THE GELLAB-II 2D GEL EXPLORATORY DATA ANALYSIS SYSTEM

Peter F. Lemkin
IPS/LMMB/NCI

July 6, 1993
*** DRAFT ***

This book describes GELLAB-II programs version *beta*-level 1.3.52 as of July 6, 1993. It was prepared using LaTeX and the index was prepared using the `makeindex` program. It was printed on a PostScript laser printer in 10 point Roman font.

Image Processing Section
Laboratory of Computational Biology, CCR
National Cancer Institute
Building 469, Room 150
Frederick, MD 21702, USA

E-mail: lemkin@ncifcrf.gov

*******PRELIMINARY INTERNAL DOCUMENT*******
DRAFT - NOT FOR CIRCULATION

Open Source License

`Written 1981-1993 by Peter F. Lemkin`
`Open2Dgel Open Source June, 2003 by Peter F. Lemkin And NCI`
`http://open2dgel.sourceforge.net`

This work was produced by Peter Lemkin of the National Cancer Institute, an agency of the United States Government. As a work of the United States Government there is no associated copyright. It will be offered as open source software under the Mozilla Public License (version 1.1) subject to the limitations noted in the accompanying LEGAL file.

LEGAL File

This document comprises the LEGAL File pursuant to Articles 3.4 and 4 of the Mozilla Public License (version 1.1) stating the intellectual property and other limitations associated with the use of Open2Dgel under this License. Dr. Peter Lemkin as an employee of The National Cancer Institute (NCI), an agency of the United States Government, is the Initial Developer of Open2Dgel (the Original Code). As such, the following limitations apply to this License:

1. This is a work of the United States Government and as such there is no copyright associated with it. Notwithstanding herein, the Government does not provide or grant any automatic license to any copyright or patent.

2. Neither the NCI, the National Institutes of Health or the United States Government endorse Open2Dgel or any product. Express or implied endorsement by these entities is prohibited. 3. The NCI, as an agency of the United States Government, is forbidden by statute to indemnify a third party. Thus, no indemnification

for any loss, claim, damage or liability is intended or provided by NCI under this License. The NCI, as an agency of the United States Government, assumes liability only to the extent provided under the federal Tort Claims Act, 28 U.S.C. 2671 et seq.

4. This License shall be construed and governed in accordance with Federal law as applied by the Federal courts in the District of Columbia. In case of conflict in law between Federal law and the laws of any other jurisdiction, Federal law as applied by the Federal courts in the District of Columbia shall prevail.

PREFACE

GELLAB-II is an integrated collection of programs for the exploratory data analysis of multiple two-dimensional (2D) electrophoretic gel images. GELLAB may be thought of as a three-dimensional spreadsheet for 2D gel analysis - a domain in which to experiment with quantitative gel data in different ways. Currently, it runs on a range of UNIX based systems from low cost computer workstations to super-computers where the same database files may be used across systems. The display of gel images is available if the UNIX workstation hardware supports a color display under X-windows. Otherwise gels and their derived images can be visualized through line graphics plots on inexpensive terminals and a laser printer.

The GELLAB paradigm is that one assembles all of the relevant quantitative and experimental information for a set of gels into a single composite gel database (DB). The investigator can then explore this data using a variety of computational techniques. The particular technique selected at any point being dependent on the "angle" at which they wish to "view" the data in hoping to "observe" relevant patterns. We will be discussing how one constructs such a composite gel DB and what "tools" are available to explore it.

This book is intended to be used primarily as a reference for the GELLAB-II system. As such, it includes a general introduction to gel analysis methods, tutorials on introductory and advanced methods for computer assisted analysis of 2D gels, and a discussion of exploratory data analysis techniques as applied to gel analysis. Many of the tutorial examples use the same data from a sample set of demonstration gels supplied with the system. This lets the user acquire some familiarity with the data - which is one of the objectives of using an exploratory data analysis system. An attempt was made to make the book sufficiently self contained such that an investigator could learn the system given minimum training and then use it later on as a reference to find answers to most of their remaining questions. For example, the discussion on search strategies is cross referenced to the tutorial examples so the investigator can see how to implement their own search strategies. It is hoped therefore that, although there is a lot of detailed material, one should be able to find their way with minimum effort. To aid in this, the book is heavily cross referenced and indexed. An extensive glossary is also available.

GELLAB-II tutorials are structured so that one can see exactly *how* to go about analyzing a set of gels using computer assisted analysis. To aid in this, a sample 12 gel adult human leukemia gel database was graciously donated by Eric Lester and is included in the GELLAB-II distribution demonstration database and is referenced extensively in the tutorial and throughout the rest of the book. You can use this data to work through the different parts of the tutorial in a coherent manner. Exploratory data analysis is something of an art-form which can proceed in different directions depending on the type and quality of the data, analysis tools and skill and interests of the investigator. Material from *2D Electophoresis gel database analysis: Aspects*

of data structures and search strategies in GELLAB [LemP83b] is included in the section on exploratory data analysis search strategies and addresses some of these issues.

GELLAB-II's main theme is in facilitating an exploratory data analysis on 2D gel data. Some knowledge of basic computer operations is required. The assumption is made that the user has some minimal knowledge of computers (such as exposure to PCs etc.) - although introductory tutorial information on UNIX is included in an Appendix to smooth gaps in this area. This will be a help but does not replace the need for some additional expertise in that area. We do not cover the production of 2D gels themselves as that is covered in a number of other references and is beyond the scope and intent of this book.

The other audience we try to serve are those individuals responsible for installing and maintaining GELLAB-II. For these readers, there are detailed instructions on setting up and maintaining the necessary file systems.

This manuscript is based on a variety of materials describing the GELLAB system. The primary source was the 1983 GELLAB-I reference manual which was brought up-to-date for the new GELLAB-II system. Unfortunately, this material did not give the background for exploratory gel analysis approaches required to fully understand and utilize GELLAB-II. To remedy this, selected material was added which was derived from some of the GELLAB-I papers discussing this exploratory data analysis concept and showing how one would analyze a set of gels. These include: Lipkin and Lemkin, *Database techniques for multiple PAGE (2D gel) analysis* [LipL80a]; Lemkin, Lipkin, and Lester *Extensions to the GELLAB 2D electrophoresis gel analysis system* [LemP82a]; and Lemkin and Lipkin *Database techniques for 2D electrophoretic gel analysis* [LemP83a]. Additional material was used from other GELLAB papers including some of the recent papers discussing GELLAB-II (*A workstation based 2D electrophoresis gel analysis system* [LemP88d], *Database and search techniques for 2D gel protein data: a comparison of paradigms for exploratory data analysis and prospects for biological modeling* [LemP89a]).

The current UNIX based GELLAB-II system was directly derived from the GELLAB-I system which had evolved over several years when it was used in a number of different biological investigations including ([LesE80], [LesE81a], [LesE81b], [LemP82a], [LesE82a], [LesE82b], [LesE83], [HowR83], [LesE84a], [LesE84b], [LemP84], [SonP85], [SonP86], [LemP89b], [StoE89], [LemP91], [RogP91], [AmbA91], [MyrJ93]). Needless to say, most of the concepts used in GELLAB are attributed to the many discussions with these collaborators and GELLAB users - especially Eric Lester, Peter Sonderegger, James Myrick and Trygve Krekling. Carl Merril and his group were instrumental in introducing me to the domain of 2D gels and posing the initial problems which got us involved. Special acknowledgement must be made to Lewis Lipkin without whose encouragement, insights and suggestions over the years GELLAB would never have been

built. Additional thanks go to Rob Ashmore who helped debug the C translation of GELLAB-II as well as reading many drafts of the book, and Wayne Main for useful suggestions in organization and style. Thanks to Kyle Upton for aiding in cleaning up the system, improving the graphical user interface and suggesting many improvements. Thanks to Yecheng Wu of CSPI has participated in the CRADA for the commercial version of GELLAB-II for Windows-NT and who has asked tough questions and suggested many improvements.

Beta version

Being at *Beta*-level version of the GELLAB-II software, we are still ironing out a few remaining bugs both in the software and in this book - hopefully these will be a minor inconvenience and not detract too much from the overall usefulness of the system. Please contact me with any new bugs you discover, comments or suggestions.

Peter F. Lemkin

Frederick, MD
July, 1993

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The 2D polyacrylamide gel electrophoresis (PAGE) technique [OFarP75] has been a rapidly developing biochemical tool, applicable to a wide variety of problems in molecular biology, basic biochemistry, genetics and clinical research ([AndN79], [AndL82], [AndL84], [CelJ89a], [NeiF89a], [KloJ89a], [SweJ89a], [CelJ92]). The 2D PAGE technique can be used to separate hundreds to several thousand polypeptide components as a matrix of spots because the variables which determine electrophoretic mobility in each of the two dimensions are effectively independent to each other. Isoelectric focusing over a pH gradient determines extent of movement in the first dimension. In the second dimension the Sodium Dodecyl Sulfate (SDS) interaction with protein fragments results in a mobility which is a function of molecular weight. The standard orientation for the isoelectric (pIe) and molecular mass (MW) axes is illustrated in Figure 1.1. A number of popular modifications and descriptions of O'Farrell's technique have been published including the Andersons' ISO-DALT system [AndL88a], Dunn's review in [DunM87], Bravo's "Guide for the Beginner" in [BraR84], Sinclair and Rickwood in [SinJ82] and Garrels [GarJ79].

what is GELLAB?

GELLAB-II is an integrated collection of computer programs for the exploratory data analysis of multiple 2D electrophoretic gel images. GELLAB may be thought of as a 3D spreadsheet with which the investigator can manipulate the data to get different "views" of proteins in the gels in trying to determine their characteristic structure. It enables correlating quantitative changes in sets of proteins with respect to experimental conditions such as gene expression, drug response, hormone regulation, development, etc. These programs were developed at the Image Processing Section (IPS) in the National Cancer Institute (NCI) starting in 1979. Currently, GELLAB runs on UNIX based systems. They may be from low cost computer workstations with color display capabilities (although the former give greatly enhanced capabilities) - to UNIX super-computers used with X-Window workstations

or X-terminals..

material covered

This book is intended to be used primarily as a reference for the GELLAB-II system. As such, it contains material which includes tutorials on introductory and advanced computer assisted analysis methods for 2D gels. It attempts to be sufficiently self contained such that an investigator could learn the system given minimum training and then use this book as a reference to find answers to most of their remaining questions.

reading further

For those interested in reading further on GELLAB, good introductory papers which describe the basic concepts of GELLAB analysis are [LipL80a] and [LesE81b] with [LemP83a] (a condensation of ([LemP81a], [LemP81b], [LemP81c]) being a more general and detailed summary. Extensions to the early system are discussed in ([LemP81d], [LemP82a], [LemP83b], [HowR83], [LemP84a], [LesE84b], [LemP89a], [LemP89b]). [LemP91], [LemP92], [LemP93]). A comparison of aspects of several 2D gel database analysis systems is given in [LemP89a]. However, much of the material contained in these papers has been incorporated into this book. Some of the major biological investigations undertaken using GELLAB include hematologic preparations ([LesE80], [LesE81a], [LesE81b], [LemP82a], [LesE82a], [LesE82b], [LesE83], [LesE84a], [LesE84b], [LemP89b]) and axonal proteins ([LemP84], [SonP85], [SonP86], [StoE89], [RogP91], [AmbA91], [MyrJ93]).

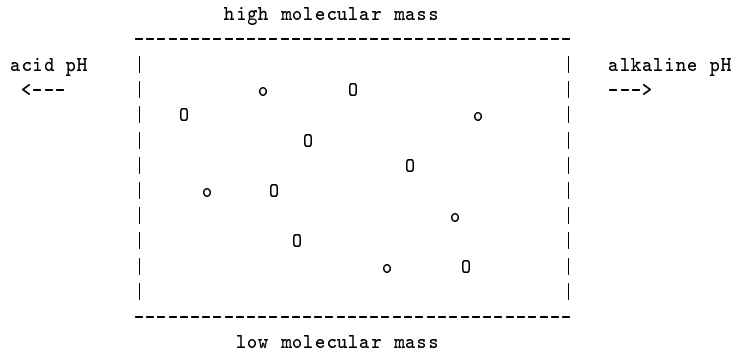


Figure 1.1. 2D PAGE gel MW/pIe axes. The standard orientation for displaying 2D PAGE gels is with the acid on the left and high molecular weight values on the top. When these values are mapped to a positive valued (x, y) coordinate system, $(0, 0)$ is the upper left hand corner and increasing values of y correspond to lower molecular weight. Increasing values of x would correspond to more alkaline proteins. Spots on the gels, indicated by the circles, are visualized using autoradiography, or stains such as Coomassie blue or the silver stain.

data reduction

Role of data reduction in 2D gel database analysis

As discussed in more detail in Section 1.3.1, page 29, data reduction of gels and their associated information is used to find quantitative and qualitative differences between samples. This can be expressed as a sequence

of steps. That is `experiment-information&gel-images` $\Longrightarrow$ `spot-lists` $\Longrightarrow$ `paired-spot-lists` $\Longrightarrow$ `composite-gel-database` $\Longrightarrow$ `significant-spots` $\Longrightarrow$ `analysis-derived-images`. Multiple 2D gel databases allow:

- Finding and clustering spots and gels which correspond to changes in experimental conditions in the gel samples.
- Visualizing where those spots are in the gels.
- Applying constraint analysis on proteins or sets of proteins. E.g. expression-profiles, finding putative precursor-product pairs, polymorphisms, etc.
- Recording and annotating results so they can be accessed and referenced as needed.
- Searching an associated gel annotation database proteins of interest for which changes have occurred in other experiments or for which information has been noted by other research groups.

A typical gel analysis may take the investigator from the wet-lab to the computer where their data is analyzed numerically and then back to the wet-lab:

WET − LAB	IMAGES	DATABASE
`sample&make-gel` $\Longrightarrow$	`gel-images` $\Longrightarrow$	`gel-database`
		$\Downarrow$
`extract&sequence-spots` $\Longleftarrow$	`synthetic-image-maps` $\Longleftarrow$	`analysis&annotate`

composite gel databases

A 2D gel experiment database is a three dimensional object which consists of a "stack" of 2D gels for that experiment. This is called a '*Composite GeL* (CGL) database.[1] Corresponding spots across gels are connected together by the computer database system such that any set of corresponding spots can be instantly retrieved. With this 3D "spreadsheet" model, each set of corresponding spots may be viewed as a distribution of polypeptide concentrations for a given polypeptide. Each component of the mixture is actually a distribution since we are dealing with real data. A component corresponds to a particular experimental condition where similar gels have experimental variance in protein concentration (σ_d^2) giving rise to the distribution.[2] We will be discussing other complications such as missing spots and split spots later. Figure 1.2.a illustrates a composite gel database. Figure 1.2.b

[1] A *Paged Composite Gel* (PCG) database is a disk based implementation of this same type of data.

[2] The assumption of a unimodal Gaussian distribution giving rise to the sample variance (σ_d^2) is not necessarily valid since what you *think* is a single sample population may in fact be multimodal as indicated by manual observation of the protein concentration distribution [LemP89a]. Various clustering techniques to be discussed can detect these multimodal distributions.

shows the basis of using mean spot positions for estimating a *canonical* spot. The canonical spot can then be used to estimate the position of spots missing from some of the other gels. The Rmap on the cover of the book is a lymphocyte autoradiograph of an acute myelocytic leukemia patient showing some putative myeloid markers. Figure 3.20, page 393 shows a derived Rmap gel image with selected Rspots labeled on a copy of one of the gels from a composite gel DB. Figure 3.21, page 399 shows a derived mosaic gel image with subregions from a set of gels which surround the selected Rspot from a composite gel DB.

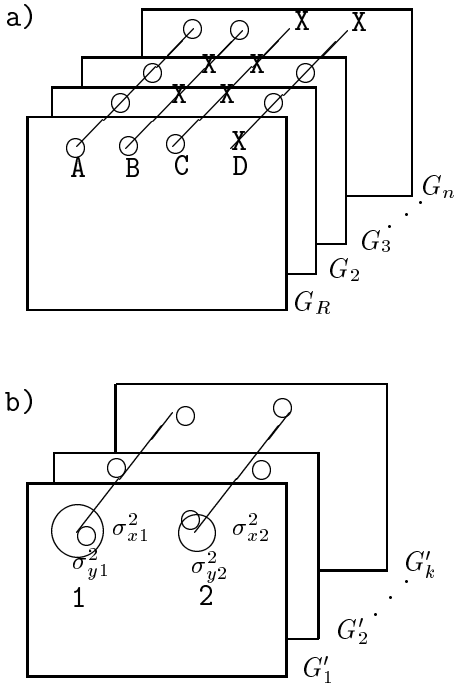


Figure 1.2. 3D composite gel database model. **a**) Illustrates a composite gel database. Corresponding paired spots (circles) are denoted by diagonal lines drawn through them. Such sets of corresponding spots are called *Rspot* sets. One of the gels is selected to be a reference or *Rgel*, denoted G_R. The circle means the spot is present and the `X` means that it is missing in that gel. Spot A occurs in all n gels. Spot B occurs in the Rgel and in one other gel. Spot C is only in the Rgel. Spot D is not present in the Rgel but is in most of the other gels. Spots A, B, and C are in the un-extended Rspot database (since they occur in the Rgel) while spot D is in the *eRspot* part of the database (since it does *not* occur in the Rgel). **b**) Shows the basis of using mean spot positions for estimating *canonical* spots for a subset of k gels from the n gel database. The canonical spot can then be used to estimate the position of spots missing from some of the other gels. The mean and variance (σ_x^2, σ_y^2) of Rspot positions across a set of gels is mapped to the coordinate system of the Rgel. When that has been done, the set of gels of the same experimental class can be replaced by a single averaged gel called the *Cgel'* (the estimate of the canonical gel). The mean displacement vector of a canonical spot from its associated landmark spot (in any gel under discussion) is used to extrapolate the position in gels where the expected canonical spot is missing.

Brief status of GELLAB-II

- The set of programs are in the *Beta*-level test version.
- GELLAB-II runs under UNIX on SUN3 and SUN4 computers (currently SUNOS-4.1.2 UNIX) with X-Windows version X11R4 or later.
- The gray scale display of gel images is available if the workstation hardware supports a color display under the X-Windows System. Image and graphic plots output can be printed to a PostScript laser printer.
- Database analyses are controlled by graphical user interface interactive menus, terminal keyboard commands or batch commands.
- Database command history (journaling) is saved and may be used to replay of previous commands.
- Much of the process of constructing the composite gel database after the initial data acquisition phase is automated by the running of batch scripts in both UNIX and in the databse program itself.

1.1 How to avoid reading this book

learning by example

Many people learn best by studying examples rather than reading a set of rules which are difficult to remember when it comes time to apply them. This book is laid out with abundant examples in the tutorials in (Chapter 2) as well as in each programs' description (Chapter 3). Additional examples are found in the sample GELLAB-II batch scripts (generated by **makjob** cf. Section 3.9, page 369). Some demonstration batch scripts are also discussed in Section 2.1, page 74. An index, page 635) and glossary, page 569 are also available.

However, before you plunge into the examples, **IT IS RECOMMENDED** that you skim this (**Introduction**) Chapter especially Sections 1.5 and 1.6, and the **Tutorial** Chapter 2.

1.2 How to read this book

reading tools

If you decide to bite the bullet and read through the book, then you might review the following suggested reading list. The **Glossary** page 569 and **Index** page 635 are available for finding definitions of gel analysis terminology which are used in gel analysis and in GELLAB-II in particular (Figures and Tables are listed in the **Index**). The book is heavily cross referenced. You are not meant to flip to every cross reference listed! To help guide you in deciding whether to do so, the Section

number is listed along with the page number in those cases where the cross reference is *only* a suggestion. After you become familiar with the organization of the book, you can use your knowledge of what a Section is about to decide whether to flip to the suggested page.

Suggested reading

This book contains detailed descriptions and examples of the GELLAB-II programs (Chapter 3, page 151). These descriptions are intended to be used primarily as a reference. *details*

Section 1.3, page 24 discusses different types of biological systems and the type of corresponding changes one might expect between gels. Section 1.4, page 40 describes the basic steps in performing a 2D gel analysis. *what is gel analysis?*

Table 1.1, page 49 lists the GELLAB-II programs and specifies which are currently operational. Section 1.5.1, page 49 gives a short synopsis of each GELLAB-II program. Section 1.6, page 54 describes how GELLAB is used in the UNIX environment. For those interested in "how long does it take", in Section 1.7, page 66 we discuss both computational and human interactive times required for an analysis. Section 1.8, page 69 lists some of the major caveats for GELLAB-II. Appendix F, page 607 gives a history of GELLAB-II and its derivation from GELLAB-I. *using GELLAB-II*

When manipulating gel data, we need to be able to visualize the original images as well as derived images of various types. Chapter 4, page 473 discusses the details of the **Xpix** program which is the X-Windows gel image display interface to GELLAB-II. Appendix D, page 603 describes how to start the X-Windows system, which is required by **Xpix**, on your machine. Many of the GELLAB-II programs use this interface. *viewing gels*

Chapter 2, page 71 is a tutorial on using GELLAB-II. It goes over the basic steps in analyzing a set of gels. Section 2.2, page 93 has detailed examples of running individual GELLAB-II programs. Section 2.3, page 105 has tutorial examples for running the **cgelp2** composite gel database program on a 12 gel experiment taken from our adult human leukemia lymphocyte database. Section 2.1, page 74 has two UNIX demonstration scripts used to first compute a series of demonstration images and then to perform a "slide" show. Additional UNIX batch scripts may be found in the examples generated by the **makjob** program Section 3.9, page 369. *tutorial*

Chapter 5, page 501 introduces the concept of exploratory data analysis methods. Section 5.1 page 503 discusses search strategies for searching the composite gel database when doing an exploratory data analysis. It is keyed to the tutorial examples in Chapter 2. Section 5.2, page 533 suggests some rules for deciding courses of action needed during the exploratory data analysis. *exploratory data analysis*

Chapter 6, page 539 discusses requirements for, installing, testing, and maintaining the GELLAB-II system. Appendix A, page 569 is a glossary of terms used

other "tools"

in GELLAB-II. Appendix C, page 593 gives a quick introduction to some of the UNIX commands. Appendix G, page 611 details some of the algorithms used for gel image segmentation into spot lists, spot list comparison and construction of the composite gel databases.

Notation

The following type fonts are used to clarify notation used in this book: GELLAB programs are indicated in lower-case bold, (e.g. **cgelp2**); specific file names are in lower case `teletype font`, (e.g. `gel.rc`); examples of programs from terminal sessions are in a smaller `footnotesize teletype font`, (e.g. `cgelp2 -d ts3pcg.pcg`); generic file names are in *italics* (for example picture file name extensions, *.ppx*), as are terms used in GELLAB-II when they are defined and the names of UNIX programs. The names of commands and switches used by the **cgelp2** interpreter are given in upper case `TELETYPE FONT`, (e.g. `INQUIRE`). Note data used in examples is derived from actual databases but may have been modified to illustrate a point or protect patient information. UNIX programs are often indicated by the standard UNIX notation of *cmd(n)* where *(n)* is the UNIX manual reference (e.g. *more(1)*). Interactive menus are referred to in the text with a double box as in `LANDMARK` while menu selections are referred to by a single box as in `Zoom Gel 2X`. *Margin notes* are used to help indicate particular subtopics. Although octal numbers are not often used in GELLAB-II, when they are, they are expressed with a leading 0 as with UNIX. For example 0177 is octal and 128 is decimal.

margin notes

1.3 Introduction to Gel Analysis

This section introduces the concept of computer aided analyses of two-dimensional electrophoretic patterns. Some of the following material has been taken from [LemP83a] and [LemP89a]. We will first discuss some of the characteristics of 2D gels, types of biological questions which can be asked using 2D gels to help get answers, and how and when this is effective. The GELLAB paradigm for multiple gel analysis is then introduced as a basis for understanding the rest of this book.

2D gel preparations

As mentioned in the introduction, the 2D polyacrylamide gel electrophoresis technique has been a rapidly developing biochemical tool since its introduction by O'Farrell [OFarP75]. It is applicable to a wide variety of problems in molecular biology, basic biochemistry, genetics and clinical research ([AndN79], [CelJ89a], [NeiF89a], [KloJ89a], [SweJ89a]). Because of the apparent independence of the two dimensions, isoelectric pH gradient and apparent molecular mass, it can be used to separate several thousand polypeptide components.[3] Isoelectric focusing

[3] Young [YouDA84], [LevR90] has reported over 5000 spots in a single giant gel and over 2000 spots can be expected with regular size gels under optimal conditions [D. Hochstrasser, personnal communication].

over a pH gradient determines extent of movement in the first, while in the second dimension sodium dodecyl sulfate (SDS) interaction with protein fragments results in a mobility which is a function of molecular mass.

Biological processes

Why do we need 2D gels and computer generated 2D gel protein databases? In short, because *major biologic processes* (growth, differentiation, malignancy, etc.) *are indeterminate*, i.e. too complex for complete determinate analysis of their component parts and explanations of all of their interrelationships and functions. Biological, and particularly eucaryotic systems, consist of a vast number of parts, of which genes and gene products are key. Most estimates suggest 30-50,000 structural genes coding for proteins in eucaryotes, of which 5-10,000 are expressed at some level in a given eucaryotic cell [AndL84]. Although with post modifications, this number of possible polypeptides will be increased. While current 1D gels usually fail to deal with such numbers, the 2D gel technique is theoretically capable of such resolution and sensitivity [TayJ83], a point confirmed by recent practical advances ([YouDA84], [KloJ84], [HocD88]). Each of these protein parts in a cell has interactions with other parts and the whole system is too complex for total description. Thus understanding these biologic systems is analogous to understanding ocean currents, the weather or macroeconomics. For all such systems, understanding begins with classifying as many component parts as possible, describing their interactions, and ultimately developing *idealized models of the system* which help serve as explanations of our understanding. The economy inherent in biological processes suggests that *while many proteins are essential for the life of the cell*, no *one* protein or metabolic pathway is of overriding importance - which is not to say that some pathways are not critical. Rather, each has its own role in concert with all the rest, and all must ultimately be included in our understanding.

how many polypeptides?

model of a cell

Thus the attraction of the 2D gel approach is its capacity to lay out in a single image, theoretically at least, a complete display of the genetic expression occurring in a population of cells. In a single "snapshot" one has measured the qualitative and quantitative setting of the genome - how the DNA is 'tuned'. By cleverly comparing the data inherent in 2D gels from different but related samples, we can begin to discern the organization of genetic expression - which genes are expressed coordinately during a biologic process and the sequence of changes in their expression. Two problems arise however:

snapshot

1. The large quantity and complexity of the data "it's too complicated, I want a simple problem!", and

2. The lack of a relationship between the 2D gel data and traditional biochemistry "What are all those spots anyway?".

The solution to the first is, obviously, creation and analysis of computerized databases. The second, is not so obvious and depends on how one designs experiments to use this information.

What can you do with "all those spots"?

Computerized gel databases permit tracking and correlating a number of these relationships.

1. Identify *individual proteins* whose presence or absence (*qualitative change*) serve as a *marker for a class of samples* (e.g. a class of leukemias or cell lines transformed by a particular oncogenic virus).

2. Identify consistent *quantitative changes* in protein expression (relative spot size) which can serve as markers for a class of samples.

3. *Compare groups of proteins* identified in 1) and 2) using *set operations* after various classes of samples have been compared and marker proteins identified. Thus proteins which are *most consistently* different between classes can be identified as markers for these differences. Such proteins, by virtue of their consistency, are more likely to represent *key gene products* which underlie the biologic differences between sample classes.

4. Ultimately, a complete annotated *catalog of proteins* observed in all the gels from each sample in various databases can be constructed. Creation of such large databases, particularly those containing *summaries of the analytic comparisons* and set operations noted above, provides a powerful inferential tool for comparison of mechanisms operative in diverse biologic systems.

5. Alternatively, gel patterns may be analyzed in a global fashion, in order to show the *relatedness of samples* to each other based upon a *summary of their features*. Various cluster analysis algorithms and other approaches will generate such 'summary statistics' and patterns of relatedness among samples. Such analyses also of course lead back to the individual marker proteins since a natural outcome is to ask 'which proteins contribute the most to the final arrangement of sample relatedness'.

Relating gel data to other biologic data

Two dimensional gels have been used extensively for finding qualitative and quantitative changes in many different biological domains. These results are reported in the literature in both domain specific journals, proceedings (such as those of the International Electrophoresis Society or American Electrophoresis Society), books and the electrophoresis literature. In particular, the journal *Electrophoresis*

has published many of these results. In addition, the journal *Clinical Chemistry* has published special issues where this type of data has been reported.

The best link, at present, between 2D gels and the rest of biology appears to be the generation of partial amino acid sequences from proteins isolated from 2D gels ([HooL87], [DelC88] and others). [EckC88] suggests computing the amino acid compositions which can be used to search protein databases, but which require 1/4 the number of gels since less material is required than with sequencing. Thus key proteins identified by comparing sets of proteins defined by statistical analysis of 2D gel databases can be eluted from the gel and partially sequenced and compared with the extant gene and protein sequence databases ([EckC88], [KenT88], [SweJ89]) to obtain clues to their function. A s pecial issue of *Electrophoresis* (11:7, July 1990) contains a paper symposium devolted to blotting and sequencing proteins from 2D PAGE gels. This includes a number of technique papers ([AebR90], [BauG90], [TemP90], [EckC90], [ChoT90], [BerT90], [HirH90], [JunP90], ChaL90]. A recent 2D gel database issue *Electrophoresis* (13:12, December 1992, lists several databases [VanB92], [HocD92], [GioD92], [CelJ92], [RasM92], [YunM92]).

sequencing

This approach also allows generation of nucleic acid probes which can be used to isolate, sequence and aid in defining the regulation of the gene coding for the original protein. By following this path for multiple members of a set of co-regulated proteins defined with a 2D gel database for a variety of conditions, we may be able to elucidate and interelate the various regulatory mechanisms which are operative. Thus our 2D gel analyses may be used to lead our efforts at DNA sequencing in a rational way. The key point here is the need to relate expression (the *evidence* of regulation) and coordination of expression with the underlying genetic structure. It is not enough to simply sequence the human genome. We must understand its order, regulation, and coordination - a process in which 2D gel database analysis will play an important role.

Automating gel analysis

Because of the additional orthogonal value of the second dimension, there is a perhaps two decimal orders of magnitude increase over 1D techniques in the number of polypeptide fragments ("spots") detectable in a mixture [OFarP75]. This is a major cause for efforts at computerized analysis. Some of the systems besides GELLAB which can do this include TYCHO/KEPLER [TayJ82], ELSIE-IV ([MilM84], [MilM86], [OlsA88]), MELANIE ([FunM87], [AppR87]), QUEST [GarJ84], PDQUEST [BloS86], and HERMeS ([VinP87], [TarP87], [TarP89], [VinP89]) as well as other systems including some of these functions ([KuiR86], [SkoM86a], [SkoM86b], [RidG84], [SmiK86], [ManR86], [TeiD89]). A number of these systems are reviewed by Dunn and Burghes [DunM83] and by this author in [LemP89a]. These gel analysis systems are all really performing an *exploratory data analysis.*

gel analysis systems

exploratory data analysis

Fisher et al. [FisD86] suggest a definition that is appropriate for 2D gel databases: "Exploratory data analysis can be characterized as a search for *regularity* or *structure* among objects in an environment, and the subsequent *interpretation* of discovered regularity." One must be able to both reduce the dimensionality of the data and be able to view in it different ways to discover possible "regularities". GELLAB-II allows this by giving the investigator the ability to easily change his view of the same data. Section 5, page 501 goes into more detail on GELLAB methods for performing an exploratory data analysis. As a result of these automation attempts, there has been an increase in complexity of intermediate analysis results. These in turn would strain the limits of unaided human analytical ability.

The need for analytic assistance is further increased by the added complication of non-linear spatial warping of corresponding moieties in comparable gels which are artifacts of the technique. A 2D PAGE electrophoretic gel is a complex of distinct polypeptides, each one of which is characterized by position relative to other polypeptide *spots* and density. However, unlike a geographic map, proximity of polypeptides on a gel is usually not an indication of related genesis or biological function. Nonetheless the large number of discrete spots in a gel and the similarity that is preserved among gels from a similar source allows one to track many proteins through the effects of experimental variables.

Gel complexity

experiment permuatations

We are progressively more concerned with the generation of data structures, strategies and tactics for their employment in the analyses of *sets* of gels. Such comparisons, both qualitative and quantitative, among multiple gels might reflect, for example, successive values of a dose or time variable in an experiment or the clinical course of a patient.

why automate?

In dealing with this material, human factor considerations place a practical limit on the number of spots for which manual density information can be obtained. Manual techniques such as optical flicker or dual-color comparison between two local regions on separate gels is useful for local alignment, especially in cases with obvious spot differences [LemP79a]. However, this method forces the user to deal with the gels as a sequence of pairs. This makes the process time consuming and difficult for the observer to visualize a pattern directly over a set of gels. Such manual comparison methods are probably capable of supporting a complete search for all major polypeptide differences, but the bookkeeping needed to identify the same spot in several gels makes computer aid attractive. Beyond some relatively small number of spots, some computer aid in matching, "remembering" and retrieving images of preserved spot correspondences is seen as indispensable in thoroughly analogous sets of gels. An added benefit of this is that after the spots have been isolated, located and tagged, the machine can use this information to produce a variety of pictorial, diagrammatic or numerical representations of patterns difficult

to grasp when attention is focused on small regions. Final output of GELLAB-II includes labeled gel image maps and mosaics where statistically interesting spots have been marked as well as numeric spot data lists to and graphics support these findings.

steps in analysis

Figure 1.3 on page 43, illustrates the major steps performed in the GELLAB-II 2D gel analysis procedure. Gels are first accessioned (assigned a unique sample number) and related experiment information associated with the sample is entered at this time into the system. The gel images are then acquired (digitized and stored on disk files). Then spots are segmented (located and quantitated) in each gel followed by pairing spots between each gel and a standard gel using a small set of manually defined landmarks. A multiple spot database (DB) is then constructed and analyzed. Final output of such a system takes several forms. This includes labeled gel image maps (superimposed on the original gel images) where statistically interesting spots have been marked as well as numeric spot data to support these findings. Various 1-, 2-, and 3-D views of spot database features may be presented both graphicly on interactive displays and numerically in tabular form.

In Section 1.4, page 40, we first briefly touch on gel image acquisition and digitization. Then image segmentation for spot extraction is discussed in enough detail so as to convey how spots are quantitated. A discussion of the nature of the gels themselves is limited to those aspects of their generation, staining, and other modes of spot detection, which bear directly on the computerized procedures for their analysis. This is followed by a discussion of spot pairing using landmarks to locally align subregions. Finally, we discuss construction of multiple gel databases using the spot pairing algorithm's output - sets of paired spots - and the subsequent types of searches and analyses which can then be performed. But before we get into the details of multiple gel analysis, we need to discuss some of the characteristics of 2D gels themselves. This will give us a better insight into what we can and should analyze.

1.3.1 Gel characteristics

factors affecting analysis

For the most part, a spot's resultant geometric position in a gel bears no relation to function or the origin of the protein it represents. Closely related polypeptides may be separated by considerable distances while functionally unrelated materials could be distributed in close proximity. In contrast to more conventional images such as microscopic fields or X-rays, the image "structure" (i.e. local adjacencies, inclusions, etc) in gels provides little information to facilitate the analysis. Individual spots unless contaminated by artifacts or overlapped by other spots are much simpler images than, say, the image of a cell in a blood smear. Thus once a spot has been isolated, its analysis and characterization as an *individual entity* at least in a single gel, is relatively simple.

gel distortions

Non-congruence is a more serious difficulty (i.e. the lack of point to point reproducibility of gels). It occurs in gels derived from the same sample and from a single run on the same apparatus. This is due to a large number of preparative factors including local temperature variations, local heterogeneities in polyacrylamide texture and/or local concentration, heterogeneities of ampholine concentration, etc. All of these variables and perhaps others less understood to reduce the reproducibility of mobility of polypeptide fragments in one or another dimension. This net result is a set of gels which are *not congruent* but which are affine. [*Affine* denotes a linear coordinate transformation in n-dimensions which permits translation, rotation and magnification.] In other words, comparable spots within a set of gels have corresponding neighbors but are not necessarily located at exactly the same distances from these neighbors in any specific instance. The set of gels show a local superimposibility, which is maintained for surrounds of areas of varying extent. Thus it is this absence of simple direct correspondence coupled with the large numbers of spots in a set of gels that makes some automated assistance a necessity.

1.3.2 Classes of problems

It is necessary when viewing gels to carefully consider the kinds of questions that biological problems pose. The questions assist in determining the nature, depth and range of the analyses to be performed. These questions have been divided into five increasingly complex areas.

isolated qualitative changes

1. Is only one or a very few different spots present in one gel and not in its experimental pair? The paradigmatic biologic systems which pose such questions are those in areas such as bacterial genetics, where both the homogeneity of and the specificity of the product generating cell line are very high. Here the gels are used as detectors and serve simply to confirm or deny the existence of a polypeptide fragment. Simple flicker analysis may be all that is required under favorable conditions while densitometry for this situation is a secondary consideration if one at all. A case in point is a single gene difference in an E.Coli mutant [LemP79a].

quantitative changes

2. Are there changes in any of several spots in a cell line as a function of time? These variations are often in polypeptide quantity so that densitometry is required. Moreover, the complexity of the analysis is increased so that n gels rather than a pair of gels must often be compared. The answer to such questions as this requires spot data structures and database management software that are both significantly larger and significantly more complex than those required for the answer to the first question.

increasing complexity ⇓ "sets" time course

3. Are there changes in several spots resulting from an applied stimulus? The less known about the outcome, i.e., the more exploratory the search, the more extensive and complex must be the gel analysis. When the cell line is only apparently homogeneous (as was the case for PHA stimulated lymphocytes [LipL80a], [LesE80])

and where the effect of the stimulus is both complex and a function of time, the gel analyses become correspondingly more extensive and complex. In such situations, where many new spots may result, there seems no alternative to automatic spot pairing using a computer.

4. Is there a "finger print" of morphologically homogeneous but biologically and functionally different cell groups as for example differences among various lymphoblastic tumors. Here, especially if stimuli are required to elicit differences, the number of gels grows to an m (# of classes) times n (# of temporal samples) times p (# of levels of stimuli) number of comparisons. This assumes minimum problems in the reproducibility of the gels whereas often multiple gels of the same sample are run. Particular interest must be focused not only on differences but on subgroup similarities as might seem to be indicated by results of applying additional clustering techniques to these intermediate results.

5. Are known polypeptides present in normal or abnormal quantities in a body fluid? Here are the quintessential problems of clinical chemistry but multiplied by the large number of spots present in the gel. At first, the answer to questions of this type might seem simpler than those above. The comparison of a single gel's contents to some internal or external standard certainly involves future developments in the area which has been called by Anderson "molecular anatomy" [AndN79]. Because of the need for extensive bookkeeping in multiple gel analysis, it is likely that some of the data structures we present here will be an aid in this development.

"standards"

Gels may be thought of as complex objects similar to a geographic map with individual polypeptides appearing in distinct local morphologic regions. Spot are not however in any way certain reference points unlike the geographic map. Adjacency of morphologic polypeptides in the gel is no particular indication of related genesis or biological function. However, characteristic patterns such as those obtained with carbamylation and other biochemical treatments do have such a basis. In general, comparing biological specimens by comparing their corresponding gel maps is one means of determining major protein differences - although this is tedious when done manually. Given a number of gels, polypeptide concentration values may be modeled as a density distribution for each set of corresponding spots and then statistical analyses performed on these distributions.

local morphologic regions

1.3.3 Complicating issues in using gels

In [LipL80a], we discussed issues which complicate the analysis of 2D gels. "The major sources of difficulty lie in four major areas: (a) variability in location of corresponding spots, (b) variability in spot intensity, (c) variability in spot detectability, and (d) the microstructure of the spot.

Photographs of star fields taken through the same telescope under comparable conditions yield quite reproducible images. Indeed, they approach geometric con-

gruence, so much so that 'flicker' comparison – the rapid switching between two images in the same area but taken at different times – may be used to compare sequential images of a star field to detect the motions of planets or comets.

Sources of error

Unfortunately, between-gel variability is such as to preclude global application of such 'flicker' comparison techniques. Gels that are otherwise comparable, even those produced on the same apparatus at the same time, do not exhibit point to point correspondence of spot positions. Many heterogeneities [HurP78], ranging from ampholyte distribution to local temperature variation during electrophoresis, contribute to this kind of variance.

gel reproducibility

Nonetheless, we can properly speak of gel reproducibility and comparability. Corresponding gels 'look alike,' and more particularly, any given local corresponding regions of these corresponding gels, 'look' even more alike. As a first approximation, we may consider gels to be dissectable into corresponding local regions of greater similarity than the gels as a whole. We denote these *local morphologic regions*. It is this that constitutes the effective reproducibility of gels noted earlier.

If the regions are small enough, it is reasonable to assume that they may be treated as if linear point-by-point correspondence is valid. Practically, using 'flicker' techniques [LemP79a] and **Xpix** or **landmark** in GELLAB-II - successive local alignments of prominent spots common to both gels - may result in an effective set of regions that show a piecewise, approximately linear correspondence. This provides a basis for gel comparisons in the absence of the ideal overall linear correspondence. This point is the basis for the role of landmark spots, the generation of landmark sets and the gel-comparison program.

The alternative to this pragmatic piecewise treatment would be a global transformation of the entire image, to bring it into linear conformity with another gel or a 'model' characteristic of the particular source material using affine transformations. However, since spots present themselves differently, simple image sutraction of transformed gels does not confirm the quantitataive equivalence of two corresponding spots.

protein expression variability

The second major problem is variability in spot density. Densitometry of autoradiographic images offers the most direct route for quantitative comparisons, provided time and duration of the labeling pulse and other labeling parameters are strictly comparable. Successive autoradiographs of the same labeled protein fragments do exhibit the ideal linear pointwise correspondence of spot positions among gels. This does not, however, imply equal densities in a spot pair. Depending on exposure duration and other photographic variables, densities of corresponding spots may vary widely, even to the extremes of the disappearance of saturation or disappearance of the spot. Moreover, to take advantage of the full range of autoradiographic detectability (which exceeds the dynamic range of most film and, even

more so, of optical Vidicon, Photodiode or CCD detectors), multiple exposures of various durations are necessary. Because of the non-linear saturation effects of these detectors, darker spots have much more error than lighter spots. This may also result in saturation of some spots and non-detection of others. Some laser scanners offer linearity over a much higher dynamic range of 0 to 4.0 OD. Used with wide range radiographic films which offer a more linear response over a similar range, there is less need for using and merging multiple gel exposures. There are advanced scanner technologies which count radiation directly and thus get around the saturation problem but most suffer from poor spatial resolution and the ability to only count radiation directly or indirectly through special plates which are "charged" by the gels and read by a special laser scanner.

detector sensors

Instances of spots with densities outside the gamma (detector response characteristic) range of the particular film/detector combination need to be programaticly excluded from densitometry to prevent invalid, although formally correct, quantitation. To deal with these variations in density of the same spots and artifacts that look like spots, the utility or rather the necessity, of a database management system that maintains the identity of spots in multiple autoradiographic images of the same gel is obvious.

Methods of spot detection include chemical stains and various isotopic labels. Coomassie Blue is still the protein stain of choice, despite the fact that its reaction with protein does not appear to be strictly stochiometric. The silver stain [MerC79], which is not stochiometric - proteins stain selectively - has sensitivity which greatly exceeds that of autoradiography. Merril has reviewed silver stains in [MerC87]. Molecular Dynamics produces an interference optics scanner device calling their process Autophoresis.

detection method

Transmission densitometric procedures on original gels, no matter what the stain or other protein detector, appears to be more difficult than quantitative autoradiography. Predominant among these difficulties must be the variable background and scatter produced by the polyacrylamide gel itself. Incomplete differentiation of regressive stains, especially in the presence of concentrated colloidal materials, makes for marked reduction of dynamic range and increased uncertainty of measurement. These difficulties of within-gel comparisons are multiplied for between-gel ones.

The fine structure of the apparently structureless spot constitutes the last major problem discussed here. From the viewpoint of classical image processing, a 2D gel autoradiograph is one of the simplest of images. The image primitives that comprise it all belong to the same class, i.e., spots. These may vary in detailed shape, but all appear to be without internal structure. The gel does not map an extrapictorial structure in the same sense that, for example, a roentgenogram represents a two-dimensional projection of three-dimensional anatomic features. Problems of substructural insidedness or outsidedness do not apply.

The apparent simplicity of a 2D gel does not bear up, however, under the detailed

the spot model and noise ...

examination necessary before computer analysis. For example, although an isolated spot does not have explicit internal structure, whole regions (i.e., the alkaline range) may show artifactual confluence. Apparently separate spots in reality may overlap (which is obvious when looking at the same material with D. A. Young's 'giant gels' [YouDA84], [LevR90]). In practice, spots are not symmetric and they frequently show tail-like extensions, especially when present in relatively high concentrations, particularly in the isoelectric dimension. Overload of a spot characteristically results in increased asymmetry in the dimension related to relative molecular mass. Thus initial assumptions as to 2D gaussian distribution within spots are useful only as first approximations. The range of spot densities may well extend beyond both extremes of the governing detection devices' gamma curve. The local, idiosyncratic distortions occurring in individual gels, which make direct spot comparison so difficult, have already been mentioned.

Summary of sources of error in gel analysis

To summarize the above discussion, there are a number of sources of error which can be attributed to different parts of the gel analysis. Some of these include:

1. Type of sample material, sample preparation and its purity, and degree of radio-labeling, will affect both number and quality of artifacts and gel background.

2. Gel preparation, including control of reagent purity, cooling, sample-loading, voltage/current regulation and autoradiography/staining.

3. Autoradiograph scanner resolution, linearity and noise. If a spot is saturated in one exposure of a gel but not in another exposure of the same gel, then it is desirable to use the second measurement. Currently, GELLAB-II has no direct way of merging different exposures of the same gel. Currently, both exposures reside in the same database and the "prefilter" is used to eliminate the saturated data.

4. Gel image segmentation failures. These may be due to noise in overlapping spots, spot fragmentation due to resolution - both spatial and signal-to-noise in the gel image, etc.

5. Gel spot list pairing failures. Some of these may be due to greatly divergent numbers of spots/gel, greater distortion between gels, landmark set selection, etc.

6. Gel database search failures. These may be due to faulty density normalization, "power" of the test, prefilter selection or insufficient data to make results significant.

1.3.4 Multiple 2D Gel Analysis

Earlier we discussed the need for computer support of 2D gel electrophoresis analysis. Such support along largely data structural lines has been shown to be essential. We have treated the problems of spot extraction and quantitation, and pairwise spot comparison and in the process have indicated that experiments involving time or dose variables require comparisons of spots from multiple gels. We now deal with multiple gel comparisons, the most powerful and demanding mode of application of 2D electrophoresis to biological and clinical investigation and describe a computer program, **cgelp2** for multiple gel analysis. When comparing corresponding spots among a number of gels, the pairing performed by pairing two gels is a prerequisite for the analysis of multiple gels where one treats the values of particular spots within a set of gels.

Associated spots and their characteristics can be partitioned by one criterion and then repartitioned as one attempts to "see" the data from several perspectives. From our early efforts at gel analysis using flicker analysis, it became evident that what was required was a system which could automatically find and measure all (or most) spots in a gel. Spots from two or more gels should be comparable which implies that the program needs to be able to partition and to concatenate lists of spots acquired at different times and from different gels. Without checking *all* or at least most of the spots in the set of two or more gels, no complete statement of the types of spot differences can otherwise be made. These constraints imply both a gel pairing program and a spot data management system.

multiple analysis constraints

The canonical gel - the "standard" gel

In a given gel, the majority (if not all) spots, once isolated, can be characterized by (to the first approximation) a triple, comprising x and y position (centroid) and an adjusted integrated density value D_{norm} proportional to polypeptide concentration. Among gels, the idiosyncratic variations of these triples due to variation in gel and sample preparation, detection etc. confound what are the "real" variations produced in the biological/clinical system by time, dose, clinical state, etc. We propose the concept of a *canonical gel* or *Cgel* composed of canonical spots, which is valid for the domain of a given experiment or a defined clinical situation. Such a Cgel provides information characterizing position and density distributions for all spots over all gels in the set (see Figure 1.2, page 21). Further, it excludes the data idiosyncratic to detection and preparative conditions unrelated to the biologic issue. A necessary but not sufficient condition for construction of a Cgel is the spot wise comparison of each gel with every other gel in the set, with the condition that comparison be commutative. In other words, if there are n gels in the set, to construct a canonical gel requires C_2^n comparisons times the number of spots. Since each element of the Cgel is a function expressing the variation of the spot descriptor triple (x, y, D_{norm}) as a function of the biomedical variable, it is not easily constructed.

the canonical gel

the representative gel

Though not easily realized in practice, the Cgel provides a model reference object against which we may weigh a pragmatic substitute, the *representative or Rgel*. In practice, we can construct an estimate of the Cgel which we call the *Cgel′*. As more is known about the Cgel, the estimate of the Cgel′ can improve.

The Rgel, in contrast to the Cgel′, is derived from a single pictorial object. It is a real gel chosen by the investigator from a set of gels representing a given experiment. Rgel selection is described below, but may be considered to be what it is named, a representative (by experimenter criteria) gel which is believed to contain most if not all spots encountered in any of the members of the set of gels. It is not necessarily an experimental control gel, but its selection by the biologist certainly reflects his knowledge of the experiment and of the resulting individual gels that constitute the set.

the Rspot set

The Rgel is used as the basis against which other gels in the set are compared with respect to spot position. Each spot in the Rgel is the index to a *Rspot set*. A Rspot set is that set of spots, with at most one from each gel in the set of gels, which corresponds to a given spot in the Rgel. The *list* of Rspot sets under ideal conditions includes all spots in all gels. Until biochemistry can provide essentially noise free gels and extensions to the analysis including methods for handling missing or very noisy spots in the Rgel, such a complete and ideal accounting is simply not attainable.

General system of analysis

composite gel database

The design philosophy underlying the part of the GELLAB-II system that deals with multiple gels is the interactive and flexible manipulation of spot data organized by corresponding-spot association. Paired spots and their locations and densities are recorded in a composite gel database denoted the CGL, which can be searched in a variety of ways. Various representations, numeric, diagrammatic, pictorial, textual or tabular, of this database or of its derivatives can be rapidly displayed in order that the researcher may quickly grasp patterns and implications when doing exploratory data analysis. Hypothesis verification is performed by interactive partitioning and testing new representations of segments of the data.

corresponding spots

Fundamental to our system of analysis of multiple gels is the concept of canonical polypeptides which give rise to sets of corresponding spots across gels. A corresponding set of polypeptides is one in which each member arises from a common group of biologic processes. The quantitative expression of such production may be muted or exaggerated under varying experimental conditions. But in each gel where it is detected, *the spot denoting the canonical polypeptide occupies the same relative position in the local gel morphology.*

Definitions: list of spots, Rspot set, and list of Rspots

We now formally define these often confused constructs: the list of spots in a single gel, the Rspot set and the list of Rspots in a composite gel database. They are used throughout in our discussions on GELLAB.

List of spots in a gel - These are the list of spots present in a *single* gel. This list stands alone and has no explicit relationship to any other gel - that is the job of the Rspot database system.

A Rspot set - This is a set of *corresponding* spots of a particular putative canonical polypeptide, having at most one member from each gel (but definitely including a particular member of the list of Rspots), corresponding to a given spot in the Rgel. The Rspot set may be regarded as a vector, each element of which is taken from a single plane of the three-dimensional (3D) stack of gels. (Using Composite Pair spots which are composed of a group of spots, this rule of a single spot may be extended.)

The list of Rspots sets - All the distinguishable Rspot sets in the Rgel, taken together, constitute a list of Rspots - the composite gel database; i.e., all the members of the list of Rspots are to be found in the Rgel and all the spots visible in the Rgel are, at least potentially, members of this list of so called Rspots. Spots that compose the list of Rspots are to be distinguished from the elements of a particular Rspot set.

list of Rspots vs. Rspot set

The linkage and reciprocal dependency between the list of Rspots and a Rspot set is this: 1) A Rspot set member (i.e., a single spot in the Rgel) must correspond to at least one other spot in the remaining $(n-1)$ gels for it to be recorded by the **cgelp2** database program as a spot pair, and 2) a set of corresponding spots will not be recorded as a normal Rspot set if it does not have a representative in the Rgel. If the spot only exists in the Rgel it will be recorded by **cgelp2** as a single unresolved spot as a special case of 1). On the other hand, if it exists in a gel *other* than Rgel, then it is recorded as an extrapolated spot.

Such *extrapolated Rspots* or *eRspots* which can be inferred from the local morphology and can be used to handle this *missing-from-the-Rgel* problem - so all corresponding spots exist or can be located where they should be found for *all* gels. A Rspot set represents a presumptive empirically derived set of canonical polypeptides. Since Rgels are real objects which are assumed incompletely representative of the totality of protein production, it is likely that some Rspot sets will not be represented in the gel chosen to be the Rgel. This is not really a problem since missing spots can be extrapolated to *any* gel - including the Rgel (in which case they are called eRspots). So all spots can be represented in all gels.

eRspots

Local morphology

We have found that for gel analysis, a most effective strategy is to concentrate on sets of local morphologies (both within and across gels) rather than to treating one gel or one spot at a time. Even if the task is defined as detection of the presence or absence of a single spot, some consideration of local morphology is necessary for any decision to be made by machine or when requiring human confirmation.

Recognition and identification (as opposed to detection) is quite difficult because of the absence of fixed shape and size of the individual spot. In dealing with spot identification, we are actually concerned with problems of local spot morphology, in which we are aided by the machine to 1) establish the proper region of regard, 2) maintain a local coordinate system (that of the Rgel) and 3) perform pictoral and numeric comparisons.

Composite Gel (CGL) spot database

We have discussed procedures that have been preparatory in that they deal with operations on individual spots or spot pairs. After constructing the set of Rspot sets, we are now in a position to use these data so as to construct a database which can be ordered as a function of biological, clinical, experimental or temporal variables. The richness of the database does not limit us to any one of these as the facilities which we now describe allow a multiplicity of orderings. A variety of representations may be chosen which may best be determined by the nature of the experiment. The biology demands that the analytic process be limited in its "attention" to the set of corresponding spots, one from each gel, a process that transcends the constraints of the individual gel. The **cgelp2** data management system permits this type of analysis to be applied successively to the majority of such Rspot sets.

partitioning gel DB

The types of operations performed consists of many computational or representational operations on the list of Rspot sets or sublists of Rspot subsets. The latter subsetting may be automatically accomplished based on an experiment dependent characteristic of a gel (from the accession file - to be discussed), on a statistical property of spot or Rspot set features, etc. Alternatively, the user may construct at will a working set of gels taken from the entire set of gels. A wide variety of representations of the data, both image and numeric, is available with many modes of display including superimposition on the original image. Important data structures include:

operations on DB

1. The set of working gels used to restrict the **cgelp2** operations to a subset of the gels in the database. Only gels in the working set are used in the computations.

2. The gel subsets structure which is used to manipulate gel subsets in order to easily redefine the working set or gel classes.

3. The classification sets which contain the names of the gels in each of up to nine classes. Thus, the user can, depending on the problem he is dealing with, classify gels by temperature, disease, metabolic condition, dose, etc.

4. A *search results list* of Rspots set number *names* which were found by one or more of the various available search options (or explicitly defined) is available to many of the **cgelp2** operators. A Rspot name is just a number used to index that Rspot set in the database.

In dealing with real data, it is frequently necessary to create a working subset of gels taken from the original database in response to different questions. The same data may be used to analyze different aspects of the same experiment by being partitioned in various ways. A related requirement is the facility to declare classes of gels and to create further subsets based on class membership. As an aid to manipulating subsets of gels, gels may be put into named subsets and treated as an entity.

Solution strategies

The properties that characterize spots, the principle of local morphology and the different objectives of different users require of an analysis system which is capable of varied analysis. Such a system offers the capability of designing a solution strategy or set of strategies rather than a direct and single solution. Among the important tools available for such strategies is the experimenter directed creation of multiple representations of the same data. Many of the system procedures are essentially procedures of presentation which allow the user to alternate between say numeric position or density data and synthetic images.

tools: Rmaps & mosaics

These tools include Rmaps and mosaic images which facilitate the backchecking of any Rspot set in both a global (the Rmap) and a local but multiple gel (mosaic) context. A mosaic is an image constructed by concatenating, in a 4x4 array, corresponding subregions from each gel, ordered in the array by spot density, surrounding the spot of interest. The mosaic provides a powerful tool whereby the user may be assured, on the basis of visual evidence, that a spot belongs to a given Rspot set. The Rmap image provides the link between the global location of an individual spot as seen numeric Rspot set data or local mosaic image. The Rmap is invaluable for rapid evaluation of the validity of spots found to be of interest by GELLAB-II statistical searches or manual examination of the database. Mosaics are insufficient for establishing a spot's context because of their locality and thus the Rmap fills this void.

numeric analysis

Numeric data, particularly functions of density presented in rank ordered tabular form, is useful for evaluating magnitude differences between spots in an Rspot set. The gray scale numeric representation of each pixel comprising a spot in a small window of the image is occasionally useful in determining whether a spot is

actually one or two or whether a spot was fragmented by the gel spot - segmentation program. The accession file information is always available for use with a data set or its derivatives. Any portion of it may be used as the associative key with which to regroup gels within the gel database.

Tools such as the foregoing are invoked as needed at user discretion to establish and/or confirm membership in a biologically significant canonical spot vector, i.e. a Rspot set. Moreover they can be used to quantitate substantive changes as a function of the biologic variable at issue.

In sum, the GELLAB-II set of programs represents a general method to organize and selectively compress the data of 2D gels so that the user may more efficiently perceive patterns out of the welter of individual spots. Each different instance of such compressed data can be thought of as a different view of the original database. Once Rspot sets of interest have been found, it is a direct process to quantitate their individual components by merely printing their Rspot sets.

Analyzing multiple gels as a continuum

Rspot set ≡ protein distribution

Each corresponding-spot polypeptide visualized as a spot may be thought of as having a distribution of spot densities when sampled in a set of gels. It is expected that this distribution will cluster multimodally in the case of significant spot density differences according to the biological state of the sample. Therefore, it is important that biologically non-significant variances be controlled and minimized (false positives). Adequate numbers of gel samples must be obtained for the database to aid in detecting these multimodal distributions.

We must assume that not all spots will be accounted for (false negatives). No automatic procedure can account for the almost infinite variety of image noise found in these gels. The semiautomation of the gel analysis may be sufficient to find spots for those biological problems where the changes are above the noise level and resolvable by the system.

1.4 Standard processing of a set of gels

This section introduces the basic scheme for analyzing a set of 2D gels using GELLAB-II. Although we use the GELLAB notation exclusively throughout the book, most of the concepts of gel analysis hold for other computer aided 2D gel analysis systems. On first glance, the large number (about 25) of GELLAB-II programs looks overwhelming. However, as shown in Figure 1.3 the basic 2D gel analysis is a sequential six or seven step procedure.

GELLAB tutorial

GELLAB-II programs are discussed in detail in Chapter 3. You might also review Chapter 2, page 71 which is a tutorial using these programs to show how gels might be analyzed. Chapter 5.1, page 503 discusses strategies for searching a

gel database.

We first list the names of some of the GELLAB-II programs and then discuss how they are used together to analyze a set of gels.

The major GELLAB-II programs are: **sg2gii** ([LipL80a], [LemP81a], [LemP83a]) used to quantitate gels into spot lists; **cmpgl2** ([LipL80a], [LemP81b], [LemP83a], [LemP91], [LemP93]) or **autopair** used to pair spots between spot lists; **cgelp2** ([LipL80a], [LemP81c], [LemP82a], [LesE82a], [LemP83a]) the composite gel database program; and **markgel** and **mosaic** used for visualizing spots of interest in the gels ([LipL80a], [LemP81c], [LemP81d], [LemP82a], [LesE82a], [LemP83a], [LemP92]). The **cgelp2** program is a disk based version of the original CGELP program [LemP81c]. *major programs*

Other ancillary programs are **getacc** and **landmark** used for data acquisition and landmarking respectively. (These concepts will be discussed). Accessioning is the entering or editing gel experiment, image and calibration information into an accession file information database for gel images. Normally, image acquisition is performed by program **getacc** running the **camera** program to acquire the gel image. Alternatively, **getacc** may be used to accession *previously* scanned gel image files from other scanners. When used this way, gel image files scanned elsewhere are converted to GELLAB-II standard Portable PiXture *.ppx* format using the program **ppxcvt** which is transparently invoked from **getacc**. *ancillary programs*

Normally, one views derived gel images using the X-Windows System on your workstation. The GELLAB-II programs **Xpix**, **accppx**, **getacc**, **landmark**, **cgelp2**, use this visualization method. *image display*

landmarking

Landmarking is the process of locating a set of corresponding spots in the reference gel and each of the other gels to be used in the database. This is performed using the **landmark** program which lets you manually compare the Rgel image with another gel image. Once the set of corresponding spots are interactively selected and defined, **landmark** saves them in a landmark database used by the spot pairing and other programs. Another alternative to using the **landmark** program is to estimate landmark sets automatically given a pair of GSFs when doing the spot pairing. This program, **autopair**, is under development.

Figure 1.3 illustrates the sequence of processing steps in the analysis of a set of gels. Section 1.6 discusses some of these data structures in more detail. Figure 1.4 illustrates the sets of associated data files generated during a gel analysis. *analysis steps*

At the time a set of gels are accessioned (see step:1) in Figure 1.3.) using **getacc** (or later using **makjob**), several batch job programs (UNIX *scripts*) are generated for later execution. [4] These include: a) a job to interactively landmark the set of gels (step:2) under the UNIX shell, b) a job to segment (step:3) and perform the gel

[4] The **makjob** program can be used at any time to create the same set of batch jobs. This is useful for creating new projects from different collections of gels.

automating the analysis

comparisons (step:4) generating *Gel Segmentation Files* (GSF) and *Gel Comparison Files* (GCF), and c) a job to build the initial *Composite GeL* (CGL) database and perform an initial analysis of it (step:5). In step:6 Rmap images (using program **markgel**) and mosaic images (using program **mosaic**) can then be computed and display (using program **accppx**) from the database results of this analysis.

Additional batch job scripts are also generated to, optionally, permit the user to manually landmark the gels by first segmenting the gels, then generate plot files with which to manually select landmark spots, and finally perform the gel comparisons. Section 3.9, page 369 gives an example of generating such batch scripts and shows the scripts produced.

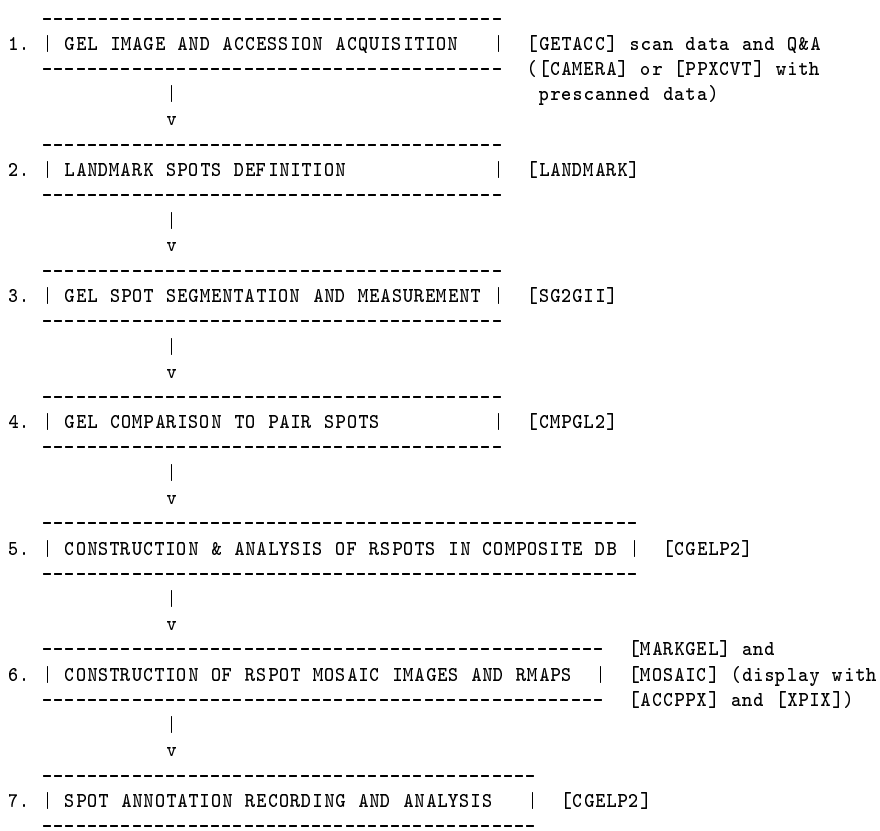


Figure 1.3. Block diagram of the 2D-gel analysis GELLAB-II system. Programs associated with major steps of GELLAB-II are indicated in "[...]". Gel images are acquired by scanning with a camera interfaced to the UNIX system and saved on the computer disk in step **1**. *Accession information* about the set of gels is also used to update an accession file. *Landmark spots* are then manually selected which are well defined spots spaced fairly evenly throughout the gel - with more landmarks in regions with higher distortion in step **2**. Using gel image flicker alignment, the landmark spots are aligned for all of the gels with a *Representative gel* (Rgel). The gel images are then segmented and measurements made of the spots which are found in step **3**. This information and the raw segmentation data is then used to pair corresponding spots in the remaining gels with the Rgel in step **4**. The set of gel pairings with the same Rgel may be merged together to form a list of sets of equivalent Rspots called the *composite gel* database (CGL) in step **5**. Thus a Rspot set (most likely) contains corresponding spots from all the gels in which it occurs. Finally, in step **6** Rmap and mosaic images of statistically significant spots can be computed and displayed. These results can be recorded in an annotation database and further analyzed against other results in step **7**.

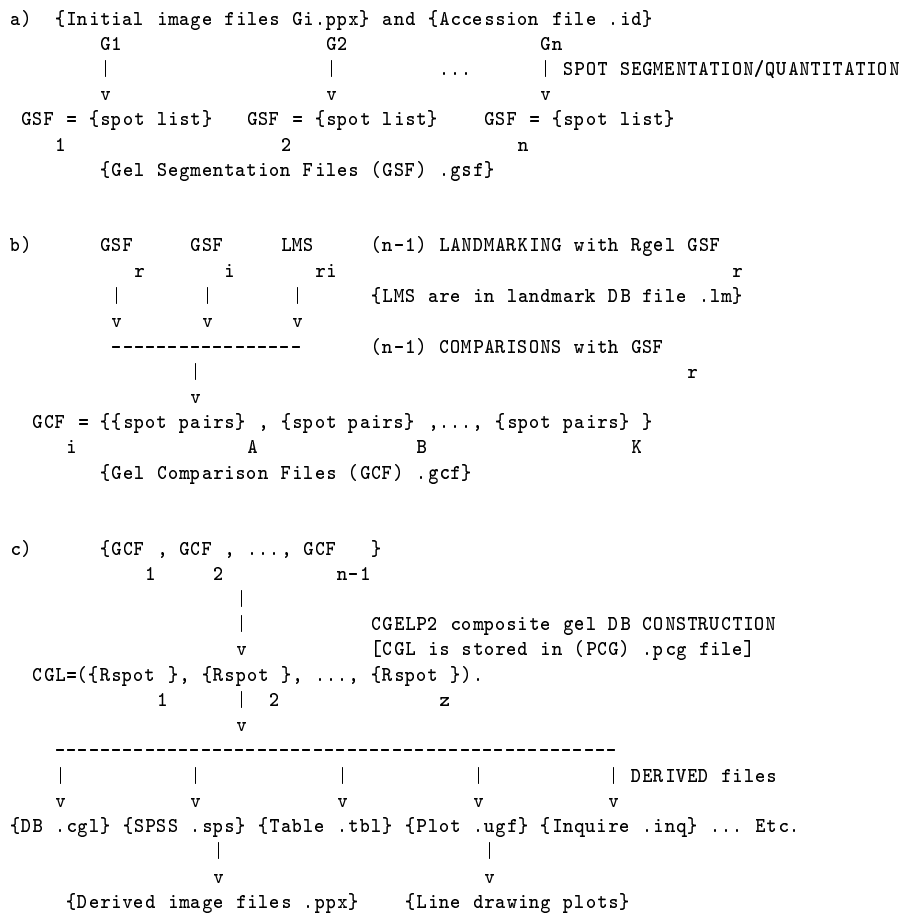


Figure 1.4. Some of the files used in gel analysis. *Data file structures* and corresponding *file extensions* used in the gel analysis. A GELLAB-II file extension is a 2 or 3 character name preceded by a '.', (e.g. `ts3pcg.cgl`). **a)** Gel Segmentation Files (GSFs) are produced by segmentation of the gel images by **sg2gii**. **b)** Gel Comparison Files (GCFs) are produced by comparing GSFs using landmark spots with **cmpgl2** or **autopair**. **c)** The Paged Composite Gel (PCG) database is constructed by merging GCFs with **cgelp2**. The PCG DB is a 3D data reduction of the original set of gel images and accession information. The derived files are other types of data derived from the PCG DB.

running programs

Programs are invoked by naming the program and specifying optional arguments (e.g. '`sg2gii 324.1 -7x7filter`' to segment gel `324.1`). All programs have several common UNIX-style command-line switches to facilitate learning the consistent user interface. This subset of switches is useful in learning how to run particular programs and is described in Section 1.4.1.

Descriptions of the algorithms used in these programs are given in many of the GELLAB-I papers listed in the references ([LipL80a], [LemP81a], [LemP81b], [LemP81c], [LemP81d], [LemP81e], [LemP82a], [LemP83b], [HowR83], [LemP84a], [LesE84b], [LemP88d], [LemP89a], [LemP91], [LemP93]). Good introductory papers which describe the basic GELLAB analysis are [LipL80a] and [LesE81b] with [LemP83a] being a more general and detailed summary. Extensions to the early system are discussed in ([LemP81d], [LemP82a], [LemP83b], [HowR83], [LemP84], [LesE84b], [LemP88d], [LemP89a]). A comparison of aspects of 2D gel database analysis systems is given in [LemP89a] with examples of visual output described in [LemP92]. Appendix G, page 611 presents summaries of some of the algorithms.

algorithms

1.4.1 On-line information in GELLAB-II

On-line help is available from each of the GELLAB-II programs using the `-info` switch with each program. This is typed *after* the name of the program. Only one UNIX *man(1)* file, `gellab.1`, is available which discusses GELLAB-II at the system - not the individual program level. If the `gellab.1` file is installed in one of the UNIX *man(1)* directories, then you need only type `man gellab` to print it. See Section 6.8, page 550 for more information on printing this file.

help!

The `-info` switch causes the program to print detailed information about the program including *what* it does, *how* to run it, and *literature references* specific to that program. Eg.

on-line information

```
cgelp2 -info
```

You can use the UNIX *more(1)* program to page through the documentation slowly. Eg.

```
cgelp2 -info | more
```

or use it to create a printable file:

```
cgelp2 -info > cgelp2.info
```

Note, all of the information in the `-info` printout is also available in this book.

The `-version` switch causes the program to print the version number of the program. Eg.

```
cgelp2 -version
```

The `-usage` switch causes the program to print a quick list of command line switch `usage` for the program. Eg.

UNIX command line

```
cgelp2 -usage
```

or

```
cgelp2 -usage | more
```

1.4.2 Invoking GELLAB-II programs from X-Windows

keyboard or menus

All GELLAB-II programs are normally invoked by typing the name of the program (with appropriate switches and arguments added). In addition, they may be invoked using a menu selection from the window system. [5]

X11 task menu

Currently, the following X11 `twm(1)` window manager menu is available which invokes the GELLAB-II programs as shown below. The menu is organized by task rather than program name. Alternatively, another menu is also available with the program names listed explicitly. The detailed sub-menus are not listed here but are self evident when invoked. The menu structure is defined in file `.twmrc`.

MENU: GELLAB TASKS	SUB-MENU
Accession gels	**getacc**
Cvt image file to PPX	**ppxcvt**
Cvt PPX to PostScript	**ppx2ps**
Debug PPX image file	**ppxodt**
Display gel images	**accppx**
Draw Rmap GSF plot	**dwrmap**
Gel Database Manager	**cgelp2**
Landmark gels	**landmark**
Make GELLAB Batch Scripts	**makjob**
Mosaic derived imaged	**mosaic**
Pair gels to GCF	**cmpgl2**
Print State-File(GELLAB)	**pgelrc**
Rmap derived image	**markgel**
Scan Datacopy image	**camera**[6]
Segment gel to GSF	**sg2gii**
Xpix *image display*	**Xpix**

Using GELLAB-II with X-windows version X11

The GELLAB-II programs **cgelp2**, **Xpix**, **getacc** and **landmark**, **plotn** are X-Windows version X11.

The window manager we use with X11 is *twm(1)* and is available on the free MIT distribution. It requires a startup file *.twmrc*. X11 is started using the following and requires an initialization file *.xinitrc*. Both files are supplied with the GELLAB-II distribution. We have edited these files to make things a little easier to use with GELLAB-II, but you can use your own versions if you prefer. In addition, you should add the following lines to your *.cshrc* startup file. The *xinit*(1) program is used to automatically start X11.[7].

[5] See Appendix D, page 603 for information on starting X-Windows if it is not already started.

[7] If you normally run OpenLook `olwm` or Motif `mwm` window managers, you may want to switch

```
1% set path=($path /usr/local/bin/X11)
2% xinit
```

1.4.3 Use of Xpix to visually flicker-compare two gels

The **Xpix** program under UNIX with the X-Window System can be used to visually compare two images by alternately displaying the two gels on the same position on the screen. The region selected from each gel being displayed can be changed. When two corresponding spot regions from different gels are aligned, the flickered sum of the two regions pulses. The original FLICKER program was implemented on the Real Time Picture Processor (with GELLAB-I) using the FLICKER program [Lem79a] which used special hardware to implement the flickering (see Appendix F, page 607). This visual comparison method assumes only that the gel images and the accession file are on the user disk. No spot data computations or data bases are required to perform flickering. There are two versions of **Xpix**: **Xpix11** and the newer **Xpix2**. The latter works better under OpenLook but currently lacks some of the functionality of the **Xpix11**. With GELLAB-II, two gels may be loaded using the **accppx** program by specifying their two accession numbers and flickered using the `Compare images` mode in **Xpix**. The **disp11** program can flicker the entire gel. Program **accppx** loads **Xpix** with the two gel images associated with the two accession numbers. Section 3.1 discusses **accppx** and Chapter 4, page 473 discusses **Xpix** in detail.

pulsing regions when aligned

Sometimes the experimental versions have a different name and must be taken into account using the following system manager operations.

```
   1% cd ~gelmgr/gellab/bin/sun4
or    cd ~gelmgr/gellab/bin/sun3
                # Enter the executable binary GELLAB directory.
   2% rm Xpix getacc landmark
                # Remove old versions since we redefine the names
   3% ln -s Xpix11 Xpix
or    ln -s Xpix2 Xpix
      ln -s getacc11 getacc
      ln -s landmark11 landmark
                # Make symbolic links from X11 programs to old names.
```

When in flickering gels in the **landmark**, program, it is possible to interactively increase or decrease the amount of time each gel region is displayed when flickering.

---

to the MIT `twm` window manager. This is because we currently have problems switching the colormap with these other window managers - especially OpenLook. The **Xpix** program behaves better under OpenLook than Xpix11. To use `twm`, edit your `.xinitrc` file to use `twm` before you run `xinit`

This is useful in slowing down the flickering to get ones bearings when aligning two radically different regions.

1.5 List of GELLAB-II programs

GELLAB programs may be run on a UNIX system. [This assumes that your user's directory *path* (cf. Sections 1.6.3, 3, and 6.7) has been set up correctly].

In general, almost all GELLAB-II programs require one or more arguments so you should read the individual programs' documentation or tutorial prior to attempting to run them. GELLAB-II programs use a resource file called `gel.rc` in the users current path to provide user state information. It is created and changed using the **pgelrc** program. This includes the names of various directories for image, gel database, and other intermediate files (see Section 1.6.5). If this file is not in your current project directory - you must create it. Running any GELLAB program will tell you that it is missing and suggest that you create it. Running **pgelrc** the first time in a new project directory causes it to prompt the user in defining the initial `gel.rc` file. Later, running **pgelrc** in the same directory prints a user-friendly form of `gel.rc`. If you wish to change the `gelr.c` resource file creating a new project, you should either use **pgelrc** or a text editor.

a set of programs

A batch script generation facility which is part of GELLAB uses the programs **makjob** (and may be called from **getacc**) automates running the major GELLAB-II programs. Using these batch jobs, minimum operator intervention is required for a major part of its operation during the initial data reduction.

The GELLAB-II programs are listed below in Table 1.1. Section 1.5.1 gives a brief synopsis of each of these programs. Chapter 3 describes each of these programs in more detail. Literature references are given which describe the algorithms in more detail. As mentioned before, no individual UNIX *man* pages are available for these programs. The equivalent information is available with `-info` switch options instead (see Section 1.4.1).

TABLE 1.1. GELLAB-II programs as of July 6, 1993.

accppx - display gel image(s) given accession number(s).

camera - capture Datacopy camera image into *.ppx* image file at NCI/IPS.

cgelp2 - interactive Paged Composite Gel database analysis.

cmpgl2 - paired two GSF spot lists using landmark DB$\Longrightarrow$GCF.

dendrogram - Rspot, gel expression profile cluster analysis.

dwrmap - draw Rmap numbered plot of GSF spot list.

getacc - multiple gel accession: image, info., calibration.

landmark - interactive graphics acquisition of landmarks for LM DB.

makjob - create GELLAB-II scripts for batch processing gels.

markgel - generate Rmap image from **cgelp2** (*.sps*) data file.

mosaic - generate mosaic image from **cgelp2** (*.sps*) file.

pgelrc - "pretty print" the `gel.rc` GELLAB-II 'state' file.

plotn - plot GELLAB-II Universal Graphics Files (*.ugf*).

ppxcvt - convert foreign formats $\Longleftrightarrow$ Portable PiXture file.

ppxodt - Portable PiXture (*.ppx*) file image debugger.

ppx2ps - convert image file to PostScript for laser printer.

sg2gii - segment gel image to Gel Segmentation File (*.gsf*).

tek2psG - convert Tektronix plot file to PostScript.

Xpix - display and manipulate Portable PiXture file images.

1.5.1 Synopsis of GELLAB-II programs

A Short synopsis is given here for each of the GELLAB-II programs with more detailed descriptions to be found in Chapter 3.

accppx display one or two gel image PPX type files using **Xpix** given the accession number(s) and optional picture prefix type or picture files. Optional picture types include: `l` (L) for landmark Rmap, `m` for Rmap images, `y` and `z` for segmented spot images produced by **sg2gii**, `c` for segmented connected component images (see *PPX file* definition in the glossary as well as **sg2gii** and **markgel** for more information). It can also display images given the image file name (see **mosaic**).

autopair (future) automatic gel pairing program (this replaces **cmpgl2** and the need to define the same set of landmarks for all gels for doing spot pairing). Only the initial gel need be landmarked with all of the landmarks. The remaining gels can be landmarked if desired (for better accuracy). **autopair** generates a *Gel Comparison File (GCF)* (*.gcf*) from two *Gel Segmentation Files (GSF)* (*.gsf*) produced by the **sg2gii** segmentation program.

camera runs the NCI-FCRDC/IPS Datacopy 612F CCD camera to scan an image generating a *.ppx* image. Standard image size is 512x512 or 1024x1024. Other sizes may also be scanned. Alternatively, the SUN VideoPix camera program *vcftool(1)* may also be used for low resolution TV camera scanning.

cgelp2 runs the Paged Composite Gel database analysis system. This builds a PCG DB file from a set of *Gel Comparison Files (GCF)* produced by the **cmpgl2** or **autopair** programs. It may be run as either interactive terminal, background batch, or X-Windows menu-oriented interfaces. When running cgelp2 additional information is available on top level commands by typing `HELP` to list all of the top level commands, which are prompted for by <`CMD`>, or `HELP specific-command`. For example, type `HELP HELP` to get more information on the HELP command. ([LipL80a], [LemP81c], [LesE81b], [LemP82a], [LemP83a], [LemP83b], [HowR83], [LemP84a], [LesE84b], [SonP85], [SonP86], [LemP88d], [LemP89a])

cmpgl2 runs the gel pairing program which generates a *Gel Comparison File (GCF)* (*.gcf*) from two *Gel Segmentation Files (GSF)* (*.gsf*) produced by the **sg2gii** segmentation program. It also requires a list of landmarks from the landmark DB (LM) entry for the two gels being paired. ([LipL80a], [LemP81b], [LesE81b], [LemP83a])

dendrogram generates a dendrogram cluster analysis plot. It uses **cgelp2** produced SPSS (*.sps*) or INQUIRE (*.inq*) files. It can cluster a set of Rspots as a function of density of a set of gels or cluster a set of gels as a function of the density profile of a set of Rspots sets. It also plots the results after they are generated and/or makes an optional *.ugf* plot file. A data file (*.dgm*) is also produced which contains numeric cluster analysis information. [SonP86]

dwrmap given a *Gel Segmentation File (GSF)* Plots a Rmap from the *.gsf* file. It plots a Rmap with spots labeled by their GSF spot number. This can be used with the `-lmsedit` option with the **landmark** program to manually generate the *landmark set* data entry. It also plots the results after they are generated and/or makes an optional *.ugf* plot file. [LemP82a]

getacc (called **getacc11** or **getacc11a**) is used in a data acquisition session is used to acquire 2D gel images previously scanned if not using the DataCopy-Camera at NCI) and their related accession information including experiment information, gel computing window and ND wedge calibration. This information is appended to the gel accession file. Alternatively, one can do the above acquisition but on previously scanned images. It can also be used at any time to edit the accession entry for any gel in the database. At the end of the session it asks if you wish to generate batch scripts for further processing using the **makjob** program and then starts **makjob** using the list of gels beging edited. ([LipL80a], [LemP81a], [LemP83a]) *start here!*

landmark (called **landmark11**) is an interactive X-windows graphics program to landmark or edit landmarks for two gels. This process defines a small set of corresponding spots (10 to 52) in each of the two gels. These spot positions are used to update an entry in the *LandMark (LM) database file* which is used by other programs including the spot pairing program **cmpgl2**. It can also use a previously defined LM DB entry to indicate where the landmarks are in the Rgel image when landmarking another gel. This makes finding the same landmarks much easier and reproducible. ([LipL80a], [LemP81b], [LesE81b], [LemP83a]) *starts analysis*

makjob generates GELLAB-II UNIX scripts. It requests a list of gel accession numbers for a subset of gels previously accessioned. It then asks a few questions regarding the type of experiment to be performed and generate UNIX batch scripts to a) interactively landmark the set of gels, b) segment the gel images into GSF spot lists, c) pair GSF spot lists into GCF paired spot lists, d) merge the GCF files by constructing a PCG DB file and e) perform an initial statistical analysis of the PCG DB. A makjob run can be customized to perform some and not other specific analyses (see `-info` switch for makjob). See example of running **makjob** in the examples in Sections 3.9, 3.7 and 2.2. ([LemP82a], [LemP83a])

markgel generates a Rmap image having specified a gel accession number and a SPSS (*.sps*) file generated by the **cgelp2** program. The Rmap is the synthetic image generated by the projection of the set of spots specified by the SPSS file onto a copy of the gel image associated with the gel accession number. ([LipL80a], [LemP81c], [LemP83a])

mosaic generates one or more mosaic derived images having specified a particular Rspot number and a SPSS (*.sps*) file generated by the **cgelp2** program. A mosaic of an Rspot for a set of gels is a composite 4x4 panel image or graphic formed from panels from each gel arranged in a regular checkerboard pattern ordered by minimum spot density (protein concentration). The panels are

taken from a subregion of each gel surrounding a particular Rspot. The mosaic can then be displayed or printed. ([LipL80a], [LemP81c], [LemP83a])

pgelrc prints a user friendly form of the `gel.rc` GELLAB state file. This file contains the default names of various database files, directories, segmentation parameters, and information on the last data processed. It is used by all GELLAB-II programs upon startup. The `gel.rc` file may be changed at any time using **pgelrc**. If you are starting a new project (i.e. have created a new directory to put the data in), then the **gel.rc** file does not exist. Running **pgelrc**in the new project directory will run an interactive question and answer session to generate the `gel.rc` file. It can find the next free accession number and picture file name for use in entering new gels. It will also extract information for accession numbers and pictures in your database. Eg. find the gel accession numbers of all gel picture files contained in a directory, or find all picture files given gel accession numbers in your database, etc. [LemP82a]

plotn reads Universal Graphics Files (*.ugf*) produced by various GELLAB-II programs using the SMDISP ("small OMNIGRAPH" emulation package). It is able to replot a *.ugf* on the same or different type of display as well as to plot the file on other devices (such as a PostScript laser printer, Tektroix 4010 and X-Windows).

ppxcvt convert different picture file formats into the GELLAB Portable PiXture file format (*.ppx*). It can convert an almost any arbitrary image file with or without a header and with possibly extra leading and trailing data per line to a 1024x1024 or 512x512 PPX image file. The larger image is sampled or optionally averaged to the smaller one. Data may be binary, or ASCII hex, decimal or octal. It may be have 8 or 16-bit pixels. It may be complemented, scaled and a log transform taken. If the image is a BioImage Systems Inc. gel image, then default options may be invoked to read the 1Kx1K pixel image and convert its gray-value to OD wedge calibration which are then stored in the PPX image header. In addition, higher resolution TIFF files from Molecular Dynamics and Truval can be converted. BioImage and Elsie gel image files can also be converted. Additional options are available to edit the PPX file header.

ppxodt is a picture debugger for opening, reading pixels, 3x3 neighborhoods, and 18x18 windows of a *.ppx* file. Data may be viewed in hex, octal or decimal. Individual pixels may be changed and the edited picture file saved.

ppx2ps converts a *.ppx* image file to a PostScript file. PostScipt files can then be printed on many laser printers such as the Apple LaserWriter.

sg2gii is a gel spot list segmentation program which generates a *Gel Segmentation File* (GSF) (*.gsf*) from the image file associated with the gel accession number.

The accession number and gel image are produced by the **getacc** gel image acquisition program. **sg5gii** is a new version of **sg2gii** which can segment gels with variable size images and with more than 8-bits/pixel. ([LipL80a], [LemP81a], [LesE81b], [LemP83a], [LemP82a], [LemP91], [LemP93])

tek2psG is a UNIX style filter used to convert Tektronix plot files to PostScript. It is typically used with the **plotn** program which converts the GELLAB Universal Graphics Files (*.ugf*) to Tektronix 4010 output. It can be used to pipe the PostScript output directly into a laser printer input queue.

Xpix (X11 versions is called **Xpix11** and **Xpix2** is a replacement) is a general purpose X-windows *.ppx* file interactive display program. It is controlled by the user moving and clicking a mouse to get menus selections and interact with the image(s). It can manipulate one or two images on the screen at a time with each image having its own real-time small zoom window. It can perform general image processing types of operations. [LemP88a]

1.5.2 GELLAB-II programs which require X-Windows

Most of the GELLAB-II programs are portable and independent of a X-Windows graphics system. However some are not and are discussed here. You must use these programs to interactively obtain information used in the analysis. The X-Window System [SchR86], [SchR88] is used for both displaying gray scale images and for obtaining specific spot position information about images by user interaction. Of course, the X-Window System itself is highly portable and is available free from MIT. It runs under most UNIX systems as well as DEC's VMS and other operating systems. If you have X-Windows, then don't bother reading further - this discusses what you can do if you don't have it.

image display

programs needing X-windows

The **getacc** program is normally used to interactively scan or convert gels and then add the names of these images and corresponding experiment accession information to the accession file. It is also used to interactively define an active gel region called the *computing window* (CW). It may also be used to analyze and calibrate the gel image in terms of optical density (OD) using a co-scanned neutral density (ND) step wedge or cpm (counts/minute) chip standard.

accessioning gels

The **landmark** program is used to interactively define 10 to 52 landmark spots for a pair of gels, a *set of landmarks* (LM set), and then lets you add this LM set to the landmark database file. Again, because of the graphics interaction required, landmarking is X-Windows dependent. You can go back later and edit the landmarks for a given gel - although those for the Reference gel are not currently allowed to change.

landmarking

Although you can process a database without any further interactive visualization, it is invaluable to performing an analysis. The **Xpix** and **cgelp2** programs

offer X-Windows visualization of images produced by various stages of the analysis and are the prefered way of viewing these images.

Using line graphics plots as alternative to displayed images

Tektronix 4010 graphics

GELLAB-II line-graphics functionality is written to use portable Tektronix 4010/4014 graphics using the OMNIGRAPH [DCRT79] plotting function emulator package called SMDISP. This is independent of X-Windows. SMDISP emulates a subset of OMNIGRAPH which is compatible with the Tektronix 4010 display. So programs **dwrmap**, **dendrogam**, **cgelp2**, **plotn**, etc. which generate line-graphics can be performed on an inexpensive graphics terminal or PostScript laser printer.[8] They do this by invoking the **plotn** program in the background which then displays the plot file. In the latter case, the default lineprinter name is `laser`. These programs can generate plot files whcihc can be redisplayed using the GELLAB-II **plotn** program as X-Windows, Tektronix output, PostScript output or printed to a laser printer. The default printer may be defined using the `LASERPRINTER` environment variable. Eg. to change this default to a printer called `laser2`, do `setenv LASERPRINTER laser2`.

You can draw gel derived images as line graphic plots which can be saved as UGF plot files. Program **dwrmap** draws GSF files as Rmaps. Program `dendrogram` clusters Rspots as a function of gels or gels as a function of Rspots and draws this analysis as a dendrogram tree. Program `cgelp2` has a number of line graphics commands: `CCPLOT`, `EXPRESSION-PROFILE`, `DCPLOT`, `DDPLOT`, `HISTOGRAM`, `MOSAIC`, `PLOT`, `RMAP`, etc. The `SET DISPLAY` is used to select the display which should be either a XWND (i.e. X-Window popup), 4010, VT240 or LASER (for dumping the plot directly to a connected Postscript laser printer). The UGF plot files are printed by conversion to Tektronix 4010 graphics format using the **plotn** program. This in turn can be converted by program **tek2psG** to Postscript for printing on a laser printer.

It is also possible to view derived images by printing a "dithered" black and white version of the gray scale on a Postscript laser printer using the **ppx2ps** program.

1.6 The GELLAB environment - conventions

learning UNIX

UNIX is the current computer operating system environment in which GELLAB-II runs. It is used to invoke specified programs as high level functions (or operators) supplying them with arguments originally specified through the UNIX shell. Appendix C, page 593 lists a small subset of UNIX which is more than sufficient

[8] Tektronix 4010 style graphics can be emulated on a large number of terminals and UNIX workstations. For instance, the DEC VT240, VT340 type terminals have a 4010 emulation mode. SUN workstations have a 4010 emulation mode in both *suntools(1)* using the *tektool(1)* program and under X-windows under the *xterm(1)* program.

for learning UNIX. These include all of the UNIX commands you might ever need in the GELLAB-II environment.

The shell is a program the investigator uses to interact with UNIX. It is part of UNIX - not GELLAB. There are several shell programs (*sh(1), csh(1), ksh(1)*) provided with UNIX. However, we prefer the `csh` because of its power and is widely available. Once the user has started UNIX, the shell is fairly transparent and unless they wish to do fancy things, can be ignored. When programs finish, they exit back to `csh` rather than the UNIX kernel. Thus, `csh` is a very high level language for controlling an essentially infinite set of programs. These programs may be compiled programs in the classic sense or script files of UNIX commands in the sense of batch or interactive batch. That is, typing the name of a batch script to the shell causes the script to be executed.

UNIX shell

UNIX batch processing

When any GELLAB program is invoked by typing its name, a state file of user specific information is read by that program. The *state file* for GELLAB is called `gel.rc` (c.f. Section 1.6.6, page 63 for description and Section 3.13, page 407 - **pgelrc** for printing and defining the state).

state file

For example, the program **accppx** displays one or two gels given their associated *accession numbers*. It does this by translating the accession numbers to picture file names and invoking **Xpix** with these image files. In order to to this, it has to look up the picture file names corresponding to accession numbers using the accession file. The name of the accession file is specified in the `gel.rc` file in the users directory path. So that a command such as

```
accppx 324.1 384.1
```

can then use the accession file to do the lookup from accession numbers to picture file names before it displays the two gels. Furthermore, the `gel.rc` file also gives the path names of the picture file directories that **accppx** should look in to find the picture files. We will see many more examples shortly, but it is important to remember that programs are invoked by typing their name followed by any required arguments and optional switches. Also remember that the `gel.rc` state file tells GELLAB where to look for special files and directories where data is to be stored.

To process a program consisting of a of list of UNIX commands, one would construct a *shell script* file. This file should have *execute* file protection set with *chmod(1)* described in Section C, page 593. This is the formalism used by the GELLAB-II **makjob** and **getacc** programs when they create batch scripts. All scripts mention the name of the UNIX shell (e.g. `/bin/csh`) to be used in the first line followed by the shell commands. The `#` indicates the remainder of the line is a comment and is ignored by the shell. For example,

```
#!/bin/csh
accppx 324.1 384.1
```

```
      accppx 324.1 369.1
#     accppx 324.1 577.1     # The '#' comment prevents this line's execution
         ...
      accppx 324.1 378.2
```

See Appendix B, page 585 for a list of some of these scripts.

1.6.1 UNIX interactive and batch GELLAB-II scripts

Given a UNIX command script file such as the one above, the GELLAB-II operator then starts it by typing the name of the script as:

```
cmdfile.do
```

where *cmdfile* is the name of the file.[9] As mentioned, the gel acquisition programs **getacc** or **makjob** generates a set of batch scripts for each gel analysis project. For example, if the project name (used in the tutorial Chapter) is "`ts3`", then some of the scripts generated are: `ts3lms.do, ts3prc.do, ts3cgl.do`, and `ts3cgl.gdo`. When a landmarking script (e.g. `ts3lms.do` - see page 376) is run, the operator responds to terminal prompts during the landmarking process (see step (2) of Figure 1 and the example in Section 3.8). The remaining two gel analysis batch jobs corresponding to steps (3+4) (e.g. `ts3prc.do` - see page 376) and (5) (e.g. `ts3cgl.do, ts3cgl.gdo` - see page 374) of Figure 1, are automatically started in turn. This occurs after landmarking is completed. No further operator interaction is required until the CGL database and initial analyses are completed. At that point exploratory data analysis may be performed.

gel analysis scripts

The **makjob** program can also be used to create these batch jobs from a set of gels already accessioned (and possibly segmented) under different exploratory analysis conditions. This might, for example, define a different group of gels taken from several different projects which you might wish to analyze together. Using batch jobs is the simplest and preferred way to run the initial part of the gel analysis which is routine and time consuming. The UNIX background batch jobs are started as:

creating script files

```
cmdfile.do >& cmdfile.log&
```

by putting a '`&`' at the end of the line it will also create a batch log file, `cmdfile.log` in this example, of all output generated when it was run. This is indicated by preceeding the log file name with a '>>'.

[9] GELLAB-II uses the convention that UNIX scripts have a *.do* file extension. A **cgelp2** input script has a *.gdo* file extension which is a special case.

1.6.2 Gaining access to the system

To gain access to UNIX and the GELLAB-II system, the user must first log onto the system.[10] This is done as follows under your UNIX account set up by the system manager for your computer.

```
login your account name (Press Return key denoted <CR>)
password:  your secret password<CR>
```

The user might then run a GELLAB command (in future examples we will often leave out the <CR>, but it is understood that all lines typed to the computer at shell level and to GELLAB program command prompts are terminated with it),

```
pgelrc<CR>
```

In general, to exit any program type the `control/C` keys at the same time - this aborts the program you are in (except for special cases like text editors). When you are done with your work, you must logoff the computer. This is done from the UNIX shell level by typing `control/D` or `exit`. One special exception is in the **cgelp2** program - where to exit you should type `exit` (see page 212). Often in the examples which follow throughout the remainder of this book, you will see UNIX shell level commands preceded by a 'number' followed by a '%'. This is the `csh` shell's current command line history number and prompt requesting you to enter a UNIX command. The history number is incremented for you each time you enter a command. For example,

UNIX history

```
25% pgelrc -usage
26% pgelrc
   .
   .
   .
35% history
```

Typing the UNIX `history` command as in step `35 %` prints the previous current entries. You can access previous UNIX commands by their history numbers, but this will not be discussed in this book and is not necessary to know in order to run GELLAB-II.

[10] Multiuser systems such as UNIX require you to identify yourself by "logging in". This is done in order to keep different user's files under separate accounts. Single user systems such as MS-DOS or OS/2 do not have this facility.

1.6.3 Running GELLAB-II from your area

The GELLAB-II system executable files are kept a special GELLAB manager's account called ~gelmgr. Then, all other GELLAB user's can then access the GELLAB programs from this manager account.

~gelmgr *account*

Lets assume that the GELLAB executable files are in this ~gelmgr account. Then, for each additional GELLAB user account, you can add this directory path to their UNIX $path environment variable GELLABMANAGER as described in Section 6.3 page 544. We reiterate here how to automatically set up GELLAB manager and user accounts.

gelusr *group*

Doing this requires that the UNIX administrator set up a UNIX "group" called gelusr (ask your administrator to do this in the UNIX system file /etc/group. Both ~gelmgr and all GELLAB user accounts need to belong to the same UNIX group gelusr so they can access the GELLAB-II executable files). The following lines should be contained in each GELLAB user's .cshrc file (which is read by UNIX when you first log in) to give you access to all of the GELLAB programs.

```
setenv GELLABMANAGER ~gelmgr
set $path = ($path GELLABMANAGER/gellab/bin/)
set $path = ($path GELLABMANAGER/gellab/bin/`arch`)
```

You should now be able to access both the GELLAB-II executable files and demonstration scripts and data. You still need to set up your *user account*. Also see Section 6.7, page 549 for a discussion on creating additional GELLAB user accounts and directories.

1.6.4 The gellab directory

Each user should have a top-level subdirectory called gellab which is created as discussed above. The gellab directory will have the sub directories, listed in the following table.

TABLE 1.2. Subdirectories in any user's project gellab directory.

ann	(future) annotation database files.
aux	auxiliary runtime directory of **.gsf, *.gcf, *.ppx* derived data etc.
gen	alternate directory for **cgelp2** generated derived files.
id	accession files.
lms	landmark set database files.
org	(optional) original images directory.
pcg	Paged Composite Gel database files.
ppx	original gel scanned image **.ppx* files.
tmp	temporary **.ppx* image directory.

In addition, the following directories are found only in the ~gelmgr/gellab directory which belongs to the GELLAB-II manager.

bin executable files and scripts in ~gelmgr.

bin/sun3 executable files in ~gelmgr.

bin/sun4 executable files in ~gelmgr.

doc GELLAB-II general documentation in ~gelmgr.

demo demonstration project in ~gelmgr (may be deleted if not wanted).

Picture disk and other data file areas

file extensions

Mention is made throughout this book of the user's *picture disk* area. For some large users of the NCI/FCRDC GELLAB system, additional short-term file space may be available on another disk on the network for storing image files. User databases can be archived to tape and restored from tape.

File types are indicated by their file extension - i.e. the last two or three characters after the last '.' in the file name. Original gel image *.ppx* files are stored on the picture disk. If other original files (e.g. TIFF files from some scanner) are used, they can be stored in the gellab/org directory or the value of *original files path* set to where they actually reside. Temporary files and images are saved on the temporary picture disk while synthesized Rmaps and mosaics images and *.gsf* and *.gcf* files are stored on the auxiliary picture disk. The *.pcg* paged composite gel database files are stored in a separate database area. Derived **cgelp2** files generated during analysis including *.inq, .sas, .sps, .srl, .tbl, .ugf* etc. may be stored in either the pcg or gen sub directories. The different picture disks may be defined to be the same path. Sections 1.6.5 2.1, and 6.2 discuss these directories in more detail.

project directories

Each of the 2D gel analysis projects that an investigator sets up in their directory will contain its own gellab directory subtree. This is illustrated below in Figure 1.6.

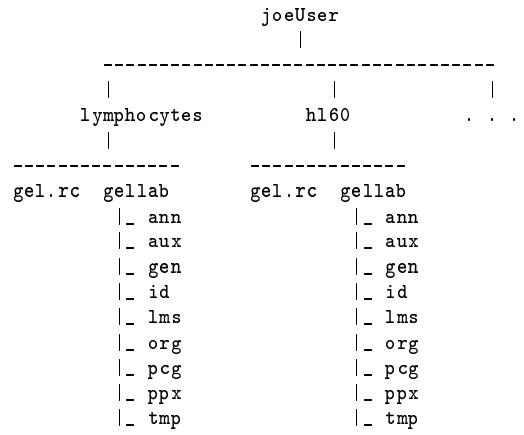


Figure 1.6. Typical investigators directory tree. Each gel project could be placed in its own directory (e.g. `lymphocytes`, `hl60`, ...). It is also possible to merge both sets of files into one `gellab` directory in which case multiple PCG databases and possibly multiple accession files could be used. However, the former scheme is clearer. As will be discussed, the `gel.rc` file in each project usually points to the sub directories in the associated `gellab` sub directory.

image size

GELLAB-II images are currently of default size, 512x512 8-bit (256 gray level) pixel elements or *pixels*, with white represented by 0 and black by 255. Alternatively, a 1024x1024 8-bit default may also be used by specifying the `-1Kx1K` switch during image acquisition. Higher resolution iamges can also be used if the files were converted to a higher resolution using **ppxcvt**. These GELLAB-I compatible images are stored on the disks as binary files with a *.ppx* file extension. A picture header is used at the start of each picture file. The picture header is defined by file `ppxfmt.h` (see Appendix H. GELLAB-II images can be defined with an arbitrary size as defined by this image header (e.g. 1024x1024, etc.) and can be processed as such.

Disk space requirement

user data files

The GELLAB executable programs themselves are generally installed in directory `~gelmgr/gellab/bin/sun4` If this is done, then no user space is taken up with private copies of executable programs. User data is another matter. For example, 1024x1024 (512x512) pixel picture files are 1.0 (0.25) megabytes each so that a 20 gel image database is about 20 (5) Mbytes. Since additional variants of the gel images will probably (although not necessarily) be generated during analysis, this space should be tripled to leave plenty of room. The uncompressed GSF and GCF files typically take about 100 Kbytes each, depending on the number of spots. A 20 gel PCG composite gel database file with 600 Rspots and 500 eRspots would take about 6 Mbytes. Note that after the GCF files are produced, the GSF and

segmented gel image files may be deleted or backed up to tape. Similarly, after the PCG DB is constructed, the GCF files may also be backed up and deleted.

You should always make permanent backup tapes of your primary data consisting of gel scanned images (*.ppx*), accession file (*.id*), landmark database (*.lm*) and batch (*.do*) scripts created by **getacc** or **makjob**. GELLAB-II under UNIX permits placing gel images on the "picture disk" and GSF, GCF and PCG files on the "auxiliary disk" or "temporary picture disk". Use of a canonical "averaged" gel called Cgel′ instead of several replicate gels tends to reduce these space requirements, since one synthetic gel may be used instead of n replicate gels. To save space, it is possible to *compress* of most intermediate data files `.gsf, .gcf, .ppx,` etc. - and then automatically uncompress them when needed (using UNIX *compress(1)* and *uncompress(1)*). This can save from (30% to 90%) of file system space. Some programs, **sg2gii** and **cmpgl2** allow you to specify a `-compress` switch to cause generated files to be compressed. **cgelp2** is able to read compressed GCF files. For more information on disk space requirements, see Section 6.1.

1.6.5 The user's GELLAB 'state' file: `gel.rc`

The GELLAB-II state file `gel.rc` contains the names of a number of other files as well as database file paths. These are indicated by a *keyword=value* syntax in the `gel.rc` file. In addition, you can leave additional keyword-value option entries for use by programs which might wish to use them. Both the `gellab` directory and `gel.rc` should be put in each project directory (denoted here as `/home/joeUser/project` Alternatively, if you wish to copy a default `gel.rc` you have set up in your home directory, the word `HOME` can initially be substituted for the `/home/joeUser/project` occurrences in the following items and the first time the **pgelrc** program is run it will replace all `HOME` entries with the value of your UNIX `$HOME` environment variable.[11] `gel.rc`

The `gel.rc` file includes the following entries which default to the following paths.

The accession database file:

```
gelFile= /home/joeUser/project/gellab/id/gel.id
```

The landmark set database file:

```
lmsFile= /home/joeUser/project/gellab/lms/lms.lm
```

The (future) annotation database file:

```
annDBfile= /home/joeUser/project/gellab/ann/ann.ann
```

The default `gellab` file paths are specified by the `ppnp`*n*`x` path keywords. *default parameters*

[11] This is the case in the sample `gel.rc.USER` and `gel.rc.DEMO` files found in directory ~`gelmgr/gellab/demo`.

The original gel picture disk path:

```
ppnp1x= /home/joeUser/project/gellab/ppx/
```

The auxiliary picture disk path:

```
ppnp2x= /home/joeUser/project/gellab/aux/
```

The temporary picture disk path:

```
ppnp3x= /home/joeUser/project/gellab/tmp/
```

The PCG composite gel database path:

```
ppnp4x= /home/joeUser/project/gellab/pcg/
```

The PCG generated derived files path:

```
ppnp5x= /home/joeUser/project/gellab/gen/
```

The Original (non-ppx) gel image files path:

```
ppnp6x= /home/joeUser/project/gellab/gen/
```

Note that the original gel images are kept on `ppnp1x`. The *.gsf* and *.gcf* files as well as derived Rmap, mosaic and segmented gel images are saved in the `ppnp2x` directory. Temporary images generated by **sg2gii** are saved in `ppnp3x`. The **cgelp2** PCG DBs are saved in `ppnp4x` while **cgelp2** generated derived data files are stored in `ppnp5x`. If the original images are not PPX files, they may optionally be save in `ppnp6x`. Some of the other `gel.rc` state keywords include:

The 3 character alphanumeric project prefix name:

```
projectPrefix= XXX
```

The current representative gel:

```
Rgel= XXXX.E
```

The current scanner image file pixel size in microns:

```
PixelSizeMicrons= 176
```

The current PPX image file number of rows in pixels:

```
PpxNrows= 1024
```

The current PPX image file number of cols in pixels:

```
PpxNcols= 1024
```

The **sg2gii** spot area sizing in pixels:

```
SG2areaLimits= A1,A2
```

The **sg2gii** spot density sizing in integrated OD:

```
SG2densityLimits= D1,D2
```

The **sg2gii** spot OD range sizing in OD:

```
SG2odRange= O1,O2
```

Keyword-value option entries

Additional options may be specified in the `gel.rc` file. Options which are used by multiple programs use the word `option`. The `option` or program name, followed

by a '.'. This is followed by a specific option keyword followed by a ':'. The values, if any, for that keyword follow. Note that entries specific for particular programs are indicated by the program name rather than the keyword `option`. Comments are indicated by a '#' prefix. Current entries include:

```
#
# Note: add '#' in front of option to ignore it.
#
option.COMPRESS: yes - do it
#option.NOCOMPRESS: NO - don't do it
option.DISPLAY: LASER
#option.DISPLAY: PLOT
option.LASERPRINTER: laser
#
Xpix.switches:
Xpix11.switches:
accppx.switches:
  . . .
sg2gii.switches: -3x3 -BUSSE:3:C -SAT:99.7 -BACK:64 -RESTOFGEL -CCMIN:4\
                 -DRAWSPOTS:PO -CH:A:25,10000,D:0.001,10000,O:0.001,4.5
  . . .
wmwait.switches:
```

If you see strange behavior in that files are not found or are not where you expect them to be, then make sure that your `gel.rc` file is properly defined. See Section 3.13, page 407 for details on defining or changing `gel.rc`.

1.6.6 The gel accession database file (`gel.id`)

Gels are referenced by an *accession number* at the time they are entered into the system using the **getacc** program. The accession number, ACC#, is a 4 digit number *XXXX* with a 1 place fraction *E* denoted *XXXX.E* assigned by the user to a particular exposure of a gel at the time it is entered (accessioned) into the system. If different exposures are made of the same physical gel the *XXXX* part would be the same but the *E* part would differ, eg. `01234.1, 01234.2,` etc. The *E* value of `9` is reserved for synthetic Cgel′ data files generated by **cgelp2.** The default value of *E* is '1' (e.g. `324` is `0324.1`). The accession file contains an initial data dictionary in the first 5 lines followed by sequential accession entries.

accession number

Format of Data Dictionary

The data associated with each gel is specified in the accession file which has a `gel` prefix and an `.id` file extension. Typically the 3 character "project-prefix"

is appended to the word `gel` as in `gelts3.id`. The name of the gel accession file database is in the name `gelFile` field of the `gel.rc` file. It will default to `./gellab/id/gel.id` in the current project directory. Notice that the accession number indexes the accession file and that the corresponding picture file name is in the accession file record for each gel. After the five line *Data Dictionary* (DD), each gel accession data record is four lines. The first three DD lines of the file define the record field DD descriptors which are separated by "/" and terminated with a "*". The 4th DD line defines the optical density (OD) values of the particular calibrated neutral density (ND) wedge (or counts/minute (CPM) standards) scanned with each gel in the database in order to map the gray value pixel values in the original image to integrated OD or CPM. This is followed by the *computing window* (CW) which specifies the active region in the gel image in raster coordinates. [(x, y) of $(0, 0)$ is the upper left hand corner, and (ROW_{max}, COL_{max}) is the lower right hand corner. There are 15 ND step wedge entries and 4 computing window values and the image size in pixels. The 5th line defines an associated *annotation* database file if it exists (no longer used).

data dictionary

Data record instances

Then, each *accession entry* record which follows the DD is four lines, the first 3 of which are data which has a 1-to-1 correspondence with the DD. The fourth line of a data record is the set of gray values corresponding to the ND wedge for each OD value in the ND standard step wedge. If no peaks were found for the high ND wedge values, the list terminates with 0 values. The last six numbers of the same line are the computing window subregion for that gel $[x1 : x2, y1 : y2]$ and the [rowsXcolumns]. A typical accession file is listed here. In the entry for gel `0094.1`, $[x1 : x2, y1 : y2]$ = `[47:441,100:374]`, and $[rowsXcolumns]$ = `[512,512]`.

OD calibr.

CW calibr.

Example of accession file: gelts3.id

```
ACCESSION#/PATIENT/BIR.DATE/RACE&SEX/EXP.DATE/EXP#/CULT.REAG/AMPH,GEL/
INTRVL BEFR LBLNG/LBLNG ISOTOPE/DURTN LABEL/DURTN OF EXPSR/STUDY/
PPX FILE#/TAPE #/OPT. BACKUP TAPE #/CAMERA,LENS,DISTANCE/EXPERMENTER*
ND:.05,.20,.35,.50,.66,.80,.95,1.10,1.25,1.41,1.56,1.72,...,2.17 CW[X1:X2,Y1:Y2] rows,columns
ANNTS3.ANN IS THE ANNOTATION DATABASE FILE
0324.1/.../?/?/1-18-82/#12/CULT #2/3:10,5-20%/
0 HRS/H3/2 HRS/96 HRS/HEME MALIG-AML,MYELOID/
B00661/-NONE-/-NONE-/VIDICON-AUTO,28MM,F8,76CM/LESTER*
 027 049 072 095 117 136 153 168 181 192 200 208 213 220 225 000 036 497 074 509 512 512
0369.1/.../?/?/3-10-82/#2/CULT #2/3:10,5-20%/
0 HRS/H3/2 HRS/96 HRS/HEME MALIG-ALL,LYMPHOID/
B00889/-NONE-/-NONE-/VIDICON-AUTO,28MM,F8,76CM/LESTER*
 028 051 075 098 118 138 155 169 183 192 200 208 215 225 229 000 061 505 068 503 512 512
0378.2/.../?/?/3-23-82/#1/CULT #1/3:10,5-20%/
0 HRS/H3/2 HRS/168 HRS/HEME MALIG-CLL,LYMPHOID (DUPL. SCAN)/
B00945/-NONE-/-NONE-/VIDICON-AUTO,28MM,F8,76CM/LESTER*
 027 051 073 097 117 137 154 168 182 191 200 208 214 223 000 000 025 482 067 509 512 512
0384.1/.../?/?/1-18-82/#17/CULT #1/3:10,5-20%/
0 HRS/H3/2 HRS/720 HRS/HEME MALIG-HCL,LYMPHOID/
B00981/-NONE-/-NONE-/VIDICON-AUTO,28MM,F8,76CM/LESTER*
```

```
 023 047 071 095 114 135 153 167 181 191 199 208 213 223 000 000 048 500 066 502 512 512
0396.1/HL-60/?/?/11-19-81/#5/CONTROL-4-EXP16/3:10,5-20%/
72 HRS/H3/2 HRS/264 HRS/HL-60 HUMAN MYELOID DIFFERENTIATION/
B01045/-NONE-/-NONE-/VIDICON-AUTO,28MM,F8,76CM/LESTER*
 016 041 063 088 108 129 147 162 176 187 196 204 211 223 000 000 175 486 077 509 512 512
0497.1/-#1/?/?/11-23-82/#13/NONE/3:10,5-20%/
0 HRS/H3/2HRS/168HRS/HEME MALIG-AML MYELOID/
B01577/-NONE-/-NONE-/VIDICON-AUTO,28MM,F8,76CM/LESTER*
 017 047 075 100 121 141 158 170 182 191 198 204 215 000 000 000 123 509 071 509 512 512
0503.1/...-#2/?/?/2-16-83/#10/NONE/3:10,5-20%/
0 HRS/H3/2HRS/166HRS/HEME MALIG-AML MYELOID/
B01601/-NONE-/-NONE-/VIDICON-AUTO,28MM,F8,76CM/LESTER*
 025 055 080 105 125 143 159 171 182 191 197 203 215 000 000 000 121 478 097 481 512 512
0511.1/...-#2/?/?/12-07-82/#9/NONE/3:10,5-20%/
0 HRS/H3/2HRS/168HRS/HEME MALIG-ALL,LYMPHOID/
B01633/-NONE-/-NONE-/VIDICON-AUTO,28MM,F8,76CM/LESTER*
 012 042 069 095 118 138 155 168 180 189 196 203 215 000 000 000 164 509 073 509 512 512
0514.1/...-#2/?/?/5-13-82/#12/NONE/3:10,5-20%/
0 HRS/H3/2HRS/167HRS/HEME MALIG-ALL,LYMPHOID/
B01645/-NONE-/-NONE-/VIDICON-AUTO,28MM,F8,76CM/LESTER*
 012 040 068 095 117 137 154 167 179 188 196 203 215 000 000 000 111 505 075 509 512 512
0515.1/...-#3/?/?/5-13-82/#15/NONE/3:10,5-20%/
0 HRS/H3/16HRS/167HRS/HEME MALIG-CLL,LYMPHOID/
B01649/-NONE-/-NONE-/VIDICON-AUTO,28MM,F8,76CM/LESTER*
 023 054 079 104 125 143 159 171 183 191 198 204 215 000 000 000 080 509 069 509 512 512
0517.1/...-#4/?/?/5-13-82/#10/NONE/3:10,5-20%/
0 HRS/H3/16HRS/24HRS/HEME MALIG-CLL,LYMPHOID/
B01657/-NONE-/-NONE-/VIDICON-AUTO,28MM,F8,76CM/LESTER*
 032 061 087 111 129 148 163 174 185 193 200 203 215 000 000 000 115 509 074 509 512 512
0393.2/...-#2/?/?/3-23-82/#20/NONE/3:10,5-20%/
0 HRS/H3/2HRS/360HRS/HEME MALIG-HCL,LYMPHOID/
B01693/-NONE-/-NONE-/VIDICON-AUTO,28MM,F8,76CM/LESTER*
 030 059 083 108 128 146 161 173 184 193 199 204 215 000 000 000 129 500 078 504 512 512
       .
       .
       .
```

1.6.7 The landmark database file (`lms.lm`)

The landmark database (LM) file consists of a sequential file of landmark sets. Entries are generated automatically using the **landmark** program. The LMS file name is specified in a manner similar to that for the accession file. It has a `lms` prefix and an `.lm` file extension. The name of the landmark database file is in the `lmsFile` field of the `gel.rc` file. It will default to `./gellab/lms/lms.lm` in the current project directory. Typically the 3 character "project-prefix" is appended to the word `lms` as in `lmsts3.lm`. The default name is `lms.lm`. The format is illustrated by the following example. It consists of: a preface (4 lines starting with "LMS: PROTOCOL-VER# [rows,cols]=[...]" which name the LMS DB entry and its creation date; up to 52 alphabetic landmark entries starting with the keyword "LANDMARK # G1[x1,y1] G2[x2,y2]"; followed by a blank line and the "ELAPSED TIME" required by the user in the **landmark** program to generate the landmark set of spots. The entry [rows,cols] indicates the coordinate system for which the landmarking was done so that appropriate scaling to other domains could be performed. The accession numbers are indicated by the keyword "FROM

corresponding spot coords.

GELS". The names of the gels are coded in the names of the "GSF" files. For example the LMS of gels `324.1, 369.1` are encoded as `P10324.GSF, P10369.GSF`. These are the name prefixes of the corresponding GSF files. The "CREATION-DATE:" is the time stamp when the LM entry was added. The following example shows two landmark sets in the front of the landmark DB file.

```
/ LMS: PROTOCOL-VER# 9-28-88 [rows,cols]=[512,512]
/ INTO /home/joeUser/project/gellab/lms/lmsts3.lm FROM GELS: 0324.1,0369.1
/ GSF: p10324.gsf,p10369.gsf
CREATION-DATE: 010/14/1988,  03:26:00 PM
 LANDMARK #A  G1[304, 190], G2[284, 176]
 LANDMARK #B  G1[335, 151], G2[315, 140]
 LANDMARK #C  G1[353, 190], G2[333, 173]
        . . .
 LANDMARK #V  G1[226, 417], G2[196, 392]

 ELAPSED TIME:      939. SECONDS

/ LMS: PROTOCOL-VER# 9-28-88 [rows,cols]=[512,512]
/ INTO /home{annotation ! DB file ! fields examples\&defs}
```

1.7 Benchmarks

There are two main areas for evaluating the time performance in a gel analysis system. One is the efficiency of the processing programs and the other is the interactive time required by an operator to use the system. Both are important. The processing time is mainly a function of the computer hardware while the interactive time depends on how far one goes with the exploratory data analysis.

CPU time benchmarks

measuring time

Central Processor Unit (CPU) time is a main measure of computer resources. Most CPU time required in analyzing a set of gels with GELLAB-II is spent primarily in segmenting gels. Run time (denoted RUN here) is the elapsed daytime required for an operation. Both times should be taken into account when evaluating benchmarks. Benchmark times of course vary between processors. The percentage of the time spent by the CPU processing your particular program is CPU/RUN time. This reflects both the load of other users on your computer as well as overhead accessing disk files. If no one else is using the computer, then RUN time is an effective benchmark measure (since most computers have comparable overhead). Most GELLAB-II programs terminate their operation by printing statistics on all three time measures. For example the following times (for segmenting a 512x512 pixel gel image with over 800 spots on a SparcStation2-32Meg with no other users but running X-Windows) indicate that UNIX overhead was about 2% indicating

that the program is compute bound. [12] For example on the SparcStation-2, for a gel with 800 spots:

```
Real TIME =00:01:21  CPU TIME =00:01:17,  95.06%
```

And on the Convex super-computer run time with some optimization,

```
Real TIME =00:00:21  CPU TIME =00:00:19,  90.48%
```

On the SparcStation-2, the segmentation (with all **sg2gii** options commonly used) of a gel with about 1000 spots takes about 1 to 2 minutes for a 512x512 pixel image and about 3 to 4 minutes for a 1024x1024 pixel gel image with over 1700 spots - although times depend on the number of spots in the gel as well as how noisy the gel is. A **cmpgl2** gel comparison takes about 20 to 130 seconds (for 800 to 2000 spots/gel). Building a PCG database with **cgelp2** takes about 1 minute/gel. Generating a Rmap image takes on the order of 5 seconds, while for a mosaic image, it is proportional to the number of panels (about 3 seconds per panel).

interactive time

Since all of the segmentation, gel comparison, database building and initial database search may be run under UNIX background 'batch', gel processing can precede in parallel with other work. A *script* in UNIX is a sequential list of UNIX top level commands normally typed by the user to run programs. This corresponds to a batch job file on most other computer systems (e.g. like a "COM" file in VMS or "BAT" file in MS-DOS for example). On a UNIX system, batch processing is equivalent to putting a script being processed into the background. That it is executing as a background process. The **cgelp2** program reports the total session time. For example, the total time to run the complete script to construct the PCG DB and run about twenty tests, compute histogram plots of the PCG DB features as well as recording the preliminary results can give you a rough estimate of the speed of the PCG DB analysis. For the twelve demonstration gels *on the same disk as the computer* (see `ts3cgl.gdo`, page 3.19) on a SparcStation-2 the total time is:

```
SUN4:
   Total session times: Real TIME =00:25:05  CPU TIME =00:19:55,  79.40%
SUN3:
   Total session times: Real TIME =00:41:48  CPU TIME =00:33:11,  79.39%
```

On a SparcStation-2 with the database on the local Sun 1.3Gbyte disk and the GCF files on another disk mounted on NFS, the run total time is:

```
   Total session times: Real TIME =00:22:49  CPU TIME =00:15:57,  69.91%
```

[12] CPU times for GELLAB-II programs given throughout the book are for the un-optimized versions. That is, the C code was *not* optimized by the SUN CC compiler - nor was the faster GNU `gcc` C compiler used. This will be true for the *Beta* release of GELLAB-II as debugging information is included in executable files which is incompatible with the C compiler `-O` optimizer.

On a Convex super-computer C200 the total run time time is:

```
Total session times: Real TIME =00:07:12  CPU TIME =00:05:11,  71.99%
```

Interactive time benchmarks

Time spent by the investigator in interactive operations can be divided into the time required to construct the composite gel PCG DB and that required for further exploratory data analysis.

During initial gel image data acquisition using **getacc**, answering the acquisition questions regarding the gel samples and interactively defining the gel computing window and ND wedge window and evaluating its calibration for the accession file takes 1 to 2 minutes/gel (exclusive of camera scanning).

Manually landmarking a pair of gels takes from 3 to 30 minutes depending on the comparability of the gels with an average time being about 3 to 10 minutes. These times depend on gel quality (the single most important factor) and the set of landmarks selected (which must be in all gels to be compared). Landmarking the initial pair of gels takes somewhat longer since the investigator must decide which set of spots to use for landmarking all of the gels.

After the initial **cgelp2** batch file is finished, the investigator should evaluate the resulting log file listing and check the initial Rmap and mosaic images of some of the initial searches. This can be done in 1 to 3 hours. At this point however, the time required to pursue the exploratory data analysis is a function of what is found and what the investigator is trying to find and so is difficult to estimate.

spot & gel capacity

Because large databases take increasingly longer time to search there are "tricks" which can be used to help reduce this time. The **cgelp2** program capabilities include being able to handle up to 65,000 Rspots for up to 8000 gels as well as other enhancements. [13] The database can handle 4Kx4Kx8-bit size images - however the Datacopy scanner can scan up to a 1728x2810 pixel image and other scanners up to 4096x4096. A size of 1024x1024 is adequate for most gels. The **cgelp2** program currently is setup to handle upt to 16253 Rspots for 128 gels but can be recompiled to handle more spots and or gels at the expense of using more memory, disk space and and time.

Only the gel spot segmentation requires directly manipulating images with this high image resolution. This is necessary to ensure maximum accuracy for spot quantitation. After quantitating spots in gels using **sg2gii**, a smaller 512x512 image may be used for subsequent visual display since it is accurate enough to visually

[13] You would need a "very" large disk as well as fast computer to handle a database for a large number of gels with this maximum resolution. A better way to handle such data would be to use averaged canonical Cgel′ gels which would drastically reduce the size of data required to be kept on line.

locate spots of interest when generating synthetic Rmap and mosaic images. So averaged 512x512 images derived from the original higher resolution images are saved and used in the exploratory data analysis while the higher resolution images are optionally archived onto tape (see Section 6.9, page 550).

using Cgel′ to save space

In practice, there also are better ways of handling large numbers of gels than creating very large databases. One can create *Cgel′* synthetic gels which are averages of subsets of gels so extremely large databases with corresponding heavy computational and storage requirements need not be faced (c.f. [LemP82a] and Section 3.3.10, page 192).

1.8 GELLAB caveats - status as of 7-1-92

GELLAB-II is in *Beta* test phase which means that not all of the programs are working 100%. [Actually no program *ever* work 100% - not even "star wars" or the phone system]. This section lists some of the known major problems. These are some of the known caveats in the reimplementation of GELLAB-II from GELLAB-I. As they are fixed, they will be removed from the following list.

autopair automatic spot pairing program is being tuned and is not currently released.

cgelp2-plot-functions are not always optimally scaled.

cgelp2-SET SRL SUBSETS//FINDKEYWORD works only with a single term. The boolean expression of terms does not work.

cgelp2-LMN The *Local Morphologic Landmarks* LMN are partially implemented in **cgelp2** but also have to be added to **autopair**. See [LemP88d] for discussion on LMN.

cgelp2-capacity The current capacity (set when GELLAB-II is compiled) is: 16253 Rspot sets, 128 gels, 512x512 (or 1024x1024) pixel images, 52 manual landmarks/gel. These limits can be increased by changing the C code macro constants and recompiling all of GELLAB-II. The absolute upper bounds are 65,000 Rspot sets, 8,128 gels, variable size images 512x512 to 4096x4096 pixels. In a future release, there will be an unlimited number of **autopair** produced *landmarks* per gel.

cgelp2-exceeding-maximum-number-Rspots Currently, if you try to build a PCG DB which would contain more than the current maximum number of spots (16254), then you will be told this by it's printing a DRYROT error. This may be gotton around by increasing the size of spots being considered. Most of these "noise" spots are at the lower limits of resolution and of very low

density. Increase the lower bounds of area or density using `SET STATISTICS` and rebuild the database using the *prj*`cgl.do` batch job.

cgelp2-GELS/FULL does not report all **sg2gii/cmpgl2** sizing parameters correctly since that part of the **cgelp2** CGF file parser is not fully enabled.

cgelp2-SET-CALIBRATION has not been fully tested.

cgelp2-history still has a minor problem when deleting certain entries from the history list.

cgelp2-SET-PARAMETERS-SUBSET does not include all of the PCG DB state at this time. Some of the SET commands (`SET RATIO LIST, SET LEAST SQUARES CALIBRATION, SET ANNOTATION, ...`) are not part of the PARAMETERS subset.

cmpgl2-Composite-Spots are no longer used. We will be able to generate these interactively with the GSF spot editor being developed.

SMOMNI-line-graphics OMNIGRAPH emulator is not fully operational. PPX, andPostScript modes have not been throughly tested or debugged. However, it correctly generates X-Window and Tektronix 4010 output which can be plotted directly on a 4010 terminal or emulated terminal as well as with the **tek2psG** filter program in which converts 4010 output to PostScript. This affects programs **cgelp2, dwrmap, plotn, dendrogram**.

sg2gii-1Kvs512-symmetry Currently there is not complete agreement between using the 512x512 sampled version of a 1Kx1K gel when segmented (as would be expected). The `sg2gii -stdppx -7x7lowpass -laplace5x5` generally works better on the sampled 512x512 gel image. As expected, because of the lower resolution, occasionally some spots are merged which should be able to be split at the higher 1Kx1K resolution. While the `sg2gii -1kx1k -13x17lowpass:1.0 -busseLaplacian:3` which should be equivalent on a larger grid size does not always seem to do as well - with some spots not being adequately detected. For some Bioimage iso-dalt size gels scanned at 1Kx1K, the following parameters seem to work well in detecting very small spots: `sg2gii -3x3Lowpass -backgroundFilterSize:26`

Xpix-caveats does not work completely for all of the menu options. We suggest using **Xpix11** under MIT X11/twm and **Xpix2** under OpenLook - although it also works under MIT/twm. See the **Xpix** caveat list on page 498.

Chapter 2

GELLAB-II Tutorial

A GELLAB analysis of a set of 2D gels is a data reduction process. It analyzes a set of gels of the same general type of material but with different experimental conditions to produce lists of spots for each gel. These spot lists are combined into a single composite database. In turn, the composite database can be searched to find subsets of spots which change as a function of experimental variable. Furthermore, these subsets of spots can then be clustered using various additional statistical techniques. This Chapter presents the overall process of gel analysis using a step by step tutorial. It should be pointed out that the tutorial is designed to instruct you on the many possibilities available in analyzing gels. Normally, one would not need to spend anywhere near that amount of time or error in doing a routine analysis.

step by step ...

Section 2.1 page 74 shows two demonstration UNIX scripts used to generate some processed images and then run a "slide" show with the images. If you choose to run this `slides11.do` demonstration, you will see the results of part of a gel database analysis. Section 2.2 page 93 gives examples of invoking individual GELLAB-II programs and its subsections are organized by "next step required" in performing the analysis. Section 2.3 page 105 gives examples of **cgelp2** command sequences one might use during composite gel construction and exploratory data analysis. Section 2.4 page 129 shows a small part of an exploratory data anaysis session which has been captured from the terminal and annotated to make it easier to follow. But first, we will review the basic steps in performing a gel analysis.

lots of examples

A set of 12 adult human leukemia lymphocyte gels (gratiously supplied by Eric Lester) are used in all of the following examples. By exercising the examples, you will become more familiar with the data which will then greatly facilitate performing the exploratory data analysis. These gels were accessioned on the old GELLAB-I RTPP Vidicon system, with pIe reversed with "acid on the right" instead of the current convention with "acid on the left" (See Figure 1.1, page 18). However, this is not a problem as GELLAB works equally well with any orientation of pIe or

sample dataset

MW when they are displayed. See Section 2.2, Examples 13 and 14. for generating Rmaps and mosaics which reorient the gels in pIe orientation. These gels and associated database files may be found in the ~gelmgr/demo directory.

In the tutorial examples, data is precomputed so you can skip any example you want without having a problem with later examples.

Steps in performing a gel analysis

The four major steps in performing analysis, illustrated in Figure 1.3 page 43, are: (1) accessioning (gel scanning), (2) spot quantitation, (3) gel pairing, (4) composite gel DB construction with subsequent searching and displaying of search results of different views of the DB. Unlike the earlier stages of gel analysis, this last phase of exploratory data analysis does not always have a well defined computational path - and can branch out in different directions. Briefly, the first four steps of data reduction are achieved as follows:

building a database

1. **Accession** a set of gels by scanning them with the **getacc** program (Figure 1.3 `step:1`) which: (a) assigns an accession number to each gel, (b) scans the gels into *Portable PiXture (.ppx)* image files, and (c) requires the experimenter to enter associated experimental study accession information, and (d) calibrate the ND step wedge scanned with the gel and define an active region in the gel image called the computing window. The information from (c) and (d) is entered into an accession file. This accession file typically has a *gel* prefix and a *.id* file extension. E.g. `gelts3.id`. During a data acquisition session, one would enter a number of gels and at the end of the session **getacc** would prompt you for a few pieces of information necessary for further processing. These include: (a) the name of the *Reference gel* or *Rgel*, (b) a three character project prefix used for all files associated with the project. and (c) the names of the different experimental classes to which the different gels belong. It will then generate UNIX batch scripts to: interactively landmark these gels (Figure 1.3 `step:2`) ; segment or extract quantitated lists of spots from the gels; pair $n-1$ of the n gels with the selected Reference gel; and construct the composite gel database and perform some initial statistical tests. The **makjob** program also lets you generate these batch scripts for different sets or subsets of gels which have been previously accessioned.

data

2. **Spot-list extraction and quantitation** is performed by the **sg2gii** program (Figure 1.3 `step:3`) which results in a *Gel Segmentation File (.gsf)* and an optional extracted spot image file. The GSF file contains position and quantitation information for all spots in a single gel and must be further processed to compare it with other gels.

quantitation

3. **Pairing of GSF spot lists from two gels** (one of which is the Rgel) is performed by the **cmpgl2** program (Figure 1.3 `step:4`). The output is called a

Gel Comparison File (GCF) and is a file with a *.gcf* file extension. It consists of the pairing of spot data from the two GSF input files. The pairing program also requires a list of a small number of corresponding landmark spots for the two gels being paired. This is stored in the landmark (LM) database file which typically has a *lms* prefix and *.lm* file extension (e.g. `lmsts3.lm`). LM DB data can be acquired several ways: (a) using the **landmark** interactive graphics program running under X-windows, (b) using program **dwrmap** to draw Rmaps from GSF data which can be plotted with **plotn**. The landmark numbers can be read manually from the plots and then entered via a terminal session using the `-lmsedit` option with the **landmark** program. A third method involves using the **Xpix** program in its "compare" mode to interactively generate the landmark coordinate pairs which can then be text edited into the proper LM DB format. Needless to say, when X-Windows is available, the use of the **landmark** program is to be encouraged as it is by far the easiest and most accurate.

gel pairing

4. **Construction of the Paged Composite Gel DB** (PCG DB) is performed by **cgelp2** (Figure 1.3 `step:5`). **Cgelp2** requires a set of $n-1$ GCF files for n gels since the Rgel is included in each GCF. By "paged" we mean that as the database is too large to fit in memory, pieces of it are brought in and out of memory from the actual PCG DB disk file (which has a *.pcg* file extension (e.g. `ts3pcg.pcg`). The accession file is also accessed to extract the "study" information for each gel in the PCG DB. This information is used for automatically classification of gels into the current experimental classes of gels and other analyses.

   composite gel database

   Exploratory data analysis really starts once the PCG DB is constructed. The particular strategy to follow will depend on the type of gel data that is brought to GELLAB. Different strategies are discussed throughout this Chapter, Chapter 5.1, page 503 as well as in many of the GELLAB papers listed in Section 1.2 page 23.

   exploratory data analysis

Using batch processing to analyze gels

The above steps [2] to [4] can also be done in UNIX background batch. [1] The initial batch script generated by **getacc** or **makjob**: (a) constructs the PCG DB file, (b) constructs an initial experimental gel classification based on accession file study information, (c) normalizes the protein concentration values between gels using the Ratio-List method and reorders spots in *all* Rspots sets in the PCG DB based on

automatic processing

[1] "Background batch" is a mode of computer operation whereby all processing for a specific set of computations which are declared to be batch processed are done in the background without any user involvement. This means that the computer can then be used for doing other things in the "forground".

this normalization. (d) It computes ordered expression profile similarity tables for the landmark spots for thresholds of 1.0, 0.5, 0.25 and 0.1. Then it (e) then performs an initial F-test at p-values of 0.90, 0.95 and 0.99 for all of the experimental classes. It then does a t-test at p-values of 0.90, 0.95 and 0.99 for experimental classes 1 and 2. Finally, it performs a Wilcoxon-rank-sum test of classes 1 and 2 at p-values 0.90, 0.95 and 0.99 for classes 1 and 2 as well as a missing-class test. (f) It also computes and displays histograms of various Rspot set spot features for the entire PCG DB in order to aid in setting the initial prefilter parameters. And finally (g), it plots Rmaps and mosaics of the landmark spots.

Subsequent analyses consist of changing the *view* of the PCG DB, performing searches in the new view and displaying this transformed data as images, plots, tables, lists, etc. See ([LipL80a], [LemP82a], [LemP84b], [LemP88d], [LemP89a]) for more discussion on using GELLAB-II for exploratory data analysis. Section 2.3 page 105 gives step by step examples of some of these tasks in **cgelp2**.

We will be seeing examples of all of these data reduction and analysis steps in the following tutorials. By referring back to this "master plan" you will begin to understand the structure of an analysis.

2.1 GELLAB-II demonstration scripts

An interactive demonstration of the processing and display aspects of GELLAB-II is done using some scripts. It assumes that you have created a `gellab` environment in *your* login area. If you have not, then you can set it up as follows *after* you have logged into your account. First, if not already defined in your default UNIX shell startup file, you must define an "environmental variable" called `GELLABMANAGER` to the where GELLAB "lives" [the distributed version of GELLAB-II assumes `~gelmgr`, but you should set it whatever it is for your system]. You might wish to add the following line to your `.cshrc` file in your home directory so you don't have to type this every time you log in. This may have already been done by the person administering GELLAB, so it may not be necessary. In fact, if they have set up your account you may be able to skip to step `3%` or `5%` below.

GELLAB manager

```
1% setenv GELLABMANAGER ~gelmgr
```

Where: `~gelmgr` is your system's account where the executable GELLAB-II programs are to be found. [If the GELLAB files "live" elsewhere, then replace `gelmgr` with the name of that account].

At this point you need to create your `gellab` directories and and so that they can access the demonstration files to *your* `~/gellab/demo` area. If this has been done previously by the GELLAB manager, you can skip to step step `3%`. More details on installing your `gellab` and `demo` directories are discussed in Section ??, page ??.

making a new user directory

The actual demonstration consists of two parts: creating the slides and then showing them. If you wish to recreate the slides, run the following script to create the images for the slide show by doing:[2]

simple demonstration

```
3% cd ~/gellab/demo
4% make-slides.do
```

As the "slides" are precomputed on the distributed GELLAB, you can also skip this step if you wish and just run the slide show in steps `5%` and `6%`. Note, you must be in `~/gellab/demo` to run the `slides11.do` or `demo11.do` scripts. The `demo11.do` runs some of the GELLAB-II programs including **getacc11** and **landmark11** to illustrate some of the data acquitision aspects of the system.

```
5% cd ~/gellab/demo
6% slides11.do
or
6% demo11.do
```

If your UNIX login account was already set up by your GELLAB-II system manager, you can ignore Section 2.1.1.

2.1.1 Creating a new gel project `gellab` directory tree

The **pgelrc** program can be used to create the required user's GELLAB directories and files for a *project* if they wish to keep separate different projects in the same user account. It first checks to see if the directories exist and, if not, then it creates them. You should first create a project directory to put the project files in as follows. For example, let the project directory be called `prj1`. Then type the following UNIX commands and respond the terminal prompts. If you are not sure of the response, just type RETURN-key. When it is finished, the account is setup with `gel.rc` and the `gellab` directory created as well as new accession and landmark database files if needed.

creating a new gel project

```
1% mkdir prj1
2% cd prj1
3% pgelrc
```

[2] the "slide show" demonstration assumes that X-Windows has previously been started (see Appendix D, page 603). If your directory has been set up correctly, just type `xini` at UNIX command level. Later, when you are wish to log off of the computer, type `CONTROL/D` in the upper left hand terminal command window to "kill" X-Windows. Then type `CONTROL/D` again to log you off of UNIX.

2.1.2 Making "slides" and then showing them

The first script, `make-slides.do`, creates a set of derived gel images which the second script, `slides.do`, displays. It assumes that the SPSS data file `ts3s02.sps` (included in the `demo` directory files) was previously created from the PCG DB `ts3pcg.pcg` using **cgelp2**. See Section 6.6 page 547 for examples of invoking these scripts. You can skip directly to `slides11.do` since these images were already computed and exist in the `demo` directory.

MAKE-SLIDES.DO: Creating a set of derived gel images

```
#!/bin/csh -v
echo "make-slides.do script for demonstrating GELLAB-II"
# P. Lemkin
# It creates images for use with slides.do script.
# The make-slides.do script may be run either interactively or in
# background batch. It assumes that the '~gelMgr/gellab/demo/gen/ts3s02.sps'
# file has previously been created from the PCG DB.
#
# $Date:$ / $Revision: $
#
#
cd ~/gellab/demo
pwd
echo "NOTE: make-slides.do is running on the [" `arch` "]."
#
echo " "
echo " NOTE: MEANINGS OF PREFIX OF PPX FILES."
echo " [a,b] original gel images."
echo " [c] propagated central core image in  sg2gii."
echo " [f] flipped image (pIe or MW reversed from original) in  markgel."
echo " [g] GraphScale pseudocolor image generated by various programs."
echo " [j] averaged image in   sg2gii."
echo " [k] magnitude of 2nd derivataive image in   sg2gii."
echo " [l] Rmap landmark image in  cmpgl2."
echo " [m] Rmap image from SPSS file in  markgel."
echo " [n] notch filtered image."
echo " [s] standard 512x512 PPX size copy of original gel images."
echo " [t] change vector Rgel labeled paired-spot image in  autopair."
echo " [u] marked Rgel labeled paired-spot image in  cmpgl2 & autopair."
echo " [v] marked gel paired-spot image in  cmpgl2 & autopair."
echo " [w] mosaic image in  mosaic."
echo " [y] (original less the segmented spot) image in  sg2gii."
echo " [z] segmented image in  sg2gii."
echo " ------------------------------------------------- "

# ---------------------- DO THE DEMO-------------------------
#
date
echo " "
echo "1.  PRINT THE GELLAB STATE FILE "
```

```
pgelrc
#

echo " "
echo "2.  Create SEGMENTED z,c,y,n,k,j .ppx images"
sg2gii 384.1 -ctl -dots -laplace5x5
sg2gii 324.1 -ctl -dots -laplace5x5
sg2gii 324.1 -ctl -restOfPPX -dots -laplace5x5
ls -l tmp/*.ppx*
#
echo " "
echo "3.  RMAP OF LANDMARK SPOTS - l images"
cmpgl2 324.1 384.1 -onlyMarkLMS
#
echo " "
echo "4.  PAIRED SPOT LABELS - u and v images"
cmpgl2 324.1 384.1 -MarkPairingLabels
#
echo " "
echo "5. CREATE LANDMARKED IMAGES FROM PCG DB - Rmap m images."
markgel 324.1 ts3s02.sps -large -graphscale
markgel 384.1 ts3s02.sps -large -graphscale
#
echo " "
echo "6.  MOSAICs of Landmark spot[c] Rspot 87 in PCG DATABASE"
mosaic 87 ts3s02.sps -graphscale
#
echo " "
echo "7.  MOSAICs of all Rspot for gel 324.1 in ts3s02.sps in PCG DATABASE"
mosaic ts3s02.sps -gels:324.1 -graphscale
#
echo " "
echo "**** THAT's ALL FOLKS  - Your slides are ready. ***"
echo "You may PRINT them on the laser printer with print-slides.do."
echo "You may SHOW them under them under X-windows with slides.do."

# ------------------ END OF SCRIPT -------------------
```

SLIDES11.DO: Showing a set of derived gel images

```
#!/bin/csh
clear;
echo "slides11.do script for demonstrating GELLAB-II."
date
pwd
# -------------------------------------------------------------
# P. Lemkin
# It assumes that the images were created previously with
# make-slides.do script This script MUST be run interactively.
#
# $Date: $ / $Revision: $
#
```

```
# [0] Fix up your local demo directory structure if it is required.
# ---------------------- DO THE DEMO-------------------------

# ------------------------------------------------------------------

echo "DEMONSTRATION OF GELLAB-II USING PRECOMPUTED IMAGES."
echo ""
echo "Press Sun 'Front' button when mouse is in a window to raise it."
echo ""
echo "
 ********************************************************
 *               WELCOME   TO   GELLAB-II               *
 *                                                      *
 *                                                      *
 *  DEMONSTRATION OF GELLAB-II USING PRECOMPUTED IMAGES. *
 *                                                      *
 *  Each example will print a short description prior to *
 *  displaying the image(s) for that example.           *
 *                                                      *
 *               Version: June 18, 1992                 *
 ********************************************************

[Press Sun 'Front' button when mouse is in a window to raise it.]
" > /tmp/demo11.tmp

echo "
DEMONSTRATION OF GELLAB-II USING PRECOMPUTED IMAGES

    Three leukemia patient lymphocyte gels will be used in
    the following examples (first two are in the first
    display). They are H3 flourographs scanned with a high
    resolution Vidicon into 512x512 pixel images (courtesy
    of Dr. Eric Lester). Gel accession number
         324.1 is a MALIG-AML (myeloid),
         378.2 is a MALIG-CLL (lymphoid),
         384.1 is a MALIG-HCL (lymphoid).
" >> /tmp/demo11.tmp

echo "
    The UNIX script   make-slides.do   was run previous to
    this one to have GELLAB process process these gels and
    construct the images being displayed here with the Xpix
    program.

=======>  Press QUIT button to continue.
" >> /tmp/demo11.tmp
cat /tmp/demo11.tmp | xless -geometry +528+1


# ------------------------------------------------------------------
echo "0.  Accessioning gels"
echo "
[Press Sun 'Front' button when mouse is in a window to raise it.]
```

```
0.  Accessioning gels.

   You normally accession a set of gels into a project directory
   by specifying it's ACCESSION #. You then either specify its PICTURE
   FILE name or direct the CAMERA program to scan it for you. The other
   accession fields may be specified by you or inherited from another
   previously accessioned gel.

CMD:      getacc

=======>  Press QUIT button when you finish this step.

" > /tmp/demo11.tmp
cat id/gelts3.id >> /tmp/demo11.tmp
cat /tmp/demo11.tmp | xless -geometry +528+1&

getacc -project ts4


# ------------------------------------------------------------------------

echo "1.  ORIGINAL GELS 324.1 (Rgel) and 384.1"
echo "
[Press Sun 'Front' button when mouse is in a window to raise it.]

1.  ORIGINAL GELS 324.1 (Rgel) and 384.1

    HINT: When in Xpix, you might try some of the Xpix
          program options such as the ZOOM in the CURSOR OPS
          menu.
" > /tmp/demo11.tmp

echo "
    NOTE: Xpix menu operation.
  a.  To get the five Xpix menus:
      hold down the keyboard CONTROL key while pressing and
      holding the MIDDLE button. Move the cursor to desired
      entry and release button to select it. To change the
      image contrast and brightness, MOVE THE MOUSE with
      MIDDLE button pressed (don't press CONTROL key).

  b.  Select the EXIT option in the VIEW OPS menu to leave
      Xpix and continue the show. This also holds for the
      following examples.

CMD:      accppx 324.1 384.1 -zoomRight -silent

=======>  Press QUIT button when you finish this step.
" >> /tmp/demo11.tmp
cat /tmp/demo11.tmp | xless -geometry +528+1&

accppx 324.1 384.1 -zoomRight -silent
```

```
# ------------------------------------------------------------------
echo "2.  ORIGINAL GEL and SEGMENTED SPOTS image for gel 324.1"
echo "
[Press Sun 'Front' button when mouse is in a window to raise it.]

2.  ORIGINAL GEL and SEGMENTED SPOTS image for gel 324.1

    GELLAB segments and quantitates spots using the
    properties of the second derivative. The resulant
    isolated spots are shown here. Other intermediate
    images used will be displayed in following examples.
" > /tmp/demo11.tmp

echo "
    HINT: Find a spot in the original image on the left
          and then look for the corresponding extracted
          spot in the right image.

    HINT: Adjust the Xpix image contrast as before and
          possibly use the zoom if looking at small spots.
" >> /tmp/demo11.tmp

echo "
    NOTE: menu operation.
  a. To get the five Xpix menus,
     hold down the keyboard CONTROL key while pressing
     and holding the MIDDLE button. Move cursor to desired
     entry and release button to select it. To change the
     image contrast and brightness, move the mouse with
     MIDDLE button pressed (don't press CONTROL key).

  b. Select the EXIT option in the VIEW OPS menu to leave
     Xpix and continue the show. This also holds for the
     following examples.

CMD:      accppx 324.1 324.1 -p2:z -zoomRight -silent

=======>  Press QUIT button when you finish this step.
" >> /tmp/demo11.tmp
cat /tmp/demo11.tmp | xless -geometry +528+1&

accppx 324.1 324.1 -p2:z -zoomRight -silent

# ------------------------------------------------------------------

echo "3.  ORIGINAL GEL and REST OF IMAGE"
echo "
[Press Sun 'Front' button when mouse is in a window to raise it.]

3.  ORIGINAL GEL and REST OF IMAGE
```

```
    The rest-of-image = (original gel less segmented spots).
    It is really an estimate of the mean background density
    in the gel and is used for that purpose in the next
    example.

    HINT: Adjust the Xpix image contrast to see the
          lighter background image.
" > /tmp/demo11.tmp

echo "
    NOTE: The darkest regions are those with high background
          in the gel or with 'leakage' from
          saturated spots.
" >> /tmp/demo11.tmp

echo "
    NOTE: menu operation.
  a. To get the five Xpix menus:
     hold down the keyboard CONTROL key while pressing
     and holding the MIDDLE button. Move cursor to desired
     entry and release button to select it. To change the
     image contrast and brightness, move the mouse with
     MIDDLE button pressed (don't press CONTROL key).

  b. Select the EXIT option in the VIEW OPS menu to leave
     Xpix and continue the show. This also holds for the
     following examples.

CMD:      accppx 324.1 324.1 -p2:y -zoomRight -silent

=======>  Press QUIT button when you finish this step.
" >> /tmp/demo11.tmp
cat /tmp/demo11.tmp | xless -geometry +528+1&

accppx 324.1 324.1 -p2:y -zoomRight -silent

# ------------------------------------------------------------------

echo "4.  ORIGINAL FOR 324.1 AND ITS BACKGROUND (NOTCH-FILTERED IMAGE)."
echo "
[Press Sun 'Front' button when mouse is in a window to raise it.]

4.  ORIGINAL FOR 324.1 AND ITS BACKGROUND
    (NOTCH-FILTERED IMAGE).

    After spots are extracted, they must be corrected for
    background density variation. This background image,
    computed from the previous 'gel less spots' image, is
    used to estimate the background density  for each spot
    by subtracting   'mnBackground(x,y)*Area'   for each
    spot.
" > /tmp/demo11.tmp
```

```
echo "
    HINT: Adjust the Xpix image contrast to see  the lighter
          background image. Possibly print out corresponding
          regions using PRINT REGION to see the relatively
          low magnitude of the slowly changing background
          function. The darkest regions are those with high
          background in the gel or with 'leakage' from
          saturated spots.
" >> /tmp/demo11.tmp

echo "
    NOTE: menu operation.
  a. To get the five Xpix menus:
     hold down the keyboard CONTROL key while pressing and
     holding the MIDDLE button. Move cursor to desired entry
     and release button to select it. To change the image
     contrast and brightness, move the mouse with MIDDLE
     button pressed (don't press CONTROL key).

  b. Select the EXIT option in the VIEW OPS menu to leave
     Xpix and continue the show. This also holds for the
     following examples.

CMD:      accppx 324.1 324.1 -p2:n -zoomRight -silent

=======>  Press QUIT button when you finish this step.
" >> /tmp/demo11.tmp
cat /tmp/demo11.tmp | xless -geometry +528+1&

accppx 324.1 324.1 -p2:n -zoomRight -silent

# ---------------------------------------------------------------------

echo "5.  PROPAGATED CENTRAL CORE IMAGES VS. ORIGINAL FOR 324.1"
echo "
[Press Sun 'Front' button when mouse is in a window to raise it.]

5.  PROPAGATED CENTRAL CORE IMAGES VS. ORIGINAL FOR 324.1

    During spot segmentation and extraction, an image based
    on the second derivative is used to find the centers of
    the spots. These 'central cores' are the light centers,
    'propagated central cores' the darker rings around the
    spots.
" > /tmp/demo11.tmp

echo "
    SUGGESTION: investigate a group of touching spots in the
    left gel image and see how well they are separated in
    the central core image.

    HINTS:
    (1) use the Xpix ZOOM 2X option in the CURSOR OPS menu,
```

```
    (2) use the Xpix PRINT REGION option in the CURSOR OPS
        menu to see the numeric values of adjacent
        propagated spots.

    NOTE: even though some adjacent spots look like they
    are touching the central core image illustrates they are
    separate spots.
" >> /tmp/demo11.tmp

echo "
    NOTE: menu operation.
  a. To get the five Xpix menus:
     hold down the keyboard CONTROL key while pressing and
     holding the MIDDLE button. Move cursor to desired entry
     and release button to select it. To change the image
     contrast and brightness, move the mouse with MIDDLE
     button pressed (don't press CONTROL key).

  b. Select the EXIT option in the VIEW OPS menu to leave
     Xpix and continue the show. This also holds for the
     following examples.

CMD:      accppx 324.1 324.1 -p2:c -zoomRight -silent

=======>  Press QUIT button when you finish this step.
" >> /tmp/demo11.tmp
cat /tmp/demo11.tmp | xless -geometry +528+1&

accppx 324.1 324.1 -p2:c -zoomRight -silent

# ---------------------------------------------------------------------

echo "6.  RMAP OF LANDMARK SPOTS FOR 324.1 and 384.1 FROM FINAL PCG DB"
echo "
[Press Sun 'Front' button when mouse is in a window to raise it.]

6.  RMAP OF LANDMARK SPOTS FOR 324.1 and 384.1
    FROM FINAL COMPOSITE GEL (PCG) DATABASE

    Spot pairing requires defining landmark spots common
    to a reference gel or 'Rgel'. An 'Rmap' is a derived
    image of a gel with specific spots labeled. These
    Rmaps show the final landmark spots previously selected
    by the investigator doing the landmarking. These
    landmarks were entered into the Landmark Database using
    the LANDMARK program. In a later example, you will have
    a chance to landmark a gel using the LANDMARK program.
" > /tmp/demo11.tmp

echo "
    HINTS:
      (1) compare the same subregions in both images
          (LMS A B C D).
```

```
      (2) use the micro-zoom window showing title area
          at bottom.
" >> /tmp/demo11.tmp

echo "
    NOTE: menu operation.
  a. To get the five Xpix menus:
     hold down the keyboard CONTROL key while pressing and
     holding the MIDDLE button. Move cursor to desired entry
     and release button to select it. To change the image
     contrast and brightness, move the mouse with MIDDLE
     button pressed (don't press CONTROL key).

  b. Select the EXIT option in the VIEW OPS menu to leave
     Xpix and continue the show. This also holds for the
     following examples.

CMD:      accppx 324.1 384.1 -prefix:l -zoomRight -silent

=======>  Press QUIT button when you finish this step.
" >> /tmp/demo11.tmp
cat /tmp/demo11.tmp | xless -geometry +528+1&

accppx 324.1 384.1 -prefix:l -zoomRight -silent

# ------------------------------------------------------------------------

echo "7.  PAIRED SPOT LABELS AFTER PROCESS GSF FILES WITH CMPGL2."
echo "
[Press Sun 'Front' button when mouse is in a window to raise it.]

7.  PAIRED SPOT LABELS AFTER PROCESS GSF FILES WITH CMPGL2.

    Using the landmarks and the spot lists, the CMPGL2
    program pairs the Rgel with another gel resulting in a
    paired-spot file. The results may also be visualized in
    an image which is displayed here.
" > /tmp/demo11.tmp

echo "
        landmarks (user specified) are  LARGE RED letters.
        S - Sure pair labeled spots are RED.
        P - Possible pair labeled spots are YELLOW.
        A - Ambiguous pair labeled spots are BLUE.
        U (+) - Unresolved labeled spots are CYAN.

    HINT: zoom up the region around landmark T or S and
          compare spots with the same labels.
" >> /tmp/demo11.tmp

echo "
    NOTE: menu operation.
  a. To get the five Xpix menus:
```

```
     hold down the keyboard CONTROL key while pressing and
     holding the MIDDLE button. Move cursor to desired entry
     and release button to select it. To change the image
     contrast and brightness, move the mouse with MIDDLE
     button pressed (don't press CONTROL key).

  b. Select the EXIT option in the VIEW OPS menu to leave
     Xpix and continue the show. This also holds for the
     following examples.

CMD:      accppx 384.1 384.1 -p1:u -p2:v -zoomRight -silent

=======>  Press QUIT button when you finish this step.
" >> /tmp/demo11.tmp
cat /tmp/demo11.tmp | xless -geometry +528+1&

accppx 384.1 384.1 -p1:u -p2:v -zoomRight -silent

# ------------------------------------------------------------------

echo "8.  LANDMARKING GELS 324.1 (Rgel) WITH 384.2"
echo "
[Press Sun 'Front' button when mouse is in a window to raise it.]

8.  LANDMARKING GELS 324.1 (Rgel) WITH 384.2

     We will be using the landmarks previously defined when
     gel 378.2 was landmarked with the Rgel, gel 324.1.

     When it starts press the LANDMARK button and then mark
     corresponding spots in the Right gel. Press HELP for
     more information. Press BINDINGS to get the accelerator
     keys and mouse bindings.
" > /tmp/demo11.tmp

echo "
     When you have had enough, select the QUIT (NOT the
     FINISH) menu option to exit. If you accidentally pressed
     FINISHED, then answer NO when it asks if you wish to
     overwrite the LMS data.
" >> /tmp/demo11.tmp

echo "
     Note that spots to be left Rgel are GREEN. After you
     select a spot to be marked in the Rgel, it turns it
     CYAN. When you mark a spot in the right gel, it turns
     the spot in the Rgel right gel RED.

     You may edit previously marked spots if you change your
     mind. Zoom and flickering may be used with landmarking.
     Press ADD LANDMARKS button to add landmarks.

CMD:      landmark 324.1 384.1 -AutoLM:378.2
```

```
=======>  Press QUIT button when you finish this step.
" >> /tmp/demo11.tmp
cat /tmp/demo11.tmp | xless -geometry +528+1&

landmark 324.1 384.1 -AutoLM:378.2

# ------------------------------------------------------------------

echo "9.  RMAPS OF LANDMARK SPOTS AS SEEN IN COMPOSITE"
echo "
[Press Sun 'Front' button when mouse is in a window to raise it.]

9.  RMAPS OF LANDMARK SPOTS AS SEEN IN COMPOSITE
    GEL DATABASE.

     After a composite gel DB is constructed, all spots are
     referred to by sequential 'Rspot' numbers. The Rspot
     numbers in the following image corresponds to the
     previous landmarks. The Rspot numbers are arbitrarily
     assigned to spots as they are constructed when merging
     gels in the Composite Gel Database.
" > /tmp/demo11.tmp

echo "
     In general, Rmaps can be generated of spots found to be
     significant using a variety of statistical or logical
     tests available during exploratory data analysis using
     the MARKGEL program.
" >> /tmp/demo11.tmp

echo "
     Such 'spot lists' are generated using the CGELP2
     Composite Gel Database program which can invoke the
     MARKGEL program with the 'RMAP' command.

     HINT: use the little zoom window to read the
     annotation and blow up spots.
" >> /tmp/demo11.tmp

echo "
    NOTE: menu operation.
  a. To get the five Xpix menus:
     hold down the keyboard CONTROL key while pressing and
     holding the MIDDLE button. Move cursor to desired entry
     and release button to select it. To change the image
     contrast and brightness, move the mouse with MIDDLE
     button pressed (don't press CONTROL key).

  b. Select the EXIT option in the VIEW OPS menu to leave
     Xpix and continue the show. This also holds for the
     following examples.
```

```
CMD:      accppx 324.1 384.1 -prefix:m -zoomRight -silent

=======>  Press QUIT button when you finish this step.
" >> /tmp/demo11.tmp
cat /tmp/demo11.tmp | xless -geometry +528+1&

accppx 324.1 384.1 -prefix:m -zoomRight -silent

# ---------------------------------------------------------------------

echo "10.  RMAP AND MOSAIC OF RSPOT 87 (LANDMARK C) IN PCG DB"
echo "
[Press Sun 'Front' button when mouse is in a window to raise it.]

10.  RMAP AND MOSAIC OF RSPOT 87 (LANDMARK C) IN
     PCG DATABASE.

     It is useful to visualize the same corresponding spot
     in several gels. This display is called a 'mosaic'
     derived image. Each mosaic panel (drawn from lightest
     to darkest, from left to right and top to bottom) is
     taken from a 'different' gel in the database.
" > /tmp/demo11.tmp

echo "
     Note each panel is labeled with the accession #, gel
     pairing label, experimental class, and density. If
     there are more than 16 gels to be displayed, then
     additional images would be generated (up to 36 mosaics
     of 576 gels).

     HINT: use the little zoom window to investigate the
     small panels or magnify the writing in the gels.

" >> /tmp/demo11.tmp

echo "
    NOTE: menu operation.
  a. To get the five Xpix menus:
     hold down the keyboard CONTROL key while pressing and
     holding the MIDDLE button. Move cursor to desired entry
     and release button to select it. To change the image
     contrast and brightness, move the mouse with MIDDLE
     button pressed (don't press CONTROL key).

  b. Select the EXIT option in the VIEW OPS menu to leave
     Xpix and continue the show. This also holds for the
     following examples.

CMD: accppx 324.1 w00087 -p1:m
            -swList:'-graphscale' -zoomRght -silent

=======>  Press QUIT button when you finish this step.
```

```
" >> /tmp/demo11.tmp
cat /tmp/demo11.tmp | xless -geometry +528+1&

accppx 324.1 w00087 -p1:m -swList:"-graphscale" -zoomRight -silent

# ------------------------------------------------------------------

echo "11.  MOSAICS OF ALL RSPOTS FOR GEL 324.1 IN SRL[2] IN PCG DB"
echo "
[Press Sun 'Front' button when mouse is in a window to raise it.]

11.  MOSAICS OF ALL RSPOTS FOR GEL 324.1 IN
     SEARCH RESULTS SUBSET [2] IN PCG DATABASE.

     This alternate form of the mosaic shows a set of
     different spots for the 'same' gel.

     HINT: use the little zoom window to investigate the
           small panels or magnify the writing in the gels.
" > /tmp/demo11.tmp

echo "
    NOTE: menu operation.
  a. To get the five Xpix menus:
     hold down the keyboard CONTROL key while pressing and
     holding the MIDDLE button. Move cursor to desired entry
     and release button to select it. To change the image
     contrast and brightness, move the mouse with MIDDLE
     button pressed (don't press CONTROL key).

  b. Select the EXIT option in the VIEW OPS menu to leave
     Xpix and continue the show. This also holds for the
     following examples.

CMD: accppx w00000 w10000
            -swList:'-graphscale' -zoomRight -silent

=======>  Press QUIT button when you finish this step.
" >> /tmp/demo11.tmp
cat /tmp/demo11.tmp | xless -geometry +528+1&

accppx w00000 w10000  -swList:"-graphscale" -zoomRight -silent

# ------------------------------------------------------------------

echo "12.  Print GELLAB-II 'state' file for this project."
echo "
[Press Sun 'Front' button when mouse is in a window to raise it.]

12.  Print GELLAB-II 'state' file for this project.

     Each 2D gel project can be kept in a separate
     directory. A special file,  gel.rc,  in that
```

```
     directory tells GELLAB about the other data files
     required for that project. For example, where the
     images, spot list, etc. data is kept for that project.
" > /tmp/demo11.tmp

echo "
     In addition, each project directory also has a 'gellab'
     subdirectory tree:
         ./gellab/ann  - annotation database if any
         ./gellab/aux  - auxiliary files GSF, GCF, Rmaps, mosaics
         ./gellab/gen  - files generated by cgelp2 sessions
         ./gellab/id   - accession file
         ./gellab/lms  - landmark database
         ./gellab/pcg  - paged composite gel database
         ./gellab/ppx  - original gel image files
         ./gellab/tmp  - temporary image files during segmentation
         ./gellab/org  - original non-PPX gel images.

     After we run the  'pgelrc' command, scroll the window
     to see the data file paths and various parameters
     available.

CMD:      pgelrc

" >> /tmp/demo11.tmp

pgelrc >> /tmp/demo11.tmp

echo "
=======>  Press QUIT button when you finish this step.
" >> /tmp/demo11.tmp

cat /tmp/demo11.tmp | xless -geometry +528+1



# ---------------------------------------------------------------------

echo "13.  CREATE BATCH SCRIPTS FOR A PROJECT WITH MAKJOB."
echo "
[Press Sun 'Front' button when mouse is in a window to raise it.]

13.  CREATE BATCH SCRIPTS FOR A PROJECTWITH MAKJOB.

     The batch jobs for a project can be generated with
     GETACC or MAKJOB. This illustrates MAKJOB. To run
     it completely from the UNIX command line, you need
     to have a CCL file containing the accession numbers
     of the gels previously accessioned that you want to
     use for the project.

     The CCL file (ts3.ccl) for a 'jnk' for this
     demonstration can be same as  for the 'ts3' project.
```

```
" > /tmp/demo11.tmp

cat -n ts3.ccl >> /tmp/demo11.tmp

echo "
     After we run the  'makjob' command, scroll the window
     to see the data file paths and various parameters
     available.

CMD:      makjob -rgel:0324.1 -class:AML:ALL:CLL:HCL:HL-60
                 -accList:ts3.ccl -prj:jnk -study:2:13

=======>  Press QUIT button when you finish this step.
" >> /tmp/demo11.tmp

makjob -rgel:0324.1 -class:AML:ALL:CLL:HCL:HL-60 -accList:ts3.ccl
-prj:jnk -study:2:13 >> /tmp/demo11.tmp

cat jnk*.do jnk*.gdo>> /tmp/demo11.tmp

cat /tmp/demo11.tmp | xless -geometry +528+1


# ------------------------------------------------------------------

echo "14.  RUN THE PAGED COMPOSITE GEL DATABASE PROGRAM - CGELP2"
echo "
[Press Sun 'Front' button when mouse is in a window to raise it.]

14.  RUN THE PAGED COMPOSITE GEL DATABASE PROGRAM - CGELP2

     We now finally run the CGELP2 composite gel database
     program with using the graphical interface option.
     The large gel image with spots indicated is the
     Dynamic Rmap which has a zoom window below it. You
     interact with cgelp2 by typing/clicking in the Dynamic
     Rmap window or selecting menu entries.
" > /tmp/demo11.tmp

echo "
    HINTS:
     (1) Put cursor on pull down menus to see what options
         are for each menu. Release button when NOT on an
         option in order to avoid invoking that function.
     (2) To find out the keyboard/mouse bindings, select
         BINDINGS in the STATUS pull down menu. Press
         PRINT button to make a laser printer copy.
     (3) To change the PREFILTER, use SET PREFILTER menu
         options.
" >> /tmp/demo11.tmp

echo "
     (4) Use HELP <particular-cmd> in STATUS menu to get
```

```
         more information about that command. Enter no
         command to get short description of all commands.
         Press PRINT button to make a laser printer copy.
     (5) The INQUIRE menu lists various search commands.
     (6) To exit, select ABORT or EXIT in the CGL DB pull
         down menu.

CMD:      cgelp2 -protect -graphics -database ts3

=======>  Press QUIT button when you finish this step.
" >> /tmp/demo11.tmp
cat /tmp/demo11.tmp | xless -geometry +528+1&

cgelp2 -protect -graphics -database ts3


# ------------------------------------------------------------------

echo "*********** THAT's ALL FOLKS ************"
echo "
************************************************************
****                 THAT's ALL FOLKS                   ***
****                                                    ***
**** GO READ THE TUTORIAL IN THE BOOK FOR MORE EXAMPLES ***
************************************************************
=======>  Press QUIT button to exit back to UNIX.
" > /tmp/demo11.tmp
cat /tmp/demo11.tmp | xless -geometry +528+1 -background yellow
rm  /tmp/demo11.tmp
# ------------------ END OF SCRIPT -------------------
```

PRINT-SLIDES.DO: Print a set of demo gel images on laser printer

```
#!/bin/csh
echo "print-slides.do script for demonstrating GELLAB-II"
# P. Lemkin
# It prints images also used with slides.do script. The images
# are generated by make-slides.do.
# The make-slides.do script may be run either interactively or in
# background batch. It assumes that the '~gelMgr/gellab/demo/tmp/ts3s02.sps'
# file has previously been created from the PCG DB.
#
# $Date: $ / $Revision: $
#

cd ~/gellab/demo
pwd
set nonoerror
echo " "
echo " LISTING OF PRINT-SLIDES.DO "
cat -n print-slides.do
```

```
set L="lpr -Plaser2"
#set T="60"
set T="1"

# ---------------------- Print the slides -------------------------
#
echo " "
echo " NOTE: MEANINGS OF PREFIX OF PPX FILES."
echo " [a,b] original gel images."
echo " [c] propagated central core image in  sg2gii."
echo " [f] flipped image (pIe or MW reversed from original) in  markgel."
echo " [g] GraphScale pseudocolor image generated by various programs."
echo " [j] averaged image in   sg2gii."
echo " [k] magnitude of 2nd derivataive image in   sg2gii."
echo " [l] Rmap landmark image in  cmpgl2."
echo " [m] Rmap image from SPSS file in  markgel."
echo " [n] notch filtered image."
echo " [s] standard 512x512 PPX size copy of original gel images."
echo " [t] change vector Rgel labeled paired-spot image in  autopair."
echo " [u] marked Rgel labeled paired-spot image in  cmpgl2 & autopair."
echo " [v] marked gel paired-spot image in  cmpgl2 & autopair."
echo " [w] mosaic image in  mosaic."
echo " [y] (original less the segmented spot) image in  sg2gii."
echo " [z] segmented image in  sg2gii."
echo " ------------------------------------------------- "

date
ppx2ps ppx/b00661.ppx -Title:"[2.1] original gel 324.1 image ppx/b00661" | $L
sleep $T
ppx2ps ppx/b00981.ppx -Title:"[2.2] original gel 384.1 image ppx/b00981" | $L
sleep $T


ppx2ps tmp/j00661.ppx -Title:"[2.3] averaged image of 324.1: tmp/j00661" | $L
sleep $T
ppx2ps tmp/j00981.ppx -Title:"[2.4] averaged image of 384.1: tmp/j00981" | $L
sleep $T


ppx2ps tmp/k00661.ppx -Title:"[2.5] magnitude Laplacian 324.1 image: tmp/k00661" | $L
sleep $T
ppx2ps tmp/k00981.ppx -Title:"[2.6] magnitude Laplacian 384.1 image: tmp/k00981" | $L
sleep $T


ppx2ps tmp/c00661.ppx -Title:"[2.7] propagated central core of 324.1: tmp/c00661" | $L
sleep $T
ppx2ps tmp/c00981.ppx -Title:"[2.8] propagated central core of 384.1: tmp/c00981" | $L
sleep $T


ppx2ps tmp/n00661.ppx -Title:"[2.9] background of gel 324.1: tmp/n00661" | $L
sleep $T
```

```
ppx2ps tmp/n00981.ppx -Title:"[2.10] background of gel 384.1: tmp/n00981" | $L
sleep $T


ppx2ps tmp/z00661.ppx -Title:"[2.11] segmented 324.1 image tmp/z00661" | $L
sleep $T
ppx2ps tmp/z00981.ppx -Title:"[2.12] segmented gel 384.1: tmp/z00981" | $L
sleep $T
ppx2ps tmp/y00661.ppx -Title:"[2.13] original - segmented gel 324.1: tmp/y00661" | $L
sleep $T


ppx2ps aux/l00661.ppx -Title:"[3.1] gel 324.1 Landmarks from LMS DB: aux/l00661" | $L
sleep $T
ppx2ps aux/l00981.ppx -Title:"[3.2] gel 384.1 Landmarks from LMS DB: aux/l00981" | $L
sleep $T

ppx2ps aux/u00981.ppx -Title:"[4.1] Pair spots (324.1,384.1) in 324.1: aux/u00981" | $L
sleep $T
ppx2ps aux/v00981.ppx -Title:"[4.2] Pair spots (324.1,384.1) in 384.1: aux/v00981" | $L
sleep $T


ppx2ps aux/m00661.ppx -Title:"[5.1] Rmap of LMs for gel 324.1: aux/m00661" | $L
sleep $T
ppx2ps aux/m00981.ppx -Title:"[5.2] Rmap of LMs for gel 324.1: aux/m00981" | $L
sleep $T


ppx2ps aux/w00087.ppx -Title:"[6.1] mosaic of Rspot 87 (LM C): aux/w00087" | $L
sleep $T


ppx2ps aux/w00000.ppx -Title:"[7.1] mosaic of 324.1 Rspot LMs: aux/w00000" | $L
sleep $T
ppx2ps aux/w10000.ppx -Title:"[7.2] mosaic of 324.1 Rspot LMs: aux/w10000" | $L

echo " All done - the slides are in the print queue of " laser2
date
% ----------------------------------------------------
```

2.2 Examples of manual program invocation

The following short list of examples illustrates some of the types of operations possible with GELLAB-II. All of the following code may be typed to try out the programs. Obviously, we can only touch on a small number of the **cgelp2** possible operations. But hopefully enough to convey the flavor of using it to help you perform an exploratory data analysis. The examples in this section are grouped according to which of the four steps listed above they belong, in a gel analysis.

No extensive examples of **cgelp2** are given here because that is beyond the scope of this Section. Instead, it illustrates individual GELLAB-II program invocation from the UNIX shell level. For more examples of using **cgelp2**, see Section 2.3 page 105 and Section 5.1, page 503 for more detailed discussion of PCG DB search strategies using **cgelp2** and examples of **cgelp2** scripts in Section 3.9 page 377.

examples notation

The text after a "`#`" denotes comments relating to the example. Do not type these comments. In these examples, commands that are typed to the *csh(1)* are prefaced with a '*number*`%`' prompt (i.e *number* is the UNIX C shell history number), those to the main command level of **cgelp2** are denoted by <`CMD`>, and those to sub command level prompts by $*$. In the interest of clarity, we will encode multiple prompt **cgelp2** commands using the '`//`' notation (see page 170).

If you have not already set up your GELLAB UNIX account as described in Section **??**, page **??**, then do so before attempting to run any of these examples. Then you should enter your ~`/gellab/demo` directory by doing

```
1% cd ~/gellab/demo        # Enter YOUR demo directory.
```

In general, skip any examples which do not apply to what you are trying to learn or are having difficultly with. Most of the examples are independent of one another because the demonstration database has precomputed most data dependencies.

2.2.1 Examples of accessioning gels

EXAMPLE 1. `Print GELLAB state.`

Print the current GELLAB state file `gel.rc`. The state file specifies the names of the various GELLAB-II directories which are required by the other programs. See section 1.6.5 page 61 for details.

the state

```
1% pgelrc               # Pretty print the existing gel.rc file.
2% mv gel.rc gel.rc.old # Save it so can restore it.
3% rm gel.rc            # DELETE IT!
4% pgelrc               # Force Q&A prompt for default gel.rc values.
5% mv gel.rc.old gel.rc # Restore old gel.rc.
```

EXAMPLE 2. `Acquire and accession gel images using` **camera**.

Alternatively, acquire and acquisition a set of gel images from the GELLAB camera. You need to have a camera on your system to use this example, otherwise skip it. Note if you will be adding new gel PPX files to your `~gellab/demo/ppx` directory, then change your `~gellab/demo/gel.rc` file to reflect this. This is the preferred method for gel image acquisition. You need to specify the name of the first gel image file `a01234.ppx` which you wish to call the scanned images. It will automatically increment the picture file number (eg. `a01235.ppx`, etc). After starting the session as follows, you need more detailed instruction. This is given in Section 3.7 page 347.

camera needed

```
1% getacc
                    # Acquire and acquisition gels from camera
                    # and display ND wedge calibration histogram
                    # for each gel. Select 'DataCopy Scan'
```

EXAMPLE 3. `Accession existing gel images.`

When using externally scanned images, acquisition a set of gel images *previously* scanned (whether with GELLAB or some other scanner). There are 12 (duplicate) demonstration gel database adult human leukemia database gels `b00661.ppx, b00889.ppx, ..., b01693.ppx` which are included for you to try out this form of data acquisition. NOTE: press `Edit` to edit an existing gel and press `Quit` when you are finished. You would need to use *this* gel accession method if you did not have a scanner as part of your GELLAB system. After starting the session, you need more detailed instruction. This is given in Section 3.7 page 347 which also covers additional options.

accession editing

```
1% getacc
                        # Similar to above, but use existing files
                        # b00661, b00889 ,..., b01693 already on the
                        # disk rather than getting them from the camera.
                        # Just fill out the form with file names
```

EXAMPLE 4. `File compression`.

save disk space

Use the *compress* and *uncompress* programs on image files to save disk space. This process preserves complete data integrity. To see how much space you save, use the `ls -l` UNIX command to list file size.

```
1% cd ~/gellab/demo/ppx   # Move to the original image PPX directory.
2% ls -l mcrew.ppx        # See how much space original image takes.
3% compress mcrew.ppx     # Compress it to save space.
4% ls -l mcrew.hex.Z      # See how much space it NOW takes.
5% uncompress mcrew.ppx.Z # Restore it to a usable form.
6% ls -l mcrew.ppx        # See if it is correctly restored - i.e. same size.
```

EXAMPLE 5. `Generate batch job scripts`.

setting up batch scripts

If you have not generated batch job scripts with **getacc** (as in Example 3 above), then you can alternatively generate (or regenerate) them with **makjob**. Generate batch scripts given a list of gel accession numbers in Concise Control List (CCL) file `tst3.ccl`, a project prefix `ts3`, five experimental classes and the Rgel. We will be discussing using these later.

```
1% cat ts3.ccl          # Print list the 12 gel accession numbers.
2% makjob -rgel:324.1 -class:AML:HCL -accs:ts3.ccl -prj:ts3 -saturation:2.0
                        # Create batch jobs with NO Q&A.
3% makjob -rgel:324.1   # Create batch jobs but answer almost
                        # all of the questions.
4% makjob               # Create batch jobs but you must answer
                        # all of the questions.
```

The following is the final batch job generation you should do in order to leave the proper set of `ts3` scripts on the disk. These files are listed in Section 3.9 page 369. The files created by **makjob** are listed on page 374. Step 5% executes a batch script which does the same thing as step 2%, but is easier to type. You could study the **cgelp2** command file `ts3cgl.gdo`. To learn how **cgelp2** behaves, are important and what commands you might commonly use. You could type them in by hand. Alternatively, you could study the resulting batch log files produced when the scripts are run as is described in Section 2.3, page 2.3.

```
1% makjob -fields       # List fields in accession file data dictionary
                        # which are used to compose the 'study'.

2% makjob -rgel:0324.1 -class:AML:ALL:CLL:HCL:HL-60 -accFile:ts3.ccl \
          -prj:ts3 -study:2:13 -saturation:2.0
                        # Generate ts3 project batch files we will
                        # use in later examples.
3% ls -l ts3*do         # List the names of the generated batch files.
4% cat makts3-demo.do   # Print full makjob specification.
5% makts3-demo.do       # Evaluate makjob script to create ts3 .do files.
```

The contents of the indirect gel accession name specification CCL file `ts3.ccl` is listed here to show the format.

```
0324.1
0369.1
0378.2
0384.1
0396.1
0497.1
0503.1
0511.1
0514.1
0515.1
0517.1
0393.2
```

2.2.2 Examples of gel segmentation and quantitation

EXAMPLE 6. `Segment gels`.

After you have accessioned images into the system, you can segment a gel into a Gel Segmentation File (GSF) quantitated spot list file and a '`z`' segmented spot image file. Several different examples are given. You might review the switch options in Section 3.18 page 452 to see what might be changed. Normally you do not segment a set of gels manually like this but rather let the generated batch jobs do it (see `ts3prc.do` in the previous Example 5). So we will not segment all of the gels in our demonstration database here, but do so in a later demonstration of the `ts3prc.do` or `ts2seg.do` batch jobs in Example 17, page 104 .

spot quantitation

```
1% sg2gii 324.1         # Segment gel with default switches.
2% accppx 324.1 324.1 -p2:z
                        # Display original and segmented images.
                        # Hint, use:  ppxz 324.1
3% sg2gii 324.1 -3x3 -ch:313:330,195:212,4:1000,0:10000,0:2.7
```

```
                        # Use smaller filter and small computing
                        # window as well as new range for D', area, OD diff.
4% accppx 324.1 324.1 -p2:z
                        # Display original and segmented images.
5% sg2gii 324.1 -3x3 -pixdmp:18 -ch:313:330,195:212,4:1000,0:10000,0:2.7
                        # Same as above but generate teletype trace
                        # of intermediate images while segmenting.
6% accppx 324.1 324.1 -p2:z
                        # Display original and segmented images.
7% sg2gii 324.1 -7x7 -ch:178:290,179:241
                        # Segment image but use different CW and filter.
8% accppx 324.1 324.1 -p2:z
                        # Display original and segmented images.
9% sg2gii 324.1 -ctlcoreimage -restofimage
                        # Segment gel but generate central core image,
                        # the original image less the segmented image,
                        # background, average and Laplacian magnitude images.
10% accppx 324.1 324.1 -p1:c -p2:y
                        # Display central core and image less spots.
11% accppx 324.1 324.1 -p2:n
                        # Display original and background OD image
                        # Hint, use:  ppxn 324.1
12% more ~/gellab/demo/aux/p10324.gsf
                        # List the GSF file.
13% sg2gii 369.1        # Segment gel with default switches.
14% sg2gii 384.1        # Segment gel with default switches.
```

2.2.3 Examples of gel pairing

EXAMPLE 7. `Pair spots between two gels.`

pairing spots

Pair two GSF spot list files into a Gel Comparison File (GCF) file. No additional images are generated. Normally, the generated "process" batch scripts (such as `ts3prc.do` or `ts3cmp.do` batch jobs) will perform these pairings and so we will not do all of the sample gel database pairings here.

```
1% cmpgl2 324.1 369.1   # Pair GSFs to make GCF file.
2% cmpgl2 324.1 384.1   # Pair GSFs to make GCF file.
3% more ~/gellab/demo/aux/c10369.gcf
                        # List the GCF file.
4% cmpgl2 324.1 369.1 -changeParameters:6,11
                        # Same as above, but change (dT1,dT2).
5% tail ~/gellab/demo/aux/c10369.gcf
                        # List the end of the GCF file.
```

EXAMPLE 8. `Pair gels with pair-label images.`

Pair two GSF spot list files into a GCF file. Also generate '`u`' and '`v`' labeled paired-spot derived images and display them. Select EXIT from the VIEW menu to terminate image display.

paired-spot images

```
1% cmpgl2 324.1 369.1 -mark
                        # Pair GSFs to make GCF file and generate
                        # the labeled paired-spot images.
2% accppx 369.1 369.1 -p1:u -p2:v
                        # Display labeled paired spot images.
                        # Hint, use:  ppxv 369.1
```

EXAMPLE 9. `Generate landmark labeled images.`

Generate the '`l`' landmark set images for the two gels, but do not pair the gels. Select EXIT from the VIEW menu to terminate image display.

landmark images

```
1% cmpgl2 324.1 369.1 -onlyMarkLMSimages
                        # Generate labeled landmark LM images.
2% accppx 324.1 369.1 -preface:l
                        # Display both labeled landmark LM images.
                        # Hint, use:  ppxl 324.1
                        #             ppxl 369.1
```

2.2.4 Examples of gel PCG DB construction

EXAMPLE 10. `Build PCG DB manually.`

Run `cgelp2` manually to build a 12 gel PCG DB. Normally you would do this with a batch job as in the later examples. Again, the <CMD> and * indicates prompts and answers respectively for **cgelp2**. More examples of running **cgelp2** are in the following Section 2.3 page 105. Step `1%` of this example takes about 10 minutes to run on a SUN3/260 computer. Once, started, it should run to completion or you will corrupt the PCG DB you are trying to build. In general, you can always type `CONTROL/C` to exit **cgelp2** and answer `yes` to the question *Do you really want to exit?*.

DB construction

```
    1% cgelp2                # Start cgelp2.
<CMD>: SET ACCESSION FILE
     * /users/joeUser/gellab/demo/id/gelts3.id
<CMD>: SET RGEL//0324.1
```

```
<CMD>: SYSTEM
     * rm /users/joeUser/gellab/demo/pcg/testpcg.pcg
<CMD>: SET DATABASE          # name of new PCG DB file
     * /users/joeUser/gellab/demo/pcg/testpcg.pcg
<CMD>: CREATE/ERspot
     * 369.1
     * 378.2
     * 369.1
     * 384.2
     * 396.1
     * 497.1
     * 503.1
     * 511.1
     * 514.1
     * 515.1
     * 517.1
     * 393.2
     *
     * PSAU                   # Declare pairing labels used when build initial DB.
     *12                     # Max dP distance allowed for eRspot creation
<CMD>: EXIT

    2% cgelp2 -d testpcg.pcg
                              # Start cgelp2 on existing PCG DB file.
<CMD>: GELS
<CMD>: BACKUP                 # Checkpoint PCG DB.
<CMD>: SYSTEM
     * ls -l /home/joeUser/gellab/demo/pcg/testpcg.pcg
<CMD>: GELS
<CMD>: EXIT

    3% rm ~gellab/demo/pcg/testpcg.pcg
                              # Delete it (unless YOU want to play
                              # with it some more).
```

EXAMPLE 11. `Build PCG DB using a batch script.`

PCG DB script

Run `cgelp2` on the command file batch script `ts3pcg.pcg` to create a PCG DB from a set of GCF files and then to run some initial statistical tests to get a feeling for the type of data in the database. Normally, this is submitted by one of the generated batch jobs as background batch (3rd part of this example). If you let this run to completion, it will generate and save the PCG DB and other auxiliary data files used in the following examples. Hint: run the example generating a batch job (i.e 2%), then print it out and study it. See Section 3.3 page 158 for more details on what the commands in the `ts3cgl.gdo` script are doing.

```
1% cgelp2 -file ts3cgl.gdo
                        # Run cgelp2 on input command script.
                        # Commands and their actions will be
```

```
                        # echoed on the terminal. Type
                        # CONTROL/C and answer 'YES' to
                        # to answer EXIT question to abort.

 2% cgelp2 -f ts3cgl.gdo > & ts3cgl.log
                        # Same as above, but save output in a
                        # log file. Otherwise it ties up your
                        # terminal for a LONG time.

 3% cgelp2 -f ts3cgl.gdo > & ts3cgl.log &
                        # Same as above, but put into UNIX
                        # background batch.
 4% jobs                # List the background batch jobs (i.e
                        # see the one you just submitted).
```

EXAMPLE 12. `Reload old PCG DB.`

Start the PCG database program on an existing database file (the one just created). Also read in and print the old history list. Generate a SPSS file from the Rspot sets listed in SRL[2].

reload PCG DB

```
    1% cgelp2 -database ts3pcg.pcg
                        # Start cgelp2 on existing PCG DB file.
<CMD>: !?               # List current history.
<CMD>: BACKUP           # Checkpoint PCG DB and history.
<CMD>: EXIT             # Checkpoint PCG DB and history, and EXIT.

    2% cgelp2 -graphics -d ts3pcg.pcg
                        # Start cgelp2 on existing PCG DB file.
                        # But start the Graphical User Interface
                        # under X-Windows. In menu GEL DB, select
                        # EXIT session to exit the program.

    3% cgelp2 -readonly -d ts3pcg.pcg
                        # Same as above, but set database to
                        # read only.
<CMD>: !?               # List current history.
<CMD>: SET SRL SUBSET//SPSS//2////
                        # Generate SRL[2] spots as ts3s02.sps SPSS data file.
<CMD>: BACKUP           # Checkpoint history ONLY!!!
<CMD>: EXIT             # Checkpoint history only and EXIT.

    4% cat gen/ts3s02.sps | more
                        # Print the SPSS data file just created.
                        # Type 'q' to exit more(1), 'SPACE' key
                        # to print the next page.
```

EXAMPLE 13. `Generate Rmap images.`

Generate an Rmap of gel 324.1 for SPSS file 'ts3s02.sps' produced from the `cgelp2`

PCG DB program. You *must* have the SPSS file prior to generating the Rmap with **markgel**. Select EXIT from the VIEW menu to terminate image display.

Rmap construction

```
1% markgel 324.1 ts3s02.sps
                    # Generate Rmap with standard options.
2% accppx 324.1 -prefix:m
                    # Display Rmap just generated by markgel
3% markgel 324.1 ts3s02.sps -Zoom:2x
                    # generate zoomed Rmap
4% accppx 324.1 -prefix:m
                    # Display Rmap just generated by markgel
5% markgel 324.1 ts3s02.sps -Zoom:2x:87
                    # Generate zoomed Rmap around Rspot[87]
6% accppx 324.1 -prefix:m
                    # Display Rmap just generated by markgel
7% markgel 324.1 ts3s02.sps -Xpix
                    # Same as above but automatically display
                    # the Rmap in an Xpix window window.
8% markgel 324.1 ts3s02.sps -HflipPIE -Xpix
                    # Display Rmap with pIe reversed.
9% markgel 324.1 ts3s02.sps -VflipMW -Xpix
                    # Display Rmap with MW reversed.
10% markgel 324.1 ts3s03.sps -Xpix
                    # Display Rmap with SP+PP+AP+US+EP.
11% markgel 324.1 ts3s03.sps -restrictLabels -Xpix
                    # Display Rmap with only SP+PP+US.
```

EXAMPLE 14. `Generate mosaic images.`

mosaic construction

Generate a mosaic image of Rspot 87 for SPSS file `ts3s02.sps` produced from the `cgelp2` PCG DB program. As with Rmaps above, you need the SPSS file to generate mosaic images. Select EXIT from the VIEW menu to terminate image display.

```
1% mosaic 87 ts3s02.sps
                    # Generate mosaic image w00087.ppx.
2% accppx w00087
                    # Look on the gel.rc paths for the image.

3% accppx 324.1 w00087
                    # Display original gel and mosaic together.
4% accppx 324.1 -p1:m w00087
                    # Display Rmap gel and mosaic together.
5% mosaic 87 ts3s02.sps -Xpix
                    # Same as above but display the Rmap in an
                    # Xpix window window.
6% mosaic 87 ts3s02.sps -HflipPIE -Xpix
                    # Display mosaic with pIe reversed.
```

```
 7% mosaic 87 ts3s02.sps -VflipMW -Xpix
                      # Display mosaic with MW reversed.
 8% mosaic 87 ts3s03.sps -restrictLabels -Xpix
                      # Display mosaic with only SP+PP+US.
 9% mosaic ts3s03.sps -gels -Xpix
                      # Display mosaic of all Rspots in SPSS file.
10% mosaic ts3s03.sps -gels:0324.1 -Xpix
                      # Display mosaic of all Rspots in SPSS file
                      # but only for the 324.1.
```

EXAMPLE 15. `Display images`.

Given an accession number display the gel in one or more `Xpix` windows. Note that when two different gels are displayed together, you can use the `compare images` option mode in **Xpix** to flicker-compare them. Select EXIT from the VIEW menu to terminate image display. *view images*

```
 1% accppx 324.1        # Given an accession number display the
                        # gel in a Xpix window.
 2% accppx 324.1 369.1  # Given two accession numbers display the
                        # gels in two Xpix windows.
 3% accppx 324.1 -prefix:z
                        # Given an accession number display the
                        # associated segmented 'z' image of gel in a
                        # Xpix window.
```

2.2.5 Examples of UNIX batch scripts

EXAMPLE 16. `Landmark gels`.

Landmark a set of gels using the interactive landmarking batch script previously generated using **getacc** or **makjob**. The script landmarks 11 gels with respect to the Rgel. The first gel to be landmarked with the Rgel will be used to define the initial landmark set which is then used in landmarking the other 10 gels. See Section 3.8 page 362 for more details on what you do next. Note that when successfully terminated, this script submits `ts3prc.do` into background batch to "process" (i.e. segment and compare) the set of gels resulting in a set of GCF files. See Section 3.8 page 362 which describes the details on landmarking before you try this example. Select COMPARE GELS from the LANDMARK menu to begin landmarking. Select EXIT from the CONTROL menu to terminate image display. *landmarking*

```
1% cp ~/gellab/demo/lms/lmsts3.lm lmsts3.lm.backup
                    # Save old landmark DB file in case you trash it.

2% cat ts3lms.do    # Print landmarking batch script

3% ts3lms.do        # Start the interactive script to do
                    # landmarking of the set of gels. You
                    # can abort this by typing CONTROL/C.

4% mv ~/gellab/demo/lms/lmsts3.lm.backup ~/gellab/demo/lms/lmsts3.lm
                    # Restore old landmark DB file in case
                    # you destroyed it while trying out
                    # landmark on the demo data.
```

EXAMPLE 17. Segment gels and pair spots between gels

process gels

Segment and pair GSF spot lists for a set of gels using "processing" batch scripts previously generated using **getacc** or **makjob**. See Section 3.18 page 452 and Section 3.4 page 322 for details on what these programs are doing. When finished, it has generated the segmented gel `z` images, and the set of GSF and GCF files. Note that when successfully terminated, this script also submits `ts3cgl.do` into background batch to build the initial PCG DB as in the following example. Normally you would not submit any of these scripts yourself as `ts3lms.do` will do it for you. They are merely a manual alternative for doing the same thing. In step `7%`, select `EXIT` from the `VIEW` menu to terminate image display and go onto the next image.

```
1% cat ts3prc.do    # Print 'gel processing' batch script.
2% ts3prc.do>&ts3prc.log
                    # Start the 'processing' script to do
                    # segmentation and pairing of the set of gels.
                    # You can abort this by typing CONTROL/C.

3% cat ts3seg.do    # Print 'only segment' gels batch script.
4% ts3seg.do>&ts3seg.log
                    # Start - only segment gels script.
                    # You can abort this by typing CONTROL/C.

5% cat ts3cmp.do    # Print list - only compare gels batch script.
6% ts3cmp.do>&ts3cmp.log
                    # Start - only compare gels script.
                    # You can abort this by typing CONTROL/C.

7% foreach x (324.1 369.1 378.2 384.1 396.1 497.1 503.1\
          511.1 514.1 515.1 517.1 393.2)
        accppx 324.1 324.1 -p2:z
   end
                    # Review the segmented gels along with originals
```

```
7'% ppxallz ts3
                    # HINT: easier alternate way to review the
                    # segmented gels along with originals.
```

EXAMPLE 18. `Construct a PCG DB.`

Construct a PCG DB given a set of GCF files previously produced by the `ts3prc.do` batch job above. See Section 3.3 page 158 for more details on what **cgelp2** is doing. The `ts3cgl.do` script is listed in step `1%`. The **cgelp2** command input script `ts3cgl.gdo` is executed by doing step `2%`. It consists of a command to puts the initial PCG DB construction process into background batch. Step `3%` *alternatively* effectively does the same thing as the batch script were you to submit it manually. Again, you would normally not submit this script yourself as `ts3prc.do` will do it for you. This example is merely a manual alternative for doing the same thing.

PCG DB construction

```
1% cat ts3cgl.do     # Print UNIX batch script to construct PCG DB
2% more ts3cgl.gdo   # Print construct PCG DB script
3% cgelp2 -f ts3cgl.gdo > & ts3cgl.log
                     # Start the contruct PCG DB
                     # script for a set of GCF files.
                     # You can abort this by typing CONTROL/C.
```

2.3 Exploring the PCG DB with cgelp2

The following is a list of short examples illustrating some of the types of operations possible with **cgelp2** PCG DB program (described in Section 3.3, page 158). All of the following code may be typed to try out the programs. See Section 5.1 for more detailed discussion of PCG DB search strategies using **cgelp2** and examples of **cgelp2** scripts in Section 3.9 page 377. The text following a "`#`" denotes comments relating to the example. Again, as with the examples in the previous Section, do not type these comments if you do the examples. In these examples, commands that are typed to the *csh(1)* are not prefaced with the '`%`' prompt, those to the main command level of **cgelp2** are denoted by <`CMD`>, and those to subcommand level prompts by $*$. For **cgelp2** commands, you only need to type enough of the command to make it unique. In the interest of clarity, we will encode multiple prompt **cgelp2** commands using the '//' notation (see page 170).

examples notation

If you have not already set up your UNIX account as described in Section **??**, page **??**, then do so before attempting to run any of these examples. Then you should enter your ~`/gellab/demo` directory by doing

```
1% cd ~/gellab/demo              # Enter YOUR demo directory.
```

EXAMPLE 1. `Build PCG DB using script.`

building PCG DB

Construct a PCG DB file `ts3pcg.pcg` using a **cgelp2** script `ts3cgl.gdo`. After the database is constructed, then access it for further exploratory data analysis in step `4%`. In steps `1%` to `3%` one could rebuild the PCG DB if it did not exist. Note that all of the `-file` or `-f` specified commands do the same thing in terms of constructing the database as you might do manually, so you will save *yourself* time by doing it in batch. We will use this database in the rest of the examples by specifying the`-database` or `-d` (database name) **cgelp2** as in `4%` below. As there is a large amount of computation in constructing the database, this takes some time (about 10 minutes on a SUN3/260).

```
1% cgelp2 -f ts3cgl.gdo     # Construct initial PCG DB using
                            # interactive batch.

2% cgelp2 -f ts3cgl.gdo >& ts3cgl.log
                            # Same as above but put output into
                            # a session log file.

3% cgelp2 -f ts3cgl.gdo >& ts3cgl.log &
                            # Same as above but background batch.

4% cgelp2 -d ts3pcg.pcg     # Start up the existing PCG DB.
```

EXAMPLE 2. `Build PCG DB manually.`

building PCG DB

Run `cgelp2` manually to build a PCG DB. Normally you would do this with a batch job as in the following examples. Again, the < `CMD`> and `*` indicates prompts and answers for **cgelp2**. Note that this PCG DB `testpcg.pcg` is not used in any of the other examples. Instead, in the following examples, we will be using the `ts3pcg.pcg` PCG DB created in the previous Example.

```
    1% cgelp2                       # Start cgelp2.
<CMD>: SYSTEM//rm /users/joeUser/gellab/pcg/testpcg.pcg
                                    # Delete it in case it exists.
<CMD>: SET DATABASE//testpcg.pcg
<CMD>: CREATE/ERspot
     * 369.1
     * 378.2
     * 369.1
     * 384.2
     * 396.1
     * 497.1
     * 503.1
     * 511.1
     * 514.1
     * 515.1
```

```
     * 517.1
     * 393.2
     *
     * PSAU                          # Pairing labels when build PCG DB
     * 12                            # Max dP for spot to join eRspot set.
<CMD>: VERIFY PCG DB                 # Check for corruption by computing
                                     # and verifying checksums. Errors
                                     # should NEVER happen!
<CMD>: GELS                          # List what gels are in the PCG DB.
<CMD>: GELS/Full/Option:0369.1
<CMD>: SYSTEM//ls -l /users/joeUser/gellab/pcg/testpcg.pcg
                                     # List file size of PCG DB.
<CMD>: INQUIRE//Print//123////       # Print Rspot sets from different
<CMD>: INQUIRE//Print//456////       # parts of the paged database.
<CMD>: EXIT                          # Checkpoint the PCG DB and exit.

    2% cgelp2 -d testpcg.pcg         # Start up the existing PCG DB.
<CMD>: GELS                          # list what is in the PCG DB.
<CMD>: EXIT

    3% rm /users/joeUser/gellab/pcg/testpcg.pcg
```

EXAMPLE 3. `Print PCG DB status.`

Print the status of the current database. This includes prefilter and other PCG DB state parameters.

prefilter status

```
    1% cgelp2 -d ts3pcg.pcg          # Start cgelp2 on existing PCG DB.
<CMD>: SET DATABASE////              # Name of PCG DB.
<CMD>: GELS                          # Gels in WS and studies.
<CMD>: LIMITS                        # Prefilter limits.
<CMD>: FEATURES                      # Maximum spot feature values.
<CMD>: SET WORKING GELS////          # Working set of gels.
<CMD>: SET CLASSES////               # Gel classifications.
<CMD>: SET GEL SUBSETS//LIST//<ALL>// # Gel subsets directory.
<CMD>: SET SRL SUBSETS//DIRECTORY////  # SRL subsets directory.
<CMD>: INQUIRE//PRINT//S////         # Names of Rspots in SRL.
<CMD>: SET FIELDS////                # Accession fields used in study.
<CMD>: SET ACCESSION FILE////        # Name of accession file.
<CMD>: VALIDLANDMARKS////            # Gel landmark data.
<CMD>: EXIT
```

EXAMPLE 4. `Dump Rspot data into SPSS file.`

Dump SPSS file of existing Search Results List (SRL). Then do it for an explicit list of Rspots. The `/FULL` switch used with `SPSS` dumps the SPSS file sorted by gel rather than by Rspot.

SPSS data

```
    1% cgelp2 -d ts3pcg.pcg          # Start cgelp2 on existing PCG DB
<CMD>: SET SRL SUBSETS//RESTORE//landmarks////
                                     # Restore SRL to landmark spots
<CMD>: SPSS//ts3lms.sps//$           # File to put it in.
<CMD>: SYSTEM//more ts3lms.sps       # Please list it.
<CMD>: SPSS//ts3srl.sps//22,53,276   # Explicit list of Rspots to dump
                                     # and redefine the SRL to these spots.
<CMD>: SPSS//ts3srlG.sps//$          # File to put it in.
<CMD>: SYSTEM//more ts3srlG.sps      # Please list it.
<CMD>: EXIT
```

EXAMPLE 5. `Find names of Rspots meeting prefilter.`

Find and list the names of all "legal" Rspot sets which meet the current prefilter criteria. This treats each Rspot set as being one experimental class (of all gels). This test is useful for finding robust spots when the prefilter constraints have been tightened. E.g. lower the CVD, make OD difference non-saturating, require spots be present in all gels, etc.

legal Rspots

```
    1% cgelp2 -d ts3pcg.pcg          # Start cgelp2 on existing PCG DB
<CMD>: SET LABEL//PSEAU
<CMD>: INQUIRE//SEARCH////
<CMD>: EXIT
```

EXAMPLE 6. `Do t-test and display Rmap.`

Do a t-test search on classes 1 and 3. Then save the resulting SRL as a SRL subset. Then write out the new SRL subset as an SPSS file also generating an associated batch script to generate mosaic images for all Rspots in the SRL subset. Display the Rmap image. Then submit the batch job to create the mosaic images (in background batch). Also create and display an Rmap of these spots. After exiting **cgelp2**, display all of the mosaic images which were computed from the simultaneously executing batch job (waiting of course until it is finished). Select EXIT from the VIEW menu to terminate image display.

t-test

```
    1% cgelp2 -d ts3pcg.pcg          # Start cgelp2 on existing PCG DB.
<CMD>: INQUIRE//t-Test//1,3          # Find SRL spots from t-test search.
<CMD>: SET SRL SUBSETS//ASSIGN//tb-test classes: 1&3 p=0.90////
<CMD>: SET SRL SUBSETS//SPSS/MOSAIC//<last>////
                                     # The SPSS file will be ts2s02.do.
<CMD>: RMAP// /PPXplot//             # Create Rmap of Rgel 324.1.
<CMD>: RMAP// /USEGEL:0369.1/PPXplot//
                                     # Create Rmap of Gel 369.1.
<CMD>: MOSAIC//87/PPXplot            # Create mosaic of Rspot 87 which
                                     # was found in search.
```

```
<CMD>: SYSTEM//ts3s02.do&           # Submit batch job to create rest of
                                    # the mosaics.
<CMD>: EXIT
                                    # Now display mosaics which have
                                    # been computed from batch job.
   2% foreach x (~/gellab/aux/w*.ppx)
           accppx 324.1 -P1:m $x    # Display Rmap of Rgel 324.1 and mosaic
                                    # image of each Rspot in turn. The
                                    # mosaics were generated from the
                                    # ts3s02.do batch job.
      end
```

EXAMPLE 7. `Print Rspot set data.`

Print Rspot set data from different parts of the PCG DB. The database manager has to "page" the Rspot sets in and out of the computer memory from the disk file if the Rspot sets are not close by (i.e. in its "cache"). Note that there is a small additional access time when the Rspot set # is not near the one previously accessed. This is not a problem - just an observation.

print Rspot

```
    1% cgelp2 -d ts3pcg.pcg          # Start cgelp2 on existing PCG DB.
<CMD>: INQUIRE//Print//23////        # Print Rspot sets from different
<CMD>: INQUIRE//Print//123////       # parts of the paged database.
<CMD>: INQUIRE//Print//124////
<CMD>: INQUIRE//Print//234////
<CMD>: INQUIRE//Print//345////
<CMD>: INQUIRE//Print//456////
<CMD>: EXIT
```

EXAMPLE 8. `Change pairing label prefilter.`

Change the spot pairing label prefilter value to 'PS' (i.e. Sure Pair (SP) + Possible Pair (PP)).

pairing label

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB.
<CMD>: SET LABEL//PS
<CMD>: EXIT
```

EXAMPLE 9. `Print current accession fields.`

Print the current accession file fields selected to define the PCG DB gel "study". If you change the fields, then it will go back out to the accession file and use it to redefine the gel study data in the PCG DB.

accession field

```
    1% cgelp2 -d ts3pcg.pcg          # Start cgelp2 on existing PCG DB
<CMD>: SET FIELDS////                # check which field we are using
<CMD>: GELS                          # Check new study
<CMD>: SET FIELDS//2,12,13////
<CMD>: GELS                          # Check new study
```

EXAMPLE 10. `Change gel working set.`

working gels

Change the working set of gels to only two gels, then change it to *all* gels which are in the PCG DB.

```
    1% cgelp2 -d ts3pcg.pcg          # Start cgelp2 on existing PCG DB
<CMD>: SET WORKING SET//DEFINE//324.1,369.1
                                     # Change it to only have 2 gels.
<CMD>: SET WORKING SET//DEFINE//ALL
                                     # Restore it to all 12 gels.
<CMD>: SET WORKING SET////
<CMD>: EXIT
```

EXAMPLE 11. `Define gel subset.`

gel subsets

Define a gel subset and then use it to redefine the working set of gels by gel subset name. Then remove two specific gels, 384.1 and 369.1, from the working set. Do a search with the reduced working set and then add the two gels back to the working set of gels.

```
    1% cgelp2 -d ts3pcg.pcg           # Start cgelp2 on existing PCG DB
<CMD>: SET GEL SUBSET//EXPLICIT//two gels//324.1,369.1////
<CMD>: SET GEL SUBSET//LIST////
<CMD>: SET WORKING SET//DEFINE//two gels//
<CMD>: SET WORKING SET////            # Just list current working set.
<CMD>: SET WORKING SET//DEFINE//ALL
<CMD>: SET WORKING SET//SUBTRACT//384.1,369.1
                                      # Remove two gels.
<CMD>: SET STATISTICS/OPTION:nbrGels=<all>
                                      # i.e 10 gels.
<CMD>: INQUIRE/QUIET//SEARCH          # Search on all gels except 384.1, 369.1
<CMD>: SET WORKING SET//ADD//384.1,369.1
                                      # Remove two gels.
<CMD>: SET STATISTICS/OPTION:nbrGels=<all>
                                      # i.e 12 gels.
<CMD>: EXIT
```

EXAMPLE 12. `On-line help`

help - apropos

There are a number of different help facilities in **cgelp2**. You can get brief summaries of the commands using the first three examples. More extensive information can be obtained by requesting help on the specific commands and subcommands.

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB
<CMD>: HELP                         # Print list of CGELP2 commands.
<CMD>: !                            # Print list of history commands.
<CMD>: INQUIRE//HELP////            # Print list of INQUIRE commands.
<CMD>: HELP HELP                    # Print manual info. on HELP command itself.
<CMD>: HELP INQUIRE                 # Print manual info. on INQUIRE command.
<CMD>: HELP INQUIRE: PRINT          # Print manual info. on PRINT subcommand.
<CMD>: # Print infomation on APROPOS
<CMD>: ??
<CMD>: # List all <CMD>s which begin with the phrase SET
<CMD>: ?SET
<CMD>: # List all <CMD>s which contain the phrase GEL
<CMD>: ?APROPOS GEL
<CMD>: EXIT
```

EXAMPLE 13. `Change density normalization mode and Print Rspot.`

Change the density normalization mode to each of the different modes. Note that some of the modes require that you have previously performed a `SET RATIO LIST` for `Ratio-density` mode or `SET LEAST SQUARE CALIBRATION` for `Least-square density` mode.

density normalization

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB
<CMD>: SET DENSITY MODE//Absolute density
<CMD>: INQUIRE//Print//123////      # Print Rspot set with new mode.
<CMD>: SET DENSITY MODE//Cpm density
<CMD>: INQUIRE//Print//123////      # Print Rspot set with new mode.
<CMD>: SET DENSITY MODE//Least-square density
<CMD>: INQUIRE//Print//123////      # Print Rspot set with new mode.
<CMD>: SET DENSITY MODE//Percent density
<CMD>: INQUIRE//Print//123////      # Print Rspot set with new mode.
<CMD>: SET DENSITY MODE//Ratio density
<CMD>: INQUIRE//Print//123////      # Print Rspot set with new mode.
<CMD>: SET DENSITY MODE//Volume density
<CMD>: INQUIRE//Print//123////      # Print Rspot set with new mode.
<CMD>: LIMITS
<CMD>: EXIT

    2% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB
<CMD>: SET DENSITY MODE//Least-square density
<CMD>: GELS                         # Note calibration values are ratio-sum
<CMD>: SET DENSITY MODE//Ratio density
<CMD>: GELS                         # Note calibration values are least-square
<CMD>: EXIT
```

EXAMPLE 14. `Change prefilter statistics limits.`

Change the Rspot set prefilter statistics limits. Note that we can put comments after the numeric answers since the program only looks at the leading numbers of a response.

changing prefilter

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB
<CMD>: SET DENSITY MODE//Absolute density
<CMD>: SET STATISTICS
     * 0,512             - relative distance of (Dx,Dy) from Landmark
     * 0,512             - mean DL
     * 0,10              - mean DP
     * 25,10000000       - mean area
     * 3,10000000        - mean density
     * 0,10.0            - CV of area
     * 0,10.0            - CV of density
     * 0,10.0            - mean OD difference
     * .90               - two class significance limit
     * 0,1000            - range of spots required/experimental class
<CMD>: LIMITS
<CMD>: EXIT
```

EXAMPLE 15. `List valid landmarks`.

valid landmarks

List the valid landmarks and their centroids for all gels and then just for a specific gel. Then list the full landmark information using the `/ListLMS` switch. You can also restrict the information to one gel. Note a T entry means that a spots exists for the landmark while a F means it did not. This does *not* affect the validity of the PCG DB, but gives and indication of gel quality.

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB
<CMD>: VALIDLANDMARKS               # Only list global gel minLSQ err.
<CMD>: VALIDLANDMARKS/FULL          # also list gel LM data for all gels.
<CMD>: VALIDLANDMARKS/FULL/OPTION:0369.1
                                    # Full information on only one gel.
<CMD>: INQUIRE//LANDMARK            # Find Rspots which are landmarks.
<CMD>: EXIT
```

EXAMPLE 16. `Declare gel class names and put gels into classes`.

gel classes

Declare gel class names and then automatically partition gels in the working set into these classes. In the second example, you can partition gels into more than one class at a time. This means that you can do searches which include different groups of gels simply by referring to their class names (i.e. numbers).

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB
<CMD>: SET WORKING GELS//DEFINE//<ALL>
<CMD>: SET CLASSES                  # Classify gels into gel subsets
     * AUTO                         # Do gel classification automatically
     * YES                          # Yes, redefine the gel class names
     * -aml                         # Class #1 name
     * -cll                         # Class #2 name
     * -all                         # Class #3 name
```

```
      * -hcl                          # Class #4 name
      * hl-60                         # Class #5 name
      * <null>
      * <null>
      * <null>
      * <null>
<CMD>: SET CLASSES//NO                # List the classes just set up.
<CMD>: INQUIRE//PRINT//123////        # Print Rspot set with new classes.
<CMD>: EXIT

    2% cgelp2 -d ts3pcg.pcg           # Start cgelp2 on existing PCG DB
<CMD>: SET WORKING GELS//DEFINE//<ALL>
<CMD>: SET CLASSES                    # Classify gels into gel subsets
      * AUTO
      * YES
      * -aml                          # Class #1 name
      * -cll                          # Class #2 name
      * -all                          # Class #3 name
      * -hcl                          # Class #4 name
      * hl-60                         # Class #5 name
      * lymphoid                      # Class #6 name
      * myeloid                       # Class #7 name
      * <null>
      * <null>
<CMD>: SET CLASSES//NO                # List the classes just set up.
<CMD>: INQUIRE//PRINT//123////        # Print Rspot set with new classes.
<CMD>: INQUIRE//tb-TEST//1,2////      # Search AML vs CLL.
<CMD>: INQUIRE//tb-TEST//6,7////      # Search LYMPHOID vs MYELOID.
<CMD>: SET CLASSES                    # Restore PCG DB to 5 classes
      * AUTO
      * YES
      * -aml                          # Class #1 name
      * -cll                          # Class #2 name
      * -all                          # Class #3 name
      * -hcl                          # Class #4 name
      * hl-60                         # Class #5 name
      * <null>
      * <null>
      * <null>
      * <null>
<CMD>: SET CLASSES//NO                # List the classes just set up.
<CMD>: EXIT
```

EXAMPLE 17. `Change study fields and reclassify gels`

study fields - reclassification

Repartition the set of 12 gels based on another feature - DURATION OF EXPOSURE - which is recorded in the accession file. If the experimental classes have a numeric class name and are ordered by increasing value, then we can fit the classes to the density data. That is, we can find Rspots which have a least squares fit of $dens = m(durationOfExposure) + b$, where: $0.25 < m < 0.75$. So first we change

the study, then reclassify, and finally we can do the least squares search. After we are done, then restore the 4 class classification.

```
    1% cgelp2 -d ts3pcg.pcg          # Start cgelp2 on existing PCG DB
<CMD>: SET WORKING GELS//DEFINE//<all>
<CMD>: SET FIELDS////                # Print current fields.
<CMD>: SET FIELDS/11                 # Set ``study' to DURATION OF EXPOSURE field.
<CMD>: SET CLASSES                   # Classify gels into gel subsets
     * AUTO                          # Do gel classification automatically
     * YES                           # Yes, redefine the gel class names
     * 24hr                          # Class #1 name
     * 96hr                          # Class #2 name
     * 166hr                         # Class #3 name
     * 167hr                         # Class #4 name
     * 168hr                         # Class #5 name
     * 220hr                         # Class #6 name
     * 264hr                         # Class #7 name
     * 360hr                         # Class #8 name
     * <null>
<CMD>: SET CLASSES//NO               # List the classes just set up.
<CMD>: INQUIRE//PRINT//123////       # Print Rspot set with new classes.
                                     # Note that spots are now in experimental
                                     # classes defined by DURATION OF LABEL.
<CMD>: INQUIRE//LEAST-SQUARE-SEARCH//0.25,0.75
                                     # Find Rspots which have a least squares
                                     # fit of Dens=M*(durationOfExposure)+B.
                                     # Where: 0.25 < M < 0.75.
<CMD>: SET SRL SUBSETS//ASSIGN//LSQ search 0.25 < M < 0.75 DurationOfExposure////
<CMD>: SET FIELDS/13                 # Restore CGELP2 'study' to acc. file stud.
<CMD>: SET CLASSES                   # Restore PCG DB to 5 classes
     * AUTO
     * YES
     * -aml                          # Class #1 name
     * -cll                          # Class #2 name
     * -all                          # Class #3 name
     * -hcl                          # Class #4 name
     * hl-60                         # Class #5 name
     * <null>
     * <null>
     * <null>
     * <null>
<CMD>: EXIT
```

EXAMPLE 18. `Perform gel subsets operations.`

gel subsets

Exercise the gel subsets operations but don't save these changes in the PCG DB. Note that we do an explicit `PROTECT` command here. Instead, if you know that you want to protect the PCG DB, start it with a `-r` (read only) switch as is shown in latter examples.

```
    1% cgelp2 -d ts3pcg.pcg          # Start cgelp2 on existing PCG DB.
<CMD>: PROTECT                       # Prevents you from saving changes.
<CMD>: SET GEL SUBSET//DELETE//1//// # just delete all gel subsets before start.
<CMD>: SET GEL SUBSET//DELETE//2////
<CMD>: SET GEL SUBSET//DELETE//3////
<CMD>: SET GEL SUBSET//DELETE//4////
<CMD>: SET GEL SUBSET//DELETE//5////
<CMD>: SET GEL SUBSET//DELETE//6////
<CMD>: SET GEL SUBSET//DELETE//7////
<CMD>: SET GEL SUBSET//DELETE//8////
<CMD>: SET GEL SUBSET//DELETE//9////
<CMD>: SET GEL SUBSET//DELETE//10////
<CMD>: SET GEL SUBSET//CLASSNAME//-aml////                    # set[1]
<CMD>: SET GEL SUBSET//CLASSNAME//-cll////                    # set[2]
<CMD>: SET GEL SUBSET//CLASSNAME//-hcl////                    # set[3]
<CMD>: SET WORKING SET//DEFINE//<all>
<CMD>: SET GEL SUBSET//WORKING SET//set of all gels////       # set[4]
<CMD>: SET GEL SUBSET//UNION//Union(AML&HCL)//-aml//-hcl////  # set[5]
<CMD>: SET GEL SUBSET//INTERSECTION//Intersect(<all>&(AML+HCL))//4//5//// # Set[6]
<CMD>: SET GEL SUBSET//SUBSTRACTION//all less(AML&HCL)//4//5//// # Set[7]
<CMD>: SET GEL SUBSET//EXPLICIT//one gel/class//324,369,378,384//// # Set[8]
<CMD>: SET GEL SUBSET//LIST//<ALL>////
<CMD>: EXIT                          # Don't save changes since protected PCG DB.
```

EXAMPLE 19. `Search with different gel working sets.`

Search with different gel working sets. The different sets of gels will have different mean density, etc. so that the prefilter will reject different Rspots depending on the sets of gels used in computing the means.

using gel subsets

```
    1% cgelp2 -r -d ts3pcg.pcg      # Start cgelp2 on existing PCG DB, readOnly.
<CMD>: SET WORKING SET//DEFINE//<all>
<CMD>: INQUIRE//SEARCH              # Search valid Rspots for ALL gels.
<CMD>: SET WORKING SET//DEFINE//Union(AML&HCL)
<CMD>: INQUIRE//SEARCH              # Search valid Rspots for just AML&HCL.
<CMD>: SET WORKING SET//DEFINE//<all>
                                    # Restore the working set to all.
<CMD>: EXIT                         # Don't save changes since protected PCG DB.
```

EXAMPLE 20. `Do least squares normalization.`

Find a set of R-spots found in all gels with mean $D' >= 3.0$. The strategy is to only normalize with robust Rspot sets consisting of SP and PP spots. These spots are used to normalize the DB and reorder it based on the new normalized density measure.

least Sq. normalization

```
    1% cgelp2 -r -d ts3pcg.pcg      # Start cgelp2 on existing PCG DB, readOnly.
<CMD>: SET LABELS//PS////           # Only normalize with Rspot sets
```

```
                                    # consisting of SP & PPs.
<CMD>: SET DENSITY MODE//ABSOLUTE DENSITY
                                    # Relax limit to include all spot
                                    # pairs for LSQ normalization.
<CMD>: SET STATISTICS               # Set to include just robust spots.
     * 0,512         # relative distance of (Dx,Dy) from Landmark
     * 0,512         # mean DL
     * 0,512         # mean DP
     * 0,10000000    # mean area
     * 3,10000000    # mean density  # Ignore very light spots.
     * 0,10.0        # CV of area
     * 0,10.0        # CV of density
     * 0,1.80        # mean OD difference (do NOT use saturating spots)
     * .90           # two class significance limit
     * 0,1000        # No restriction on number of spots/Rspot set.
<CMD>: SET REGION                   # Set (pIe,MW) region to good part of gel.
     * 156,425        # pIe
     * 139,475        # MW
<CMD>: LIMITS                       # Check it by printing it out new prefilter.
<CMD>: SET SRL SUBSETS//DELETE//1// # Make space in SRL[1].
<CMD>: SET SRL SUBSETS//EXPLICIT//norm. spots//11,70,97,181,612////
                                    # Define SRL subset spots to use in calib.
<CMD>: SET LEAST SQUARES CALIB/SRL:1
     * YES                          # Compute Least Sq. calib. for SRL[1] spots.
<CMD>: SET LEAST SQUARES CALIB//YES # Compute Least Sq. on all prefiltered spots.
<CMD>: SET REGION//0,511//0,511     # Restore full (pIe,MW) region.
<CMD>: SET DENSITY MODE/LEAST-SQ    # Set normalization mode.
<CMD>: GELS                         # Print out new values .
<CMD>: BACKUP                       # Checkpoint the new calibration.
<CMD>: SET LABEL//PSEAUX            # Open up to all labels.
<CMD>: INQUIRE//PRINT//123////      # Print Rspot set with new calibration.
<CMD>: EXIT                          # Don't save changes since protected PCG DB.
```

EXAMPLE 21. Do ratio-list normalization.

ratio-list norm.

Search for normalization R-spots where we restrict prefilter limits to include all robust spots for normalization. Alternatively, we can specify a specific list of spots with which to compute the ratio sum. You should have all gels in the working set present to do this normalization otherwise.

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB.
<CMD>: SET LABELS//PS////           # Only normalize with Rspot sets.
                                    # consisting of SP & PPs.
<CMD>: SET STATISTICS   # restrict to include all spots for normalization
     * 0,512            # relative distance of (Dx,Dy) from Landmark
     * 0,512            # mean DL
     * 0,512            # mean DP
     * 0,1000000        # mean area
     * 3,10000000       # mean density
     * 0,10.0           # CV of area
```

```
     * 0,10.0           # CV of density
     * 0,1.80           # mean OD difference (do not use saturating spots)
     * .90              # two class significance limit
     * ALL              # test only R-spot sets with all gels present.
<CMD>: SET DENSITY MODE//ABSOLUTE   # i.e. D'
<CMD>: INQUIRE/WORRY//INDEX-SEARCH  # to print spot headers
<CMD>: SET RATIO LIST               # Compute ratio norm. lists using SRL.
     * $                           # Use SRL spots if a reasonable number found.
<CMD>: SET RATIO LIST               # Or use explicit list in case INQUIRE failed!
     * 11,70,97,181,612
<CMD>: SET SRL SUBSETS/ASSIGN//normalization set of spots for RATIO mode////
<CMD>: SET SRL SUBSETS/LIST//<last>////
<CMD>: INQUIRE//PRINT//S////        # Print Spots found.
<CMD>: SET DENSITY MODE//RATIO mode # Change normalization method.
<CMD>: INQUIRE//PRINT//123////      # Print Rspot set with new calibration.
<CMD>: GELS                         # List Ratio sums for all gels.
<CMD>: EXIT
```

EXAMPLE 22. `Perform searchs with different switch modifiers.`

The `INQUIRE` searches can be performed using various switch modifiers. Some of these, `/WORRY`, `/QUIET`, `/EXPLAIN`, `/CHANGEHISTOGRAM` only affect what is printed out during or at the end of a search - not the contents of the SRL. Others, `/LOG`, `/MEDIAN`, `/EPSPOT`, `/NOEPSPOT` affect *how* the statistics are in the prefilter and in the test. The `/FILE` switch causes a session log file to be created for that search operation only.

search modifiers

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB.
<CMD>: INQUIRE/WORRY//INDEX         # Print out where you are every 30 Sec.
<CMD>: INQUIRE/QUIET//INDEX         # Print summary not each Rspot accepted.
<CMD>: INQUIRE//INDEX/LOG           # Do LOG(density) data before test.
<CMD>: INQUIRE//INDEX/MEDIAN        # Do Median of density data before test.
<CMD>: INQUIRE/EXPLAIN//INDEX       # Explain why prefilter fails and summary.
<CMD>: INQUIRE/EXPLAIN//TB-TEST//1,3
                                    # Like above, but explain why either
                                    # class 1 or class 3 fails and summarize.
<CMD>: INQUIRE//INDEX/FILE//tstidx.inq
                                    # Save search in a file as well.
<CMD>: INQUIRE//tb-Test/CHANGEHISTOGRAM//1,3
                                    # Print Change Histogram when done.
<CMD>: EXIT
```

EXAMPLE 23. `Extrapolate EP spots.`

Extrapolate EP spots from SP+PP+US spots. Missing spots are extrapolated into gels which need them. Doing this permanently adds the EP spots to the PCG DB.

EP spots

```
    1% cgelp2 -d ts3pcg.pcg          # Start cgelp2 on existing PCG DB.
<CMD>: SET LABEL//PSEUAX
<CMD>: GELS                          # Note initial EP total statistics.
<CMD>: INQUIRE//PRINT//123////       # Print spot before extrapolate.
<CMD>: EXTRAPOLATE/QUIET
<CMD>: GELS                          # Note change in EP total statistics.
<CMD>: INQUIRE//PRINT//123////       # Print spot after extrapolate.
<CMD>: EXIT
```

EXAMPLE 24. `Reorder the PCG DB.`

reorder PCG DB

Reorder the PCG DB. To illustrate how the rank order of spots in the Rspot set changes, print it out in Percent mode (if this is the current ordering). Then change the density mode view to Ratio-mode and print it again. Spots will be out of order. By doing the `REORDER` the spots will now be printed in the correct order.

```
    1% cgelp2 -d ts3pcg.pcg          # Start cgelp2 on existing PCG DB.
<CMD>: SET DENSITY MODE//PERCENT
<CMD>: INQUIRE//PRINT//123////       # print spot before reorder % dens/Mode.
<CMD>: SET DENSITY MODE//RATIO
<CMD>: INQUIRE//PRINT//123////       # print spot before reorder Ratio Dens/Mode.
<CMD>: REORDER                       # based on  Ratio Dens/Mode.
<CMD>: INQUIRE//PRINT//123////       # print spot after reorder Ratio Dens/Mode.
<CMD>: EXIT
```

EXAMPLE 25. `Generate printable CGL DB file.`

ASCII CGL file

Generate a printable ASCII Composite GeL (CGL)file of the prefiltered PCG DB. This is a *.cgl* type file. You can let the system generate a file name or specify it yourself. In addition, you can restrict the Rspot sets with the prefilter including the `/SRL:n` switch (spots which are also in SRL[n]).

```
    1% cgelp2 -d ts3pcg.pcg          # Start cgelp2 on existing PCG DB.
<CMD>: DUMP CGL////                  # Let it assign numeric file name.
<CMD>: DUMP CGL/SRL:2////            # Only use spots also in SRL subset 2.
<CMD>: DUMP CGL
     * /home/joeUser/gellab/gen/ts3pcg.cgl
                                     # We assign a specific file name.
<CMD>: EXIT
```

EXAMPLE 26. `Find Rspots which are landmarks.`

find landmarks

Find the list of landmarks in the database and save the resulting landmark SRL as a SRL subset. Then restrict a subsequent t-Test to only test the Rspot sets which are landmark spots.

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB
<CMD>: INQUIRE//LANDMARKS////
<CMD>: SET SRL SUBSETS//ASSIGN//landmark spots////
<CMD>: SET SRL SUBSETS//LIST//<last>////
                                   # This  set will be SRL[2].
<CMD>: INQUIRE//PRINT//S////       # Print names of Rspots in SRL
<CMD>: INQUIRE//PRINT//$////       # Print all Rspots sets in SRL
<CMD>: INQUIRE/SRL:2//tb-TEST//1,3   # Do t-Test restricted by SRL[2].
<CMD>: EXIT
```

EXAMPLE 27. `Change the prefilter statistical limits.`

Change the prefilter statistical limits. Change only area, density, OD diff and DP sizing for the entire PCG DB (i.e. both unextended and extended eRspot). This is also done using the *key=value* constructions - take your choice. Then the pairing label is changed and a %-test search performed. The resulting Rspot set names are placed in a SRL subset. The second part of this example changes the CVD part of the prefilter and show this effect on the number of candidate Rspot sets found in a simple search.

change prefilter

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB
<CMD>: LIMITS                      # Note before change.
<CMD>: SET STATISTICS              # set limit to allow only robust G2 US spots.
     * 0,512
     * 0,512
     * 0,10          # DP
     * 0,10000000    # area
     * 0,10000000       # density
     * 0,10.0
     * 0,10.0
     * 0,10.0
     * .90
     * 2,1000        # minimum of 2 gels/class
<CMD>: LIMITS                      # Note after change.
<CMD>: SET STATISTICS/OPTION:dp=0,11
<CMD>: SET STATISTICS/OPTION:area=25,10000000
<CMD>: SET STATISTICS/OPTION:density=3.0,10000000
<CMD>: SET STATISTICS/OPTION:nbrgels=2,200
<CMD>: SET LABEL//PSUX
<CMD>: LIMITS                      # Note after change.
<CMD>: INQUIRE//%-CHANGE TEST//1,3,400%
<CMD>: SET SRL SUBSETS//ASSIGN//%-change test of 400% classes 1,3////
<CMD>: SET SRL SUBSETS//LIST//<last>////
<CMD>: EXIT
```

EXAMPLE 28. `Search restricted pIe and MW.`

If you suspect that spots on interest lie in a particular (pIe,MW) range, you can restrict the search to that region.

restrict (pIe,MW)

```
    1% cgelp2 -d ts3pcg.pcg        # Start cgelp2 on existing PCG DB
<CMD>: SET REGION                  # Set (pIe,MW) region to good part of gel.
     * 156,425        # pIe
     * 139,475        # MW
<CMD>: INQUIRE/QUIET//SEARCH
<CMD>: SET REGION                  # Restore full (pIe,MW) region.
     * 0,511          # pIe
     * 0,511          # MW
<CMD>: INQUIRE/QUIET//SEARCH
<CMD>: EXIT
```

EXAMPLE 29. `Search restricted by pairing label.`

restrict pairing-label

Test how many Rspot sets meet the other prefilter criteria as we change the pairing label to increasingly robust spots.

```
    1% cgelp2 -d ts3pcg.pcg        # Start cgelp2 on existing PCG DB
<CMD>: SET LABEL//psauex           # All spots everywhere.
<CMD>: INQUIRE/QUIET//SEARCH
<CMD>: SET LABEL//psauex           # All spots everywhere.
<CMD>: INQUIRE/EPspot/QUIET//SEARCH
                                   # Count EP spots as having zero density.
<CMD>: SET LABEL//psau             # All measured spots.
<CMD>: INQUIRE/QUIET//SEARCH
<CMD>: SET LABEL//psux             # All non-fragmented non-ambiguous spots.
<CMD>: INQUIRE/QUIET//SEARCH
<CMD>: SET LABEL//a                # All ambiguous spots.
<CMD>: INQUIRE/QUIET//SEARCH
<CMD>: SET LABEL//ps               # All well defined spots.
<CMD>: INQUIRE/QUIET//SEARCH
<CMD>: SET LABEL//s                # All very well defined spots.
<CMD>: INQUIRE/QUIET//SEARCH
<CMD>: SET LABEL//s*               # All very well defined landmark spots.
<CMD>: INQUIRE/QUIET//SEARCH
<CMD>: EXIT
```

EXAMPLE 30. `Do searches on a range of CVD values.`

restrict COV D′

Perform searches on a seqence of increasing restrictive CVD prefilter values.

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB
<CMD>: SET STATISTICS/OPTION:cvd=0,10.0
<CMD>: INQUIRE/QUIET//SEARCH
<CMD>: SET STATISTICS/OPTION:cvd=0,5.0
<CMD>: INQUIRE/QUIET//SEARCH
<CMD>: SET STATISTICS/OPTION:cvd=0,2.5
<CMD>: INQUIRE/QUIET//SEARCH
<CMD>: SET STATISTICS/OPTION:cvd=0,1.0
<CMD>: INQUIRE/QUIET//SEARCH
```

```
<CMD>: SET STATISTICS/OPTION:cvd=0,0.5
<CMD>: INQUIRE/QUIET//SEARCH
<CMD>: SET STATISTICS/OPTION:cvd=0,0.25
<CMD>: INQUIRE/QUIET//SEARCH
<CMD>: EXIT
```

EXAMPLE 31. `Perform t-Test searches on range of p-values.`

Perform a t-Test search of the PCG DB using the existing prefilter. Do this for the probability p-value set to 0.80, 0.90, 0.95 and 0.99.

restrict p-value

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB
<CMD>: SET LABEL//PSUX
<CMD>: SET STATISTICS/OPTION:p-Value=0.80
<CMD>: INQUIRE//TB-TEST//1,3
<CMD>: SET SRL SUBSETS//ASSIGN//tb-Test p=0.80 classes 1&3////
<CMD>: SET SRL SUBSETS//LIST//<last>////
<CMD>: BACKUP                       # Checkpoint what we have done
<CMD>: SET STATISTICS/OPTION:p-Value=0.90
<CMD>: INQUIRE//TB-TEST//1,3
<CMD>: SET SRL SUBSETS//ASSIGN//tb-Test p=0.90 classes 1&3////
<CMD>: SET SRL SUBSETS//LIST//<last>////
<CMD>: BACKUP                       # Checkpoint what we have done
<CMD>: SET STATISTICS/OPTION:p-Value=0.95
<CMD>: INQUIRE//TB-TEST//1,3
<CMD>: SET SRL SUBSETS//ASSIGN//tb-Test p=0.95 classes 1&3////
<CMD>: SET SRL SUBSETS//LIST//<last>////
<CMD>: BACKUP                       # Checkpoint what we have done
<CMD>: SET STATISTICS/OPTION:p-Value=0.99
<CMD>: INQUIRE//TB-TEST//1,3
<CMD>: SET SRL SUBSETS//ASSIGN//tb-Test p=0.99 classes 1&3////
<CMD>: SET SRL SUBSETS//LIST//<last>////
<CMD>: EXIT
```

EXAMPLE 32. `Do different parametric test searches.`

Perform different parametric test searches on the PCG DB using the existing prefilter. Note that the `/WORRY` and `/QUIET` switches only change the amount of output on the terminal while the search is going on - not the results of the search. Also save the results of each of the searches.

parametric tests

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB
<CMD>: SET LABEL//PSUX
<CMD>: INQUIRE//TB-TEST (standard tb-test)//1,3
<CMD>: SET SRL SUBSETS//ASSIGN//tb-Test p=0.90 classes 1&3////
<CMD>: INQUIRE/WORRY//TB-TEST (standard tb-test)//1,3
<CMD>: INQUIRE/QUIET//TB-TEST (standard t-test)//1,3
<CMD>: INQUIRE//TC-TEST (confidence limits T-test)//1,3
```

```
<CMD>: SET SRL SUBSETS//ASSIGN//TC-Test p=0.90 classes 1&3////
<CMD>: INQUIRE//BEHRENS-FISHER-T-TEST//1,3
<CMD>: SET SRL SUBSETS//ASSIGN//BF-Test p=0.90 classes 1&3////
<CMD>: INQUIRE//TB-TEST (use F-statistic to pick T or F test)//1,3
<CMD>: SET SRL SUBSETS//ASSIGN//TB-Test p=0.90 classes 1&3////
<CMD>: INQUIRE//F-TEST//1,3
<CMD>: SET SRL SUBSETS//ASSIGN//F-Test p=0.90 classes 1&3////
<CMD>: INQUIRE//F-TEST//1,2,3,4
<CMD>: SET SRL SUBSETS//ASSIGN//F-Test p=0.90 classes 1,2,3,4////
<CMD>: SET SRL SUBSETS//LIST//<last>////
<CMD>: EXIT
```

EXAMPLE 33. `Perform non-parametric WMW-Test.`

non-parametric tests

Perform a non-parametric rank order WMW-Test search of the PCG DB. Note that there are two different names for the same test.

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB
<CMD>: INQUIRE//RankOrder-test//1,3
<CMD>: INQUIRE//WMW-test//1,3
<CMD>: SET SRL SUBSETS//ASSIGN//WMW-Test p=0.90 classes 1&3////
<CMD>: EXIT
```

EXAMPLE 34. `Perform missing-spot-Test search.`

missing-spot

Perform a missing-spot-Test search of the PCG DB using the existing prefilter.

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB
<CMD>: INQUIRE//Missing-spot test//1,3
<CMD>: SET SRL SUBSETS//ASSIGN//missing-class-Test classes 1&3////
<CMD>: INQUIRE//Missing-spot test/MustBeinIn1stClass//1,3
<CMD>: SET SRL SUBSETS//ASSIGN//missing-class-Test classes in 1 & not in 3////
<CMD>: EXIT
```

EXAMPLE 35. `Perform Upper-Lower missing-spot-Test search.`

Perform a missing-spot-Test search of the PCG DB using the existing prefilter. The maximum number of gels for a class to be considerfed present is 5. The minimum

missing-spot

number of gels for a class to be considerfed present is 12.

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB
<CMD>: INQUIRE//Upper-Lower-Missing-Spot test//1,3//3,12
<CMD>: SET SRL SUBSETS//ASSIGN//Upper-lowerL missing-class-Test classes 1&3////
<CMD>: EXIT
```

EXAMPLE 36. `Perform Expression Profile search.`

Perform an Expression-Profile-Test search of the PCG DB using the existing prefilter.

expression-profile

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB
<CMD>: INQUIRE
     * Expression-profile-test
     * 1,2,3,4                     # Names of classes to test.
     * 0.3, 1.0 : 0.5 : 3.0 : 0.5   # lsqERR threshold and EP.
<CMD>: SET SRL SUBSETS//ASSIGN//EProfile classes:1-4, 0.3, 1.0:0.5:3.0:0.5////
<CMD>: INQUIRE/SRL:2//EProfile//1,2,3,4//0.3, 1.0 : 0.5 : 3.0 : 0.5
<CMD>: EXIT
```

EXAMPLE 37. `Perform Coordinate-Pair search.`

Perform a Coordinate-Pair search of the PCG DB using the existing prefilter. Compare the set landmark Rspots for classes 1 and 2. Let the threshold be 100% and the change in pIe and MW be 20 each.

coordinate-pair

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB
<CMD>: INQUIRE//COORDINATE-PAIR//1,2,100%,20,20
<CMD>: EXIT
```

EXAMPLE 38. `Search extended PCG DB.`

Search just the extended eRspot part of the PCG DB when doing a t-test.

extended PCG DB

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB
<CMD>: SET LABEL//XXUE
<CMD>: INQUIRE//tb-Test//1,3
<CMD>: SET SRL SUBSETS//ASSIGN//tb-Test p=0.90 classes 1&3 lbl=XXUE////
<CMD>: EXIT
```

EXAMPLE 39. `Search using EP spots in statistics.`

Do a t-Test search on PCG DB for any spots with SP, PP, US or EP spots which use `/EPspot` extension. This counts EP spots as having zero density in statistics calculations (when switch is set). Note that you can look at EP spots which are found in the possibly more robust un-extended Rspots.

using EP spots

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB
<CMD>: SET LABEL//XXUE              # just (extended) eRspots
<CMD>: INQUIRE/EPspot//tb-Test//1,3
<CMD>: SET SRL SUBSETS//ASSIGN//tb-Test(/EPspot) p=0.90 classes 1&3 lbl=XXUE////
<CMD>: SET LABEL//PSEU              # just unextended Rspots (with no EP in Rgel)
```

```
<CMD>: INQUIRE/EPspot//tb-Test//1,3
<CMD>: SET SRL SUBSETS//ASSIGN//tb-Test(/EPspot) p=0.90 classes 1&3 lbl=PSEU////
<CMD>: SET LABEL//PSEUX             # allow EP spots from anywhere
<CMD>: INQUIRE/EPspot//tb-Test//1,3
<CMD>: SET SRL SUBSETS//ASSIGN//tb-Test(/EPspot) p=0.90 classes 1&3 lbl=PSEUX////
<CMD>: EXIT
```

EXAMPLE 40. `Compute global histograms.`

histograms

Compute global histograms of selected database features. Put the histogram files in the `ppnp5x` (i.e. the `gen`) directory.

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB
<CMD>: SET DISPLAY//TTY
<CMD>: HISTOGRAM/QUIET//Number of gels per Rset//tstNBR.tbl
<CMD>: HISTOGRAM//Individual spot density//tstISD.tbl
<CMD>: HISTOGRAM/QUIET//Density of Rspot set//tstMRD.tbl
<CMD>: HISTOGRAM/QUIET//Maximum spot density//tstMXD.tbl
<CMD>: HISTOGRAM/QUIET//Area of spots//tstARA.tbl
<CMD>: HISTOGRAM/QUIET//P is DP values of spots//tstDP.tbl
<CMD>: HISTOGRAM/QUIET//L is DL values of spots//tstDL.tbl
<CMD>: HISTOGRAM/QUIET//Variation (CVD) of Rspot set//tstV.tbl
<CMD>: HISTOGRAM/QUIET//OD difference of spot densities//tstO.tbl
<CMD>: HISTOGRAM/QUIET//Individual spot density/MW//ISD.tbl
<CMD>: HISTOGRAM/QUIET//Individual spot density/pIe//tstISD.tbl
<CMD>: EXIT
```

EXAMPLE 41. `Define subsets of gels and do operations on them.`

gel subsets

Define sub sets of gels and perform operations on them. They may be used to redefine the working set of gels. Note the use of `REMOVEALL` to delete the 10 gel subsets instead of using 10 `DELETE` subcommands.

```
    1% cgelp2 -r -d ts3pcg.pcg       # Start cgelp2 on existing PCG DB, readOnly.
<CMD>: SET GEL SUBSET//REMOVEALL//// # Delete ALL gel subsets before start
<CMD>: SET GEL SUBSET//CLASS//-all////
<CMD>: SET GEL SUBSET//CLASS//-cll////
<CMD>: SET GEL SUBSET//CLASS//-hcl////
<CMD>: SET GEL SUBSET//LIST//2////                # List SRL[2] which is -cll
<CMD>: SET GEL SUBSET//UNION//all+cll//1//2////   # Union(ALL+CLL) gels
<CMD>: SET GEL SUBSET//UNION//lymphoid//4//3////  # Union(ALL+CLL+HCL) gels
<CMD>: SET GEL SUBSET//DIRECTORY//<last>////
<CMD>: SET WORKING GELS//DEFINE//-aml
<CMD>: SET WORKING GELS//DEFINE//lymphoid
<CMD>: SET WORKING GELS//DEFINE//<all>
<CMD>: EXIT                          #  Don't save changes since protected PCG DB
```

EXAMPLE 42. `Calculate gel-gel correlation table.`

Compute and print correlation tables for gel-gel density correlation table for the working set of gels. Prior to doing the correlation, set up the working set of gels from the gel subset *lymphoid* previously defined.

gel correlation

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB
<CMD>: SET WORKING GELS//DEFINE//lymphoid # restrict WS gels to lymphoid
<CMD>: TABLE//CORRELATION//tstMNV.tbl     # Gel-gel correlation and
                                          # save table in file.
<CMD>: SET WORKING GELS//DEFINE//<ALL>    # restore WS to all gels
<CMD>: EXIT
```

EXAMPLE 43. `Calculate Rspot-Rspot correlation table.`

Compute and print correlation tables for Rspot-Rspot density correlation table for the "landmarks" SRL subset of Rspots to the working SRL prior to doing the correlation.

Rspot correlation

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB
<CMD>: SET SRL SUBSETS//RESTORE SRL//landmarks////
<CMD>: TABLE//SRL-TABLE//tstSRL.tbl//$
<CMD>: EXIT
```

EXAMPLE 44. `Calculate Rank-Order table.`

Compute and print the Rank order table of landmark SRL subset Rspot sets.

rank order

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB.
<CMD>: SET SRL SUBSETS//RESTORE SRL//landmarks////
<CMD>: TABLE//RANKORDER-TABLE//tstROT.tbl//$
<CMD>: EXIT
```

EXAMPLE 45. `Generate Order Expression Profile.`

Compute the Order Expression Profile landmark spots. This also attempts to find Rspots with similar expression profiles such that they have a lsqErr $<$ a threshold (0.5 here). The table is saved in a file.

order expression profile

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB
<CMD>: INQUIRE//OEProfile/File//tstEPT.inq//0.5 min LSQ profile match err////
<CMD>: EXIT
```

EXAMPLE 46. `Compute Change Histogram on SRL data.`

change histogram

Compute and print the Change Histogram of %-test 1&3 SRL subset.

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB
<CMD>: SET SRL SUBSETS//RESTORE//%-change test of 400% classes 1,3////
<CMD>: INQUIRE//CHANGE-HISTOGRAM//1,3////
<CMD>: EXIT
```

EXAMPLE 47. `Compute and print Order Table for SRL data.`

order table

Compute and print the Order Table for Landmark SRL[2] subset. Then print the Order Table for the 4-class F-test search.

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB.
<CMD>: SET SRL SUBSETS//RESTORE//2//          # By number.
<CMD>: SET SRL SUBSETS//RESTORE//landmarks// # By name.
<CMD>: INQUIRE//ORDER-TABLE/FILE//tstORT.inq//// # Save table in file.
<CMD>: SET SRL SUBSETS//RESTORE//F-Test p=0.90 classes 1,2,3,4// # By name.
<CMD>: INQUIRE//ORDER-TABLE                   # Just print it.
<CMD>: EXIT
```

EXAMPLE 48. `Exercise some SRL subset operations`

SRL subsets

Exercise some the SRL subset operations. They are divided into categories of similar commands.

```
    1% cgelp2 -r -d ts3pcg.pcg      # Start cgelp2 on existing PCG DB, readOnly.
          # 1. Set operations between sets.
<CMD>: SET SRL SUBSETS//UNION//1//2////
<CMD>: SET SRL SUBSETS//INTERSECTION//1//2////
<CMD>: SET SRL SUBSETS//SUBTRACT//1//2////
          # 2. Relational search of set contents and set titles.
<CMD>: SET SRL SUBSETS//DIRECTORY////
<CMD>: SET SRL SUBSETS//LIST//10////
<CMD>: SET SRL SUBSETS//QUERYRSPOT//273////
<CMD>: SET SRL SUBSETS//QUERYRSPOT//110////
<CMD>: SET SRL SUBSETS//FINDKEYWORD//aml////
<CMD>: SET SRL SUBSETS//FINDKEYWORD//cll////
          # 3. Dump and restore SRL subsets to .srl files.
<CMD>: SET SRL SUBSETS//WRITE(to ts3s01.srl)//<all>////
<CMD>: SET SRL SUBSETS//DIRECTORY////
<CMD>: SET SRL SUBSETS//CLEAR////
<CMD>: SET SRL SUBSETS//DIRECTORY////
<CMD>: SET SRL SUBSETS//READ//<all>//ts3s01.srl////
<CMD>: SET SRL SUBSETS//DIRECTORY////
          # 4. Dump and restore SRL subsets to .srl files.
<CMD>: SET SRL SUBSETS//RESTORE//2////
<CMD>: INQUIRE//PRINT//S////        # list contents of SRL
```

```
          # 5. Define SRL subsets.
<CMD>: SET SRL SUBSETS//ASSIGN//Rspots in current SRL////
<CMD>: SET SRL SUBSETS//EXPLICIT//explicitly specified Rspots//201,273,523////
          # 6. Create SPSS fields and optional mosaic batch jobs.
<CMD>: SET SRL SUBSETS//SPSS/2////  # Generate ts3s02.sps
<CMD>: SET SRL SUBSETS//SPSS/MOSAIC/2////
                                    # Same as above, but generate ts3s02.do job.
<CMD>: SET SRL SUBSETS//SPSS/MOSAIC//2-7////
                                    # Same as above, but do it for sets 2 to 7.
<CMD>: EXIT                         # Don't save the changes.
```

EXAMPLE 49. `Compare t-Test and WMW-search.`

Do two statistical searches saving the results in SRL subsets. Then compare the sets of Rspots found in the two searches. This will then tell you the spots found in one search but not in another.

compare multivariate searches

```
    1% cgelp2 -r -d ts3pcg.pcg      # Start cgelp2 on existing PCG DB, readOnly.
<CMD>: INQUIRE//tb-Test//1,3
<CMD>: SET SRL SUBSETS//ASSIGN//tb-test////
<CMD>: INQUIRE//WMW-Test//1,3
<CMD>: SET SRL SUBSETS//ASSIGN//WMW-test////
<CMD>: SET SRL SUBSETS//INTERSECTION//WMW-test//tb-test////
<CMD>: SET SRL SUBSETS//ASSIGN//Rspots in both tb-Test and WMW-test////
<CMD>: SET SRL SUBSETS//LIST//<last>>////
<CMD>: SET SRL SUBSETS//SUBTRACTION//WMW-test//tb-test////
<CMD>: SET SRL SUBSETS//ASSIGN//Rspots in WMW-Test but not in t-test////
<CMD>: SET SRL SUBSETS//LIST//<last>>////
<CMD>: SET SRL SUBSETS//SUBTRACTION//tb-test//WMW-test////
<CMD>: SET SRL SUBSETS//ASSIGN//Rspots in t-Test but not in WMW-test////
<CMD>: SET SRL SUBSETS//LIST//<last>>////
<CMD>: EXIT                         # Don't save changes since protected PCG DB.
```

EXAMPLE 50. `Compare t-Tests of three subclasses.`

Do three statistical searches comparing AML vs ALL (1,2), AML vs CLL (1,3) and AML vs HCL (1,4). Find the set of spots common to all three tests and those which are unique to the ALL class.

compare SRL subsets

```
    1% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB.
<CMD>: INQUIRE//tn-Test//1,2
<CMD>: SET SRL SUBSETS//ASSIGN//all////
<CMD>: INQUIRE//tn-Test//1,3
<CMD>: SET SRL SUBSETS//ASSIGN//cll////
<CMD>: INQUIRE//tn-Test//1,4
<CMD>: SET SRL SUBSETS//ASSIGN//hcl////
<CMD>: SET SRL SUBSETS//INTERSECTION//all//cll////
<CMD>: SET SRL SUBSETS//ASSIGN//I(all,cll)////
```

```
<CMD>: SET SRL SUBSETS//LIST//<last>>////
<CMD>: SET SRL SUBSETS//INTERSECTION//I(all,cll)//hcl////
<CMD>: SET SRL SUBSETS//ASSIGN//I(all,cll,hcl)////
<CMD>: SET SRL SUBSETS//LIST//<last>>////

    2% cgelp2 -d ts3pcg.pcg         # Start cgelp2 on existing PCG DB
<CMD>: SET SRL SUBSETS//SUBTRACTION//all//cll////
<CMD>: SET SRL SUBSETS//ASSIGN//all-cll////
<CMD>: SET SRL SUBSETS//LIST//<last>>////
<CMD>: SET SRL SUBSETS//SUBTRACTION//all//hcl////
<CMD>: SET SRL SUBSETS//ASSIGN//all-hcl////
<CMD>: SET SRL SUBSETS//LIST//<last>>////
<CMD>: SET SRL SUBSETS//UNION//all-cll//all-hcl////
<CMD>: SET SRL SUBSETS//ASSIGN//(all-hcl)+(all-hcl)////
<CMD>: SET SRL SUBSETS//LIST//<last>>////
<CMD>: SET SRL SUBSETS//SUBTRACT//all//(all-hcl)+(all-hcl)////
<CMD>: SET SRL SUBSETS//ASSIGN//Unique to ALL=all-((all-hcl)+(all-hcl))////
<CMD>: SET SRL SUBSETS//LIST//<last>>////
<CMD>: EXIT
```

EXAMPLE 51. `Run T-test over range of p-values.`

changing p-value, Rmaps

Run the t-test at different p-values of 0.80 to 0.99, making Rmaps for each test to visually see how many spots are false positives. After each t-Test, it uses using `RMAP// /PPXplot` to display the Rmap. You might decide, before going on to the next p-value test, to make a few mosaics using `MOSIAC//`*Rspot#*`/PPXplot`. Select `EXIT` from the `VIEW` menu to terminate image display.

```
    1% cgelp2 -r -d ts3pcg.pcg      # Start cgelp2 on existing PCG DB, readOnly.
<CMD>: SET STATISTICS/OPTION:p-Value=0.80
<CMD>: INQUIRE/QUIET//tn-Test//1,4
<CMD>: RMAP// /PPXplot//            # Create and display Rmap of Rgel 324.1.
<CMD>: SET STATISTICS/OPTION:p-Value=0.90
<CMD>: INQUIRE/QUIET//tn-Test//1,4
<CMD>: RMAP// /PPXplot//            # Create and display Rmap of Rgel 324.1.
<CMD>: SET STATISTICS/OPTION:p-Value=0.95
<CMD>: INQUIRE/QUIET//tn-Test//1,4
<CMD>: RMAP// /PPXplot//            # Create and display Rmap of Rgel 324.1.
<CMD>: SET STATISTICS/OPTION:p-Value=0.99
<CMD>: INQUIRE/QUIET//tn-Test//1,4
<CMD>: RMAP// /PPXplot//            # Create and display Rmap of Rgel 324.1.
<CMD>: EXIT                         # Don't save changes since protected PCG DB.
```

EXAMPLE 52. `Run a search for spots with minimum values of CVD.`

changing CVD, Rmaps

Run a search for spots with minimum values of different coefficients of deviation (CVD) for density in the range of 2.0 to 0.25, making Rmaps for each test. Analyze

the set of gels as a whole. Then compute a histogram of CVD. Then change the prefilter to different upper bounds for CVD to find how many have CVD < these values and to visualize Rspots which are. By checking the failCode histogram at the end of the searching, you can see how many spots were rejected because of CVD. Select EXIT from the VIEW menu to terminate image display.

```
    1% cgelp2 -r -d ts3pcg.pcg      # Start cgelp2 on existing PCG DB, readOnly.
<CMD>: RMAP// /PPXplot//            # Create and display Rmap of Rgel 324.1.
<CMD>: SET STATISTICS/OPTION:cvd==0,2.0
<CMD>: INQUIRE/QUIET/EXPLAIN//SEARCH
<CMD>: RMAP// /PPXplot//            # Create and display Rmap of Rgel 324.1.
<CMD>: SET STATISTICS/OPTION:cvd==0,1.0
<CMD>: INQUIRE/QUIET/EXPLAIN//SEARCH
<CMD>: RMAP// /PPXplot//            # Create and display Rmap of Rgel 324.1.
<CMD>: SET STATISTICS/OPTION:cvd==0,0.5
<CMD>: INQUIRE/QUIET/EXPLAIN//SEARCH
<CMD>: RMAP// /PPXplot//            # Create and display Rmap of Rgel 324.1.
<CMD>: SET STATISTICS/OPTION:cvd==0,0.25
<CMD>: INQUIRE/QUIET/EXPLAIN//SEARCH
<CMD>: RMAP// /PPXplot//            # Create and display Rmap of Rgel 324.1.
<CMD>: SET DISPLAY//TTY             # Plot on terminal as printable file
<CMD>: HISTOGRAM//Variation(CVD)//tstCVD.tbl
<CMD>: SET DISPLAY//PLOT            # Generate UGF file which can be
                                    # plotted with PLOTN program.
<CMD>: HISTOGRAM//Variation(CVD)
<CMD>: SET DISPLAY//LASER           # Plot on your laser printer if set up.
<CMD>: HISTOGRAM//Variation(CVD)
<CMD>: EXIT                         # Don't save changes since protected PCG DB.
```

EXAMPLE 53. Create mosaic image from PCG DB.

Create a mosaic image from using the specified spot which is currently in the search results list. For example, spot 87 is in the landmark set, SRL[2]. Select EXIT from the VIEW menu to terminate image display. *mosaic*

```
    1% cgelp2 -r -d ts3pcg.pcg      # Start cgelp2 on existing PCG DB, readOnly.
<CMD>: SET SRL SUBSETS//RESTORE//2//// # Restore SRL from subset with Rspot 87.
<CMD>: MOSAIC//87/PPXplot//         # Create and display mosaic of Rspot 87.
<CMD>: EXIT                         # Don't save changes since protected PCG DB.
```

2.4 Sample of exploratory data analysis session

This session document was created by typing the UNIX command: `script` and then editing the resulting session log by adding subsequent "annotation" before each command typed by the investigator. It had assumed that the PCG DB `jj1pcg.pcg` had

previously been created with the script generated automatically by either **makjob** or **getacc**. All annotation starts with a #.

session annotation

Part [A.1] to [A.4] starts up the **cgelp2** program on the database and review its current status. In parts [B.1] to [B.2] we now start to ask questions about the relationships between some of the existing SRL subsets. In part [C.1] to [C.5] we compare classes 1 and 2 since the previous searches had only compared classes 1 and 2. In part [D.1] to [D.16] we start off thinking we are going to make a Rmap but change our mind. Notice that it is easy to switch contexts and get on-line help. In part [E.1] to [E.3], we review what we have done recently by looking at the command history and see that we have compared classes 2 and 3 but not 1 and 3 or 1 and 2 so lets try that. In part [F.1] to [F.2] we look at some Rspot-Rspot correlations for these same Rspots in the current SRL. Finally, in part [G.1] to [G.10] we return to display the Rmap we had initially start out to draw. We also generate a UNIX batch script which when evaluated under the UNIX shell will generate derived images for these SRL spots. This is then started in the background. And we return to reload the PCG DB and continue the analysis.

```
# [A.1] Start the PCG DB database program cgelp2 on the previously
# created PCG DB for the jj1 project (see jj1cgl.log).
36% cgelp2 -d jj1pcg

CGELP2 - September 20, 1989. Todays date:09/21/1989, 06:34:48AM
 Type 'HELP' for list of CMDS. Type ?? for hints on APROPOS usage.
 Written 1981-1989, P. Lemkin.

Using existing PCG paged composite gel database:
        /home/joeUser/gellab/pcg/jj1pcg.pcg
   Date created: 09/20/1989, 06:31:01
   Date last session: 09/20/1989, 08:28:36
[06:34:52AM] Real TIME =00:00:02  CPU TIME =00:00:01,  50.00%

# [A.2] Print latest command history from previous session.
117<CMD>: !?
 96:SET SRL//ASSIGN//Intersection of LANDMARKS and NORMALIZATION spots////////
 97:SET SRL SUBSET//LIST/DIR////
 98:SET STATISTICS - set wide open//0,1024//0,1024//0,1024//0,5000//0,6000 \
                     //0,10.0//0,10.0//0,10.0//1//0 - test all Rspot sets
 99:SET DISPLAY//PLOT
 100:HISTOGRAM/QUIET//Number of gels per Rset
 101:HISTOGRAM/QUIET//Individual spot density
 102:HISTOGRAM/QUIET//Set mean Rspot set density
 103:HISTOGRAM/QUIET//Maximum spot density
 104:HISTOGRAM/QUIET//Area of spots
 105:HISTOGRAM/QUIET//P - DP values of spots
 106:HISTOGRAM/QUIET//L - DL values of spots
 107:HISTOGRAM/QUIET//Variation of std dev/mean Rset density
 108:HISTOGRAM/QUIET//OD difference of spot densities
 109:TABLE//Correlation//jj1mnv.tbl
 110:GELS
```

```
 111:FEATURES
 112:SET GEL SUBSETS//LIST//<ALL>////
 113:SET SRL SUBSETS//LIST/DIR//<ALL>////
 114:LIMITS
 115:EXIT
 116:SET DATABASE

# [A.3] Re-execute command 112.
117<CMD>: !112
112: SET GEL SUBSETS//LIST//<ALL>////
Gel Subset commands
 (Classname DElete Explicit Intersection List REMOVE Subtract Union Workingset)
 ?: LIST
Gel subset names?:
Gel subset name (or # or <ALL>)?: <ALL>
Sub set # 1  =
Sub set # 2  =
Sub set # 3  =
Sub set # 4  =
Sub set # 5  =
Sub set # 6  =
Sub set # 7  =
Sub set # 8  =
Sub set # 9  =
Gel Subset commands
 (Classname DElete Explicit Intersection List REMOVE Subtract Union Workingset)
 ?:
[06:35:24AM] Real TIME =00:00:00  CPU TIME =00:00:00,  0.00%

# [A.4] List SRL subsets previously created.
118<CMD>: set srl subsets
SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], FIndkeywd,
  Intersection[/ListOfSRLs], List[/Dir/ListofSRLs], QueryRspot, REAd,
  RENUMBER, REStore, SPss[/Mosaic], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?: list/dir
SRL subset names:
  [ 1] NORMALIZATION SPOTS NON-SATURATING AND FOUND IN ALL GELS |13|
  [ 2] LANDMARKS SET OF SPOTS |24|
  [ 3] F-TEST OF ALL CLASSES AT P-VALUE=0.90 |49|
  [ 4] F-TEST OF ALL CLASSES AT P-VALUE=0.95 |18|
  [ 5] F-TEST OF ALL CLASSES AT P-VALUE=0.99 |12|
  [ 6] TB-TEST OF CLASSES (1,2) AT P-VALUE=0.90 |13|
  [ 7] TB-TEST OF CLASSES (1,2) AT P-VALUE=0.95 |12|
  [ 8] TB-TEST OF CLASSES (1,2) AT P-VALUE=0.99 |11|
  [ 9] WMW-TEST OF CLASSES (1,2) AT P-VALUE=0.90 | 7|
  [10] WMW-TEST OF CLASSES (1,2) AT P-VALUE=0.95 |76|
  [11] WMW-TEST OF CLASSES (1,2) AT P-VALUE=0.99 | 5|
  [12] MISSING-CLASS-TEST OF CLASSES 1 AND 2 MIN #GELS/CLASS=3 |41|
  [13] ALL PSEUX RSPOTS FOR RGEL 0010.1 |984|
  [14] INTERSECTION OF LANDMARKS AND NORMALIZATION SPOTS | 5|
Subset name Or #?:
```

```
[06:35:42AM] Real TIME =00:00:09  CPU TIME =00:00:00,  0.00%

# [B.1] Ask if Landmark SRL intersects any spots found in .90 p-value T-test.
119<CMD>: set srl subsets
SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], FIndkeywd,
  Intersection[/ListOfSRLs], List[/Dir/ListofSRLs], QueryRspot, REAd,
  RENUMBER, REStore, SPss[/Mosaic], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?: intersection
1st subset # or name?: 6
2nd subset # or name (or SRL)?: 2
Result in working SRL.
Search Results List =
SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], FIndkeywd,
  Intersection[/ListOfSRLs], List[/Dir/ListofSRLs], QueryRspot, REAd,
  RENUMBER, REStore, SPss[/Mosaic], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?:
[06:35:58AM] Real TIME =00:00:14  CPU TIME =00:00:00,  0.00%

# [B.2] Print current prefilter parameter limits.
120<CMD>: limits
Rspot Mapping file [].
Map Rspots to Fspots is OFF.
Ratio-List density mode calibration EXISTS.
Least-Squares density mode calibration EXISTS.
Image size [rowsXcolumns]=[512X512]
Relative distance limits [0.00,1024.00] pixels
Mn DL limits [0.00,1024.00] pixels
Mn DP limits [0.00,1024.00] pixels
MN area limits [0.00,5000.00] pixels**2
MN density (Mode: L, tot-density/spot) limits [0.00,6000.00]
Coef. variation: S.D./Mn Rset area limits [0.00,10.00]
Coef. variation: S.D./Mn Rset dens limits [0.00,10.00]
Spot OD difference limits [0.00,10.00] OD
Class difference t-Test, F-test, Rank order p-value limit is .99:
Check if # gels in Rspot set [0:1000]
(x,y) calibration file: <NONE>
pIe Rgel limits [0.00,511.00] pixels
MW Rgel limits [0.00,511.00] pixels
Pairing-Search-Label [PSU]
No SRL Subset search restriction is active.
[06:36:00AM] Real TIME =00:00:00  CPU TIME =00:00:00,  0.00%

# [C.1] Change prefilter limits.
121<CMD>: set statistics
Answer '@' to backup to previous Q&A question,
       '$' to exit Q&A immediately,
       '?' print help for setting a parameter by keyword,
       'RETURN-key' to leave limits unchanged,
       Otherwise enter numeric values to change parameter limits.
```

```
Relative distance limits [0.00,1024.00]?:
Mean DL [0.00,1024.00]?:
Mean DP [0.00,1024.00]?:
Mean area  [0.00,5000.00]?:
Mean density (Mode:L) [0.00,6000.00]?:
Coef. variation (S.D./Mn Rset) area [0.00,10.00]?:
Coef. variation (S.D./Mn Rset) density [0.00,10.00]?:0,1.5
Spot (max-min) OD difference [0.00,10.00]?: @   - oops wrong entry so backup.
Coef. variation (S.D./Mn Rset) density [0.00,1.50]?: 0,10
Spot (max-min) OD difference [0.00,10.00]?: 0,1.5  - so not saturating
Class difference t-(F-, or WMW-)Test, p-value or confidence level
 (1%, 5%, 10%, 20% or .99, .95, .90, .80) is 1%?: 10%
Check # gels in Rspot set (per class) [0:1000](ALL or #s)?: 3,1000 - min 3/class

[06:36:40AM] Real TIME =00:00:32  CPU TIME =00:00:00,  0.00%

# [C.2] Review current gel - class partitioning.
122<CMD>: set classes
Class # 1(WITHOUT)=0001.1, 0002.1, 0003.1, 0004.1, 0005.1, 0006.1,
Class # 2(20UM)=0010.1, 0007.1, 0008.1, 0009.1,
Class # 3(50UM)=0011.1, 0012.1, 0013.1, 0014.1, 0015.1, 0016.1,
Class # 4(<NULL>)=
Class # 5(<NULL>)=
Class # 6(<NULL>)=
Class # 7(<NULL>)=
Class # 8(<NULL>)=
Class # 9(<NULL>)=
Change classes (Yes/Auto/Subsets/No)[N]?:
[06:36:45AM] Real TIME =00:00:02  CPU TIME =00:00:00,  0.00%

# [C.3] Do a TB test of classes 2 and 3 not checked previously.
123<CMD>: inquire
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%).
 HELP to list CMDS?: tb
Std t-/BehrensFisher t-Test (using F-stat) class search at .90 significance.
Which two classes are to be compared?: ?    # i.e. print class names.
These are the current CLASS NAMES
---------------------------------
  Class# 1 = WITHOUT
  Class# 2 = 20UM
  Class# 3 = 50UM
Which two classes are to be compared?: 2,3

Rspot[ 407] mnXYDA=(355,257,180.64,45)+-(3.46,2.45,243.08,17) CVD=1.35 #G=8
 [407] (m2,m3)=(177.33,6.11), m3/m2=0.03 dF=3.02 t-BF=3.22] n2=4 n3=3 f=453.48 s1=106.33 s2=4.99

Rspot[ 442] mnXYDA=(329,287,92.72,73)+-(2.45,3.87,146.27,47) CVD=1.58 #G=10
 [442] (m2,m3)=(7.08,149.88), m3/m2=21.16 dF=5.01 t-BF=2.06] n2=3 n3=6 f=1461.43 s1=4.43 s2=169.

Rspot[ 712] mnXYDA=(84,357,47.70,59)+-(2.24,1.00,51.69,32) CVD=1.08 #G=9
 [712] (m2,m3)=(12.15,78.26), m3/m2=6.44 varPooled=1796.94 t-T=2.14 f=13.83

Rspot[ 832] mnXYDA=(161,473,203.17,136)+-(3.32,3.32,130.25,93) CVD=0.64 #G=9
```

```
 [832] (m2,m3)=(112.58,303.47), m3/m2=2.70 varPooled=11740.95 t-T=2.31 f=2.19

Rspot[ 846] mnXYDA=(230,496,16.84,55)+-(2.83,3.16,19.16,54) CVD=1.14 #G=8
 [846] (m2,m3)=(35.30,6.93), m3/m2=0.20 varPooled=181.04 t-T=2.76 f=14.76

Found 5 Rspots, mean density+-'sd/mn' = 2.64+-1.76
[06:38:03AM] Real TIME =00:01:16  CPU TIME =00:00:55,  72.37%

# [C.4] Save the spots we just found.
124<CMD>: set srl subsets
SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], FIndkeywd,
  Intersection[/ListOfSRLs], List[/Dir/ListofSRLs], QueryRspot, REAd,
  RENUMBER, REStore, SPss[/Mosaic], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?: assign
New subset # or name?: manual TB-test p=.90 class:2,3, maxOD=1.5, min 3gels/class
Saved SRL in Set[15] 'MANUAL TB-TEST P=.90 CLASS:2,3, MAXOD=1.5, MIN 3GELS/CLASS'.

SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], FIndkeywd,
  Intersection[/ListOfSRLs], List[/Dir/ListofSRLs], QueryRspot, REAd,
  RENUMBER, REStore, SPss[/Mosaic], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?: list/dir
SRL subset names:
  [ 1] NORMALIZATION SPOTS NON-SATURATING AND FOUND IN ALL GELS |13|
  [ 2] LANDMARKS SET OF SPOTS |24|
  [ 3] F-TEST OF ALL CLASSES AT P-VALUE=0.90 |49|
  [ 4] F-TEST OF ALL CLASSES AT P-VALUE=0.95 |18|
  [ 5] F-TEST OF ALL CLASSES AT P-VALUE=0.99 |12|
  [ 6] TB-TEST OF CLASSES (1,2) AT P-VALUE=0.90 |13|
  [ 7] TB-TEST OF CLASSES (1,2) AT P-VALUE=0.95 |12|
  [ 8] TB-TEST OF CLASSES (1,2) AT P-VALUE=0.99 |11|
  [ 9] WMW-TEST OF CLASSES (1,2) AT P-VALUE=0.90 | 7|
  [10] WMW-TEST OF CLASSES (1,2) AT P-VALUE=0.95 |76|
  [11] WMW-TEST OF CLASSES (1,2) AT P-VALUE=0.99 | 5|
  [12] MISSING-CLASS-TEST OF CLASSES 1 AND 2 MIN #GELS/CLASS=3 |41|
  [13] ALL PSEUX RSPOTS FOR RGEL 0010.1 |984|
  [14] INTERSECTION OF LANDMARKS AND NORMALIZATION SPOTS | 5|
  [15] MANUAL TB-TEST P=.90 CLASS:2,3, MAXOD=1.5, MIN 3GELS/CLASS | 5|
Subset name Or #?: 15
Set #15<<<MANUAL TB-TEST P=.90 CLASS:2,3, MAXOD=1.5, MIN 3GELS/CLASS>>>= 407

 442 712 832 846

SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], FIndkeywd,
  Intersection[/ListOfSRLs], List[/Dir/ListofSRLs], QueryRspot, REAd,
  RENUMBER, REStore, SPss[/Mosaic], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?: inter
1st subset # or name?: 2,
```

```
2nd subset # or name (or SRL)?: 15
Result in working SRL.
Search Results List =

SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], FIndkeywd,
  Intersection[/ListOfSRLs], List[/Dir/ListofSRLs], QueryRspot, REAd,
  RENUMBER, REStore, SPss[/Mosaic], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?:
[06:39:11AM] Real TIME =00:01:03  CPU TIME =00:00:00,  0.00%
# NOTE: spots just found  with the TB test for classes 2 and 3 are
# are NOT landmarks since the intersection was empty.

# [C.5] Checkpoint the PCG DB so we save the results of the TB search we
# just performed.
127<CMD>: backup
Checkpointing PCG database
        /home/joeUser/gellab/pcg/jj1pcg.pcg
for later use.

Using existing PCG paged composite gel database:
        /home/joeUser/gellab/pcg/jj1pcg.pcg
   Date created: 09/20/1989, 06:31:01
   Date last session: 09/21/1989, 06:39:31
[06:39:34AM] Real TIME =00:00:03  CPU TIME =00:00:01,  33.33%

# [D.1] Change the pairing label to include EXTRAPOLATED spots since
# we will want to include them when we generate the mosaic images.
128<CMD>: set label

Enter spot pairing-reliability label `search-pattern`
 consisting of (A, S, P, U, E, C, X (and *))
    A is Ambiguous Pair, S is sure Pair, P is Possible Pair,
    U is Unresolved Spot, C is composite Pair,
    E is Extrapolated Pair,
    X allows accessing the eRspot database,
    XX allows accessing ONLY the eRspot database.
 [PSU] ?: psue
New pairing search label [PSUE]
[06:39:54AM] Real TIME =00:00:04  CPU TIME =00:00:00,  0.00%

# [[D.2] Change only the ODDF limit so now no restriction on MAX OD - i.e.
# now include saturated spots.
129<CMD>: set statistics
Answer '@' to backup to previous Q&A question,
       '$' to exit Q&A immediately,
       '?' print help for setting a parameter by keyword,
       'RETURN-key' to leave limits unchanged,
       Otherwise enter numeric values to change parameter limits.
Relative distance limits [0.00,1024.00]?: ?

Legal <keys> are:  DISTANCE DL DP AREA DENSITY CVA CVD ODDF P-VALUE NBRGELS
```

```
with 'SET STATISTICS/Option:<key>=<value-range>'
E.g.                    SET STATISTICS/Option:DENSITY=3,100

[06:40:00AM] Real TIME =00:00:03  CPU TIME =00:00:00,  0.00%

# [D.3] Ok - now use the short form for setting a single parameter.
130<CMD>: set statistics/option:oddf=0,4.0
GLOBAL CMD SWITCHES: /Option:ODDF=0,4.0
Answer '@' to backup to previous Q&A question,
       '$' to exit Q&A immediately,
       '?' print help for setting a parameter by keyword,
       'RETURN-key' to leave limits unchanged,
       Otherwise enter numeric values to change parameter limits.
Spot (max-min) OD difference [0.00,1.50]?:
[06:40:25AM] Real TIME =00:00:00  CPU TIME =00:00:00,  0.00%

# [D.4] Check that change was actually made.
131<CMD>: limits
Rspot Mapping file [].
Map Rspots to Fspots is OFF.
Ratio-List density mode calibration EXISTS.
Least-Squares density mode calibration EXISTS.
Image size [rowsXcolumns]=[512X512]
Relative distance limits [0.00,1024.00] pixels
Mn DL limits [0.00,1024.00] pixels
Mn DP limits [0.00,1024.00] pixels
MN area limits [0.00,5000.00] pixels**2
MN density (Mode: L, tot-density/spot) limits [0.00,6000.00]
Coef. variation: S.D./Mn Rset area limits [0.00,10.00]
Coef. variation: S.D./Mn Rset dens limits [0.00,10.00]
Spot OD difference limits [0.00,4.00] OD      - YES it was.
Class difference t-Test, F-test, Rank order p-value limit is .90:
Check if # gels in Rspot set [3:1000]
(x,y) calibration file: <NONE>
pIe Rgel limits [0.00,511.00] pixels
MW Rgel limits [0.00,511.00] pixels
Pairing-Search-Label [PSUE]
No SRL Subset search restriction is active.
[06:40:30AM] Real TIME =00:00:00  CPU TIME =00:00:00,  0.00%

# [D.5] Forgot to also change the lower bound on #gels/class - so go do that.
133<CMD>: set stat/opt:nbrgels=ALL
GLOBAL CMD SWITCHES: /Option:NBRGELS=ALL
Answer '@' to backup to previous Q&A question,
       '$' to exit Q&A immediately,
       '?' print help for setting a parameter by keyword,
       'RETURN-key' to leave limits unchanged,
       Otherwise enter numeric values to change parameter limits. imgfile.h
Check # gels in Rspot set (per class) [3:1000](ALL or #s)?:
[06:40:57AM] Real TIME =00:00:01  CPU TIME =00:00:00,  0.00%

# [D.6] Check it - notice it is [16:1000] not [0:16].
134<CMD>: limits
```

```
Rspot Mapping file [].
Map Rspots to Fspots is OFF.
Ratio-List density mode calibration EXISTS.
Least-Squares density mode calibration EXISTS.
Image size [rowsXcolumns]=[512X512]
Relative distance limits [0.00,1024.00] pixels
Mn DL limits [0.00,1024.00] pixels
Mn DP limits [0.00,1024.00] pixels
MN area limits [0.00,5000.00] pixels**2
MN density (Mode: L, tot-density/spot) limits [0.00,6000.00]
Coef. variation: S.D./Mn Rset area limits [0.00,10.00]
Coef. variation: S.D./Mn Rset dens limits [0.00,10.00]
Spot OD difference limits [0.00,4.00] OD
Class difference t-Test, F-test, Rank order p-value limit is .90:
Check if # gels in Rspot set [16:1000]
(x,y) calibration file: <NONE>
pIe Rgel limits [0.00,511.00] pixels
MW Rgel limits [0.00,511.00] pixels
Pairing-Search-Label [PSUE]
No SRL Subset search restriction is active.
[06:41:00AM] Real TIME =00:00:00  CPU TIME =00:00:00,  0.00%

# [D.7] So fix it to [0:16] using the <CMD> history mechanism. Print the
# history and then look for the command to set it. Later re-execute
# that command but change the ALL feature to 0,16.
136<CMD>: !?
 118:set srl//lsit/dir////
 119:set srl//intersecti//6//2////
 120:limits
 121:set stat/////////////////////////0,1.5//@//0,10//0,1.5//10%//3,1000
 122:set classes////
 123:inq//tb//?//2,3
 124:set srl subsets//assign//manual TB-test p=.90 class:2,3, maxOD=1.5, min 3gels/class////
 125:SET SRL SUBSET//list/dir//15////
 126:SET SRL SUBSET//inter//2,//15////
 127:backup
 128:set label//psue
 129:set stat//?
 130:set stat/option:oddf=0,4.0
 131:lim
 132:set stat//?
 133:set stat/opt:nbrgels=ALL
 134:lim
 135:set stat/////////////////////////////////////

# [D.8] Re-execute with proper argument by substitute 0,16 for ALL.
137<CMD>: !133^ALL^0,16
133: set stat/opt:nbrgels=0,16
GLOBAL CMD SWITCHES: /Option:NBRGELS=0,16
Answer '@' to backup to previous Q&A question,
       '$' to exit Q&A immediately,
       '?' print help for setting a parameter by keyword,
       'RETURN-key' to leave limits unchanged,
```

```
      Otherwise enter numeric values to change parameter limits.
Check # gels in Rspot set (per class) [16:1000](ALL or #s)?:
[06:42:09AM] Real TIME =00:00:00  CPU TIME =00:00:00,  0.00%

# [D.9] Verify the change.
138<CMD>: limits
Rspot Mapping file [].
Map Rspots to Fspots is OFF.
Ratio-List density mode calibration EXISTS.
Least-Squares density mode calibration EXISTS.
Image size [rowsXcolumns]=[512X512]
Relative distance limits [0.00,1024.00] pixels
Mn DL limits [0.00,1024.00] pixels
Mn DP limits [0.00,1024.00] pixels
MN area limits [0.00,5000.00] pixels**2
MN density (Mode: L, tot-density/spot) limits [0.00,6000.00]
Coef. variation: S.D./Mn Rset area limits [0.00,10.00]
Coef. variation: S.D./Mn Rset dens limits [0.00,10.00]
Spot OD difference limits [0.00,4.00] OD
Class difference t-Test, F-test, Rank order p-value limit is .90:
Check if # gels in Rspot set [0:16]
(x,y) calibration file: <NONE>
pIe Rgel limits [0.00,511.00] pixels
MW Rgel limits [0.00,511.00] pixels
Pairing-Search-Label [PSUE]
No SRL Subset search restriction is active.
[06:42:16AM] Real TIME =00:00:01  CPU TIME =00:00:00,  0.00%

# [D.10] Checkpoint the change.
139<CMD>: backup

Checkpointing PCG database
        /home/joeUser/gellab/pcg/jj1pcg.pcg
for later use.

Using existing PCG paged composite gel database:
        /home/joeUser/gellab/pcg/jj1pcg.pcg
   Date created: 09/20/1989, 06:31:01
   Date last session: 09/21/1989, 06:42:23
[06:42:26AM] Real TIME =00:00:03  CPU TIME =00:00:01,  33.33%

# [D.11] Changed my mind - only look at valid (i.e. non-extrapolated) spots.
140<CMD>: set label

Enter spot pairing-reliability label `search-pattern`
 consisting of (A, S, P, U, E, C, X (and *))
    A is Ambiguous Pair, S is sure Pair, P is Possible Pair,
    U is Unresolved Spot, C is composite Pair,
    E is Extrapolated Pair,
    X allows accessing the eRspot database,
    XX allows accessing ONLY the eRspot database.
 [PSUE] ?: ps
New pairing search label [PS]
[06:42:36AM] Real TIME =00:00:02  CPU TIME =00:00:00,  0.00%
```

```
# [D.12] Lets generate ordered expression profile  tables based on the these
# Rspots just found.
141<CMD>: inquire
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%).
 HELP to list CMDS?: ?           # list the commands.

INQUIRE COMMANDS
----------------
                            --- SEARCHES ---
Behrens-Fisher t-test search for Rspot sets with given significance
        between classes (assuming different variance).
Expression-profile-test - for spots with min LSQ err < T.
F-test search for Rspot sets with given confidence limit between N classes.
Index search for entire Rspots meeting statistical prefilter limits,
*Kruskol-Wallis rank order search for Rspot sets with given significance
         between all defined classes for minimum of 5 gels/class.
LAndmark search for Rspot sets which are landmarks,
LEast squares linear search fitting Dens=m*ClassValue+b for
        corr.coef. < confidence limit.
Missing class search for Rspots where (only 1 of 2 classes present),
% search for Rspots where mean ratios of 2 classes: MAX(c1/c2,c2/c1) > T%,
Rank order (or WMW) search for Rspot sets with given significance between
        2 classes.
Search for entire Rspot sets meeting statistical prefilter limits,
T-test search for Rspot sets with given confidence limit between
        classes (assuming same variance).
TB-search using F-stat. to select T- or B-F-T-test per Rspot set.
TC-search using confidence-limits t-test search.

                            --- TABLES ---
CHange histogram of mean ratios of last SRL for two classes.
COordinate pairs of last SRL for two classes and %Err,Tpie,Tmw.
OExpression-profile print Expression Profile table and linkage of SRL spots.
ORder Rspots table by all class pairs (>,<,-) for spots in SRL.
Print Rspot set i, '$' (search results Rspots), 'S' to list SRL #s,

                    --- OPTIONAL SECONDARY SWITCHES ---
     (1) Append /FILE to any command to log its output in a file
         with a '.inq' file to be specified.
     (2) Append /LOGDENSITY to any command for density to
         be recomputed as LOG(1+density) prior to its use in any
         operation.
     (3) Append /STUDYVALUE or default /CLASSVALUE to determine the
         independent variable for 'LEAST-SQRS-LINEAR-SCH' command.
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%).
 HELP to list CMDS?: oep
Order Expression Profile of Rspots in SRL.
Enter minimum LSQ profile threshold
 ?: .5
[06:43:04AM] Real TIME =00:00:24  CPU TIME =00:00:00,  0.00%

# [D.13] !!! Nothing there since we set the SRL to NULL by doing the
# intersection previously which resulted in no common spots. This
```

```
# is easily fixed by RESTOREing SRL[15].
142<CMD>: set srl subsets
SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], FIndkeywd,
  Intersection[/ListOfSRLs], List[/Dir/ListofSRLs], QueryRspot, REAd,
  RENUMBER, REStore, SPss[/Mosaic], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?: restore
Subset name Or #?: manual          # NOTE: access set by partial name.
Restoring SRL from subset [15] with 5 spots.
<<<MANUAL TB-TEST P=.90 CLASS:2,3, MAXOD=1.5, MIN 3GELS/CLASS>>>
Search Results List =  407 442 712 832 846

SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], FIndkeywd,
  Intersection[/ListOfSRLs], List[/Dir/ListofSRLs], QueryRspot, REAd,
  RENUMBER, REStore, SPss[/Mosaic], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?: list
Subset name Or #?: man
Set #15<<<MANUAL TB-TEST P=.90 CLASS:2,3, MAXOD=1.5, MIN 3GELS/CLASS>>>= 407

 442 712 832 846

SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], FIndkeywd,
  Intersection[/ListOfSRLs], List[/Dir/ListofSRLs], QueryRspot, REAd,
  RENUMBER, REStore, SPss[/Mosaic], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?:
[06:43:23AM] Real TIME =00:00:14  CPU TIME =00:00:00,  0.00%

# [D.14] Checkpoint again - it can not hurt. This is especially true since
# we had to fiddle around with getting the PREFILTER correct.
144<CMD>: backup

Checkpointing PCG database
        /home/joeUser/gellab/pcg/jj1pcg.pcg
for later use.

Using existing PCG paged composite gel database:
        /home/joeUser/gellab/pcg/jj1pcg.pcg
   Date created: 09/20/1989, 06:31:01
   Date last session: 09/21/1989, 06:43:26
[06:43:29AM] Real TIME =00:00:03  CPU TIME =00:00:01,  33.33%

# [D.15] Redo the previous INQUIRE <CMD> with the appropriate sub-arguments.
# Note: that since we had done a '?' - unfortunately this is repeated as well.
# Generally you do NOT want to repeat help messages.
145<CMD>: !inq
141: inq//?//oep//.5
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%).
 HELP to list CMDS?: ?
```

```
INQUIRE COMMANDS
----------------
                          --- SEARCHES ---
Behrens-Fisher t-test search for Rspot sets with given significance
        between classes (assuming different variance).
Expression-profile-test - for spots with min LSQ err < T.
F-test search for Rspot sets with given confidence limit between N classes.
Index search for entire Rspots meeting statistical prefilter limits,
*Kruskol-Wallis rank order search for Rspot sets with given significance
         between all defined classes for minimum of 5 gels/class.
LAndmark search for Rspot sets which are landmarks,
LEast squares linear search fitting Dens=m*ClassValue+b for
        corr.coef. < confidence limit.
Missing class search for Rspots where (only 1 of 2 classes present),
% search for Rspots where mean ratios of 2 classes: MAX(c1/c2,c2/c1) > T%,
Rank order (or WMW) search for Rspot sets with given significance between
        2 classes.
Search for entire Rspot sets meeting statistical prefilter limits,
T-test search for Rspot sets with given confidence limit between
        classes (assuming same variance).
TB-search using F-stat. to select T- or B-F-T-test per Rspot set.
TC-search using confidence-limits t-test search.

                          --- TABLES ---
CHange histogram of mean ratios of last SRL for two classes.
COordinate pairs of last SRL for two classes and %Err,Tpie,Tmw.
OExpression-profile print Expression Profile table and linkage of SRL spots.
ORder Rspots table by all class pairs (>,<,-) for spots in SRL.
Print Rspot set i, '$' (search results Rspots), 'S' to list SRL #s,

                  --- OPTIONAL SECONDARY SWITCHES ---
     (1) Append /FILE to any command to log its output in a file
         with a '.inq' file to be specified.
     (2) Append /LOGDENSITY to any command for density to
         be recomputed as LOG(1+density) prior to its use in any
         operation.
     (3) Append /STUDYVALUE or default /CLASSVALUE to determine the
         independent variable for 'LEAST-SQRS-LINEAR-SCH' command.
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%).
 HELP to list CMDS?: oep
Order Expression Profile of Rspots in SRL.
Enter minimum LSQ profile threshold
 ?: .5


Mean-Dens-Class-I/Mean-Dens-Class-J
Rspot:  m1/1 m2/1 m3/1
-------------------------------------------------------------
  407    1.0  0.2  0.0
  442    1.0  1.1 22.4
  712    1.0  7.8 50.4
  832    0.0  0.0  0.0
```

```
  846    1.0 33.1  6.5
 Similar EPs Sorted by Minimum lsqErr of profiles (Rspot#:lsqErrValue)
-----------------------------------------------------------------------
  407
  442
  712
  832
  846
[06:43:35AM] Real TIME =00:00:02  CPU TIME =00:00:00,  0.00%

# [D.16] Print some CHANGE HISTOGRAMS of these Rspots in the SRL
147<CMD>: inquire
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%).
 HELP to list CMDS?: change histogram
Which two classes are to be compared?: 2,3
The 5 Rspots are:  407 442 712 832 846
-----------------------------------------------------------------
m3/m2 Rspot sets
 0.00 407
 0.05
 0.10
 0.15 846
 0.20
 0.25
        .
        .
        .
 2.60
 2.65 832
 2.70
 2.75
        .
        .
        .
 6.35
 6.40 712
 6.45
 6.50
        .
        .
        .
 9.95
10.00 442
[06:44:26AM] Real TIME =00:00:08  CPU TIME =00:00:00,  0.00%

# [E.1] Look at the HISTORY to see what we have done so far.
149<CMD>: !?
 129:set stat//?
 130:set stat/option:oddf=0,4.0
 131:lim
 132:set stat//?
 133:set stat/opt:nbrgels=ALL
 134:lim
```

```
 135:set stat/////////////////////////////////////
 136:set stat/opt:nbrgels=ALL
 137:set stat/opt:nbrgels=0,16
 138:lim
 139:backup
 140:set label//ps
 141:inq//?//oep//.5
 142:set srl//res//manual////
 143:SET SRL SUBSET//list//man////
 144:backup
 145:inq//?//oep//.5
 146:inq//oep//1.0////
 147:inq//change histogram//2,3
 148:ch

# [E.2] Redo CHANGE HISTOGRAM but this time for 1,3 instead of previous
# classes 2,3.
149<CMD>: !inq^2,3^1,3
147: inq//change histogram//1,3
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%).
 HELP to list CMDS?: change histogram
Which two classes are to be compared?: 1,3
The 5 Rspots are:  407 442 712 832 846
----------------------------------------------------------------
m3/m1 Rspot sets
 0.00 407 832
 0.05
 0.10
 0.15
        .
        .
        .
 6.40
 6.45 846
 6.50
 6.55
        .
        .
        .
 9.95
10.00 442 712
[06:44:49AM] Real TIME =00:00:01  CPU TIME =00:00:00,  0.00%

# [E.3] Redo previous command (using the !! notation) but for this time
# for classes 1,2 instead of previous classes 2,3.
150<CMD>: !!^1,3^1,2
149: inq//change histogram//1,2
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%).
 HELP to list CMDS?: change histogram
Which two classes are to be compared?: 1,2
The 5 Rspots are:  407 442 712 832 846
----------------------------------------------------------------
m2/m1 Rspot sets
```

```
 0.00 832
 0.05
 0.10
 0.15
 0.20 407
 0.25
 0.30
        .
        .
        .
 1.00
 1.05 442
 1.10
 1.15
        .
        .
        .
 7.75
 7.80 712
 7.85
 7.90
        .
        .
        .
 9.95
10.00 846
[06:45:01AM] Real TIME =00:00:02  CPU TIME =00:00:00,  0.00%

# [F.1] Take a look at  SRL-SRL-correlation table.
151<CMD>: table
Table type?(Rank-order, Correlate:gel-gel, Srl-Srl-correlate)?: srl
Output file:[/home/joeUser/gellab/gen/000001.tbl]?: jj1s15.tbl
File: /home/joeUser/gellab/gen/jj1s15.tbl 09/21/1989, 06:45:33AM

# 1[0010.1] study: / NCTC P300/ -/ -/ -/ 120HR 20UM EXPER-COND
# 2[0001.1] study: / NCTC P301/ -/ -/ -/ 84HR WITHOUT EXPER-COND
# 3[0002.1] study: / NCTC P301/ -/ -/ -/ 84HR WITHOUT EXPER-COND
# 4[0003.1] study: / NCTC P303/ -/ -/ -/ 132HR WITHOUT EXPER-COND
# 5[0004.1] study: / NCTC P303/ -/ -/ -/ 132HR WITHOUT EXPER-COND
# 6[0005.1] study: / NCTC P303/ -/ -/ -/ 132HR WITHOUT EXPER-COND
# 7[0006.1] study: / NCTC P303/ -/ -/ -/ 132HR WITHOUT EXPER-COND
# 8[0007.1] study: / NCTC P300/ -/ -/ -/ 120HR 20UM EXPER-COND
# 9[0008.1] study: / NCTC P300/ -/ -/ -/ 84HR 20UM EXPER-COND
#10[0009.1] study: / NCTC P300/ -/ -/ -/ 84HR 20UM EXPER-COND
#11[0011.1] study: / NCTC P308/ -/ -/ -/ 132HR 50UM EXPER-COND
#12[0012.1] study: / NCTC P308/ -/ -/ -/ 132HR 50UM EXPER-COND
#13[0013.1] study: / NCTC P308/ -/ -/ -/ 132HR 50UM EXPER-COND
#14[0014.1] study: / NCTC P308/ -/ -/ -/ 132HR 50UM EXPER-COND
#15[0015.1] study: / NCTC P308/ -/ -/ -/ 132HR 50UM EXPER-COND
#16[0016.1] study: / NCTC P308/ -/ -/ -/ 132HR 50UM EXPER-COND
Correlation between Rspots in database. Pairing labels: PS
Paged CGL database file: /home/joeUser/gellab/pcg/jj1pcg.pcg
Using least square normalization.
```

```
Cor.Coef. of paired Rspot-set density
Rspot |  442 |  712 |  832 |  846
-----------------------------------
[ 407]|-0.385|-0.544|0.9108|-0.032
[ 442]|      |0.5807|0.2620|-0.470
[ 712]|      |      |0.4974|-0.668
[ 832]|      |      |      |-0.518

Number of spot pairs for two Rspot-sets considered
Rspot |  442 |  712 |  832 |  846
-----------------------------------
[ 407]|     6|     5|     5|     5
[ 442]|      |     8|     7|     7
[ 712]|      |      |     8|     6
[ 832]|      |      |      |     6

rFactor (Taylor's) of paired Rspot-set density
Rspot |  442 |  712 |  832 |  846
-----------------------------------
[ 407]|0.9531|1.1167|0.8926|0.6962
[ 442]|      |0.4754|0.7650|0.7808
[ 712]|      |      |0.7728|0.7942
[ 832]|      |      |      |0.8185
[06:45:35AM] Real TIME =00:00:20  CPU TIME =00:00:01,  5.00%

# [F.2] Generate ORDER TABLE for current SRL Rspots.
152<CMD>: inquire
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%).
 HELP to list CMDS?: ?

INQUIRE COMMANDS
----------------
                           --- SEARCHES ---
Behrens-Fisher t-test search for Rspot sets with given significance
        between classes (assuming different variance).
Expression-profile-test - for spots with min LSQ err < T.
F-test search for Rspot sets with given confidence limit between N classes.
Index search for entire Rspots meeting statistical prefilter limits,
*Kruskol-Wallis rank order search for Rspot sets with given significance
         between all defined classes for minimum of 5 gels/class.
LAndmark search for Rspot sets which are landmarks,
LEast squares linear search fitting Dens=m*ClassValue+b for
        corr.coef. < confidence limit.
Missing class search for Rspots where (only 1 of 2 classes present),
% search for Rspots where mean ratios of 2 classes: MAX(c1/c2,c2/c1) > T%,
Rank order (or WMW) search for Rspot sets with given significance between
        2 classes.
Search for entire Rspot sets meeting statistical prefilter limits,
T-test search for Rspot sets with given confidence limit between
        classes (assuming same variance).
TB-search using F-stat. to select T- or B-F-T-test per Rspot set.
TC-search using confidence-limits t-test search.
```

```
                          --- TABLES ---
CHange histogram of mean ratios of last SRL for two classes.
COordinate pairs of last SRL for two classes and %Err,Tpie,Tmw.
OExpression-profile print Expression Profile table and linkage of SRL spots.
ORder Rspots table by all class pairs (>,<,-) for spots in SRL.
Print Rspot set i, '$' (search results Rspots), 'S' to list SRL #s,

                  --- OPTIONAL SECONDARY SWITCHES ---
     (1) Append /FILE to any command to log its output in a file
         with a '.inq' file to be specified.
     (2) Append /LOGDENSITY to any command for density to
         be recomputed as LOG(1+density) prior to its use in any
         operation.
     (3) Append /STUDYVALUE or default /CLASSVALUE to determine the
         independent variable for 'LEAST-SQRS-LINEAR-SCH' command.
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%).
 HELP to list CMDS?: order table
Order Rspots by class pairs for SRL.


Rspot:  m1/2 m1/3 m2/3
------------------------
  407    >    >    >
  442    <    <    <
  712    <    <    <
  832    -    -    <
  846    <    <    >


Rspot:  m1/2 m1/3 m2/3
------------------------
  407    4.0117.4 29.0
  442    0.9  0.0  0.1
  712    0.1  0.0  0.2
  832    -    -    0.4
  846    0.0  0.2  5.1
[06:46:18AM] Real TIME =00:00:19  CPU TIME =00:00:01,  5.26%

# [G.1] OK - now we are going to get back to generating a Rmap image
# and display it (on X-windows so you won't see it in the listing!).
# First add the eRspots back to the PREFILTER labels.
153<CMD>: set label

Enter spot pairing-reliability label `search-pattern`
 consisting of (A, S, P, U, E, C, X (and *))
    A is Ambiguous Pair, S is sure Pair, P is Possible Pair,
    U is Unresolved Spot, C is composite Pair,
    E is Extrapolated Pair,
    X allows accessing the eRspot database,
    XX allows accessing ONLY the eRspot database.
 [PS] ?: psue
New pairing search label [PSUE]
```

```
[06:46:29AM] Real TIME =00:00:04  CPU TIME =00:00:00,  0.00%

# [G.2] Draw a Rmap of the Rgel (i.e. gel 0010.1).
154<CMD>: rmap
Draw a Rmap (optionally around Rspot#) ?: /ppxplot    # image instead of plot.
Output file:[/home/joeUser/gellab/gen/rmap.sps]
Creating SPSS data file from PCGL DB:
   /home/joeUser/gellab/pcg/jj1pcg.pcg
markgel 0010.1 rmap.sps -large -graphscale -Xpix -name:./rmap.ppx
[06:46:51AM] Real TIME =00:00:17  CPU TIME =00:00:01,  5.88%

# [G.3] Generate a SPSS file for SRL[15].
155<CMD>: set srl
GLOBAL CMD SWITCHES: /Option:
PLOT SWITCHES: /PPXplot
SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], FIndkeywd,
  Intersection[/ListOfSRLs], List[/Dir/ListofSRLs], QueryRspot, REAd,
  RENUMBER, REStore, SPss[/Mosaic], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?: spss
Subset name (or # Or # range Or <ALL> Or <LAST> Or SRL[i])?: 15
Restoring SRL from subset [15] with 5 spots.
<<<MANUAL TB-TEST P=.90 CLASS:2,3, MAXOD=1.5, MIN 3GELS/CLASS>>>
Output file:[/home/joeUser/gellab/gen/jj1s15.sps]
Creating SPSS data file from PCGL DB:
   /home/joeUser/gellab/pcg/jj1pcg.pcg

SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], FIndkeywd,
  Intersection[/ListOfSRLs], List[/Dir/ListofSRLs], QueryRspot, REAd,
  RENUMBER, REStore, SPss[/Mosaic], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?:
[06:47:14AM] Real TIME =00:00:15  CPU TIME =00:00:01,  6.67%

# [G.4] I forgot to also generate the MOSAIC generation batch script
# so just do it again (overide the .SPS file as well).
156<CMD>: set srl subsets
GLOBAL CMD SWITCHES: /Option:
PLOT SWITCHES: /PPXplot
SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], FIndkeywd,
  Intersection[/ListOfSRLs], List[/Dir/ListofSRLs], QueryRspot, REAd,
  RENUMBER, REStore, SPss[/Mosaic], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?: spss/mosaic
Subset name (or # Or # range Or <ALL> Or <LAST> Or SRL[i])?: 15
Restoring SRL from subset [15] with 5 spots.
<<<MANUAL TB-TEST P=.90 CLASS:2,3, MAXOD=1.5, MIN 3GELS/CLASS>>>
Output file:[/home/joeUser/gellab/gen/jj1s15.sps]
Creating SPSS data file from PCGL DB:
   /home/joeUser/gellab/pcg/jj1pcg.pcg
```

```
Creating Batch script job: '/home/joeUser/jj1s15.do' to compute MOSAIC images.
Output file:[/home/joeUser/jj1s15.do]

SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], FIndkeywd,
  Intersection[/ListOfSRLs], List[/Dir/ListofSRLs], QueryRspot, REAd,
  RENUMBER, REStore, SPss[/Mosaic], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?:
[06:47:38AM] Real TIME =00:00:13  CPU TIME =00:00:01,  7.69%

# [G.5] Leave the database program. Note the total REAL TIME was about 12
# minutes.
157<CMD>: exit
GLOBAL CMD SWITCHES: /Option:
PLOT SWITCHES: /PPXplot
Total session times: Real TIME =00:12:59  CPU TIME =00:01:13,  9.37%

Saving PCG database
        /home/joeUser/gellab/pcg/jj1pcg.pcg
for later use.
To use database at a later time, run CGELP2 then declare the data
base using the SET DATABASE command or restart cgelp2 by
    cgelp2 -d /home/joeUser/gellab/pcg/jj1pcg.pcg

# [G.6] List the batch script to create MOSAIC PPX images from SRL subset
# number SRL[15].
37% cat jj1s15.do

#!/bin/csh
#JOB /home/joeUser/jj1s15.do - 09/21/1989, 06:47:35AM
#   Create mosaics for /home/joeUser/gellab/pcg/jj1pcg.pcg
# SRL subset[15]
# SRL title:MANUAL TB-TEST P=.90 CLASS:2,3, MAXOD=1.5, MIN 3GELS/CLASS
mosaic 407 jj1s15.sps -Quickbkgrd -Zoom:2X
mosaic 442 jj1s15.sps -Quickbkgrd -Zoom:2X
mosaic 712 jj1s15.sps -Quickbkgrd -Zoom:2X
mosaic 832 jj1s15.sps -Quickbkgrd -Zoom:2X
mosaic 846 jj1s15.sps -Quickbkgrd -Zoom:2X

# [G.7] List the SPSS data for just to show what it looks like. Note
# the replication of the PCG DB state at the time the SPSS file
# was created.
39% cat gellab/gen/jj1s15.sps

File: /home/joeUser/gellab/gen/jj1s15.sps 09/21/1989, 06:47:33AM from:
PCG DB: /home/joeUser/gellab/pcg/jj1pcg.pcg
Title:MANUAL TB-TEST P=.90 CLASS:2,3, MAXOD=1.5, MIN 3GELS/CLASS
Rspot Mapping file [].
Map Rspots to Fspots is OFF.
Ratio-List density mode calibration EXISTS.
Least-Squares density mode calibration EXISTS.
Image size [rowsXcolumns]=[512X512]
```

```
Relative distance limits [0.00,1024.00] pixels
Mn DL limits [0.00,1024.00] pixels
Mn DP limits [0.00,1024.00] pixels
MN area limits [0.00,5000.00] pixels**2
MN density (Mode: L, tot-density/spot) limits [0.00,6000.00]
Coef. variation: S.D./Mn Rset area limits [0.00,10.00]
Coef. variation: S.D./Mn Rset dens limits [0.00,10.00]
Spot OD difference limits [0.00,4.00] OD
Class difference t-Test, F-test, Rank order p-value limit is .90:
Check if # gels in Rspot set [0:16]
(x,y) calibration file: <NONE>
pIe Rgel limits [0.00,511.00] pixels
MW Rgel limits [0.00,511.00] pixels
Pairing-Search-Label [PSUE]
No SRL Subset search restriction is active.
Using least square normalization.

Class # 1(WITHOUT)=0001.1 0002.1 0003.1 0004.1 0005.1 0006.1
Class # 2(20UM)=0010.1 0007.1 0008.1 0009.1
Class # 3(50UM)=0011.1 0012.1 0013.1 0014.1 0015.1 0016.1
Class # 4(<NULL>)=
Class # 5(<NULL>)=
Class # 6(<NULL>)=
Class # 7(<NULL>)=
Class # 8(<NULL>)=
Class # 9(<NULL>)=
RSPOT# ACC# INDEX DENSL D' LABEL[0:5] LMSET[1:25] DP DL DX DY XABS YABS CLASS MNBKOD
  407 0012.1   252    2.79    2.30   2   8   7.0    42    13   -33   375   263 3  0.020
  407 0013.1   477    3.69    1.60   2   8   5.8    47    23   -41   363   275 3  0.020
  407 0014.1   362   11.85    1.70   2   8   2.2    42    20   -37   382   263 3  0.020
  407 0010.1   386   22.90   22.90   2   8   4.0    44    17   -39   355   256 2  0.020
  407 0009.1   466  191.79    6.30   2   8   4.0    44    22   -39   351   260 2  0.020
  407 0007.1   412  241.72   33.10   2   8   4.1    42    15   -35   350   242 2  0.020
  407 0008.1   262  252.91   33.70   2   8   2.2    42    17   -36   354   269 2  0.020
  407 0001.1   529  717.49    5.60   2   8   4.0    44    21   -39   350   277 1  0.020
  442 0010.1   477    2.60    2.60   1   8   3.5    10    -9    -7   329   288 2  0.020
  442 0012.1   306    4.50    3.70   2   8   5.0    14   -13    -7   349   289 3  0.020
  442 0002.1   302    6.68    1.40   1   8   3.5    10   -10    -3   375   322 1  0.020
  442 0008.1   320    7.19    1.40   1   8   3.0    10    -5    -6   332   299 2  0.020
  442 0007.1   497   11.46    2.30   1   8   4.4    10   -10    -2   325   275 2  0.020
  442 0016.1   342   24.01   15.50   2   8   6.0    15    -9   -12   357   288 3  0.020
  442 0014.1   410   78.79   11.30   2   8   9.0    17    -9   -15   353   285 3  0.020
  442 0015.1   433  149.51   65.40   1   8   0.0    10    -8    -6   341   296 3  0.020
  442 0011.1   334  175.01   40.00   1   8   2.2    13   -10    -9   359   313 3  0.020
  442 0013.1   578  467.45  108.20   1   8   4.0    10    -4    -6   336   310 3  0.020
  712 0004.1   445    1.55    1.20   2  15   2.2    25   -23     6    84   369 1  0.020
  712 0010.1   646    2.90    2.90   2  15   2.2    25   -26     9    83   357 2  0.020
  712 0008.1   495    5.65    1.10   2  15   1.3    26   -25     9    88   368 2  0.020
  712 0015.1   570   21.38   15.30   1  15   2.2    27   -26     9   109   359 3  0.020
  712 0007.1   650   27.91    5.60   1  15   4.4    25   -20    10    91   332 2  0.020
  712 0012.1   457   57.24   30.00   1  15   1.0    25   -23     8   112   361 3  0.020
  712 0016.1   490   57.87   28.10   1  15   3.0    25   -21     7   122   357 3  0.020
  712 0013.1   747  100.21   28.50   1  15   0.0    25   -24     8   101   377 3  0.020
```

```
  712 0011.1   452  154.62   36.40   1  15   3.0    25   -21     7   121   370 3  0.020
  832 0007.1   856   27.91    5.60   2  17   4.0    35   -22    24   158   441 2  0.020
  832 0003.1   604   95.04   62.60   2  17   5.0    35   -25    25   154   491 1  0.020
  832 0010.1   907  116.50  116.50   2  17   3.0    36   -21    28   163   473 2  0.020
  832 0004.1   658  181.87   78.60   2  17   4.4    36   -25    26   158   483 1  0.020
  832 0016.1   642  190.34   77.40   2  17   7.1    41   -28    30   186   479 3  0.020
  832 0008.1   729  193.32   27.90   2  17   4.0    35   -17    28   167   479 2  0.020
  832 0015.1   729  235.70   99.10   2  17   3.5    38   -23    31   177   478 3  0.020
  832 0011.1   641  318.93   65.40   2  17   2.2    36   -24    27   192   489 3  0.020
  832 0012.1   681  468.91  199.40   2  17   9.1    44   -27    35   176   488 3  0.020
  832 0001.1   842 42313.62  172.70   2  17   3.0    36   -20    31   169   490 1  0.020
  846 0006.1   579    1.07    1.10   2  17   4.0    69    43    50   228   504 1  0.020
  846 0016.1   677    2.55    1.90   2  17   3.0    70    46    51   260   500 3  0.020
  846 0015.1   764    3.20    2.50   2  17   6.0    73    51    53   251   500 3  0.020
  846 0012.1   693    8.02    6.60   2  17   6.0    70    44    48   247   501 3  0.020
  846 0014.1   783   13.95    2.00   2  17   5.0    75    48    58   259   504 3  0.020
  846 0008.1   766   19.01    3.70   2  17   6.4    70    49    49   233   500 2  0.020
  846 0007.1   899   28.90    5.80   2  17   2.2    70    44    52   224   469 2  0.020
  846 0010.1   964   58.00   58.00   2  17   4.0    69    45    54   229   499 2  0.020

# [G.8] Create Mosaic images to view later by running the generated script in
# the UNIX background. The '>&' means save all program output in a log file
# and the final '&' means put it into the background. The '[1] 19938' means
# that UNIX has created job [1] and given it a UNIX process number 19938.
# It will tell you later when it is finished. You can type 'jobs' to
# see if it is still running.
41% jj1s15.do >& jj1s15.log &

[1] 19938

# [G.9] Meanwhile you can restart PCG DB program can continue doing additional
# searches and exploratory data analysis.
42% cgelp2 -d jj1pcg

CGELP2 - September 20, 1989. Todays date:09/21/1989, 06:50:06AM
 Type 'HELP' for list of CMDS. Type ?? for hints on APROPOS usage.
Written 1981-1989, P. Lemkin.

Using existing PCG paged composite gel database:
        /home/joeUser/gellab/pcg/jj1pcg.pcg
   Date created: 09/20/1989, 06:31:01
   Date last session: 09/21/1989, 06:47:47
[06:50:12AM] Real TIME =00:00:03  CPU TIME =00:00:01,  33.33%

# [G.10] Continuing with the data exploration...
159<CMD>:

   .... etc. ....
```

Chapter 3

GELLAB-II Program Descriptions

This chapter describes the GELLAB-II programs including usage, lists of command line switches, and examples. The programs which are fully implemented and reliably working are listed in the table on page 49.

These programs have a consistent command line switch interface. In general, you specify the name of the program followed by additional optional arguments. Note that in general, you need only type that initial part of a switch which makes it unique from other switches for the same program. This is indicated throughout the lists of switches in this Chapter by that part of the switch being upper case in the usage printout. Any switch may be negated by preceding it with a 'no' eg. `-noinfo`. In addition, switches are case independent (eg `-auto` is equivalent to `-Auto` or `-AUTO`). Section 1.4.1 page 45 discusses the `-info`, `-version` and `-usage` switches common to all of these programs.

common user interface

3.1 accppx - Display gel images by accession numbers

Program **accppx** is used to lookup the picture file name associated with the accession number argument(s) and then invoke **Xpix** on that file(s). If two gels ACC# arguments are given, then `Xpix -two` is invoked. [1] The additional **Xpix** switches `-cycle -mark` are also used. If you specify *GraphScale*, the switch is also passed on to **Xpix**. PPX files which begin with `l, m, t, u, v, w` will have the `-GraphScale` switch set automatically unless you overide it by specifying `-NOGraphScale`. The `-laser<:`*opt. printer name*> switch can be used to dump the iamges to the laser printer using `ppx2ps -height:7` *Rmap.ppx* `| lpr -Pprinter` instead of to **Xpix**. If <:*opt. printer name*> is prefaced with `@`, then it is assumed to be a script and `ppx2ps -height:7` *Rmap.ppx* `| printer` is used instead. If you want to display a derived image (on paths `ppnp2x` or `ppnp3x`) use `-Prefix:letterName` as well (eg. `-Prefix:m` to display the current Rmap image, `-Prefix:z` for the **sg2gii** segmented image, etc). If `-P1:letterName` and/or `-P2:letterName` are specified, then only use the prefix letter for the corresponding image. If you specify a picture file name instead of a gel accession number, it tries to use the name. It looks for the image file on `ppnp1x, ppnp2x, ppnp3x` paths in that order. If the image is not found in these directories, then it looks in all the directories in your UNIX `$PATH` environment variable which is normally defined in your `.cshrc` file. You may mix accession numbers and picture file names, but remember that the appropriate prefix switch `-P1, -P2` should be used. The default switches passed to **Xpix** are `-cycle -mark -ppx -full`.

gel image display

Flicker comparison

Note that **Xpix** has a `compare gels` mode in the `CURSOR OPS` menu to flicker two images. When used with two different gels, you can flicker compare two local gel regions. By moving one image against the other, the right zoom-window display performs a kind of eye-brain differencing of the images. [You must be positioned close to the zoom window to get this effect.] Using the mouse, you should select what you suspect are the same prominant spots in similar morphologic regions in the two gels. As you flicker the two gels, you will see a shift of these morphologic regions. Move the mouse to minimize this shift. When you are almost aligned, the flickering image will begin to pulse. This is strongest when they are ideally aligned. It is helpful to slow down and speedup the flicker which can be done using the mouse as described in the menu when `compare images` starts. Select `EXIT` in the `VIEW` menu to terminate **Xpix** and **accppx**.

it uses **Xpix**

USAGE:

[1] The X-Window System should be running - see Appendix D, page 603.

```
accppx <gel 1 ACC#> [gel 2 acc#>] [<opt. -Switches>]
```

Type `accppx -info` for more information on accppx.

SWITCHES

-Graphscale Display image in graphscale color instead of grayscale. This switch is passed on to **Xpix** to display "GraphScale" pseudo-color images generated by **markgel**, **mosaic**, bf cmpgl2, etc. This is the default for prefix letters: `l, m, t, u, v, w`.

-Info print more information on accppx.

-LASer:<opt. printer-name> use `ppx2ps | lpr -Plaser` to print the images to the laser printer instead of to **Xpix**. If printer-name starts with `@`, then assume it is a script instead of using `lpr -P`.

-Prefix:letter display the image for the gel acc# but with letter prefix instead of the original gel image.

-P1:letter display the image for the gel1 acc# but with letter prefix instead of the original gel image.

-P2:letter display the image for the gel2 acc# but with letter prefix instead of the original gel image.

-SIlent do not print output on the terminal.

-SWlist:"<List of switches>" allows passing of a switch list to child **Xpix** process. For example: `-swList:"-complement -rgb"` passes the switches `-complement -rgb`.

-Usage print UNIX command level switch usage.

-Version print the version of the program

-WmWait When done, wait until do CLICK TO EXIT `mwwait` widget to exit.

-ZoomRight:*n*`X`*,xC,yC* put the `Xpix` micro zoom windows on the right, or if `-LASER` is used, then print a *n*`X` zoomed region at (*xC,yC*).

EXAMPLES OF USAGE

```
accppx 324.1
          # Display one gel image

accppx 324.1 384.1
          # Display two gel images side by side which can be flickered
```

```
accppx 324.1 95.2 -Prefix:z
          # Display two segmented gel images

accppx 324.1 -Prefix:m
          # Display Rmap gel image

accppx 324.1 -Prefix:m -NOgraphScale
          # Display Rmap gel image, but without GraphScale.

accppx 324.1 384.1 -Prefix:m
          # Display two gel Rmaps side by side which can be flickered.

accppx 384.1 384.1 -P2:z
          # Display original and segmented image of same gel.

accppx 384.1 384.1 -P1:m -P2:z
          # Display Rmap and segmented image of same gel.

accppx 384.1 384.1 -P1:m -P2:z -Laser:laser
          # Print same images as above to 'laser' laser printer.

accppx 384.1 384.1 -P1:m -P2:z -Laser:@qms
          # Print same images as above to 'qms' laser printer
          # using 'qms' script.

accppx w00087.ppx
          # Display mosaic image w00087.ppx

accppx w00087
          # Display mosaic image w00087.ppx

accppx 324.1 w00087.ppx
          # Display original gel image and a mosaic image.

accppx 324.1 w00087.ppx -p1:m
          # Display Rmap gel image and a mosaic image.
```

3.2 camera - Acquire PPX image at NCI/IPS

The **camera** program is used to acquire and transfer an image from the Datacopy 612F charge coupled device (CCD) camera[2] to a PPX formated disk file. It is also used to control other camera options. A UNIX device driver for the GPII was written by Chris Johnson.[3]

The default image size is 1024x1024x8-bits with white being 0 and black being 255 (using the `-1Kx1K` switch). A larger size image is be specified by using either the `-CW:`*x1,x2,y1,y2* or `-SIZE:nRows,nCols` switches. The specified scanning frame is relative to the center of field of the camera. It creates a PPX formated image. The `-COMPLEMENT` switch can reverse the sense of black and white (255,0) to (0,255) in the scanned image. *image size*

Before performing a series of scans of similar material, it is recommended that you adjust the light level. This is done by rescanning the center line repeatedly using the `-SCANLINE` switch. As the center scan line is rescanned, it prints the minimum and maximum gray values seen in the line as well as the entropy of the gray scale histogram of the line. Minimizing entropy when scanning a sharp edged object (such as a wedge) can be used to attain optimal focus. By adjusting the light intensity or camera *f*-stop, one can also optimize the dynamic range as seen in the gray scale minima and maxima. You must type `CONTROL/C` to exit this mode.

Other types of scanner maintenance is possible with **camera**. Using the `-PRINTPARAMBLOCK` and `-CHANGEPARAMBLOCK` it is possible to print and change the Datacopy camera GPII controller parameter block. Normally, the Datacopy-SUN interface automatically resets it setl when turned on. However, the `-RESET` option forces the Datacopy GPII camera controller and camera to be reset with a default parameter block (1728x2810 size image). *NOTE: you should always do a* `-RESET` *if the camera was turned off prior to doing a series of scans.* After doing a reset, wait about a minute for the camera to become ready. Only one reset is necessary for doing a series of scans.

The **camera** program is used in GELLAB-II by the **getacc** program to do the actual image scanning. This is done via a UNIX `rsh` to the machine which actually has the camera attached. Note that the **camera** program must actually be executed on the computer which has the camera attached whereas the **getacc** program need not. If you try to run the **camera** program on the wrong computer, it will do a `rsh` to the correct machine - so it will work out in the end but just take a little *use with* **getacc**

[2] The 612F CCD camera with a GPII (General Purpose Instrument Interface) is made by Datacopy Inc., Mountain View, California. The camera is capable of 1728 8-bit pixels/line with 2810 lines. A 1024x1024 pixel image takes about 1 minute, a 512x512 about 20 seconds.

[3] Sun Coast SoftWorks, Inc. P.O.Box 1992 Palm Harbor, FL 34682. (813)784-0072. Binary device drivers are available from NCI (free) and driver source code from Sun Coast SoftWorks, Inc.

longer. The `-PATH:pathToUse` switch is included for use by **camera** itself when it must recursively run itself on the proper machine and needs to specify the current path when it does the `rsh`.

USAGE:

```
camera <pix-file-name> [<optional switches>]
```

Type `camera -info` to get more information.

SWITCHES

-Average:n use average of n Datacopy images instead of one.

-CHangeParamBlock read, edit and write back the camera parameter block.

-COmplement complement the gray value image, from [0:255] [black:white] to [white:black], before saving it on disk (default).

-CW:x1,x2,y1,y2 force Computing Window to this new value.

-DBug:cmd evaluate GPII `/dev/gpii0` camera driver debugging option (only one may be specified), where:

- DPRINT:level - set DPRINT trace level to 0, 1 or 2.
- TRACE:level - set TRACE procedure call trace level to 0, or 1.
- SCSI:level - set SCSI command block trace level to 0, 1 or 2.
- CLOSEDRIVER - force /dev/gpii0 to close().

-DUmpline when doing a RESCANLINE, also dump the actual gray values.

-GelMgr:<*GELLABMANAGER path*> needed if camera is to be called recursively.

-Info print more information on camera.

-OverridePPXfilename overide existing PPX file with same name.

-PAth:newpath use the new path to read the `gel.rc` file instead of the current path.

-PRintParamBlock read and print the camera parameter block.

-RESET do a RESET on the Datacopy GPII camera controller.

-SCanLine:<Opt. width> continuously rescan center line of camera until type `CONTROL/C`.

-SIze:nRows,nColumns - specify image size constrained by *nRows* $*$ *nColumns* is a multiple of 512 pixels.

-Usage print UNIX command level switch usage.

-Version print the version of the program

-WmWait When done, wait until do CLICK TO EXIT `mwwait` widget to exit.

-1Kx1K scan a 1024x1024 pixel size PPX file.

EXAMPLES OF USAGE

```
camera a00123
      # Scan a 512x512 image into the file a00123.ppx.

camera a00123 -overwrite
      # Same as above, but overwrite the file if it
      # existed previously.

camera a00123.ppx -1KX1K
      # Scan a 1024x1024 image into the file.

camera a00123.ppx -size:1536,1536
      # Scan a 1536x15364 image into the file. Note
      # rowsXcolumns must be multiple of 512.

camera a00123.ppx -complement
      # Scan a 512x512 image but swap black and white.

camera -print
      # Print current camera parameter block.

camera -scanline
      # Rescan center 1024 pixel line and print min, max, entropy.

camera -scanline -dumpline
      # Same as above but also print the gray scale values

camera -scanline:512 -dumpline
      # Same as above but also print the gray scale values

camera -reset
      # RESET the Datacopy GPII controller.

camera -dbug:DPRINT:2
      # Enable full DPRINT tracing in /dev/gpii0 driver.

camera -dbug:SCSI:0
      # Disable SCSI trace in /dev/gpii0 driver.
```

3.3 cgelp2 - multiple 2D gel spot analysis

The **cgelp2** program is used to create a composite 2D gel (CGL) database and then analyze polypeptide spots in 2D gels. It is an interactive program and is used to help perform the exploratory data analysis. This Section contains more tutorial information then the other program descriptions because of its increased complexity.

This part of GELLAB was first described as CGEL in [LipL80a], with extensions discussed in ([LipL84], [LemP81c], [LesE81b], [LemP81d], [LemP83a], [LemP82a], [LemP83b], [HowR83], [LemP84], [SonP85], [SonP85], [LemP89a], [LemP89b],). Other papers discussing the use of **cgelp2** are ([LemP80a], [LipL80a], [LesE80], [LesE81a], [LesE82a], [LesE82b], [LesE83], [LemP83b], [HowR83], [LesE84a], [LemP84], [SonP85], [LesE84b], [SonP86]). Section 5.1 discusses **cgelp2** search strategies and is updated material initially presented in [LemP83a].

Section 2.3, page 105 gives a tutorial with examples using **cgelp2**. When a large number of gels is used, the entire composite gel database can not be kept in memory at the same time. This led us to rewrite CGEL to us a disk based database and to *page* that part of the database required when it was needed. The database is then called a Paged Composite Gel (PCG) DB. This program was called CGELP (P for paged) in GELLAB-I. When it was converted to UNIX, it was renamed **cgelp2**.

why **cgelp2***?*

Part of performing an analysis may be in investigating the significance of spot differences for each Rspot set. That is, sweeping through the database and performing a particular statistica test on each Rspot set and then keeping track of which Rspot sets passed the test.

Cgelp2 is an interpreter program which, once started, accepts commands typed by the investigator to perform various operations on the PCG DB.[4] Alternatively, **cgelp2** may be started with a graphical user interface and command can be entered either by clicking on menu buttons or by typing commands as before. This continues until they issue the command (`EXIT`) to terminate the program. To minimize confusion, in most of the following discussion, we will illustrate commands as if they were typed, but as just stated, you can enter commands by clicking on menu selections. Some commands have sub-interpreters which behave in a similar manner. The philosophy being that a sub-interpreter presents an environment where one would typically issue several commands to manipulate specific type data (e.g. gel subsets or Rspot subsets). All commands and subcommands are entered in the history journal (to be discussed) where they may be reviewed, edited and re-evaluated. The journal is saved as a permanent file and restored when leaving and restarting **cgelp2** respectively.

interpreter

[4] **cgelp2** comands are prompted for by the computer typing <CMD> and we sometimes refer to these top level commands by this symbol.

Constructing a PCG DB

First, **Cgelp2** constructs the composite gel database from a set of paired gel spot lists (i.e. GCF files). This set of gels is assumed to have been previously segmented with the **sg2gii** program (Section 3.18, page 452) and then compared (two gels at a time) with a representative *Rgel* (or reference) gel using the **cmpgl2** program (Section 3.4, page 322). Initially, the set of **cmpgl2** *.gcf* output files are read into corresponding spot *Rspot sets*. Individual spots in Rspot sets can be rank-ordered by gel density). Although we call it an Rspot *set*, it is actually an ordered *list* of spots which can be thought of as ordered or not depending on how it is used. The particular spot order is a function of density calibration and when the `REORDER` command is used which will be discussed later.

creating a PCG DB

The assumption is made that a single, fairly-good gel, denoted the representative gel or Rgel, is used in all gel comparisons. Thus the Rgel is used to make the connection between all gels in the data base (termed the *PCG* or *paged CGL database* which is a file with a *.pcg* extension).[5] Since all spots are mapped to the coordinate domain of the Rgel, $n-1$ gel comparisons are required for n gels. For example given a set of six gels 1.3, 2.3, 3.1, 4.3, 5.1 and 6.1. Let the Rgel be gel `2.3`. Then the set of **cmpgl2** produced input files would be:

Rgel

`c30001.gcf, c10003.gcf, c30004.gcf, c10005.gcf, c10006.gcf`.

Then these files imply that gels `1.3, 3.1, 4.3, 5.1` and `6.1` are the gels paired with gel 2.3. The `CREATE` command is used to specify gels to add when constructing or extending the PCG DB.

These Rspot sets of corresponding spots (see Figure 1.2, page 21) constructed are a subset of all spots in the GCF files restricted by the subset of allowable pairing labels (described in Section 3.4 and ([LipL80a], [LemP81b])). Possible pairing labels allowed are: SP (sure-pair), PP (possible-pair), AP (ambiguous-pair), US (unresolved-spot), CP (Composite-Pair) are discussed in ALGORITHM CMPGL2 in Appendix G, page 611 and in ALGORITHM CREATE-CGL-DB.

Rspot sets

When building the composite gel database, spots for gels *missing* from Rspot sets may be extrapolated into these Rspot sets. Such spots are called EP (extrapolated pairs). They are created two different ways. First, during the construction of the initial PCG DB if the US pairing label is specified, then US spots found in the Rgel are always included in the database as Rspots. However, US *not* found in the Rgel are included as extended or *eRspot* sets. This initial extended PCG DB is constructed using the `CREATE/ERSPOT` command described on page Section 196) also in the same Appendix.

eRspot sets

Later, one can extrapolate missing spots to *all* gels in all Rspot and eRspot sets. Then, EP spots are created where required to fill in for missing spots using the `EXTRAPOLATE` command (see ([LemP82a], [LemP83a], [LemP89a])). The algorithm

extrapolated spots

[5] The Rgel is selected *before* **cmpgl2** pairs the gels and its name is encoded in the GCF files.

is described in ALGORITHM EXTRAPOLATE in Appendix G, page 611. As just discussed, spots not found in the Rgel are extrapolated in the Rgel (and other gels as well) so that all spots found in *any* gel in the PCG DB can be handled.

There is also another class of spots called GS spots (Garbage Spots) which can be manually assigned to spots using the **cgelp2** spot `EDIT` command.

The resultant PCG database consists of a sequential list of *Rspot sets* in the Rgel and all corresponding spots in other gels which belong to those Rspot sets. In [LipL80a] and Section 1.3.4, page 35, we discuss this concept of Rspot as an approximation to the *canonical spot* - the ideal morphologically-similar *corresponding spot* found in a set of gels. This is represented by a single ideal *canonical gel* or *Cgel*. An approximation to the Cgel called the *Cgel'* can be constructed using the `C-GEL'` command in **cgelp2** (see page 192).

canonical gel

Section 1.3, page 24 and Section 5.1, page 503 discusses how **cgelp2** fits into the strategy for analyzing a set of gels.

Using CGELP2

Being an interpreter program, **cgelp2** accepts commands typed in response to a `<CMD>:` prompt to perform specific operations. Alternatively, you can use the pull down menus in the when the graphical interface option is enabled (see page 162). However, in this Chapter we will discuss the commands for the most part as if they were typed. Typing `HELP` lists these commands (see Table 3.1 in Section 3.3.6). As was mentioned, some commands have their own subcommands. In particular, the `INQUIRE` command has a large number of database search subcommands. Section 3.3.8 lists the `INQUIRE` subcommands in Table 3.7. Detailed descriptions of all of these commands along with examples of usage are given in Section 3.3.10. Some of the other commands start a sub-interpreter which the investigator issues subcommands. When these terminate, control returns to the top command level. Thse sub-interpreters include: `INQUIRE, SET SRL SUBSETS, SET GEL SUBSETS`.

commands

A typical processing command sequence is illustrated in Section 3.9, Example 3, page 374 in **makjob** (see `ts3cgl.gdo` sample **cgelp2** batch script program in Section 3.19, page 375). In that example (see Figures 1 and 2 in the **Introduction** to clarify this), the **cgelp2** composite database is constructed by applying a sequence of GELLAB operations which first segment each gel into a Gel Segmentation File (GSF), then compare each segmented gel with a Rgel into a Gel Comparison File (GCF). Finally the list of GCFs is read into the **cgelp2** program to construct the paged database file (PCG DB). The **cgelp2** program may be initialized as illustrated by the `ts3cgl.gdo` script where lines proceeded by ';' or '#' are comments and by '*' are input commands to **cgelp2**.

batch scripts

Aborting an operation

Many of the commands sweep through the PCG DB performing various operations on Rspot sets. Any operation of this type can be aborted by typing `CONTROL/C`. If using the Graphical User Interface (GUI) then move the cursor into the dynamic Rmap before typing `CONTROL/C`.

aborting a command

3.3.1 Program usage and switches

Cgelp2, being an interpreter program, accepts terminal commands to perform specific operations. These may also be run in an interactive batch mode (see either the `-f` SWITCH below or `DO` command on page 207) (e.g. `cgelp2 -f ts3cgl.gdo`). By convention, we use the `.gdo` file extension only for **cgelp2** command files. See Section 1.6.1, page 56 on how to put commands into the UNIX background batch so that instead of being interactive, you can execute **cgelp2** command files in UNIX background batch.

USAGE:

```
cgelp2 [<Opt. -switches>]
```

Type `cgelp2 -info` to get more information.

SWITCHES

-d (-database) (-pcg) pcgDBfile startup with `SET DATABASE` on this file.

-f (-file) scriptFile get commands from a script file.

-g (-graphicMenu) use Dynamic Rmap with X-Windows MENU mode.

-i (-info) print more information on **cgelp2**.

-l (-log) logFile save terminal output in log file.

-path userPath new user path to put table files.

-r (-readonly) (-protect) set PCG DB to read-only (`PROTECT`).

-u -usage display command line format information.

-v or -version print current version number of the program.

-wmwait when done, wait until do CLICK TO EXIT `wmwait` widget to exit.

-zoomGui in `-GraphicMenu` mode, enable Dynamic Rmap zoom. Default is not to update zoom with mouse motion. This is useful with low speed network connection between the client and server.

3.3.2 Running cgelp2

Cgelp2 may be started with or without an existing PCG DB file by just typing in response to the UNIX *csh(1)* prompt (i.e. *#%*) : [6]

```
25% cgelp2
```

If the database exists, (for example as file `ts3pcg.pcg`) you might type:

```
1<CMD>: SET DATABASE FILE
Enter name of PCG paged composite gel database to use
 []?: ts3pcg.pcg<CR>
```

accessing a PCG DB

The default PCG DB directory is specified by the `gel.rc` file `ppnp4x` keyword associated entry (ususally being `./gellab/pcg`). Alternatively with an *existing* PCG DB file (e.g. `ts3pcg.pcg`), just type it all in one UNIX command line:

```
26% cgelp2 -database ts3pcg.pcg
```

or

```
27% cgelp2 -d ts3pcg.pcg
```

If you want to run a **cgelp2** <CMD> level script such as `ts3cgl.gdo`, then type:

```
28% cgelp2 -file ts3cgl.gd
```

or

```
29% cgelp2 -f ts3cgl.gdo
```

If you want to save the output from a script session in a log file, you can put it in UNIX background batch. The following example saves the output in log file `ts3cgl.log`.

```
30% cgelp2 -f ts3cgl.gdo >& ts3cgl.log&
```

You can also save the output in a log file using the `-log` switch.

```
31% cgelp2 -f ts3cgl.gdo -log ts3cgl.log
```

3.3.3 Using CGELP2 with X-Windows Graphical User Interface

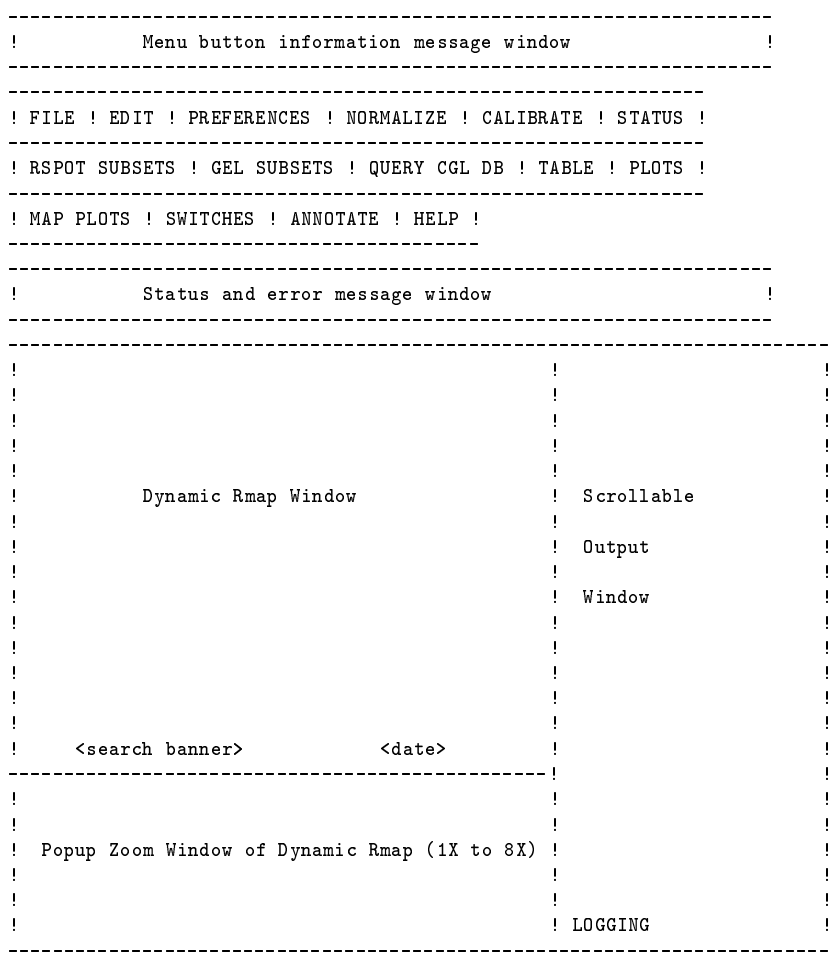
GUI interface

You can start **cgelp2** with an X11 graphical user interface for and existing PCG DB. When started, it puts up a composite of several windows illustrated below. It always sets up the *dynamic Rmap* for the Rgel. Therefore, only use the graphical user interface on existing PCG databases.

[6] The '`#`' in '`#%`' is the current UNIX shell history number (a decimal integer which is incremented each time a UNIX command line is evaluated). This is *different* from the **cgelp2** command history number (see Section 3.3.7, page 173). See references on the UNIX `csh` on how to use the history number.

```
32% cgelp2 -g -d ts3pcg.pcg
```

In the graphical user interface to **cgelp2**, the mouse may be used to select commands by pressing one of the pull-down menus and then selecting the command of interest. If you change your mind, then release the button when it is NOT on a menu entry. There are a number of accellerator keys and mouse-key combination commands which are listed in the table below.



```
        DYNAMIC RMAP MENU
 User Actions          Operations
============          ========================
Ctrl <Btn1Down>:      roi StartDrag
Ctrl <Btn1Motion>:    roi Drag
Ctrl <Btn1Up>:        roi StopDrag
Shift <Btn1Down>:     delete rspot for SRL by mouse
      <Btn1Down>:     add rspot for SRL by mouse
Shift <Btn3Down>:     print rspot by mouse
      <Btn3Down>:     query rspot by mouse
Button2 <Motion>:     change contrast
<Motion>:             zoom update
Ctrl <Key>     A:     toggle label rspots with annotation
Ctrl <Key>     C:     abort operation
Ctrl <Key>     D:     toggle menu debug
Ctrl <Key>     F:     toggle foreign spot mode
Ctrl <Key>     G:     toggle RGB  grayscale colors
Ctrl <Key>     H:     print image menu help
Ctrl <Key>     K:     clear all rspots from SRL
Ctrl <Key>     L:     toggle label dots all rspots in rmap
Ctrl <Key>     M:     change gel for dynamic rmap
Ctrl <Key>     P:     print rmap on laser printer
Ctrl <Key>     R:     toggle label rspots with rspot number
Ctrl <Key>     S:     define foreign spot
Ctrl <Key>     W:     toggle zoom enable
Ctrl <Key>     Z:     change zoom magnification
      <KeyPress>:     process CMD key
```

NOTE: to build a PCG DB with the `CREATE` command you *must* have previously constructed the Gel Comparison Files (GCF) with the **cmpgl2** program.

help!

When running **cgelp2**, you can get help by typing `HELP` or `HELP cmd-name` or `?cmd-name` for specific commands.

3.3.4 The 'PREFILTER' for the PCG DB

prefilter

The *prefilter* is discussed in [LemP83b] and Section 5.1.10, page 524 which defines it and discusses its role as a primary search constraint. Search constraints are divided into a primary and a secondary constraint. The prefilter is a primary statistical and logical test applied to a Rspot set prior to it's being used in the specified (secondary) operation. pIt checks: a) if the gels in the Rspot set are in the Working Set (WS) of gels; b) if the gels are in the desired experimental classes; c) if the Rspot sets are in the desired (pIe,MW) sub-region; d) if individual spots meet the spot-pairing label; e) if the Rspot set meets the current statistical limits for Rspot set features. If the prefilter fails for any reason, then the secondary test is not performed and assumed to have failed.

Prefilter failCode statistics

One of the problems when using the `INQUIRE` or other commands which use the PREFILTER is that if an Rspot set is rejected for computation by the PREFILTER, there is normally no way to find out why it failed. When used with the `/EXplain` switch (switches will be explained shortly) it causes the PREFILTER to print out why it failed and also to accumulate global *failCode* statistics. With `/NOEXplain` set, a failure will print out:

explain prefilter failures

```
Rspot[n] is NULL
```

for some spot n. If `/EXplain` is set it will report it as:

```
Rspot[n] FAILED-PREFILTER '<failcode> = failure explanation'.
```

The reported failure also prints out the class number if it is for a specific class. (eg):

```
Rspot[n] FAILED-PREFILTER '[Class#:n]<failcode> = failure explain'.
```

If some search is done in `INQUIRE` and the `/EXplain` switch is set, at the point where the Mean and ±StdErr is printed, it will also print out a summary histogram of the failCodes. E.G.

```
PREFILTER failCode Histogram
Freq=341  failCode=0 0 = PASSES ALL TESTS!
Freq=0    failCode=1 1 = Rspot set is NULL
Freq=0    failCode=2 2 = No gels in W.S. meet label, DP, DL or ODdf limits
Freq=382  failCode=3 3 = # gels outside of #-gels-required limits
Freq=0    failCode=4 4 = # gels meeting relative (dx,dy) limits == 0
Freq=0    failCode=5 5 = Rspot set Rgel position outside of pIe-MW region
Freq=356  failCode=6 6 = No gels in W.S. meet label limits
Freq=64   failCode=7 7 = No gels in W.S. meet DP limits
Freq=0    failCode=8 8 = No gels in W.S. meet DL limits
Freq=0    failCode=9 9 = No gels in W.S. meet ODdf limits
Freq=3    failCode=10 10 = Failed mean AREA statistics
Freq=203  failCode=11 11 = Failed mean DENSITY statistics
Freq=0    failCode=12 12 = Failed COV of AREA statistics
Freq=0    failCode=13 13 = Failed COV of DENSITY statistics
Freq=1302 failCode=14 14 = Failed Statistical test
Freq=47   failCode=15 15 = Passed Statistical test
```

3.3.5 cgelp2 generated files

If an output file specification is mentioned by the user in response to a file name request (rather than <CR>), the operation will use the user specified file name rather than generating a 6-digit numeric file name *nnnnnn* as: *nnnnnn*`.cgl`. Other file extensions include `.tbl, .ugf, .sps, .sas` and `.inq` (e.g. `000003.tbl`, etc). When creating a CGL database, the set of input files may alternatively be specified indirectly using the construction *@file.ccl* as an indirect input file.

generated files

extending PCG DB

The *.pcg* database file is a special image mode (i.e. binary) random access file the first approximately 300 blocks of which contains information on the state of the current **cgelp2** database. The remainder of the file is allocated sequentially to Rspot set data.

Because Rspot sets can grow, the Rspot set portion of the DB file may expand by extending the file and shuffling data. For the average user, the only facts about the database that need be remembered are: 1) it is a disk file with *.pcg* extension, 2) for a large database it could be a large file so sufficient space should be allocated, and 3) it is saved along with the current state of **cgelp2** by typing the `EXIT` or `BACKUP` commands. Running **cgelp2** on this PCG file at some time in the future will restore the state to the point *it was* prior to exiting **cgelp2**.

Setting user graphics display terminal if plotting

graphics plots

Some **cgelp2** commands (`DCPLOT, DDPLOT, DENDROGRAM, HISTOGRAM, MOSAIC, PLOT, RMAP` can generate graphics plots. A display selected from the test of legal displays needs to be assigned prior to performing that command. If a plot file is specified then it will have a ùgf file extension and may be plotted at some later time using the **plotn** program.. Note that the display/plot device is "sticky". To change it, use the `SET DISPLAY` command.

Use of the /SRLsubset:n switch with CGELP2 <CMD>s

/switch modifiers

Any of the **cgelp2** commands which operate on the entire PCG DB at top level (i.e. <CMD> level) may be restricted, as part of the prefilter, to that part of the PCG DB belonging to a particular set of Rspots (see `SET SRL SUBSETS`) denoted a *SRL subset*. Subsets are given numeric references n with respect to the prefilter. The Rspots to be used must have been set to a SRL subset previously (see `SET SRL SUBSET <CMD>`). The syntax to add a SRL subset restriction to the prefilter is,

<CMD> ··· /SRLsubset:n

or

<CMD> ··· /SRLSS[n]

<CMD< /SRLSS[n] modifier

Where <CMD> is one of the following commands which operates on Rspot set and n is a legal SRL subset. That is any Rspot sets which might be operated on by the command are prefiltered against those in the specified SRL subset first and only those in the specified SRL subset are used. The <CMD>s affected are: `C-GEL′`, `DCPLOT`, `DDPLOT`, `DENDROGRAM`, `EXTRAPOLATE`, `HISTOGRAM`, `INQUIRE`, `MOSAIC`, `PLOT`, `REORDER`, `RMAP`, `SAVE`, `SET RATIO LIST`, `SET SRL SUBSET`, `SPSS`, `TABULATE` (i.e. Correlate <SUBCMD>). The '*' prefixing some commands indicates those which are not fully operational at this time.

3.3.6 cgelp2 commands

The top level **cgelp2** commands are listed here in Table 3.1. This list is printed in response to the `HELP` command. The user types a command in response to the `<CMD>:` prompt. Commands which search through the PCG DB may be aborted by typing (both keys together) `CONTROL/C`. You need only type enough of the command to make it unique (indicated by prefix upper case letters). HINT: to get detailed help, type `HELP HELP`. For summaries of commands use `??` or `?APROPOS phrase`.

TABLE 3.1. Top-level **cgelp2** <CMD>s. Top-level **cgelp2** commands available at '<CMD>' level. Commands with '*' prefix are not fully operational at this time.

ABORT session and do NOT save PCG database for later use.

BACkup backup the working PCG DB onto the current paged DB file.

BIndings list key and mouse bindings in Menu mode.

***BUbblePlot** probability from t-statistic that means of c1,c2 are different.

C-GEL′ create a C-gel′ to estimate a set of replicate gels.

CCplot plot (*.ugf*) LOG density/density class-class plots from CGL DB.

***CLOSE** the previously opened PCG database.

COALESCE the PCG DB to optimize Rspot set access.

CReate create a CGL database from a set of CMPGL2 *.gcf* files.

DBUG:0nnnnn enter debugging option bits (if debugging enabled).

DCplot plot (*.ugf*) $\log density/classValue$ for specified Rspot.

DDplot plot (*.ugf*) $\log density/density$ spot plots from PCG DB.

DELete pcg db delete all file for the specified PCG database.

DENDROGRAM plot (*.ugf*) of gels=Fct(SRL) or SRL=fct(expr-profiles).

DO execute commands from a script file

DUmp CGL DUmp the CGL database in an ASCII (*.cgl*) file.

EDit spots in the PCG database.

EXPplot plot (*.ugf*) Expression Profile for Rspots in SRL.

EXIt cgelp2 to monitor and save PCG database file for later use.

EXTrapolate missing spots in Rsets from mean $(dx, dy) + LM_{position}$.

Features list current maximum values for various spot features.

Gels lists names, total densities, study of current gels.

HElp Print this message.

HISTOGram - compute/plot Rspot sets feature histograms (*.tbl*/*.ugf*).

HISTORy - list the **cgelp2** history commands.

INFormation print the CGELP2 general information message.

INQuire Interrogate the CGL database for particular spots.

Limits print the current statistical limits.

***MEMo** use Rspot memos.

***MERgeAP** merge AP's with SP or PP if meet DP limits.

MOsaic generate a mosaic plot/image around the specified Rspot.

***OPen PCG DB** specify and open a new PCG database.

PLot feature vs. feature plot of 2 (or 3) spot features over PCG DB.

***PROBabilityPlot** of all Rspots with D' vs $(D' - MeanD')/(stdDevD'/\sqrt{(n)})$.

PROTect paged CGL database for read-only (toggle).

REOrder Rspot sets (after changing density mode).

***REMove** a gel from the CGL paged database.

RMap generate a Rmap plot/image surrounding optional Rspot.

SEQuential set operation of adjacent search result list subsets.

SET ACcession file name change the default 'gel.id' name.

SET ANnotation spot feature annotation.

SET CAlibration for density/unit-area, file for(pIe,MW) as fct of (X,Y)

SET CLasses Define gel experimental class partition.

SET DAtabase file define or access CGL data paged database.

SET DEnsity mode in Abs(D'), Uncorrected, Percent, Ratio, Volume, LSQ, CPM units.

SET DIsplay plotting device to 4010, XWIND, PPX, LASER, PS, xxxxPLOT.

SET FIelds Set the list of accession fields desired for gel labeling.

SET FOreign spot map define Rspot to/from Fspot protein name mapping.

SET Gel subset define/operate on gel subsets.

SET LAbel Set the 'Label' code to (S,P,A,U,E,C) used in searching.

SET LEast squares density normalization calibration to Rgel.

SET MOre toggle 'more' style output switch for the terminal.

***SET NAmes of gels** change the alternative gel names.

SET PArameters subset define/operate on parameters subsets.

SET PRefilter limits define PREFILTER limits from X-window dialog form.

SET RAtio compute Ratio-Mode Rspots spot mean density normalization.

SET REgion of pIe and MW subregion in the Rgel to restrict CGL DB.

SET RGel Set the name of the Rgel used in searching.

***SET SPot view** define 'view' of a spot's features to dump on SAS/SPSS/etc.

SET SRl subset define/operate on Search Results List subsets.

SET STatistics limits Set statistics limits for use in searching.

SET Working gels Define working set of gels from CGL database.

SPss Dump a SPSS .sps summary file of part of the CGL database.

SYstem Evaluate a command on the underlying operating system.

TAbulate Correlate-gels, SRL-correlate, or Rank-order table (.tbl).

TImer print the run and cpu times for commands (toggle).

VAlid landmarks list the valid landmarks for each gel in a table.

VErify PCG DB test if PCG DB corrupted. Verify Rspot checksums.

!<CR> list the <CMD> history options.

?<partial CMD> list commands which start with the partial CMD.

?APROPOS <phrase> list commands which contains this phrase.

Optional Global <CMD> level Switches

The following switches may be appended to any of the above top level commands. Top level switches listed in Table 3.2 are appended to the command. Unlike UNIX, these switches are indicated by a '/' prefix. Switches may be negated using a `/NO` prefix (e.g. `/NOChangeHistogram`). For example, `INQUIRE/EXPLAIN`. If a command expects additional arguments, these may be supplied in advance on the same command line using a '//' notation. Be careful to note that a single '/' prefaces a switch while a double '//' prefaces an additional argument(s) expected by the command. If you do not supply the additional argument(s), the command will prompt for them.

use of '/' and '//'

For example `INQUIRE/EXPLAIN//t-test//1,2classes`. In the case, the first extra argument `t-test` is specified, then the second `1,2classes`. You can embed switches on inner arguments as for example `INQUIRE//PRINT/LOG//123////`. Note that `PRINT/LOG` is the first argument, `123` the second, and the third `////` is equivalent to <CR>. If you are in command mode, i.e *not* menu mode, then switches are valid only for the top level command in which they appear. In GUI menu mode, switches are sticky.

'//' subarguments

TABLE 3.2. Top-level **cgelp2** '/' switches.

/CAlibrateMWpIe print(x,y) as (pIe,MW) if calibrated with `SET CALIBRATE`.

/CHangeHistogram print it after `INQUIRE` 2-class search.

/EDifferencePrint w/`CREATE` print Euclidian Difference in LM data.

/EPspot use EP (0 density) spots in Rspot distribution calculations.

/ERspot with `CREATE`, it will include non-Rgel US as Rgel EP spots.

/EXplain why Rspot set failed the prefilter.

/FIND:ACC#,CC#,DBUGcode print out Rspot# if ever find this spot&set DBUG.

/FSpot use foreign spot instead of Rspot number.

/FUll print full specification for the operation.

/HeaderOnly print only Rspot set summary header - not tabular data.

/LOGDensity use log of density instead of density in all calculations.

/MEDian use median instead of mean calculation in Rspot statistics.

/Option:stringArg extra argument used by some operations.

/PRecision use 7 digits of precision in printing means, std-devs.

/Quiet do not output to terminal during calculations if applicable.

/SASformat generate SAS instead of SPSS output if appropriate.

/SINgle gel for Cgel' estimate if appropriate.

/SOrtByGel SPSS (*.sps*) or SAS (*.sas*) output file by gel number rather than density.

/SRLSS:n prefilter the operation by SRL subset *n* if appropriate.

/WorryMsg print DON'T WORRY message every 30 seconds during searches.

Some of these switches are discussed in more detail whereas others are discussed where they are used in the various commands.

The `/NOEPspot` switch can be used to disable counting zero EP density values in statistical calculations. If `SET LABEL` has 'EP' then it will automatically invoke `/EPspot` - unless `/NOEPspot` switch is explicitly set. The implication is that, with `/EPspot` set, the mean and standard deviation statistics computed on an Rspot set use a zero value for density for all EP spots which are counted as a spot for statistical purposes. *EP spot data*

The `/OPTION:stringArg` switch can be used to supply additional optional arguments to commands. For example `GELS/Option:0324.1` and `VALIDLANDMARKS/Option:0324.1` will restrict the `GELS` and `VALIDLANDMARKS` commands to only report on gel `324.1` respectively.

The `/EXPLAIN` switch can be used with `INQUIRE` to explain why individual Rspots failed the PREFILTER. It also prints a summary at the end. This is described in more detail on page 164.

Optional plot-type commands secondary switches

plot switches

The plot commands which include `CCPLOT, DCPLOT, DDPLOT, DENDROGRAM, EXPPLOT, HISTOGRAM, MOSAIC, PLOT, RMAP` have switch modifiers which can alter the plots. Not all switches apply to all plot commands. Check the individual plot commands to see which ones apply. Note that the display used by these commands is declared with `SET DISPLAY`. These switches are given in Table 3.3.

TABLE 3.3. Plot-type **cgelp2** '**/**' secondary switches typed with the user's response to various plot-type commands.

/ALLgels - use all gels in W.S in mosaics.

/ANGLE:nDegrees - to rotate 3D feature plot n degrees.

/CENTER:x,y - center Rmap using explicit x,y.

/CLASSNAMES - use class names instead of numbers in mosaics.

/COLOR - grayscale or color spots RED(S+P), GREEN(A+E), BLUE(U+C).

/DUMPPPXPLOT - dump the PPXplot image on the laser printer.

/FILL Label - fill labeled spots in BLACK.

/GELacc#LABELS - draw gel ACC# labels.

/HFLIPpIe - flip pIe (horizontal) axis.

/LABEL - label drawn spots.

/LINE - draw lines instead of dots in scatter plot.

/LOGPLOT - draw log of data.

/MAKEUP:nXn - set mosaic makeup in range of 2x2 to 12x12.

/MW - project 2D gel values on MW axis to estimate 1D gel.

/PIE - project 2D gel values on pIe axis to estimate 1D gel.

/PPXplot - generate PPX image on **Xpix** display instead of plot.

/SCALE:densS.F., classS.F. - scale plot by these scale factors.

/SIZEbyD' - draw spots the size estimated by it's Density else use spot's (S_x, S_y) estimate.

/SRLlabel - draw labels on just SRL Rspots.

/TITLE:'title' - enter title for use when do plot.

/USEgel:acc# - use the ACC# gel for Rmaps.

/VFLIPmw - flip the presentation of the MW (vertical axis).

/ZOOM:nX - zoom factor nX (1 to 32) for Rmaps and mosaics.

TABLE 3.4. INQUIRE <CMD> **cgelp2** '**/**'secondary switches typed with the user's response to various INQUIRE commands.

/ABsoluteDiff - use $|m_1 - m_2|/(m_1 + m_2)/2$ instead of m_2/m_1 for CHANGE HIST.

/CLassValue - use class Dens value in LSQ search (instead of StudyValue).

/FILE logging - put output into *.inq* file.

/RElativeDiff - use $(m_1 - m_2)/(m_1 + m_2)/2$ instead of $m2/m1$ for CHANGE HIST.

/STudyValue - use study field value in LSQ search (instead of ClassValue).

/DIrectory - print directory of SRL subsets.

/LISTofSRLs - do Gather-Scatter SRL subset operation.

/MOsaic scripts - generate mosaics scripts.

/RMap scripts - generate Rmap scripts.

TABLE 3.5. SET SRL SUBSET <CMD> **cgelp2** '/'secondary switches typed with the user's response to various SET SRL SUBSET commands.

/DIrectory - print directory of SRL subsets.

/LISTofSRLs - do Gather-Scatter SRL subset operation.

/MOsaic scripts - generate mosaics scripts.

/RMap scripts - generate Rmap scripts.

3.3.7 History (!) of <CMD> commands

Cgelp2 has a *history* interpreter which can be used to keep track of previous commands and to re-execute them if desired. A command history is a journal of commands previously entered and executed. You may list or re-evaluate these old commands to repeat an action. All history commands start with a '!'. Otherwise the command is just passed on through. Commands starting with ';' (i.e comments) are mapped to NULL and ignored. When entering or exiting **cgelp2**, the history is read or written to the file `cgelp2.hrc` in the current working directory. Using the `!+file` or `!-file` history commands, you can read (append) or write other history files. Subsequences of commands may be repeated using the `!p,q` command (where `p,q` is the history entry range numbers). Up to 200 history commands may be saved, but you have the option of viewing (default) the last 20. The history commands are listed in Table 3.6. This list of commands is printed using either the `HISTORY <CMD>` or if you type an illegal history ! type command.

<CMD> *history*

TABLE 3.6. **cgelp2** history commands. History commands are all prefaced with a !.

!? list the current command history (last 20).

!?? list the current command history (last 200).

!! execute the previous command again.

!nnn execute history command *nnn* number (digits) again.

!XXX execute history command matching letters `XXX` again.

!p^oldPattern^newPattern execute history command *p* where: *p* is *!, nnn* or *XXX* after substituting *newPattern* for all occurrences of *oldPattern*.

!* clear the current history list.

!+file append history list from a file *file.hrc*.

!-file write history list into a file *file.hrc*.

!@n1,n2:file redo history numbers in range [*n1:n2*], *:file* is opt.

!_nnn delete *nnn*'th or last history entry.

!/CCC if string CCC then extend last history entry else terminate it.

EXAMPLE of listing the latest command history.

```
201<CMD>: !?
 181:LIMITS
 182:SET DENSITY MODE//RATIO - change the density mode to Ratio
 183:LIMITS
 184:SET LABEL//PSEAUX//// - reorder all spots in the database
 185:INQUIRE//PRINT//33////
 186:INQUIRE//PRINT//22////
 187:REORDER
 188:INQUIRE//PRINT//22////
 189:INQUIRE//PRINT//33////
 190:SET LABEL//PS//// - only operate on PP and SP
 191:INQ/FIND:0000.1,133,0200400057
 192:INQUIRE//PRINT//22////
 193:SET SRL SUBSETS//SPSS/MOSAIC//2//// - create SPSS file & mosaics script
 194:SET SRL SUBSETS//LIST/DIR////
 195:BACKUP
 196:EXTRAPOLATE/QUIET
 197:BACKUP
 198:GELS
 199:SET LABEL//PSEAU
 200:INQUIRE//PRINT//22////
 201:EXIT
201<CMD>:
```

Notice that comments were added at the ends of some of the commands. The command interpreter only checks for matching commands at the beginning of the command and ignores trailing information. So you may add whatever you wish as long as it does not have '/'s which are interpreted as switches. For example, the following are equivalent.

```
201<CMD>: INQuire search - test for missing spots in DB abcpcg.pcg
202<CMD>: INQuire
203<CMD>: INQ
204<CMD>: inq
```

EXAMPLE of re-evaluating the previous command.

```
6<CMD>: !!
 5: LIMITS
Rspot Mapping file [].
Map Rspots to Fspots is OFF.
Ratio-List density mode calibration EXISTS.
Least-Squares density mode calibration EXISTS.
Relative distance limits [0.00,512.00] pixels
     .
     .
     .
7<CMD>:
```

EXAMPLE of redoing a command by number.

```
120<CMD>: !93
 93: INQUIRE//PRINT//278////
Inquire cmds (B,CH,CO,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%).
 HELP to list CMDS?: PRINT
Print Rspot set #?[#,s,$,0 to exit]?: 278

Rspot[ 278] mnXYDA=(271,168,8.84,43)+-(2.83,2.00,13.09,29) CVD=1.48 #G=19
     .
     .
     .
121<CMD>:
```

EXAMPLE of redoing a command by searching backwards for partial command match

```
121<CMD>: !inq
 118: INQUIRE//PRINT//301////
     .
     .
     .
122<CMD>:
```

EXAMPLE of redoing a command by replacing a substring in the command with another string (everywhere the substring would appear).

```
122<CMD>: !120^278^123
 120: INQUIRE//PRINT//123////
     .
     .
     .
123<CMD>: !inq^23^44
 124: INQUIRE//PRINT//144////
     .
     .
     .
125:<CMD>:
```

EXAMPLE of redoing a range of history commands specified by a numeric range. It saves the specified commands in a temporary file `HISTORY.GDO` and then starts it using the `DO` command. If you specify the option *:fileName*, then it just saves the history commands specified by the range in the file indicated. You can then evaluate this later using the `DO` command.

```
125<CMD>: !@120,124                  # Save and eval in HISTORY.GDO
126<CMD>: DO//HISTORY.GDO////
127: DO//HISTORY.GDO////
Script file to use?: HISTORY.GDO
```

```
      . . .
128<CMD>: INQUIRE//PRINT//278////
      . . .
129<CMD>: INQUIRE//PRINT//301////
      . . .
130<CMD>: INQUIRE//PRINT//123////
      . . .
131<CMD>: INQUIRE//PRINT//144////
      . . .
132<CMD>: !@128,131:print4spots.do     # Save it in specific file name
132<CMD>: DO//print4spots.do
133<CMD>: INQUIRE//PRINT//278////
      . . .
134<CMD>: INQUIRE//PRINT//301////
      . . .
135<CMD>: INQUIRE//PRINT//123////
      . . .
136<CMD>: INQUIRE//PRINT//144////
      . . .
137<CMD>:
```

The `!_nnn` deletes *nnn*'th or last history entry if no number is specified.

```
132<CMD>:!_129
132<CMD>:!?
         . . .
 127: DO//HISTORY.GDO////
 128: INQUIRE//PRINT//278////
 130: INQUIRE//PRINT//123////
 131: INQUIRE//PRINT//144////
132<CMD>:
```

You can add additional arguments to or terminate the last history entry. `!/CCC` extends the history entry if some string *CCC* is specified. If `!/` is specified by itself, then the history entry is terminated.

```
   142<CMD>: !/
would cause
    141: SET LABEL//PS////
```

3.3.8 INQUIRE subsystem commands

INQUIRE *subsystem*

The INQUIRE command is one of the more extensive **cgelp2** commands having many more subcommands than the other **cgelp2** commands. These subcommands, listed in Table 3.7, are primarily concerned with searching the PCG DB and secondarily with printing out different views of SRL subsets of spots. It is used to make inquiry of the database. These commands are printed by typing HELP or ? to the request for subcommand when using the INQUIRE command as in:

```
Inquire cmds (B,CH,CO,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%)
```

TABLE 3.7. `INQUIRE`-level **cgelp2** <SUBCMD>'s.[7] [8] [9] [10]

Behrens-Fisher test search for Rspot sets with p-value < prefilter p-value limit assuming unequal variance (2 classes).

CHange-histogram (table) of mean ratios of last SRL for two classes.

COordinate-pairs (table) of Rspots in last SRL for two classes and $\%Err, T_{pIe}, T_{mw}$.

Darkest-spots search for N darkest Rspots.

Expression-profile test for spots with minimum LSQ error between expected and Rspot set's EP is $< T$.

F-test search for Rspot sets with given confidence limit between two to the maximum number of classes.

Index search for entire Rspots meeting statistical prefilter limits.

***Kruskol-Wallis** rank order search for Rspot sets with given significance between all defined classes for minimum 5 gels/class.

LAndmark search for Rspot sets which are landmarks.

LEast-squares-class linear search for Rspots fitting $Dens = m * ClassValue + b$ for correlation coefficient < confidence limit and specified slope (m) range.

Missing class search for Rspots where (only one of two classes present).

OExpression-profile (table) print Ordered Expression Profile table of SRL spots.

ORder-Rspots-table (table) by all class pairs $(>, <, -)$ for spots in SRL.

% search for Rspots where mean ratios of two classes: $MAX(c1/c2, c2/c1) > T\%$.

Print (table) Rspot set *i*, *$* or *** (all search results Rspots), *S* to only list Rspot # in the SRL.

Rank-order search for Rspot sets with given significance between two classes. Also invoked by `WMW` (Wilcoxon-Mann-Whitney) - another name for the same test.

SEarch for entire Rspot sets meeting statistical prefilter limits.

SOrt SRL by probability (t-statistic) for any 2 classes and print table..

T-test test search for Rspot sets with p-value < prefilter p-value limit assuming equal variance (2 classes).

TB search using F-statistic to select T- or BF-test per Rspot set.

TC search using confidence-limits t-test search, 2-sided test (2 classes).

TP perform T-test search of 2 classes < p-value (continuous).

Upper-lower Missing class search for Rspots (only 1 of 2 classes present. The missing gel class has #gels/class $\leq N_{lower}$ and present gel class has #gels/class $\geq N_{lower}$.

---

[7] Appending `/FILE` to any command will cause its output to be sent to an *.inq* file to be specified.

[8] Appending `/LOGDENSITY` to any command will cause density to be recomputed as $\log(1 + density)$ prior to its use in any operation.

[9] Appending `/STUDYVALUE` or default `/CLASSVALUE` determines the independent variable for `LEAST-SQUARES-LINEAR-SEARCH`.

[10] Appending `/RELATIVEDIFF` or `/ABSOLUTEDIFF` uses $0.5(m_i - m_j)/(m_i + m_j)$ or $0.5|m_i - m_j|/(m_i + m_j)$ instead of m_j/m_i for the CHANGE HISTOGRAM operation.

3.3.9 Graphical User Interface Menus

If you are running under X-Windows, then switch to the Graphical User Interface (GUI) *menu-mode*. NOTE: instead of using this command, you should start **cgelp2** in the GUI mode at the start. E.g.,

```
32% cgelp2 -g -d ts3pcg.pcg
```

A set of command buttons appears at the top of your workstation screen. A Dynamic Rmap and zoom windows which tracks the cursor in the Dynamic Rmap. Instead of getting commands from the "<CMD>:" prompt, you may set them from a set of pull-down menus. In addition, you may also type commands as before if the cursor is in the Dynamic Rmap window. Type `Control/H` to popup a list of special mouse buttons and Control-key bindings. When message windows appear on the screen, click on `Done` to make them go away. GELLAB-II is always dumb-terminal stream compatible - even when X-Windows System menu control is used. Command script input streams allow us to run **cgelp2** in batch mode. The following lists the top-level pull-down menus from left to right.

Description of top level GUI menus

FILE operations on PCG database

EDIT operations on PCG DB after it is constructed

PREFERENCES operations to set or review the PREFILTER

NORMALIZE operations for normalizing PCG DB data

CALIBRATE operations for normalizing PCG DB data

STATUS operations for info on PCG DB state

RSPOT SUBSETS operations on subsets of Rspots

GEL SUBSETS operations on subsets of gels

QUERY CGL DB search operations

TABLE operations for correlating SRLs or Gels

PLOTS operations for derived plots

MAP PLOTS operations for Rmaps and Mosaics, etc.

SWITCHES for modifying global <CMD>s

ANNOTATE operations on the dynamic Rmap

HELP operations for info on CGELP2 commands

executing a CMD

Selecting an entry from one of the pull-down menus will execute that command as a <CMD> - just as if you had typed it. If the entry requires additional arguments (e.g the history command '`!::?n`' requires an argument '`n`', it will prompt

you for the additional '`n`' argument. In general, the format for an entry is *command:optionalArg - comment...'*. A single '`:`' preserves the '`:`' in the command while a double '`::`' removes itself after getting the optional argument and appending it to the command. You may also type a command name (e.g `LIMITS` for the above example) instead selecting the command with the mouse from the pull down menu.

List of CGELP2 Sub Menus

pull-down menus

The list of other major pull-down menus are given here for completeness. Note that selections from the submenus may have sub-submenus themselves (e.g `SET SRL SUBSETS` - although they are not listed here.

```
MAIN MENU 1:
------------------------------------------------------------------------
|        FILE operations on PCG database                               |
========================================================================
CReate PCG DB - create paged CGL database from CMPGEL .gcf files.
/ERspot - with CREATE, it will include non-Rgel US as Rgel EP spots.
------------------------------------------------------------------------
CLOSE the PCG DB - close the previously opened PCG database.
COALESCE - the PCG DB to optimize Rspot set access.
DELete pcg db - delete all file for the specified PCG database.
OPen PCG DB - specify and open a new PCG database.
PRotect PCG DB - paged CGL database for read-only (toggle).
SET DAtabase file name - declare or query name of Paged Composite Gel DB file.
------------------------------------------------------------------------
SET ACcession file name - change the default accession file.
------------------------------------------------------------------------
*CGEL' - create C-gel' est. cononical gel (.gsf) file from replicate gels.
Dump CGL - dump prefiltered CGL database to an ASCII (.cgl) file.
SPss - dump prefiltered SPSS .sps summary file of part of the CGL database.
SET SPot view - define 'view' of a spot's features to dump on SAS/SPSS/etc.
------------------------------------------------------------------------
BACkup PCG DB - checkpoint working CGL DB onto the current paged DB file.
ABORT session - and do NOT save PCG database for later use, just exit.
EXIt session - save PCG database file for later use and exit CGELP2.
------------------------------------------------------------------------
```

```
MAIN MENU 2:
------------------------------------------------------------------------------
|         EDIT operations on PCG DB after it is constructed                  |
==============================================================================
EDit Rspots - spots in Rspot sets in the CGL database.
EXTrapolate missing spots - in Rsets from mean (dx,dy)+LM positions.
*MERgeAP spots - merge AP's with SP or PP if meet DP limits.
*REMove gel - from the Paged Composite Gel DataBase.
REOrder PCG DB - Rspot sets (use after change density mode to sort PCG DB).
VErify PCG DB - test if PCG DB corrupted by verifyin Rspot checksums.
------------------------------------------------------------------------------
```

```
MAIN MENU 3:
------------------------------------------------------------------------
|        PREFERENCES operations to set or review the PREFILTER          |
========================================================================
SET CLasses - define gel experimental gel class partition.
SET CLasses::Auto - put gels into classes by current study (see SET FIELDS).
SET CLasses::No - don't change the gel classes.
SET CLasses::Subsets - change the gel classes by gel subsets .
SET CLasses::Yes - change classes by prompting which class to put each gel.
------------------------------------------------------------------------
SET DEnsity mode - Uncorrected(D'), Abs(D'), %, Ratio, Volume, LSQ, CPM units
SET FIelds - set current studies (from access. file) desired for gel labeling.
SET LAbel limits - set Pairing-Label code from (S,P,A,U,E,C,X) for prefilter.
SET PRefilter limits - set most PREFILTER limits from a dialog form.
/EPspot - use EP (0 density) spots in distribribution calculations.
/EXplain - why Rspot set failed the prefilter in the current operation.
/LOGDensity - use log of density instead of density in all calculations.
/Median - use median instead of mean calculation in Rspot statistics.
------------------------------------------------------------------------
SET RGel - set the name of the Rgel used in searching.
------------------------------------------------------------------------
SET Working gels - change working set of gels from CGL database.
SET Working gels::Add - specific gels to current working set of gels.
SET Working gels::Define - specify list of gels as new working set of gels.
SET Working gels::Subtract - remove gels from current working set of gels.
SET Working gels::Yes (Edit) - prompt for gels to include in working set.
------------------------------------------------------------------------
SET PArameter subset - define/operate on subsets of (PREFILTER) parameters.
SET PArameter subset::CLEAR subsets - clear all parameter subsets.
SET PArameter subset::DELETE subset - parameter subset to be specified.
SET PArameter subset::DIrectory - list directory of parameters subsets.
SET PArameter subset::List - list parameter subsets to be specified.
SET PArameter subset::Restore - current params. from subset to be specified.
SET PArameter subset::Save - current parameters into subset.
------------------------------------------------------------------------
```

```
MAIN MENU 4:
------------------------------------------------------------------------
|        NORMALIZE operations for normalizing PCG DB data               |
========================================================================
SET LEast squares calibration - density normalization calibration to Rgel.
SET LEast squares::No (i.e status) - use the old LSQ normalization.
SET LEast squares::Yes-Recalibrate - do new Least Squares normalization
------------------------------------------------------------------------
SET RAtio calibration - compute Ratio-Mode Rspot mean density normalization.
SET RAtio cal::Input list of Rspots - to define Rspots to calc. mean Ratio calibs.
SET RAtio cal::$ (i.e. SRL) - to indicate use SRL Rspot sets to calc. Ratio calibs.
SET RAtio cal::Edit - previous ratio-spot-list, then compute Ratio calibs.
SET RAtio cal::Recompute - Ratio calibrations using previous ratio-spot-list.
------------------------------------------------------------------------
```

```
MAIN MENU 5:
------------------------------------------------------------------------
|        CALIBRATE operations for normalizing PCG DB data               |
========================================================================
SET CAlibration - calibrating spot position (pIe,MW) as function of (x,y).
SET CAlibration::DEFINE pIe Rspots - as SRL Rspots and their pIe-X values.
SET CAlibration::DEFINE MW Rspots - as SRL Rspots and their MW-Y values.
SET CAlibration::EDIT Rspots' pIe&MW - values of previously DEFINED markers.
SET CAlibration::READ pIe-MW - read .cal calibration file and recalibrate.
SET CAlibration::WritepIe-MW - save current calibr. as .cal calibration file.
SET CAlibration::List pIe-MW - list current calibration points.
SET CAlibration::Format .cal file - print legal format of a .cal calibration file.
SET CAlibration::CAlibrate pIe-MW - calibrate (pIe,MW) from defined (x,y) or Rspots.
SET CAlibration::CLEAR calibration - clear old calibration so can use DEFINE again.
SET CAlibration::Unit-area(density) - set mode report spot integr.dens/unit-area.
SET CAlibration::Total-density - set mode to report spot total integrated density.
/AReaInMM - print area in square millimeters and density/mm**2.
/CAlibrateMWpIe - print(x,y) as (pIe,MW) if calibrated.
------------------------------------------------------------------------
SET FOreign spot map - define Rspot<==>Fspot protein name mapping.
SET FOreign spot::CLEAR map - clear the ENTIRE Foreign spot map properties.
SET FOreign spot::DEFine spot map - entry (Rspot#, Fspot#, protein name).
SET FOreign spot::DELETE spot map - entry (Rspot#, Fspot#, protein name).
SET FOreign spot::EDITSpot - entry (Rspot#,Fspot#,{annotation #s},protein name).
SET FOreign spot::ENTERSpot - into SRL & dyn. Rmap, but don't change Fspot map.
SET FOreign spot::FINDProtein spots - find Rspots by protein name pattern.
SET FOreign spot::LISTMap - list all (Rspot#,Fspot#,{annotation #s},protein name)s.
SET FOreign spot::LISTSpot - list a (Rspot#,Fspot#,{annotation #s},protein name).
SET FOreign spot::Mode - toggle between Rspot and Fspot mode.
SET FOreign spot::READ map file - get or merge foreign spot map (.map) file.
SET FOreign spot::WRITE map file - save current foreign spot map in (.map) file.
/DRawSpotName - draw the spot name when drawing the Fspot in the Dynamic Rmap.
/FSpot - use foreign spot instead of Rspot number.
/RSpotFromGui - get Rspot# prompt by clicking in Dyn. Rmap else from keyboard.
------------------------------------------------------------------------
```

```
MAIN MENU 6:
-------------------------------------------------------------------------
|        STATUS operations for info on PCG DB state                     |
=========================================================================
Gels - lists names, total densities and study of current working gels.
Features - list current maximum values for various spot features.
Limits - print the current statistical limits.
TImer - print the run and cpu times for commands (toggle).
-------------------------------------------------------------------------
VAlid landmarks - list valid landmarks table for each gel in PCG DB.
/EDifferencePrint - w/CREATE print Euclidian Difference in LMS data
-------------------------------------------------------------------------
DBUG:Onnnnn - enter debugging option bits (if debugging enabled).
/FIND:?ACC#,?CC#,?DBUGBITS - print Rspot# if ever find this spot&set DBUG.
-------------------------------------------------------------------------
DO:?scriptFile - execute commands from a script file
SYstem:?cmd - Evaluate command on the underlying operating system.
-------------------------------------------------------------------------
! print the command history HELP message.
!? print command history list.
!?? print full command history list.
!! redo the last command history entry.
!::?n re-evaluate command history entry n.
-------------------------------------------------------------------------
```

```
MAIN MENU 7:
-------------------------------------------------------------------------
|        RSPOT SUBSETS operations on subsets of Rspots                  |
=========================================================================
SET SRl subset operations - define/operate on Search Results List subsets.
SET SRl subsets::Assign - current SRL to new SRL subset.
SET SRl subsets::CLEAR - all SRL subset(s), (it will ask you to verify this).
SET SRl subsets::Explicit - define new SRL subset from explicit list of Rspots.
SET SRl subsets::DElete - [/ListOfSRLs] specified SRL subset(s).
SET SRl subsets::DIrectory - list subset titles & # Rspots in each subset.
SET SRl subsets::FIndkeywd - list SRL subsets which have keyword in title.
SET SRl subsets::Intersection - [/ListOfSRLs] define new SRL subset as intersect. of 2 subsets.
SET SRl subsets::List - [/ListOfSRLs] list subset titles & spots in subset(s).
SET SRl subsets::QueryRspot - list SRL subsets which contain spot to be specified.
SET SRl subsets::REAd - read (and create) SRL subset(s) from .srl file.
SET SRl subsets::RENUMBER - all SRL's, removing null subset(s).
SET SRl subsets::REStore - the SRL from SRL subset to be specified
SET SRl subsets::SPss - [/Mosaic/Rmap/StartBatch] create SPSS/SAS data file & opt. Mosaic/Rmap scripts.
SET SRl subsets::SUbtract - [/ListOfSRLs] define new SRL subset as diff. of 2 SRL subsets.
SET SRl subsets::Union - [/ListOfSRLs] define new SRL subset as union of 2 SRL subsets.
SET SRl subsets::Write - [Filename] write SRL subset(s) into 'filename'.srl file.
/LISTofSRLs - do Gather-Scatter SRL subset operation.
/MOsaic scripts - generate mosaics scripts when do SPSS subcommand.
```

```
 /RMap scripts - generate Rmap scripts when do SPSS subcommand.
 /STartBatchJob - if SPSS subcommand, start batch job(s) from /MOSAIC and/or /RMAP.
-------------------------------------------------------------------------
 SEQuential SRL subset operations - of groups of adjacent SRL subsets.
 SEQ SRL subset::All - given w do sequential <set-opers.> on (SSRL[1] to SSRL[w]).
 SEQ SRL subset::List - given i,j,...,m do <set-operation> on list of sets to process.
 SEQ SRL subset::Operator - define or change current <set-operation>.
 SEQ SRL subset::Sequential - given i,w do <set-operation> on SSRL[i] to SSRL[i+w-1].
 -------------------------------------------------------------------------


 MAIN MENU 8:
 -------------------------------------------------------------------------
 |        GEL SUBSETS operations on subsets of gels                      |
 =========================================================================
 SET Gel subset operations - define/operate on subsets of gels.
 SET Gel subset::Classname - define new gel subset of working set gels by class name.
 SET Gel subset::DElete - gel subset.
 SET Gel subset::DIrectory - list directory of gel subsets.
 SET Gel subset::Explicit - define new gel subset by explicit list of gels.
 SET Gel subset::Intersection - define new gel subset as intersection of 2 gel subsets.
 SET Gel subset::List - contents of gel subset.
 SET Gel subset::REMOVE - ALL gel subsets (it asks you to verify this).
 SET Gel subset::Subtract - define new gel subset as difference of 2 gel subsets.
 SET Gel subset::Union - define new gel subset as union of 2 gel subsets.
 SET Gel subset::Workingset - define new gel subset from Working Set of gels.
 -------------------------------------------------------------------------
```

```
 MAIN MENU 9:
 ------------------------------------------------------------------------
 |        QUERY CGL DB search operations                                |
 ========================================================================
 INQuire operations - interrogate and search CGL DB for spot differences.
 INQuire::BF-T-test - perform Behrens-Fisher T-test search.
 INQuire::COordinate-pairs-test - search for C-P within SRL.
 INQuire::Expression-profile-test - search for spots with min LSQerr<T.
 INQuire::F-test - search for Rspots with p-value limit for difference between N classes.
 INQuire::FP-test - search for Rspots w/p-value (continuous) limit for difference between N classes.
 INQuire::Help - print INQUIRE HELP message.
 INQuire::Index - perform search for Rspots meeting prefilter cond.
*INQuire::KW-test - perform Kruska-Wallis 2 to 9 class search.
 INQuire::Landmarks - search for Rspots which are landmarks.
 INQuire::Least-Squares-test - perform LSQ search of 2 classes.
 INQuire::Missing-class - perform search of 2 classes.
 INQuire::Rank-test - perform Wilcoxon-Mann-Whitney 2 class search.
 INQuire::Search - perform search for Rspots meeting prefilter cond.
 INQuire::T-test - perform T-test search of 2 classes (1-sided).
 INQuire::TB-test - perform f-statistic selected BF-t or T-test.
 INQuire::TC-test - perform confidence-limits T-test search (2-sided).
 INQuire::TP-test - perform T-test search of 2 classes > p-value (continuus).
*INQuire::T2-test - perform T-test search of 2 classes (2-sided).
 INQuire::Upper-lower-Missing-class - perform search of 2 classes with range.
 INQuire::%-Test - percent variation search of 2 classes.
 ------------------------------------------------------------------------
 /ABsoluteDiff - use |m1-m2|/(m1+m2)/2 instead of m2/m1 for CHANGE HIST.
 /CHangeHistogram - print it after INQUIRE 2-class search.
 /CLassValue - use class Dens value in LSQ search (instead of StudyValue).
 /FILE logging - put output of a test into a .inq logging file.
 /MUstBeIn1stClass - gels must be in 1st class for missing-spot test.
 /RElativeDiff - use (m1-m2)/(m1+m2)/2 instead of m2/m1 for CHANGE HIST.
 /STudyValue - use study field value in LSQ search (instead of ClassValue).
 ------------------------------------------------------------------------
```

```
MAIN MENU 10:
------------------------------------------------------------------------------
|        TABLE operations for correlating SRLs or Gels                       |
==============================================================================
TAbulate operations - compute Rank-order, gel-gel-Cor, Rspot-Rspot-Cor table.
TAbulate::GEl-gel-Correlation - compute and print table.
TAbulate::RAnk-order - compute and print table/plot of Rspots.
TAbulate::SRl-Srl-Correlation - compute and print table.
------------------------------------------------------------------------------
INQuire::CHangeHistogram - computed and print Mj/Mi for SRL.
INQuire::OExpressionProfile - print Expr. Profile table of SRL spots.
INQuire::ORderTable - Mj/Mi by all class pairs (>,<,-) for SRL spots.
------------------------------------------------------------------------------
INQuire::Print a Rspot - print table of a Rspot set(s) data.
INQuire::Print::s (SRL spot #s) - print NAMES of spots in Rspot set .
INQuire::Print::$ (all SRL Rspots) - print table of all Rspots in the SRL.
/HeaderOnly - print only Rspot set summary header - not tabular data.
------------------------------------------------------------------------------


MAIN MENU 11:
------------------------------------------------------------------------
|        PLOTS operations for derived plots, etc.                      |
========================================================================
CCplot (class vs class) scatter - plot LOG density/density class-class plots from CGL DB.
DCplot (density vs class) scatter - plot LOG density/classValue for Rspot.
DDplot (density vs density) scatter - plot LOG %density/%density CGL DB spots.
/BOxLabel - draw additional boxes as appropriate.
/CCoefLabel - label correlation corefficient as is appropriate.
/CLAssNames - draw class names in plot otherwise use class numbers.
/LINElabel - draw 40 degree line or additional lines as appropriate.
/LOGPlot - do log-plot instead of linear plot.
------------------------------------------------------------------------
DEndrogram clustering - plot of gels=fct(SRL) or SRL=fct(expr-profiles).
/CLUSTERGels - cluster gels as fct of SRL Rspot density features.
/CLUSTERRspots - cluster Rspots as fct of mean class densities features.
/DENCOLOR - label objects in plots different colors.
/DENDTABLE - list all the partial data used in computing clusters.
------------------------------------------------------------------------
EXPression profile - plot Expression Profile for Rspots in SRL.
HISTOGram of spot data - compute/plot Rspot sets feature histograms.
/File - print the histogram as 'bin#[frequency] ***...' to log file.
/MW - project MW onto Y-axis to 1D gel in plot.
/PIe - project PIE onto X-axis to 1D gel in plot.
PLot 2D or 3D features - plot Rspot CGL DB 2D/3D feature scatter plot.
/ANgle:?nDegrees - degrees to rotate 3rd dimension (default 30 degrees).
------------------------------------------------------------------------
SET DIsplay for plots - plotting device to XWND, 4012, PPX, LASER, xxxxPLOT.
/CLAssNames - draw class names in plot otherwise use class numbers.
/GelAcc#labels - use gels or label gel's ACC# as appropriate.
/SCale:?dS,?cS - scale density by (dS,cS) factor else by each Rspot maxima.
/SRlLabel - label SRL Rspots otherwise don't label spots.
/UGf label - label plots with .ugf file number.
```

```
MAIN MENU 12:
------------------------------------------------------------------------
|        MAP PLOTS operations for Rmaps and Mosaics, etc.               |
========================================================================
BUbblePlot - of probability from t-statistic that means of c1,c2 are different.
MOsaic of Rspot in gels - generate a mosaic plot/image around the specified Rspot.
RMap of SRL in a gel - generate a Rmap plot/image surrounding optional Rspot.
/ALlgels - plot all Mosaic(SRL spots) or Rmaps (gels).
/CEnter:?x,?y - center Rmap at (x,y) instead of on Rspot if no spot specified.
/COlor - draw spot as red:S,P blue:A,E green:U,C,G else black.
/DUmpPPXplot - dump the PPXplot image on the laser printer.
/FIllSpot - fill in spots as appropriate.
/HFlipPIE - flip gel plot about the pIe (horizontal) axis else don't flip.
/LAbelSpots - label additional information on all plotted spots.
/MAkeup:?nXn - use nXn instead of Working Set size for mosaic plots.
/PLusLabel - label spots with '+' (i.e. /NOPLUS for dots).
/PPxplot - generate PPX image on Xpix display instead of grapics plot.
/SIzebyD' - spot size is proportional to spot D' else its (sX,sY).
/VFlipMW - flip gel graphic about the MW (vertical) axis else don't flip.
/USe:?gelAcc# - use specified gel for Rmap otherwise use the Rgel.
/Zoom:?nX - zoom gel graphic by nX for Rmaps and mosaics.
------------------------------------------------------------------------
```

```
MAIN MENU 13:
------------------------------------------------------------------------
|        SWITCHES for modifying <CMD>s                                 |
========================================================================
/FUll - print full specification for the operation.
/Option:?stringArg - extra argument used by some operations.
/PRecision7digits - use 7 digits of precision in printing means,std-devs.
/Quiet - do not output status during calculations if applicable.
/SAsFormat - generate SAS instead of SPSS output if appropriate.
/SOrt - SPSS or SAS output file by gel number rather than density.
/SIngle - gel for Cgel' estimate if appropriate.
/SRLSS:?n - filter operation by SRL[n] subset if appropriate.
/WorryMsg - print DON'T WORRY message every 30 seconds.
------------------------------------------------------------------------



MAIN MENU 14:
------------------------------------------------------------------------
|        ANNOTATE operations on the dynamic Rmap                       |
========================================================================
DRMAP::DElete Text - in the Dynamic Rmap (drawn with 'Draw Text').
DRMAP::DRaw Text - in the Dynamic Rmap.
DRMAP::Move Text - in the Dynamic Rmap (drawn with 'Draw Text').
------------------------------------------------------------------------
SET ANnotation - spot feature annotation.
SET ANnotation::ANNotation - edit property names of Spot map properties.
SET ANnotation::ADD annotation - feature to all SRL spots (set Fspot=Rspot if not defined).
SET ANnotation::CLEAR annotation - clear all annotation properties.
SET ANnotation::DELETE annotation  - feature [1:32] or by name.
SET ANnotation::EDIT spot - annotation list of features for a specific Rspot/Fspot.
SET ANnotation::FINDAnnotation spots - find Rspots by annotation FEATURE # [1:32] or name.
SET ANnotation::FINDExpression spots - find Rspots by annotation feature Boolean expression
SET ANnotation::SUBtract annotation - feature from all spots in SRL if feature was defined.
SET ANnotation::TABle annotation - print Rspot features in a table for annotated spots.
------------------------------------------------------------------------



MAIN MENU 15:
------------------------------------------------------------------------
|        HELP operations for info on CGELP2 commands                   |
========================================================================
BIndings (key & mouse) - list keyboard and mouse-key bindings available.
INFormation - print the CGELP2 general information message.
HElp on <CMDS>s - print general help message or help for a specific command/subcommand.
HISTORy Help - list the CGELP2 history commands.
INQuire::Help:::: - print list of INQUIRE search and table subcommands.
------------------------------------------------------------------------
```

```
 SUBMENU (SET GEL SUBSET) 2:
-------------------------------------------------------------------------
|        GEL SUBSET MENU                                                |
-------------------------------------------------------------------------
| Classname  - define new gel subset as W.S. gels with class name.      |
| DElete     - gel subset.                                              |
| Explicit   - define new gel subset by explicit list of gels.          |
| Intersection - define new gel subset as intersection of 2 gel subsets. |
| List       - directory of gel subsets & particular subsets contents.  |
| REMOVE     - ALL gel subsets.                                         |
| Subtract   - define new gel subset as difference of 2 gel subsets.    |
| Union      - define new gel subset as union of 2 gel subsets.         |
| Workingset - define new gel subset as the Working set of gels.        |
| <CMD>      - Return to top command level.                             |
-------------------------------------------------------------------------
```

```
SUBMENU (SET SRL SUBSET) 3:
 ---------------------------------------------------------------------------
 |       SRL (Search Results LIST) SUBSET MENU                             |
 ---------------------------------------------------------------------------
 | Assign - current SRL to new SRL subset.                                 |
 | CLEAR - all SRL subset(s).                                              |
 | Explicit - define new SRL subset from explicit list of Rspots.          |
 | DElete - [/ListOfSRLs] specified SRL subset(s).                         |
 | FIndkeywd - list SRL subsets which have keyword in title.               |
 | Intersection - [/ListOfSRLs] def. new SRL subset as intersect. 2 subsets. |
 | List - [/Dir/ListOfSRLs] list subset titles (names if /DIR) & subset(s). |
 | QueryRspot - list SRL subsets which contain spot to be specified.      |
 | REAd - read (and create) SRL subset(s) from .srl file.                  |
 | RENUMBER - all SRL's, removing null subset(s).                          |
 | REStore - the SRL from SRL subset to be specified                       |
 | SPss - [/Mosaic/Rmap] create SPSS/SAS data file & opt. Mosaic/Rmap scripts. |
 | SUbtract - [/ListOfSRLs] define new SRL subset as diff. of 2 SRL subsets. |
 | Union - [/ListOfSRLs] define new SRL subset as union of 2 SRL subsets.  |
 | Write - [Filename] write SRL subset(s) into 'filename'.srl file.        |
 | <CMD> - Return to top command level                                     |
 ---------------------------------------------------------------------------
```

3.3.10 Detailed cgelp2 command descriptions

Detailed descriptions of the **cgelp2** commands are given in this section by both explanation and by example. In this Section, command subheadings are prefixed with '<*CMD*> `command-name`' and commands are listed alphabetically. Some commands have a number of subcommand options. These subcommands are prefixed in this section as '<*SUBCMD*> `command-name`'.

the gory details...

<CMD> ABORT
Abort **cgelp2** to UNIX to *not* saving the current state of the PCG database file. Use command `EXIT` if you want to checkpoint the database when you exit. Because using `ABORT` is so dangerous to losing your data, it will ask you to confirm. You must type the word `yes` to actually abort the database session.

```
44<CMD>: ABORT
Are you sure you REALLY want to exit CGELP2 without saving
the PCG DB(yes/no)?:  [no]
?: yes
47%
```

<CMD> BACkup
Backup the working CGL database state in the current paged database file previously defined with `SET DATABASE`. Control returns to **cgelp2** so that the interactive session may continue. The `EXIT` command may also be used to save the CGL database. However, it does not return control to **cgelp2** but rather exits to the UNIX shell. For example:

```
25<CMD>: BACKUP
Checkpointing PCG database
      /home/joeUser/gellab/pcg/hempcg.pcg
for later use.

Using existing PCG paged composite gel database:
      /home/joeUser/gellab/pcg/hempcg.pcg
   Date created: 12/01/1988, 10:44:51
   Date last session: 12/07/1988, 12:04:19
```

Note that if the file is write protected or the `PROTECT` command was invoked, then you will not be allowed to do a `BACKUP`.

<CMD> BIndings
When working with the Graphical User Interface enabled, you may want to find out what the key and mouse bindings are. This command lists then but only when the GUI is running.

GUI only

<CMD> *BUbble plot
This generates an Rmap plot where the size of each Rspot circle is proportional to the t-statistic probability value for the difference of two classes.

<CMD> C-GEL′

canonical gel′

It is possible to *estimate* a C-gel′ ([LemP82a], [LesE82a]) from the PCG DB under particular conditions. This estimate is called the C-gel′. The existing PCG database must consist of a set of replicate gels. By replicate we mean: gels of the same sample or of parallel tissue cultures run under the same conditions and at the same time. Having created such a PCG DB, sizing (see `SET STATISTICS` and `INQUIRE -` search) is used to discriminate robust spots from noise. For example, requiring 80% of the gels to be present and for the coefficient of variation to be less than some small value would find robust spots in a majority of the gels. The C-gel′ has the same accession number as the Rgel in this replicate gel set *but* the accession number extension is defined to be `'9'` (e.g. Rgel `250.2` has a C-gel′ of `250.9`).

A *synthetic* GSF file is produced which uses the mean Rspot set centroid (x,y) mapped to the Rgel as well as mean Rspot set D′ and area. The standard deviations of these parameters and #gels/Rspot set are also output for future use in building a new DB based on Cgel′. Spots are renumbered sequentially from 1 where only spots meeting sizing criteria are output.

The `C-GEL′` command first may request the name of the landmark database file (if *XXX* in the accession file *gelXXX.id* is different from *YYY* in the landmark file *lmsYYY.lm*). (The names of these two files may be obtained by running PGEL). The landmark database file is then searched for Rgel landmark set entries. A duplicate of each entry is inserted where the s name is *changed* to that of the Cgel′ and the mean centroids of the Rspot sets of the corresponding Rgel landmark spots are substituted. The typical use of the Cgel′ command is as follows:

1. Accession a set of gels where there are replicates to experimental gels.
2. Landmark these gels to the replicate gel (or define one as Rgel). Segment and pair all gels as is usually done.
3. Using **cgelp2** build a PCG DB of *only* replicate gels from each set of replicate gels. Of course you could build a PCG DB which includes these gels as well as others, but then `SET WORKING GELS` to just this subset of gels.
4. Create the Cgel′ *.gsf* and LM′ database entry (as detailed above) using the `C-GEL′` command - then `EXIT` **cgelp2**. This should be done for each set of replicate gels.
5. Run **cmpgl2** on the *other* experimental gels combined with Cgel′s (replicate gels may also be included if desired).

6. Build a new PCG DB around these GCF files and these Cgel′s - but not the gels used to make the Cgel′.

Note: For the Cgel′, all Rspots in any of the gels used to make the Cgel′ are in the Cgel′. Similarly, all Cgel′ eRspots are US spots in the experimental gels (or were spots originally sized out of the set of replicate gels when creating the Cgel′).

Example of building a Cgel′ from eight replicate gels

Cgelp2 prints the name of the .gsf file and each LM database entry which was synthesized with the Cgel′ (c.f. **sg2gii** discussion in Section 3.18 for synthetic Cgel′ *.gsf* format). The default GSF file path is specified by the `gel.rc` state file keyword `ppnp2x` associated entry. The default landmark DB file path is specified by the `gel.rc` state file keyword `lmsFile` associated entry. After setting the prefilter and make invoking the `Cgel′` command,

```
<27>CMD: C-GEL'
Output file:(p90250.gsf)?: <CR>
Total of 313 accepted D' spots accumulated density= 4372., area=23596
Generated estimated C-gel GSF file: p90250.gsf
LMS[0250.9,0250.1]
LMS[0250.9,0250.2]
LMS[0250.9,0251.1]
LMS[0250.9,0251.2]
     .
     .
     .
```

The following is part of the Cgel′ synthetic gel segmentation file.

```
SG2GII : Version Sept 3, 1981 - 5:12AM
Today's date is 09/24/1981,  08:26:13 AM
User:/home/joeUser/gellab/aux
Gel Segmentation File is: p90250.gsf
0250.9/P388D1/-/-/8-21-80/#A95/FISCHER'S/3:10, 10%/
0 HRS/C14/8/1 WEEK/ALUMINUM,TO,CONTROL,BOTTLE#1/
L00157/-NONE-/--NONE--/VIDICON-AUTO,28MM F8,69CM/LIPKIN*
 40 65 88 109 126 144 159 170 181 190 196 208 213 0 0 0 31 460 51 400
Switches:  -SHORTGSF
Window [0:511,0:511]
Spot Area sizing limits ( .00, 500.00)
Integrated Density sizing limits ( .00, 1000.00)
Density difference sizing limits ( .00, 3.16)
Background range [ .00: .00] OD
 CC#1[205:210, 239:242][ .30: .54] .122, .000
  [ 207.43, 240.29]29, 3.50, 3.50, .09
   2.10, 1.30, .00, 4.65 1.76, 6.58, 1.59, 1.28, 8
 CC#2[213:215, 241:245][ .32: .57] .138, .000
  [ 214.29, 242.57]27, 3.77, 3.77, .10
   1.10, 2.00, .00, 3.90 2.39, 12.27, 1.28, .90, 7
```

```
 CC#3[201:207, 244:248][ .25: .73] .164, .000
  [ 203.75, 245.75]60, 9.80, 9.80, .25
   3.00, 1.90, .00, 19.40 3.32, 8.00, 1.30, 1.48, 8
      .
      .
      .
 CC#313[76:80, 234:238][ .13: .39] .064, .000
  [ 77.71, 236.43]45, 2.84, 2.84, .07
   2.20, 2.00, .00, 8.11 1.15, 7.87, 2.60, .90, 8
Total of 1878 accepted D spots accumulated density= 4372., area=23596
Total of 313 accepted D' spots accumulated density= 4372., area=23596
Total of 0 omitted spots accumulated density= 0.0, area=0
Omitted/Accepted density =0%
Done at 09/24/1981,  08:28:32 AM
```

The following is an example of part of the updated landmark DB with the Cgel′ entry. The first entry is the set of landmarks of gel 250.1 paired with the Rgel 250.2 and the second entry is that of 250.1 paired with the Cgel′ 250.9.

```
/ CMPGL2: VER# 9/18/81 - 9:09AM
/ INTO SYS@:JUNK@@.DA FROM GSF FILES: P20250.GS AND P10250.GS
/ SURE!PAIR THRESHOLD= 205, POSSIBLE!PAIR THRESHOLD= 336
11/15/1980,  07:45:23 PM
 LANDMARK #A  G1[211, 262], G2[204, 284]
 LANDMARK #B  G1[173, 235], G2[168, 259]
 LANDMARK #C  G1[180, 219], G2[175, 243]
      .
      .
      .
/ CMPGL2: VER# 9/18/81 - 9:09AM
/ INTO SYS@:JUNK@@.DA FROM GSF FILES: P90250.GS AND P10250.GS
/ SURE!PAIR THRESHOLD= 5, POSSIBLE!PAIR THRESHOLD= 10
09/21/1981,  04:01:48 PM
 LANDMARK #A  G1[208, 263], G2[204, 284]
 LANDMARK #B  G1[173, 236], G2[168, 259]
 LANDMARK #C  G1[177, 219], G2[175, 243]
      .
      .
      .
```

<CMD> CCplot

class-class scatter-plots

Draw (optionally generating a plot *.ugf* file) a class vs. class density plot of specified (default all) classes of gels. The plot is drawn as a linear (default /NOLOG switch) plot. `CCPLOT` prompts the user to enter the names of the current classes to be plotted. Optionally, it defaults plots of `ALL` of the classes. Use `SET DISPLAY` to define the plot or display type to be used (as well as associated plot switches: `/CLASS`, `/FILL`, `/LABEL`, `/LINE`, `/LOG`, `/SRLlabel`, `/UGFlabel`). Figure 3.1 shows a sample plot output. A response to a typical prompt follows:

```
244<CMD>: ccplot
CLASS vs. CLASS scatter plots for gel classes (Options)[all]
 ?: ?
 DEFAULT SWITCH OPTIONS
 ----------------------
  /NOCLASSNAMES - default is label 'CLASS# c' else use class names.
  /FILL - default is label spots with '+' (i.e. /NOFILL for dots).
  /NOLABEL - default is not to display corr. coefs and # spot pairs.
  /NOLINE - default is not to draw 45 degree line.
  /NOLOG - default is do linear-linear instead of log-log scatter plot.
  /NOSRL - default is to label no spots (i.e. /SRL to label Rspots).
  /UGF - label plot with UGF file number.
[04:24:12PM] Real TIME =00:00:04  CPU TIME =00:00:00,  0.00%

245<CMD>: CCLASS
CLASS vs. CLASS scatter plots for gel classes (Options)[all]
 ?: /class/label
CLASS[1] = WC-A
CLASS[2] = WC-B
CLASS[3] = WC-C
CLASS[4] = WC-D
Printing plot file on laser printer
 plotn -display:4010 /home/joeUser/gellab/gen/000051.ugf | tek2psG | \
                                                          lpr -Plaser
Plot file: /home/JoeUser/gellab/gen/000051.ugf
CLASSES[2,1] r=0.775, 441 pairs.
CLASSES[3,1] r=0.788, 468 pairs.
CLASSES[3,2] r=0.710, 448 pairs.
CLASSES[4,1] r=0.119, 244 pairs.
CLASSES[4,2] r=0.161, 235 pairs.
CLASSES[4,3] r=-0.062, 234 pairs.
[04:26:38PM] Real TIME =00:02:20  CPU TIME =00:01:22,  58.57%
```

Figure 3.1. Sample **cgelp2** `CCPLOT` plot. The plot file was ploted using the `LASER` display option set with `SET DISPLAY`.

<CMD> CLOSE DATABASE FILE
If a PCG DB was opened, you can close it in order to open a different one. It Checkpoints the database before closing it.

. **<CMD> CReate**
Create or extend an existing PCG database from a set of **cmpgl2** *.gcf* files (see page 3.4). No sizing is done on SP or PP spots. During database creation, any Rgel spots which have a AP label in some of the gels used in the initial part of the database construction are given the tentative AP label. If in finding the Rgel defined as a SP or PP later on, the system redefines the Rgel spot label from AP to SP or PP. If there is a higher ratio of AP to (SP or PP) than 2.5 to 1, you can speed up the process of constucting the PCG DB by specifying the *nodeSaftyFactor* value (greater than 2.5) using the `/OPTION:`*nodeSaftyFactor*.

Hint: to add a set of gels to an existing PCG DB (say it is called `tc3pcg.pcg`), first copy it to a new file to which we will be adding the new gels (call it `tc4pcg.pcg`). Then run **cgelp2** on the latter PCG DB file and use `CREATE` to add the new gels. Since this is effectively a new database it should have a new name to keep it from being confused with the old database. In case of corruption of the new database one can always then fall back on the older database. Note that when adding new gels to an existing PCG DB, it is impossible that a new un-extended Rspot set could be added which have a Rspot number name greated than existing eRspots numbers. If if were, then by definition it would be an un-extended Rspot for the Rgel.

building a PCG DB

If `/ERSPOT` is appended to the `CREATE/ERSPOT` command, then the set of *.gcf* files is scanned a second time for US spots *not* in the Rgel which have (area, D' (density corrected for background in absolute OD units), and OD difference *ODdf* (i.e. (*maxOD−mnBackground*)) within the limits set by `SETSTATISTICS`. When these spots are read from the files, they are first tested to see if they belong to an existing eRspot set (within DP limits of the extrapolated spot in the Rgel). If the US spot meets this test, it is then put into that eRspot set. If not, a new eRspot set is created along with an EP spot for the Rgel. Note that the numbers of eRspots are treated just the same as normal Rspots. However, eRspots only have EP and US spot labels.

`/ERSPOT` *option*

If the `/EDIFF` switch (Euclidean distance between the **sg2gii** estimated spot centroid and the user interactively specified position) may be appended to the `CREATE` command. It then prints out the landmark-GSF matched spot estimates and the **cmpgl2** landmark validity check. The validity check entry is either `OK` (landmark matched well to segmented spots and thus the segmented spot's centroid is used), `NG` (landmark did not match well and the coordinates of the interactively defined LM are used) or `SM` (this landmark matched the *same* spot as another landmark and is rejected - a fatal error in which case that pair of gels should be re-landmarked, **cmpgl2** re-run on that gel, and the **cgelp2** database rebuilt). As each gel pair GCF file name is entered into the database, it prints the following information:

`/EDIFF` *option*

```
3<CMD>: CREATE/EDiff
Input file or ACC#?: c10253.gcf<CR>
Input file or ACC#?: c10254.gcf<CR>
Input file or ACC#?: c10255.gcf<CR>
Input file or ACC#?: <CR>
```

Typing a null file name or null accession number terminates the list. The default GCF file path is specified by the `gel.rc` state file keyword `ppnp2x` associated entry.

The GCF files may also be specified by their accession numbers: `253.1, 254.1, 255.1` etc. However you *must* be in the directory where the GCF files may be found (probably ∼`/joeUser/gellab/gcf` or ∼`/joeUser/gellab/aux`.

```
4<CMD>: CREATE
```

```
Input file or ACC#?: 327.1<CR>
File /home/joeUser/gellab/c10327.gcf not found - ignoring entry.
Input file or ACC#?: 253.1<CR>
Input file or ACC#?: 254.1<CR>
Input file or ACC#?: 255.1<CR>
Input file or ACC#?: <CR>
```

Alternatively, an indirect file containing the names of the files may be specified using the `@listOfgcfFiles.ccl` response.

```
5<CMD>: CREATE
Input file or ACC#?: @ts3.ccl<CR>
Input file or ACC#?: <CR>
```

acceptable pairing label

When you have entered the last file and typed the empty line, it prompts you for the pairing labels you wish to use in creating the PCG DB.

```
Search for A or S or P or U or E or C or X (and *) [PSAUE]
  A is Ambiguous Pair, S is sure Pair, P is Possible Pair,
  U is Unresolved Spot, C is composite Pair,
  E is Extrapolated Pair,
  X allows accessing the eRspot database,
  XX allows accessing ONLY the eRspot database.
    ?: PS<CR>
```

This last question is used to restrict the data in the data base to specific label types. It is really forcing you to do a `SET LABEL` and so the semantics of that operation still apply. If the `/EDIFF` switch was specified, then it reads in the GCF files but prints out the landmark information as it scans it.

```
[1] c10253.gcf from gel ACC#'s 0250.2 and 0253.1.
Converting D' to % total D'.
Short GCF.
G1[A,549][210,263],E.Diff= 1.4,G2[A, 80][167,249],E.Diff= 4.1-OK
G1[B,445][174,236],E.Diff= 1.4,G2[B, 63][129,221],E.Diff= 1.4-OK
G1[C,348][180,219],E.Diff=  .0,G2[C, 49][136,202],E.Diff= 2.2-OK
G1[D,183][177,176],E.Diff=  .0,G2[D, 25][132,160],E.Diff= 1.0-OK
G1[E,227][231,187],E.Diff= 2.0,G2[E, 32][190,170],E.Diff= 4.5-OK
G1[F,327][238,212],E.Diff= 1.4,G2[F, 45][198,196],E.Diff= 3.6-OK
G1[G,176][318,170],E.Diff= 2.0,G2[G, 24][282,153],E.Diff= 3.2-OK
G1[H,219][305,182],E.Diff=  .0,G2[H, 31][267,165],E.Diff= 3.2-OK
G1[I,360][287,218],E.Diff= 5.0,G2[I, 48][260,201],E.Diff= 6.1-OK
G1[J,403][325,227],E.Diff= 1.0,G2[J, 58][286,211],E.Diff= 1.4-OK
G1[K,460][362,242],E.Diff= 1.4,G2[K, 66][323,228],E.Diff=  .0-OK
G1[L,283][409,201],E.Diff= 1.4,G2[L, 40][359,187],E.Diff= 2.2-OK
G1[M, 85][355,138],E.Diff=  .0,G2[M, 24][282,153],E.Diff= 48.2-NG
G1[N,565][411,263],E.Diff= 2.8,G2[N, 84][364,251],E.Diff= 5.4-OK
G1[O,681][320,310],E.Diff= 2.8,G2[O, 88][317,300],E.Diff= 36.9-NG
G1[P,792][304,363],E.Diff= 3.2,G2[P,102][266,388],E.Diff= 40.2-NG
G1[Q,813][319,383],E.Diff= 2.2,G2[Q,102][266,388],E.Diff= 21.0-NG
```

```
G1[R,760][249,347],E.Diff= 1.4,G2[R, 92][136,349],E.Diff= 72.8-NG
G1[S,794][149,367],E.Diff=  .0,G2[S, 94][108,357],E.Diff= 3.2-OK
G1[T,710][155,324],E.Diff= 3.2,G2[T, 93][130,350],E.Diff= 41.4-NG
G1[U,714][ 88,326],E.Diff= 2.2,G2[U, 90][ 43,313],E.Diff= 1.0-OK
G1[V,806][113,376],E.Diff= 2.2,G2[V, 94][108,357],E.Diff= 32.0-NG
G1[W,270][ 97,199],E.Diff= 2.2,G2[W, 44][115,195],E.Diff= 59.0-NG
Found 135 pairs.

[2] c10254.gcf from gel ACC#'s 0250.2 and 0254.1.
Converting D' to % total D'.
Short GCF.
G1[A,549][210,263],E.Diff= 1.4,G2[A, 80][174,287],E.Diff= 2.0-OK
G1[B,445][174,236],E.Diff= 1.4,G2[B, 62][136,258],E.Diff=  .0-OK
Short GCF.        .
.
.
Etc.
```

Similarly, when the `CREATE` command *without* the `/EDIFF` is specified, then the output is as follows. This is what is normally used.

```
[1] c10253.gcf from gel ACC#'s 0250.2 and 0253.1.
Converting D' to % total D'.
Short GCF.
Found 135 pairs.

[2] c10254.gcf from gel ACC#'s 0250.2 and 0254.1.
Converting D' to % total D'.
Short GCF.
     .
     .
     .
Etc.
```

Finally, it will print a database creation *summary* after it has read in all of the GCF files into the database. For example with the above database with the pairing label set to only (SP+PP):

database summary

```
     .
     .
     .
[83] c20186.gcf from gel ACC#'s 0102.2 and 0186.2.
Converting D' to % total D'.
Short GCF.
Found 203 pairs.


Done processing 85 gels for 325 Rspot sets consisting of 12314 spots.
There are 325 UNextended Rspots.
There are 0 eRspots.
Spot free store has 380902 spots available.
```

```
   2820 sure pairs,
   9494 possible pairs,
   0 ambiguous pairs,
   0 unresolved spots,
   0 extrapolated spots.
   0 composite-spot pairs.
```

/ERSPOT option

The /ERSPOT switch appended to CREATE is used to create an extended CGL database consisting of eRspots as well as Rspots. In the following example, a typical 15 gel database was generated with the prefilter statistics parameters set (using the SET STATISTICS command) to the following *prior* to the create operation:

(area=[20:1000], D′=[3:1000], OD diff=[0:10], DP=[0:2.7]).

Note that eRspots start at Rspot #534 and range up to 711. The following statistics were reported:

```
Done processing 15 gels with 711 Rspot sets consisting of 5630 spots.
There are 533 UNextended Rspot sets.
There are 178 eRspot sets.
Spot free store has 387586 spots available.
   1195 sure pairs,
   2120 possible pairs,
   1799 ambiguous pairs,
   338 unresolved spots,
   178 extrapolated pairs.
   0 composite-spot pairs.
```

<CMD> DBUG

for guru's only

An extensive runtime database debugging facility is available in **cgelp2** for guru level users. This is only available *if* **cgelp2** *was compiled to enable the* DBUG:Onnnn command. The current *dbugCode* and list of available debug bits is listed by entering the DBUG command with no argument.

```
2<CMD>: DBUG
    DBUG bit allocation
    BIT(octal)      function
    ----------      ------
    0400000000       run VERIFY_RSPOT_SET
    0200000000       DUMP_NODE_LINKED_LIST in where Rspot accessed
    0100000000       enable malloc_debug(2) do one malloc_verify()
     040000000       enable malloc_debug(2) and malloc_verify()
     020000000       trace /FIND:ACC#:CC#:DBUG
     010000000       MAKE_OUT_FILE, DMPCGL, DMPSPOT_SETS
      04000000       VALIDLANDMARKS, SET_xxxx, SET_LSQ_NORM_CALIB
      02000000       EXTEND_PCG, VERIFY_PCG_DB and their calls
      01000000       Print LMN data with PRT_SPOTS, CVSPOTSTR
       0400000       NEW_RCRD
       0200000       PRT_SPOTS
```

```
        0100000       MORE_OUTSTR print local line count
         040000       PAGCGL RD/WT hdr bOffset's, MAK_,INI_,SET_,FIN_
         020000       RD_FIELD/WT_FIELD
         010000       PRT_SPOTS
          04000       CK_RSPOT_METRIC - print failCodes > 0
          02000       PAGCGL - RD_BLOCK
          01000       PLT_MOSAIC, POP_ENV, onINTR, ERR_RESTART
           0400       CK_RSPOT_METRIC - *** always succeed ***
           0200       CK_LIMITS_RSPOT - *** always succeed ***
            040       CK_LIMITS_RSPOT - debug printout
            020       TAB_PRT_CORRELATION_TABLE
            010       PUT_IN_ORDER
             04       MAK_PAIR, MAK_RCD
             02       PUSH_PAIR
             01       RD_GELPAIRS, EXTRAPOLATE_CGL
dBugCode=00
```

It may be set by specifying the inclusive-OR of the above dbug options desired.

```
2<CMD>: DBUG:0620616
dBugCode=0620616
```

which selects:

```
      0400000          NEW_RCRD
      0200000          PRT_SPOTS
      020000           RD_FIELD/WT_FIELD
      0400             CK_RSPOT_METRIC - *** always succeed ***
      0200             CK_LIMITS_RSPOT - *** always succeed ***
      010              PUT_IN_ORDER
      04               MAK_PAIR, MAK_RCD
      02               PUSH_PAIR
```

You can have it report why the prefilter failed (if it did) by setting `DBUG:04000`. You can force the prefilter to always accept an Rspot set by setting `DBUG:0600`.

Rspot daemon

It is possible to activate debugging for a particular spot specified by the 2-tuple (ACC#,CC#). Alternatively, if the ACC# field is 0000 or null, then the CC# field is interpreted as the Rspot set name itself. This is called a *Rspot daemon*. It is activated at global command level by adding the `/FInd:ACC#,CC#,DBUGcode` or `/FInd:0000,Rspot#,DBUGcode` switch. This causes it to print out debugging information (specified by the DBUGcode) for that Rspot# if it is ever found.

If the dBugCode is set to `02000000` when extending the PCG DB with an `EXTRAPOLATE` command, then it does a *verify* pass through the database checking that the maximum number of links per Rspot-set are $\leq$ the number of nodes which might occur in the Rspot set. Thus we can detect possible corruption in the PCG DB.

It is possible, in the debugging version of **cgelp2**, to gather statistics on dynamic memory utilization. By doing a `DBUG:0100000000` before and after an operation you can see how the memory was used. Some of the statistics are meaningful only if you read the storage allocation C code.

```
388<CMD>: DBUG:0100000000
dBugCode=0100000000
Pcalloc()/Pfree() Statistics.
Note: memStat[N] is frequency of Pcalloc() requests of size N to 2*N.
memStat[32]=1416232
memStat[64]=13382
memStat[128]=56625
memStat[256]=64594
memStat[512]=2947
memStat[1024]=250
memStat[2048]=217
memStat[4096]=1
memStat[8192]=79
memStat[16384]=2
minDynamicPtr = 0Xfb16c
maxDynamicPtr = 0X139167
totSpaceUsed = 183707
totMemCalloc = 253947
% memory used = tot/(max-min) = 72
# nBadMagicBytes= 0
# nPfreeNullPtrs= 0
# nPfreeLTminDynPtr= 8509
# nPfreeGTmanDynPtr= 0
# nfLocal= 10
```

<CMD> DCplot

gel-titration scatter-plot

Draw (optionally generating a plot *.ugf* file) a density-class feature plot of a Rspot set of gels. A least squares linear fit of the data (m, b) in $(density = m * classfeature + b)$ is computed along with its correlation coefficient and standard error (from the residual sum of squares). This line and features are also drawn on the display. The plot is drawn in linear space with the scale factor defaulting to the maximum of the two ranges (density and class feature). This may be explicitly specified using the `/SCALE:S1,S2` switch. `DCPLOT` prompts the user to enter the name of the Rspot set to be plotted. Optionally, it plots `ALL` of the Rspots in the current SRL (Search Results List) may be plotted by specifying '`*`'. Use `SET DISPLAY` to define the plot or display type to be used (as well as associated plot switches: `/CLASS`, `/GELS`, `/LINElabel`, `/SCALE`, `/UGFlabel`). Figure 3.2 shows a sample plot output. A response to a typical prompt follows:

```
14<CMD>: DCPLOT
Plot Rspot# vs. Class values. (Optional switches):
Rspot#, abscissa (class) label ?: <CR>
Bad gels specification - ignoring DCPLOT command.
```

```
15<CMD>: DCPLOT
Plot Rspot# vs. Class values. (Optional switches):
Rspot#,abscissa (class) label)?: 27,uGrams/GEL/SCALE:200x200<CR>
```

Figure 3.2. Sample **cgelp2** `DCPLOT` plot. The plot file was ploted using the `LASER` display option set with `SET DISPLAY`.

<CMD> DDplot

gel-gel scatter plot

Draw (optionally generating a plot *.ugf* file) a density-density plot of a pair of gels. The correlation coefficient and rFactor [AndN81] are also computed and drawn on the display. The plot is drawn in $log - log$ space. It will prompt the user to enter the names of two different gels to be plotted. Optionally, Rspots in the current SRL (Search Results List) may be labeled. It will then request the type of display or plot to be performed (if it has not been defined before). Use `SET DISPLAY` to define the plot or display type to be used (as well as associated plot switches: `/CLASSnames`, `/LINE`, `/FILLabel`, `/SRLlabel`, `/UGFlabel`). Figure 3.3 shows a sample plot output. A response to a typical prompt follows:

```
17<CMD>: DDPLOT
Plot two gels (XXXX.E, XXXX.E)?: <CR>
Bad gels specification - ignoring DDPLOT command.
```

```
    18<CMD>: DDPLOT
    Plot two gels (XXXX.E, XXXX.E)?: 324.1,384.1<CR>
       . . .
```

or to label Rspots in the current SRL,

```
    38<CMD>: DDPLOT
    Plot two gels (XXXX.E, XXXX.E)?:  324.1,384.1 /SRLlabel <CR>
    Printing plot file on laser printer
    plotn -display:4010 ~joeUser/gellab/gen/000015.ugf | tek2psG | lpr -Plaser
    Plot file: /home/joeUser/gellab/gen/000015.ugf
[07:55:13AM] Real TIME =00:01:34  CPU TIME =00:00:58,  61.70%
```

Figure 3.3. Sample **cgelp2** `DDPLOT` plot. The plot file was ploted using the `LASER` display option set with `SET DISPLAY`. The `/SRLlabel` option was used to label those spots in the search results list.

<CMD> Dendrogram

Generate a Dendrogram cluster analysis plot. It plots the Dendrogram on the display or *.ugf* plot file. Since the PREFILTER is active, you can restrict spots which you wish to view in the Dendrogram plot which are normally specified in the SRL. It defaults to generating a Dendrogram of gels in the working set as objects and the set of Rspots in the SRL (their densityies/gel) as the features. If you specify the `/SRL` plot switch, it will cluster the Rspots in the SRL as a function of the expression profiles features of the set of gels in the working set of gels. Use `SET DISPLAY` to define the plot or display type to be used (as well as associated plot switches). Optionally, the `/TITLE:` switch may be used to specify different title text for the plot. This causes: (1) file `dendrogram.sps` to be generated, and (2) the invoking of `system("dendrogram dendrogram.sps -DISPLAY:laser -MeanClasses")`.

Dendrogram Use `SET DISPLAY` to define the plot or display type to be used. The following

plot switches are active: `/COLORlabel`, `/LABEL`, `/SRL`, `/TITLE:'...'`). The default switches are: `/NOSRL`. The following prompt is given the user for generating a Dendrogram plot:

```
25<CMD>: Dendrogram
Drawing a Dendrogram of SRL.
    . . .

26<CMD>: Dendrogram
Drawing a Dendrogram of SRL.
    . . .
```

<CMD> DO

Get future **cgelp2** commands from a script file instead of typing them in. You may not currently nest 'DO' files. By convention the files have a *.gdo* file extension. UNIX batch script files used with GELLAB are given a *.do* file extension. The *.gdo* files, although a script file is only used as input to **cgelp2**. The `DO` command is automatically invoked when starting **cgelp2** with the `-f` switch. Also, doing the `!@n1,n2` "do range" history command (see Section 3.3.7, page 173) invokes the `DO` command. NOTE: if you specify the file with '//' notation, it will convert it to upper case. See the example **cgelp2** script in Section 3.9, page 369. For example, a **cgelp2** script `setup.gdo` can be executed from within **cgelp2** as:

additional `.gdo` *scripts*

```
3<CMD>: DO
Script file to use?: setup.gdo<CR>
```

<CMD> DUmp CGL

Dump the CGL database in an ASCII (*.cgl*) file which may be printed. Only Rspot sets meeting the prefilter criteria are output. The following example was obtained for a small region of replicate gels.

ASCII CGL file

```
28<CMD>: DUMP CGL
Output file:[000001.cgl]?: hm9pcg.cgl<CR>

29<CMD>: DUMP CGL/SRLSS[5]
Output file:[000001.cgl]?: hm9s05.cgl<CR>
```

The file has the following format:

```
File: hm9pcg.cgl 12/19/1988, 02:33:39PM /home/joeUser/gellab/pcg/hm9pcg.pcg
Date database created: 11/02/1988, 06:46:47
Date last terminal session: 11/02/1988, 07:58:46
Rspot Mapping file [].
Map Rspots to Fspots is OFF.
Ratio-List density mode calibration EXISTS.
Least-Squares density mode calibration EXISTS.
```

```
Image size [rowsXcolumns]=[512X512]
Relative distance limits [0.00,512.00] pixels
Mn DL limits [0.00,512.00] pixels
Mn DP limits [0.00,10.00] pixels
MN area limits [25.00,10000000.00] pixels**2
MN density (Mode: R) limits [2.00,10000000.00]
Coef. variation: S.D./Mn Rset area limits [0.00,10.00]
Coef. variation: S.D./Mn Rset dens limits [0.00,10.00]
Spot OD difference limits [0.00,1.80]
Class difference t-Test, F-test, Rank order p-value limit is .90:
Check if # gels in Rspot set [0:1000]
(x,y) calibration file: <NONE>
pIe Rgel limits [0.00,511.00] pixels
MW Rgel limits [0.00,511.00] pixels
Pairing-Search-Label [PSEAU]
List of Rspots used in Ratio list normalization:
 11 70 97 181 612

Class # 1(-AML)=0524.1 0497.1 0505.1
Class # 2(-CLL)=0578.2 0515.1 0517.1
Class # 3(-ALL)=0569.1 0511.1 0514.1
Class # 4(-HCL)=0584.1 0593.2
Class # 5(HL-60)=0596.1
Class # 6(<NULL>)=
Class # 7(<NULL>)=
Class # 8(<NULL>)=
Class # 9(<NULL>)=
There are 12 gels with 1006 Rspot sets consisting of 13968 spots.
There are 680 UNextended Rspot sets [0:680]
There are 326 eRspot sets [681:1006]
Spot free store has 1165680 spots available. The Rgel is 0524.1
   958 sure pairs,
   2708 possible pairs,
   2349 ambiguous pairs,
   568 unresolved spots,
   7385 extrapolated pairs.
   0 composite-spot pairs.

(1)  0524.1
   GSF(Tot D'=5889, #spots=680), Ratio D' sum=185.80 eF=1.000
   Study: / HEME MALIG-AML,MYELOID

(2)  0569.1
   GSF(Tot D'=15954, #spots=789), Ratio D' sum=255.80 eF=1.000
   Study: / HEME MALIG-ALL,LYMPHOID

(3)  0578.2
   GSF(Tot D'=7934, #spots=521), Ratio D' sum=107.00 eF=1.000
   Study: / HEME MALIG-CLL,LYMPHOID (DUPL. SCAN)
        .
        .
        .
(12)  0593.2
```

```
   GSF(Tot D'=3676, #spots=396), Ratio D' sum=168.10 eF=1.000
   Study: / HEME MALIG-HCL,LYMPHOID

Rspot[   1] mnXYDA=(302,172,3.17,29)+-(6.00,8.12,5.28,25) CVD=1.66 #G=18
 ACC#[Index] C   RDens  area pkOD    D' Lbl LM  DP  DL  Dx   Dy  Xabs Yabs ODdf
------------ -  ------- ---- ----    -- --- -- ---- -- ---- ---- ---- ---- ----
0511.1[ 172] 3   20.43R   60 0.68  13.2  PP A   4.1 24 (  0,-21) (352,172) 0.43
0515.1[ 177] 2   11.71R   63 0.50  11.0  PP A   2.2 24 (  5,-23) (341,179) 0.35
0578.2[  55] 2    7.29R   81 0.28   7.8  AP A   4.0 24 (  7,-23) (265,106) 0.28
0514.1[ 111] 3    3.92R   13 0.28   1.3  PP B   6.0 30 (-21, 13) (322,155) 0.11
0596.1[ 122] 5    3.70R   66 0.46  11.0  PP A   3.0 24 (  3,-21) (313,133) 0.35
0569.1[ 168] 3    2.89R   48 0.46   7.4  AP A   8.5 24 ( -5,-21) (278,156) 0.35
0569.1[ 163] 3    2.11R   38 0.40   5.4  PP A   2.2 24 (  1,-23) (284,154) 0.22
0524.1[ 161] 1    1.13R   35 0.14   2.1  PP A   2.2 24 (  3,-24) (305,167) 0.14
0497.1[ 339] 1    1.00R   30 0.36   1.3  AP A   7.1 24 ( -3,-20) (318,209) 0.11
0517.1[ 163] 2    0.81R   14 0.22   0.9  PP A   3.2 24 (  4,-21) (347,163) 0.05
0497.1[ 335] 1    0.77R   16 0.36   1.0  PP A   4.0 24 (  3,-20) (324,209) 0.11
0593.2[  39] 4    0.65R   43 0.08   1.1  PP A   5.0 24 (  2,-19) (326,148) 0.08
0578.2[  64] 2    0.47R   16 0.06   0.5  AP A   7.8 24 ( -2,-18) (256,111) 0.06
0593.2[  44] 4    0.18R   10 0.08   0.3  AP A   7.0 24 ( -2,-19) (322,148) 0.07
0514.1[8127] 3    0.00R    0 0.00   0.0  EP A   0.0  0 (  2,-21) (317,161) 0.00
0505.1[8127] 1    0.00R    0 0.00   0.0  EP A   0.0  0 (  2,-21) (274,124) 0.00
0584.1[8127] 4    0.00R    0 0.00   0.0  EP A   0.0  0 (  2,-21) (292,163) 0.00
0578.2[8127] 2    0.00R    0 0.00   0.0  EP A   0.0  0 (  2,-21) (260,108) 0.00

Rspot[   2] mnXYDA=(306,168,2.98,27)+-(8.60,11.53,4.93,27) CVD=1.65 #G=13
 ACC#[Index] C   RDens  area pkOD    D' Lbl LM  DP  DL  Dx   Dy  Xabs Yabs ODdf
------------ -  ------- ---- ----    -- --- -- ---- -- ---- ---- ---- ---- ----
0514.1[ 105] 3   13.25R   53 0.28   4.4  PP A   0.0 29 (  1,-29) (316,153) 0.13
0515.1[ 178] 2   13.21R   46 0.58  12.4  PP A   0.0 27 ( 13,-24) (349,178) 0.33
0578.2[  55] 2    7.29R   81 0.28   7.8  PP A   2.0 26 (  7,-23) (265,106) 0.28
0596.1[ 102] 5    1.18R   71 0.40   3.5  PP A   2.2 28 (  8,-27) (318,127) 0.33
0511.1[ 162] 3    1.08R   29 0.34   0.7  PP A   0.0 30 (  6,-29) (358,164) 0.09
0524.1[ 163] 1    0.70R   14 0.14   1.3  PP A   5.8 28 (  7,-25) (309,166) 0.14
0497.1[ 317] 1    0.69R   19 0.34   0.9  PP B   1.0 27 (-23, 14) (323,200) 0.09
0517.1[ 157] 2    0.54R   13 0.22   0.6  PP A   4.0 26 (  6,-21) (349,163) 0.08
0593.2[  26] 4    0.42R   20 0.10   0.7  PP A   0.0 33 (  8,-32) (332,135) 0.09
0569.1[ 150] 3    0.39R   12 0.26   1.0  PP A   5.8 28 (  2,-28) (285,149) 0.09
0505.1[8127] 1    0.00R    0 0.00   0.0  EP A   0.0  0 (  6,-26) (278,119) 0.00
0497.1[8127] 1    0.00R    0 0.00   0.0  EP A   0.0  0 (  6,-26) (327,203) 0.00
0584.1[8127] 4    0.00R    0 0.00   0.0  EP A   0.0  0 (  6,-26) (296,158) 0.00

Rspot[   7] mnXYDA=(302,185,5.28,34)+-(3.00,3.00,6.72,29) CVD=1.27 #G=19
 ACC#[Index] C   RDens  area pkOD    D' Lbl LM  DP  DL  Dx   Dy  Xabs Yabs ODdf
------------ -  ------- ---- ----    -- --- -- ---- -- ---- ---- ---- ---- ----
0505.1[  11] 1   22.80R   50 0.42   5.7  AP A* 7.3  7 (  0,  0) (272,145) 0.42
0593.2[  67] 4   17.25R  106 1.04  29.0  AP A* 7.3  7 (  0,  0) (324,167) 1.04
0515.1[ 203] 2   14.91R   58 0.76  14.0  SP A   2.2  7 (  4, -6) (340,196) 0.61
0497.1[ 360] 1   10.60R   93 0.72  13.8  AP A   5.8 13 (  5,-12) (326,217) 0.47
0514.1[ 139] 3    7.23R   33 0.30   2.4  SP A   1.0  7 (  1, -7) (316,175) 0.16
0569.1[ 205] 3    6.45R   36 1.34  16.5  SP A   2.0  7 (  0, -7) (283,170) 1.09
0511.1[ 204] 3    6.35R   64 0.42   4.1  SP A   1.0  8 (  2, -8) (354,185) 0.17
0578.2[  82] 2    6.26R   33 0.42   6.7  SP A   2.2  7 (  4, -6) (262,123) 0.28
```

```
0517.1[ 180] 2    2.96R   34 0.42   3.3  SP A   0.0  7 (  2, -7) (345,177) 0.27
0578.2[  78] 2    1.96R   35 0.18   2.1  AP A   5.0 10 ( -2,-10) (256,119) 0.18
0524.1[ 197] 1    1.61R   23 0.28   3.0  SP A   2.0  7 (  2, -7) (304,184) 0.25
0596.1[ 150] 5    0.64R   17 0.48   1.9  SP A   2.0  7 (  0, -7) (310,147) 0.23
0515.1[ 202] 2    0.43R   10 0.24   0.4  AP A   4.0  8 ( -2, -8) (334,194) 0.10
0569.1[ 187] 3    0.31R   31 0.32   0.8  AP A  10.0 14 ( -6,-13) (277,164) 0.31
0584.1[ 181] 4    0.30R   23 0.26   0.7  SP A   1.0  7 (  2, -6) (292,178) 0.26
0569.1[ 210] 3    0.23R   13 0.42   0.6  AP A   9.0 10 ( -7, -7) (276,170) 0.31
0593.2[8127] 4    0.00R    0 0.00   0.0  EP A   0.0  0 (  1, -6) (325,161) 0.00
0505.1[8127] 1    0.00R    0 0.00   0.0  EP A   0.0  0 (  1, -6) (273,139) 0.00
0497.1[8127] 1    0.00R    0 0.00   0.0  EP A   0.0  0 (  1, -6) (322,223) 0.00

Rspot[   9] mnXYDA=(302,191,27.72,88)+-(2.00,1.73,15.57,26) CVD=0.56 #G=14
 ACC#[Index] C   RDens  area pkOD    D' Lbl LM  DP  DL  Dx   Dy  Xabs Yabs ODdf
------------ -  ------- ---- ----    -- --- -- ---- -- ---- ---- ---- ---- ----
0578.2[  92] 2   52.06R   89 1.82  55.7  SP A*  0.0  0 (  0,  0) (258,129) 1.77
0569.1[ 222] 3   48.16R  116 2.00 123.2  SP A*  0.0  0 (  0,  0) (283,177) 1.75
0514.1[ 144] 3   46.99R   78 0.92  15.6  SP A*  0.0  0 (  0,  0) (315,182) 0.78
0511.1[ 221] 3   44.12R  116 1.10  28.5  SP A*  0.0  0 (  0,  0) (352,193) 0.85
0497.1[ 382] 1   36.10R  109 1.50  47.0  SP A*  0.0  0 (  0,  0) (321,229) 1.25
0515.1[ 210] 2   31.84R   86 1.24  29.9  SP A*  0.0  0 (  0,  0) (336,202) 1.09
0524.1[ 210] 1   23.25R  110 1.38  43.2  SP A*  0.0  0 (  0,  0) (302,191) 1.38
0505.1[  11] 1   22.80R   50 0.42   5.7  SP A*  0.0  0 (  0,  0) (272,145) 0.42
0517.1[ 189] 2   20.02R   79 1.02  22.3  SP A*  0.0  0 (  0,  0) (343,184) 0.90
0596.1[ 169] 5   19.33R  127 1.42  57.5  SP A*  0.0  0 (  0,  0) (310,154) 1.22
0593.2[  67] 4   17.25R  106 1.04  29.0  SP A*  0.0  0 (  0,  0) (324,167) 1.04
0584.1[ 199] 4   14.29R   81 1.50  33.2  SP A*  0.0  0 (  0,  0) (290,184) 1.45
0569.1[ 226] 3    6.14R   56 1.20  15.7  AP A   8.1  8 ( -8,  1) (275,178) 1.09
0596.1[ 190] 5    5.68R   41 0.98  16.9  AP A   7.0  7 (  1,  7) (311,161) 0.73
```

.
.
.

<CMD> EDit

Edit spots in the PCG database. This non-graphical spot editor allows moving, defining or altering spots or their data in the PCG database. It has the following commands.

RSPOT EDIT CODES

j spot j is defined as 3-tuple $S_j = (g : r, l)$ where:

g is the ACC# of an entry in r.

r is the Rspot set.

i (i_1 and i_2) is the spot segmenter index of an entry in r.

A$(g : r, i)$ Alter fields in existing spot $(g : r, i)$.

D$(g : r, i)$ Delete spot $(g : r, i)$.

H or ? print this menu.

I$(g : r, i)$ Insert new spot $(g : r, i)$ and alter values.

M$(g : r1, i1 < r2, i2)$ Merge spot $(g : r2, i2)$ with $(g : r1, i1)$ deleting $(g : r2, i2)$ spot.

S$(g : r1, i1 < r2, i2)$ Swap spot $(g : r2, i2)$ with $(g : r1, i1)$ for same gel.

T$(g : r1, < r2, i2)$ Transfer spot $(g : r2, i2)$ to Rspot set $(g : r1,)$.

The 'Alter' subcommand presents the current field value and asks for a new value. The following illustrates the use of 'Alter'.

```
33<CMD>: EDIT
 ?:A(0252.2:6,S)<CR>
Field Density[ 5.3930]?:16.0<CR>
Field DP[ .0000]?:<CR>
Field sX[ 5.0000]?:<CR>
Field sY[ 3.0000]?:<CR>
Field SG2GII index[277]?:<CR>
Field DL[0]?:<CR>
Field LMset[A]?:<CR>
Field Pairing label[S]?:<CR>
Field ACC#[0252.2]?:<CR>
Field xRel[0]?:<CR>
Field yRel[0]?:<CR>
Field Xabs[196]?:<CR>
Field Yabs[281]?:<CR>
Field Area[240]?:<CR>
Field maxD[1.2100]?:<CR>
Field minD[0.23]?:<CR>
Altered Rspot[0252.2:6, label=S]<CR>


34<CMD>: EDIT
 ?:A(253.2,42,A)<CR>
Field Density[ .1291]?:2.13<CR>
Field DP[ 3.0000]?:<CR>
Field sX[ 1.2000]?:<CR>
Field sY[ 1.1000]?:<CR>
Field SG2GII index[106]?:<CR>
Field DL[15]?:<CR>
Field LMset[F]?:<CR>
Field Pairing label[A]?:S<CR>
Field ACC#[0253.2]?:<CR>
Field xRel[25]?:26<CR>
Field yRel[-4]?:-5<CR>
Field Xabs[277]?:278<CR>
Field Yabs[198]?:197<CR>
Field Area[16]?:<CR>
```

```
Field maxD[0.9400]?:<CR>
Field minD[0.2100]?:<CR>
Altered Rspot[0253.2:42, label=S]<CR>
 ?:
```

The null request terminates the Rspot `EDIT`.

<CMD> EXIt

Exit **cgelp2** to UNIX to save the PCG database file for later use. The total real time (wall clock time) and CPU time used for the entire session are printed. It also prints the name of the database to remind the user what PCG database was being used as follows. Unlike command `BACKUP` which checkpoints the database and continues in **cgelp2**, `EXIT` does not continue after it saves the database. Although typing "BYE" is equivalent to an program exit on some computer systems, typing `BYE` is not equivalent to typing `EXIT` in **cgelp2** and it will warn you of that fact.

```
44<CMD>: EXIT
Total session times: Real TIME =00:41:21  CPU TIME =00:34:40,  83.84%

Saving PCG database
        /home/joeUser/gellab/pcg/apzpcg.pcg
for later use.
To use database at a later time, run CGELP2 then declare the data
base using the SET DATABASE command or restart cgelp2 by
    cgelp2 -d /home/joeUser/gellab/pcg/apzpcg.pcg
```

Note that if the PCG DB file is write protected or the `PROTECT` command was invoked, then you will not be allowed to checkpoint the database when doing an `EXIT`. Similarly,

```
20<CMD>: BYE
If you want to exit cgelp2, use 'EXIT'
[05:30:49PM] Real TIME =00:00:00  CPU TIME =00:00:00,  0.00%
21<CMD>:
```

<CMD> EXPplot

expression profile plots

Draw (optionally generating a plot *.ugf* file) expression profile plots of all Rspots in the current SRL. `EXPPLOT` prompts the user to enter additional switch options. Use `SET DISPLAY` to define the plot or display type to be used (as well as associated plot switches: `/CLASS`, `/LINE`, `/SCALE:`*dS,cS*, `/UGFlabel`). Figure 3.4 shows a sample plot output. A response to a typical prompt follows:

```
252<CMD>: expplot
Plot Rpspot Expression-Profiles vs. Class values.
Options ?: ?

 DEFAULT SWITCH OPTIONS
```

```
 ----------------------
  /NOCLASS - default don't draw class names in plot.
  /NOLINE - default is to not to draw boxes around each subgraph.
  /NOSCALE:dSfactor,cSfactor - default is scale Dens by each Rspot maxima.
  /UGF - label plot with UGF file number.
[04:36:36PM] Real TIME =00:00:00  CPU TIME =00:00:00,  0.00%

253<CMD>: expplot
Plot Rpspot Expression-Profiles vs. Class values.
Options ?: /classnames
Rspot[232] failed the PREFILTER - continuing.
Rspot[281] failed the PREFILTER - continuing.
Rspot[347] failed the PREFILTER - continuing.
Rspot[379] failed the PREFILTER - continuing.
Rspot[428] failed the PREFILTER - continuing.
Rspot[468] failed the PREFILTER - continuing.
Rspot[539] failed the PREFILTER - continuing.
Rspot[613] failed the PREFILTER - continuing.
Printing plot file on laser printer
 plotn -display:4010 /home/joeUser/gellab/gen/000054.ugf | tek2psG | \
                                                           lpr -Plaser
Plot file: /home/joeUser/gellab/gen/000054.ugf
[04:36:48PM] Real TIME =00:00:09  CPU TIME =00:00:02,  22.22%
```

Figure 3.4. Sample **cgelp2** `EXPLOT` plot. The plot file was ploted using the `LASER` display option set with `SET DISPLAY`.

<CMD> EXTrapolate
Extrapolate missing spots for gels g in Rspot sets from the mean $(\bar{d}_x, \bar{d}_y) + LM_{x,y,g}$ position creating new spots in the corresponding Rspot sets with zero density and an EP label. Only SP, PP and US labeled spots from the same landmark are used for estimating the mean spot. If an EP label already exists for a gel, then do not create an EP spot for that gel. This means that doing an `EXTRAPOLATE` more than once will not add any additional EP spots. This also implies that if you add new gels to the database at a later time with `CREATE`, and then do an EXTRAPOLATE, it will extrapolate only spots missing from the the new gels and not change the older EP spots. The entire PCG DB is automatically searched (equivalent to a `SET LABEL` to `SPUEAX`). However, the `SET STATISTICS` limits should be opened up to include all Rspot sets. Note that `EXTRAPOLATE` is an atomic operation which should not be interrupted or corruption of the PCG DB could result.

As a side effect, any Rgel spots in the Rspot set with an AP label for which a

SP or PP label exists for another spot in the Rspot set will be changed to that SP or PP label. This operation is a consistency check on this condition and may be used to upgrade earlier databases where this condition was not detected. Typical output is as follows:

```
28<CMD>: EXTRAPOLATE/NOQUIET
Creating EXTRAPOLATED Rspot sets for gels missing from the
Rspot set by adding the mean Rspot set (Dx,Dy) to the missing gel's
landmark position. Extrapolated Rspots are put into the search
results list. Note the /NOQUIET switch may be appended to the command
as 'EXTRAPOLATE/NOQUIET' in order to print the names of gels and
Rspot sets as the spots are being extrapolated.
Estimating Rspot[9] for gels: 0260.2,
Estimating Rspot[13] for gels: 0251.2, 0256.2, 0265.2,
Estimating Rspot[14] for gels: 0251.2, 0253.2, 0257.2, 0261.2,
                               0265.2,
Estimating Rspot[19] for gels: 0265.2,
Estimating Rspot[25] for gels: 0252.2, 0257.2, 0260.2, 0264.2,
                               0265.2,
.
.
.
Extrapolated 127 Rspots.
```

If you use the default /QUIET switch, it will not tell you about each EP added, just the total.

```
273<CMD>: EXTRAPOLATE/QUIET
GLOBAL CMD SWITCHES: /Quiet
Creating EXTRAPOLATED Rspot sets for gels missing from the
 Rspot set by adding the mean Rspot set (Dx,Dy) to the missing gel's
landmark position. Extrapolated Rspots are put into the search
results list. Note the /NOQUIET switch may be appended to the command
as 'EXTRAPOLATE/NOQUIET' in order to print the names of gels and
Rspot sets as the spots are being extrapolated.

Extrapolated 906 Rspots.
[07:13:40AM] Real TIME =00:04:21  CPU TIME =00:01:54,  43.68%
```

<CMD> Features

Print maximum values of Rspot set features found in the PCG DB when it was constructed. This can be useful in helping estimate the upper bound to try when setting the prefilter parameters.

```
68<CMD>: FEATURES
CURRENT MAXIMUM VALUES OF FEATURES IN CGL DATABASE
  AREA          = 308.00
  DENSITY       = 137.62
  MAX-OD        = 2.19
```

```
MIN-OD        = 0.26
OD-DIFFERENCE = 4.80
DP            = 99.00
DL            = 215.00
PIE           = 512.00
MW            = 512.00
CV-DENSITY    = 2.50
#GELS/RSET    = 3.00
```

<CMD> Gels

Gels lists the accession number names, total (and normalized total) densities, total number of spots/gel, and study of the current gels. Some typical output is for Ratio-list normalization and then for least-squares normalization is given.

```
87<CMD>: GELS
There are 12 gels with 1006 Rspot sets consisting of 6909 spots.
There are 680 UNextended Rspot sets [0:680]
There are 326 eRspot sets [681:1006]
Spot free store has 1172739 spots available. The Rgel is 0524.1
   958 sure pairs,
   2708 possible pairs,
   2349 ambiguous pairs,
   568 unresolved spots,
   326 extrapolated pairs.
   0 composite-spot pairs.
(1)  0524.1
   GSF(Tot D'=5889, #spots=680), RatioSUM D'=185.80 eF=1.000
   Study: / HEME MALIG-AML,MYELOID

(2)  0569.1
   GSF(Tot D'[=15954, #spots=789), RatioSUM D'=255.80 eF=1.000
   Study: / HEME MALIG-ALL,LYMPHOID

(3)  0578.2
   GSF(Tot D'[=7934, #spots=521), RatioSUM D'=107.00 eF=1.000
   Study: / HEME MALIG-CLL,LYMPHOID (DUPL. SCAN)

(4)  0584.1
   GSF(Tot D'=8399, #spots=697), RatioSUM D'=232.40 eF=1.000
   Study: / HEME MALIG-HCL,LYMPHOID
              .
              .
              .
```

Where `TotD'` is the sum of integrated density corrected for background for the entire gel; `# spots` is the total number of spots quantitated for the gel; `Ratio D' sum` is the ratio-normalization scale factor used to divide a D' spot value for this gel; and `eF` is the optional exposure correction factor. For the least-squares density normalization,

```
88<CMD>: GELS
There are 12 gels with 1006 Rspot sets consisting of 13968 spots.
There are 680 UNextended Rspot sets [0:680]
There are 326 eRspot sets [681:1006]
Spot free store has 1165680 spots available. The Rgel is 0524.1
   958 sure pairs,
   2708 possible pairs,
   2349 ambiguous pairs,
   568 unresolved spots,
   7385 extrapolated pairs.
   0 composite-spot pairs.

(1)  0524.1
   GSF(Tot D'=5889, #spots=680), Mj=1.000 b=0.000 r=1.000 #pairs=5 eF=1.000
   Study: / HEME MALIG-AML,MYELOID

(2)  0569.1
   GSF(Tot D'=15954, #spots=789), Mj=0.672 b=2.800 r=0.950 #pairs=5 eF=1.000
   Study: / HEME MALIG-ALL,LYMPHOID

(3)  0578.2
   GSF(Tot D'=7934, #spots=521), Mj=0.351 b=29.656 r=0.600 #pairs=5 eF=1.000
   Study: / HEME MALIG-CLL,LYMPHOID (DUPL. SCAN)

(4)  0584.1
   GSF(Tot D'=8399, #spots=697), Mj=0.589 b=9.782 r=0.860 #pairs=5 eF=1.000
   Study: / HEME MALIG-HCL,LYMPHOID
              .
              .
              .
```

The (M_j, b) are the least square normalization coefficients; r is its correlation coefficient; *# spots* are the number of paired spots used in calculating the coefficients.

The specific features of a particular gel may be obtained by restricting the output using the `/OPTION=ACC#` switch. For example,

```
89<CMD>: GELS/OPTION:0524.1
GLOBAL CMD SWITCHES: /Option:0524.1
There are 12 gels with 1006 Rspot sets consisting of 13968 spots.
There are 680 UNextended Rspot sets [0:680]
There are 326 eRspot sets [681:1006]
Spot free store has 1165680 spots available. The Rgel is 0524.1
   958 sure pairs,
   2708 possible pairs,
   2349 ambiguous pairs,
   568 unresolved spots,
   7385 extrapolated pairs.
   0 composite-spot pairs.

(1)  0524.1
   GSF(Tot D'=5889, #spots=680), Ratio D' sum=185.80 eF=1.000
   Study: / HEME MALIG-AML,MYELOID
```

. . .

/FULL *option*

The /FULL switch causes **sg2gii** and **cmpgl2** parameters associated with each gel to be printed. Since there is a lot of output, you should probably restrict it with the /Option switch.

***** EDIT NOTE: not fully operational at this time****

```
90<CMD>: GELS/OPTION:0742.1/FULL
     . . .
(1)  0524.1
   GSF(Tot D'=5889, #spots=680), Ratio D' sum=185.80 eF=1.000
   Study: / TC-27
GEL[] FULL GSF/GCF PARAMETERS
GSF SW: -ALLOWTCHEDGES -SHORTGSF -7X7LOWPASS
CW[36:497,74:508]
AREA-LIMITS[10.0:2000.0]
D'-LIMITS[0.30:500.0]
OD-RANGE-LIMITS[0.03:2.7]
GAUSSIAN-FILTER SIZE: 7x7
ZONAL-NOTCH FILTER SIZE: 32x32
BACKGROUND-OD-SEEN[0.0:0.88]
# SPOTS-BEFORE-SIZING: 730
# SPOTS-OMITTED: 11036
# SPOTS-AFTER-SIZING: 680
D' SPOTS-BEFORE-SIZING: 7605.22
D' SPOTS-OMITTED: 1474.70
D' SPOTS-AFTER-SIZING: 5889.24
AREA-BEFORE-SIZING: 28720
AREA-OMITTED: 67613
AREA-AFTER-SIZING: 27872
DENSITY-SPOTS-OMITTED-TO-ACCEPTED: 19%
GCF SW:
CMPGEL T1=5.0 T2=10.0
#-US(INITIAL,SECONDARY)[212:200]
#-SP(INITIAL,SECONDARY)[210:210]
#-PP(INITIAL,SECONDARY)[502:530]
#-AP(INITIAL,SECONDARY)[304:288]
#-CP(INITIAL,SECONDARY)[0:0]
MEAN DP(SP+PP)=5.28 MEAN DP'((|G1|+|G2|)/(SP+PP))=8.05
```

<CMD> HElp

Print the list of top level commands. In addition, help may be obtained on specific commands or subcommands. (In which case the SET MORE option is automatically enabled during the printout of the help entry). Help is invoked by typing:

<CMD>: HELP *command*

or on a particular subcommand by:

<CMD>: HELP *command: subcommand*

For example,

```
<CMD>:  HELP INQUIRE
```

or on a particular subcommand by:

```
<CMD>:  HELP INQUIRE: PRINT
```

When a specific 'command' type argument is specified for a `HELP` search, it searches the file `cgelp2.hlp` (if it exists). The file is produced by using the *detex(1)* program on the LaTeX source file for this chapter.

?APROPOS help

There are two additional short help facilities in **cgelp2**. At the <CMD> level, typing ?command-prefix will list all top level comamnds which begin with that prefix. Typing `?APROPOS` keyword will list all top level commands which have that *keyword* somewhere in their one-line description. For example,

```
20<CMD>: ??
?<partial CMD> - list commands which start with the partial CMD.
?APROPOS <phrase> - list commands which contains this phrase.

21<CMD>: ?set
SET ACcession file name - change the default 'gel.id' name.
SET ANnotation - spot feature annotation.
SET CAlibration - for density/unit-area, file for(pIe,MW) as fct of (X,Y).
SET CLasses - Define gel experimental class partition.
SET DAtabase file - define or access CGL data paged database.
SET DEnsity mode - in Abs(D'), Uncor, Percent, Ratio, Vol, LSQ, CPM units.
SET DIsplay - plotting device to TTY, 4012, XWIND, xxxxPLOT.
SET FIelds - Set the list of accession fields desired for gel labeling.
SET FOreign spot map - define Rspot<==>Fspot protein name mapping.
SET Gel subset - define/operate on gel subsets.
SET LAbel - Set the 'Label' code to (S,P,A,U,E,C) used in searching.
SET LEast squares - density normalization calibration to Rgel.
SET MOre - toggle 'more' style output switch for the terminal.
SET NAmes of gels - change the alternative gel names.
SET PArameters subset - define/operate on parameters subsets.
SET PRefilter limits - define PREFILTER limits from X-window dialog form.
SET RAtio - compute Ratio-Mode Rspots spot mean density normalization.
SET REgion - of pIe and MW subregion in the Rgel to restrict CGL DB.
SET RGel - Set the name of the Rgel used in searching.
SET SPot view - define 'view' of a spot's features to dump on SAS/SPSS/etc.
SET SRl subset - define/operate on Search Results List subsets.
SET STatistics limits - Set statistics limits for use in searching.
SET SUbcmd switches - parse and change specified <SUBCMD> switches
SET Working gels - Define working set of gels from CGL databas

22<CMD>: ?apropos srl
SET ANnotation - Rspots&Fspots features - Edit, List, Map<==>SRL sets.
SET SRl subset - define/operate on Search Results List subsets.
/SRL:n - filter operation by SRLSS[n] if appropriate.
```

```
23<CMD>: ?apropos gel
CGEL' - create a C-gel' to estimate a set of replicate gels
DENdrogram - plot (.ugf) of gels=fct(SRL) or SRL=fct(expr-profiles).
EXIt - CGELP2 to monitor and save PCG database file for later use.
Gels - gels lists names, total densities, study of current gels.
HISTORy - list CGELP2 history commands.
INFormation - print the CGELP2 general information message.
*REMove - a gel from the CGL paged database.
SET ACcession file name - change the default 'gel.id' name.
SET CLasses - Define gel experimental class partition.
SET FIelds - Set the list of accession fields desired for gel labeling.
SET Gel subset - define/operate on gel subsets.
SET LEast squares - density normalization calibration to Rgel.
SET NAmes of gels - change the alternative gel names.
SET REgion - of pIe and MW subregion in the Rgel to restrict CGL DB.
SET RGel - Set the name of the Rgel used in searching.
SET Working gels - Define working set of gels from CGL database.
VAlid landmarks - list the valid landmarks for each gel in a table.
/ERspot - with CREATE, it will include non-Rgel US as Rgel EP spots.
/SINgle - gel for Cgel' estimate if appropriate.
/SOrt - SPSS or SAS output file by gel number rather than density.
```

<CMD> HIstogram

Compute histograms of functions of Rspot sets. Then plot them on either an X-window, the user's graphics terminal or into a plot file. An optional UGF plot file can also be generated. These plot files can be printed using the `plotn` program. It prints the following additional prompt. Normally you would use `SET DISPLAY` to define the plot or display type to be used (as well as associated plot switches: `/LOGPLOT`, `/MWprojection`, `/PIEprojection`, `/TITLE:text`, `/UGFlabel`).

simulated 1D gel projections

A response to a typical prompt follows for a printed (TTY) histogram:

```
77<CMD>: HISTOGRAM
Histogram [can restrict to function of /MW or /pIe] of
  'A'rea spot density
  'D'ensity mean Rspot integrated density
  'I'ndividual spot integrated density
  'L' DL spot pair distance from center of a pair to landmark
  'M'ax OD of any pixel in a spot
  'N'umber gels present in the R-spot set
  'O'D difference of spots as measured by (maxD-minD)
  'P' DP spot pairing distance between spots in a pair
  'V'ariation as (std dev/mean) R-spot set density
  ?: Number of gels/Rset<CR>
```

It is possible to create *synthetic 1D gel density profile* plots using the `HISTOGRAM` command by integrating in the MW or pIe axes. The `/pIe` switch creates a projection of the isoelectric direction while `/MW` creates a projection of the molecular weight.

```
202<CMD>: HIST//DENSITY/MW////
GLOBAL CMD SWITCHES:
Histogram [can restrict to fct of /MW or /pIe] of
    'A'rea spot density
    'D' - Rspot set mean integrated density.
    'I'ndividual spot integrated density
    'L' - DL spot pair distance from pair center to landmark
    'M'ax OD of any pixel in a spot
    'N'umber gels present in the R-spot set
    'O'D difference of spots as measured by (maxD-minD)
    'P' - DP spot pairing distance between spots in a pair
    'V'ariation as (std dev/mean) R-spot set density
 ?: DENSITY/MW
 Range[8.20 ratio D' : 49.70 ratio D']
 MIN:MAX[18.00 ratio D' : 29.50 ratio D']
 mode = 29.50 ratio D'
 median = 26.60 ratio D'
 mean = 27.30 ratio D' +/-8.20 ratio D'
[07:13:03PM] Real TIME =00:01:22  CPU TIME =00:01:13,  89.02%
203<CMD>:
```

If the display is set to LASER, then

```
42<CMD>: HISTOGRAM
Histogram [can restrict to function of /MW or /pIe] of
    'A'rea spot density
    'D'ensity mean Rspot integrated density
    'I'ndividual spot integrated density
    'L' DL spot pair distance from pair center to landmark
    'M'ax OD of any pixel in a spot
    'N'umber gels present in the R-spot set
    'O'D difference of spots as measured by (maxD-minD)
    'P' DP spot pairing distance between spots in a pair
    'V'ariation as (std dev/mean) R-spot set density
 ?: Density
File: /home/joeUser/gellab/gen/000016.ugf 06/12/1989, 08:14:24AM
PCGL database: /home/joeUser/gellab/pcg/ts3pcg.pcg
Pairing labels used are [PSC]
 Printing plot file on laser printer
 plotn -display:4010 ~joeUser/gellab/gen/000017.ugf | tek2psG | lpr -Plaser
Plot file: /home/joeUser/gellab/gen/000017.ugf
 Range[0.00 LSQ D' : 196.00 LSQ D']
 MIN:MAX[30.00 LSQ D' : 0.00 LSQ D']
 mode = 0.00 LSQ D'
 median = 5.00 LSQ D'
 mean = 18.00 LSQ D' +/-196.00 LSQ D'
 # Rspots= 721 # spots= 4183
[08:15:24AM] Real TIME =00:01:18  CPU TIME =00:00:53,  67.95%
```

Figure 3.5 shows a sample plot output.

Figure 3.5. Sample **cgelp2** `HISTOGRAM` plot. The plot file was ploted using the `LASER` display option set with `SET DISPLAY`.

<CMD> HISTORy

cgelp2 <CMD> *history*

List the **cgelp2** `CMD` history commands. See Section 3.6, page 173 for more details on using **cgelp2** command history.

```
4<CMD>: history
Illegal HISTORY command

<CMD> COMMAND HISTORY
---------------------
   !?        - list last 20 command history entries.
   !??       - list entire command history entries with [more?].
   !!        - execute the previous command again.
   !nnn      - execute history command nnn again.
   !XXX      - execute history command matching letters XXX again.
   !p^oldPattern^newPattern   - execute history command p where
                p is ! or nnn or XXX after substituting newPattern
                for all occurrences of oldPattern.
```

```
!*        - clear the current history list.
!+file    - append history list from a file 'file.hrc'.
!-file    - write history list into a file 'file.hrc'.
!@n1,n2:f - redo history numbers in range [n1:n2], opt. ':filename'.
!_nnn     - delete nnn'th or last history entry.
!/CCC     - if CCC then extend last history entry else terminate it.
```

<CMD> Inquire

Interrogate the PCG DB for particular spots. The set of `INQUIRE` subcommands, listed in Table 3.7, page 177 in Section 3.3.8 are used for inquiry of the database. It contains parametric, non-parametric and other types of searches as well as commands to present different views of Rspot set data. The various searches performed under `INQUIRE` terminate by printing a summary line of the form

spot searches

```
Found 238 Rspots, mean density+-'sd/mn' = 2.85+-1.78
```

to indicate the number of Rspot sets found meeting the search criteria. The names of Rspot sets found in a search are saved in the *Search Results List* (SRL). The SRL may be listed using the `PRINT` subcommand. It is used to specify a list of Rspots on which to operate for some of the following subcommands as well as other commands. The SRL may be saved in (restored from) SRL subset sets (see `SET SRL SUBSETS` command). A search operation, once started, may be aborted by typing `CONTROL/C`. The SRL will contain the list of Rpots found up to the point the search was aborted.

SRL creation

Several non-search subcommands, `Print, Change histogram, Order Rspot sets, OExpression Profile` are part of the `INQUIRE` command set. They are used to present a different view of the current SRL. The constraint based tests `Missing-class`, `%-search-above-threshold`, `Expression-profile`, `least-squares-fit`, and `coordinate-pair` as well as several others are non-statistical searches which may find interesting spot differences which the standard statistical density distribution based tests do not. This is discussed here in detail and in Section 5.1 and [LemP83a].

views of SRL

Note that appending `/FILE` to any `INQUIRE` subcommand will cause its output to be sent to an *.inq* file to be specified by the user. This option is useful if you wish to document a particular search or to export the particular search results to other statistical packages. For example, the **dendrogram** program can use the `ORDER TABLE` generated *.inq* file as input for clustering a set of Rspots as a function of the expression profile for the set of gels in different experimental profiles.

`/FILE` *suboption*

Another switch, `/LOG`, may also be appended to a subcommand. It causes density be transformed to $log(1+density)$. This transformation is useful for creating a view which minimizes differences in the variance between classes.

`/LOG` *suboption*

Density for all inquire commands is interpreted as the current density mode which is defined using the `SET DENSITY MODE` command. If you set the global

`/MEDIAN` *suboption*

switch /MEDIAN then the median rather than the mean is computed and used in all statistical test calculations.

/SRLSS[n] *option*

Normally the search is performed on all Rspot sets meeting the prefilter criteria. It may be further restricted. The top level /SRLSS[n] switch modifier can be used to restrict (e.g. INQUIRE/SRLSS[n]) the inquiry to only those spots in the specified SRL subset n.

Prefilter

prefilter

In any INQUIRE operation in which an Rspot set is to be used or tested, a candidate Rspot set must pass a preliminary set of tests or filters collectively called the *PREFILTER* in order for the specified operation to be performed on it. These prefilter tests include:

1. The pairing label of gels to be considered in the Rspot set must be one of those specified for the prefilter by SET LABEL in order for that particular gel′s spot to be considered. To search only that part of the database which contains the Rgel, PUS (PP+US+SP) is specified. To search only the eRspot sets, UX (US+Xtended) is specified. To search both parts of the PCG DB, the PUSX would be specified. To include EP spots, add E to the search label. To count EP spots as spots with zero density then invoke the operation as INQUIRE/EPspots.

2. Each gel to be considered in the Rspot set must be in the working set of gels defined for the prefilter using the SET WORKING command. Working sets of gels may be saved (and later used by SET WORKING GELS) using the SET GEL SUBSET command.

3. The prefilter statistical limits are applied to the Rspot set as a whole in operations which are class independent (such as INDEX search, etc) or to each subset of gels defined under a class for a Rspot set when class dependent operations are performed (such as t-test, etc). The limits are defined using the SET STATISTICS command. One of the limits includes # gels/class (in which that spot is found) which is useful in defining how robust a set of gels must be to be counted. The class partitions may be defined using the SET CLASSES command.

4. The Rgel spot (if present) must be in the MW-pIe window region defined for the prefilter using the SET REGION command. Only the Rgel position is checked.

5. The Rspots which are eligible for a search can be further restricted by appending global level switch /SRL:n to the INQUIRE command. Then, for some SRL subset n, the prefilter will add the additional constraint that any potential Rspot also be in the SRL subset n.

Search failure analysis

Therefore, if in a search through the PCG DB the features of the Rspot set do not meet the above prefilter criteria, then the specified test is not performed. Section 5.1.10, page 527 discusses the prefilter and how to determine values for setting it. Note that you can determine *why* the prefilter rejected an Rspot set by enabling reporting using the global `/EXplain` switch. This is illustrated with the following examples. `DBUG:04000` bit enables the explanation on a Rspot by Rspot basis. Note that for a single class test such as `INDEX`, it explains the Rspot sets as a whole, whereas for a multiclass test such as the `T-TEST`, it explains each of the sub distributions for each Rspot set.

`/EXPLAIN` *option*

```
3<CMD>: INQ/EXPLAIN
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%).
 HELP to list CMDS?: INDEX<CR>
Statistical limits search

Rspot[   1] mnXYDA=(302,172,3.17,29)+-(6.00,8.12,5.28,25) CVD=1.66 #G=18
Rspot[2] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statistics
Rspot[3] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statistics
Rspot[4] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statistics
Rspot[5] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statistics
Rspot[6] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statistics
Rspot[   7] mnXYDA=(302,185,5.28,34)+-(3.00,3.00,6.72,29) CVD=1.27 #G=19
Rspot[8] FAILED-PREFILTER failCode= 10 = Failed mean AREA statistics
Rspot[   9] mnXYDA=(302,191,27.72,88)+-(2.00,1.73,15.57,26) CVD=0.56 #G=14
Rspot[10] FAILED-PREFILTER failCode= 10 = Failed mean AREA statistics
Rspot[  11] mnXYDA=(304,202,25.35,76)+-(1.00,2.45,15.65,28) CVD=0.62 #G=12
Rspot[  12] mnXYDA=(283,204,3.65,35)+-(2.45,2.65,5.15,29) CVD=1.41 #G=15
Rspot[  13] mnXYDA=(295,206,3.51,33)+-(3.87,3.46,4.26,28) CVD=1.21 #G=16
Rspot[  14] mnXYDA=(342,149,10.29,36)+-(10.30,15.03,15.32,42) CVD=1.49 #G=22
Rspot[15] FAILED-PREFILTER failCode= 10 = Failed mean AREA statistics
Rspot[  16] mnXYDA=(375,82,6.32,25)+-(3.87,3.00,18.08,37) CVD=2.86 #G=17
Rspot[17] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statistics
Rspot[18] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statistics
        .
        .
        .
Rspot[680] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statistics

Found 238 Rspots, mean density+-'sd/mn' = 2.85+-1.78

PREFILTER failCode Histogram
Freq=476  failCode = 0 = PASSES ALL TESTS!
Freq= 0   failCode = 1 = Rspot set is NULL
Freq=38   failCode = 2 = No gels in W.S. meet label, DP, DL or ODdf limits
Freq= 0   failCode = 3 = # gels outside of #-gels-required limits
Freq= 0   failCode = 4 = # gels meeting relative (dx,dy) limits == 0
Freq= 0   failCode = 5 = Rspot set Rgel position outside of pIe-MW region
Freq= 0   failCode = 6 = No gels in W.S. meet label limits
Freq= 0   failCode = 7 = No gels in W.S. meet DP limits
```

```
Freq= 0   failCode = 8 = No gels in W.S. meet DL limits
Freq= 0   failCode = 9 = No gels in W.S. meet ODdf limits
Freq=18   failCode = 10 = Failed mean AREA statistics
Freq=381  failCode = 11 = Failed mean DENSITY statistics
Freq= 5   failCode = 12 = Failed COV of AREA statistics
Freq= 0   failCode = 13 = Failed COV of DENSITY statistics
Freq=238  failCode = 14 = Failed Statistical Test
Freq=442  failCode = 15 = Passed Statistical Test

4<CMD>: INQ/EXPLAIN
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%).
  HELP to list CMDS?: TB-TEST<CR>
Std t-/Behrens-Fisher t-Test (using F-stat) class search at 0.90 significance
Which two classes are to be compared?: ?<CR>
These are the current CLASS NAMES
---------------------------------
  Class# 1 = AML
  Class# 2 = ALL
  Class# 3 = CLL
  Class# 4 = HCL
  Class# 5 = HL-60
Which two classes are to be compared?: 1,2<CR>

Rspot[1] [Class:1] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statis.
Rspot[2] [Class:1] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statis.
Rspot[3] [Class:1] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statis.
Rspot[3] [Class:2] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statis.
Rspot[4] [Class:1] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statis.
Rspot[4] [Class:2] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statis.
Rspot[5] [Class:1] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statis.
Rspot[5] [Class:2] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statis.
Rspot[6] [Class:1] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statis.
Rspot[6] [Class:2] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statis.
Rspot[8] [Class:2] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statis.
Rspot[10] [Class:2] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statis.
Rspot[ 11] mnXYDA=(304,202,25.35,76)+-(1.00,2.45,15.65,28) CVD=0.62 #G=12
  [11] (m1,m2)=(22.12,44.79), m2/m1=2.02 varPooled=93.14 t-T=2.88 f=9.51
Rspot[12] [Class:1] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statis.
Rspot[13] [Class:2] FAILED-PREFILTER failCode= 10 = Failed mean AREA statis.
Rspot[15] [Class:1] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statis.
Rspot[16] [Class:1] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statis.
Rspot[17] [Class:1] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statis.
Rspot[17] [Class:2] FAILED-PREFILTER failCode= 11 = Failed mean DENSITY statis.
        .
        .
        .
```

<SUBCMD> INQUIRE: Print

Print data from a Rspot set `i`, `'*'` or `'$'` (search results list), `'S'` to list SRL #s. The '*' indicates using *all* of the search results list (SRL) found with the last search.

The 'S' prints only the Rspot set numbers which are in the SRL whereas the '*' prints the Rspot sets themselves which are in the SRL.

Typical Rspot set output is illustrated with several examples. Note that the first line includes: the Rspot #; the estimated (x,y) Cgel′ centroid in the Rgel and mean density and area; ± the standard deviations of these same features; the coefficient of variation of the density as well as the # of gels meeting sizing criteria. The table entries include: ACC# and segmenter CC# which taken together correspond to a particular spot entry; density in the current mode; spot area; maximum spot OD; raw spot density D' (corrected for background); pairing label (see **cmpgl2** [LemP81b]); landmark set name to which it belongs; DP and DL (see **cmpgl2** [LemP81b]); relative (x, y) offset from the spot to the landmark; the absolute (x, y) of the spot in the corresponding gel image; and spot OD difference ($maxOD-meanBackground$). The following example is in percent D' of total gel D' (`% density mode`). A short form of the command might be `INQUIRE//Print//2//` to print Rspot[2], etc. It has a table entry `%Dens` mode.

% density mode

```
236<CMD>: INQUIRE
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%).
 HELP to list CMDS?: PRINT<CR>
Print Rspot set #?[#,s,$,0 to exit]?: 2

Rspot[   2] mnXYDA=(207,240, .09,30) +-( 1.90, 1.30, .04,7) CVD= .47 #G=10
 ACC#[Index] C  %Dens  area pkOD  D'   Lbl LM  DP  DL  Dx   Dy  Xabs Yabs ODdf
------------ - ------- ---- ----  ---- --- -- ---- -- ---- ---- ---- ---- ----
0260.2[ 417] 2   .132%   38  .58   9.0  PP A   4.5 25 ( -4,-25) (208,245)  .32
0250.2[ 363] 1   .161%   36  .54   6.3  PP A   1.0 23 (  0,-23) (208,240)  .25
0255.2[ 295] 1   .112%   32  .36   3.5  PP A   4.2 23 ( -3,-20) (186,239)  .17
0251.2[ 349] 1   .111%   33  .47   5.4  PP A   1.0 23 ( -1,-23) (113,243)  .17
0252.2[ 284] 1   .112%   36  .41   4.1  PP A   2.0 23 ( -2,-23) (194,258)  .14
0261.2[ 356] 2   .074%   37  .43   4.4  PP A   3.0 23 ( -3,-23) (223,219)  .22
0254.2[ 316] 1   .068%   21  .36   2.3  PP A   1.4 23 (  1,-22) (178,261)  .10
0264.2[ 399] 2   .053%   19  .49   3.2  PP A   1.4 23 (  1,-22) (170,240)  .20
0257.2[ 375] 1   .035%   23  .29   1.7  PP A   2.2 24 (  2,-24) (232,215)  .10
0256.2[ 263] 1   .028%   20  .32   1.2  PP A   1.4 24 ( -1,-24) (214,260)  .10
Print Rspot set #?[#,s,$,0 to exit]?:
```

The first line of the Rspot set table is a summary of the entire Rspot set treated as a single class. The `mnXYDA` is the mean spot (x, y) position (in the Rgel domain), mean density (in the current normalization mode) and mean area. The `CVD` is the coefficient of variation of density. The `#G` is the number of gels visible at the current prefilter settings. Depending on the density mode (set with `SET DENSITY MODE`) a Rspot set can be printed in various formats. These are enumerated below for the same Rspot set.

`mnXYDA` *Rspot summary*

The Rspot set printed using Least Squares Ratio density mode is calibrated by `SET LEAST SQUARES` calibration. This normalization maps D′ of a non-Rgel gel to

least-square density mode

the domain of D′ for the Rgel and has a table entry `LDens`.

```
Rspot[   2] mnXYDA=(207,240, 3.94,30) +-( 1.90, 1.30, 2.09,7) CVD= .53 #G=10
 ACC#[Index] C  LDens  area pkOD  D'   Lbl LM  DP  DL  Dx   Dy  Xabs Yabs ODdf
------------ - ------- ---- ----  ---- --- -- ---- -- ---- ---- ---- ---- ----
0260.2[ 417] 2   8.32L   38  .58   9.0  PP A   4.5 25 ( -4,-25) (208,245)  .32
0250.2[ 363] 1   6.30L   36  .54   6.3  PP A   1.0 23 (  0,-23) (208,240)  .25
0255.2[ 295] 1   4.81L   32  .36   3.5  PP A   4.2 23 ( -3,-20) (186,239)  .17
0251.2[ 349] 1   4.55L   33  .47   5.4  PP A   1.0 23 ( -1,-23) (113,243)  .17
0252.2[ 284] 1   4.19L   36  .41   4.1  PP A   2.0 23 ( -2,-23) (194,258)  .14
0261.2[ 356] 2   3.35L   37  .43   4.4  PP A   3.0 23 ( -3,-23) (223,219)  .22
0254.2[ 316] 1   2.71L   21  .36   2.3  PP A   1.4 23 (  1,-22) (178,261)  .10
0264.2[ 399] 2   2.46L   19  .49   3.2  PP A   1.4 23 (  1,-22) (170,240)  .20
0257.2[ 375] 1   1.47L   23  .29   1.7  PP A   2.2 24 (  2,-24) (232,215)  .10
0256.2[ 263] 1   1.20L   20  .32   1.2  PP A   1.4 24 ( -1,-24) (214,260)  .10
```

ratio-list density mode

The Rspot set in ratio mode where the ratio of a spot is computed by dividing its D′ by the sum of D′ for that gel for a set of normalization spots (see `SET RATIO LIST`) meeting some particular criteria. It has a table entry `RDens`.

```
Rspot[   2] mnXYDA=(207,240, .34,30)+-( 1.90, 1.30, .16,7) CVD= .47 #G=10
 ACC#[Index] C  RDens  area pkOD  D'   Lbl LM  DP  DL  Dx   Dy  Xabs Yabs ODdf
------------ - ------- ---- ----  ---- --- -- ---- -- ---- ---- ---- ---- ----
0260.2[ 417] 2    .69R   38  .58   9.0  PP A   4.5 25 ( -4,-25) (208,245)  .32
0250.2[ 363] 1    .47R   36  .54   6.3  PP A   1.0 23 (  0,-23) (208,240)  .25
0255.2[ 295] 1    .43R   32  .36   3.5  PP A   4.2 23 ( -3,-20) (186,239)  .17
0251.2[ 349] 1    .43R   33  .47   5.4  PP A   1.0 23 ( -1,-23) (113,243)  .17
0252.2[ 284] 1    .38R   36  .41   4.1  PP A   2.0 23 ( -2,-23) (194,258)  .14
0261.2[ 356] 2    .30R   37  .43   4.4  PP A   3.0 23 ( -3,-23) (223,219)  .22
0254.2[ 316] 1    .25R   21  .36   2.3  PP A   1.4 23 (  1,-22) (178,261)  .10
0264.2[ 399] 2    .24R   19  .49   3.2  PP A   1.4 23 (  1,-22) (170,240)  .20
0257.2[ 375] 1    .15R   23  .29   1.7  PP A   2.2 24 (  2,-24) (232,215)  .10
0256.2[ 263] 1    .12R   20  .32   1.2  PP A   1.4 24 ( -1,-24) (214,260)  .10
```

The same mode is printed here but the `/LOG` option is specified to the Print subcommand. The `/LOG` option computes density as $log(1 + density)$.

```
Rspot[   2] mnXYDA=(207,240, .29,30)+-( 1.90, 1.30, .12,7) CVD= .41 #G=10
 ACC#[Index] C  RDens  area pkOD  D'   Lbl LM  DP  DL  Dx   Dy  Xabs Yabs ODdf
------------ - ------- ---- ----  ---- --- -- ---- -- ---- ---- ---- ---- ----
0260.2[ 417] 2    .52R   38  .58   9.0  PP A   4.5 25 ( -4,-25) (208,245)  .32
0250.2[ 363] 1    .38R   36  .54   6.3  PP A   1.0 23 (  0,-23) (208,240)  .25
0255.2[ 295] 1    .36R   32  .36   3.5  PP A   4.2 23 ( -3,-20) (186,239)  .17
0251.2[ 349] 1    .35R   33  .47   5.4  PP A   1.0 23 ( -1,-23) (113,243)  .17
0252.2[ 284] 1    .32R   36  .41   4.1  PP A   2.0 23 ( -2,-23) (194,258)  .14
0261.2[ 356] 2    .26R   37  .43   4.4  PP A   3.0 23 ( -3,-23) (223,219)  .22
0254.2[ 316] 1    .22R   21  .36   2.3  PP A   1.4 23 (  1,-22) (178,261)  .10
0264.2[ 399] 2    .21R   19  .49   3.2  PP A   1.4 23 (  1,-22) (170,240)  .20
0257.2[ 375] 1    .14R   23  .29   1.7  PP A   2.2 24 (  2,-24) (232,215)  .10
0256.2[ 263] 1    .12R   20  .32   1.2  PP A   1.4 24 ( -1,-24) (214,260)  .10
```

The Rspot set printed using Absolute (raw D′ - i.e. total spot density in OD corrected for gel background density) density format. It has a table entry 'Dens. *absolute density' mode*

```
Rspot[   2] mnXYDA=(207,240, 4.11,30) +-( 1.90, 1.30, 2.22,7) CVD= .54 #G=10
 ACC#[Index] C  'Dens  area pkOD  D'   Lbl LM  DP  DL  Dx   Dy  Xabs Yabs ODdf
------------ - ------- ---- ----  ---- --- -- ---- -- ---- ---- ---- ---- ----
0260.2[ 417] 2   9.00'   38  .58   9.0  PP A   4.5 25 ( -4,-25) (208,245)  .32
0250.2[ 363] 1   6.30'   36  .54   6.3  PP A   1.0 23 (  0,-23) (208,240)  .25
0255.2[ 295] 1   3.50'   32  .36   3.5  PP A   4.2 23 ( -3,-20) (186,239)  .17
0251.2[ 349] 1   5.40'   33  .47   5.4  PP A   1.0 23 ( -1,-23) (113,243)  .17
0252.2[ 284] 1   4.10'   36  .41   4.1  PP A   2.0 23 ( -2,-23) (194,258)  .14
0261.2[ 356] 2   4.40'   37  .43   4.4  PP A   3.0 23 ( -3,-23) (223,219)  .22
0254.2[ 316] 1   2.30'   21  .36   2.3  PP A   1.4 23 (  1,-22) (178,261)  .10
0264.2[ 399] 2   3.20'   19  .49   3.2  PP A   1.4 23 (  1,-22) (170,240)  .20
0257.2[ 375] 1   1.70'   23  .29   1.7  PP A   2.2 24 (  2,-24) (232,215)  .10
0256.2[ 263] 1   1.20'   20  .32   1.2  PP A   1.4 24 ( -1,-24) (214,260)  .10
```

The Rspot set printed with *Volume* density format where the volume density estimate [AndN81] is computed using the formula:

$$\sqrt{4\pi} * dMax * Sx * Sy,$$

Where: $dMax$ is the highest density pixel value in the spot, Sx and Sy are the Gaussian estimates of the spot widths computed by the segmenter **sg2gii** (cf. Section 3.18. It has a table entry VDens. *volume density mode*

```
Rspot[   2] mnXYDA=(207,240, 3.12,30) +-( 1.90, 1.30, 1.44,7) CVD= .46 #G=10
 ACC#[Index] C  VDens  area pkOD  D'   Lbl LM  DP  DL  Dx   Dy  Xabs Yabs ODdf
------------ - ------- ---- ----  ---- --- -- ---- -- ---- ---- ---- ---- ----
0260.2[ 417] 2   6.00V   38  .58   9.0  PP A   4.5 25 ( -4,-25) (208,245)  .32
0250.2[ 363] 1   4.65V   36  .54   6.3  PP A   1.0 23 (  0,-23) (208,240)  .25
0255.2[ 295] 1   3.39V   32  .36   3.5  PP A   4.2 23 ( -3,-20) (186,239)  .17
0251.2[ 349] 1   3.06V   33  .47   5.4  PP A   1.0 23 ( -1,-23) (113,243)  .17
0252.2[ 284] 1   2.92V   36  .41   4.1  PP A   2.0 23 ( -2,-23) (194,258)  .14
0261.2[ 356] 2   4.26V   37  .43   4.4  PP A   3.0 23 ( -3,-23) (223,219)  .22
0254.2[ 316] 1   1.32V   21  .36   2.3  PP A   1.4 23 (  1,-22) (178,261)  .10
0264.2[ 399] 2   2.50V   19  .49   3.2  PP A   1.4 23 (  1,-22) (170,240)  .20
0257.2[ 375] 1   1.66V   23  .29   1.7  PP A   2.2 24 (  2,-24) (232,215)  .10
0256.2[ 263] 1   1.40V   20  .32   1.2  PP A   1.4 24 ( -1,-24) (214,260)  .10
```

<SUBCMD> INQUIRE: Landmark search

Search for Rspot sets which are landmarks. It saves the names of Rspot sets which meet the test criteria in the SRL. It finds the names of the corresponding Rspot sets in the database and prints the following summary as well as saving the Rspot names in the search results list: *landmarks*

```
198<CMD>: INQUIRE
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%).
  HELP to list CMDS?: LANDMARK SEARCH<CR>
```

```
Landmark set search
LM[ A ]=Rspot[   8]
LM[ B ]=Rspot[  55]
LM[ C ]=Rspot[  89]
LM[ D ]=Rspot[  98]
LM[ E ]=Rspot[ 105]
LM[ F ]=Rspot[ 141]
LM[ G ]=Rspot[ 150]
LM[ H ]=Rspot[ 173]
LM[ I ]=Rspot[ 208]
LM[ J ]=Rspot[ 233]
LM[ K ]=Rspot[ 240]
LM[ L ]=Rspot[ 292]
LM[ M ]=Rspot[ 331]
LM[ N ]=Rspot[ 465]
LM[ O ]=Rspot[ 528]
LM[ P ]=Rspot[ 576]
LM[ Q ]=Rspot[ 597]
LM[ R ]=Rspot[ 619]
LM[ S ]=Rspot[ 630]
LM[ T ]=Rspot[ 661]
LM[ U ]=Rspot[ 693]
LM[ V ]=Rspot[ 723]
Found 22 Rspots, mean density+-'sd/mn' = 0.78+-0.32
```

<SUBCMD> INQUIRE: Index search

prefilter search

Search for Rspot sets meeting the current prefilter statistical limits. It saves the names of Rspot sets which meet the test criteria in the SRL. The Rspot set for statistical purposes is treated as a single class with single mean and standard deviation, etc. The index search just prints the first line of each Rspot set meeting the statistical criteria (set with `SET STATISTICS`). The following output illustrates this command where the coefficient of variation of density limit was set to only find spots with the prefilter parameter coefficient of variation of density CVD > 0.5.

```
199<CMD>: INQUIRE
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%)
 HELP to list CMDS?: INDEX SEARCH<CR>
        Statistical limits search
Rspot[   1] mnXYDA=(302,174,5.03,38)+-(8.00,11.58,6.74,21) CVD=1.34 #G=9
Rspot[   2] mnXYDA=(305,169,3.87,35)+-(9.85,13.19,5.35,25) CVD=1.38 #G=10
Rspot[   7] mnXYDA=(303,185,5.19,35)+-(1.41,0.00,4.54,15) CVD=0.88 #G=9
Rspot[   9] mnXYDA=(302,191,31.35,95)+-(0.00,0.00,13.63,22) CVD=0.43 #G=12
Rspot[  10] mnXYDA=(314,191,7.95,30)+-(2.24,2.00,14.66,24) CVD=1.84 #G=8
         .
         .
         .
Found 287 Rspots, mean density+-'sd/mn' = 2.19+-1.43
```

<SUBCMD> INQUIRE: Search

Search for Rspot sets meeting statistical limits like `Index Search` above but only print the Rspot set number. It saves the names of Rspot sets which meet the test criteria in the SRL. The Rspot set for statistical purposes is treated as a single class with single mean and standard deviation, etc. The search subcommand behaves as the index search command but only prints the Rspot set numbers meeting the statistical criteria. The following output illustrates this command.

```
301<CMD>: INQUIRE
 Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%)
  HELP to list CMDS?: SEARCH<CR>
  Statistical limits search
   3 6 8 ...
```

<SUBCMD> INQUIRE: Missing class

Search for Rspot sets where one of the two specified classes is completely missing while the other class meets the statistical sizing criteria set with `SET STATISTICS` number of gels per class PREFILTER. It saves the names of Rspot sets which meet the test criteria in the SRL. Note that you can set the minimum number of gels/experimental class be > 1 (i.e. some minimum number) to decrease the number of false negatives (spots not found) due to noise. You may further restrict the search to only find Rspots which occur in a specific class *c1* when you are comparing classes *c1,c2*. Specify this using the */OPTION:c1* switch to `INQUIRE` (eg. `INQUIRE/OPTION:`*c1*).

missing-class search

Normally, you set the PREFILTER variable "minimum # of gels/Class" for a class to be considered present. I.e. set it to 5 for say 10 gels or whatever. If an Rspot set has say 2 gels in class 1, then it effectively says (for the MISSING CLASS test) that class 1 is NOT present. However, in reality it is. You then need to check the mosaic to see if the 2 gels are bogus and the missing change is real.

eg. Rspot[j] has
Class 1: 3 gels out of 15 class 2 gels.
Class 2: 11 gels out of 15 class 2 gels.

- if "minimum # of gels/Class" is set to 5, then Rspot[j] is considered a MISSING CLASS Rspot.
- if "minimum # of gels/Class" is set to 2, then Rspot[j] is *not* considered a MISSING CLASS Rspot.

To test it with *no* gels present in the missing class, then set "minimum # of gels/Class" to 0. However, you lose the noise immunity this provides.

Some typical output might be:

```
302<CMD>: INQUIRE
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%)
 HELP to list CMDS?: MISSING CLASS<CR>
 Which two classes are to be compared?: 1,2<CR>

Rspot[4] mnXYDA=(234,244,.21,32)+-(1.00,.00,.13,15) CVD= .61 #G=2
  n1=2, n2=0
Rspot[9] mnXYDA=(233,262,.21,16)+-(.00,.00,.00,0) CVD= .00 #G=1
  n1=0, n2=1
Rspot[13] mnXYDA=(255,259,.30,25)+-(1.50,1.00,.18,11) CVD= .59 #G=2
  n1=2, n2=0
Rspot[14] mnXYDA=(261,260, .20,22)+-( 3.00, 1.00, .04,4) CVD= .21 #G=2
  n1=2, n2=0
        .
        .
        .
```

Analysis of missing class problem - subtle statistics problem

There is a suble way in which the MISSING CLASS test should be used. This operational problem is best described using an example and then the solution is suggested. The criteria it uses is:

$$((c_{1ok} \wedge \sim c_{2ok}) \vee (c_{2ok} \wedge \sim c_{1ok}))$$

Where:

- c_{1ok} (c_{2ok}) is the state of the PREFILTER for the class 1 (class 2) subset of the gels.
- `n1` (`n2`) are the number of gels of class 1 (class 2) subset of the gels.

So it is important to remember that not only must $n_j > 0$ but it must meet the PREFILTER *min # gels/class* criteria.

The following are SRLSS generated by setting the prefilter to 10/class for SRLSS[12] and 5/class for SRLSS[13]. Since 5/class would appear to be less restrictive, one would expect the spots in SRLSS[13] to be in SRLSS[12] as well! However, that is *not* strictly true as we shall see.

```
[12] MISSING CLASS SEARCH - MIN # GELS/CLASS= 10 |49|  6-23-93 15:27
[13] MISSING CLASS SEARCH - MIN # GELS/CLASS= 5 |22|  6-23-93 15:31
[14] INTERSECTION (10/CLASS, 5/CLASS) FOR MISSING CLASS TESTS |2|  6-24-93 6:5
```

Then, the intersection of the two searchs to find common spots and put these in SRLSS[14]. There were only 2 Rspots.

The contents of these SRL subsets are:

```
Subset name (or # Or # range Or <ALL> Or <LAST> <LOS> Or SRL[i])?: 12-14
```

```
Set #12<<<MISSING CLASS SEARCH - MIN # GELS/CLASS= 10>>>= 24 177 179
 180 205 206 208 209 211 214 374 420 421 423 432 433 487 521 529 535
 560 576 586 596 597 614 707 732 784 790 813 817 819 821 822 824 961
 1115 1183 1185 1205 1208 1210 1213 1351 1425 1428 1431 1436

Set #13<<<MISSING CLASS SEARCH - MIN # GELS/CLASS= 5>>>= 207 429 430
 431 440 442 444 449 522 597 733 790 815 828 833 1212 1427 1434 1441
 1442 1443 1447

Set #14<<<INTERSECT. (10/CLASS, 5/CLASS) FOR MISSING CLASS TESTS>>>= 597 790
```

Some of the questions we can ask are:

Q1 Why are Rspots 597 790 present in both and the rest of SRLSS[13] not present?

Q2 Why are there Rspots in 5/class NOT in the 10/class?

Q3 Why are there Rspots in 10/class which are NOT in the less restrictive 5/class?

Q4 Why are there MORE Rspots in 10/class than in what we think is the less restrictive 5/class?

The analysis of the data helps explain this situation:

[Q1] If you look at Rspot[597] and R[790] you see they have > 10 gels/class in class 1 BUT < 5 gels/class for the other class:

```
Rspot[597] mnXYDA=(184.0,838.0,5639.42,91)+-(3.2,6.9,9518.34,81) CVD=1.69 #G=18
  18 gels (|class 1|=14, |class 2|=4).

Rspot[790] mnXYDA=(916.0,709.0,2230.64,52)+-(8.1,5.0,4261.80,28) CVD=1.91 #G=15
  15 gels (|class 1|=11, |class 2|=4).
```

[Q2] whereas all of the other SRLSS[13] Rspots do not have 10 gels/class! For example in Rspot[207] found only in SRLSS[13],

```
Rspot[207] mnXYDA=(922.0,491.0,7952.84,104)+-(2.0,2.4,5219.94,49) CVD=0.66 #G=9
  9 gels (|class 1|=5, |class 2|=4).
```

will meet the minimum criteria for 5/class but not for 10/class. All of the other Rspots found only in SRLSS[13] but not in SRLSS[12] are similar.

[Q3] Rspots in SRLSS[12] which are NOT found in the 5/class SRLSS[13] have >5 gels/class for BOTH classes.

```
Rspot[24] mnXYDA=(77.0,195.0,1808.19,50)+-(5.1,2.0,1737.46,21) CVD=0.96 #G=25
  25 gels (|class 1|=14, |class 2|=9).
```

[Q4] The reason there are MORE gels in 10/class than 5/class is, I think, accidental.

In conclusion, then you should probably ALWAYS titrate the MISSING CLASS test with n_1/class n_2/class (...???... nK/class) for $[n_1 < n_2 < ... < n_K]$ and then take the intersection. The intersection operation gets rid of the noise and leaves the Rspots which appear to be really changing.

<SUBCMD> INQUIRE: Upper-Lower-Missing class

missing-class search

Search for Rspot sets where one of the two specified classes has less than a lower bound number of gels/class and the other has more than the upper bound number of gels/class . The lower and upper bounds are specified using `SET STATISTICS` number of gels per class PREFILTER. It is similar to the `MISSING-CLASS` test which you might refer to for its general operation. Where it differs is in what we consider to be missing and present. Let N_l be the upper bound on the number of gels/class for gels in a class to be considered *missing* and N_u be the lower bound on the number of gels/class for gels in a class to be considered *present*. Then

- Missing gels do not have more than N_l gels/class
- Present gels have at least N_u gels/class.

This means Then the number of gels in $C_{missing}$ is $\leq N_l$, and the number of gels in $C_{present} \geq N_u$. The bounds are requested after you enter the two classes to be tested. You should probably leave the prefilter number of gels/class wide open since the test will restrict things itself.

```
Enter (Nlower:max# gels/missing, Nupper:min# gels/present) class?: 5,20
```

The test then performs the following test:

$$((c_{1ok} \wedge n_1 \geq N_u \wedge (\sim c_{2ok} \vee n_2 \leq N_l)) \vee (c_{2ok} \wedge n_2 \geq N_u \wedge (\sim c_{1ok} \vee n_1 \leq N_l)) \wedge \sim mustBeIn1stClass) \vee (n \tag{3.1}$$

The following is an sample search with the number of gels/class is set to [0:10000] so it is wide open.

```
203<CMD>: inq
Inquire cmds(B,CH,CO,D,E,F,FP,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,TP,T2,U,%).
 HELP to list CMDS?: u
Which two classes are to be compared?: 1,2
Enter (Nlower:max# gels/missing, Nupper:min# gels/present) class?: 5,10

Rspot[597] mnXYDA=(184.0,838.0,5639.42,91)+-(3.2,6.9,9518.34,81) CVD=1.69 #G=18
 [597] n1=14  n2=4
```

```
Rspot[790] mnXYDA=(916.0,709.0,2230.64,52)+-(8.1,5.0,4261.80,28) CVD=1.91 #G=15
 [790] n1=11  n2=4

Rspot[817] mnXYDA=(742.0,900.0,1282.00,42)+-(3.7,7.2,3432.99,23) CVD=2.68 #G=15
 [817] n2=10  n1=5

Rspot[1183] mnXYDA=(922.0,737.0,2141.83,58)+-(10.1,22.2,1934.39,28) CVD=0.90 #G=18
 [1183] n1=13  n2=5

Rspot[1213] mnXYDA=(932.0,864.0,23484.50,82)+-(7.1,15.7,33792.95,71) CVD=1.44 #G=17
 [1213] n1=12  n2=5

Found 5 Rspots, mean (sd/mn) +-'std-dev (sd/mn)'= 2.48+-1.32
```

<SUBCMD> INQUIRE: % search

Search for class density ratios above a percent threshold. It saves the names of Rspot sets which meet the test criteria in the SRL. It searches for Rspot sets where the MAX of the class mean density ratios $m2/m1$ and $m1/m2$ is $>$ (100%+threshold%). You may further restrict the search to only find Rspots where $m1/m2 > threshold$ or vice versa for a specific class *c1* when you are comparing classes *c1,c2*. Specify this using the */OPTION:c1* switch to `INQUIRE` (eg. `INQUIRE/OPTION:`*c1*). Both classes meets the statistical sizing criteria set with `SET STATISTICS`. Some typical output might be:

% change search

```
303<CMD>: INQUIRE
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%)
 HELP to list CMDS?: % SEARCH<CR>
 Which two classes are to be compared and % threshold?: 1,2,100%<CR>

Rspot[5] mnXYDA=(227,248,.32,31)+-( 1.29,2.93,.19,13) CVD= .60 #G=12
 m2/m1= 2.655,CV1= .000,CV2= .000,n1=4,n2=7,s1= .037,s2= .179
Rspot[10] mnXYDA=(222,261,1.53,29)+-( 3.19,3.23,2.50,20) CVD= 1.63 #G=12
 m2/m1= .148,CV1= 1.000,CV2= .000,n1=4,n2=7,s1= 3.704,s2= .217
Rspot[15] mnXYDA=(206,273,4.21,100)+-( 1.60,3.26,7.09,128) CVD= 1.68 #G=14
 m2/m1= 14.006,CV1= .000,CV2= 1.000,n1=6,n2=7,s1= .275,s2= 8.591
Rspot[17] mnXYDA=(255,272,.34,41)+-( 3.03,2.67,.18,17) CVD= .53 #G=14
 m2/m1= 2.591, CV1= .000, CV2= .000, n1=6, n2=7, s1= .075, s2= .125
        .
        .
        .
```

<SUBCMD> INQUIRE: LEast Square fit search

Do a linear least square fit search for a line $density = M\,classFeature + B$ of a Rspot set such that M is within the range specified. It assumes a linear model which may be inappropriate for some studies and will have false negative and positives if the data is bimodal or non-linear. It saves the names of Rspot sets which meet the test criteria in the SRL. This test requires that the class names be

least-square search

numeric values corresponding to appropriate experimental conditions. Examples might include a time or dose response. For example, duration-in-culture response (Classes(1,2,3,...) = (1mg,5mg,10mg,...); drug-dose response (Classes(1,2,3,...) = (0Hr,24H4,48Hr,...). The default values are the class values (specified by the default `/ClassValue` subcommand switch which uses numeric values parsed from the experimental class names (see `SET CLASSES`). Alternatively, if the numeric information for the dependent variable is in the front part of the "study" field (created using `SET FIELDS`), then you can specify it using the `/StudyValue` subcommand command switch. Typical output might be:

`/STUDYVALUE` *suboption*

```
107<CMD>: INQUIRE
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%)
 HELP to list CMDS?: LEAST SQUARES<CR>
Least Squares Rspot set search
Range of slope to restrict search[.000 : .100E31]?: -1.0E30,1.0E30<CR>

Rspot[  4] mnXYDA=(99,33,2.48,19)+-(1.00,1.00,1.28,3) CVD=.52 #G=2
 Cor.Coef.=1.000, stdErr=.000, [Density=.043*ClassValue-.075]
Rspot[ 34] mnXYDA=(93,136,2.31,23)+-(1.50,.50,1.36,5) CVD=.59 #G=4
 Cor.Coef.=.997, stdErr=.129, [Density=.059*ClassValue-.352]
Rspot[ 48] mnXYDA=(149,36,2.60,22)+-(2.86,1.80,1.49,4) CVD=.57 #G=4
 Cor.Coef.=.993, stdErr=.322, [Density=.050*ClassValue+ .370]
Rspot[137] mnXYDA=(102,170,114.88,146)+-(2.49,1.73,60.19,39) CVD=.52 #G=8
 Cor.Coef.=.997, stdErr=10.883, [Density=2.589*ClassValue-1.628]
        .
        .
        .
```

<SUBCMD> INQUIRE: T-test search

Perform a t-test search for Rspot sets with given significance or confidence limit between two classes ([NatM66], [SneG80]) assuming equal or unequal numbers of spots with equal or unequal variance. It saves the names of Rspot sets which meet the test criteria in the SRL. The t-test significance/confidence limit is set with `SET STATISTICS`. With the `TB-test`, the F-statistic is used to select the standard t-test or Behrens-Fisher t-Test depending on equal or unequal variance as determined by the F-statistic. The t-test forces the equal variance t-test. The `B-test` forces the Behrens-Fisher unequal variance t-test (to be discussed). The `TC-test` forces use of the less sensitive confidence-limits t-test [NatM66]. At the end of the search, the change histogram of m_j/m_i Rspots is printed for the two classes if the `/ChangeHistogram` switch was set when invoking `INQUIRE` (see page 243). Appending `/RELATIVEDIFF` or `/ABSOLUTEDIFF` uses $0.5(m_i - m_j)/(m_i + m_j)$ or $0.5|m_i - m_j|/(m_i + m_j)$ instead of m_j/m_i for the CHANGE HISTOGRAM operation. Some typical searches might be:

four t-Test searches

`/CHANGEHISTOGRAM` *option*

```
231<CMD>: INQUIRE
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,%).
```

```
   HELP to list CMDS?: T-TEST (standard t-Test)<CR>
Std t-Test class search at 0.90 significance
Which two classes are to be compared?: 1,3<CR>

Rspot[   9] mnXYDA=(302,191,31.35,95)+-(0.00,0.00,13.63,22) CVD=0.43 #G=12
 [9] (m1,m3)=(27.38,46.42), m3/m1=1.70 varPooled=30.67 t-T=4.21 f=13.17
Rspot[  59] mnXYDA=(345,207,14.34,52)+-(2.00,9.17,8.44,21) CVD=0.59 #G=10
 [59] (m1,m3)=(13.11,22.87), m3/m1=1.74 varPooled=25.83 t-T=2.35 f=6.20
Rspot[  71] mnXYDA=(324,227,31.68,89)+-(0.00,0.00,15.43,29) CVD=0.49 #G=12
 [71] (m1,m3)=(21.85,45.96), m3/m1=2.10 varPooled=15.16 t-T=7.58 f=3.69
Rspot[  76] mnXYDA=(324,244,12.60,61)+-(1.00,2.24,8.65,15) CVD=0.69 #G=8
 [76] (m1,m3)=(5.41,16.89), m3/m1=3.12 varPooled=1.16 t-T=10.65 f=32.13
       .
       .
       .
Rspot[ 600] mnXYDA=(367,398,16.56,84)+-(2.00,2.00,10.07,32) CVD=0.61 #G=10
 [600] (m1,m3)=(5.34,21.36), m3/m1=4.00 varPooled=13.53 t-T=4.36 f=1.95

Found 20 Rspots, mean density+-'sd/mn' = 1.43+-1.09

242<CMD>: INQUIRE
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,%).
 HELP to list CMDS?: TB-TEST (use F-statistic to pick T or F test)<CR>
Std t-/Behrens-Fisher t-Test (using F-stat) class search at 0.90 significance
Which two classes are to be compared?: 1,3<CR>

Rspot[   9] mnXYDA=(302,191,31.35,95)+-(0.00,0.00,13.63,22) CVD=0.43 #G=12
 [9] (m1,m3)=(27.38,46.42), m3/m1=1.70 varPooled=30.67 t-T=4.21 f=13.17
Rspot[  59] mnXYDA=(345,207,14.34,52)+-(2.00,9.17,8.44,21) CVD=0.59 #G=10
 [59] (m1,m3)=(13.11,22.87), m3/m1=1.74 varPooled=25.83 t-T=2.35 f=6.20
Rspot[  71] mnXYDA=(324,227,31.68,89)+-(0.00,0.00,15.43,29) CVD=0.49 #G=12
 [71] (m1,m3)=(21.85,45.96), m3/m1=2.10 varPooled=15.16 t-T=7.58 f=3.69
Rspot[  76] mnXYDA=(324,244,12.60,61)+-(1.00,2.24,8.65,15) CVD=0.69 #G=8
 [76] (m1,m3)=(5.41,16.89), m3/m1=3.12 varPooled=1.16 t-T=10.65 f=32.13
        .
        .
        .
Rspot[ 600] mnXYDA=(367,398,16.56,84)+-(2.00,2.00,10.07,32) CVD=0.61 #G=10
 [600] (m1,m3)=(5.34,21.36), m3/m1=4.00 varPooled=13.53 t-T=4.36 f=1.95

Found 20 Rspots, mean density+-'sd/mn' = 1.43+-1.09

253<CMD>: INQUIRE
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,%).
 HELP to list CMDS?: TC-TEST (confidence limits T-test)<CR>
t-Test (confidence limits) class search at 0.90 significance
Which two classes are to be compared?: 1,3<CR>

Rspot[   9] mnXYDA=(302,191,31.35,95)+-(0.00,0.00,13.63,22) CVD=0.43 #G=12
 [9] (m1,m3)=(27.38,46.42),m3/m1=1.70,Lim1-3[17.1:37.6,43.6:49.2] n1=3 n3=3 f=13.2
Rspot[  71] mnXYDA=(324,227,31.68,89)+-(0.00,0.00,15.43,29) CVD=0.49 #G=12
 [71] (m1,m3)=(21.85,45.96),m3/m1=2.10,Lim1-3[18.4:25.3,39.3:52.6] n1=3 n3=3 f=3.7
Rspot[  76] mnXYDA=(324,244,12.60,61)+-(1.00,2.24,8.65,15) CVD=0.69 #G=8
```

```
 [76] (m1,m3)=(5.41,16.89),m3/m1=3.12,Lim1-3[4.9:6.0,13.8:20.0] n1=2 n3=2 f=32.1
        .
        .
        .
Found 13 Rspots, mean density+-'sd/mn' = 1.35+-1.04

821<CMD>: INQUIRE//change histogram/AbsoluteDiff //1,2
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%).
 HELP to list CMDS?: change histogram/AbsoluteDiff
Which two classes are to be compared?: 1,2
The 64 Rspots are:  223 231 233 235 236 238 250 276 286 287 292
                295 302 309 323 337 345 347 349 354 368 394 408
                421 449 459 486 490 529 581 592 593 610 666 667
                668 673 689 716 733 739 744 757 759 761 765 767
                773 786 858 862 872 877 910 913 915 917 919 921
                922 932 936 938 945
---------------------------------------------------------------
|m1-m2|/(m1+m2)/2 Rspot sets
 0.00 238 250 286 287 295 349 529 593 610 667 689 739 744 761 767 786 858...
 0.05 233 235 236 302 323 354 368 421 449 459 486 490 581 592 673 716 733...
 0.10 223 231 276 292 309 337 347 408 759 773 936
 0.15 666 668 910
 0.20 345 394
 0.25
 0.30
[09:42:47AM] Real TIME =00:00:03  CPU TIME =00:00:03,  100.00%

822<CMD>: INQUIRE//change histogram/RelativeDiff //1,2
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%).
 HELP to list CMDS?: change histogram/RelativeDiff
Which two classes are to be compared?: 1,2
The 64 Rspots are:  223 231 233 235 236 238 250 276 286 287 292
                295 302 309 323 337 345 347 349 354 368 394 408
                421 449 459 486 490 529 581 592 593 610 666 667
                668 673 689 716 733 739 744 757 759 761 765 767
                773 786 858 862 872 877 910 913 915 917 919 921
                922 932 936 938 945
---------------------------------------------------------------
(m1-m2)/(m1+m2)/2 Rspot sets
-0.35
-0.30
-0.25 345 394
-0.20 666 668 910
-0.15 276 347 759 773 936
-0.10 235 236 354 459 673 716 733 765 877 913
-0.05 286 287 349 610 667 689 739 744 862 915 922 932 938
 0.00 238 250 295 529 593 761 767 786 858 919 921 945
 0.05 233 302 323 368 421 449 486 490 581 592 757 872 917
 0.10 223 231 292 309 337 408
 0.15
 0.20
[09:43:47AM] Real TIME =00:00:07  CPU TIME =00:00:03,  42.86%
```

<SUBCMD> INQUIRE: B-test search

Perform the B-test search for Rspot sets with given confidence limit between two classes assuming equal or unequal numbers of spots with unequal variance. It saves the names of Rspot sets which meet the test criteria in the SRL. This form of the t-test is called the Behrens-Fisher t′-test [SneG80]. The B-test significance limit is set with `SET STATISTICS` as for the standard t-test. At the end of the search, the change histogram of m_j/m_i Rspots is printed for the two classes if the `/ChangeHistogram` switch was set when invoking `INQUIRE`. Some typical output might be:

Behrens-Fisher test

```
299<CMD>: INQUIRE
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,%).
 HELP to list CMDS?: BEHRENS-FISHER-T-TEST<CR>
Behrens-Fisher t-Test class search at 0.90 significance
Which two classes are to be compared?: 1,3<CR>

Rspot[   9] mnXYDA=(302,191,31.35,95)+-(0.00,0.00,13.63,22) CVD=0.43 #G=12
[9](m1,m3)=(27.38,46.42),m3/m1=1.70 dF=2.30 t-BF=4.21] n1=3 n3=3 f=13.17 s1=7.55 s2=2.08
Rspot[  71] mnXYDA=(324,227,31.68,89)+-(0.00,0.00,15.43,29) CVD=0.49 #G=12
[71](m1,m3)=(21.85,45.96),m3/m1=2.10 dF=3.01 t-BF=7.58] n1=3 n3=3 f=3.69 s1=2.54 s2=4.88
Rspot[  76] mnXYDA=(324,244,12.60,61)+-(1.00,2.24,8.65,15) CVD=0.69 #G=8
[76](m1,m3)=(5.41,16.89),m3/m1=3.12 dF=1.06 t-BF=10.65] n1=2 n3=2 f=32.13 s1=0.26 s2=1.50

Rspot[ 600] mnXYDA=(367,398,16.56,84)+-(2.00,2.00,10.07,32) CVD=0.61 #G=10
[600](m1,m3)=(5.34,21.36),m3/m1=4.00 dF=1.81 t-BF=4.36] n1=2 n3=2 f=1.95 s1=3.03 s2=4.23
        .
        .

Found 16 Rspots, mean density+-'sd/mn' = 1.34+-1.05
```

<SUBCMD> INQUIRE: F-test search

Perform F-test search for Rspot sets with given confidence limit between any number of classes (up to 9) classes [SneG80] for up to 9 classes. It saves the names of Rspot sets which meet the test criteria in the SRL. At the end of the search, the change histogram of m_j/m_i Rspots is printed for the first two specified classes if the `/ChangeHistogram` switch was set when invoking `INQUIRE`. An example follows for three classes (`Control, treatment 1, treatment 2`). Note that setting the prefilter confidence limits to 1%, 5%, 10%, 20% corresponds to setting the p-value for the F-test to 0.995, 0.975, 0.90 and 0.75 respectively.

F-test

```
311<CMD>: INQUIRE
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,%).
 HELP to list CMDS?: F-TEST<CR>
F-test class search at 0.90 significance
Which two or more classes are to be compared?: 1,2,3,4<CR>

Rspot[   9] mnXYDA=(302,191,31.35,95)+-(0.00,0.00,13.63,22) CVD=0.43 #G=12
```

```
[9]mnD[27.38,34.64,46.42,15.77] m2/m1=1.26, mSqWthn=93.15, mSqBtwn=411.21, F=4.41
Rspot[  11] mnXYDA=(304,202,25.35,76)+-(1.00,2.45,15.65,28) CVD=0.62 #G=12
[11]mnD[22.12,44.79,23.07,13.63] m2/m1=2.02, mSqWthn=128.78, mSqBtwn=474.82, F=3.69
Rspot[  61] mnXYDA=(317,198,46.79,104)+-(1.00,1.41,25.69,28) CVD=0.55 #G=11
[61]mnD[19.72,75.13,53.41,24.01] m2/m1=3.81, mSqWthn=254.85, mSqBtwn=1680.76, F=6.60
Rspot[  71] mnXYDA=(324,227,31.68,89)+-(0.00,0.00,15.43,29) CVD=0.49 #G=12
[71]mnD[21.85,34.22,45.96,11.19] m2/m1=1.57, mSqWthn=65.59, mSqBtwn=574.71, F=8.76
Rspot[  76] mnXYDA=(324,244,12.60,61)+-(1.00,2.24,8.65,15) CVD=0.69 #G=8
[76]mnD[5.41,22.19,16.89,5.88] m2/m1=4.10, mSqWthn=27.28, mSqBtwn=138.19, F=5.07
      .
      .
      .
Rspot[ 594] mnXYDA=(355,381,60.27,160)+-(0.00,0.00,24.51,47) CVD=0.41 #G=12
[594]mnD[54.52,53.05,89.76,37.16] m2/m1=0.97, mSqWthn=380.32, mSqBtwn=1310.47, F=3.45

Found 15 Rspots, mean density+-'sd/mn' = 2.56+-2.73
```

<SUBCMD> INQUIRE: WMW-test search

rank order test

Perform a Rank order Wilcoxon-Mann-Whitney search for Rspot sets with given significance between 2 classes [NatM66]. There must be at least 2 gels in one of the classes. It saves the names of Rspot sets which meet the test criteria in the SRL. Either `rank` or `wmw` may be used to invoke this test. The rank order test confidence limit is set with `SET STATISTICS`. At the end of the search, the change histogram of m_j/m_i Rspots is printed for the two classes if the `/ChangeHistogram` switch was set when invoking `INQUIRE`. A typical search might produce the following:

```
297<CMD>:INQUIRE
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%).
 HELP to list CMDS?: RankOrder (WMW)-test<CR>
Wilcoxson-Mann-Whitney Rank order test class search at 0.90 significance
REMINDER - PCG DB must first be sorted using REORDER use non-
parametric tests. Do REORDER in this density mode and try again.
Which two classes are to be compared?: 1,3<CR>

Rspot[   1] mnXYDA=(302,174,5.03,38)+-(8.00,11.58,6.74,21) CVD=1.34 #G=9
 [1] n1=2 n2=3 n=5 R=9 R'=3 Ralpha=6
Rspot[   2] mnXYDA=(305,169,3.87,35)+-(9.85,13.19,5.35,25) CVD=1.38 #G=10
 [2] n1=2 n2=3 n=5 R=7 R'=5 Ralpha=6
Rspot[   3] mnXYDA=(316,175,2.46,39)+-(1.41,1.41,1.52,14) CVD=0.61 #G=9
 [3] n1=2 n2=3 n=5 R=6 R'=6 Ralpha=6
Rspot[  14] mnXYDA=(334,138,17.09,65)+-(8.83,12.49,16.04,37) CVD=0.94 #G=11
 [14] n1=2 n2=3 n=5 R=4 R'=8 Ralpha=6
Rspot[  18] mnXYDA=(423,118,1.41,35)+-(4.47,2.24,1.14,35) CVD=0.81 #G=7
 [18] n1=2 n2=3 n=5 R=6 R'=6 Ralpha=6
Rspot[  20] mnXYDA=(351,131,2.72,38)+-(2.65,3.00,3.83,24) CVD=1.41 #G=11
 [20] n1=2 n2=3 n=5 R=9 R'=3 Ralpha=6
Rspot[  24] mnXYDA=(353,147,3.24,44)+-(3.16,1.41,1.99,16) CVD=0.61 #G=10
 [24] n1=2 n2=3 n=5 R=5 R'=7 Ralpha=6
Rspot[  25] mnXYDA=(303,148,3.73,38)+-(3.00,2.24,4.64,38) CVD=1.24 #G=9
 [25] n1=2 n2=3 n=5 R=9 R'=3 Ralpha=6
```

```
Rspot[  60] mnXYDA=(330,202,7.91,46)+-(9.11,7.68,11.76,21) CVD=1.49 #G=10
 [60] n1=2 n2=3 n=5 R=9 R'=3 Ralpha=6
Rspot[  72] mnXYDA=(314,234,1.52,28)+-(1.73,2.24,1.67,10) CVD=1.10 #G=9
 [72] n1=2 n2=3 n=5 R=6 R'=6 Ralpha=6
Rspot[  73] mnXYDA=(322,237,9.91,52)+-(2.45,1.73,15.09,55) CVD=1.52 #G=11
 [73] n1=2 n2=3 n=5 R=5 R'=7 Ralpha=6
Rspot[  78] mnXYDA=(331,253,7.81,53)+-(2.00,4.47,8.54,22) CVD=1.09 #G=9
 [78] n1=2 n2=3 n=5 R=3 R'=9 Ralpha=6
        .
        .
        .
Rspot[ 648] mnXYDA=(264,404,12.37,57)+-(39.41,11.70,18.13,57) CVD=1.47 #G=11
 [648] n1=2 n2=3 n=5 R=7 R'=5 Ralpha=6
Don't worry, I am still working on it - Rspot set [996]

Found 63 Rspots, mean density+-'sd/mn' = 2.07+-1.29
[12:30:31AM] Real TIME =00:02:36  CPU TIME =00:01:16,  48.72%
```

<SUBCMD> INQUIRE: *Kruskol-Wallis search

Perform a Kruskol-Wallis rank order search for Rspot sets with given significance between all classes for min 5 gels/class [NatM66]. It saves the names of Rspot sets which meet the test criteria in the SRL. The significance limit is set for this test with `SET STATISTICS`. At the end of the search, the change histogram of m_j/m_i Rspots is printed for the first two classes if the `/ChangeHistogram` switch was set when invoking `INQUIRE`.

Darkest-spots search for N darkest Rspots. **<SUBCMD> INQUIRE:**

Darkest spots

Search for the N darkest Rspot sets meeting statistical limits like `Index Search` above but only print the sorted Rspot set numbers and densities for those spots meeting the criteria. It saves the names of Rspot sets which meet the test criteria in the SRL. The Rspot set for statistical purposes is treated as a single class with single mean and standard deviation, etc. The following output illustrates this command.

```
301<CMD>: INQUIRE
 Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%)
  HELP to list CMDS?: DARKEST<CR>
  Enter maximum number of darkest spots to find?: 10<CR>
  Darkest spots search for N darkest Rspots
   Found 738 Rspots, mean (sd/mn) +-'std-dev (sd/mn)' = 1.23+-0.58
   #   1  Rspot[ 154] Density = 179.43L
   #   2  Rspot[ 149] Density = 151.87L
   #   3  Rspot[ 145] Density = 145.19L
   #   4  Rspot[ 157] Density = 129.37L
   #   5  Rspot[ 626] Density = 128.05L
   #   6  Rspot[ 719] Density = 126.41L
   #   7  Rspot[ 596] Density = 126.22L
   #   8  Rspot[ 195] Density = 126.18L
```

```
#   9  Rspot[ 508] Density = 125.60L
#  10  Rspot[ 147] Density = 114.86L
[08:48:09AM] Real TIME =00:00:52  CPU TIME =00:00:41,  78.85%
```

<SUBCMD> INQUIRE: Expression Profile test

expression profile test

Search for Rspots which have a protein expression profile similar to the model profile provided with a minimum least square error $\leq$ a threshold. It saves the names of Rspot sets which meet the test criteria in the SRL. An expression-profile (see [SonP85]) is an n-tuple numeric pattern discussed below. It consists of an ordered list of ratios of mean protein concentrations of a particular protein as a function of experimental class relative to the first class. Given n experimental classes, let protein k for class c have density m_{ck}. Then, the expression profile is the n-tuple:

$$(1.0 : \frac{m_{2k}}{m_{1k}} : \ \ldots : \ \frac{m_{nk}}{m_{1k}}). \tag{3.2}$$

Similarity between profiles of two different proteins j and k is $S_{j,k}$ and is measured by the inverse of the least square error between profiles:

$$LSQ_{j,k} = \sqrt{\frac{\sum_{j=1,c}(m_{cj} - m_{ck})^2}{(c-1)}}. \tag{3.3}$$

then,

$$S_{j,k} = \frac{1}{LSQ_{j,k}}. \tag{3.4}$$

An expression profile search may be performed through a database looking for spots whose expression profile is similar according to the previous definition such that the least square error < some threshold. The following example is an expression profile search with a least square error threshold = 0.4. The search is for expected expression profile (c1 : c2 : c3 : c4) = (1.0 : 0.5 : 3.0 : 0.5).

This list of Rspots (or any list for that matter) can be used to create an expression profile table of spots linked by maximum similarity. Here the threshold was set to 0.40. For each row entry corresponding to a single Rspot is a list of similar spots. Each entry of this list is 2-tuples ($Rspot : lsqErrValue$). Those entries on the front of the list are most similar. The *lsqErrValue* is with respect to the Rspot which contains that list.

```
331<CMD>: INQUIRE
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,%).
 HELP to list CMDS?: EXPRESSION-PROFILE SEARCH<CR>
Profile-expression (min LSQ) search
Which two or more classes are to be compared?: 1,2,3,4<CR>
Enter minimum LSQ profile threshold and ratios of the
desired expression profiles we are looking for.
```

```
Eg. thresh,c1:c2:c3:c4 might be 0.5,1.0:2.3:7.6:0.5.
 ?: 0.4, 1.0 : 0.5 : 3.0 : 0.5<CR>

Rspot[ 145] mnXYDA=(305,277,15.44,83)+-(1.41,1.73,12.21,27) CVD=0.79 #G=9
 [145] lsqErr=0.38 mnDens=[11.29,13.01,32.40,6.26] EP=[1.00:1.15:2.87:0.55]
Rspot[ 269] mnXYDA=(265,135,10.66,40)+-(1.73,1.41,15.16,18) CVD=1.42 #G=9
 [269] lsqErr=0.37 mnDens=[8.75,8.09,29.11,1.41] EP=[1.00:0.92:3.33:0.16]
Rspot[ 279] mnXYDA=(280,167,3.74,34)+-(1.00,1.00,4.44,13) CVD=1.19 #G=10
 [279] lsqErr=0.12 mnDens=[2.71,0.97,8.52,1.48] EP=[1.00:0.36:3.15:0.55]
Rspot[ 440] mnXYDA=(218,264,4.05,42)+-(2.24,1.73,5.97,17) CVD=1.47 #G=9
 [440] lsqErr=0.22 mnDens=[3.39,1.40,10.43,2.90] EP=[1.00:0.41:3.08:0.86]

Found 4 Rspots, mean density+-'sd/mn' = 3.97+-3.99

Mean-Dens-Class-I/Mean-Dens-Class-J
Rspot:  m1/1 m2/1 m3/1 m4/1 m5/1
-----------------------------------------------------------------
  145    1.0  1.2  2.9  0.6  0.0
  269    1.0  0.9  3.3  0.2  0.0
  279    1.0  0.4  3.1  0.5  0.0
  440    1.0  0.4  3.1  0.9  0.0

 Sorted by Minimum lsqErr of profiles (Rspot#:lsqErrValue) list
-----------------------------------------------------------------
  145
  269  145:0.32
  279  440:0.16 269:0.36
  440  279:0.16
```

<SUBCMD> INQUIRE: Change histogram of SRL table

The Change Histogram computes the 2-class (i, j) mean-densities ratio m_j/m_i for each Rspot in the SRL. Then each Rspot name is plotted in a histogram bin according to where it's ratio falls in the range of 0.00 to 20.0 in steps of 0.05. Clusters of Rspots which may putatively be coordinately regulated may appear in adjacent bins giving the appearance of multimodal distributions. Note these results are merely suggestive of possible correlations which need to be further investigated. The set of spots used to compute this histogram are those in the Search Results List - which may result from the previous search or as the result of SRL subset operations (`SET SRL SUBSETS`). Note that the `/CHANGEHISTOGRAM` switch invoked with any of the 2-class tests computes the change histogram after the test completes. The following example illustrates its operation.

expression change histogram

```
44<CMD>: INQUIRE
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%)
 HELP to list CMDS? CHANGE HISTOGRAM<CR>
 Which two classes are to be compared?: 1,2<CR>
```

```
Class#1=CONTROL, Class#2=EXPERIMENTAL
The 12 Rspots: 15 61 167 232 244 335 340 348 360 423 440 482
-----------------------------------------------------------------
m2/m1   Rspot sets
  .00  61 167
  .05
         .
         .
         .
  .60
  .65  15
  .70
  .75 360
  .80
  .85 482
  .90
  .95
 1.00 244 348 423 440
 1.05
         .
         .
         .
 1.30
 1.35 335
 1.40
 1.45
 1.50 232
 1.55
         .
         .
         .
 1.95
 2.00 340
```

With more spots in the SRL, these peaks become more apparent.

```
45<CMD>: INQUIRE
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%)
 HELP to list CMDS? CHANGE HISTOGRAM<CR>
 Which two classes are to be compared?: 5,6<CR>

Class#5=CONTROL, Class#6=EXPERIMENTAL
The 37 Rspots: 4 9 37 44 52 57 64 82 87 94 97 99 117 133 142
164 170 172 176 182 185 190 191 200 225 239 252 285 323 371 393
426 474 506 532 544
-----------------------------------------------------------------
m6/m5        Rspot sets
  .20 426
  .25
  .30
  .35  64 225 252 532
  .40  96 190
  .45  97  99 142 200
```

```
.50  37  52  57  94 172 544
.55   9 133 176 185 191 285 474
.60   4  82  84 182 239 323 506
.65 170 371
.70
.75 117 164
.80
.85
       .
       .
       .
1.60
1.65  44
1.70
       .
       .
       .
2.55
2.60 393
```

<SUBCMD> INQUIRE: COordinate Pairs in SRL table

The search for coordinated pairs of proteins is conducted on those polypeptide spots that are in the SRL of a previous search for differences between the protein pattern of two experimental classes [LemP84b]. A pair of Rspots is said to be *coordinate* if the ratio of the sums of the mean density values of two different spots in two different classes of gels has a value of approximately 1.0. Coordinate pairs are found by tracking the upper and lower error bounds of this ratio and applying an error fraction f parameter to the computed ratios.

coordinate-pair expression

By further restricting coordinate-pair's to constant isoelectric point (pIe) or constant apparent molecular mass (MW), additional candidate pairs for such properties (phosphorylation, cleavage or other post-translational modification mechanisms) might be detected. By "constant" we mean relatively constant as defined by an allowable deviation: dpIe or dMW. Of course, pIe and MW are nonlinear on the gel, so that a constant deviation would have to be applied nonlinearly over the 2D gel (for large deviations). Proper treatment of constant deviation would take into account the nonlinear mapping of image picture elements (pixels) to pIe and MW. This implies that pIe and MW standards need to be built into the gel-and that the gel-analysis system takes them into account. This calibration facility is partially available in GELLAB.

Computation of Coordinate Pairs of Rspots

A coordinate-pair exists when the condition specified below in the last equation is true. The first set of equations computes the ratios and the bounds on the ratios. We define the following symbols which are used in the derivation.

Let M_{ck} be the mean density (normalized density) of Rspot[k] for class c.

Let S_{ck} be the standard deviation of mean density (normalized density) of Rspot[k] for class c.
Let pIe_k be the pIe of Rspot[k], in pixels.
Let MW_k be the MW of Rspot[k], in pixels.
Let T_{mw} be the maximum allowable MW deviation threshold, in pixels.
Let T_{pIe} be the maximum allowable pIe deviation threshold, in pixels.
Let f be the % error allowed in the deviation of R_{ij} from 1.0 considering whether R_{ij} is about 1.0.

Then for Rspots i and j, and classes p and q, compute the ratios R_{ij}, upper bound RU_{ij}, and lower bound RL_{ij}.

$$R_{ij} = \frac{(M_{pi} + M_{pj})}{(M_{qi} + M_{qj})} \tag{3.5}$$

$$RU_{ij} = \frac{(M_{pi} + S_{pi}) + (M_{pj} + S_{pj})}{(M_{qi} - S_{qi}) + (M_{qj} - S_{qj})} \tag{3.6}$$

$$RL_{ij} = \frac{(M_{pi} - S_{pi}) + (M_{pj} - S_{pj})}{(M_{qi} + S_{qi}) + (M_{qj} + S_{qj})} \tag{3.7}$$

The two following cases for evaluating R in the context of ij the error bounds on the real line may be visualized as follows:

$$(R_{ij} >= 1.0) * * * * * RL_{ij} - - - -1.0 - - - - R_{ij} * * * * * * \tag{3.8}$$

with

$$RL_{ij} \leq 1.0 \; and \; R_{ij} >= 1.0, \tag{3.9}$$

$$(R_{ij} \leq 1.0) * * * * * R_{ij} - - - -1.0 - - - - RU_{ij} * * * * * * \tag{3.10}$$

with

$$RU_{ij} \geq 1.0 \; and \; R_{ij} \leq 1.0, \tag{3.11}$$

pair constraints

Finally, a coordinate pair (i, j) exists when the following logical condition, described in the following constraint equation, is true.

$$(|1.0 - R_{ij}| \leq f) \; and \tag{3.12}$$

$$(((R_{ij} \geq 1.0) \; and \; (RL_{ij} \leq 1.0)) \; or \; ((R_{ij} < 1.0) \; and \; (RU_{ij} \geq 1.0))) \; and \tag{3.13}$$

$$(|pIe_i - pIe_j| < T_{pIe}) \; and \tag{3.14}$$

$$(|MW_i - MW_j| < T_{mw}). \tag{3.15}$$

If there is a shift expected toward a *particular* pIe or MW, then the direction of the test is important. The next two equations should then be used to express the last two constraints in the above Equation as an ordered relation rather than as an absolute difference.

$$(pIe_i - pIe_j < T_{pIe}). \tag{3.16}$$

shift toward basic spot i from acid spot j.

$$(MW_i - MW_j < T_{mw}). \tag{3.17}$$

shift toward lower MW spot j from spot i.

The following example illustrates the results of a search with coordinate spot pairs found applying the Coordinate-Pair search to 13 spots in the SRL [LemP84b]. Potential spots were from a previous F-Test at .95 with CV area < 0.5, maximum OD < 1.8 and 7 Gel samples/Class. The deviation factor f set to 20% for two classes with T_{mw} And T_{pIe} set to infinity. Given a SRL consisting of the following spots {85, 86, 90, 162, 335, 359, 361, 435, 462, 484, 516, 582},

```
33<CMD>: INQUIRE
  Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%)
   HELP to list CMDS? COORDINATE PAIR SEARCH
  Which two classes are to be compared & %Err,TpIe,Tmw?:1,2,20%,512,512
C.P. found      Rij      [RLij : RUij]  (dpIe,dMW)
Rspot[85,90]    0.98     [.19: 2.67]    (1,12)
Rspot[85,462]   1.19     [.02: 8.78]    (21,210)
Rspot[85,484]   0.85     [.11: 3.63]    (155,171)
Rspot[86,90]    0.94     [.31: 2.60]    (12,11)
Rspot[86,462]   1.06     [.23: 6.62]    (34,209)
Rspot[86,484]   0.84     [.19: 3.47]    (168,170)
Rspot[90,516]   0.80     [.20: 2.02]    (228,79)
Rspot[162,335]  0.92     [.43: 1.69]    (15,74)
Rspot[162,516]  0.92     [.22: 2.25]    (240,122)
Rspot[335,359]  0.96     [.43: 1.90]    (7,38)
Rspot[335,361]  0.94     [.38: 2.07]    (23,37)
Rspot[335,435]  0.94     [.40: 1.91]    (13,30)
Rspot[335,462]  0.82     [.33: 1.88]    (19,167)
Rspot[335,516]  1.04     [.32: 2.33]    (225,48)
Rspot[335,582]  0.93     [.42: 1.75]    (151,40)
Rspot[359,516]  0.95     [.20: 2.56]    (232,10)
Rspot[361,516]  0.93     [.17: 2.80]    (248,11)
Rspot[435,516]  0.94     [.23: 2.42]    (212,18)
Rspot[462,516]  0.83     [.15: 2.57]    (206,119)
Rspot[516,582]  0.93     [.23: 2.25]    (74,8)
```

<SUBCMD> INQUIRE: ORder SRL table

order expression table

Order Rspot sets, named in the SRL, by all combinations of class pairs. By non-null classes we mean that there exists gels in the working set and whose Rspot subsets in each class meet the limit preconditions specified above in the initial description of the `INQUIRE` command). This creates, for the classes which are not null, the following type of table consisting of entires >, <, -. The >, < indicate that for classes c1 and c2 (each meeting the current sizing limits) that mean density of c1 is > that of c2 or vice versa. For those entries not meeting sizing criteria or with no gels exist in that Rspot set, the entry is set to −.

```
2<CMD>: INQUIRE
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%).
 HELP to list CMDS?: ORDER SRL TABLE<CR>
Order Rspots by class pairs for SRL.

Rspot: m1/2 m1/3 m1/4 m2/3 m2/4 m3/4
----------------------------------------
   14    <    >    >    >    >    >
   76    >    >    >    >    >    <
  126    <    >    >    >    >    >
  176    >    <    >    <    >    >
  193    >    <    >    <    <    >
  213    <    <    >    >    >    >
  248    <    <    >    >    >    >
  257    <    <    >    >    >    >
  283    <    <    >    >    >    >
  308    <    <    >    >    >    >
  325    <    <    >    >    >    >
  335    <    <    >    >    >    >
  337    <    <    <    >    >    >
  342    <    >    >    >    >    >
  371    <    <    >    >    >    >
  395    <    <    >    >    >    >
  455    <    >    >    >    >    >
  466    <    <    <    >    >    >
  510    <    <    >    >    >    >

Rspot:  m1/2 m1/3 m1/4 m2/3 m2/4 m3/4
----------------------------------------
   14     .7  1.1  1.8  1.6  2.7  1.7
   76    1.8  3.2  2.1  1.8  1.2   .7
  126     .8  2.6  4.3  3.3  5.5  1.6
  176    1.8   .6  3.8   .3  2.1  6.6
  193    2.3   .8  2.0   .4   .9  2.4
  213     .1   .4  2.4  3.3 20.1  6.1
  248     .1   .3  1.4  3.6 15.3  4.3
  257     .1   .3  1.9  2.5 18.0  7.1
  283     .1   .3  1.4  3.8 16.4  4.4
```

```
 308     .1   .3  1.5  3.7 18.8  5.1
 325     .2   .6  2.6  3.1 13.2  4.3
 335     .3   .7  1.2  2.1  3.5  1.7
 337     .1   .2   .7  3.1 10.0  3.2
 342     .2  1.1  3.7  6.2 21.6  3.5
 371     .7   .7  2.7  1.1  4.2  3.7
 395     .1   .3  3.0  3.3 38.6 11.5
 455     .2  1.5  3.1  7.9 15.9  2.0
 466     .1   .2   .8  3.0 11.3  3.8
 510     .1   .3  1.9  4.2 27.1  6.4
```

<SUBCMD> INQUIRE: OExpression Profile SRL table

First compute the Order SRL Rspots table and then compute the minimum least square error between the profiles of all combinations of Rspots in the SRL. List these by minimum lsqErr for those with $lsqErr < threshold$. See discussion on `INQUIRE` subcommand `Expression-Profile` search on page 242. This is discussed in [LemP88d]. The Ordered Expression Profile table indicates those spots with similar expression profiles.

similar expression profiles

```
359<CMD>: INQUIRE
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,%).
 HELP to list CMDS?: OEXPRESSIONPROFILE<CR>
Order Expression Profile of Rspots in SRL.
Enter minimum LSQ profile threshold
 ?: 0.5 min LSQ profile match error<CR>

Mean-Dens-Class-I/Mean-Dens-Class-J for LsqErr-threshold=0.5
Rspot:  m1/1 m2/1 m3/1 m4/1 m5/1
-----------------------------------------------------------------------
    9    1.0  1.3  1.7  0.6  0.7
   29    1.0  0.6  0.8  0.2  0.1
   53    1.0  0.8  1.9  0.9  1.3
   63    1.0  1.2  1.3  0.6  0.6
   71    1.0  1.6  2.1  0.5  2.4
   94    1.0  1.7  0.7  0.8  6.2
  101    1.0  2.5  7.7  1.9  3.4
  120    1.0  2.3  0.7  1.2  0.8
  155    1.0  0.9  1.5  0.6  0.7
  182    1.0  1.0  1.0  0.5  1.1
  190    1.0  1.2  1.5  0.6  0.3
  237    1.0  6.7 10.6  2.1  0.0
  453    1.0  0.1  0.3  0.4  0.4
  499    1.0  2.8  2.1  0.7  0.5
  523    1.0  1.3  1.3  1.1  1.3
  551    1.0  0.5  8.7  0.1  3.5
  594    1.0  1.0  1.6  0.7  1.0
  629    1.0  0.3  0.1  0.6  0.2
  652    1.0  1.0  1.9  0.9  0.5
 Sorted by Minimum lsqErr of profiles (Rspot#:lsqErrValue) list
-----------------------------------------------------------------
```

```
  9  594:0.23  63:0.23 190:0.24 652:0.25 523:0.41  53:0.42 182:0.42
 29  629:0.46  63:0.46 190:0.50
 53  594:0.22 523:0.38   9:0.42 652:0.44 182:0.48
 63  155:0.21 190:0.22   9:0.23 594:0.32 561:0.35 652:0.37 523:0.42  29:0.46
 71
 94
101
120
155  594:0.20  63:0.21   9:0.21 652:0.28 190:0.28 182:0.33  53:0.39
182   63:0.31 594:0.33 155:0.33 523:0.39   9:0.42  53:0.48
190   63:0.22   9:0.24 155:0.28 652:0.30 594:0.42  29:0.50
237
453  629:0.24  29:0.40
499
523  594:0.33  53:0.38 182:0.39   9:0.41  63:0.42 155:0.44
551
594  155:0.20  53:0.22   9:0.23  63:0.32 182:0.33 523:0.33 652:0.33 190:0.42
629  453:0.24  29:0.46
652    9:0.25 155:0.28 190:0.30 594:0.33  63:0.37  53:0.44
```

<CMD> Limits

prefilter status

Print the current statistical limits, (MW,pIe) region and pairing label as part of the prefilter illustrated by the following output. The prefilter statistical limits may be set using the `SET STATISTICS` command.

```
3<CMD>: LIMITS
Rspot Mapping file [].
Map Rspots to Fspots is OFF.
Ratio-List density mode calibration EXISTS.
Least-Squares density mode calibration EXISTS.
Image size [rowsXcolumns]=[512X512]
Relative distance limits [0.00,512.00] pixels
Mn DL limits [0.00,512.00] pixels
Mn DP limits [0.00,10.00] pixels
MN area limits [0.00,10000000.00] pixels**2
MN density (Mode: R) limits [0.00,10000000.00]
Coef. variation: S.D./Mn Rset area limits [0.00,10.00]
Coef. variation: S.D./Mn Rset dens limits [0.00,10.00]
Spot OD difference limits [0.00,10.00] OD
Class difference t-Test, F-test, Rank order p-value limit is .90:
Check if # gels in Rspot set [0:1000]
(x,y) calibration file: <NONE>
pIe Rgel limits [0.00,511.00] pixels
MW Rgel limits [0.00,511.00] pixels
Pairing-Search-Label [PSEAU]
```

<CMD> *MErgeAP
Merge AP's with SP or PP if meet DP limits. (See `EDIT` command, page 210 on how to do this manually).

<CMD> MOsaic
Generate a mosaic style derived gel image or plot around the specified Rspot (with optional *.ugf* plot file). Since the PREFILTER is active, you can restrict spots which you wish to view in the mosaic plot. The following prompt is given to the user with a typical response given. Use `SET DISPLAY` to define the plot or display type to be used. The associated plot switches are active: `/ALLgels`, `/CLASSname`, `/COLORlabel`, `/DUMPPPXplot`, `/FILLlabeledSpots`, `/HFLIPpIe`, `/LABEL`, `/MAKEUP:nXn`, `/PPXplot`, `/SIZEbyD'`, `/SRLlabel`, `/TITLE:'...'`, `/UGFlabel`, `/VFlipMW`, `/ZOOM:nX`). The default switches are: `/FILLlabeledSpots`, `/SRLlabel`, `/UGFlabel`, `/ZOOM:1X`. Figure 3.6 shows a sample plot output.

mosaic images or plots

```
    44<CMD>: MOSAIC
    Draw a mosaic around Rspot (or ALLSRL)#?: 67
    List Gels with which to construct mosaic
     [ALL (i.e. working set)] ?: all
    Building 4X4 mosaic with PPX image range=128 mosaic-size=128 from 12 gels:
    324.1 0369.1 0378.2 0384.1 0396.1 0497.1 0503.1 0511.1 0514.1 0515.1
    0517.1 0393.2
    Drawing mosaic at 1X magnification.
    Title:
    Making a mosaic of 12 gels found in Rspot[67].
    Mosaics ordered by minimum spot density as follows:
    503.1 0517.1 0514.1 0393.2 0511.1 0515.1 0497.1 0324.1 0378.2 0384.1
    0369.1 0396.1
    Mosaic centered at xyRgel=[354,191] limits pIe[258:450] MW[95:287]
          .
          .
          .


   6<CMD>: MOSAIC
    Draw a mosaic around Rspot (or ALLSRL)#?: 69/zoom:1/SRLlabel/Title:'Landmarks'<CR>
or
   7<CMD>: MOSAIC
    Draw a mosaic around Rspot# (or ALLSRL)?: 69/zoom:2/All<CR>
or
   8<CMD>: MOSAIC
   Draw a mosaic around Rspot (or ALLSRL)#?: 69/zoom:2/<CR>
or
   9<CMD>: MOSAIC
   Draw a mosaic around Rspot (or ALLSRL)#?: 69/zoom:2/PPXplot<CR>
or
   10<CMD>: MOSAIC
   Draw a mosaic around Rspot (or ALLSRL)#?: 69/zoom:2/DumpPPXplot<CR>
```

Figure 3.6. Sample **cgelp2** `MOSAIC` plot. The plot file was ploted using the `LASER` display option set with `SET DISPLAY`.

After you respond with the spot number, it which forces the user to respond with which gels are to be used in constructing the mosaic. It will then generate the mosaic plot.

```
10<CMD>: MOSAIC//166/PPXPLOT
GLOBAL CMD SWITCHES: /Option:
PLOT SWITCHES: /PPXplot
Draw a mosaic around Rspot (or ALLSRL)#?: 166<CR>
List Gels with which to construct mosaic
 ?[ALL(i.e. working set)]: 324.1, 369.1, 378.2, 384.1<CR>
  .
  .
  .
```

mosaic images or plot

Alternatively, if the `/PPXplot` switch is specified, then generate a mosaic derived image and invoke **Xpix** to display it. This causes: (1) file `mosaic.sps` to be

generated, and (2) the invoking of `system("mosaic` *Rspot#* `mosaic.sps -zoom:2x -large -Xpix -name:./mosaic.ppx")`. It will draw the mosaic derived image or plot around the specified spot (if it is specified as a non-zero numeric value). If you instead supply `ALLSRL`, it will generate mosaics for all of the Rspots in the search results list. Select EXIT from the VIEW menu to terminate image display to return you to where you left off in **cgelp2**. The `/DumpPPXplot` switch will dump the image on the laser printer instead of displaying it.

```
11<CMD>: MOSAIC//166/PPXPLOT
GLOBAL CMD SWITCHES: /Option:
PLOT SWITCHES: /PPXplot
Draw a mosaic around Rspot# (or ALLSRL)?: 166<CR>
Output file:[mosaic.sps]
Creating SPSS data file from PCGL DB:
   /home/joeUser/gellab/pcg/tst.pcg
mosaic 166 mosaic.sps -zoom:1 -large -Xpix -name:./mosaic.ppx
MOSAIC - Version November 30, 1988
Today's date: 01/29/1989, 07:20:12AM
Written 1981-1988, P. Lemkin.
Magnification of gel inserts is 1X.
Labeling R-spot[166] in SPSS file:mosaic.sps
Creation date:01/29/1989, 07:20:04AM
Title:
There are 12 gels.
These are divided into 1 pictures.
Creating Rspot Image file: /home/joeUser/mosaic.ppx
Doing frames [1:9]
#1 Doing Gel 0566.1 index 315 at [290,258]
#2 Doing Gel 0524.1 index 491 at [342,320]
     .
     .
     .
#12 Doing Gel 0544.1 index 516 at [314,300]
Drawing labels in mosaic.
0566.1 L:PP D:0.37R
 -CLL
0524.1 L:PP D:0.38R
 -AML
     .
     .
     .
0544.1 L:PP D:2.78R
 -ALL
Real TIME =00:00:49  CPU TIME =00:00:33,  67.35%

Finished processing R-spot[166]: mosaic.ppx
;    Xpix -full mosaic.ppx

   ********************************************************
   * Xpix - October 23, 1988, type '?' for help.
   * Written 1987-1990, P. Lemkin.
```

```
* Position cursor over image, then press the keyboard
* CONTROL-key and mouse MIDDLE-button to get menus.
*********************************************************
     .
     .  [Select 'EXIT' menu option with mouse - see Xpix]
     .
[07:21:47AM] Real TIME =00:01:43  CPU TIME =00:00:05,  4.85%
```

In addition, you can pass additional arguments to the **mosaic** program if you hide them in the `/OPTIONS:arg` switch.

```
111<CMD>: MOSAIC/OPTION:-gel//0/PPXplot

or
112<CMD>: MOSAIC/OPTION:-gel:0524.1//0/PPXPlot
```

<CMD> *OPEN DATABASE FILE

Open a new data paged database (PCG DB) at the start of an interactive session if there is no actiave database. IF it is already opened, then use `CLOSE DATABASE FILE` first.

<CMD> PLot

Plot feature vs. feature plot of 2 (or 3) spot features over the PCG DB. The plot may be either two or three features against each other. The three dimensional plot is a 30 degree projection onto two dimensions. The following features and switches are available. Use `SET DISPLAY` to define the plot or display type to be used (as well as associated plot switches: `/ANGLE`, `/LINES`, `TITLE`, `/UGF`). Figure 3.7 shows a sample mosaic plot output.

Figure 3.7. Sample **cgelp2** `PLOT` plot. The plot file was ploted using the `LASER` display option set with `SET DISPLAY`.

```
9<CMD>: PLOT
CURRENT MAXIMUM VALUES OF FEATURES IN CGL DATABASE
  AREA          = 308.00
  DENSITY       = 137.62
  MAX-OD        = 2.19
  MIN-OD        = 0.26
  OD-DIFFERENCE = 4.80
  DP            = 99.00
  DL            = 215.00
  PIE           = 512.00
  MW            = 512.00
  CV-DENSITY    = 2.50
  #GELS/RSET    = 12.00
Plot feature f1,f2 (opt, f3) features:
[area, density, maxD, minD, ODdiff, dP, dL, pI, MW, CVD, #gels/Rset]
 ?: <CR>
```

A typical request might be,

```
        AREA,DENSITY
or
        AREA,DENSITY,MW
```

The **plotn** program may be used to draw plot files on a laser plotter. See the section on **plotn**. The PLOT command also lists the maximum values of all of these features seen in this PCG DB to date (as in the FEATURES command. If no plot is desired, just reply with <CR> to the prompt.

```
39<CMD>: PLOT
CURRENT MAXIMUM VALUES OF FEATURES IN CGL DATABASE
  AREA          = 308.00
  DENSITY       = 290.00
  MAX-OD        = 2.47
  MIN-OD        = 0.25
  OD-DIFFERENCE = 6.10
  DP            = 99.00
  DL            = 224.00
  PIE           = 512.00
  MW            = 512.00
  CV-DENSITY    = 2.50
  #GELS/RSET    = 12.00
Plot feature f1,f2 (opt., f3) features.
 [Area, Cv-density, DEnsity, DP, DL, MAx-od, MIn-od, Mw, Od-diff, Pie,
  #gels/Rset]
 ?: area,density
AREA vs. DENSITY 06/12/1989, 08:09:42AM
Printing plot file on laser printer
plotn -display:4010 ~joeUser/gellab/gen/000016.ugf | tek2psG | lpr -Plaser
Plot file: /home/joeUser/gellab/gen/000016.ugf
[08:10:55AM] Real TIME =00:01:30  CPU TIME =00:00:59,  65.56%
```

<CMD> *PROBability Plot

The probability plot is not available at this time.

<CMD> PROTect

Protect the paged CGL database for read-only use *during a session* for a database which was not originally read-only. If you want to start out with a protected database, use the `-readonly` or `-protect` switch (as in `cgelp2 -protect -d ...`). By invoking the PROTECT command again, the protection may be lifted. Several users may independently log into the computer and access the same PCG DB if they do it with the protect option. If the user sets the system to protected prior to issuing a SET DATABASE, then two users may share the same database. The user who has not protected the database may change it. For example, the first time the PROTECT command is used, it causes the following response:

sharing PCG DBs

```
        Database is protected.
```

The next time, it unprotects and re-opens access to the data base as follows:

```
23<CMD>: PROTECT
        Database is UNprotected.
        Using existing PCG paged composite gel database:
              /home/joeUser/gellab/pcg/ts3pcg.pcg
        Date created: 10/03/1988,  08:01:02 AM
        Date last session: 10/29/1988,  08:51:00 AM
```

<CMD> *REMove

Remove a gel from the CGL paged database. [Use `SET WORKING SET` at this time to effectively remove gels.]

<CMD> REOrder

Reorder the Rspot sets by density rank order. This is usually performed after changing the density mode normalization with `SET DENSITY MODE`. For example, you must reorder the PCG DB in order to use the WMW non-parametric test. Note: this changes the PCG DB contents on the disk as it is being computed so you must *not* interrupt this *atomic* operation or the database will become corrupted.

sort Rspots by density normalization

If the `/FULL` global switch is added when doing a `REORDER`, then it first checkpoints the PCG DB by 'copying' it into a file with the same name but *.ckpt* suffix. This file is removed when the operation is all done. This is implemented using the `system("cp abcpcg.pcg abcpcg.pcg.ckpt")` and `system("rm abcpcg.pcg.ckpt")` calls respectively for some PCG DB (`abcpcg` is used in this example). The following example illustrates how the Rspot sets get sorted after doing a `REORDER`.

```
21<CMD>: INQUIRE
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%).
 HELP to list CMDS?: PRINT<CR>
Print Rspot set #?[#,s,$,0 to exit]?: 27<CR>

Rspot[  27] mnXYDA=(322,151,0.11,32)+-(2.24,2.00,0.13,22) CVD=1.21 #G=11
 ACC#[Index] C   %Dens  area pkOD    D' Lbl LM  DP  DL  Dx   Dy  Xabs Yabs ODdf
------------ -  ------- ---- ----    -- --- -- ---- -- ---- ---- ---- ---- ----
0497.1[ 276] 1    0.38%   69 0.86  19.9  SP B   3.5 13 (-11, -6) (335,180) 0.61
0511.1[ 145] 3    0.25%   61 0.62   8.8  PP B   7.0 16 (-16, -3) (365,157) 0.37
0517.1[ 129] 2    0.29%   67 0.46   7.3  SP B   4.0 13 (-13, -3) (355,149) 0.32
0514.1[  87] 3    0.06%   12 0.30   1.0  SP B   4.1 13 (-12, -6) (331,136) 0.05
0384.1[ 111] 4    0.06%   31 0.40   5.2  SP B   2.2 10 (-10, -1) (309,147) 0.33
0396.1[  72] 5    0.08%   33 0.50   6.2  SP B   3.2 10 (-10,  0) (337,112) 0.35
0515.1[ 156] 2    0.02%   13 0.26   1.0  SP B   3.2  9 ( -8,  0) (357,160) 0.06
0378.2[  36] 2    0.01%   12 0.14   1.0  SP B   1.0  9 ( -8, -3) (279, 91) 0.07
0324.1[ 118] 1    0.02%   14 0.18   1.4  SP B   1.0  9 ( -9, -3) (323,150) 0.09
0393.2[  25] 4    0.03%   26 0.10   1.0  SP B   3.2 12 (-12, -2) (340,134) 0.10
0369.1[ 126] 3    0.00%   15 0.24   0.4  SP B   1.0  9 ( -9, -2) (305,140) 0.12
```

```
22<CMD>: REORDER
[09:42:59AM] Real TIME =00:00:34  CPU TIME =00:00:25,  73.53%

23<CMD>: INQUIRE
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%).
 HELP to list CMDS?: PRINT<CR>
Print Rspot set #?[#,s,$,0 to exit]?: 27<CR>

Rspot[  27] mnXYDA=(322,151,0.11,32)+-(2.24,2.00,0.13,22) CVD=1.21 #G=11
 ACC#[Index] C   %Dens  area pkOD    D' Lbl LM  DP  DL  Dx   Dy  Xabs Yabs ODdf
------------ -  ------- ---- ----    -- --- -- ---- -- ---- ---- ---- ---- ----
0497.1[ 276] 1    0.38%   69 0.86  19.9  SP B   3.5 13 (-11, -6) (335,180) 0.61
0517.1[ 129] 2    0.29%   67 0.46   7.3  SP B   4.0 13 (-13, -3) (355,149) 0.32
0511.1[ 145] 3    0.25%   61 0.62   8.8  PP B   7.0 16 (-16, -3) (365,157) 0.37
0396.1[  72] 5    0.08%   33 0.50   6.2  SP B   3.2 10 (-10,  0) (337,112) 0.35
0384.1[ 111] 4    0.06%   31 0.40   5.2  SP B   2.2 10 (-10, -1) (309,147) 0.33
0514.1[  87] 3    0.06%   12 0.30   1.0  SP B   4.1 13 (-12, -6) (331,136) 0.05
0393.2[  25] 4    0.03%   26 0.10   1.0  SP B   3.2 12 (-12, -2) (340,134) 0.10
0324.1[ 118] 1    0.02%   14 0.18   1.4  SP B   1.0  9 ( -9, -3) (323,150) 0.09
0515.1[ 156] 2    0.02%   13 0.26   1.0  SP B   3.2  9 ( -8,  0) (357,160) 0.06
0378.2[  36] 2    0.01%   12 0.14   1.0  SP B   1.0  9 ( -8, -3) (279, 91) 0.07
0369.1[ 126] 3    0.00%   15 0.24   0.4  SP B   1.0  9 ( -9, -2) (305,140) 0.12
```

<CMD> RMap

Generate a Rmap derived gel image or plot. It plots the Rmap surrounding the Rspot using density (or **sg2gii** supplied s_x, s_y) estimates on the display or *.ugf* plot file. Since the PREFILTER is active, you can restrict spots which you wish to view in the Rmap plot. It defaults to generating a Rmap for the Rgel unless you specify another gel using the `/USEGEL:`*xxxx.e* plot switch. It will draw the Rmap around the specified spot (if it is specified as a non-zero numeric value). If you instead supply `ALLGELS`, it will generate Rmaps for all of the gels in the working set of gels. Use `SET DISPLAY` to define the plot or display type to be used (as well as associated plot switches). Optionally, the `/TITLE:` switch may be used to specify different title text for the plot. If the `/PPXplot` switch is specified, then generate a Rmap image and invoke **Xpix** to display it. If the `/PPXplot` switch is specified, then generate a Rmap image and invoke **Xpix** to display it instead of generating a graphics plot. This causes: (1) file `rmap.sps` to be generated, and (2) the invoking of `system("markgel` *gelACC#* `rmap.sps -large -norestrictEP -Xpix -name:./rmap.ppx -zoom:nX")`. Select EXIT from the VIEW menu to terminate image display to return you to where you left off in **cgelp2**. The `/DumpPPXplot` switch will dump the image on the laser printer instead of displaying it. Use `SET DISPLAY` to define the plot or display type to be used. The following plot switches are active: `/CENTER:x,y`, `/COLORlabel`, `/DUMPPPXplot`, `/FILLlabeledSpots`, `/HFLIPpIe`, `/LABEL`, `/PPXplot`, `/SIzeByD'`, `/SRLlabel`, `/TITLE:'...'`, `/USEgel:acc#`, `/VFlipMW`, `/ZOOM:nX`). The default

Rmap images or plots

switches are: `/FILLlabeledSpots`, `/SRLlabel`, `/UGFlabel`, `/ZOOM:1X`. The following prompt is given the user for generating a Rmap plot:

```
25<CMD>: Rmap
Draw a Rmap (optionally around Rspot# (or ALL gels)) ?: 233/Zoom:2x<CR>
Drawing Rmap of 0524.1 at 1X with label.
Title:
Plot on display
  (TTY, 4010, VT240, XWND, PPX, PS, PLOT, 4010PLOT, VT240PLOT, XWNDPLOT,
        PPXPLOT, PSPLOT, LASER)[TTY] ?: 4010PLOT<CR>
Selecting new display [4010].

26<CMD>: Rmap
Draw a Rmap (optionally around Rspot# (or ALL gels)) ?: 233/Zoom:2x<CR>
Drawing Rmap of 0524.1 at 1X with label.
Title:
Plot on display
  (TTY, 4010, VT240, XWND, PPX, PS, PLOT, 4010PLOT, VT240PLOT, XWNDPLOT,
        PPXPLOT, PSPLOT, LASER)[4010PLOT] ?: LASER<CR>
Selecting new display [LASER].
     . . .



 47<CMD>: RMAP
 Draw a Rmap (optionally around Rspot# (or ALL gels)) ?: /SRLlabel/Zoom:2x
 Drawing Rmap of 0324.1 at 2X with nolabel.
 Title:
 Printing plot file on laser printer
  plotn -display:4010 ~joeUser/gellab/gen/000018.ugf | tek2psG | lpr -Plaser
 Plot file: /home/joeUser/gellab/gen/000018.ugf
 [08:23:54AM] Real TIME =00:01:15  CPU TIME =00:00:54,  72.00%
```

Figure 3.8 shows a sample Rmap plot output.

Figure 3.8. Sample **cgelp2** `RMAP` plot. The plot file was ploted using the `LASER` display option set with `SET DISPLAY`.

The **plotn** program may be used to draw the plots on the laser printer. Alternatively, you can generate a Rmap image and display it.

```
26<CMD>: Rmap
Draw a Rmap (optionally around Rspot# (or ALL gels)) ?: 233/PPXplot<CR>
Output file:[rmap.sps]
Creating SPSS data file from PCGL DB:
   /home/joeUser/gellab/pcg/ts3pcg.pcg
markgel 0524.1 rmap.sps -zoom:1 -large -Xpix -name:./rmap.ppx
MARKGEL :Version October 30, 1988
Today's date: 01/28/1989, 12:59:15AM
Written 1981-1988, P. Lemkin.
Magnification is 1X.
Generating a map of R-spots for gel 0524.1 from file rmap.sps.
0524.1/.../?/?/1-18-82/#12/CULT #2/3:10,5-20%/
0 HRS/H3/2 HRS/96 HRS/HEME MALIG-AML,MYELOID/
A00661/-NONE-/-NONE-/VIDICON-AUTO,28MM,F8,76CM/LESTER*
```

```
 027 049 072 095 117 136 153 168 181 192 200 208 213 220 225 000 036 497 074 509
Creation date: pcg
Title: Title:
Estimating Centroid [29 Rspots] at (279,254)
Creating Rspot Image file: /home/joeUser/rmap.ppx
R-spot[7] at (304,184)
R-spot[9] at (302,191)
R-spot[25] at (300,146)
R-spot[61] at (317,199)
R-spot[71] at (324,227)
R-spot[93] at (406,218)
R-spot[97] at (357,221)
       . . .
R-spot[445] at (210,267)
R-spot[505] at (244,329)
R-spot[533] at (280,359)
R-spot[594] at (355,381)
R-spot[649] at (262,413)
R-spot[660] at (245,448)
Real TIME =00:00:13  CPU TIME =00:00:10,  76.92%

Finished creating Rspot Image file: /home/joeUser/rmap.ppx
 Xpix -full rmap.ppx

*********************************************************
* Xpix - October 23, 1988, type '?' for help.
* Written 1987-1990, P. Lemkin
* Position cursor over image, then press the keyboard
* CONTROL-key and mouse MIDDLE-button to get menus.
*********************************************************
      .
      .  [Select 'EXIT' menu option with mouse - see Xpix]
      .
[12:59:29AM] Real TIME =00:00:35  CPU TIME =00:00:07,  20.00%
```

In addition, you can pass additional arguments to the **mosaic** program if you hide then in the `/OPTIONS:arg` switch.

```
    111<CMD>: RMAP/OPTION:-noLargeLables//0/PPXplot

or
    112<CMD>: RMAP/OPTION:-rspot:236//0/PPXPlot
or
    113<CMD>: RMAP/OPTION:-rspot:236//0/DumpPPXPlot
```

<CMD> SEQuential set operation
Compute the sequential set operations of adjacent SRL subsets using the following algorithm. The SRL subsets need to have been created previously using the `SET SRL SUBSETS` command. Using the `INTERSECTION` subcommand of the `SEQUENTIAL`

`SET OPR`, one could compute the same results but with considerably more effort in typing.

Let the number of sequential SRL subsets be n.
Let $SSRL_i$ be the i'th SRL subset.
Let Rk_i be the k'th spot in $SSRL_i$.
Let C_{ki} be the coefficient of variation of R_{ki}.
Let Rel be the relation $>$ or $<$.
Let T be a threshold.
Then,

$$SSRL_i = \{R_{ki} | (C_{ki}\ Rel\ T)\}. \tag{3.18}$$

The transitive intersection T_{jw} is defined as:

$$T_{jw} = \bigcap_{p=k}^{j+(w+1)} SSRL_p. \tag{3.19}$$

Similarly, a histogram type display similar to $m2/m1$ ratio histogram can be produced where instead of the $m2/m1$ value, we have one of the up to 9 T_{jw} entries. The following subcommands are available.

<SUBCMD> SEQUENTIAL SET OPERATION: List
List intersection $(SRL_i, SRL_j, ..., SRL_m)$ - compute working SRL as:

$$SRL = \bigcap(SRL_{i,j,...,m}). \tag{3.20}$$

<SUBCMD> SEQUENTIAL SET OPERATION: Sequential
Sequential intersection ($SRL_i\ of\ width\ w$) - compute the sequential intersection of SRL_i to SRL_{i+w-1} and store it in the SRL.

<SUBCMD> SEQUENTIAL SET OPERATION: All
All sequential intersection of width w - compute the sequential intersections of all SRL_i and print the results on the terminal as well as putting into a *.tbl* file.

Examples:
If the SRL subset lists are defined with the following Rspots,

```
Sub set list#1[T0-T24]=   48 134 234 425 437 461
Sub set list#2 [T24-T48]= 144 178 356 446
Sub set list#3 [T48-T72]= 16 105 185 186 244 446
Sub set list#4 [T72-T96]= 15 61 167 232 244 335 340 348 360 423
                          440 482
```

then,

```
5<CMD>: SEQUENTIAL SET OPR
Subcommands (SRL sublists specified as #s):
---------------------------------------
   All w - sequential <set-operation>s of (SRLSS[1] to SRLSS[w]).
   List i,j,...,m - <set-operation> on list of sets to process.
   Operator - change <set-operation> from [INTERSECTION].
   Sequential i,w - <set-operation> of SRLSS[i] to SRLSS[i+w-1].
 ?: LIST 2,3,4<CR>
 Search Results List =  446
```

<CMD> SET ACcession file name

Change the default `gel.id` accession file name. The accession file is used to acquire the gel *STUDY* information using the `SET FORMAT` command. When a PCG database is `CREATE`d all gels are initially put into class 1 (*all gels*). The study information can then be used to automatically classify the gels according to the study field membership by the classes defined by the `SET CLASSES` command. The default accession file is obtained from the `gel.rc` state file keyword `gelFile` associated entry. Normally one does not change the accession file once the database is constructed. This should *only* be done if the gels in the current database are a subset of the new accession file.[11]

accession file

```
4<CMD>: SET ACCESSION FILE
Enter accession file [gel.id]
 ?: /home/joeUser/gellab/id/gelhz1.id<CR>

5<CMD>: SET ACCESSION FILE
Enter accession file [/home/joeUser/gellab/id/gelhz1.id]
 ?: /home/joeUser/gellab/id/gelhz2.id<CR>
```

<CMD> *SET ANnotation

`PFL- REWRITE - does not correspond to GUI`

Change the associated PCG DB gel annotation.[12] An associated annotation DB file is specified by the `.map` file extension. Each Rspot set has associated up to 32 annotation fields. You may `ADD`, `EDIT` or `DELETE` an annotation feature for a particular Rspot. You can list the `TABLE` for the current annotation map. The functions available are:

annotation DB file

```
ANNotation - edit property names of Spot map properties.
```

[11] Currently, note that if moving a PCG DB between disks or systems you need to reset the accession file to the correct path. You can not currently do this with the GUI interface. So start the database without the GUI, change it, EXIT and then rerun it with the GUI interface if you wish.

[12] *Annotation* is textual information pertaining to data objects which are associated with those objects or to sets of objects.

```
 ADD annotation - feature to all SRL spots (set Fspot=Rspot if not defined).
 CLEAR annotation - clear all annotation properties.
 DELETE annotation  - feature [1:32] or by name.
 EDIT spot - annotation list of features for a specific Rspot/Fspot.
 FINDAnnotation spots - find Rspots by annotation FEATURE # [1:32] or name.
 FINDExpression spots - find Rspots by annotation feature Boolean expression
 SUBtract annotation - feature from all spots in SRL if feature was defined.
 TABle annotation - print Rspot features in a table for annotated spots.

207<CMD>: SET FOREIGN SPOT MAP
Foreign spot maps commands
 (CLEARall DEFine DELETEspot EDITSpot ENTERspot FINDProtein LISTMap LISTSpot
  Mode READmapFile WRITEmapFile)
 ?: read
Rspot Mapping file[fspot.map]?:
Reading Foreign Spot map file: fspot.map
Read 11 SPOTFEATURES and 12 Rspot<==>Fspot entries from 'fspot.map'.

208<CMD>: SET ANNOTATION
Set annotation commands
 (ANNnames ADDann CLEARall DELETEann EDITspot FINDann FINDExprAnn
  SUBann TABleAnn)
 ?: table
Computing Rspot annotation feature table...Rspot-Annotation Feature table
/home/joeUser/gellab/pcg/ts3pcg.pcg
Todays date:07/02/1993, 06:12:20PM
Rspot Feature Table
[1] Adult Human Leukemia - lymphocytes (E.Lester HM5 DB)
[2] Increased in CLL vs AML. Increased diff. in HL-60
[3] Interactively manually defined landmark spot
[4] Increased in CLL&HCL vs AML
[5] Decreased diff. in HL-60. Sequenced by Hood, etal.
[6] Sequence: P?(DR)MH(DR)N(SQ)(QJ)S(QR)WP
[7] Increased in Lymphocytes
[8] Sequence: (KA)FGADWWTLMR
[9] Increased in Lymphs. & stim. Increased in AML vs CLL
[10] Sequence: MREIVHLQAGQCGNQIGAKF
[11] Is landmark Spot

Rspot  1  2  3  4  5  6  7  8  9 10 11  Protein Name
-------------------------------------  ------------
 116   -  -  -  -  -  -  T  T  -  -  -  no-name
 144   -  -  -  -  T  T  -  -  -  -  -  no-name
 162   -  -  -  -  -  -  -  -  T  T  T  LM[G] - tubulin
 222   -  -  -  -  -  -  -  -  -  -  -  actin
 318   -  -  -  T  -  -  -  -  -  -  T  LM[L] - no-name
 365   -  T  -  -  -  -  -  -  -  -  T  LM[M] - no-name
 610   -  -  -  -  -  -  -  -  -  -  -  HLA class-I-c?
 612   -  -  -  -  -  -  -  -  -  -  -  HLA class-I-b?
 613   -  -  -  -  -  -  -  -  -  -  -  HLA class-I-a?
 659   -  -  -  -  -  -  -  -  -  -  -  cyclin?
 669   -  -  -  -  -  -  -  -  -  -  -  tropomyosin-b
 671   -  -  -  -  -  -  -  -  -  -  -  tropomyosin-a
```

```
209<CMD>: SET ANNOTATION
Set annotation commands
 (ANNnames ADDann CLEARall DELETEann EDITspot FINDann FINDExprAnn
  SUBann TABleAnn)
 ?: findann
Enter annotation feature for search, [1:32] or name?: 11
 162 318 365

Found 3 Rspots with annotation feature Is landmark Spot
```

<CMD> SET CAlibration

Calibrate density, (x) pIe and (y) MW. You select to calibrate density (as spot density/unit-area or total-density/spot). Or you define the (pIe,mw) $\Leftrightarrow$ (x,y) calibration from a calibration (*.cal*) file and set up the calibration lookup tables. MW is interpolated using a piecewise linear interpolation from a set of (y,MW) standard entries and pIE from a set of (x,pIe) standard entries.

GUI only

```
SET CAlibration - calibrating spot position (pIe,MW) as function of (x,y).
DEFINE pIe Rspots - as SRL Rspots and their pIe-X values.
DEFINE MW Rspots - as SRL Rspots and their MW-Y values.
EDIT Rspots' pIe&MW - values previously DEFINED to change a value.
READ pIe-MW - read .cal calibration file and recalibrate.
Write pIe-MW - save current calibration as .cal calibration file.
List pIe-MW - list current calibration points.
Format .cal file - print the legal format of a .cal calibration file.
CAlibrate pIe-MW - calibrate (pIe,MW) from defined (x,y) or Rspots.
CLEAR calibration - clear the old calibration so can use DEFINE again.
Unit-area(density) - set mode to report spot integrated density/unit-area.
Total-density - set mode to report spot total integrated density.
  /AReaInMM - print area in square millimeters and density/mm**2.
  /CAlibrateMWpIe - print(x,y) as (pIe,MW) if calibrated.
```

```
16<CMD>: SET CALIBRATION
Calibration subcommand (total density per spot)
 (DEFINE pIe, DEFINE MW, Edit, READ .cal, Write .cal, List, Format,
  CAlibrate, CLEAR calib, Unit-area, Total-density)
?:
```

[PFL-CHECK]

If the keyword `"MICRONS-PER-PIXEL=value"` is in the composite study which was set from the `SET FIELDS` command, then get the values for each gel from there. Density (in any mode) is then recalculated as for an individual spot i in gel g with density D, area A and square mm per pixel constant K as

$$\frac{D^u_{i,g} = D_{i,g}}{A_{i,g} K_{g.}} \tag{3.21}$$

If K_g is not known for each gel, and if a global calibration for `PixelSizeMicrons` in the `gel.rc` file is known, then K_g is computed from that value.

pIe, MW calibration

You elect to read a calibration from a file. The file should have up to 15 entries each for (x,pIe) and (y,MW) of the form:

```
(x,pIe)=35,5.02pH
(x,pIe)=120,6.02pH
    . . .
(x,pIe)=501,8.07pH
(y,MW)=16,210000Daltons
(y,MW)=121,125000Daltons
    . . .
(y,MW)=503,15000Daltons
```

<CMD> SET CLasses

Define gel class partition by defining or redefining up to 9 gel classes either manually, by gel subset or automatically by field information. The following example illustrates the use of the automatic mode. The class names are tested against the gel *STUDY* information which is specified by field using the `SET FIELD` command. If a class name appears anywhere within the study, then it is considered a match. Therefore, since several class names could match, a gel could belong to several classes. A typical example follows illustrating a 9 class system where a gel can exist in several classes using automatic classification.

experimental classes

```
8<CMD>: SET CLASSES
Class #1(ALL GELS)= 0250.2, 0250.1, 0251.1, 0251.2, 0252.1, 0252.2, 0253.1,
                    0253.2, 0254.1, 0254.2, 0255.1, 0255.2, 0256.1,
                    0256.2, 0257.1, 0257.2, 0258.1, 0258.2, 0259.1,
                    0259.2, 0260.1, 0260.2, 0261.1, 0261.2, 0262.1,
                    0262.2, 0263.1, 0263.2, 0264.1, 0264.2, 0265.1,
                    0265.2,
Class #2()=
Class #3()=
Class #4()=
Class #5()=
Class #6()=
Class #7()=
Class #8()=
Class #9()=
Change classes?(Yes/Auto/Subsets/N): AUTO<CR>
Change class names?(Y/N/Clear)  Yes<CR>
Answer with <NULL> to delete a class.
Class name[  1](ALL GELS)?: CONTROL<CR>
Class name[  2]()?: AL2O3-HC<CR>
Class name[  3]()?: AL2O3-18U<CR>
Class name[  4]()?: AMOSITE<CR>
Class name[  5]()?: T0<CR>
Class name[  6]()?: T24<CR>
```

```
Class name[  7]()?: TOXIC<CR>
Class name[  8]()?: AL2O3<CR>
Class name[  9]()?: PHAGOCYTIC<CR>
Put 0250.2 into CONTROL
Put 0250.2 into T0
Put 0250.1 into CONTROL
Put 0250.1 into T0
Put 0251.1 into CONTROL
Put 0251.1 into T0
Put 0251.2 into CONTROL
Put 0251.2 into T0
Put 0252.1 into AL2O3-HC
Put 0252.1 into T0
Put 0252.1 into TOXIC
Put 0252.1 into AL2O3
Put 0252.1 into PHAGOCYTIC
Put 0252.2 into AL2O3-HC
Put 0252.2 into T0
    .
    .
    .

Put 0262.2 into PHAGOCYTIC
Put 0263.1 into T24
Put 0263.1 into PHAGOCYTIC
Put 0263.2 into T24
Put 0263.2 into PHAGOCYTIC
Put 0264.1 into AMOSITE
Put 0264.1 into T24
Put 0264.1 into TOXIC
Put 0264.1 into PHAGOCYTIC
Put 0264.2 into AMOSITE
Put 0264.2 into T24
Put 0264.2 into TOXIC
Put 0264.2 into PHAGOCYTIC
Put 0265.1 into AMOSITE
Put 0265.1 into T24
Put 0265.1 into TOXIC
Put 0265.1 into PHAGOCYTIC
Put 0265.2 into AMOSITE
Put 0265.2 into T24
Put 0265.2 into TOXIC
Put 0265.2 into PHAGOCYTIC
```

To simply list the current classification scheme, respond NO after it prints the current scheme when doing `SET CLASSES` again.

```
9<CMD>: SET CLASSES
Class #1(CONTROL)=0250.2, 0250.1, 0251.1, 0251.2, 0258.1, 0258.2, 0259.1,
                  0259.2,
Class #2(AL2O3-HC)=0252.1, 0252.2, 0253.1, 0253.2, 0260.1, 0260.2, 0261.1,
                  0261.2,
```

```
Class #3(AL2O3-18U)=0255.1, 0255.2, 0262.1, 0262.2,
Class #4(AMOSITE)=0256.1, 0256.2, 0257.1, 0257.2, 0264.1, 0264.2, 0265.1,
                  0265.2,
Class #5(T0)=0250.2, 0250.1, 0251.1, 0251.2, 0252.1, 0252.2, 0253.1,
                  0253.2, 0254.1, 0254.2, 0255.1, 0255.2, 0256.1,
                  0256.2, 0257.1, 0257.2,
Class #6(T24)=0258.1, 0258.2, 0259.1, 0259.2, 0260.1, 0260.2, 0261.1,
                  0261.2, 0262.1, 0262.2, 0263.1, 0263.2, 0264.1,
                  0264.2, 0265.1, 0265.2,
Class #7(TOXIC)=0252.1, 0252.2, 0253.1, 0253.2, 0256.1, 0256.2, 0257.1,
                  0257.2, 0260.1, 0260.2, 0261.1, 0261.2, 0264.1,
                  0264.2, 0265.1, 0265.2,
Class #8(AL2O3)=0252.1, 0252.2, 0253.1, 0253.2, 0255.1, 0255.2, 0260.1,
                  0260.2, 0261.1, 0261.2, 0262.1, 0262.2,
Class #9(PHAGOCYTIC)=0252.1,0252.2,0253.1,0253.2,0254.1,0254.2,0255.1,
                  0255.2, 0256.1, 0256.2, 0257.1, 0257.2, 0260.1,
                  0260.2, 0261.1, 0261.2, 0262.1, 0262.2, 0263.1,
                  0263.2, 0264.1, 0264.2, 0265.1, 0265.2,
Change classes?(Y/AUTO/SUBSETS/N): NO<CR>
```

It is also possible to generate the gel classification manually by responding to a prompt for each gel. In this case, a gel can only belong to one class.

```
202<CMD>: SET CLASSES
GLOBAL CMD SWITCHES: /EPspot
Class # 1(-AML)=0524.1, 0497.1, 0505.1,
Class # 2(-CLL)=0578.2, 0515.1, 0517.1,
Class # 3(-ALL)=0569.1, 0511.1, 0514.1,
Class # 4(-HCL)=0584.1, 0593.2,
Class # 5(HL-60)=0596.1,
Class # 6(<NULL>)=
Class # 7(<NULL>)=
Class # 8(<NULL>)=
Class # 9(<NULL>)=
Change classes (Yes/Auto/Subsets/No)[N]?: YES<CR>
Change class names?(Yes/No/Clear)[N]?: NO<CR>
Put 0524.1 into class #?: 1<CR>
Put 0569.1 into class #?: 3<CR>
Put 0578.2 into class #?: 2<CR>
Put 0584.1 into class #?: 4<CR>
Put 0596.1 into class #?: 5<CR>
Put 0497.1 into class #?: 1<CR>
      . . .
```

<CMD> SET DAtabase file

old or new PCG DB

Setup the new or old CGL data paged database (PCG DB) at the start of an interactive session. Note that by changing the name of the database in mid stream, the old database will be saved before the new database is instantiated. A typical example of setting up a new database follows:

```
294<CMD>: SET DATABASE
What CGL data paged database do you want to use[]?: jan18.pcg<CR>
Creating new paged CGL database file: jan18.pcg
```

Similarly, when accessing an existing database,

```
327<CMD>: SET DATABASE FILE
What CGL data paged database do you want to use[]?: Jan14.pcg<CR>
Using existing paged CGL database: jan14.pcg
     Date created: 01/16/1981,  12:34:24 AM
     Date last session: 01/19/1981,  02:06:42 PM
```

If a user change is made in response to the prompt (i.e. you type CR), no change is made in the database. This is useful for checking on the name of database currently opened. Currently, the database can not be changed once it is opened initially during a session. Currently, you must exit the **cgelp2** program and start it again with the name of a different database if you wish to change databases.

Corrupted PCG DB

If an existing database is corrupted, this is determined by a parity check when this command is invoked and the following message is printed. Try to use a backed up version of the database file if this occurs. A database can become corrupted if you interrupt an *atomic operation* such as CREATE, REORDER, EXTRAPOLATE or your system crashes during one of these operations when writing out the PCG DB header. When one of these commands is started, it sets a *LOCK* flag in the *.pcg* file header area. When the operation is completed, it clears the flag in the file header area. [For some of these atomic commands, REORDER, the /FULL switch will cause the PCG DB to be checkpointed into a separate file prior to the operation.] This flag is checked whenever a SET DATABASE command is invoked.

trouble?

```
[SET_PCGL_DB] DRYROT! You have REAL-TROUBLE!
Your .pcg database file is corrupted. Try to get
an older copy from a backup and recover from that.
You can override this by setting DBUG:0600, then do SET DATABASE again!
```

If you do override the corrupt lock by doing DBUG:0600, it will print out the above followed by the following note. Beware that you database is probably corrupted so use any data obtained from it with this in mind.

```
Continuing... clearing the lock IN MEMORY ONLY!
```

In addition, you can invoke an even more extensive test of the PCG DB by using the VERIFY PCG DB <CMD>.

<CMD> SET DEnsity mode

Report results and perform computations of density in Absolute (D′), CPM (counts/minute), Percent, Ratio, Uncorrected, Volume or CPM (counts/min) units. The density mode is usually set to Ratio mode when the database has been normalized using the `SET RATIO LIST` command. The ratio is the mean of a set of spots. In general the same number of spots should be present in all gels. This is the optimal case if a sufficient number of normalization spots can be found. Another (somewhat less desirable) method is the least squares normalization which is initialized by `SET LEAST SQUARES` calibration. Percent density mode unfortunately normalizes the databy the sum of all spots segmented. So that in the case of noisy gels, the total density values may be considerably higher than they should be because of a large number of artifactual spots. Volume (see Section 3.18, page 452) is estimated as $V = 4.0\sqrt{\pi}D_{max}S_xS_y$. Volume is found to consistently under represent spot density by about 20% (i.e. $\frac{(D'-V)}{D'} \sim 20\%$) especially for light fuzzy spots where the (S_x, S_y) may be poorly estimated. CPM mode is currently not implemented and defaults to Absolute mode. Section 5.1.4, page 516 discusses these various normalization methods in more detail. It will not let you set the density modes to R or L if you have not performed the normalization using `SET RATIO LIST` or `SET LEAST SQUARES` calibration. The default density mode is `Percent`. If the exposure correction factor (see page 272) is set, it is used to scale D′ in all density mode calculations.

"view" of density

If the exposure correction factor *eFactor* is specifed from the accession file by `SET FIELDS` (see Section 3.8, page 272), then the `Efactor`, `Least-square` and `Ratio` modes used the D' corrected by multiplying by the *eFactor*. NOTE that `UNcorrected`, `CPM`, and `Volume` modes do not use the *eFactor*.

The command asks the following prompt:

```
15<CMD>: SET DENSITY MODE
Enter Density-Mode (exposure-corrector-factor is 'OFF')
 (Absolute, Cpm, Least-square, Percent, Ratio, Uncorrected, Volume) [L]
 ?: RATIO<CR>
```

<CMD> SET DISPLAY

Set the display plotting device to XWND (i.e. X-Windows), 4010, LASER, PLOT or a combination of both display and plot (i.e *xxx*`PLOT`). Some commands allow plotting on the `TTY` (e.g `HISTOGRAM`) whereas others, which generate 2D or 3D plots (e.g. `DDPLOT`, `DCPLOT`, `HISTOGRAM`, `MOSAIC`, `RMAP`), do not. For any of the plotting operations, you can type '`?`' in response to the prompt for specific options to print the names and default values of the plot switches which are active for that operation.

plot graphics device

The 4010 is a generic interactive Tektronix 4010/4014 type display for any "dumb" terminal which can interpret this type of graphics. The VT240 is also a generic terminal type which is intended to be used interactively. It will first

switch to 4010 emulation mode, display the plot, and then when you are finished switch back to a VT100 emulation mode for the VT240. Any terminal which is a superset of the VT240 (VT241, VT340, VT341, Visual 240, etc.) can be used here. If you specify the 4010, VT240 or XWND, it will pause at the end of drawing the plot so you can view it. When you want to continue, just type `done<CR>` to erase the screen. You can select the `PPXPLOT` display (which will have the same effect as `/PPXplot` for `MOSAIC` and `RMAP` commands). The `PS` or `PSPLOT` display will generate a PostScript *.ps* plot file which could be printed on a laser printer. If the `PLOT` suboption is selected and `/UGFlabel` is set (the default), it will put the UGF file number in the plot. The `LASER` option generates the plot file and then plots it on the system's PostScript laser printer. The default device is `laser` which is overridden by the UNIX environment variable `$LASERPRINTER`.

```
18<CMD>: SET DISPLAY
Plot on display
(4010, VT240, XWND, PPX, PS, PLOT, 4010PLOT, VT240PLOT, XWNDPLOT,
 PPXPLOT, PSPLOT, LASER)[XWND] ?: 4010<CR>

19<CMD>: SET DISPLAY
Plot on display
(4010, VT240, XWND, PPX, PS, PLOT, 4010PLOT, VT240PLOT, XWNDPLOT,
 PPXPLOT, PSPLOT, LASER)[4010] ?: 4010PLOT<CR>
Selecting new display [4010PLOT].
```

There are in addition, a number of global plot modification switches which apply to all plot type commands (`DDPLOT, DCPLOT, HISTOGRAM, MOSAIC, PLOT, RMAP`). These switches (see Table on page 172) are supplied in response to a sub-prompt from one of these commands (i.e not from global level). Note that as with other '/' style switches, they may be negated (e.g. `/NOLINE`).

The *.ugf* plot files can be viewed later or printed on a laser printer using the **plotn** program. For example, if the plot file is `000003.ugf` (these UGF files are normally automatically put on the `ppnp5x` generated file directory), then

```
87% plotn 000003.ugf -display:vt240 -wait
                      # Will redisplay the file on a VT240 terminal.

88% plotn 000003.ugf -display:4010 | tek2psG | lpq -Plaser
                      # Will print the file on an PostScript printer.

89% plotn 000003 -display:laser
                      # equivalent to the previous command.
```

<CMD> SET FIelds

Set the list of accession information fields desired for gel labeling. Each gel has associated information in the accession file. Selected parts for each gel may be strung together to denote a **cgelp2** *field*. This field can be used for automatic

accession information

classification (see SET CLASSES command) and for identifying the subject of the gels (see GELS command). Typing just <CR> with no changes does not reload the study titles from the accession file - you must enter the field numbers to make a change. If you are using an exposure correction factor with the model

$$D^e_{ig} = F^e_g D^{'}_{ig} \tag{3.22}$$

where D^e_{ig} is the eFactor corrected D' value for Rspot i for gel g, and $D^{'}_{ig}$ is the D' value for Rspot i for gel g. then you must specify one of the fields which are used to contain the entry EFACTOR=F^e_g or DECAY=F^e_g for each gel g.[13] A typical change of format follows:

```
22<CMD>: SET FIELDS
ACCESSION#/PATIENT/BIRTHDATE/RACE&SEX/EXP DATE/EXP #/CULTURE REAG/AMPH,GEL/
     1        2        3         4        5      6         7          8
INTRVL BEFR LBLNG/LBLNG ISOTOPE/DURTN LABEL/DURTN OF EXPSR/STUDY/ FILE #/
         9              10           11            12        13     14
TAPE #/OPT. BACKUP TAPE #/ CAMERA,LENS,DISTANCE/EXPERIMENTER/
   15           16                  17              18
 ? (in range 1 to 18) [12,13,]?:  2,3,10,12,13<CR>
   Changing gel study titles...
[0250.2]: /P388D1/-/C14/1 WEEK/ALUMINUM,TO,CONTROL,BOTTLE#1
[0250.1]: /P388D1/-/C14/1 DAY/ALUMINUM,TO,CONTROL,BOTTLE#1
[0251.1]: /P388D1/-/C14/1 DAY/ALUMINUM,TO,CONTROL,BOTTLE#2
[0251.2]: /P388D1/-/C14/1 WEEK/ALUMINUM,TO,CONTROL,BOTTLE#2
[0252.1]: /P388D1/-/C14/1 DAY/ALUMINUM,TO,AL2O3-HC,BOTTLE#3
[0252.2]: /P388D1/-/C14/1 WEEK/ALUMINUM,TO,AL2O3-HC,BOTTLE#3
                .
                .
                .
```

<CMD> *SET Foreign Spot Map

PFL - REWRITE - does not correspond to GUI

GUI only

Change the associated PCG DB gel foreign spot map. The Foreign Spot Map maps Rspot numbers to/from spot numbver used in other (foreign) databases.objects. An associated foreign spot map DB file is specified by the .map file extension. The concept of spot mapping files is discussed in ([LemP88d], [LemP89a]).

foreign spot map

Note that the SET MAPPING FILE command declares a Rgel mapping file to be used in creating a pre-defined Rspot numbering *prior* to using the CREATE command to create a *new* PCG DB. This command can be used in two different ways: (1) use with *.sps* file of another PCG DB to pre-declare numbering for CREATE of a new PCG DB or (2) to define a mapping table: $rMapTable[oldRspotNbr] = newRspotNbr$.

[13] Note that the Density Mode should be set to EFactor for D^e_{ig} to be used as the normalized data. It will also be used in computing the least square and ratio density modes as well. (See Section 3.8, page 270.)

That is, it (1) sets up $\{RsetPtr[CC\#] = rSpotNbr\}$ for PSUA Rgel spots derived from the CC's of SG2GII, and (2) uses externally specified mappings $\{EPspotCentroid[rSpotNbr] = (xAbs, yAbs)_{rGel}\}$. The latter can be used to find spots closest to EP spots in the Rgel.

Mapping foreign spot names to the local database names

foreign spot DBs'

GELLAB-II supports the new concept of *Fspot* or *foreign* spot. Fspots are the mapping of most spot numbers from a similar sample 2D gel database made in another laboratory to those Rspots numbers in the database under discussion. Obviously it is not one-to-one given the state of the art of inter-laboratory gel reproducibility. Therefore, not all spots would be mapped. GELLAB-II permits the use of mapping files to discourse about the Rspot data base using Fspot numbers or names. Using Fspots, GELLAB-II is able to use spot data from other sources by scanning a gel from another system. GELLAB-I was only able to read its own spot pairing files. Other gel databases can then be accessed through *mapping files* so that different Rspot numberings for different databases can be used within GELLAB-II. Previously, GELLAB-I only allowed analyses of a single database to which additional gels could be added.

- The mapping concept is upwards compatible between GELLAB-II and other systems.
- All **internal** database computations deal with system dependent spot numbers.
- All **external** user database interactions deal with **foreign** spot numbers.
- Spot numbers **from** user are converted to internal spot numbers.
- Spot numbers to be seen by user are converted **to** foreign spot numbers.
- Concept works even if spot names are not numbers - use associative map: **external-spot-names** $\Longleftrightarrow$ **internal-spot-numbers**.
- Could simultaneously maintain several foreign spot mapping tables and address a particular spot by **(database-name, spot-name)** pairs.
- A mapping file is obtained by agreement on identification of a subset of spots between two databases.

GELLAB-II $\Longleftrightarrow$ User view is of **Fspot** names
$\updownarrow$ $\updownarrow$
Rspot#s **Fspot-names (or #s)**
$\updownarrow$ $\updownarrow$
PCG-database Foreign-database(s)
$\updownarrow$
Mapping-file(s) [Rspot $\leftrightarrow$ Fspot]

Because the later stages of data reduction in GELLAB are independent of where the paired spot list data comes from, GELLAB-II could use as input paired-spot data from non-GELLAB spot pairing files using the concept of *Local Morphologic Neighborhoods* [LemP89a]. The LMN maps arbitrary spot pairing data into the Rspot set notation required by the GELLAB-II `cgelp2` database program. Data file scanners or file conversion would have to be built for particular file formats.

```
CLEAR map - clear the ENTIRE Foreign spot map properties.
DEFine spot map - entry (Rspot#, Fspot#, protein name).
DELETE spot map - entry (Rspot#, Fspot#, protein name).
EDITSpot - entry (Rspot#, Fspot#, {annotation #s}, protein name).
ENTERSpot - into the SRL and dynamic Rmap, but don't change Fspot map.
FINDProtein spots - find Rspots by protein name pattern.
LISTMap - list all (Rspot#, Fspot#, {annotation #s}, protein name) entries.
LISTSpot - list a (Rspot#, Fspot#, {annotation #s}, protein name) entries.
Mode - toggle between Rspot and Fspot mode.
READ map file - to get or merge a foreign spot map (.map) file.
WRITE map file - to save the current foreign spot map in a (.map) file.
 /DRawSpotName - draw the spot name when drawing the Fspot in Dynamic Rmap.
 /FSpot - use foreign spot instead of Rspot number.
 /RSpotFromGui - get Rspot# prompt by clicking in Dynamic Rmap else from keyboard.

207<CMD>: SET FOREIGN SPOT MAP
Foreign spot maps commands
 (CLEARall DEFine DELETEspot EDITSpot ENTERspot FINDProtein LISTMap LISTSpot
  Mode READmapFile WRITEmapFile)
 ?: read
Rspot Mapping file[fspot.map]?:
Reading Foreign Spot map file: fspot.map
Read 11 SPOTFEATURES and 12 Rspot<==>Fspot entries from 'fspot.map'.

213<CMD>: SET FOREIGN SPOT MAP
Foreign spot maps commands
 (CLEARall DEFine DELETEspot EDITSpot ENTERspot FINDProtein LISTMap LISTSpot
  Mode READmapFile WRITEmapFile)
 ?: listMap
SPOTFEATURE[1]=Adult Human Leukemia - lymphocytes (E.Lester HM5 DB)
SPOTFEATURE[2]=Increased in CLL vs AML. Increased diff. in HL-60
SPOTFEATURE[3]=Interactively manually defined landmark spot
SPOTFEATURE[4]=Increased in CLL&HCL vs AML
SPOTFEATURE[5]=Decreased diff. in HL-60. Sequenced by Hood, etal.
```

```
  SPOTFEATURE[6]=Sequence: P?(DR)MH(DR)N(SQ)(QJ)S(QR)WP
  SPOTFEATURE[7]=Increased in Lymphocytes
  SPOTFEATURE[8]=Sequence: (KA)FGADWWTLMR
  SPOTFEATURE[9]=Increased in Lymphs. & stim. Increased in AML vs CLL
  SPOTFEATURE[10]=Sequence: MREIVHLQAGQCGNQIGAKF
  SPOTFEATURE[11]=Is landmark Spot
  R[116] F[75] { 7 8} no-name
  R[144] F[92] { 5 6} no-name
  R[162] F[106] { 9 10 11} LM[G] - tubulin
  R[222] F[161] {} actin
  R[318] F[235] { 4 11} LM[L] - no-name
  R[365] F[273] { 2 11} LM[M] - no-name
  R[610] F[552] {} HLA class-I-c?
  R[612] F[549] {} HLA class-I-b?
  R[613] F[545] {} HLA class-I-a?
  R[659] F[611] {} cyclin?
  R[669] F[670] {} tropomyosin-b
  R[671] F[665] {} tropomyosin-a

214<CMD>: set for
  Foreign spot maps commands
   (CLEARall DEFine DELETEspot EDITSpot ENTERspot FINDProtein LISTMap LISTSpot
    Mode READmapFile WRITEmapFile)
   ?: findProtein
  Enter protein pattern for search?: tubulin
  Fspot[106] (Rspot[162]) is LM[G] - tubulin
  Found 1 Rspots with protein 'TUBULIN'
```

<CMD> SET Gel subset

Define or operate on a gel subset(s) from the set of all gels in the database. There may be up to 10 gel subsets having user defined names. The subsets may be defined either explicitly or implicitly by a *class name* or by the current *working set* of gels. The gel subsets may be listed deleted or have set operations performed on them to derive new gel subsets. Note that requests for sub commands are repeated to the user until no command is given (i.e. <CR>). The gel subsets may be used to redefine the working set (see `SET WORKING SET` command). For example:

```
31<CMD>: SET GEL SUBSET
Gel Subset commands
 (Classname DElete Explicit Intersection List REMOVE Subtract Union Workingset)
    ?: EXPLICIT<CR>
 Subset name?: NEW SET OF GELS<CR>
 Enter a list of gel accession numbers
    ?: 250.2,251.2,252.2<CR>
```

A set of gels can be defined by class name,

```
32<CMD>: SET GEL SUBSET
Gel Subset commands
```

```
 (Classname DElete Explicit Intersection List REMOVE Subtract Union Workingset)
    ?: CLASSNAME<CR>
Class name to be used as subset name?: CONTROL<CR>
```

You can list the current gel subsets,

```
33<CMD>: SET GEL SUBSET
Gel Subset commands
 (Classname DElete Explicit Intersection List REMOVE Subtract Union Workingset)
    ?: LIST<CR>
 Gel subset name (or # or <ALL>):
    [ 1] DARK GELS |16|
    [ 2] CONTROL GELS |4|
    [ 3] NEW SET OF GELS |3|
    Subset name?: dark gels
    Sub set list[DARK GELS]=
    0250.2 0251.2 0252.2 0253.2 0254.2 0255.2 0256.2 0257.2 0258.2
    0259.2 0260.2 0261.2 0262.2 0263.2 0264.2 0265.2
```

or alternatively,

```
 34<CMD>: SET GEL SUBSET
Gel Subset commands
  (Classname DElete Explicit Intersection List REMOVE Subtract Union Workingset)
    ?: LIST<CR>
 Gel subset name (or # or <ALL>):
    [ 1] DARK GELS
    [ 2] CONTROL GELS
    [ 3] NEW SET OF GELS
    Subset name?: 2<CR>
    Sub set list[CONTROL GELS]= 0250.2 0251.2 0258.2 0259.2
```

A subset may be created from the set union or intersection of two other gel subsets as follows.

```
35<CMD>: SET GEL SUBSET
Gel Subset commands
 (Classname DElete Explicit Intersection List REMOVE Subtract Union Workingset)
    ?:UNION<CR>
  Subset name?: BOTH<CR>
    1st subset name? CONTROL<CR>
    2nd subset name?: NEW SET OF GELS<CR>
```

A gel subset may be defined as the current working set of gels as follows.

```
36<CMD>: SET GEL SUBSET
Gel Subset commands
 (Classname DElete Explicit Intersection List REMOVE Subtract Union Workingset)
    ?: WORKING SET<CR>
 Subset # or name?: WORKING SET OF GELS<CR>
```

<CMD> SET Label

Set the 'Label' code to (`S, P, A, U, C, E, X`) used in searching and in various operations. The pairing label is part of the **cgelp2** *prefilter* discussed in Section 5.1. Pairing labels are discussed in Section 3.4 and Appendix G for **cmpgl2** and ([LipL80a], [LemP81b], [LemP83a]). If `X` is included, the eRspot database is accessible otherwise it is ignored. A typical prompt is as follows:

pairing labels

```
 31<CMD>: SET LABEL
Enter spot pairing-reliability label 'search-pattern'
  consisting of any combination of (A, S, P, U, E, C, G, X (and *))
     A is Ambiguous Pair, S is sure Pair, P is Possible Pair,
     U is Unresolved Spot, C is composite Pair,
     G is Garbage Spot, E is Extrapolated Pair,
     X allows accessing both the Rspot and the eRspot database,
     XX allows accessing ONLY the eRspot part of the database.
 [PSUX]
 ?: PAS<CR>
 New pairing search label [PAS]
```

Note the global switches `/EPspot` causes EP spots to be counted in prefilter and statistics tests as if they had zero integrated density. The `/NOEPspot` causes EP spots to be ignored when computing the mean density and area values.

<CMD> SET LEast squares calibration

A least squares approach to normalizing the set of gels to the density range of the Rgel may be used. A linear function $d'^{R}_{j} = m_j * d'_j + b_j$, with forced zero intercept using a piecewise-linear approximation from d'_j in the range of $[0 : 2b_j]$ being approximated by $2m_j$ for $b > 0$ and $m_j/2$ for $b < 0$ may be used to map density in gel j to the domain of the Rgel. The correlation coefficient is computed for fitting $(mD' + b)$ and printed for each gel. The SP+PP spots for Rspot sets meeting the sizing criteria (e.g. CV of area is a useful sized parameter) in both gels are used to estimate (m_j) for all gels and is used as the normalizing density (in OD, i.e. D′ units). When `SET DENSITY MODE` is set to `Least squares`, then the LSQ calibration is used and the printed Rspot sets have a *Ldens* normalization name. Note that the prefilter is applied when computing the LSQ normalization. For example:

density normalization

```
243<CMD>: SET LEAST SQUARES
Calibrate Least Squares density normalization
LSQ calibration doesn't exist.
Calibrate LSQ normalization?(Y/N)[N]: Y
Doing Least Squares normalization.
Fitting function f(mj,bj,D'j):      D'jR = mj*D'j + bj
which maps D'j to the range of OD in the Rgel.
Least Sq. [0524.1] m=1.000, b=0.000,  cor.coef.=1.000, #pairs=332
Least Sq. [0569.1] m=0.353, b=1.941,  cor.coef.=0.699, #pairs=247
Least Sq. [0578.2] m=0.431, b=6.492,  cor.coef.=0.665, #pairs=251
```

```
Least Sq. [0584.1] m=0.322, b=7.470,  cor.coef.=0.640, #pairs=245
Least Sq. [0596.1] m=0.373, b=7.918,  cor.coef.=0.696, #pairs=234
Least Sq. [0497.1] m=0.908, b=4.746,  cor.coef.=0.775, #pairs=208
Least Sq. [0505.1] m=2.239, b=19.767, cor.coef.=0.632, #pairs= 90
Least Sq. [0511.1] m=0.545, b=12.262, cor.coef.=0.557, #pairs=198
Least Sq. [0514.1] m=1.263, b=12.132, cor.coef.=0.680, #pairs=192
Least Sq. [0515.1] m=0.453, b=11.053, cor.coef.=0.623, #pairs=218
Least Sq. [0517.1] m=1.304, b=8.427,  cor.coef.=0.736, #pairs=208
Least Sq. [0593.2] m=0.733, b=8.300,  cor.coef.=0.713, #pairs=208
```

Thereafter, `SET DENSITY MODE` to `least squares` will cause the least squares estimate to be used. This is indicated in Rspot set print out by the header `Ldens` instead of `Rdens` (when in ratio-mode) as in the examples on page 228.

<CMD> SET MOre

The 'more' paged text facility is used to print up to one screen full of text and then prompt the user with a `[more]?`. Depending on their response, it will continue printing text one page at a time or abort the current operation. The `SET MORE` command toggles 'more' style output for the terminal on and off.

stoping display output

If the "more" switch is set then 22 lines will be printed at which point it will stop and ask `[more]?`. It waits for you to type one of the following:

<RETURN> to continue printing the next 22 lines.

q or **Q** to restart **cgelp2** <CMD> prompt.

? to print this menu.

To disable `[more]?` prompts, do the `SET MORE` <CMD> to turn it off. NOTE: until the current terminal I/O is completely debugged, you must type an extra <CR> to execute the above *more*-response commands. The default is that `SET MORE` is *not* enabled.

<CMD> SET PArameters subsets

Save and restore many **cgelp2** parameters using named sets. This could alternatively be done using the explicit commands `SET CLASSES`, `SET DENSITY MODE`, `SET DISPLAY`, `SET FIELDS`, `SET LABEL`, `SET MAPPING FILE`, `SET REGION`, `SET STATISTICS`, `SET WORKING GELS`, `global switches`. The parameters subsets are part of the PCG DB so that they are saved along with other information. The following `SET PARAMETERS SUBSETS` commands are available:

```
Delete:n         - PARAMETER subset to be specified.
Directory        - list directory ofPARAMETERS subsets titles.
List             - list PARAMETERS subsets values.
Restore:n        - current PARAMETERS from PARAMS subset n.
Save             - current PARAMETERS into PARAMS subset (to be allocated).
```

Note that although most parameters are restored, there are some subtle side effects in that some derived parameters are not restored. The following example illustrates its operation.

```
8<CMD>: SET PARAMETERS SUBSETS
PARAMS subset commands are
 (DElete, DIrectory, List, Restore, Save)
 ?: SAVE<CR>
Saving parameters in set #1
Title for parameters subset
 ?: State at time of completion of ts3cgl.gdo batch job

9<CMD>: SET LABEL
GLOBAL CMD SWITCHES: /NOEPspot

Enter Search-pattern consisting of (A, S, P, U, E, C, X (and *)) [PSU]
    A is Ambiguous Pair, S is sure Pair, P is Possible Pair,
    U is Unresolved Spot, C is composite Pair,
    E is Extrapolated Pair,
    X allows accessing the eRspot database,
    XX allows accessing ONLY the eRspot database.
 ?: PS<CR>
New pairing search label [PS]

10<CMD>: SET WORKING GELS
The current gel working set, of size 12, is:
 0324.1 0369.1 0378.2 0384.1 0396.1 0497.1 0503.1 0511.1 0514.1 0515.1
 0517.1 0393.2
Enter new working set?(Yes/No/Define/acc#s/Add/Subtract)[No]
 ?: DEFINE<CR>
Input a list of accession numbers or gel subset name [ALL]
 ?: 324.1<CR>

11<CMD>: SET PARAMETERS SUBSETS
PARAMS subset commands are
 (DElete, DIrectory, List, Restore, Save)
 ?: SAVE<CR>
Saving parameters in set #2
Title for parameters subset
 ?: PS and just Rgel

12<CMD>: SET PARAMETERS SUBSETS
PARAMS subset commands are
 (DElete, DIrectory, List, Restore, Save)
 ?: DIRECTORY<CR>
[1] STATE AT TIME OF COMPLETION OF TS3CGL.GDO BATCH JOB
[2] PS AND JUST RGEL
List parameters subset number?: 1<CR>
Rspot Mapping file [].
Map Rspots to Fspots is OFF.
Ratio-List density mode calibration EXISTS.
Image size [rowsXcolumns]=[512X512]
Least-Squares density mode calibration EXISTS.
```

```
Relative distance limits [0.00,512.00] pixels
Mn DL limits [0.00,512.00] pixels
Mn DP limits [0.00,512.00] pixels
MN area limits [0.00,10000000.00] pixels**2
MN density (Mode: L) limits [0.00,10000000.00]
Coef. variation: S.D./Mn Rset area limits [0.00,10.00]
Coef. variation: S.D./Mn Rset dens limits [0.00,10.00]
Spot OD difference limits [0.00,10.00] OD
Class difference t-Test, F-test, Rank order p-value limit is .99:
Check if # gels in Rspot set [0:1000]
(x,y) calibration file: <NONE>
pIe Rgel limits [0.00,511.00]
MW Rgel limits [0.00,511.00]
Pairing-Search-Label [PSU]

13<CMD>: SET PARAMETERS SUBSETS
PARAMS subset commands are
 (DElete, DIrectory, List, Restore, Save)
 ?: RESTORE<CR>
RESTORE parameters from subset number?: 1<CR>
Restoring PARAMS subset #1
 STATE AT TIME OF COMPLETION OF TS3CGL.GDO BATCH JOB

14<CMD>: LIMITS
Rspot Mapping file [].
Map Rspots to Fspots is OFF.
Ratio-List density mode calibration EXISTS.
Least-Squares density mode calibration EXISTS.
Image size [rowsXcolumns]=[512X512]
Relative distance limits [0.00,512.00] pixels
Mn DL limits [0.00,512.00] pixels
Mn DP limits [0.00,512.00] pixels
MN area limits [0.00,10000000.00] pixels**2
MN density (Mode: L) limits [0.00,10000000.00]
Coef. variation: S.D./Mn Rset area limits [0.00,10.00]
Coef. variation: S.D./Mn Rset dens limits [0.00,10.00]
Spot OD difference limits [0.00,10.00] OD
Class difference t-Test, F-test, Rank order p-value limit is .99:
Check if # gels in Rspot set [0:1000]
(x,y) calibration file: <NONE>
pIe Rgel limits [0.00,511.00]
MW Rgel limits [0.00,511.00]
Pairing-Search-Label [PSU]
```

<CMD> SET PRefilter limits

When in the graphical user interface mode of operation, it is useful to see all of the PREFILTER limits at one time. The `SET PREFILTER LIMITS` command presents the fields and current values for the PREFILTER limits in one popup "form" dialog. The following fields are used:

```
Relative distance? 0.00,1024.00
```

```
Mean DL? 0.00,1024.00
Mean DP? 0.00,1024.00
Mean Area? 0,10000000
Mean density (Mode:L)? 0.000,10000000.00
Coef. variation (S.D./Mn Rset) area? 0.00,10.00
Coef. variation (S.D./Mn Rset) density? 0.00,10.00
Spot (max-min) OD difference? 0.000,2.100
P-value or conf-level (1%, 5%, 10%, 20% or .99, .95, .90, .80)? 1
Check # gels in Rspot set (per class) (enter ALL or #s)? 3,1000
pIe Rgel window limits? 0.00,1023.00
MW Rgel window limits? 0.00,1023.00
Pairing label 1 or more (A, S, P, U, E, C, G, X (and *))? PASEUC
```

<CMD> SET RAtio list

Set the list of Rspots for normalizing spot densities for Ratio mode. The old ratio list may be edited (spots may be deleted) or the search results list may be used to define the new list. When `SET DENSITY MODE` is used to set ratio mode, the ratio-sum normalized value is $D^R_{ig} = (1/n)D'_{ig}/D^R_{gTot}$ where the ratio sum value $D^R_{gTot} = \sum_{i \in RatioList} D'_{ig}$ for n spots. In general, n should be the same of all gels, unless for example, a charge train of spots is used which apprears as different numbers of spots in different gels. The printed Rspot sets have a *Rdens* normalization name. Note that the prefilter is applied when computing the normalization sums. You can view the current ratio list normalization sum values by doing a `SET DENSITY MODE//RATIO` then do the `GELS` command.

density normalization

```
33<CMD>: SET RATIO LIST
Old Ratio-list [1, 5, 23, 55, 74, 210, 504, 1011]
Input list of Rspots to define Ratio-spot-list.
Use '$' to indicate use current SRL Rspots for Ratio-spot-list.
Use 'E' to Edit previous ratio-spot-list.
Use 'R' to Recompute using previous ratio-spot-list.
```

This, [1, 5, 23, 55, 74, 210, 504, 1011], is the old list. You can type in numbers or use the editing commands.

```
?: $<CR>
```

will define a new set of spots found with an `INQUIRE` type search *or* a new list may be explicitly defined as:

```
?: 6 7 96 103 104 105 107 146 239 355 356 460 492 603 654 744 745 750<CR>
  Normalizing Mean (x100) of density[0324.1]=    25.73  #Rspots/gel= 18
  Normalizing Mean (x100) of density[0369.1]=    65.60  #Rspots/gel= 18
  Normalizing Mean (x100) of density[0378.2]=    36.60  #Rspots/gel= 18
  Normalizing Mean (x100) of density[0384.1]=    26.83  #Rspots/gel= 18
  Normalizing Mean (x100) of density[0396.1]=    33.94  #Rspots/gel= 18
  Normalizing Mean (x100) of density[0497.1]=    30.86  #Rspots/gel= 18
  Normalizing Mean (x100) of density[0503.1]=     3.52  #Rspots/gel= 18
```

```
Normalizing Mean (x100) of density[0511.1]=    16.77  #Rspots/gel= 18
Normalizing Mean (x100) of density[0514.1]=    10.68  #Rspots/gel= 18
Normalizing Mean (x100) of density[0515.1]=    22.03  #Rspots/gel= 18
Normalizing Mean (x100) of density[0517.1]=    14.83  #Rspots/gel= 18
Normalizing Mean (x100) of density[0393.2]=    15.42  #Rspots/gel= 18
  .
  .
  .
```

changing list of Rspots

If the `Edit` option is used, then spots may be pruned from the ratio spot list which currently exists by answering `Yes/No` to a prompt on whether to use each of the current ratio list spots. The Search Results List is also set to the edited list which permits it to be saved as a SRL SUBSET or used to make a SPSS file for generating Rmap images.

```
35<CMD>: SET RATIO LIST
Old Ratio-list [6,7,96,103,104,105,107,146,239,355,356,460,492,
                603,654,744,745,750]
Input list of Rspots to define Ratio-spot-list.
Use '$' or '*'to indicate use current SRL Rspots for Ratio-spot-list.
Use 'E' to Edit previous ratio-spot-list.
Use 'R' to Recompute using previous ratio-spot-list.
  ?: EDIT<CR>
Include Rspot     6 (Y/N)[Y]?:n<CR>
Include Rspot     7 (Y/N)[Y]?:n<CR>
Include Rspot     96 (Y/N)[Y]?:y<CR>
  .
  .
  .
```

<CMD> SET REGion

restricting (pIe,MW)

Set the pIe and MW region window. This window is applied to the position of the spot in the Rspot set belonging to the Rgel. It may also be set using the X-Window graphical interface using the region of interest (see mouse bindings). If the spot is in the window, then it can be used for the operation (search, etc.) otherwise if it is not in the window, then the set will be ignored. If the Rgel spot is missing from the Rspot set, then the test is always considered to succeed. The default window is the size of the image - i.e. either 1024x1024 or 512x512 pixels as determined by the GCF data used at the time the PCG DB was constructed. This means the limits would be [0:1023] or [0:511]. Then MW and pIe are expressed as pixel numbers in this range, when there is no calibration. If there is a calibration (see `SET CALIBRATION`), then it would be in the calibration (pIe,MW) units. Note that by observing the position of Rgel Rspots in the corners of a region, the extent of the domain can be deduced. The `LIMITS` command may also be used to find out the current pIe and MW regions.

```
39<CMD>: SET REGION
Answer '@' to backup to previous Q&A question.
Answer '#' to exit Q&A immediately.
pIe Rgel limits are[.00,511.00]: 123,198
MW Rgel limits are[.00,511.00]:  225,429
```

It is also possible to set a single feature's limits using the `/OPTION:` switch as follows. Note that if you type a bad keyword, it will tell you what the legal keywords are.

```
23<CMD>: SET REGION/Option:isoelectric
GLOBAL CMD SWITCHES: /Option:ISOELECTRIC
Answer '@' to backup to previous Q&A question.
Answer '$' to exit Q&A immediately.
Illegal keyword. Pick one of
    pie mw
with 'SET REGION/Option:<key>=<value-list>

24<CMD>: SET REGION/Option:pIe=233,412
GLOBAL CMD SWITCHES: /Option:PIE=233,412
Answer '@' to backup to previous Q&A question.
Answer '$' to exit Q&A immediately.
pIe Rgel window limits [0.00,511.00]?:

25<CMD>: !!
24: SET REGION/Option:pIe=233,412
GLOBAL CMD SWITCHES: /Option:PIE=233,412
Answer '@' to backup to previous Q&A question.
Answer '$' to exit Q&A immediately.
pIe Rgel window limits [233.00,412.00]?:

26<CMD>: SET REGION/Option:MW=122,501
GLOBAL CMD SWITCHES: /Option:MW=122,501
Answer '@' to backup to previous Q&A question.
Answer '$' to exit Q&A immediately.
MW Rgel window limits [0.00,511.00]?:

27<CMD>: LIMITS
Rspot Mapping file [].
Map Rspots to Fspots is OFF.
Ratio-List density mode calibration EXISTS.
Least-Squares density mode calibration EXISTS.
Image size [rowsXcolumns]=[512X512]
Relative distance limits [0.00,512.00] pixels
Mn DL limits [0.00,512.00] pixels
Mn DP limits [0.00,10.00] pixels
MN area limits [25.00,10000000.00] pixels**2
MN density (Mode: R) limits [3.00,10000000.00]
Coef. variation: S.D./Mn Rset area limits [0.00,10.00]
Coef. variation: S.D./Mn Rset dens limits [0.00,10.00]
Spot OD difference limits [0.00,10.00]
Class difference t-Test, F-test, Rank order p-value limit is .90:
```

```
Check if # gels in Rspot set [0:1000]
(x,y) calibration file: <NONE>
pIe Rgel limits [233.00,412.00] pixels
MW Rgel limits [122.00,501.00] pixels
Pairing-Search-Label [PSEAU]

28<CMD>:
```

<CMD> SET RGel

reference gel

Print the name of the Rgel. If it was not defined, you may define it the first time for a new database. Note that if you forgot to set it when building the initial PCG DB with the `CREATE` command, it will set the Rgel from the first GCF input file. The `LIMITS` command also reports the name of the Rgel. Once a database has been built, you should not change the Rgel. The output appears as:

```
201<CMD>: SET RGEL
 Enter Rgel [(null)] ?: 0250.2<CR>

202<CMD>: SET RGEL
Enter Rgel [0250.2] ?: <CR>
```

Where to redefine it one might type:

```
203<CMD>: SET RGEL
Enter Rgel [0250.2] ?: 251.2<CR>
```

<CMD> SET SRl subsets

sets of Rspots

Define or operate on a Search Results List(s) (SRL) either from the working SRL or an operation on existing SRL subs in the data base. The up to 88 SRL subsets may be defined which have user assigned names and are assigned the next-free subset numbers by **cgelp2**. SRL subsets may be referenced by either name or number. The subsets may be defined either explicitly or implicitly by a *subset name, subset number* or the current SRL from the last **cgelp2** command. The subsets may be listed or deleted as well as having set operations performed on them. The subset command prompt is repeated until the user specifies no command (i.e. <CR>).

Several keywords may be used to refer to SRL subsets. <`ALL`> stands for all existing SRL subsets in the range of 1 to 88. <`LAST`> stands for the last SRL subset created using the `ASSIGN` subcommand. <`LOS`> stands for the List of SRL subsets created using the `FINDKEYWORD`, `QUERYRSPOT` or `EXPLICIT/LISTOFSRLS` subcommands.

List of SRL Subsets (LOS)

This latter concept of *list of SRL subsets* is somewhat tricky but is very useful and worth the effort to understand. Some subcommands such as `FINDKEYWORD`,

`QUERYRSPOT` and `EXPLICIT/LISTOFSRLS` are used to find a list of SRL subset *names* (in terms of SRLSS[i] numbers). For example, we can talk about SRL subsets by their *titles* or SRLSS[i] numbers. In the following examples we refer to SRL subsets titles `TO-T24`, `T48-T72` and `T72-T96` which are SRLSS[1], SRLSS[3] and SRLSS[4]. The list of SRL subsets, `LOS`, would be (1,3,4). The `DIRECTORY/LISTOFSRLS` subcommand can be used to print the names in the current list of SRL subsets. The `LIST/LISTOFSRLS` will print the contents of the the SRL subsets in the list of SRL subsets.

The power in using the LOS is in some of the other operations such as `UNION/LISTOFSRLS`, `INTERSECTION/LISTOFSRLS` or `SUBTRACT/LISTOFSRLS` which are normally binary operations. Other operations which take an explicit SRL subset or a range of SRL subset numbers (such as `READ/LISTOFSRLS`, `WRITE/LISTOFSRLS`, or `SPSS/LISTOFSRLS`) can now use the <LOS> argument instead. For example, let the LOS be a list of n SRL subsets $(p, q, \cdots, t)$ such that $f(1) = a$, $f(2) = b$, $\cdots$ $f(n) = z$. Then, UNION (i.e. $\bigcup$), INTERSECTION (i.e. $\bigcap$) and SUBTRACT would be:

$$\bigcup_{i=1}^{n} SRL(f(i)) = SRL_a \bigcup SRL_b \bigcup \cdots \bigcup SRL_z,$$

$$\bigcap_{i=1}^{n} SRL(f(i)) = SRL_a \bigcap SRL_b \bigcap \cdots \bigcap SRL_z,$$

and

$$Sub_{i=1}^{n} SRL(f(i)) = SRL(f(1)) - SRL(f(2)) - SRL(f(3)) \cdots - SRL(f(n)).$$

or,

$$Sub_{i=1}^{n} SRL(f(i)) = SRL_a - SRL_b - SRL_c \cdots - SRL_z.$$

respectively.

The `FINDKEYWORD` subcommand is a way of generating a list of SRL subsets which matches a specified keyword expression to each SRL subsets title string. Examples of these concepts will be given in some of the following examples.

This command may be used to assign the working SRL to a SRL subset. *assigning SRL*

```
101<CMD>: SET SRL SUBSET
SRL subset commands are
(Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
 Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
 RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
 Union[/ListOfSRLs], Write[:Filename])
 ?: ASSIGN<CR>
Subset name?: T24-T48<CR>
```

or alternatively an explicit list of Rspot numbers may be assigned.

defining explicit SRL

```
102<CMD>: SET SRL SUBSET
SRL subset commands are
(Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
 Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
 RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
 Union[/ListOfSRLs], Write[:Filename])
 ?: EXPLICIT<CR>
Subset name?: T24-T48<CR>
Input list of Rspot set numbers
  ?: 144, 178, 356, 446<CR>
```

A SRL subset may be used to preload working SRL for which other **cgelp2** operations may be performed.

```
103<CMD>: SET SRL SUBSET
SRL subset commands are
(Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
 Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
 RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
 Union[/ListOfSRLs], Write[:Filename])
 ?: RESTORE<CR>
Subset name (or # or # range or <ALL> for list cmd)?: T24-T48<CR>
```

listing subsets

You can list the names and sizes of all SRL subsets and then particular subsets. This is useful for reviewing the current stat of the SRL subsets database.

```
104<CMD>: SET SRL SUBSET
SRL subset commands are
(Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
 Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
 RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
 Union[/ListOfSRLs], Write[:Filename])
 ?: DIRECTORY<CR>
SRL sub names:
 [ 1] T0-T24  | 6|
 [ 2] T24-T48 | 4|
 [ 3] T48-T72 | 6|
 [ 4] T72-T96 |12|
Subset name (or # or # range or <ALL> for list cmd)?: T24-T48<CR>
 Set #2 <<<T24-T48>>>= 144 178 356 446
```

Note that you only have to type enough of the SRL subset name to make it unique. There is a generic name for all subsets called <ALL> which stand for 1-88 inclusive. There is also a generic name for the last subset created called <LAST>.

```
105<CMD>: SET SRL SUBSET
SRL subset commands are
(Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
 Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
```

```
RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
Union[/ListOfSRLs], Write[:Filename])
?: LIST<CR>
Subset name (or # or # range or <ALL> for list cmd)?: <ALL><CR>
 Set #1<<<TO-T24>>>= 48 134 234 425 437 461
 Set #2<<<T24-T48>>>= 144 178 356 446
 Set #3<<<T48-T72>>>= 16 105 185 186 244 446
 Set #4<<<T72-T96>>>= 15 61 167 232 244 335 340 348 360 423 440 482
```

A subset may be created from the set union, intersection or set subtraction of two other SRL subsets as follows.

subset operations

```
106<CMD>: SET SRL SUBSET
SRL subset commands are
(Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
 Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
 RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
 Union[/ListOfSRLs], Write[:Filename])
 ?: UNION<CR>
1st subset name?: T24-T48<CR>
2nd subset name (or SRL)?: T48-T72<CR>
Result in working SRL.
Search Results List =  16 105 144 178 185 186 244 356 446

107<CMD>: SET SRL SUBSET
SRL subset commands are
(Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
 Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
 RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
 Union[/ListOfSRLs], Write[:Filename])
 ?: INTERSECTION<CR>
1st subset name?: T48-T72<CR>
2nd subset name (or SRL)?: T72-T96<CR>
Result in working SRL.
 Search Results List =  244

108<CMD>: SET SRL SUBSET
SRL subset commands are
(Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
 Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
 RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
 Union[/ListOfSRLs], Write[:Filename])
 ?: SUBTRACT<CR>
1st subset name?: T48-T72<CR>
2nd subset name (or SRL)?: T72-T96<CR>
Result in working SRL.
Search Results List =  16 105 185 446
```

A SRL subset may be deleted as follows.

deleting a set

```
109<CMD>: SET SRL SUBSET
SRL subset commands are
```

```
(Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
 Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
 RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
 Union[/ListOfSRLs], Write[:Filename])
 ?: DELETE<CR>
Subset name (or # or # range or <ALL> for list cmd)?: T24-T48<CR>
Deleting Set #4 <<<T24-T48>>>.
```

groups of sets

The LIST and DELETE subcommands allow the user to specify a range of subset numbers as illustrated by the following examples.

List the directory but no individual SRL subset,

```
110<CMD>: SET SRL SUBSET
SRL subset commands are
(Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
 Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
 RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
 Union[/ListOfSRLs], Write[:Filename])
 ?: DIRECTORY<CR>
SRL subset names:
[ 1] NORMALIZATION |44|
[ 2] LANDMARKS |23|
[ 3] 1VS3F01 |179|
[ 4] 1VS3R01 |41|
  .
  .
  .
[38] [S4] |302|
[39] U(1,2,3) |315|
[40] U(4,5,6) |262|
Subset name (or # or # range or <ALL> for list cmd)?: <CR>
```

List SRL subsets 7 through 9,

```
111<CMD>: SET SRL SUBSET
SRL subset commands are
(Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
 Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
 RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
 Union[/ListOfSRLs], Write[:Filename])
 ?: LIST<CR>
Subset name (or # or # range or <ALL> for list cmd)?: 7-9<CR>
Set #7<<<INTERSECT5AND6>>>= 74 75 146 189 192 234 274 395 493 539 582
       585 595 610
Set #8<<<MISSING1VS3>>>= 43 94 399 436 467 521 558 586 608 659 662 663
       665 669 670 671
Set #9<<<MISSING 1VS4>>>= 5 6 17 28 167 177 201 399 434 477 536 558 580
       586 659 662 665 669 677
```

Delete subsets 38 to 40,

```
112<CMD>: SET SRL SUBSET
SRL subset commands are
(Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
 Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
 RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
 Union[/ListOfSRLs], Write[:Filename])
 ?: DELETE<CR>
Subset name (or # or # range or <ALL> for list cmd)?: 38-40<CR>
Deleting Set #38<<<[S4>>>
Deleting Set #39<<<U(1,2,3)>>>
Deleting Set #40<<<U(4,5,6)>>>
```

SPSS data

The SPSS command may be invoked for a specific SRL subset. It will create an SPSS file with the 3-character project prefix of the *.pcg* file followed by S followed by the SRL subset number with leading 0 if needed. If the /SAS switch is added to the top level SET SRL SUBSETS command, then a SAS file with *.sas* extension is generated instead of the SPSS file. SPSS is the Statistical Package for the Social Services [NieH75]. SAS is the Statistical Analysis System [SAS85]. In addition if the /MOSAIC switch is also specified as SPSS/MOSAIC, a UNIX batch job with the same file name but .do file extension (rather than *.sps* for SPSS file) is generated. This batch job calls the **mosaic** program with the names of the Rspots in the SRL subset in order to make mosaic images for those spots. For example:

mosaic batch scripts

```
113<CMD>: SET SRL SUBSET
SRL subset commands are
(Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
 Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
 RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
 Union[/ListOfSRLs], Write[:Filename])
 ?: SPSS/MOSAIC/RMAP<CR>
Subset name (or # Or # range Or <ALL> Or <LAST> Or SRLSS[i])?: 2
Restoring SRL from subset [2] with 23 spots.
<<<LANDMARKS SET OF SPOTS>>>
Output file:[/home/joeUser/gellab/gen/ts3s02.sps]
Creating SPSS data file from PCGL DB:
   /home/joeUser/gellab/pcg/ts3pcg.pcg
Creating Batch script job: '/home/JoeUser/ts3s02m.do'
to compute MOSAIC images.
Output file:[/home/JoeUser/ts3s02m.do]
Creating Batch script job: '/home/JoeUser/ts3s02r.do'
to compute RMAP images.
Output file:[/home/JoeUser/ts3s02r.do]
```

The batch script files looks like:

```
#!/bin/csh -v
#JOB /home/joeUser/ts3s02m.do - 08/17/1990, 08:50:38AM
#   Create mosaics for /home/joeUser/gellab/pcg/ts3pcg.pcg
# SRL subset[2]
# SRL title:LANDMARKS SET OF SPOTS
```

```
mosaic 7   ts3s02.sps -noGraphScale -Zoom:2X
mosaic 54  ts3s02.sps -noGraphScale -Zoom:2X
mosaic 88  ts3s02.sps -noGraphScale -Zoom:2X
    . . .
mosaic 688 ts3s02.sps -noGraphScale -Zoom:2X
mosaic 720 ts3s02.sps -noGraphScale -Zoom:2X
mosaic 750 ts3s02.sps -noGraphScale -Zoom:2X

echo "To display these run"
echo "/home/joeUser/ts3s02-disp-mosaics.do"

#!/bin/csh -v
#JOB /home/joeUser/ts3s02r.do - 08/17/1990, 08:50:38AM
#   Create rmaps for /home/joeUser/gellab/pcg/ts3pcg.pcg
# SRL subset[2]
# SRL title:LANDMARKS SET OF SPOTS
markgel 0324.1 ts3s02.sps -noGraphScale
markgel 0369.1 ts3s02.sps -noGraphScale
markgel 0378.2 ts3s02.sps -noGraphScale
markgel 0384.1 ts3s02.sps -noGraphScale
markgel 0396.1 ts3s02.sps -noGraphScale
markgel 0497.1 ts3s02.sps -noGraphScale
markgel 0503.1 ts3s02.sps -noGraphScale
markgel 0511.1 ts3s02.sps -noGraphScale
markgel 0514.1 ts3s02.sps -noGraphScale
markgel 0515.1 ts3s02.sps -noGraphScale
markgel 0517.1 ts3s02.sps -noGraphScale
markgel 0393.2 ts3s02.sps -noGraphScale

echo "To display these run"
echo "/home/joeUser/ts3s02-disp-rmaps.do"

#!/bin/csh -v
#JOB /home/joeUser/ts3s02-disp-mosaics.do - 08/17/1990, 08:50:38AM
#   Display Rmap and mosaics for /home/joeUser/gellab/pcg/ts3pcg.pcg
# SRL subset[2]
# SRL title:LANDMARKS SET OF SPOTS
markgel 0324.1 ts3s02.sps -noGraphscale
accppx -p1:m 0324.1 w00007
accppx -p1:m 0324.1 w00054
accppx -p1:m 0324.1 w00088
accppx -p1:m 0324.1 w00097
accppx -p1:m 0324.1 w00105
accppx -p1:m 0324.1 w00142
accppx -p1:m 0324.1 w00151
accppx -p1:m 0324.1 w00174
accppx -p1:m 0324.1 w00209
accppx -p1:m 0324.1 w00234
accppx -p1:m 0324.1 w00240
accppx -p1:m 0324.1 w00316
accppx -p1:m 0324.1 w00355
accppx -p1:m 0324.1 w00492
accppx -p1:m 0324.1 w00555
```

```
accppx -p1:m 0324.1 w00556
accppx -p1:m 0324.1 w00603
accppx -p1:m 0324.1 w00624
accppx -p1:m 0324.1 w00646
accppx -p1:m 0324.1 w00657
accppx -p1:m 0324.1 w00688
accppx -p1:m 0324.1 w00720
accppx -p1:m 0324.1 w00750

#!/bin/csh -v
#JOB /home/joeUser/ts3s02-disp-rmaps.do - 08/17/1990, 08:50:38AM
#   Display Rmap and rmaps for /home/joeUser/gellab/pcg/ts3pcg.pcg
# SRL subset[2]
# SRL title:LANDMARKS SET OF SPOTS
accppx -prefix:m 0324.1 0369.1
accppx -prefix:m 0324.1 0378.2
accppx -prefix:m 0324.1 0384.1
accppx -prefix:m 0324.1 0396.1
accppx -prefix:m 0324.1 0497.1
accppx -prefix:m 0324.1 0503.1
accppx -prefix:m 0324.1 0511.1
accppx -prefix:m 0324.1 0514.1
accppx -prefix:m 0324.1 0515.1
accppx -prefix:m 0324.1 0517.1
accppx -prefix:m 0324.1 0393.2
```

The SPSS file has the search banner generated by `SET SRL SUBSETS` to be the name of the SRL subset. The format is as follows:

```
File: /home/joeUSer/gellab/gen/ts3s02.sps 08/17/1990, 08:50:29AM from:
PCG DB: /home/joeUser/gellab/demo/pcg/ts3pcg.pcg
Title:SRLSS[2]=LANDMARKS SET OF SPOTS

Rspot Mapping file [].
Map Rspots to Fspots is OFF.
Ratio-List density mode calibration EXISTS.
Least-Squares density mode calibration EXISTS.
Image size [rowsXcolumns]=[512X512]
Relative distance limits [0.00,1024.00] pixels
Mn DL limits [0.00,1024.00] pixels
Mn DP limits [0.00,1024.00] pixels
MN area limits [0.00,10000000.00] pixels**2
MN density (Mode: L) limits [0.000,10000000.000]
Coef. variation: S.D./Mn Rset area limits [0.00,10.00]
Coef. variation: S.D./Mn Rset dens limits [0.00,10.00]
Spot OD difference limits [0.000,10.000] OD
Class difference t-Test, F-test, Rank order p-value limit is .99:
Check if # gels in Rspot set [0:1000]
(x,y) calibration file: <NONE>
pIe Rgel limits [168.00,360.00] pH
MW Rgel limits [0.00,511.00] Daltons
Pairing-Search-Label [PSUX]
```

```
Using least square normalization.

Class # 1(AML)=0324.1 0497.1 0503.1
Class # 2(ALL)=0369.1 0511.1 0514.1
Class # 3(CLL)=0378.2 0515.1 0517.1
Class # 4(HCL)=0384.1 0393.2
Class # 5(HL-60)=0396.1
Class # 6(<NULL>)=
Class # 7(<NULL>)=
Class # 8(<NULL>)=
Class # 9(<NULL>)=
RSPOT# ACC# INDEX DENSL D' LABEL[0:5] LMSET[1:25] DP DL DX DY XABS YABS CLASS
  7 0503.1   9  1.00  4.60   1   1 0.0   0   0   0 272 146 1
  7 0514.1 328  5.00 13.50   1   1 0.0   0   0   0 315 183 2
  7 0517.1 376 11.44 22.30   1   1 0.0   0   0   0 344 185 3
  7 0393.2  61 30.54 28.40   1   1 0.0   0   0   0 325 168 4
  7 0515.1 337 32.39 24.30   1   1 0.0   0   0   0 337 203 3
  7 0324.1 301 35.00 35.00   1   1 0.0   0   0   0 302 192 1
  7 0511.1 421 53.00 24.40   1   1 0.0   0   0   0 353 194 2
      . . .
 54 0393.2  28  0.53  0.70   1   2 0.0   0   0   0 352 136 4
 54 0517.1 270  1.58  4.20   1   2 0.0   0   0   0 369 150 3
 54 0514.1 224  3.19  8.60   1   2 0.0   0   0   0 344 143 2
 54 0396.1  67 13.91  9.50   1   2 0.0   0   0   0 347 112 5
 54 0515.1 248 20.71 17.80   1   2 0.0   0   0   0 365 161 3
 54 0324.1 198 27.20 27.20   1   2 0.0   0   0   0 333 154 1
 54 0511.1 340 36.83 19.40   1   2 0.0   0   0   0 382 161 2
      . . .
```

.srl *files*

The SRL subsets may be written out to a *.srl* text file which may be printed, edited or later read back into the database in whole or in part. The command to write out the entire SRL subset data base is:

```
114<CMD>: SET SRL SUBSET
SRL subset commands are
(Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
 Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
 RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
 Union[/ListOfSRLs], Write[:Filename])
 ?: WRITE<CR>
Subset name (or # or # range or <ALL> for list cmd)?: 1-88<CR>
 Creating SRL subset data Output file:(727s01.srl):
 From PCG DB: 727pcg.pcg
```

After deleting space of one interpretation of SRL subsets in the PCG DB, you can bring in selected older SRL subsets with the same set index numbers. Or if you prefer, have the system inform you of any SRL subset naming conflicts and ask what the new number should be. For example to restore the entire SRL subset database, CLEAR the SRL subsets then,

```
115<CMD>: SET SRL SUBSET
```

```
   SRL subset commands are
   (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
    Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
    RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
    Union[/ListOfSRLs], Write[:Filename])
    ?: READ<CR>
   Subset name (or # or # range or <ALL> for list cmd)?: 1-88<CR>
   What is the name of the SRL input file?: 727s01.srl<CR>
    NEW Set #1 <<<NORMALIZATION>>>
    NEW Set #2 <<<LANDMARKS>>>
    NEW Set #3 <<<1VS3F01>>>
    NEW Set #4 <<<1VS3R01>>>
        .
        .
        .
 Finished reading 727s01.srl
```

In the following example, SRL subsets 3 to 5 are deleted. We then read back sets 1 to 7. Those sets with the same SRL subset numbers which are not empty cause a prompt to be made to the user to decide whether to really replace them or move them to another set number.

```
   116<CMD>: SET SRL SUBSET
   SRL subset commands are
   (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
    Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
    RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
    Union[/ListOfSRLs], Write[:Filename])
    ?: READ<CR>
   Subset name (or # or # range or <ALL> for list cmd)?: 1-7<CR>
   What is the name of the SRL input file?: 727s01<CR>
   EXISTING SRL Set #[1]
    OLD <<<NORMALIZATION>>>
   Replace with:
    NEW <<<NORMALIZATION>>>
   Next free Set #67
    (Y/N/# of new SRL Set to use)?[N]: 67
   EXISTING SRL Set #[2]
    OLD <<<LANDMARKS>>>
   Replace with:
    NEW <<<LANDMARKS>>>

   Next free Set #68
    (Y/N/# of new SRL Set to use)?[N]: 68
    NEW Set #3 <<<1VS3F01>>>
    NEW Set #4 <<<1VS3R01>>>
    NEW Set #5 <<<INTERSECT3AND4>>>
   EXISTING SRL Set #[6]
    OLD <<<1VS4R01>>>
   Replace with:
    NEW <<<1VS4R01>>>
```

```
 (Y/N/# of new SRL Set to use)?[N]: n
EXISTING SRL Set #[7]
 OLD <<<INTERSECT5AND6>>>
Replace with:
 NEW <<<INTERSECT5AND6>>>
 (Y/N/# of new SRL Set to use)?[N]: y
Finished reading 727s01.srl
```

`.srl` *file format*

The *.srl* file format is as follows:

```
File: 727s01.srl 08/16/1982, 09:30:12 PM from: 727pcg.pcg
Pairing labels: PSUX
Density Mode: R
Relative distance limits are[ .00, 512.00] pixels
Mn DL limits are[ .00, 512.00] pixels
Mn DP limits are[ .00, 512.00] pixels
MN area limits are[ .00, 10000000.00] pixels**2
MN density limits are[ .00, 10000000.00]
Coef. variation: S.D./Mn Rset area limits are[ .00, 10.00]
Coef. variation: S.D./Mn Rset dens limits are[ .00, 10.00]
Spot OD difference limits are[ .00, 10.00]
Class difference t-test, F-test, Rank order significance limit is .99:
Check if # gels in Rspot set [0:100]
(x,y) calibration file: <NONE>
pIe Rgel limits are[ 110.00, 511.00] pixels
MW Rgel limits are[ .00, 511.00] pixels
Using least square normalization.
Class #1(AML)=0524.1, 0524.2, 0524.3, 0525.1, 0525.2, 0525.3, 0526.1,
               0526.2, 0526.3,
Class #2(ALL)=0569.1, 0569.2, 0569.3,
Class #3(CLL)=0574.1, 0574.2, 0574.3, 0578.2, 0578.3,
Class #4(HCL)=0582.1, 0582.2, 0582.3, 0584.1, 0584.2, 0584.3,
Class #5(<NULL>)=
Class #6(<NULL>)=
Class #7()=
Class #8()=
Class #9()=
Set #1<<<NORMALIZATION>>>= 10 13 57 63 67 72 74 75 78 92 97 100 105
 109 111 118 150 151 154 159 160 162 181 189 190 194 381 395 456 466
 482 485 493 509 523 550 553 554 582 583 587 593 625 649
Set #2<<<LANDMARKS>>>= 10 29 57 67 75 97 105 126 128 160 181 190 234
 274 381 436 485 509 542 553 587 625 649
Set #3<<<1VS3F01>>>= 3 5 8 10 12 17 20 21 23 24 25 28 29 31 32 37
 48 57 61 63 65 69 72 73 74 75 76 77 78 79 80 82 91 97 98 100 101
 102 105 106 107 108 109 111 112 113 116 118 120 121 124 126 128 143
 146 148 150 155 160 161 168 169 170 172 174 176 180 181 185 189 190
 191 192 193 194 196 201 203 226 229 230 231 232 233 234 235 241 243
                .
                .
                .
```

The range specification can be used with the `DIRECTORY` subcommand as well in

order to list part of the SRL directory. The following lists only a portion of the subset directory.

```
117<CMD>: SET SRL SUBSET
SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
  Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, READ,
  RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?: DIRECTORY:3-6
SRL subset names:
  [ 3] F-test 3,4 AT .99, CV AREA<.5, MIN 3 GELS/CLASS | 2|
  [ 4] MISSING CLASS-test 3,4 AT .99, CV AREA<.5, MIN 3 GELS/CLASS |69|
  [ 5] F-test 5,6 AT .99, CV AREA<.5, MIN 3 GELS/CLASS |30|
  [ 6] MISSING CLASS-test 5,6 AT .99, CV AREA<.5, MIN 3 GELS/CLASS |27|
```

When building a PCG DB with many SRL subsets, it is useful to be able to locate SRL subsets by associative context. Two relational database type commands are available: `FINDKEYWORD` and `QUERYRSPOT`. The former is used to find SRL subsets which have a keyword expression as part of the name of those subsets. The following lists the names of subsets which contain the study word `control`.

querying SRL subsets by pattern

```
118<CMD>: SET SRL SUBSET
SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
  Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, READ,
  RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?: FINDKEYWORD<CR>
Enter search key phrases composed with '~', '^', '|', '(', ')'
 ?: CONTROL<CR>
SRL subset names:
  [11] [S1] CONTROL VS AMOSITE ASBESTOS, INDUCED FIBER TOXICITY. |13|
  [13] [S3] (CONTROL+AL2O3-18U) VS (AMOSITE+AL2O3-HC), INDUCED POOLED |57|
  [14] [S4] CONTROL VS AL2O3-18U, CHANGES DUE TO AL2O3. |16|
```

Similarly, the subsets which contain the keyword `amosite` are:

by keyword

```
119<CMD>: SET SRL SUBSET
SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
  Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, READ,
  RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?: FINDKEYWORD<CR>
Enter search key phrases composed with '~', '^', '|', '(', ')'
 ?: AMOSITE<CR>
SRL subset names:
  [11] [S1] CONTROL VS AMOSITE ASBESTOS, INDUCED FIBER TOXICITY. |13|
  [13] [S3] (CONTROL+AL2O3-18U) VS (AMOSITE+AL2O3-HC), INDUCED POOLED |57|
  [15] [S5] AMOSITE VS AL2O3-HC, CHANGES BETWEEN TOXIC MATERIALS. |16|
```

A search for subsets which contain *both* `control` and `amosite` may be specified as in the following example. Up to 10 key words or phrases may be specified.

```
120<CMD>: SET SRL SUBSET
SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
  Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
  RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?: FINDKEYWORD<CR>
Enter search key phrases composed with '~', '^', '|', '(', ')'
 ?: CONTROL^AMOSITE<CR>
SRL subset names:
  [11] [S1] CONTROL VS AMOSITE ASBESTOS, INDUCED FIBER TOXICITY. |13|
  [13] [S3] (CONTROL+AL203-18U) VS (AMOSITE+AL203-HC), INDUCED POOLED |57|
```

what sets have Rspot n?

The `QueryRspot` suboperation is used to find which SRL subsets the Rspot (to be specified) is found (out of all existing SRL subsets). This is useful if a spot is found to be interesting in one context and the user desires to check whether it is present in others.

```
121<CMD>: SET SRL SUBSET
SRL subset commands are
(Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
 Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
 RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
 Union[/ListOfSRLs], Write[:Filename])
 ?: QUERYRSPOT<CR>
Check SRL subsets for Rspot#?: 423<CR>
SRL subset names:
  [ 2] MISSING CLASS-test 1,2 AT .99, CV AREA<.5, MIN 3 GELS/CLASS |13|
  [11] [S1] CONTROL VS AMOSITE ASBESTOS, INDUCED FIBER TOXICITY. |13|
  [16] S1 UNION S2 |84|4
  [17] (S1 UNION S2) UNION S3 |129|
  [19] [S6] = UNION(S1,S2,S3) - UNION(S4,S5). |109|
```

List of SRL Subsets (LOS)

creating the list of SRL subsets

As discussed before, several of the subcommands can create on lists of SRL subsets (LOS). The following examples illustrate this.

```
201<CMD>: SET SRL SUBSET
SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
  Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
  RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?: EXPLICIT/ListOfSRLs
List of SRL subset #s?: 1,2,6
```

```
SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
  Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
  RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?: DIRECTORY/ListOfSRLs
SRL subset names:
  [ 1] NORMALIZATION SPOTS NON-SATURATING AND FOUND IN ALL GELS |14|
  [ 2] LANDMARKS SET OF SPOTS |23|
  [ 6] TB-TEST OF CLASSES (1,2) AT P-VALUE=0.90 | 7|
Subset name (or # Or # range Or <ALL> Or <LAST> <LOS> Or SRLSS[i])?: <LOS>
'LIST OF' SRL's is <LOS>
Set #1<<<NORMALIZATION SPOTS NON-SATURATING AND FOUND IN ALL GELS>>>= 8
 97 104 105 107 240 331 332 433 465 576 717 718 723
Set #2<<<LANDMARKS SET OF SPOTS>>>= 8 55 89 98 105 141 150 173 208
 233 240 292 331 465 528 529 576 597 619 630 661 693 723
Set #6<<<TB-TEST OF CLASSES (1,2) AT P-VALUE=0.90>>>= 59 228 299 307
 327 437 591

SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
  Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
  RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?: FINDKEYWORD
Enter SEARCH key expression, composed with '~', '^', '|', '(', ')'
 ?: tb-test
SRL subset names:
  [ 6] TB-TEST OF CLASSES (1,2) AT P-VALUE=0.90 | 7|
  [ 7] TB-TEST OF CLASSES (1,2) AT P-VALUE=0.95 | 1|
  [ 8] TB-TEST OF CLASSES (1,2) AT P-VALUE=0.99 | 0|

SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
  Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
  RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?: QUERYRSPOT
Check SRL subsets for Rspot#?: 89
SRL subset names:
  [ 2] LANDMARKS SET OF SPOTS |23|
  [13] ALL PSEUX RSPOTS FOR RGEL 0324.1 |709|
```

As discussed before, several of the subcommands can operate on lists of SRL subsets. The following examples illustrate this.

operations on list of SRL subsets

```
202<CMD>: SET SRL SUBSET
SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
  Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
```

```
  RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?: UNION/ListOfSRLs
Set operation on all sets in List-Of-SRLs.
Result in working SRL.
Search Results List =  8 55 59 89 97 98 104 105 107 141 150 173 208
 228 233 240 292 299 307 327 331 332 433 437 465 528 529 576 591 597
 619 630 661 693 717 718 723

SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
  Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
  RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?: INTERSECTION/ListOfSRLs
Set operation on all sets in List-Of-SRLs.
Result in working SRL.
Search Results List =

SRL subset commands are
 (Assign, CLEAR, Explicit[/ListOfSRLs], DElete[/ListOfSRLs], DIrectory, FIndkeywd,
  Intersection[/ListOfSRLs], List[/ListofSRLs], QueryRspot, REAd,
  RENUMBER, REStore, SPss[/Mosaic/Rmap], SUbtract[/ListOfSRLs],
  Union[/ListOfSRLs], Write[:Filename])
 ?: SUBTRACT/ListOfSRLs
Set operation on all sets in List-Of-SRLs.
Result in working SRL.
Search Results List =  59 228 299 307 327 437 591
```

<CMD> SET STatistics limits

changing prefilter

Set statistics limits in the prefilter for use in searching. The following sample dialogue illustrates the parameters. Answering <CR> preserves the current setting. Typing `@` permits backing up to a previous question when you realize that you gave the wrong answer the first time through. The `$` permits exiting the `SET STATISTICS` command so that no more questions need to be answered. Note that the `CREATE/ERSPOT` uses sizing limits for (area, D′, OD difference and DP) set by `SET STATISTICS` for sizing US spots to be used in the construction of eRspot sets. The prefilter limits may be quickly viewed using the `LIMITS` command. Section 5.1.11 in the tutorial discusses how one goes about deciding which prefilter parameters to change.

```
207<CMD>: SET STATISTICS LIMITS
Answer '@' to backup to previous Q&A question,
       '$' to exit Q&A immediately,
       '?' print help for setting a parameter by keyword,
       'RETURN-key' to leave limits unchanged,
       Otherwise enter numeric values to change parameter limits.
Relative distance limits [0.00,512.00]?: <CR>
Mn DL [0.00,512.00]?: <CR>
```

```
Mn DP [0.00,10.00]?: <CR>
MN area [0.00,10000000.00]?: 25,10000000<CR>
MN density (Mode:R) [0.00,10000000]?: 3,10000000<CR>
Coef. variation: S.D./Mn Rset area [0.00,10.00]?: 0,1.5 <CR>
Coef. variation: S.D./Mn Rset density [0.00,10.00]?: 0,1.5<CR>
Spot (max-mnBkgrd) OD difference [0.00,10.00]?:  0,1.70<CR>
Class difference t-(F-, or WMW-)Test, p-value or confidence level
 (1%, 5%, 10%, 20% or .99, .95, .90, .80) is 10%?: .95<CR>
Check # gels in Rspot set (per class) [0:1000](ALL or #s)?: <CR>
```

It is also possible to set a single feature's limits using the `/OPTION:` switch as follows. Note that if you type a bad keyword, it will tell you what the legal keywords are.

```
16<CMD>: SET STATISTICS
Answer '@' to backup to previous Q&A question,
       '$' to exit Q&A immediately,
       '?' print help for setting a parameter by keyword,
       'RETURN-key' to leave limits unchanged,
       Otherwise enter numeric values to change parameter limits.
Relative distance limits [0.00,511.00]?: ?

<keys>=  DISTANCE DL DP AREA DENSITY CVA CVD ODDF P-VALUE NBRGELS
with 'SET STATISTICS/Option:<key>=<value-list>'
```

It will only respond to an exact keyword match as for example:

`/OPTION:`
switch option

```
17<CMD>: SET STATISTICS/Option:arra=25,500
GLOBAL CMD SWITCHES: /Option:ARRA=25,500
Answer '@' to backup to previous Q&A question,
       '$' to exit Q&A immediately,
       '?' print help for setting a parameter by keyword,
       'RETURN-key' to leave limits unchanged,
       Otherwise enter numeric values to change parameter limits.
Illegal keyword. Pick one of

<keys>= DISTANCE DL DP AREA DENSITY CVA CVD ODDF P-VALUE NBRGELS
with 'SET STATISTICS/Option:<key>=<value-list>'

19<CMD>: SET STATISTICS/Option:area=25,500
GLOBAL CMD SWITCHES: /Option:AREA=25,500
Answer '@' to backup to previous Q&A question,
       '$' to exit Q&A immediately,
       '?' print help for setting a parameter by keyword,
       'RETURN-key' to leave limits unchanged,
       Otherwise enter numeric values to change parameter limits.
MN area limits [0.00,500.00]?:

20<CMD>: !!
19: SET STATISTICS/Option:area=25,500
GLOBAL CMD SWITCHES: /Option:AREA=25,500
```

```
Answer '@' to backup to previous Q&A question,
       '$' to exit Q&A immediately,
       '?' print help for setting a parameter by keyword,
       'RETURN-key' to leave limits unchanged,
       Otherwise enter numeric values to change parameter limits.
MN area limits [25.00,500.00]?:

21<CMD>: SET STATISTICS/Option:density=3,1000
GLOBAL CMD SWITCHES: /Option:DENSITY=3,1000
Answer '@' to backup to previous Q&A question,
       '$' to exit Q&A immediately,
       '?' print help for setting a parameter by keyword,
       'RETURN-key' to leave limits unchanged,
       Otherwise enter numeric values to change parameter limits.
MN density (Mode:R) limits [0.00,1000.00]?:
```

checking PREFILTER

The `LIMITS` command just checks the changes just made by `SET STATISTICS`:

```
22<CMD>: LIMITS
Rspot Mapping file [].
Map Rspots to Fspots is OFF.
Ratio-List density mode calibration DOES NOT EXIST.
Least-Squares density mode calibration DOES NOT EXIST.
Image size [rowsXcolumns]=[512X512]
Relative distance limits [0.00,511.00] pixels
Mn DL limits [0.00,511.00] pixels
Mn DP limits [0.00,511.00] pixels
MN area limits [0.00,10000000.00] pixels**2
MN density (Mode: P, tot-density/spot) limits [0.000,10000000.000]
Coef. variation: S.D./Mn Rset area limits [0.00,10.00]
Coef. variation: S.D./Mn Rset dens limits [0.00,10.00]
Spot OD diff(maxOD-mnBkgrdOD) limits [0.000,10.000] OD
Class difference t-Test, F-test, Rank order p-value limit is .90:
Check if # gels in Rspot set [0:10000]
(x,y) calibration file: <NONE>
pIe Rgel limits [0.00,511.00] pixels
MW Rgel limits [0.00,511.00] pixels
Pairing-Search-Label [PASEUC]
No SRL Subset search restriction is active.
Restrict US spots to be added to eRspots (during CREATE cmd)
  if LEQ distance:20.0
```

<CMD> SET Working gels

defining visible gels

Define working set of gels as defined by a list of their accession numbers from PCG database which are visible for the prefilter. This may be done either explicitly by specifying a list of accession numbers or by specifying a gel subset name.

GUI only

If the GUI is active, when you select YES, it pops up a compacted list of accession numbers which are in the entire database. Those which are in the current working set of gels are maked in GREEN, while BLACK means they are not in the

working set. By clicking on any accession number you can toggle it in or out of the current working set of gels. In addition, any time a n accession number is clicked on it displays the associated current study information for that gel in the message windows. Press Done to actually make the change to the working set of gels or Cancel to cancel the operation.

The existing working set of gels can be edited gel by gel,

```
207<CMD>: SET WORKING GELS
The current gel working set, of size 12, is:
 0250.2 0251.2 0252.2 0253.2 0254.2 0255.2 0256.2 0257.2
 0258.2 0259.2 0260.2 0261.2 0262.2 0263.2 0264.2 0265.2
Enter new working set?(Yes/No/Define/acc#s/Add/Subtract)[No]
 ?: YES<CR>
Include gel 0250.2?(Y/N)[N]: <CR>
Include gel 0250.1?(Y/N)[N]: y<CR>
Include gel 0251.1?(Y/N)[N]: y<CR>
Include gel 0251.2?(Y/N)[N]: n<CR>
Include gel 0252.1?(Y/N)[N]: <CR>
Include gel 0252.2?(Y/N)[N]: <CR>
Include gel 0253.1?(Y/N)[N]: <CR>
Include gel 0253.2?(Y/N)[N]: y<CR>
Include gel 0254.1?(Y/N)[N]: <CR>
Include gel 0254.2?(Y/N)[N]: <CR>
Include gel 0255.1?(Y/N)[N]: <CR>
        .
        .
        .
Include gel 0265.2?(Y/N)[N]: <CR>
```

Alternatively, an explicit list of gel accession numbers may be specified, *explicit declaration*

```
208<CMD>: SET WORKING GELS
The current gel working set, of size 12, is:
0250.2 0251.2 0252.2 0253.2 0254.2 0255.2 0256.2 0257.2
0258.2 0259.2 0260.2 0261.2 0262.2 0263.2 0264.2 0265.2
Enter new working set?(Yes/No/Define/acc#s/Add/Subtract)[No]
 ?: DEFINE<CR>
Input a list of accession numbers or gel subset name [ALL]
 ?: ?
These are the current GEL SUBSET NAMES
--------------------------------------
  class# 1 = 24-HR
  class# 2 = 48-HR
  class# 3 = DARK GELS
Input a list of accession numbers or gel subset name [ALL]
 ?: 258.2,259.2,273.2,264.2,265.2<CR>
0273.2 not in the database!
```

Since gel 273.2 was not in the database, the question is asked again.

```
Input a list of accession numbers or gel subset name [ALL]
 ?: 258.2,259.2,264.2,265.2<CR>
```

by gel subset The working set may be defined in terms of a gel subset as follows:

```
209<CMD>: SET WORKING GELS
The current gel working set, of size 12, is:
 0250.1 0251.1 0253.2
Enter new working set?(Yes/No/Define/acc#s/Add/Subtract)[No]
 ?: DEFINE<CR>
Input a list of accession numbers or gel subset name [ALL]
 ?: DARK GELS<CR>
Defining the working gels as DARK GELS
```

modify existing W.S. Individual gels may be added or subtracted from the working set of gels.

```
210<CMD>: SET WORKING GELS
The current gel working set, of size 12, is:
 0524.1 0569.1 0578.2 0584.1 0596.1 0497.1 0505.1 0511.1 0514.1 0515.1
 0517.1 0593.2
Enter new working set?(Yes/No/Define/acc#s/Add/Subtract)[No]
 ?: SUBTRACT<CR>
Input a list of accession numbers
 ?: 569.1, 596.1<CR>
        .
        .
        .

211<CMD>: SET WORKING GELS
The current gel working set, of size 10, is:
 0524.1 0578.2 0584.1 0497.1 0505.1 0511.1 0514.1 0515.1 0517.1 0593.2
Enter new working set?(Yes/No/Define/acc#s/Add/Subtract)[No]
 ?: ADD<CR>
Input a list of accession numbers
 ?: 569.1<CR>
        .
        .
        .

212<CMD>: SET WORKING GELS
The current gel working set, of size 11, is:
 0524.1 0569.1 0578.2 0584.1 0497.1 0505.1 0511.1 0514.1 0515.1 0517.1
 0593.2
Enter new working set?(Yes/No/Define/acc#s/Add/Subtract)[No]
 ?: <CR>
```

<CMD> SPss

Dump an SPSS (*.sps* extension) summary file of part of the PCG database for selected Rspot sets. SPSS is Statistical Package for the Social Sciences [NieH75]. The SPSS data file is an ASCII data file consisting of numeric data in a fixed format

suitable for input to the SPSS Statistical Package. It is also used as input to the **markgel** and **mosaic** programs to construct Rmap and mosaic images respectively. Note that when making Rmaps, the `SET LABEL` should probably be set to *not* include EP spots whereas for making mosaics it probably *should* include EP. Generally, one does not include AP spots. If the `/SAS` switch is specified, then generate a *.sas* file readable by the SAS program ([SAS85]) instead of SPSS. A typical prompt might be:

SPSS or SAS data file

```
110<CMD>: SPSS
Creating SPSS data file from PCGL DB:
   /home/joeUser/gellab/pcg/118pcg.pcg
Output file:(000001.sps)?: 118pcg.sps<CR>
Input list of Rspot sets to output in SPSS file.
Use '$' to indicate use spot subset found with INQUIRE.
?: $<CR>
```

The `$` indicates the SRL Rspot list is to be used. Alternatively the user may specify a list of Rspot sets,

```
111<CMD> SPSS/TITLE:'t-test search class 1 Vs 2 at 99%'
Creating SPSS data file from PCGL DB:
   /home/joeUser/gellab/pcg/as3pcg.pcg
Input list of Rspot sets to output in SPSS file.
Use '$' to indicate use SRL Rspot sets.
 ?: 9, 38, 101, 304<CR>
Output file: (000002.sps)?: as3t99.sps<CR>
```

A typical SPSS data file has the following format. The 2nd line is the last "search banner" created during the use of `INQUIRE` or `SET SRL SUBSETS` when the SRL was changed. Optionally, the `/TITLE:` switch may be used to specify different title text when issuing the original SPSS command as illustrated above. Fields of the SPSS record are defined as follows:

SPSS file format

RSPOT# is the Rspot set number.

ACC# is the gel accession number.

INDEX is the connected component number for the gel as defined in the GSF file.

density-*mode* is the current density mode: DENSR, DENSL, DENS%, D′

D′ is unnormalized density.

LABEL is a number in the range [0:5] where: 0=US, 1=SP, 2=PP, 3=AP, 4=EP, 5=CP.

LMSET is a number in the range [1:25] which correspond to landmarks [A:Y].

DP is dP for that spot (distance from spot to corresponding Rgel spot).

DL is dL for that spot (distance from spot/Rgel-spot centroid to landmark spot).

DX is dx from spot to landmark spot.

DY is dx from spot to landmark spot.

XABS is absolute x coordinate of spot.

YABS is absolute y coordinate of spot.

CLASS is experimental class number to which this gel belongs.

```
File: as3t99.sps 12/20/1988, 07:30:10AM from:
PCG DB: /home/joeUser/gellab/pcg/as3pcg.pcg
Title:    T-test SEARCH CLASS 1 VS 2 AT 99%
Rspot Mapping file [].
Map Rspots to Fspots is OFF.
Ratio-List density mode calibration EXISTS.
Least-Squares density mode calibration EXISTS.
Image size [rowsXcolumns]=[512X512]
Relative distance limits [0.00,512.00] pixels
Mn DL limits [0.00,512.00] pixels
Mn DP limits [0.00,10.00] pixels
MN area limits [0.00,10000000.00] pixels**2
MN density (Mode: R) limits [0.00,10000000.00]
Coef. variation: S.D./Mn Rset area limits [0.00,10.00]
Coef. variation: S.D./Mn Rset dens limits [0.00,10.00]
Spot OD difference limits [0.00,10.00]
Class difference t-Test, F-test, Rank order p-value limit is .90:
Check if # gels in Rspot set [0:1000]
(x,y) calibration file: <NONE>
pIe Rgel limits [0.00,511.00] pixels
MW Rgel limits [0.00,511.00] pixels
Pairing-Search-Label [PSEAU]
List of Rspots used in Ratio list normalization:
 11 70 97 181 612

Class # 1(-TC#1)=0524.1 0497.1 0505.1
Class # 2(-TC#2)=0578.2 0515.1 0517.1
Class # 3(-TC#3)=0569.1 0511.1 0514.1
Class # 4(-TC#4)=0584.1 0593.2
Class # 5(-TC#5)=0596.1
Class # 6(<NULL>)=
Class # 7(<NULL>)=
Class # 8(<NULL>)=
Class # 9(<NULL>)=
RSPOT# ACC# INDEX DENSR D' LABEL[0:5] LMSET[1:25] DP DL DX DY XABS YABS CLASS MNBK
    9 0596.1   190    5.68   16.90     3     1   7.0     7     1     7   311   161 5 0.54
    9 0569.1   226    6.14   15.70     3     1   8.1     8    -8     1   275   178 3 0.48
    9 0584.1   199   14.29   33.20     1     1   0.0     0     0     0   290   184 4 0.37
        .
        .
        .
    9 0578.2    92   52.06   55.70     1     1   0.0     0     0     0   258   129 2 0.55
```

The SPSS file can be sorted by *gel* rather than *Rspot*. This is invoked by specifying the global <CMD> level switch `/SOrtByGel` with either the `SPSS` or `SET SRL SUBSET` (with `SPSS` subcommand) commands. For example, <CMD>: `SPSS/SortByGel`. `/SORTBYGEL` *option*

```
112<CMD>: SPSS/SORTBYGEL
Output file:[000001.sps]?: ab1s07.sps<CR>
```

The file is of the form:

```
File: ab1s07.sps 01/07/1983, 08:49:59 AM from: ab1pcg.pcg
SRLSS[7]=F-test 1,3 AT .99, CV AREA<.5, MIN 3 GELS/CLASS
        .
        .
```

```
        .
SRL Rspots |9| :279 283 297 304 331 337 395 418 466
GEL#, {RSPOT#S}
0269.1                                                                  1
   6.80    6.90    3.80    6.70    1.30   13.60    7.10    9.70   19.50      2
0266.1                                                                  3
   3.30    3.90    1.50    7.20    3.20   12.90     .70    1.20   10.10      4
0267.1                                                                  5
   4.20    5.50    1.00    5.50    1.30    5.00    4.50   14.80    8.60      6
0268.1                                                                  7
   2.20    3.10    1.80    5.40     .80    4.60    3.50   17.10    1.50      8
0270.1                                                                  9
   8.40    9.30    8.40    4.80     .70   11.40    5.80    1.10   31.00     10
0270.2                                                                 11
  25.60   32.60   34.50   18.20    3.00   44.70   32.30    8.10   68.60     12
0271.2                                                                 13
   2.30    4.70     .00    3.50    5.10    4.60    5.80    4.60    3.10     14
        .
        .
        .
```

SPSS /FULL file format

Alternatively, the SPSS file can be sorted by *gel* in a different format. This is invoked by specifying the global <CMD> level switch `/FULL` with either the `SPSS` or `SET SRL SUBSET` (with `SPSS` subcommand) commands. For example, <CMD>: `SPSS/FULL`. The file is of the form:

```
File: ab1s07.sps 12/20/1988, 07:46:24AM from:
PCG DB: /home/joeUser/gellab/pcg/ab1tst.pcg
Title: User defined spot list

Rspot Mapping file [].
Map Rspots to Fspots is OFF.
Ratio-List density mode calibration EXISTS.
Least-Squares density mode calibration EXISTS.
Image size [rowsXcolumns]=[512X512]
Relative distance limits [0.00,512.00] pixels
Mn DL limits [0.00,512.00] pixels
Mn DP limits [0.00,10.00] pixels
MN area limits [0.00,10000000.00] pixels**2
MN density (Mode: R) limits [0.00,10000000.00]
Coef. variation: S.D./Mn Rset area limits [0.00,10.00]
Coef. variation: S.D./Mn Rset dens limits [0.00,10.00]
Spot OD difference limits [0.00,10.00]
Class difference t-Test, F-test, Rank order p-value limit is .90:
Check if # gels in Rspot set [0:1000]
(x,y) calibration file: <NONE>
pIe Rgel limits [0.00,511.00] pixels
MW Rgel limits [0.00,511.00] pixels
Pairing-Search-Label [PSEAU]
List of Rspots used in Ratio list normalization:
 11 70 97 181 612
```

```
Class # 1(-TC#1)=0524.1 0497.1 0505.1
Class # 2(-TC#2)=0578.2
Class # 3(-TC#3)=0569.1 0511.1 0514.1
Class # 4(-TC#4)=0584.1
Class # 5(-TC#5)=0596.1
Class # 6(<NULL>)=
Class # 7(<NULL>)=
Class # 8(<NULL>)=
Class # 9(<NULL>)=
Rspot#\GELS:  0524.1  0569.1  0578.2  0584.1  0596.1  0497.1  0505.1  0511.1  0514.1
   2:          0.70    0.39    7.29    0.00    1.18    0.69    0.00    1.08   13.25
   9:         23.25   48.16   52.06   14.29   19.33   36.10   22.80   44.12   46.99
  38:          0.22    0.00    0.00    0.00   13.61    0.38    0.00    1.55    0.00
 101:         19.59   17.63    1.68   21.51   37.55   11.21    2.40  186.69   51.81
 304:          0.00   17.63    1.68    0.00   37.55   11.21    2.40  186.69   51.81
```

<CMD> SYSTEM

Evaluate a UNIX shell level command from **cgelp2**. If the UNIX command is *not* put in the background, then **cgelp2** waits until the UNIX command terminates before continuing with the **cgelp2** prompt. This command is useful for making last minute file system changes, invoking the text editor or checking the status of various files. Note that `csh` shell wildcard file name and alias expansion does not occur so that you must type the full path when specifying files outside of the current directory.

UNIX commands

```
201<CMD>: system
System command?: date<CR>
Wed Jan  21 11:33:37 EST 1989

202<CMD>: system
System command?: ls -l /home/joeUser/gellab/pcg/jj7pcg.pcg<CR>
-rw-------  1 joeUser  3240960 Jan 20 17:41 /home/joeUser/gellab/pcg/jj7cgl.pcg

203<CMD>: system
System command?: rm /home/joeUser/gellab/pcg/jj7pcg.pcg<CR>
```

<CMD> TAbulate

Generate and print one of the following three types of tables: (1) Rspot density-rank order, (2) correlation of each gel compared with each of the other gels, (3) cross-correlation of each Rspot set with every other Rspot in the SRL. Up to 18 gels or 18 Rspots can be correlated with each other at a time as only 18 objects will fit across a 132 column lineprinter output. If more objects need to be correlated, break them up into smaller sets of objects and do a separate correlation.

correlation tables

The rank order and ratio tables are density plots of several selected Rspot sets. The correlation coefficients are computed as in [SneG80]. This allows visualization

of several Rspots sets in a plot as is illustrated in Example 3. Example 1 shows the intra-class gel-gel correlation table produced by computing correlation coefficients for each of the gels in duplicate scans of the same gel while Example 2 illustrates intergel correlation for replicate gels produced from parallel tissue cultures. In the correlation tables the feature *rFactor* [AndN81] is defined as the mean $|d1-d2|/(d1+d2)$ for all spot pairs being considered. In addition to being printed on the terminal, the output is also put into a file (*.tbl* extension) for which a file name is requested.

`.tbl` *file extension*

Example 1: Intra-class gel-gel correlation of replicate scans.

Compute the intral gel-gel correlation of gels in the same experimental class. In this case, these were replicate scans of data from a vidicon scanner and we were observing scanner noise.

```
222<CMD>: TABULATE
Table type?(Rank-order, Correlate-gel, SRL-subset-correlate)?: CORRELATION<CR>
Output file:(000004.tbl)?: cov1.tbl<CR>
File: cov1.tbl 10/01/1981,  02:27:50 PM
#  1[0504.4] study: / 4DAY/ KD MULTIPLE EXPOSURE STUDY
#  2[0504.5] study: / 4DAY/ KD MULTIPLE EXPOSURE STUDY
#  3[0504.7] study: / 4DAY/ KD MULTIPLE EXPOSURE STUDY
#  4[0504.8] study: / 4DAY/ KD MULTIPLE EXPOSURE STUDY
#  5[0504.9] study: / 4DAY/ KD MULTIPLE EXPOSURE STUDY

Mean Variation for gels in database. Labels: PS

Correlation Coefficient of paired spot density
      |0504.4|0504.5|0504.7|0504.8|0504.9
-----------------------------------------
0504.4|      | .9202| .9142| .9179| .9149
0504.5|      |      | .9715| .9640| .9687
0504.7|      |      |      | .9685| .9653
0504.8|      |      |      |      | .9617

Number of spot pairs for two gels considered
      |0504.4|0504.5|0504.7|0504.8|0504.9
-----------------------------------------
0504.4|      |   955|   913|   903|   901
0504.5|      |      |   913|   903|   901
0504.7|      |      |      |   912|   906
0504.8|      |      |      |      |   905
```

Example 2: Inter-class gel-gel mean variation and correlation of different scans.

Compute the inter gel correlation between different experimental conditions.

```
223<CMD>: TABULATE
Table type?(Rank-order, Correlate-gel, SRL-subset-correlate)?: CORRELATION<CR>
Output file:(000003.tbl)?: as8t0a.tbl<CR>
File: as8t0a.tbl 03/19/1982,  03:40:52 PM
#  1[0250.9] study: / C14/ ALUMINUM,T0,CONTROL,BOTTLE#1
```

```
#  2[0251.2] study: / C14/ ALUMINUM,T0,CONTROL,BOTTLE#2
#  3[0252.2] study: / C14/ ALUMINUM,T0,AL2O3-HC,BOTTLE#3
#  4[0253.2] study: / C14/ ALUMINUM,T0,AL2O3-HC,BOTTLE#4
#  5[0254.2] study: / C14/ ALUMINUM,T0,AL2O3-18U,BOTTLE#5
#  6[0255.2] study: / C14/ ALUMINUM,T0,AL2O3-18U,BOTTLE#6
#  7[0256.2] study: / C14/ ALUMINUM,T0,AMOSITE,BOTTLE#7
#  8[0257.2] study: / C14/ ALUMINUM,T0,AMOSITE,BOTTLE#8
Correlation between gels in database. Pairing labels: PS
Paged CGL database file: as8pcg.pcg
List of Rspots used in Ratio Normalization Spot list:
    3    6    7    8   16   19   23   24   25   27   28   30   31
   32   33   36   38   39   42   46   49   55   56   57   58   59
   67   68   69   71   79   82   85   91   94   96  101  103  104
  108  110  114  115  118  119  120  121  125  126  129  130  139
  140  144  147  149  151  152  157  159  171  172  187  209  217
  235  258  260  265  268  275  285  292  293  296  303  305
Cor.Coef. of paired spot density
      |0250.9|0251.2|0252.2|0253.2|0254.2|0255.2|0256.2|0257.2
------------------------------------------------------------
0250.9|      | .9270| .9836| .6700| .9789| .9576| .9649| .9707
0251.2|      |      | .8915| .6277| .8882| .8849| .8651| .8972
0252.2|      |      |      | .6935| .9686| .9527| .9581| .9562
0253.2|      |      |      |      | .6444| .5183| .5830| .5357
0254.2|      |      |      |      |      | .9647| .9573| .9638
0255.2|      |      |      |      |      |      | .9476| .9535
0256.2|      |      |      |      |      |      |      | .9577
0257.2|      |      |      |      |      |      |      |

Number of spot pairs for two gels considered
      |0250.9|0251.2|0252.2|0253.2|0254.2|0255.2|0256.2|0257.2
------------------------------------------------------------
0250.9|      |   287|   249|   165|   246|   253|   263|   274
0251.2|      |      |   243|   158|   238|   246|   255|   265
0252.2|      |      |      |   158|   217|   226|   236|   238
0253.2|      |      |      |      |   157|   160|   162|   161
0254.2|      |      |      |      |      |   224|   232|   240
0255.2|      |      |      |      |      |      |   237|   245
0256.2|      |      |      |      |      |      |      |   256
0257.2|      |      |      |      |      |      |      |

rFactor (Taylor's) of paired spot density
      |0250.9|0251.2|0252.2|0253.2|0254.2|0255.2|0256.2|0257.2
------------------------------------------------------------
0250.9|      | .1666| .1950| .2065| .1822| .1821| .1908| .1674
0251.2|      |      | .2675| .2876| .2433| .2404| .2621| .2241
0252.2|      |      |      | .2443| .2342| .2434| .2218| .2339
0253.2|      |      |      |      | .2395| .2783| .2353| .2614
0254.2|      |      |      |      |      | .2091| .2286| .2229
0255.2|      |      |      |      |      |      | .2487| .2385
0256.2|      |      |      |      |      |      |      | .2574
0257.2|      |      |      |      |      |      |      |
```

Example 3: Correlation coefficients matrix of SRL Rspot sets.

It is possible to correlate Rspots sets with one another. Up to 18 Rspot sets (specified by the current SRL list) may be correlated. To speed up the process the /SRL switch may be used to specify the same set that was restored (using SET SRL SUBSETS//RESTORE) into the current SRL.

```
224<CMD>: TABULATE/SRL:7
Table type?(Rank-order, Correlate-gel, SRL-subset-correlate)?: SRL<CR>
Output file:(000001.tbl)?: <CR>
File: 000001.tbl 01/07/1983, 07:12:11 AM
  [ 7] F-test 1,3 AT .99, CV AREA<.5, MIN 3 GELS/CLASS | 9|
#  1[0269.1] study: / 1 WEEK/ CONTROL,NON-TOX BOTTLE#4
#  2[0266.1] study: / 1 WEEK/ CONTROL,NON-TOXBOTTLE#1
#  3[0267.1] study: / 1 WEEK/ CONTROL,NON-TOX BOTTLE#2
      .
      .
      .
# 22[0283.1] study: / 1 WEEK/ AL2O3-HC,TOXIC, BOTTLE#16, DUPL-B

Correlation between gels in database. Pairing labels: PSUE
Paged CGL database file: ab1pcg.pcg
Using least square normalization.

Cor.Coef. of paired Rspot-set density
      |  279 |  283 |  297 |  304 |  331 |  337 |  395 |  418 |  466
------------------------------------------------------------------------
[ 279]|      | .9961| .9987| .9729| .8402| .9752| .9960| .7431| .9894
[ 283]|      |      | .9966| .9648| .6199| .9582| .9971| .7737| .9825
[ 297]|      |      |      | .9661| .6316| .7612| .9979| .7550| .9298
[ 304]|      |      |      |      | .6924| .7918| .9702| .6929| .9266
[ 331]|      |      |      |      |      | .6805| .6217| .4477| .6648
[ 337]|      |      |      |      |      |      | .9670| .6578| .9247
[ 395]|      |      |      |      |      |      |      | .7694| .9798
[ 418]|      |      |      |      |      |      |      |      | .6993

Number of spot pairs for two Rspot-set considered
      |  279 |  283 |  297 |  304 |  331 |  337 |  395 |  418 |  466
------------------------------------------------------------------------
[ 279]|      |   11 |   10 |   11 |   10 |   11 |   10 |   10 |   11
[ 283]|      |      |   14 |   17 |   13 |   17 |   14 |   13 |   17
[ 297]|      |      |      |   15 |   11 |   15 |   12 |   11 |   15
[ 304]|      |      |      |      |   14 |   20 |   14 |   14 |   20
[ 331]|      |      |      |      |      |   14 |   11 |   12 |   14
[ 337]|      |      |      |      |      |      |   14 |   14 |   20
[ 395]|      |      |      |      |      |      |      |   11 |   14
[ 418]|      |      |      |      |      |      |      |      |   14

rFactor (Taylor's) of paired Rspot-set density
      |  279 |  283 |  297 |  304 |  331 |  337 |  395 |  418 |  466
------------------------------------------------------------------------
[ 279]|      | .1450| .1580| .2619| .6751| .4435| .2023| .7040| .5111
[ 283]|      |      | .2346| .2421| .5635| .4569| .1546| .5996| .4618
```

```
    [ 297]|      |      |      | .3887| .6151| .5978| .2226| .7376| .6247
    [ 304]|      |      |      |      | .5488| .4214| .2461| .5233| .4496
    [ 331]|      |      |      |      |      | .6894| .6500| .4795| .7804
    [ 337]|      |      |      |      |      |      | .4565| .6177| .2906
    [ 395]|      |      |      |      |      |      |      | .6405| .5194
    [ 418]|      |      |      |      |      |      |      |      | .6222
```

Example 4: Rank Order Table of selected spots.

Plot selected Rspot sets adjacent to one another in a Rank Order Table so one can track how different gels change as a function of Rspot set. It is useful to connect different colored lines for the same gels for different spots. This lets you view changes in the expression profile of all of the gels at the same time.

```
Table type?(Rank-order, Correlate-gel, SRL-subset-correlate)?: RANK<CR>
        Output file:(000004.tbl)?: rnktb1.tbl<CR>
        Input list of Rspots to tabulate in rank order.
        Use '*' to indicate use spot subset found with INQUIRE.
         ?: 44,88,117,393,512<CR>
        File: rnktb1.tbl 02/26/1981,  10:17:23 AM
        RANK-ORDER table: <ACC#>&<LMset>&<Class #>
        Paged CGL database file: j5.pcg
        Using least square normalization.
        User defined spot list

Density
  22.3 |                    0263.2G2
  21.8 |
  21.3 |
  20.8 |
  20.3 |                    0260.2G2
  19.7 |                    0262.2G2
  19.2 |
  18.7 |                    0261.2G2
  18.2 |
  17.7 |
  17.2 |                                        0263.2V2
  16.7 |  0260.2B2
  16.2 |  0262.2B2          0264.2G2
  15.6 |  0263.2B2          0259.2G2            0262.2V2
  15.1 |
  14.6 |                    0250.2G1
       |                    0265.2G2
  14.1 |                    0258.2G2
  13.6 |  0261.2B2                              0259.2V2
  13.1 |
  12.6 |                                        0250.2V1
  12.1 |
  11.5 |                              0262.2P2
  11.0 |                                        0265.2V2
       |                                        0260.2V2
  10.5 |  0264.2B2
```

```
 10.0 |  0265.2B2 0262.2E2
  9.5 |  0259.2B2 0263.2E2
      |  0258.2B2
  9.0 |                                        0261.2V2
  8.5 |                               0260.2P2
  8.0 |
  7.4 |
  6.9 |  0250.2B1                     0263.2P2 0264.2V2
  6.4 |           0260.2E2            0259.2P2
  5.9 |           0250.2E1            0264.2P2
      |           0259.2E2
  5.4 |           0264.2E2                     0258.2V2
  4.9 |                               0258.2P2
  4.4 |           0261.2E2            0265.2P2
  3.9 |           0258.2E2
  3.3 |
  2.8 |
  2.3 |
  1.8 |           0265.2E2
------------------------------------------------------
Rspot |        44       88      117      393      512
Class #1=T0, Class #2=T24
```

<CMD> TImer

Enable or disable printing the run and cpu times for **cgelp2** commands. It also prints the time-of-day stamp. This command will toggle, that is, it will turn this printout on and off on successive calls. It is normally on. The following illustrates the prompt with it ON and then with OFF. The times printed out are for the previous command. Note *run* time is clock time while *cpu* time is the actual time that your program was running on the Central Processing Unit of the time shared system. When the timer is on, **cgelp2** prompts for the new command by printing the time and then the <CMD> prompt:

what time is it?

```
[03:51:20PM] Real TIME =00:12:51  CPU TIME =00:07:38,  59.40%
33<CMD>:
```

When the timer is off, it prints the time of day command prompt:

```
[08:12:26 PM] 34<CMD>:
```

In performing a search that may take a long time and one would like to know where the system is in processing the PCG DB. Add the /NOWORRYMSG <CMD> modifier to print out the number of the Rspot set currently being processed every 30 seconds. For example,

```
4<CMD>: INQ/NOWORRYMSG
Inquire cmds (B,CH,CO,D,E,F,H,I,LA,LE,K,M,OE,OR,P,R,S,T,TB,TC,%).
  HELP to list CMDS?: TB-TEST<CR>
```

```
  Std t-/Behrens-Fisher t-Test (using F-stat) class search at 0.99 significance
  Which two classes are to be compared?: ?2,3<CR>
    Don't worry, I am still working on it - Rspot set [236]
    Don't worry, I am still working on it - Rspot set [532]
. . .
```

<CMD> VAlid landmarks

List the valid landmarks in a table. Landmark validity was previously determined by the **cgelp2** program. Entry `T` indicates that the landmark was the valid **cmpgl2** label `OK` and was found to be a segmented spot. Table entry `F` indicates that the landmark was not (i.e. either **cmpgl2** `NG` or `SM` label and that the user interactively selected coordinates were used rather than a segmented spot's centroid). At the bottom of the table the number of landmarks as a function of the number of gels is given. A typical output is as follows. Note that the validity table and metric can be used to indicate that something is wrong with the landmarking for gels 271.2 and 282.1 and/or they are geometrically quite different from the Rgel. Missing landmarks are indicated by a gap in the table.

checking landmarks

```
55<CMD>: VALIDLANDMARKS
 Valid landmarks [T is OK (spot exists for landmark), F is NG
 (spot does not exist for landmark) or SM (same spot for several landmarks)
 in set of landmarks validity check]

 GEL   | A B C D E F G H I J K L M N O P Q R S T U V W X Y
-------------------------------------------------------
0269.1| T T T T T F T T T T T T T T F F F F T T
0266.1| T T T T T T T T T T T T T T T F T F T T
0267.1| T T T T T T T T T T T T T T T T T T T T
0268.1| T T T T T T T T T T T T T T T T T T T F
0270.1| T T T T T T T T T T T T T T T F T T T T
0270.2| T T T T T T T T T T T T T T T T T T T T
0271.2| T T T T T T T T T T T T T T T T T T T
0272.2| T T T T T T T T T T T T T T T T T T T T
0273.1| T T T T T T T T T T T T T T T T T T T T
0273.2| T T T T T F T T T T F T T T T F F F F T T
0274.1| T T T T T T T T T T T T T T T T T T T T
0275.1| T T T T T T T T T T T T T T T T T T T T
0276.1| T T T T T T T T T T T T T T T T T T T T
0277.1| T T T T T T T T T T T T T T T T T T T T
0278.1| T T T T T T T T T F T T T T T T T F T T
0279.1| T T T T T T T F T F T T T T T T F T F T F
0280.1| T T T T T F T F T F F T T T T T F F F T F
0281.1| T T T T T T T T T T T T T T T T T T T T
0282.1| T T T T T F T T T T T T T T F F T F T T
0283.1| T T T T T F T T T T T T T T T F F F F T T

Percentage of gels in which landmark is present (T=100%)
------------------------------------------------------------------------
  A  B  C  D  E  F  G  H  I  J  K  L  M  N  O  P  Q  R  S  T  U  V  W  X  Y
```

```
------------------------------------------------------------------------
  T  T  T  T  T 75  T 90  T 85 90  T  T  T 95 80 65 75 65  T 75 %

Global estimate of LM centroid of gel and RMS deviation from Rgel
------------------------------------------------------------------
[0269.1] Mean LM centroid (276,167), LM rmsDev w.r.t.-Rgel= .0
[0266.1] Mean LM centroid (258,186), LM rmsDev w.r.t.-Rgel= 4.8
[0267.1] Mean LM centroid (280,195), LM rmsDev w.r.t.-Rgel= 7.3
[0268.1] Mean LM centroid (263,185), LM rmsDev w.r.t.-Rgel= 8.2
[0270.1] Mean LM centroid (269,185), LM rmsDev w.r.t.-Rgel= 5.9
[0270.2] Mean LM centroid (257,179), LM rmsDev w.r.t.- Rgel= 4.8
[0271.2] Mean LM centroid (262,186), LM rmsDev w.r.t.- Rgel= 112.7
[0272.2] Mean LM centroid (286,185), LM rmsDev w.r.t.- Rgel= 7.0
[0273.1] Mean LM centroid (263,168), LM rmsDev w.r.t.- Rgel= 7.0
[0273.2] Mean LM centroid (301,177), LM rmsDev w.r.t.- Rgel= 4.5
[0274.1] Mean LM centroid (264,186), LM rmsDev w.r.t.- Rgel= 6.7
[0275.1] Mean LM centroid (267,161), LM rmsDev w.r.t.- Rgel= 5.5
[0276.1] Mean LM centroid (260,188), LM rmsDev w.r.t.- Rgel= 6.8
[0277.1] Mean LM centroid (260,180), LM rmsDev w.r.t.- Rgel= 6.2
[0278.1] Mean LM centroid (274,249), LM rmsDev w.r.t.- Rgel= 12.6
[0279.1] Mean LM centroid (266,264), LM rmsDev w.r.t.- Rgel= 15.6
[0280.1] Mean LM centroid (266,255), LM rmsDev w.r.t.- Rgel= 16.3
[0281.1] Mean LM centroid (257,253), LM rmsDev w.r.t.- Rgel= 9.6
[0282.1] Mean LM centroid (262,243), LM rmsDev w.r.t.- Rgel= 110.1
[0283.1] Mean LM centroid (268,264), LM rmsDev w.r.t.- Rgel= 14.5
```

viewing landmark coordintes

Another option available is `/FULL` to list the individual landmark coordinates as used by the **cmpgl2** program in producing *.gcf* files. It is specified as: `VALIDLANDMARKS/FULL`. It extends the information in the second (root mean square deviation) table as follows: [gel accession number, landmark name, <LM validity>], (absolute x,y coordinates in gel), (deviation of this LM from projection in Rgel), (Euclidean distance of projected LM deviation).

This detailed printout of individual landmark deviation is useful for checking particular landmarks which might be suspect. An example from part of a database is used to illustrate this.

```
56<CMD>: VALIDLANDMARKS/FULL
       .
       .
       .
Global estimate of LM centroid of gel and RMS deviation from Rgel
------------------------------------------------------------------
   LM[0269.1,A,<T>] (x,y)=(304,116) (Dx,Dy)=(  0,  0) disToCent= .0
   LM[0269.1,B,<T>] (x,y)=(345,107) (Dx,Dy)=(  0,  0) disToCent= .0
   LM[0269.1,C,<T>] (x,y)=(347, 67) (Dx,Dy)=(  0,  0) disToCent= .0
         .
         .
         .
   LM[0269.1,T,<T>] (x,y)=(343,299) (Dx,Dy)=(  0,  0) disToCent= .0
   LM[0269.1,U,<T>] (x,y)=(428,284) (Dx,Dy)=(  0,  0) disToCent= .0
```

```
[0269.1] Mean LM centroid (276,167), LM rmsDev w.r.t.- Rgel= .0

   LM[0266.1,A,<T>] (x,y)=(287,136) (Dx,Dy)=( -1, -1) disToCent= 1.4
   LM[0266.1,B,<T>] (x,y)=(328,127) (Dx,Dy)=( -1, -1) disToCent= 1.4
   LM[0266.1,C,<T>] (x,y)=(330, 89) (Dx,Dy)=( -1, -3) disToCent= 3.2
        .
        .
        .
   LM[0266.1,T,<T>] (x,y)=(324,313) (Dx,Dy)=(  1,  5) disToCent= 5.1
   LM[0266.1,U,<T>] (x,y)=(405,302) (Dx,Dy)=(  5,  1) disToCent= 5.1
[0266.1] Mean LM centroid (258,186), LM rmsDev w.r.t.- Rgel= 4.8

   LM[0267.1,A,<T>] (x,y)=(311,147) (Dx,Dy)=( -4, -3) disToCent= 5.0
   LM[0267.1,B,<T>] (x,y)=(352,140) (Dx,Dy)=( -4, -5) disToCent= 6.4
   LM[0267.1,C,<T>] (x,y)=(354,102) (Dx,Dy)=( -4, -7) disToCent= 8.1
        .
        .
        .
   LM[0267.1,T,<T>] (x,y)=(350,319) (Dx,Dy)=( -4,  8) disToCent= 8.9
   LM[0267.1,U,<T>] (x,y)=(419,306) (Dx,Dy)=( 12,  6) disToCent= 13.4
[0267.1] Mean LM centroid (280,195), LM rmsDev w.r.t.- Rgel= 7.3
```

<CMD> VErify PCG DB

If the file system is suspected of being corrupted such as in the case of a disk crash, file transfer, etc., then the PCG DB may be verified. The `VERIFY PCG DB` command causes the PCG DB to be verified. This consists of checking each Rspot set's spot count and checksum of the CC# indices for all of the spots in it against precomputed counts saved in each Rspot sets internal data dictionary.

If there *is* a problem, it will report the name of the offending Rspot set and the checksum and count values. You may wish to restore your PCG DB from a previously backed up version.

```
79<CMD>: VERIFY PCG DB

Verifying ALL Rspots sets in PCG DB - be patient.
VERIFYING the PCG DB #Rsets=1338 current rspot set block size=6
Starting at 09/09/1989, 01:41:51PM
Finished VERIFYING PCG DB at 09/09/1989, 01:42:02PM!
[01:42:02PM] Real TIME =00:00:11  CPU TIME =00:00:06,  54.55%
```

3.3.11 cgelp2 error messages

In general, **cgelp2** is very robust. However, errors can occur. Typing an incorrect command will most often do nothing or in any case generally be easily recovered from by entering another command to undo the change. This is because **cgelp2** only writes to the PCG DB on the disk for a few selected commands `BACKUP, EXIT, REORDER, EXTRAPOLATE, CREATE` and no harm can be done to the database unless one of these *atomic operation* commands is involved.

types of errors

If you do not want to continue and do not want to checkpoint the PCG DB, just type `CONTROL/C` to kill the program. If you are doing an atomic operation command, it will inform you "`This is an ATOMIC operation. If you exit, you will corrupt the PCG DB`". It will ask you "`Are you sure you want to exit CGELP2(yes/no)[no]?:`". If you answer `yes`, it does not save the state of the current session. Otherwise, it will just continue. You may also abort any operation which processes the PCG DB by typing `CONTROL/C`. This causes the operation to be aborted and control to return to <CMD> command level. Normally, one uses the `EXIT` command to checkpoint the database (if the database was not protected) and terminate the program. Next time you come into the PCG DB, it will be in the state last checkpointed.

fatal erros

There are a number of warning and fatal error messages which can appear in **cgelp2**. These are listed below with a discussion of what should be done (if anything) as a result. WARNING messages may simply indicate a change in the "state" or alternatively a condition which could be a problem which you might want to correct. A DRYROT error is a fatal error condition (i.e BUG) in the operation of **cgelp2** - the module name, line number, and the location where the bug occurred is given in the table. To help us locate the problem, you should report DRYROT errors (with as much detail as to us at the Image Processing Section in the NCI.

WARNING errors

```
(general): WARNING, Accession number 'ACC#' does not exist!

(general): Extending existing .pcg database file from <oldBlkSize> to
            <newBlkSize> blocks/Rspot-set size. Be patient!

(general DBUG-daemon): [<msg>] FOUND DBUG-DAEMON [ACC#<acc#>,CC#<CC#>]
           in Rspot[<Rspot#>]

(general DBUG-daemon): [<msg>] Turn OFF DBUG-DAEMON [ACC#<acc#>,CC#<CC#>]
           in Rspot[<Rspot#>]

(general Prefilter): FAILED-PREFILTER failCode= <n1> <explanation>

BACKUP: Checkpointing PCG database <PCG DB file> ...

BACKUP: Sorry, can't CHECKPOINT if PCG DB is protected!

CREATE: Can't find file: <.gcf input file>

CREATE: Converting D' to % total D'.

CREATE: Defining initial Rgel as <acc#> from file '<.gcf input file>'

CREATE: NOTE: Reading in the eRspot database consisting of US spots
           from gels other than the Rgel! Rgel spots are EP spots.
```

```
CREATE: File <.gcf file name> not found - ignoring entry.

CREATE: Found <# spots accepted> pairs.

CREATE: Ignoring US spots not in Rgel.

CREATE: Long GCF.

CREATE: Redefining Rgel from <acc#> from file '<.gcf input file>'.
          - DB will be inconsistent.

CREATE: Short GCF.

CREATE: Sorry, you can't enter the same gel twice!

DCPLOT: Bad gel Plot spec - ignoring PLOT command.

DDPLOT: Bad gel Plot spec - ignoring PLOT command.

DO: Can't run another script within script file - ignored.

DO: Can't find script file: <scriptFile>

EDIT: Altered Rspot[<g1>:<r1>] index=<CC #>

EDIT: Can't find spot to ALTER.

EDIT: Can't find <ACC#> Rspot[<n1>] to TRANSFER 'from'.

EDIT: Can't find <ACC#> spot <Rspot#> to MERGE 'from'.

EDIT: Can't find slot for spot to TRANSFER 'to'.

EDIT: Can't find spot to DELETE.

EDIT: Can't find spot <Rspot#> to MERGE 'to'.

EDIT: Changed Rspot[<g1>:<r1>] index=<CC #>

EDIT: Deleted Rspot[n1] index=<CC#>

EDIT: Found duplicate spot - can't INSERT unless you DELETE it first.

EDIT: Gel <ACC#> not visible.

EDIT: Illegal spot edit specification.

EDIT: Illegal syntax for specifying spot(s) in Rspot sets.

EDIT: Inserted Rspot[<ACC#>:<Rspot#>]

EDIT: LMsets are different for <g1> Rspot[r1] LM#<lm1> and
```

```
          Rspot[<r2>] LM#<lm2>.

EXIT: Saving PCG database <PCG DB file> ...

EXTRAPOLATE: By aborting, you have an inconsistent database.
          EXTRAPOLATE again to correct this problem.

EXTRAPOLATE: Creating EXTRAPOLATED Rspot sets for gels missing from
          the Rspot set by adding the mean Rspot set (Dx,Dy) to the
          missing gel's landmark position. Extrapolated Rspots are put
          into the search results list. Note the /NOQUIET switch may
          be appended to the command as 'EXTRAPOLATE/NOQUIET' in order
          to print the names of gels and Rspot sets as the spots are
          being extrapolated.

EXTRAPOLATE: Rspot[<Rpost#>] was AP label, changing it to a <SP, PP or CP>

HELP: No help file `cgelp2.hlp' available for on-line HELP.

HELP: Sorry, the `cgelp2.hlp' doesn't have help on `<cmd name>'.

INQUIRE//(general WMW, T, B, F tests): Sorry, but there aren't enough gels
           in these classes to do the test #gels in C<c1>=<n1> #
           gels in C<c2>=<n2>

INQUIRE//EXPRESSION-PROFILE: Can't build expression-profile table for more
           than <n1> entries. Continuing with initial SRL entries.

INQUIRE//EXPRESSION-PROFILE: Must have threshold > 0.0

INQUIRE//EXPRESSION-PROFILE: You entered more ratios than classes:
           ignoring rest of ratios.

INQUIRE//EXPRESSION-PROFILE: You entered fewer ratios than classes:
           assuming 1.0 for rest of classes.

INQUIRE//F-TEST: [FTEST!LKUP] arg err, nClasses=<n1> nGels=<n2> t=<n3>

INQUIRE//PRINT: Can't print illegal Rspot[<n1>] - legal range is Rspots[1:<n2>]

INQUIRE//WMW-test or Kruska-Wallis-test:
          REMINDER - PCG DB must first be sorted using REORDER use non-
          parametric tests. Do REORDER in this density mode and try again.

MOSAIC: Rspot does not exist for gel <g1> - ignoring cmd.

PLOT: You don't really want to plot the feature against itself!

SEQUENTIAL SET OPR: Bad args: index in [1:87], w in [1:87]

SEQUENTIAL SET OPR: Bad arg: w in [2:87]

REORDER: Can't reorder a PCG DB which has not been created!
```

```
REORDER: Checkpointing PCG DB in <PCG DB file>.ckpt BEFORE doing REORDER.

REORDER: Finished checkpointing - now starting REORDER.

REORDER: PROBLEM: Checkpoint failed - aborting REORDER.

REORDER: Sorry, you can't abort without destroying the database - continuing.

SET ACCESSION FILE: Can't find accession file '<acc file>'. No
           change in [<acc file>].

SET CLASSES: Can't do <opr> since NULL study

SET CLASSES: Put <gel acc#> into <class #>

SET CLASSES: There are no gels in the DB. CREATE gel DB then redo SET CLASSES.

SET CLASSES: Gel subset <sub set name> does not exist for class #<c1>
           - try again.

SET DATABASE FILE: Creating new paged CGL database file: ...

SET DATABASE FILE: PCG file was write-protected, doing PROTECT.

SET DATABASE FILE: Using existing PCG paged composite gel database: ...

SET DENSITY MODE: Can't set density mode to Ratio-List mode until you
          have computed the Ratio sums calibration. Use the
          'SET RATIO LIST' normalization command. Then try
          'SET DENSITY MODE' again.

SET DENSITY MODE: Can't set density mode to Least-Squares until you
           have computed the least squares calibration. Use the
           'SET LEAST SQUARES' normalization command. Then try
           'SET DENSITY MODE' again.

SET DENSITY MODE: NOTE: CPM mode currently defaults to ABSOLUTE MODE.

SET DISPLAY: Selecting new display [<display>].

SET FIELDS: Can't find accession file: <gel accession file>

SET FIELDS: Changing gel study titles...

SET GEL SUBSETS: Can't have more than 9 sub sets!

SET GEL SUBSETS//EXPLICIT: <get ACC#> not in the database!

SET GEL SUBSETS: Class name '<gel subset name>' does not exist.

SET GEL SUBSETS: Sorry, you didn't define 1st set - try again.
```

```
SET GEL SUBSETS: Sorry, you didn't define 2nd set - try again.

SET LABEL: New pairing search label [<new search labels>]

SET LEAST SQR CALIB: Doing Least Squares normalization.

SET LEAST SQR CALIB: Warning for gel[ACC#] |b=<val>| > 64.0

SET RGEL: Gel <ACC#> not found in gels you entered so far. No change.

SET SRL SUBSETS//(general): Bad SRL subset name(s).

SET SRL SUBSETS//(general): Can't have more than 88 sub sets!

SET SRL SUBSETS//(general): No set name!

SET SRL SUBSETS//(general): Result in working SRL.

SET SRL SUBSETS//(general): Saved SRL in Set[n1] `<name>'.

SET SRL SUBSETS/CLEAR: Cleared all SRL subsets!

SET SRL SUBSETS//DELETE: Deleting Set #<n1> <<<subset name>>>

SET SRL SUBSETS//FINDKEYWORD: Bad syntax! Try again.

SET SRL SUBSETS//READ: Can't find <.srl file>

SET SRL SUBSETS//READ: NEW Set #<n1> <<<new name>>> |<nbrfound>|

SET SRL SUBSETS//RENUMBER: Renumbered all SRL subsets!");

SET SRL SUBSETS//RESTORE: Restoring SRL from subset [<n1>] with
          <nbrfound> spots. <<<set name>>>

SET SRL SUBSETS//WRITE: Creating SRL subset data File: ...

SET SRL SUBSETS//WRITE: Creating Batch script job: `<smosaicfile>'
          to compute MOSAIC images.

SET WORKING SET: Defining the working set of gels as `<gel subset name>'

SET WORKING SET: Gel <acc#> not in the database - ignoring it!

SET WORKING SET: Gel subset `<sub set name>' is not defined.

SET WORKING SET: Restoring working set to ALL gels.

SET WORKING SET: The current gel working set, of size <n1>, is: ...

TABLE//CORRELATE-GEL-GEL: Can't fit <n1> working gels in the table printout.
          Change the working set of gels to have between 2 and 19 gels
          using the SET WORKING GELS command.
```

```
TABLE//SRL-SRL: You need either Least-square or Ratio SET DENSITY MODE.
          Change the mode from '<mode-density>' and try again.

TABLE//SRL-SRL: Can't fit <n1> Rspot set comparisons in the table printout.
          Edit the SRL to have between 2 and 19 Rspots the SET SRL SUBSETS
          command.
```

DRYROT fatal errors

```
[HASH!C] DRYROT! HASH!C <symbol> table overflow failure.
[rd_block] DRYROT - Lookup error <pcgppn><pcgfilename>
[rd_field] DRYROT - exceeding max allowed PCG DB file size <n>
[rd_field] DRYROT - BAD ARGS rsetnumber=<R> p=<pz> field=<f>
[wt_field] DRYROT - exceeding max allowed PCG DB file size <n>
[wt_field] DRYROT - BAD ARGS rsetnumber=<R> p=<pz> field=<f>
[wt_header] DRYROT - n=<n1> (WRDSPERBLOCK*PCGL_HEADERSIZE)=<n2>
[rd_gsubsets] DRYROT - acc# '<accN>' not an item!
[SET_PCGL_DB] DRYROT! You have REAL-TROUBLE!
            Your .pcg database file is corrupted. Try to get
            an older copy from a backup and recover from that.
[FIN_PCGL_DB] DRYROT - corrupted .pcg database file!
            Restore .pcg file from last backup!
[EXTEND_PCGL] DRYROT - VERIFY ERROR
            RSetNumber=<n1> pz=<n2> prevPz=<n3> count=<n4>
            maxnodesperRset=<n5> accIdx=<n6>
[NEW!RCRD] DRYROT! > <n1> max # blocks/PCG - too many gels and/or spots.
[NEW!RCRF] DRYROT - can't allocate new record for RSetNumber=<n1>
[CK!LIMITS!RSPOT] DRYROT - Rspot[<n1>] nodeCnt=<n1> p1=<n2> oldp1=<n3>.
[CK!LIMITS!RSPOT] DRYROT.1 spot[p1=<n1>]=...
[CK!LIMITS!RSPOT] DRYROT.2 spot[oldp1=<n1>]=...
[PRT!SPOTS] - DRYROT - rSetNumber=<n1>
[DMPSPSS!CGL] DRYROT - out_file_flag is OFF!
[DMPSPOTFEATURES] DRYROT - out_file_flag is OFF!
[READ!RCD] DRYROT - BAD accN1Ptr=<n1> or accN2Ptr=<n2>
[PUT!IN!ORDER] DRYROT - Tried to put NULL record into Rspot[<n1>] list.
[PUT!IN!ORDER] DRYROT - no EOL nCnt=<n1>
[PUT!IN!ORDER].1 DRYROT - no EOL nCnt=<n1>
[SET!SRL!SUBSET] DRYROT - Set #<n1>
[EVAL!OPEN] DRYROT - bad smstate->pixnbr=<n1>
[Wilcoxon-Mann Test WMWT] DRYROT - (n1<1 | n2<1 | n2<n1 | t<1 | t>4).
[tab_prt_rank_order_table()] DRYROT too man duplicates - aborting table.
[PICK!MENU!ENTRY] DRYROT - Bad Menu array.
```

3.4 cmpgl2 - pair spots between two 2D gels

The **cmpgl2** program pairs corresponding spots from two GSF spot list files for use with GELLAB-II to later construction of the composite database. This pairing is performed on two Gel Segmentation Files (GSF) produced by the **sg2gii** gel segmentation program. The *.gsf* files are specified by their accession numbers. There must also exist a landmark (LM) *.lm* database file containing a list of landmarks for these two gels (as well as other gels).

If composite spots are to be defined during gel pairing, it also expects a composite CC# spot declaration (*.cg*) file specified with the `-CCfile` switch.

paired spot lists

The output is a Gel Comparison File (GCF) or *.gcf* file which may be used as input to the **cgelp2** program. **Cmpgl2** is documented in ([LipL80a], [LemP81b], [LemP83a]) and in ALGORITHM CMPGL2 in Appendix G. The GSF and GCF files and generated images are on the path specified by `ppnp3x` (see `gel.rc` discussion in Section 1.6.5 page 61).

Cmpgl2 first reads in the landmark set from the landmark database file and then reads the two GSF spot list files. The mutually best fitting spot (for each gel) is found for each corresponding landmark in the landmark set. If the euclidean distance is $> dT_2$, then the landmark is marked `NG` (no good). If the spot connected component (CC#) index (**sg2gii**) is the same for two landmarks, the later duplicate landmarks are marked `SM` (the same). This means that the LM entry is probably corrupted. Otherwise if the distance from the estimate LM to the CC# spot centroid $< dT_1$, then it is marked `OK`.

using landmarks

The landmark coordinates for `OK` spots are replaced with the centroid of the segmented GSF spot otherwise it contains the coordinates specified by the user in the LM landmark entry. It then computes the set of *half-radii* R_i for each landmark which is 1/2 the distance from a LM spot to the nearest LM spot. Spots in each gel are then sorted into these landmark sets. They are then paired within each set. Spots not paired are reported as US spots (unresolved spots). Figure 3.9 shows a set of landmark spots generated using the `-ONLYMARKLANDMARKS`.

If either LM coordinate is outside of the computing window (CW) for the corresponding gel, then that LM set is not used when partitioning GSF spots but is output later as an empty GCF LM set.

Figure 3.9. Sample **cmpgl2** set of landmarks image. The paired spots derived images are created using the `-ONLYMARKLANDMARKS` switch option.

labeled-pair images

Cmpgl2 can also be used to generate optional spot-pairing labeled images using the `-Mark` switch. It will generate two images designated `u`*xxxxx*`.ppx` and `v`*xxxxx*`.ppx` (corresponding to G1 and G2) where *xxxxx* is the last 5 characters of the G2 picture filename given in the accession file entry.

landmark images

When using the `-OnlyMarkLMS` switch, it can alternatively generate Rmap images with `l` (letter L) prefix. These consist of the original images with the LM set positions and names superimposed - no GCF file is produced. In both cases, pseudo-color spot labels can be generated using the `-GraphScale` switch (default `-GRAPHSCALE`). In which case, instead of using white labels: landmarks and SP are red, PP are yellow, AP are blue, US are cyan and CP are green. Figure 3.10 shows a pair of spot-pairing images.

Figure 3.10. Sample **cmpgl2** paired spots derived images. The paired spots derived images are created using the `-MARK` switch option. The sure-pair spots are S, possible-pair are P, ambiguous-pairs are A and unresolved are U. Landmarks are larger letter spots.

In addition to using **sg2gii** *.gsf* data for its input, one can specify *composite spot groups* which are treated as a single spot. Their density being the sum of the composite spot densities and the (x,y) centroid being the weighted centroid. These *Composite Pair* spots are only defined in **cmpgl2** - not in the old GELLAB-I CMPGEL program. The CP spots are specified as a list of corresponding CC#s for each gel and are entered via a `ccg`*xxx*`.cg` file where xxx is the project name. The format for CP entries is:

`.cg` file format

```
<spot-name>/<ACC#>={CCi,Ccj,...,CCz}

For example,
        I/0050.1={244,250,256,258,259,263,266,279,284,291}
        II/0050.1={551,552,553,557,558,559,560,563,565}
```

```
III/0050.1={633,634,635,636,637,638,639,642}
. . .
I/0050.2={66,69,71}
II/0050.2={95,96,97,98}
III/0050.2={101,102}
```

So in the above example, CP spots `II/0050.1` and `II/0050.2` are equivalent in the two gels even though they contain different numbers (as well as spot labels themselves) of spots.

Format of the *.gcf* file

The paired gels database is dumped in a *.gcf* format ASCII data file (See **EXAMPLE 1**. This file will be read later by **cgelp2**. The output is an initial header region, followed by single line records (for each spot in one or both gels), followed by a trailer region. The single line record is of the form:

`.gcf` *file format*

```
#<LM#> G1:<i1>[<x1>,<y1>]G2:<i2>[<x2>,<y2>] <code>,<dP>,<dL>,
  <D'1>,<D'2>,<maxD1>,<maxD2>,<Area1>,<Area2>,<minD1>,<minD2>,
  <sX1>,<sX2>,<sY1>,<sY2>, <sXcentroid>,<sYcentroid>,
  <sMnDprime>,<sdMnArea>,<nCgel>,<CC#-of-group1>,<CC#-of-group2>,
  <#Entries-Grp1>,<#Entries-Grp2>,<List-group1>,<List-group2>,
  B<mnBackground1>,<mnBackground2>
```

NOTE: <nCgel> is greater than 0 if the spot is a Cgel′ spot or it is a composite CP spot. The $<CC\#$-of-group> is non-zero if the spot is a member of a CP spot and the CC# is that of the CP spot (which also points to itself). This lets **cgelp2** build an inverted spot-group list. The <#Entries-Grp> of values which are used to determine the strategy for reading the subsequent group lists (<List-group>).

Spot labels

As spots are paired between gels, they are labeled with one of the following five labels:

```
US - unresolved spot code 0
SP - sure pair code 1
PP - possible pair code 2
AP - ambiguous pair code 3
EP - Extrapolated pair code 4 (Only used in CGELP2)
CP - Composite pair code 5
```

```
USAGE:
          cmpgl2 [<Opt. -switches>] <acc# Rgel> <acc# gel2>
```

Type `cmpgl2 -info` to get more information. NOTE: the two gel accession number arguments MUST appear consecutively on the command line with the Rgel appearing first.

SWITCHES

-Black use black labels in **-MARKed** pictures instead of white.

-CCfile:ccgxxx.cg create CP (Composite Pair) spots from CC data in the ccgxxx.cg (for project xxx) multiple lines of data of the form: <`spot-name`>/<`ACC#`>={`CCi,Ccj,...,CCz`}.

-CHange:dT_1**,**dT_2 change the dT_1 and dT_2 parameters specification (default is (5,10 for 512x512 and 10,20 for 1024x1024).

-COMMutativelandmarks search for either (G1,G2) or (G2,G1) instead of just (G1,G2) in the LM database file.

-COMPress derived images and GCF file after they are created on the disk to save space.

-Debug print debugging information on terminal and *.gcf* file.

-Graphscale Display image in GraphScale (for **Xpix**) pseudo color graphics where: instead of using white labels: landmarks and SP are red, PP are yellow, AP are blue, US are cyan and CP are green (default).

-Info print additional information about **cmpgl2**.

-LatchLandmarkSpots upgrade the LMS DB landmark spot position to that of the nearest GSF spot if the distance is less than T1 (default).

-Mark:<alphabetic> label all spots in the images generating u*i*`.ppx` and v*i*`.ppx` for G1 and G2 respectively (i is G2 picture number). The labels are +=US, S=SP, P=PP, A=AP, C=CP, unless the <`Alphabetic`> option is specified, then print letters of the alphabet in ascending order to label paired spots.

-OnlyMarkLMS label the LM spots only and do not perform spot pairing. I.e. no *.gcf* file is generated. The images generated are `l`*nnnnn*`.ppx` and *mmmmm*`.ppx` for G1 and G2 respectively (*nnnnn* and *mmmmm* are the G2, non-Rgel, 5-digit picture number). Eg. For picture name *a12345*, the picture number is `12345`.

-Percentdensity output percent density measurement instead of D'.

-SEcondaryPairing perform secondary pairing on AP and US spots (default).

-SHortGCF Output the *.gcf* without as much ASCII feature labeling (default).

-SWlist:"<List of switches>" allows passing of a switch list to child processes. For example: `-swList:"-ppx -rgb"` passes the switches `-ppx -rgb`.

-UncorrectedDensity Use the uncorrected density D instead of the default density D' (default is `-NOUnCorrectedDensity`).

-VErsion print the version number of the program.

-VOlume use volume density measurement instead of D'.

-WmWait When done, wait until do CLICK TO EXIT `mwwait` widget to exit.

-Xpix display processed images using **Xpix** X Windows display program.

EXAMPLES OF USAGE

```
cmpgl2 324.1 369.1
        # Default.

cmpgl2 324.1 369.1 -Mark
        # Also generate the U and V paired spot label images.

cmpgl2 324.1 369.1 -Mark
        # Generate paired spot images.

cmpgl2 324.1 369.1 -Mark -Xpix -SWlist:"-graphscale -full"
        # Same as above, but display them with Xpix and
        # pass switches to Xpix.

cmpgl2 324.1 369.1 -OnlyMarkLMS
        # Generate landmark 'l' images of LMS. Do not make GCFs.

cmpgl2 56.1 50.2 -CCFILE:ccgps7.cg
        # Also specify composite pair spots
```

EXAMPLE 1. Pairing of two gels using the standard dT_1 and dT_2 threshold defaults values. *pairing two gels*

```
28% cmpgl2 324.1 384.1 -changeParamters:5,10 -shortGSF

CMPGL2: Version February 9, 1989
Today's date is 02/23/1989, 01:30:43PM
User:/home/opus/lemkin/gellab/demo/
Written 1981-1989, P. Lemkin.
Gel Comparison File is: /home/opus2/lemkin/gellab/demo/aux/c10384.gcf from p10324.gsf and p10384.gsf
0324.1/.../?/?/1-18-82/#12/CULT #2/3:10,5-20%/
0 HRS/H3/2 HRS/96 HRS/HEME MALIG-AML,MYELOID/
B00661/-NONE-/-NONE-/VIDICON-AUTO,28MM,F8,76CM/LESTER*
```

```
 027 049 072 095 117 136 153 168 181 192 200 208 213 220 225 000 036 497 074 509
0384.1/.../?/?/1-18-82/#17/CULT #1/3:10,5-20%/
0 HRS/H3/2 HRS/720 HRS/HEME MALIG-HCL,LYMPHOID/
B00981/-NONE-/-NONE-/VIDICON-AUTO,28MM,F8,76CM/LESTER*
 023 047 071 095 114 135 153 167 181 191 199 208 213 223 000 000 048 500 066 502
Distance sizing limits (dT1 = 5.00, dT2 = 10.00)

Switches:  -SHORTGSF -CHANGEPARAMTERS:5,10
G1 CW[36:497,74:509]
G2 CW[48:500,66:502]
The LM[A] is INside CW
The LM[B] is INside CW
The LM[C] is INside CW
The LM[D] is INside CW
The LM[E] is INside CW
The LM[F] is INside CW
The LM[G] is INside CW
The LM[H] is INside CW
The LM[I] is INside CW
The LM[J] is INside CW
The LM[K] is INside CW
The LM[L] is INside CW
The LM[M] is INside CW
The LM[N] is INside CW
The LM[O] is INside CW
The LM[P] is INside CW
The LM[Q] is INside CW
The LM[R] is INside CW
The LM[S] is INside CW
The LM[T] is INside CW
The LM[U] is INside CW
The LM[V] is INside CW
G1 HAS  723, G2 HAS  764 SPOTS, AND WITH    0 CP SPOTS.
TOTAL DENSITY G1=5984.80, G2=8613.70
OMITTED TOTAL DENSITY G1= 842.70, G2= 465.50


PAIRING STATISTICS
------------------
After Initial pairing:
US 245
SP 184
PP 544
AP 304
CP 0
0.5(SP+PP)/(|G1| MIN |G2|)=50.3%

After secondary pairing:
US 234
SP 184
PP 568
AP 291
CP 0
```

```
0.5(SP+PP)/(|G1| MIN |G2|)=52.0%

mean dP(SP+PP)=5.37 mean dP'((|G1|+|G2|)/(SP+PP))=6.90

Real TIME =00:00:57  CPU TIME =00:00:51,  89.47%
Finished pairing 0324.1 with 0384.1.
Output in /home/opus2/lemkin/gellab/demo/aux/c10384.gcf
```

EXAMPLE 2. Pairing of two gels with Composite Pair spots database specified.

pairing gels using composite spots

```
26% cmpcg2 56.1 50.1 -Short -Changep:5,10 -Ccfile:ccgps7.cg


CMPGL2: Version January 16, 1989
Today's date is 01/19/1989, 02:34:35PM
User:/home/joeUser
Written 1981-1989, P. Lemkin.
Gel Comparison File is: c10050.gcf from p10056.gsf and p10050.gsf
0056.1/.../10-23-86/-/5-5-86/#56/MEDIUM/1,6% 5-8,0.5% 3-10,0.8% 4-6.5, 10%GEL/
5DAYS/S35/40HRS/35DAYS/ONE/
B00156/-NONE-/-NONE-/VIDICON-AUTO,55mm1:2.8,F11,92CM/JOEUSER*
 080 094 113 133 149 164 175 185 192 198 203 206 208 000 000 000 066 511 042 511
0050.1/.../10-23-86/-/4-28-86/#50/MEDIUM/1,6% 5-8,0.5% 3-10,0.8% 4-6.5, 10%GEL/
5DAYS/S35/40HRS/42DAYS/SPECIAL/
B00150/-NONE-/-NONE-/VIDICON-AUTO,55mm1:2.8,F11,92CM/JOEUSER*
 083 097 115 135 151 166 178 187 195 200 205 209 211 000 000 000 066 511 042 511
Distance sizing limits (dT1 =  5.00, dT2 =  10.00)

Switches:  -SHORT -CHANGEP:5,10 -CCFILE:ccgps7.cg

lmsps1.lm from gel ACC#'s 0056.1 and 0050.1
The (Representative) R-gel is: 0056.1
 LANDMARK #A  G1[204, 188], G2[174, 204]
 LANDMARK #B  G1[260, 190], G2[232, 210]
 LANDMARK #C  G1[363, 169], G2[375, 183]
 LANDMARK #D  G1[364, 133], G2[369, 148]
        . . .
 LANDMARK #S  G1[201, 279], G2[166, 302]
G1 CW[66:511,42:511]
G2 CW[66:511,42:511]
The LM[A] is INside CW
The LM[B] is INside CW
The LM[C] is INside CW
The LM[D] is INside CW
        . . .
The LM[S] is INside CW
Reading Composite spots for GEL#1 from file CCGPS7.CG
Reading Composite spots for GEL#2 from file CCGPS7.CG
G1[A,  62][ 204, 188],E.Diff=  .0, G2[A, 113][ 174, 204],E.Diff=  .0-OK
```

```
G1[B,  71][ 260, 190],E.Diff=  .0, G2[B, 123][ 232, 210],E.Diff=  .0-OK
G1[C,  59][ 363, 169],E.Diff=  .0, G2[C,  96][ 375, 183],E.Diff=  .0-OK
G1[D,  47][ 364, 133],E.Diff=  .0, G2[D,  75][ 369, 148],E.Diff=  .0-OK
       . . .
R[A]=  23 to nearest LMs[R,R], next nearest LMs[B,B]
R[B]=  28 to nearest LMs[A,A], next nearest LMs[C,C]
R[C]=  18 to nearest LMs[D,D], next nearest LMs[B,B]
R[D]=  18 to nearest LMs[C,C], next nearest LMs[B,B]
       . . .
R[S]=  14 to nearest LMs[N,N], next nearest LMs[E,E]
G1 segmenter parameters:
Switches:  -ALLOWTCHEDGES -SHORTGSF -7X7LOWPASS
Window [66:508,42:508]
Spot Area sizing limits ( 16.00: 500.00)
Integrated Density sizing limits ( .70: 500.00)
Density difference sizing limits ( .03: 3.90)
Zonal Notch filter background window 32x32

G2 segmenter parameters:
Switches:  -ALLOWTCHEDGES -SHORTGSF -7X7LOWPASS
Window [66:508,42:508]
Spot Area sizing limits ( 16.00: 500.00)
Integrated Density sizing limits ( .70: 500.00)
Density difference sizing limits ( .03: 3.90)
Zonal Notch filter background window 32x32
G1 HAS  332, G2 HAS  685 SPOTS, AND WITH   8 CP SPOTS.
TOTAL DENSITY G1= 4609.31, G2= 9566.71
OMITTED TOTAL DENSITY G1= 5001.29, G2= 6305.56
LM[A] G1 HAS       7, G2 HAS      20 SPOTS
LM[B] G1 HAS 6, G2 HAS 20 SPOTS
LM[C] G1 HAS 2, G2 HAS 13 SPOTS
       . . .
LM[S] G1 HAS 52, G2 HAS 58 SPOTS

PAIRING STATISTICS
------------------
After Initial pairing:
US 398
SP 104
PP 242
AP 144
CP 8
0.5(SP+PP)/(|G1| MIN |G2|)= 52.1%

After secondary pairing:
US 391
SP 104
PP 252
AP 141
CP 8
0.5(SP+PP)/(|G1| MIN |G2|)= 53.6%

mean dP(SP+PP)=5.09 mean dP'((|G1|+|G2|)/(SP+PP))=10.45
```

```
CC#=329[CP] grp=5 (0056.1,V) |n|=1{64}
  xC=193 yC=187 D'C= 26.23 AreaC=72 dMin= .19 dMax= 1.02
  dP= 2.80 dL=0 sdX= 1.61 sdY= 2.55 sdXY= 1.66 in LM[A] n-n-LM[B] nxtLMSptr=330
 paired with CC#=682[CP] grp=5 (0050.1,V) |n|=1{117}
  xC=161 yC=205 D'C= 109.81 AreaC=134 dMin= .24 dMax= 2.22 bst-prCC#329
  dP= 2.80 dL=0 sdX= 2.10 sdY= 2.55 sdXY= 2.11 in LM[A] n-n-LM[A] nxtLMSptr=1015
CC#=330[CP] grp=6 (0056.1,VI) |n|=1{62}
  xC=204 yC=188 D'C= 92.31 AreaC=135 dMin= .03 dMax= 2.36
  dP= .00 dL=0 sdX= 1.90 sdY= 2.55 sdXY= 1.95 in LM[A] n-n-LM[B] nxtLMSptr=0
 paired with CC#=683[CP] grp=6 (0050.1,VI) |n|=1{113}
  xC=174 yC=204 D'C= 257.35 AreaC=254 dMin= .26 dMax= 2.26 bst-prCC#330
  dP= .00 dL=0 sdX= 2.55 sdY= 2.55 sdXY= 3.04 in LM[A] n-n-LM[A] nxtLMSptr=0
CC#=331[CP] grp=7 (0056.1,VII) |n|=1{70}
  xC=242 yC=191 D'C= 11.26 AreaC=59 dMin= .12 dMax= .45
  dP= 2.00 dL=0 sdX= 2.16 sdY= 1.99 sdXY= 1.54 in LM[B] n-n-LM[A] nxtLMSptr=332
 paired with CC#=684[CP] grp=7 (0050.1,VII) |n|=1{128}
  xC=214 yC=209 D'C= 90.03 AreaC=142 dMin= .17 dMax= 2.03 bst-prCC#331
  dP= 2.00 dL=0 sdX= 2.55 sdY= 2.55 sdXY= 2.23 in LM[B] n-n-LM[B] nxtLMSptr=1017
CC#=332[CP] grp=8 (0056.1,VIII) |n|=1{71}
  xC=260 yC=190 D'C= 41.37 AreaC=79 dMin= .05 dMax= 1.76
  dP= .00 dL=0 sdX= 1.93 sdY= 2.36 sdXY= 1.51 in LM[B] n-n-LM[A] nxtLMSptr=0
 paired with CC#=685[CP] grp=8 (0050.1,VIII) |n|=1{123}
  xC=232 yC=210 D'C= 168.75 AreaC=196 dMin= .18 dMax= 2.16 bst-prCC#332
  dP= .00 dL=0 sdX= 2.55 sdY= 2.55 sdXY= 2.39 in LM[B] n-n-LM[B] nxtLMSptr=0
CC#=326[CP] grp=2 (0056.1,II) |n|=9{258,259,260,264,267,268,269,270,271}
  xC=351 yC=419 D'C= 198.75 AreaC=458 dMin= .11 dMax= 1.51
  dP= 6.00 dL=0 sdX= 1.75 sdY= 2.55 sdXY= 1.43 in LM[Q] n-n-LM[M] nxtLMSptr=327
 paired with CC#=679[CP] grp=2 (0050.1,II) |n|=9{551,552,553,557,558,559,560,563,565}
  xC=317 yC=439 D'C= 204.30 AreaC=542 dMin= .12 dMax= 1.93 bst-prCC#326
  dP= 6.00 dL=0 sdX= 1.89 sdY= 2.55 sdXY= 1.60 in LM[Q] n-n-LM[Q] nxtLMSptr=1012
. . .
Real TIME =00:01:35 CPU TIME  =00:01:13,   76.842%
```

EXAMPLE 3. Example of Gel Comparison File. *GCF output file*

```
CMPGL2: Version October 18, 1988
Today's date is 10/21/1988, 07:06:24AM
User:/home/joeUser
Written 1981-1988, P. Lemkin.
Gel Comparison File is: /home/joeUser/gellab/aux/c10569.gcf from p10524.gsf and p10369.gsf
0524.1/TCA/?/?/1-18-82/#12/CULT #2/3:10,5-20%/
0 HRS/H3/2 HRS/96 HRS/CULT 1/
A00661/-NONE-/-NONE-/VIDICON-AUTO,28MM,F8,76CM/JOEUSER*
 027 049 072 095 117 136 153 168 181 192 200 208 213 220 225 000 036 497 074 509
0569.1/TCA/?/?/3-10-82/#2/CULT #2/3:10,5-20%/
0 HRS/H3/2 HRS/96 HRS/CULT 2/
A00889/-NONE-/-NONE-/VIDICON-AUTO,28MM,F8,76CM/JOEUSER*
 028 051 075 098 118 138 155 169 183 192 200 208 215 225 229 000 061 505 068 503
Distance sizing limits (dT1 = 5.00, dT2 = 10.00)
```

```
Switches:  -MARK

/home/joeUser/gellab/lms/lms.lm from gel ACC#'s 0524.1 and 0569.1
The (Representative) R-gel is: 0524.1
 LANDMARK #A  G1[304, 190], G2[284, 176]
 LANDMARK #B  G1[335, 151], G2[315, 140]
 LANDMARK #C  G1[353, 190], G2[333, 173]
        . . .
 LANDMARK #V  G1[226, 417], G2[196, 392]
G1 CW[36:497,74:509]
G2 CW[61:505,68:503]
The LM[A] is INside CW
The LM[B] is INside CW
The LM[C] is INside CW
        . . .
The LM[V] is INside CW
G1[A,206][302,192],E.Diff= 2.8, G2[A,242][283,177],E.Diff= 1.4-OK
G1[B,119][332,153],E.Diff= 3.6, G2[B,133][315,138],E.Diff= 2.0-OK
G1[C,199][354,190],E.Diff= 1.0, G2[C,234][330,177],E.Diff= 5.0-OK
        . . .
G1[V,611][223,418],E.Diff= 3.2, G2[V,708][194,393],E.Diff= 2.2-OK
R[A]=13 to nearest LMs[D,D], next nearest LMs[B,C]
R[B]=21 to nearest LMs[C,C], next nearest LMs[A,A]
R[C]=15 to nearest LMs[D,D], next nearest LMs[B,B]
        . . .
R[V]=45 to nearest LMs[U,U], next nearest LMs[P,P]
G1 segmenter parameters:Switches:  CTL
Window [36:497,74:507]
Spot Area sizing limits (10:2000)
Integrated Density sizing limits (0.30:500.00)
Density difference sizing limits (0.03:2.70)
Zonal Notch filter background window 32x32
Background range [0.00:0.89] OD

G2 segmenter parameters:Switches:
Window [61:505,68:503]
Spot Area sizing limits (10:2000)
Integrated Density sizing limits (0.30:500.00)
Density difference sizing limits (0.03:2.70)
Zonal Notch filter background window 32x32
Background range [0.00:0.66] OD

G1 HAS  676, G2 HAS  813 SPOTS, AND WITH    0 CP SPOTS.
TOTAL DENSITY G1=5951.60, G2=16302.40
OMITTED TOTAL DENSITY G1= 845.10, G2= 472.10

LM[A] G1 HAS 9, G2 HAS 20 SPOTS
#A,237[6,-24]187[3,-28]P,5.0,28,4.880,2.570,0.140,0.270,117,40,0.020,0.120,2.640,0.940,1.980,0.810,0.00,0.00,0.00,17
#A,253[18,-16]216[14,-16]P,4.0,24,4.030,11.920,0.150,0.570,60,72,0.050,0.140,1.720,1.130,1.620,1.170,0.00,0.00,0.00,1
#A,271[21,-11]225[22,-13]P,2.2,25,1.570,4.700,0.150,0.460,35,82,0.030,0.030,1.400,1.070,1.210,1.450,0.00,0.00,0.00,9
#A,272[5,-8]248[7,-7]S,2.2,9,6.470,1.480,0.280,0.610,80,32,0.030,0.230,1.980,0.410,1.850,1.110,0.00,0.00,0.00,13.30,0
#A,297[0,0]258[0,0]S,0.0,0,34.560,243.040,1.380,2.010,55,240,0.180,0.250,1.330,2.150,1.490,2.560,0.00,0.00,0.00,9.80,
#A,325[5,14]288[4,12]P,2.2,14,18.260,72.440,0.640,1.500,84,166,0.050,0.250,1.870,1.780,1.870,2.290,0.00,0.00,0.00,13
#A,330[-19,16]295[-20,13]P,3.1,24,13.950,16.600,0.540,0.860,56,99,0.050,0.090,1.370,1.280,1.460,1.440,0.00,0.00,0.00,
#A,335[-6,17]287[-5,12]P,5.0,18,11.730,60.790,0.360,1.270,88,129,0.030,0.150,1.620,1.400,2.290,2.020,0.00,0.00,0.00,1
```

```
#A,237[6,-24]202[1,-22]A,5.3,24,4.880,9.850,0.140,0.400,117,87,0.020,0.110,2.640,1.070,1.980,1.660,0.00,0.00,0.00,17.20,0,0,0,0,0,,,B0.0
#A,0[0,0]206[-4,-21]U,24.0,24,0.000,14.060,0.140,0.460,117,92,0.020,0.150,2.640,1.120,1.980,1.710,0.00,0.00,0.00,17.20,0,0,0,0,0,,,B0.02
#A,0[0,0]224[-5,-13]U,13.0,13,0.000,2.150,0.280,0.310,80,74,0.030,0.050,1.980,1.240,1.850,1.210,0.00,0.00,0.00,13.30,0,0,0,0,0,,,B0.050,
#A,272[5,-8]229[2,-13]A,5.8,13,6.470,5.220,0.280,0.440,80,61,0.030,0.100,1.980,1.420,1.850,1.110,0.00,0.00,0.00,13.30,0,0,0,0,0,,,B0.050
#A,297[0,0]243[-7,-7]A,9.8,9,34.560,1.890,1.380,0.460,55,32,0.180,0.110,1.330,0.910,1.490,0.580,0.00,0.00,0.00,9.80,0,0,0,0,0,,,B0.050,0
#A,272[5,-8]244[1,-7]A,4.1,9,6.470,15.540,0.280,1.260,80,26,0.030,0.250,1.980,0.970,1.850,0.550,0.00,0.00,0.00,13.30,0,0,0,0,0,,,B0.050,
#A,271[21,-11]249[15,-5]A,8.4,23,1.570,6.190,0.150,0.440,35,102,0.030,0.230,1.400,1.780,1.210,1.530,0.00,0.00,0.00,9.10,0,0,0,0,0,,,B0.0
#A,0[0,0]259[-19,-1]U,24.0,24,0.000,7.240,0.540,0.620,56,46,0.050,0.100,1.370,1.120,1.460,0.800,0.00,0.00,0.00,10.00,0,0,0,0,0,,,B0.030,
#A,297[0,0]260[-8,2]A,8.2,8,34.560,32.020,1.380,1.210,55,69,0.180,0.250,1.330,0.640,1.490,2.270,0.00,0.00,0.00,9.80,0,0,0,0,0,,,B0.050,0
#A,0[0,0]265[12,0]U,12.0,12,0.000,9.910,0.280,0.970,80,36,0.030,0.250,1.980,0.930,1.850,0.640,0.00,0.00,0.00,13.30,0,0,0,0,0,,,B0.050,0.
#A,0[0,0]271[-26,4]U,26.0,26,0.000,105.490,0.540,1.360,56,227,0.050,0.250,1.370,3.470,1.460,1.650,0.00,0.00,0.00,10.00,0,0,0,0,0,,,B0.03
LM[B] G1 HAS 50, G2 HAS 64 SPOTS
#B,247[-20,19]203[-16,17]P,4.4,27,1.390,2.060,0.100,0.250,37,33,0.030,0.100,1.020,0.980,1.770,0.770,0.00,0.00,0.00,9.30,0,0,0,0,0,,,B0.0
#B,30[-48,-79]0[0,0]U,92.0,92,3.840,0.000,0.620,0.050,10,64,0.200,0.000,0.800,1.000,0.000,0.890,0.00,0.00,0.00,0.00,0,0,0,0,0,,,B0.020,0
#B,31[-32,-79]0[0,0]U,85.0,85,3.610,0.000,0.470,0.050,14,64,0.110,0.000,1.210,1.000,0.000,0.890,0.00,0.00,0.00,0.00,0,0,0,0,0,,,B0.020,0
#B,32[12,-79]0[0,0]U,79.0,79,0.890,0.000,0.200,0.050,10,26,0.020,0.010,0.870,0.850,0.000,0.250,0.00,0.00,0.00,0.00,0,0,0,0,0,,,B0.010,0.
      . . .
LM[V] G1 HAS 62, G2 HAS 72 SPOTS
#V,712[46,13]719[40,9]P,7.2,47,9.640,70.650,0.430,1.120,65,166,0.030,0.020,1.290,1.750,1.740,1.850,0.00,0.00,0.00,10.40,0,0,0,0,0,,,B0.0
#V,640[-6,-57]609[-5,-57]P,1.0,57,1.280,40.760,0.100,0.720,48,107,0.020,0.120,1.030,1.140,1.660,1.760,0.00,0.00,0.00,8.90,0,0,0,0,0,,,B0
#V,642[-34,-52]613[-29,-52]P,5.0,62,0.920,12.920,0.030,0.270,82,117,0.000,0.020,1.660,1.250,2.330,1.740,0.00,0.00,0.00,15.10,0,0,0,0,0,,
#V,643[-102,-51]0[0,0]U,99.0,114,5.350,0.000,0.230,0.070,66,138,0.000,0.000,1.230,1.840,1.700,1.490,0.00,0.00,0.00,9.30,0,0,0,0,0,,,B0.0
      . . .
#V,0[0,0]831[-130,104]U,99.0,166,0.000,0.710,0.440,0.020,32,84,0.000,0.000,1.080,1.040,1.110,1.400,0.00,0.00,0.00,7.30,0,0,0,0,0,,,B0.00



PAIRING STATISTICS
------------------
After Initial pairing:
US 229
SP 224
PP 488
AP 319
CP 0
0.5(SP+PP)/(|G1| MIN |G2|)=52.7%

After secondary pairing:
US 212
SP 224
PP 520
AP 304
CP 0
0.5(SP+PP)/(|G1| MIN |G2|)=55.0%

Real TIME =00:01:27  CPU TIME =00:01:20,  91.95%
```

3.5 dendrogram - cluster analysis plots

clustering

The **dendrogram** program [SonP86] computes and plots a dendrogram tree clustering a group of *np* objects (Operational Taxonomic Units, OTU) based on their *nv* features in an *nv*-dimensional space. It generates dendrogram plots from *.inq* and *.sps* data files. The algorithm was derived from a BASIC program which appeared in BYTE [SpeR84].

A set of Rspots may be clustered as a function of their expression profile across different gel experimental classes. Alternatively, the working set of gels may be clustered as a function of a Search Results List (SRL) set of marker Rspots. Figure Figure 3.11 shows an example of Rspots (in the SRL) clustered as a function of the expression profile. Figure 3.12 shows an example of gels in the working set clustered as a function of "marker" Rspots in the SRL. There are three types of data input files for **dendrogram**:

I. GELLAB *order histogram .inq* data files generated by the `ORDER HISTOGRAM` subcommand in the `INQUIRE/FILE` command. The file format is specified below. The *objects*={Rspots} and the *features*={mi/mj} mean density value permutations for $\{i, j | (i \neq j)\ And\ (i, j \in Classes - of - gels)\}$. That is, Rspots are clustered as a function of gel expression profiles.

II. GELLAB 'SPSS' *.sps* data files generated using the SPSS command to generated a set of *objects*={gels} and *features*=density values of {Rspots}. Alternatively, the SPSS data can be used to cluster Rspots objects with mean class densities as the features using the `-MEANCLASSES` switch. If `-BothMeanClassAndRatios` is specified, then the objects are Rspots and the features are Mean Class Densities as well as Ratios. The file format is specified below.

III. Non-GELLAB data files with any other file extension. These consist of:

1. TITLE LINE.
2. FEATURE NAMES LINE (n feature names with no spaces in the names).
3. OBJ-1 (n feature values separated by spaces or tabs).
4. . . .
5. OBJ-k (n feature values separated by spaces or tabs).

Figure 3.11. Sample **dendrogram** shows an example of Rspots (in the SRL) clustered as a function of the expression profile.

Figure 3.12. Sample **dendrogram** shows an example of gels in the working set clustered as a function of "marker" Rspots in the SRL.

DATA FILE FORMAT

The three types of data input files which **dendrogram** can accept are described below. The data for the **cgelp2** 'Order-Histogram' data *.inq* file is generated as follows: In using **cgelp2** in GELLAB, it is assumed that we have done the preliminary exploration data analysis and thus are *given* the k classes and np Rspots found to be of interest. Then,

```
(1) Open up all K classes and gels of interest in CGELP2.
(2) Set the SRL to the set of spots of interest, e.g.
        <CMD>: SET SRL SUBSETS
        *Restore SRL subset
        *70 - i.e. SRL[70]
(3) Generate the order histogram of the current SRL in a FILE.
        <CMD>: INQUIRE
        *Order histogram/FILE
        *File [000001.inq]?: <CR>
```

```
<CMD>: EXIT
```

You are now ready to use file `000001.inq` with the **dendrogram** program. Note: values which are missing in the order histogram are indicated by '-' entries. These are mapped here to '0'.

Typical *.inq* Input Data File

1. The first line is repeated in the DENDROGRAM at the bottom.
2. Classes are read into a list of class names for possible future use.
3. The first M_i/M_j table is read and ignored since it only contains sign-change $(<, =, >)$ type data.
4. The second M_i/M_j table data is read as feature names. Then the following data is read 1 object (i.e. Rspot)/line. Any entries which are '-' are defaulted to a 0 value.

```
File: 000001.inq 10/03/1984, 09:58:09 AM from: tstpcg.pcg
                        . . .
Class #  1(CULTURE VSC)=0014.1,  . . .
Class #  2(W.O. CO-CULT)=0001.4,  . . .
Class #  3(SCNN)=0027.1,         . . .
Class #  4(DRGNN)=0033.1,         . . .
Class #  5(VSCCM)=0040.1,         . . .
                . . .
Class #  9(<NULL>)=
Order Rspots by class pairs for SRL.
Rspot: m1/2 m1/3 m1/4 m1/5 m2/3 m2/4 m2/5 m3/4 m3/5 m4/5
-------------------------------------------------------------
   85    >    >    >    <    <    >    <    >    <    <
   86    >    >    >    >    <    <    <    >    >    <
                        . . .

Rspot:  m1/2 m1/3 m1/4 m1/5 m2/3 m2/4 m2/5 m3/4 m3/5 m4/5
---------------------------------------------------------------
   85    4.3  1.1  5.5   .9   .3  1.3   .2  4.8   .8   .2
   86   10.3  1.1  1.8  1.6   .1   .2   .2  1.7  1.5   .9
                        . . .
```

Typical *.sps* Input Data File

The SPSS (*.sps*) data file format is specified in the GELLAB manual. Part of a sample *.sps* file is given below which was taken from [SpeR84]:

```
File: 000002.sps 10/14/1984, 06:48:13AM from: hm3pcg.pcg
SRL[4]=RANK SUM TEST [S2] AML VS CLL, .95 SIG, 1 GEL/PT,PERCENT NRM
Pairing labels: PSUX
Density Mode: R
Relative distance limits are[ .00, 512.00]
       . . .
MW Rgel limits are[ .00, 511.00]
Using least square normalization.
Class #  1(-AML)=0324.1, 0325.1, 0326.1, 0336.1, 0346.1,
Class #  2(-ALL)=0369.1, 0370.1,
Class #  3(-CLL)=0374.1, 0371.1, 0373.1,
Class #  4(-HCL)=0384.1, 0382.1, 0383.1, 0385.1,
       . . .
Class #  9(<NULL>)=
RSPOT# ACC# INDEX DENSR D' LABEL[0:4] LMSET[1:25] DP DL DX DY XABS YABS CLASS
  36  0382.1  30    1.36  2.20   2   2  9.4  25  -19  16  337  128  4
  36  0374.1  103   2.42  3.40   1   2  2.2  18  -12  13  313  156  3
       . . .
  36  0369.1  143  42.71  13.30  1   2  3.2  17  -10  14  304  156  2
  36  0336.1  206  62.01  25.90  2   2  5.8  16  -14   6  304  187  1
  122 0325.1  114    .81  1.80   1   8  2.0   9   3  -9  390   213  1
  122 0370.1  88    1.37  1.60   2   8  5.4  11  -5  -10  401  229  2
       . . .
```

Typical Format of a General Input Data File

A sample general data file format is given below:

```
File: BYTE.DTA 10/03/1984 from article describing cluster algorithm
SOFTWARE MSDOS RAM DISK KEYPAD FNKEYS COLMNS PORTABLE PRICE
Ace 2 0 64 140 1 0 80 0 2058
AppleIIe 0 0 64 140 0 0 80 0 1919
Atari 0 0 48 0 0 0 38 0 470
C64 0 0 64 0 0 4 40 0 293
Compaq 0 1 128 320 1 10 80 1 3010
IbmPC 0 1 64 180 1 10 80 0 2558
Kaypro 3 0 64 195 1 0 80 1 1600
Osborne 2 0 64 204 1 0 80 1 1250
Ti99 0 0 16 0 0 0 28 0 205
TrsColor 0 0 16 0 0 0 32 0 385
Trs80 0 0 64 184 1 3 80 0 1700
Vic20 0 0 5 0 0 4 22 0 193
```

HINTS

(1) If you want to run it in a batch mode and do not care about the plots on your terminal (or batch job) then use `-NOWAIT`.

(2) The current limits are:
max Classes = 20
max Features = 100
max Objects = 128.

USAGE:

```
dendrogram <data file> [<optional switches>]
```

Type `dendrogram -info` to get more information.

SWITCHES

-AbsDistance use $\sum |X_i - X_j|$ instead of by the value $(X_i - X_j)^2$ to compute distances. The distance scale is the average distance instead of standard error used with the Euclidean distance measure.

-BothMeanClassesAndRatio use both the mean class density and ratios of mean class density as features from a *.sps* file to cluster Rspot objects.

-Color draw each class associated object a different color if the display permits.

-DEbug print debugging information.

-DIsplay:<4010,VT240,XWIND,PPX,PS,PLOT,xxxxPLOT,LASER> select plot device, 4010PLOT is default.

-DRaw draw the dendrogram (default). Otherwise, just compute tree.

-File create an output file with the same name but a *.dgm* file extension for `-List`.

-Help List more information on using **dendrogram**.

-Info Display general information about this program.

-List list the unnormalized and normalized data as well as trace the building of the binary cluster tree. Then list the binary tree and distance values prior to plotting the tree.

-MeanClasses use Mean class density in *.sps* to cluster Rspots. I.e. *objects*={Rspots} and *features*={class-Means}. Otherwise, *objects*={gels} and *features*={Rspots}.

-RangeScaling scale feature data to [0:1.0] instead of autoscaling by $(X_i - \overline{X})/\sigma_X$.

-Usage Display command line format information.

-Version Print current version number of the program.

-WAit if plotting, wait for `DONE` to be typed to erase the display.

-WMwait When done, wait until do CLICK TO EXIT `mwwait` widget to exit.

EXAMPLES OF USAGE

```
dendrogram ts3s02.sps -display:plot
       # Cluster gels as a function of Rspot expression profile.
       # Generate a UGF plot file.

dendrogram ts3s02.sps -bothMeansAndRatios -display:VT240 -wait
       # Cluster gels as a function of Rspot expression profile.

dendrogram ts3t95.inq -display:4010 -wait
       # Cluster Rspots as a function of gel expression profiles.
       # Display it on a 4010 graphics terminal.

dendrogram ts3t95.inq -display:4010 | tek2psG -p | laser -laser
       # Same as above, but print plot on laser printer.
```

3.6 dwrmap - Plot RMAP from GSF, GCF or SPSS file

Program **dwrmap** plots an Rmap of all of the spots in a segmented gel as estimated spot boundaries. Their size is proportional to their density or optionally the spot's variance (σ_x, σ_y). As input, it uses as input the Gel Segmentation File (GSF) produced by the **sg2gii** gel segmentation program. Alternatively, it can read in an SPSS file and plot that instead. When the GSF file is plotted, the spot numbers are the connected component (CC#) spot numbers whereas for the SPSS file the spot numbers in the plot are Rspot numbers.

Rmap plot of GSF

It first reads in the spot list file and then begins the Rmap plot on the specified graphics output device (4010PLOT is the default). The graphics device is set with `-DISPLAY:device`. Only the top *thrDens* spots are labeled unless `-ALLSPOTSLABEL:`*minDensity* is used (in which case spots darker than *minDensity* are labeled). A list of these darkest spots is generated and output in a file with the name `r`*exxxx.drm*. If `-DISPLAY:VT240` is used, then emulate a 4010 terminal. It is put into 4010 mode when plotting and returned to VT100 mode when done when the `-VT240` option is used. If `-DISPLAY:LASER` is used, send the plot to the laser printer using **plotn** and **tek2psG** where the laser printer is specified in the `$LASERPRINTER` environment variable.

The `.gellabrc` GELLAB-II initialization file has several options used by **dwrmap** and other programs. The `#` denotes disabled options which are included in the file for convenience in changing the options.

```
option.DISPLAY: 4010
#option.DISPLAY: PLOT
option.LASERPRINTER: laser
#option.LASERPRINTER: qms
```

When the `PLOT` display option is specified, a plot file numbered *nnnnnn.ugf* is generated (e.g. `000012.gsf`). It has control information for the **plotn** program which causes the spots to be plotted in red while all labels are drawn in black (see Section 3.14, page 420). Colors will only be visible if the graphics device supports color otherwise it fill spots with black.

Figure 3.14 shows a Rmap plot of a gel using the default labeling options. It includes spots with integrated density > 3.0 OD. Figure 3.14 shows a Rmap plot of the same gel but with only those spots with an integrated density > 15.0 OD.

Figure 3.13. Sample **dwrmap** Rmap plot of gel generated using the default options (label all spots with $D' > 3.0$ OD).

Figure 3.14. Sample **dwrmap** Rmap plot of gel generated only drawing spots whose D' is darker than 15.0 OD.

USAGE:

```
dwrmap <ACC#> [<Optional switches>]
```

Type `dwrmap -info` to get more information.

SWITCHES

-AllSpotsLabel:<min density value> label all spots greater than the specified minimum value of D'.

-DArkest:<n> label the n darkest spots (default `-NODARKEST:35`).

-DEbug print debugging information on TTY and *.drm* file.

-DIsplay:<4010,VT240,XWIND,PPX,PS,PLOT,LASER,xxxxPLOT>,TTY select the device to plot on (4010PLOT is default). If TTY, then print DRM data stdout and do not plot the Rmap.

-DPrime use corrected density D' rather than D (default).

-Fill fill labeled spots (`-NOFILL` is the default).

-GCf:G2 draw Rmap of **CMPGL2** GCF file instead of GSF file. Plot both gels G1 and G2.

-GRayscale fill all spots regardless of labeling (`-NOGRAYSCALE` is the default).

-HFlip pIe reverse output image about horizontal pIe axis.

-Info print more information on dwrmap.

-Plt:<additional plot switches> `/Zoom`=nX (default is 1X), `/Nolabel` (default is label *thrDens* darkest spots), `/Size:Area` or `Density` or `SxSy`, [`SxSy` the default), `/Center:x,y` Rmap at (x,y) (Centroid of set of spots).

-RestrictPlot plot only those spots specified by either `-ALLSPOTSLABEL` or `-DARKEST` switches.

-SPss:file(or:G2) draw Rmap of CGELP2 SPSS file istead of GSF file. If G2 is specified, then plot both G1 and G2.

-STudy:n_1, n_2, n_k - Label drawing with accession entry fields $n_1, n_2, ..., n_k$ (default field 13).

-TAble draw table of darkest spots in the Rmap (default `-NOTABLE`).

-TItle:text title string to use in plot instead of (GSF, GCF or SPSS) spot list file name.

-Usage print UNIX command level switch usage.

-VErsion print the version of the program.

-VFlip MW reverse output image about MW axis.

-WAit if plotting on display, wait for `DONE` to be typed to erase the display (`-NOWAIT` is the default).

-WMwait When done, wait until do CLICK TO EXIT `mwwait` widget to exit.

EXAMPLES OF USAGE

```
dwrmap 324.1 -Display:4010PLOT -NOWAIT
      # Use Tektronix 4010 output and generate plot (default).

dwrmap 324.1
      # Same as above (above defaults).

dwrmap 324.1 -Display:VT240
      # Use Tektronix 4010 emulation in VT240 and no plot.

dwrmap 324.1 -Display:4010 -Darkest:45
      # Use Tektronix 4010 output, label darkest 45 spots in plot.

dwrmap 324.1 -Display:TTY -Darkest:45
      # Do not plot but generate and print DRM darkest 45 spots.

dwrmap 324.1 -Display:PLOT -Darkest:45 -Table
      # Just make .ugf plot file, label dark 45 spots and table.

dwrmap 324.1 -Darkest:45 -RestrictPlot
      # Only plot the darkest 45 spots.

dwrmap 324.1 -AllSpotsLabel:10.0 -RestrictPlot
      # Only plot spots > 10.0 D'.

dwrmap 324.1 -Darkest:45 -Table
      # Label darkest 45 spots and also put table into plot.

dwrmap 324.1 -Spss:ts3s02.sps -Display:4010PLOT
      # Make the Rmap from the .sps Rspot file.

dwrmap 324.1 -Display:4010 -Plt:-Size:Area -Wait
      # Plot based on area rather than Sx,Sy.

dwrmap 324.1 -Display:LASER
      # Generate plot file and print it on laser printer.
```

The *.drm* output file duplicates the ordered list of dark spots as in the following examples for gels 324.1.

EXAMPLE 1. File containing a table darkest GSF spots.

```
49% dwrmap 324.1 -disp:plot
 DISP:PLOT
DWRMAP V-1.3.46 - Version August 14, 1992
Today's date is 09/26/1992, 05:40:11PM
User:/home/joeUser/demo
Written 1982-1989, P. Lemkin.
Reading GSF: p10324.gsf
Found 792 spots. Global centroid is (244,246)
Drawing Rmap of 0324.1 at 1X at(266,291)
Spot's CW[36,497:74,509]
Can't have '000000.ugf' filename, start looking at 000001.ugf
[MAKE!UGF!FILE] using first free file: /home/joeUser/gellab/gen/000017.ugf
Plot file: /home/joeUSer/gellab/gen/000017.ugf
Real TIME =00:00:22  CPU TIME =00:00:18,  81.82%

Table darkest spots file: /home/joeUser/gellab/gen/r10324.drm
```

The following is the contents of a *.drm* file.

```
DWRMAP V-1.3.46 - Version August 14, 1992
Today's date is 09/26/1992, 05:40:11PM
User: /home/joeUser/gellab/demo
Written 1982-1992, P. Lemkin.
/HEME MALIG-AML,MYELOID
Table darkest GSF spots file: /home/joeUser/demo/gellab/gen/r10324.drm
Found 792 spots. Global centroid is (244,246)
CC#414 [353,243] D'=226.8
CC#288 [353,191] D'=128.1
CC#693 [397,406] D'=107.5
CC#673 [355,382] D'=107.2
CC#546 [330,296] D'=102.5
CC#679 [406,387] D'=98.2
CC#696 [361,408] D'=89.4
      . . .
CC#528 [207,286] D'=39.1
CC#158 [308,142] D'=38.5
CC#579 [264,314] D'=37.4
CC#763 [307,460] D'=37.2
CC#390 [324,227] D'=36.2
CC#108 [88,119]  D'=35.7
CC#405 [276,234] D'=35.7
Max density=262.7 max area=240 max sxsy=4.9
Drawing Rmap of 0324.1 at 1X at(256,256)
Spot's CW[0,512:0,512]
```

3.7 getacc - Gel image data acquisition & accessioning

The **getacc**[14] program provides a mouse and menu based interface that allows the user to aquire 2D gel images and specify experimental and *additional* information for each gel. Additional (non-experimental) information refers to information that is required for further GELLAB processing (e.g. name of the gel image file, grayscale to OD calibration, image computing window of interest). All the information associated with each gel is known as *accession* information. The accession information is stored in a file (with an *.id* extension) called an accession file database. Image data comes from either prescanned image files or the DataCopy CCD 612F scanner. Images are found (or saved) in the GELLAB system's `ppnp1x` picture disk area for the specific user project.

accessioning

Upon completion of an image data acquisition session, a number of UNIX batch job scripts may be generated by the **makjob** program for processing the set of gels in the project. These include:

1. An interactive UNIX script to interactively *landmark* the set of gels just accessioned.
2. A UNIX background batch script to *segment* gels into gel segmentation files (GSF) and then perform *spot-pairing* of these files into gel comparison files (GCF).
3. A UNIX background batch script to *build* the initial PCG database from the set of GCF files, attempt to normalize it, and perform some initial statistical tests.

script generation

The first interactive script, upon completion, will automatically submit the second batch script which in turn will start the third script upon its completion. After the third script is finished, interactive exploratory data analysis can begin. Additional batch scripts are generated for other methods of processing gels in GELLAB.

If a neutral density (ND) wedge (available from Kodak and other sources) is scanned with the autoradiograph, it is then possible to calibrate image pixel gray values in terms of optical density. This is modeled using a piecewise linear function of gray value to optical density (OD). **getacc** can be used to compute this gray scale to OD calibration function using the scanned calibrated ND wedge. The operator

OD calibration

[14] There are two version of the program for X11. **getacc11** and the newer **getacc11a**. **getacc11** uses **ppxcvt** to convert images to 8-bit PPX files while **getacc11a** allows you to accession 12 or 16 bit laser scanner gel scan files. **getacc11a** is able to automatically read and figure out the types of original image files. It replaces the use of **ppxcvt** since it can downscale the original image into an `a0`*nnnn*`.ppx` file as well as make the standard `s0`*nnnn*`.ppx` image file.

will be requested to enclose the wedge with a rectangular window. In addition, a rectangular *computing window* (CW) is required for each to delimit the region in the gel where the valid spots occur. This gel specific grayscale-OD calibration and CW values are also saved in the accession file for each gel.

Flow chart of getacc operation:

What you need to do

Even though gels can be acquired two different ways, by scanning or converting gel image files scanned on other scanners, the accessioning process is effectively the same. The difference being – where the gel image file comes from. The following flow chart shows the steps required in accessioning. We do not show all of the combinations of different steps we could take, but rather the overall process which needs to be performed.

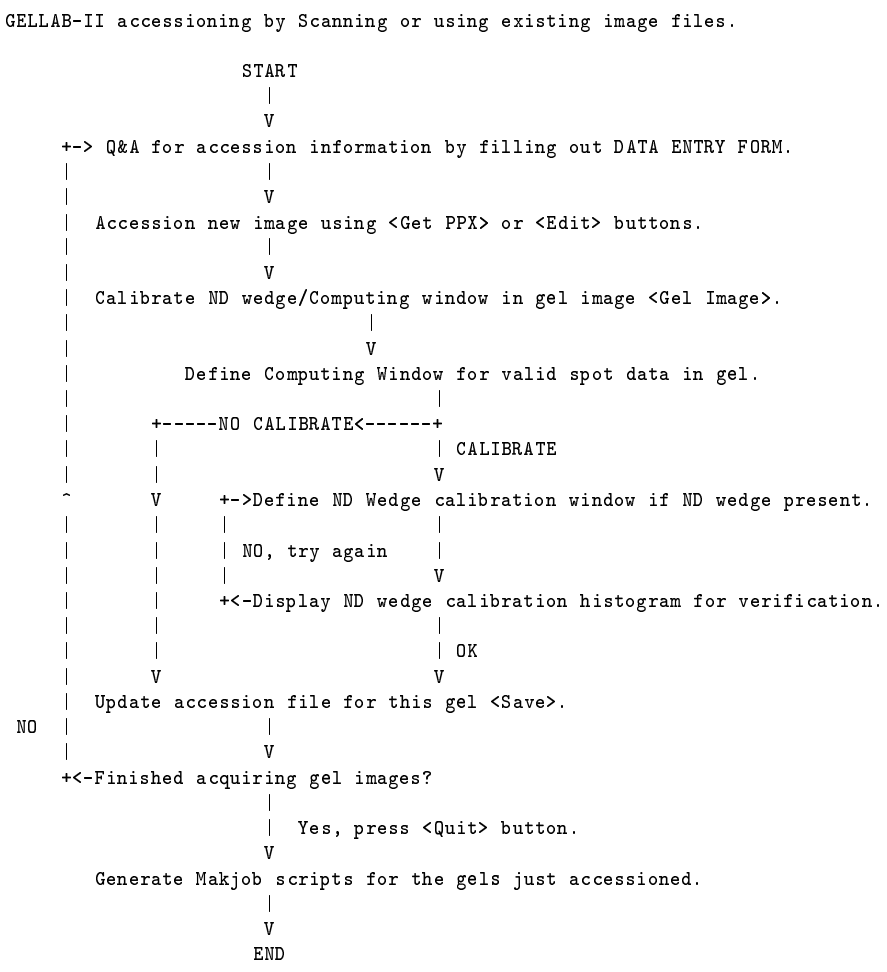


Figure 3.15. Accession gel images from either a scanner or from existing scanned image filesinto GELLAB using **getacc**. The user specifies the new gel accession number for the gel corresponding to the image file to be used. The user answers questions reguarding auxiliary accession information for the gel's experimental conditions. When images are displayed, the user interacts with the system using the mouse to define a gel computing window and possibly ND wedge calibration. All of this information is used by the other programs in GELLAB. When finished, you can generate batch scripts for doing the additional processing required to build a composite gel database.

Running getacc

The **getacc** program has three base windows organized vertically. The top window, or *info window*, displays information about items in the second window.

Controlling interaction

The second window, or "control window", is a row of push buttons and pull-down menu buttons. As you move the mouse over the buttons in the control window, information about each button is displayed in the information window. The third window, or "main form", displays a column of data entry fields where you enter data about each gel.

The Control Window

buttons

The control window is made up of a number of buttons. The buttons' labels and types are listed here in the order they appear (left to right). In the following discussion, buttons will be shown enclosed in boxes.

Change Names	change the *names* of the fields in the accession database.
Change Values	change the default *values* in the template.
Get PPX File	get the image in PPX file format (pull down menu).
Gel Image	popup the gel image to define the computing window and ND wedge.
Save	save the current information in the accession database.
Edit	edit an existing entry from the accession database.
Help	popup a window with more detailed help/information.
Bindings	popup a window describing the keyboard accelerator actions.
Quit	quit the application.

The order of these buttons is designed such that the you will generally interact with them in a left to right order, although you are not required to. When you first start your data aquisition session, you can change the names (labels) of the gel information fields using `Change Names`. Next, you can specify default values for each gel information field using `Change Values`.

The main form and the next three buttons are then used in an iterative manner. You repeatedly enter gel specific information in the main form, get the image file for a gel with `Get PPX file`, enter more gel specific information (that requires displaying the PPX file) using `Gel Image` and finally save the information in your accession file using `Save`. These and the remaining buttons are discussed in detail below.

In the following discussion, *clicking* a button means moving the mouse cursor over a (screen) button and pressing the left mouse button.

`Change Names` - (push button)

default accession field Names

Clicking this button pops up a window with a data entry form. The field values in the popup form are the accession file field *labels* of the form window in the main window. You can edit any of these values by moving the mouse cursor over a popup form value and editing the text (the keyboard edit keys are described below). However, there are two exceptions; the field labels `ACCESSION #` and `FILE #` are special and you're not allowed to change them since that would corrupt the accession database. Normally, you only change the field names at the start of a project and only if the default values are not adequate. The default values are probably adequate for many projects.

When you are finished, click the `Done` button at the bottom of the popup to make the new field values in the popup form become the new field labels in the main form. (By convention, any lower case letters you've entered are converted to upper case at this point). This also updates the field names in the accession file, so be sure these are the values you want. If you change your mind and don't want to make any changes, click the `Cancel` button. Clicking the `Help` button will pop up a window with some help information.

`Change Values` - (push button)

default accession Values

Before you actually start editing field values in the main form, you can define default values. Clicking the `Change Values` button will popup a window with a form. The default field values are displayed in the form; initially they are all empty. You can edit any of these values by moving the mouse cursor over a popup form value and editing the text (the keyboard edit keys are described below).

When you are finished, click the `Done` button at the bottom of the popup to make the new field values in the popup form become the default field values in the main form. Now, whenever it's time to start entering new gel information (usually after you've `saved` previous gel info) these default values will appear as the field values in the main form. As above, there are two exceptions; the fields labeled `ACCESSION #` and `FILE #` are calculated for you each time, so their defaults are ignored.

If you change your mind and don't want to make any changes, click the `Cancel` button. Clicking the `Help` button will pop up a window with some help information.

Editing The Main Data Entry Form

entering your accession info

If you've defined default values using the `Change Values` button, these values will appear in the main form. At this point an accession number and file name are calculated for you and also displayed in the main form. You can edit any or all of these values by moving the mouse cursor over a value and editing the text (the keyboard edit keys are described below). If you have not defined any defaults, you will have to enter all the values yourself, however, empty fields are allowed.

`Get PPX File` - (pull down menu button)

adding image to database

Once you've edited all the values in the main form (or at the very least the one labeled `FILE #`) you can get the PPX file using the `Get PPX File` button. Clicking on this button will *pull down* a menu from which you select the appropriate way to get the PPX file. The choices and their actions are grouped into commands which get the gel image by scanning it, and those which convert an existing gel image file.

```
Convert CSPI          - use PPXCVT to convert a CSPI file
Convert Mol Dynamics  - use PPXCVT to convert a Molecular Dynamics file
Convert BioImage      - use PPXCVT to convert a BioImage file
Convert Fuji          - use PPXCVT to convert a Fuji Phosphoimager file
Convert Elsie         - use PPXCVT to convert an Elsie file)
Convert Generic TIFF  - use PPXCVT to convert a TIFF file)
------------
CSPI Scan             - not implemented yet
DataCopy Scan         - use CAMERA program to scan the image
Mol Dyn Scan          - not implemented yet
VideoPix Scan         - use SUN's vfctool to grab TV image to file.
```

DataCopy Scan

using a camera

Under normal gel data acquisition where we are scanning images with GELLAB, the **camera** program is used by **getacc** to do the actual image scanning . This is done automatically by **getacc** transparently to the user via a UNIX `rsh` to the machine which actually has the camera attached. This means that if there is a network of several workstations running GELLAB, only one need have the camera attached to it so that the others can access it.[15] That particular machine then is effectively a "camera server". Note that the **camera** program must be executed on the computer which has the camera attached whereas the **getacc** program need not.

At NCI/FCRDC we use the Datacopy model 612F autofocus camera (Datacopy Corp., Mountain View, CA). This camera is connected via their General Purpose

[15] The particular computer on the network which acts as a camera server must be specified at the time GELLAB is recompiled.

Imaging Interface (GPII) interface to the SCSI port on a SUN3/50. Gels are placed on a Gordon Light Box model TVS (Gordon Instruments, Orchard Park, NY). This light box has adjustable edges to block out light from the edges of the gel which would normally contribute to "flair" image noise. It also has an adjustable stabilized power supply drives incandescent lights which minimize 60Hz noise the CCD camera would pick up if a fluorescent light source such as the Aristo light box were used. The saturation of the camera can be controlled by adjusting either the light source intensity or the *f*-stop (in the latter case you must refocus).

Datacopy camera

CSPI Scan

This scans an image directly from the CSPI scanner. `not implemented yet` *TODO*

Molecular Dynamics Scan

This scans an image directly from the Molecular Dynamics scanner. `not implemented yet`. *TODO*

VideoPix Scan

This scans an image directly from the SUN VideoPix frame grabber camera. Note: you must include a ND wedge in the image in order to calibrate the grayscale image in terms of OD. The NTCS VideoPix has a resolution of 640x480x7-bits. It uses the the SUN *vfctool* to scan the image. You must use the same file name to specify to same the image as you have specified in the accession entry PPX FILE field. The following steps should be followed when useing the VideoPix *vcftool(1)* program to grab a TIFF image. Remember to:

VideoPix

1. Select `B&W` mode.
2. Press `Preview` to continuously display the image while you focus, adjust the *f*-stop (to avoid saturation), and adjust the gel position in front of the camera.
3. Then, grab the image using the `Grab` button.
4. Press the `File` menu button and select `Save as...`.
5. In, the `File Save` option,(where *xxxx* is the current file name number):
   - select 'TIFF' file type,
   - select '8-Bit Grayscale' image format,
   - set Directory to: <*project*>`/gellab/ppx`,
   - set File to: `a0`*xxxx*`.tiff`.
6. Then, kill the *vcftool()* program using twm by selecting the `Destroy Window` option and clicking on the VideoPix image.

Note: getacc will invoke: `vp.do` *ppnpix* when the GET PPX menu `VideoPix Scan` is selected. Note that *ppnp1x* is the FULL path gotton from the `gel.rc` file. It also pops up the above message to tell them what to do. After the scan is completed, it will then call **ppxcvt** to convert the tiff file to a PPX file.

```
#!/bin/csh -f
# File: vp.do - run the VideoPix camera recursively on the system with the camera.

# Set p to the path where vpcamera.do lives
set p = $GELLABMANAGER/gellab/bin

setenv VP_HOST bigsun

rsh VP_HOST $p/vpcamera.do $argv[1]
```

This in turn calls camera on the computer where it actually resides.

```
#!/bin/csh -f
# File: vpcamera.do - run the VideoPix camera on bigsun in the specified directory.

cd  $argv[1]
pwd

# Set p to the path for your VideoPix board
set p = /home/bigsun/local/bin/vfc1.0

setenv LD_LIBRARY_PATH /usr/openwin/lib
$p/vfctool -p 1
```

Convert CSPI TIFF image file

TODO

It does not handle CSPI TIFF image OD calibrations at this time. Use the Generic TIFF files mode for now. Set the desired image pixel depth in the **getacc** preferences pull-down menu.

Convert BioImage `gel` image file

The BioImage 2D gel system is able to scan 1Kx1K images. These have the OD calibration in the header of the image file which is extracted and used with GELLAB. The gel files of a BioImage project are directories rather than image files. In each directory there is a `gel` image file. *That* is the file which is converted.

Convert Molecular Dynamics TIFF file

Currently, it handles the MD300 densitometer data which has a 12-bit linear dynamic range over 0 to 4.095 OD. These are currently mapped to 8-bits using several options. The MD400 phosphoimager options is not fully implemented at this time.

TODO

Convert Fuji TIFF file

It does not handle Fuji TIFF image files at this time.

Convert Elsie image `data.ave` file

The Elsie 2D gel system is able to scan roughly 1Kx1K to 2Kx2K images. The image pixel data is in file `data.ave`. These have the OD calibration in an associated `geldata` file. This is extracted and used with GELLAB. The gel files of a Elsie project are directories rather than image files. In each directory there is a `data.ave` image file, `geldata` image description file, and a `stdlines` grayscale to OD/CPM calibration file. The `data.ave` file is the one specified to be converted.

Convert Generic TIFF file

This converts a generic TIFF file. It is assumed that an ND wedge has been scanned with the image in order to obtain the OD calibration. The sampling size is 1:1 (for now).

Instead of using the GELLAB-II CCD camera, images can be obtained from other sources (such as the Molecular Dynamics scanner). Selecting one of these choices from the pulldown menu pops up a form that allows you to specify the input file that is then converted to a GELLAB PPX file by the **ppxcvt** program.

don't need to calibrate

In those cases where the gray scale is a linear function of optical density (such as the Molecular Dynamics scanner), **getacc** computes the equivalent ND wedge calibration. The grayscale OD calibration is read from the PPX file header where it was put by the **ppxcvt** program when the file was converted.

`Gel Image` - (push button)

Once you have acquired the PPX file, you can view it in a popup window and enter additional gel information that requires the image (e.g. the grayscale to optical density calibration and image computing window). This additional information is required before further GELLAB-II processing can proceed on your gel images. Note that depending on the source of your scanned gel image, you may not be required to explicitly calibrate OD since the information may be contained in the gel image file. In that case, it is automatically transfered to **getacc** when it reads the gel image.

calibrating the image

You start this process this by clicking `Gel Image`. This pops up a window with an information line and a row of buttons at the top (similar to the main window). The buttons are: `Done`, `Restore CW` and `Calib ND`. Below that are three status message lines and the image.

region of interest

As in all image windows in GELLAB, pressing the middle mouse button and dragging (moving the mouse while the mouse button is still depressed) will change

the contrast in the image. Dragging the left mouse button and the right mouse button will change the ND Wedge area and the Computing Window area, respectively. These two areas are shown in the image as rectangles.

OD wedge calibration

Once you have defined the ND Wedge area (by dragging the left mouse button in the image and releasing it) you can define a window over the ND wedge steps. Make sure you cover the lightest to the darkest steps. Also avoid going right up to the edge of the wedge - it will give you a better estimate.

computing windows

You also need to define the *computing window* in the gel image. This is the region of interest where there is actual spot data. This lets you avoid quantitating regions of the gel with writing, etc. in it. You define the computing window area (by dragging the right mouse button in the image and releasing it).

Figure 3.16. ND computing window. It is defined interactively over the neutral density wedge to be calibrated.

Figure 3.17. ND wedge calibration histogram. It is used to verify the computer estimation of the grayscale to OD calibration values which correspond to the peaks of the histogram. These peak values are used in other GELLAB-II software to map grayscale to OD.

Figure 3.18. Computing Window. It defines the region of the gel with valid spot data.

3.7.1 Setup procedure for scanning gels with light box/Datacopy camera

DataCopy setup

1. Make sure that the Datacopy 612F autofocus CCD camera and its GPII interface as well as the light box are turned *on*. Take the lens cap off of the camera. Let the camera and lightbox warm up for at least 15 minutes before you do any data acquisition. When finished acquiring data, shut them *both* off and put the lens cap back on the camera. Note that the light box gets rather warm, so do not leave wet gels or the ND calibration wedge on it longer than is necessary.

2. A dark cardboard or glass mask is mounted on top of the Gordon light box and adjusted so that it blocks out light on the outer edges of the gel to minimize scattered light to the camera. The Gordon light box has a light instensity control which varies over a scale between 0 (off) and 10 (full).

3. Various distances and lenses are involved for different materials.

   Datacopy-612F/GordonLight box setup

   a. Autoradiographs 197 microns/pixel scanned - 88.6 cm, 55mm Datacopy lens at f4 (lightbox=7). For 512x512 pixels, image view is about 10.1cmx10.1cm, and for 1024x1024 it is about 20.1cmx20.1cm. The 512x512 ''standard'' image has 394 microns/pixel resolution.

   b. Autoradiographs 185 microns/pixel scanned - 83 cm, 55mm Datacopy lens at f4 (lightbox=7). For 512x512 pixels, image view is about 9.5cmx9.5cm, and for 1024x1024 it is about 18.9cmx18.9cm. The 512x512 ''standard'' image has 370 microns/pixel resolution.

   c. Autoradiographs 169 microns/pixel scanned - 75.5 cm, 55mm Datacopy lens at f4 (lightbox=7)[standard]. For 512x512 pixels, image view is about 8.6cmx8.6cm, and for 1024x1024 it is about 17.3cmx17.3cm. The 512x512 ''standard'' image has 338 microns/pixel resolution.

   d. Autoradiographs 122 microns/pixel scanned - 56.6 cm, 55mm Datacopy lens at f4 (lightbox=7). For 512x512 pixels, image view is about 6.2cmx6.2cm, and for 1024x1024 it is about 12.4cmx12.4cm. The 512x512 ''standard'' image has 244 microns/pixel resolution.

   e. Autoradiographs 78 microns/pixel scanned - 37.7 cm, 55mm Datacopy lens at f4 (lightbox=7). For 512x512 pixels, image view is about 4.0cmx4.0cm, and for 1024x1024 it is about 8.0cmx8.0cm. The 512x512 ''standard'' image has 156 microns/pixel resolution.

4. To focus the camera, mount the gel on the mask, with masking tape, if needed, and set up the camera distance and lens *f*-stop. The following directions are for the Datacopy 612F autofocus camera. Note that the term "autofocus" is somewhat of a misnomer since you must actually do the focusing, but it will project a white grid through the camera lens onto the subject material being scanned. Turn *off* the light box if *on*, and then turn *on* the autofocus light (switch located on the front of the camera), focus the projected grid on the gel (which is on the lightbox) until the two sets of bands in the center of the image line up [i.e. rotate the lens focus adjustment], turn *off* the autofocus light and turn *on* the lightbox. If the autofocus light overheats, it will shut itself off. Wait a minute or two and it will reset itself automatically.

5. Gels are mounted *upside down* on the light box so that (facing the gel mounted on the lightbox) the acid end of the gel is to the left and the high MW end is toward you. If necessary, the gel is aligned on the mask such that there are no spots of interest on the right side of the light box to the right of the low OD value end of the ND wedge rectangle.

6. The CCD camera system white level needs to be adjusted so that there is maximum dynamic range for black. This can be set approximately by adjusting the camera *f*-stop and/or light source intensity. The values for *f*-stops in the above table are only suggestions since the values required will depend on the background density of the gels and the light level of the scanner. This is done either by adjusting the *f*-stop or light source intensity by rotating the Gordon light box "light intensity" control while running program **camera -scanline** on the Sun3/50 connected to the camera. This will continuously rescan the line, printing the minimum and maximum gray scale values seen in the last scan. Type <CR> to terminate rescanning. Insert the ND step wedge horizontally in the center of the light box so that the line being scanned is along the step wedge. Then maximize (without saturating) the range of the steps. If you adjust dynamic range by changing the *f*-stop, then you should recheck the focus. When done, type `control/C` to exit the program.

7. At this point remove the step wedge and put the gel to be scanned (along with the step wedge on the mask) back onto the light box. You are now ready to scan gels.

USAGE:

```
getacc [<Opt. -switches>]
```

SWITCHES

-Info print more information about **getacc**.

-Project:*prj* overide the 3 character alphanumeric project prefileprefix (the default comes from `gel.rc`). Newly created accession numbers are saved in the file *prj*`.ccl`. *prj* is also passed to **makjob**

-Usage print UNIX command level switch usage.

-Version print the version of the program.

-WmWait When done, wait until do CLICK TO EXIT `mwwait` widget to exit.

3.8 landmark - Landmark generation between gels

auto landmarking

Landmark is used to load two gels [Rgel,G2] onto the X-windows display for interactively define a set of corresponding landmark spots. Using the `-AUTOMODE:` *G3* option will cause it to draw each (Rgel,G3) LM in the left Rgel (i.e G1) image to serve as prompts for the the user to mark the corresponding LM in the right or G2 image.

Getting help

After marking the spot in the right G2 image, it will go on and draw the next landmark in the left Rgel image until there are no more landmarks to draw. Interaction is done by pressing mouse buttons. Press the `BINDINGS` command button to pops up a list of mouse actions. Pressing `HELP` button pops up a scrollable description of how the program works.

manual landmarking

If `-NOAUTOMODE` is used (default), then the user must define each landmark themselves in the left Rgel image. This can be done without the switch by selecting the `ADD LANDMARK` command button option. In either case, you may erase the last landmark spot marked in the right G2 image. It will undraw it from the images and let you try again.

deleting a landmark

You may not delete a landmark from the Rgel image which was defined prior to this session. This preserves the integrity of the landmark database. In the future, the `REMOVE LANDMARK` command will let you do this, but it is not implemented at this time.

updating landmark DB

The two gels to be landmarked are specified by accession numbers from the current accession file. When finished landmarking, the landmark data base file will be updated with this data. Once started, the user interacts using the mouse.

What you have to do

1 Adjust the *contrast* of the two gel images. This is done by moving the mouse around one of the images while at the same time depressing the middle mouse button.

2 Select the small zoom windows' magnification levels by choosing an entry from the `ZOOM` button. Alternatively, you can cycle through magnification levels by typing `z` in one of the image windows.

3 If you are not using `-AUTOMODE` or you want to add landmarks to an existing LMS database, you must enter *add landmark* mode to add landmarks to the Rgel. To do this, press the `ADD LANDMARK` button.

1. Select new spots in the Rgel by positioning the cursor in the left image and pressing the *left* mouse button. The new spots will have the color green.

2. If you decide you want to delete a newly added spot, position the cursor over that spot and press the *right* button. You can only delete spots that you have added during the session. **Landmark** will not let you delete spots from the Rgel that are already in the database.

4 Then, enter *landmark* mode to enter/edit landmarks in G2 by pressing the `LANDMARK` button. This will automatically disable *add landmark* mode. (Note, in *landmark* mode the zoom windows do not track the cursor through the image windows as they do in the other modes. Tracking is where a zoom window displays whatever is under the cursor as the cursor moves through the image.

1. The program automatically chooses a spot from the unpaired Rgel (green) spots for you to landmark. You can override this by positioning the cursor near a different unpaired Rgel (green) spot and pressing the `left` mouse button. This spot's color will change to cyan to indicate that it is the spot currently being paired and the previous Rgel spot is turned back to green. The spot currently being paired will also be centered in the left small zoom window.
2. Enable flickering by typing `f` to toggle flickering on. This enables tracking in the right G2 zoom window as you try to find the best spot in G2 corresponding to the cyan spot in the Rgel.
3. Position the cursor at the desired position in G2 corresponding to the cyan spot in the Rgel and press the *left* mouse button to add a cyan colored spot to G2. If you make a mistake, you can remove the G2 landmark by pressing the *right* mouse button near the cyan G2 spot.
4. Then, *mark* the landmark by turning off flickering (by typing `f` again) at which point the two cyan landmarks turn red.
5. Alternatively, you can landmark without flickering in *landmark* mode by positioning the cursor in G2 and pressing the *left* mouse button. This changes the cyan landmark in the Rgel to red immediately and puts a red landmark at the cursor position in G2.

5 When you are done landmarking all of the spots you want, press `FINISHED`. The landmark data you just defined is used to update the landmark database file defined in the gel.rc state file.

Warning

Pressing `QUIT` could cause you to lose any landmark information you have just generated. Thus, it asks you 'Are you sure?' before exiting to give you another chance. Normally, you press the `FINISHED` button to save the LMS DB and exit the program. *Warnings*

Flickering without landmarking

flickering gels

You can use **landmark** to flicker the two gels without landmarking.

1. Make sure you are not in `LANDMARK` mode or `ADD LANDMARK` mode.
2. Position the cursor over a spot in the Rgel where you want to flicker and press the `f` key to turn flickering on. This disables the left zoom window from tracking the cursor in the Rgel, effectively locking the Rgel position in the left zoom window.
3. Move the cursor around in G2. Flickering occurs in the right zoom window.

Editing existing landmark sets

Using the `LANDMARK` button, lets you edit a previously created landmark set if, for example, a landmark was incorrectly entered and you recognized this at a later time.

editing

Defining landmarks

The procedure for interactively defining the landmarks involves using the mouse to first select the landmarking using the `LANDMARK` button and then to operate within a mode selected from that menu.

Landmark spot positions and GSF spot coodinates

spot positioning

If the `-GSF` option is specified (or you press `SPOT POSITION`, `SHOW SPOTS PLUS` or `SHOW SPOTS DOTS`, then the two gel spot list files are read and the spot positions saved for each spot in the image. Then you have the option of indicating all spots' positions (the `SHOW SPOT` buttons). If the `SPOT POSITION` button is active, then landmarking near a spot will automatically move the cursor to the nearest spot if it is close enough and use that position. Otherwise the default *exact* mode is used to put the landmark exactly where you set it.

Adding additional landmarks to an existing LMS DB

adding landmarks

You may add additional landmarks to an existing LMS DB entry. After the landmark program starts, press the `ADD LANDMARK` button and then add landmarks in the left Rgel image by pressing the left button over the new landmarks. When you are do, press the `LANDMARK` button to landmark the corresponding spots in the right G2 image as before.

Landmark keyboard bindings

keyboard accelerators

The keyboard and mouse bindings are invoked by pressing the `BINDINGS` menu button. An accelerator key is simply an alternative way of invoking the same function as would be done by pressing another menu button.

```
ACCELERATOR KEYS (for all modes)
l - Toggle LANDMARK mode to mark Gel2 landmarks.
n - Toggle ADD LANDMARK mode to add new landmarks to Rgel.
z - Cycle through zoom factors (1X : 8X).
f - Toggle FLICKER mode:
  a - Decrease left flicker rate,
  s - Decrease right flicker rate,
  q - Increase left flicker rate,
  w - Increase right flicker rate.

DEFAULT MODE
Mouse motion (no buttons)    - magnify image in zoom window.
Mouse motion (middle button) - change contrast.

ADD LANDMARK MODE
Left button (in Rgel)        - add a new landmark spot.
Right button (in Rgel)       - delete an added landmark spot.
Mouse motion (no buttons)    - magnify image in zoom window.
Mouse motion (middle button) - change contrast.

LANDMARK MODE
Left button (in Rgel)        - select spot for pairing.
Left button (in Gel2)        - select LM for selected Rgel spot.
Right button (in Gel2)       - delete selected spot.
Mouse motion (middle button) - change contrast.

REMOVE LANDMARK MODE
  (Not Implemented Yet)
```

USAGE: Any information which is needed and is not supplied through the command line switches will be prompted for from the operator. Any case independent switch may be negated by preceeding it with a `no` eg. `-noinfo`. The UNIX syntax to invoke it is:

```
landmark <acc# Rgel> <acc# gel2> [<Opt. -switches>]
```

Type `landmark -info` to get more information.

SWITCHES

-Automode:gelACC#Gj get the landmarks from a *previous* gel landmark session [Rgel,Gj].

-CCl:*prjPrefix* process a set of gels defined by the CCL file which has a 3 character *prjPrefix*. Eg. `-CCL:ts3` implies `ts3.ccl`. [NOTE: not implemented yet.]

-CommutativeLMS if doing `-Automode` then look for either [Rgel,Gj] or [Gj,Rgel] and make the Rgel in the LM file be 'Rgel' for [Rgel,Gj] and 'Gj' for [Gj,Rgel].

-DEviation only of existing LM DB entry is computed for the two gels and printed. No landmarking is done.

-DIsplayG3insteadOfRgel Use G3 image for display instead of Rgel image. This switch requires the use of `-Automode:` *G3* so G3 can be specified.

-Gsf use *.gsf* file spot information to find landmarks.

-Info print more information.

-UPdateLMS update the landmark database file when finished (default). The `-NOUpdateLMS` can be used with `-AUTOMODE`.

-USage print UNIX command level switch usage.

-Version print the RCS version of the program.

-WmWait When done, wait until do CLICK TO EXIT `mwwait` widget to exit.

EXAMPLES OF USAGE

```
landmark 324.1 369.1
           # Define initial landmark set for [324.1,369.1].

landmark 324.1 369.1
           # Same as above, but position the two small zoom
           # windows to the right so they do not obscure the
           # terminal window.

landmark 324.1 384.1 -Automode:369.1
           # Landmark using previous LM DB entry for [324.1,369.1].

landmark 324.1 384.1 -Automode:369.1
           # Short form.

landmark 324.1 384.1 -Automode:369.1 -DisplayG3insteadOfRgel
           # Same as above, but display 369.1 in place of the Rgel.

landmark 369.1 384.1 -Automode:324.1 -Commutative
           # Landmark using previous LM DB entry for [324.1,369.1].
```

LMS DB format

An example of a LM database file is given below and the format discussed in Section 1.6.7, page 65.

```
/ LMS: PROTOCOL-VER# 9-28-88 [rows,cols]=[1024,1024]
/ INTO /home/joeUser/gellab/lms/lms.lm FROM GELS: 8006.1,8009.1
/ GSF: p18006.gsf,p18009.gsf
CREATION-DATE: 05/28/1992, 04:07:47PM
 LANDMARK #A  G1[176, 100], G2[156, 106]
 LANDMARK #B  G1[182, 202], G2[166, 220]
```

```
 LANDMARK #C  G1[608, 116], G2[606, 126]
    . . .
 LANDMARK #W  G1[882, 726], G2[884, 772]
 LANDMARK #X  G1[222, 862], G2[194, 916]
 LANDMARK #Y  G1[618, 878], G2[626, 938]
 LANDMARK #Z  G1[714, 460], G2[718, 482]
 LANDMARK #a  G1[640, 444], G2[642, 468]
 LANDMARK #b  G1[406, 308], G2[394, 340]
 LANDMARK #c  G1[400, 788], G2[390, 814]
    . . .
 LANDMARK #w  G1[636, 758], G2[654, 806]
 LANDMARK #x  G1[788, 544], G2[792, 574]
 LANDMARK #y  G1[726, 338], G2[732, 350]
 LANDMARK #z  G1[506, 546], G2[508, 578]

ELAPSED TIME:       455. SECONDS


/ LMS: PROTOCOL-VER# 9-28-88 [rows,cols]=[1024,1024]
/ INTO /home/joeUser/gellab/lms/lms.lm FROM GELS: 8006.1,8011.1
/ GSF: p18006.gsf,p18011.gsf
CREATION-DATE: 05/28/1992, 04:16:09PM
 LANDMARK #A  G1[176, 100], G2[126, 118]
 LANDMARK #B  G1[182, 202], G2[160, 236]
 LANDMARK #C  G1[608, 116], G2[606, 136]
    . . .
 LANDMARK #W  G1[882, 726], G2[894, 724]
 LANDMARK #X  G1[222, 862], G2[196, 856]
 LANDMARK #Y  G1[618, 878], G2[634, 872]
 LANDMARK #Z  G1[714, 460], G2[728, 492]
 LANDMARK #a  G1[640, 444], G2[652, 484]
 LANDMARK #b  G1[406, 308], G2[394, 354]
 LANDMARK #c  G1[400, 788], G2[398, 810]
    . . .
 LANDMARK #w  G1[636, 758], G2[666, 780]
 LANDMARK #x  G1[788, 544], G2[818, 568]
 LANDMARK #y  G1[726, 338], G2[746, 356]
 LANDMARK #z  G1[506, 546], G2[516, 568]

ELAPSED TIME:       496. SECONDS
    . . .
```

3.8.1 Use of landmark program to estimate gel deviation

The **landmark** program may also be used to compute the root-mean square deviation of an existing LM entry in any gel with respect to the same LM in the Rgel. First the centroids of the two gels are computed. Then each spot of a list of landmarks in a LM DB entry is offset relative to these centroids. Finally, the root mean square (RMS) error is computed of the differences between corresponding landmarks when the two gels are aligned by the centroids of each of their respective

global gel distortion

sets of landmarks. The following illustrates this option.

Example 1. compute the RMS error between two gels. Do *not* do landmarking.

```
25% landmark 269.1 270.2 -deviationOnly

LANDMARK: Version June 10, 1992
Today's date is 06/12/1992, 10:07:35AM
User: /home/joeUser
Written (C) 1982-1992, P. Lemkin.
0270.2/P388D1 72HR FIBER STUDY/-/-/8-31-81/#A139/FISCHER'S/3:10, 10%/
72 HRS/C14/8 HRS/2 WEEKS/AMOSITE,TOXIC, BOTTLE#5/
L00775/-NONE-/--NONE--/VIDICON-AUTO,28MM F8,69CM/LIPKIN&LEMKIN*

0269.1/P388D1 72HR FIBER STUDY/-/-/8-31-81/#A139/FISCHER'S/3:10, 10%/
72 HRS/C14/8 HRS/1 WEEK/CONTROL,NON-TOX BOTTLE#4/
L00715/-NONE-/--NONE--/VIDICON-AUTO,28MM F8,69CM/LIPKIN&LEMKIN*

The (Representative) Rgel is: 0269.1 with 21 landmarks.
For [0269.1,0270.2]
 LANDMARK #A  G1[307, 114], G2[286, 127]
 LANDMARK #B  G1[348, 105], G2[329, 118]
 LANDMARK #C  G1[348,  67], G2[329,  81]
 LANDMARK #D  G1[260,  69], G2[241,  82]
 LANDMARK #E  G1[207,  88], G2[189, 101]
 LANDMARK #F  G1[132, 119], G2[115, 130]
 LANDMARK #G  G1[151, 199], G2[133, 210]
 LANDMARK #H  G1[229, 291], G2[208, 301]
 LANDMARK #I  G1[191, 219], G2[172, 229]
 LANDMARK #J  G1[225, 197], G2[208, 208]
 LANDMARK #K  G1[282, 178], G2[262, 189]
 LANDMARK #L  G1[343, 221], G2[323, 234]
 LANDMARK #M  G1[416, 183], G2[392, 198]
 LANDMARK #N  G1[395, 138], G2[375, 153]
 LANDMARK #O  G1[410, 244], G2[385, 260]
 LANDMARK #P  G1[406,  97], G2[385, 112]
 LANDMARK #Q  G1[ 97, 118], G2[ 80, 130]
 LANDMARK #R  G1[171, 166], G2[154, 176]
 LANDMARK #S  G1[134,  85], G2[116,  96]
 LANDMARK #T  G1[347, 300], G2[326, 314]
 LANDMARK #U  G1[428, 284], G2[398, 302]

 LMS Centroid deviation from Rgel= 17.1
```

3.9 makjob - Generate GELLAB-II batch processing scripts

The **makjob** program is a *program generator* used to define a set of batch jobs (UNIX scripts) to process a set of gels. These scripts will include all phases of processing including landmarking, spot segmentation, gel pairing, composite gel PCG DB construction and initial statistical tests of the PCG DB. Several alternate scripts are also generated to draw Rmap plots of the GSF files to be used with `-lmsedit` option with the **landmark** program which is alternate method of generating landmark set data for the LM DB. However, the regular scripts will work adequatedly on most systems. The **getacc** program also invokes the same procedures to generate scripts as **makjob** and therefore is documented here for that part of **getacc**.

batch scripts

The **makjob** program is a *program generator* used to define a set of batch jobs (UNIX scripts) to process a set of gels. These scripts will include all phases of processing including landmarking, spot segmentation, gel pairing, composite gel PCG DB construction and initial statistical tests of the PCG DB. Several alternate scripts are also generated to draw Rmap plots of the GSF files to be used with `-lmsedit` option with the **landmark** program which is alternate method of generating landmark set data for the LM DB. However, the regular scripts will work adequatedly on most systems. The **getacc** program also invokes the same procedures to generate scripts as **makjob** and therefore is documented here for that part of **getacc**.

batch scripts

The `.gellabrc` GELLAB-II initialization file has several options used by **makjob** and **getacc**. The `#` denotes disabled options which are included in the file for convenience in changing the options.

```
option.COMPRESS: yes - do it
#option.NOCOMPRESS: NO - don't do it
option.DISPLAY: LASER
#option.DISPLAY: PLOT
```

Some additional information is required in order to generate these scripts. This includes: (1) Rgel name, (2) 3 character alphabetic prefix for the project which is used in all script files generated, (3) list of experimental classes in which to put the set of gels after constructing the PCG DB. The names of the Rgels can be specified either interactively or through a Concise Control List (CCL) file (with a `.ccl` file extension). If a CCL file does not exist, then it will be created from the list of gels you enter interactively.

CCL list of gels

The initial **cgelp2** batch input script does a search to find candidate spots for a ratio density normalization and then computes the ratio normalization. It also computes a least-square density normalization. It then does a statistical search comparing all classes with a F-test, t-test and WMW-test at p-values of 0.90, 0.95

and 0.99. Some other searches are also performed. The Rspots found in these searches are saved in SRL subsets. It also computes feature histograms of various Rspot set features as an aid in further setting statistical prefilter limits. This information is illustrated in the following flow chart. *initial PCG DB*

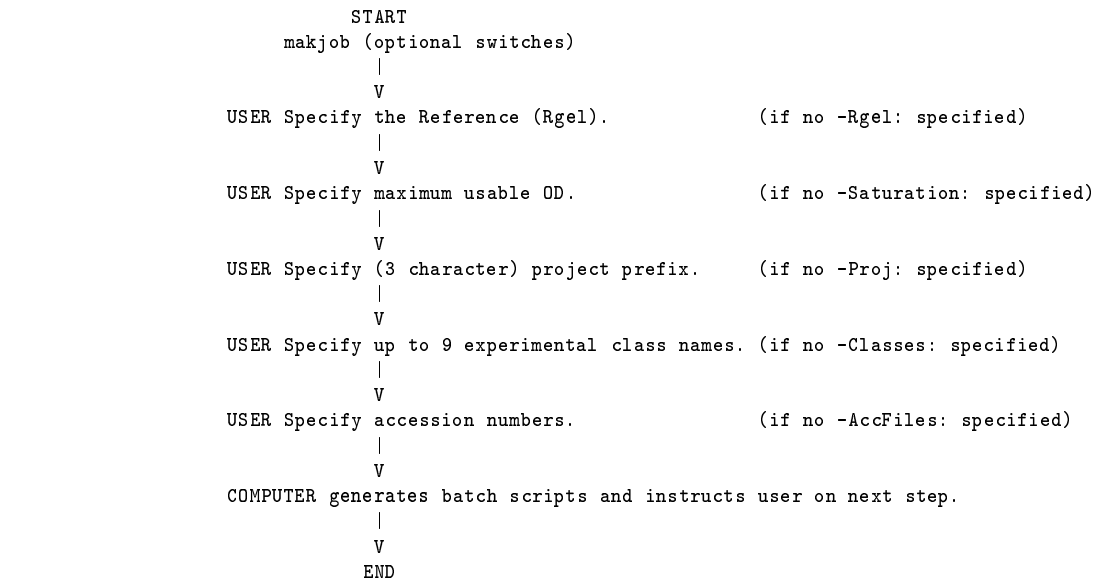


Figure 3.19. Information required by **makjob** in order to generate GELLAB batch scripts. Questions are asked with response expected from the user if the corresponding command line switch options are *not* specified.

USAGE:

```
makjob <opt. -Switches>
```

Type `makjob -info` for more information on makjob.

SWITCHES

-ACcListFile:CCLfileName specify list of gels to be used by a CCL file containing list of accession numbers.

-AUtopairInsteadOfCmpgl2 use **autopair** instead of **cmpgl2** (`-NOAutopair` is the default).

-CCfile:ccgxxx.cg file to create CP (Composite Pair) spots from CC data.

-CLasses:c1:c2:...cN define experimental classes (no spaces).

-CMpgl2Opts:< *options* > additional **cmpgl2** or **autopair** switch options.

-Erspots include eRspots commands in CGL script (default).

-Fields get and print the accession file Data Dictionary fields.

-Info print more information on markgel.

-Histogram include Histogram commands in CGL script (default).

-PAth set the path to 'ppnp1x' from gel.rc (default).

-PLotrmaps construct Rmaps of GSF files after gel comparison and start parallel CGL batch job.

-PRojectPrefix:*3-char-name* define project prefix for batch scripts.

-RGel:RgelName specify the Rgel.

-RMapAndMosaics plot the Rmaps and mosaics for the landmark set of spots. (Default).

-SAturation:*maxOD* define maximum OD where scanner saturation occurs (default 2.0 OD).

-SEgmentation segment and compare the gels specified (default).

-SG2giiOpts:< *options* > additional **sg2gii** switch options.

-STDPPX force `-STD512PPX` processing for **SG2GII** program (equivalent to `-NO1Kx1K`).

-STudy:$f_1 : f_2 : \cdots : f_n$ print accession file files as enter gels.

-Tests include F-test search commands in CGL script (default).

-Usage print UNIX command level switch usage.

-Version print the version of the program.

-WmWait When done, wait until do CLICK TO EXIT `mwwait` widget to exit.

-1Kx1K force `-NOSTD512PPX` processing for **SG2GII** program (default).

EXAMPLES OF USAGE

```
makjob -Rgel:324.1

makjob -AccessionListFile:hm6.ccl
          # Read list of gels from a file.

makjob -AccessionListFile:hm6.ccl -Study:2:13
          # Also print out accession file fields 2 and 13.

makjob -Rgel:324.1 -PRojPrefix:hm6 -CLasses:AML:ALL:CLL:HCL -saturation:2.0OD
          # Specify Rgel, project prefix and classes.

makjob -R:324.1 -PR:hm6 -CL:AML:ALL:CLL:HCL -A:hm6.ccl
          # Short form switches.

makjob -R:0056.1 -CL:CENTER:SIDE -A:ts3.ccl -CCfile:ccgts3.cg -PR:ts3
          # Complete specification from command line.
```

interactive project definition

EXAMPLE 1. - running **makjob** with no switches. The output files are listed at the end of this section. Note the Concise Control List file `prj.ccl` is generated since it did not exist.

```
36% makjob

MAKJOB V-1.3.52 - Version January 26, 1993
Today's date is 07/02/1993, 07:04:47PM
User: /home/joeUser
Written 1982-1992, P. Lemkin.
Searching accession file gel.id for ACC#s
NOTE: Terminate answers with '@' to back up to previous question.
What is the Reference gel?: 324.1 <CR>
What is this project's prefix (3 char)[prj]?: ts3<CR>
Default Project name to 'prj'
Enter gel class [1] name?: aml <CR>
Enter gel class [2] name?: all <CR>
Enter gel class [3] name?: cll <CR>
Enter gel class [4] name?: hcl <CR>
Enter gel class [] name?:  <CR>
ACC# to be processed?: 1024 <CR>
Accession number [1024.1] not in ACCESSION-FILE - ignoring it.
ACC# to be processed?: 369.1 <CR>
ACC# to be processed?: 385.1 <CR>
ACC# to be processed?: 389.1 <CR>
ACC# to be processed?: 384.2 <CR>
ACC# to be processed?: 378.2 <CR>
ACC# to be processed?: 555.2 <CR>
Accession number [0555.2] not in ACCESSION-FILE - ignoring it.
ACC# to be processed?:  <CR>
What is the maximum OD (where saturation begin) [2.00]?: <CR>

Finished creating:
    1.   ts3lms.do - perform interactive landmarking, then start job ts3prc.do.
    2.   ts3prc.do - job to segment and pair gels then start job ts3cgl.do
```

```
     2.1 ts3prc.log - spot segmentation and pairing output log file.
    3.   ts3cgl.do - job to perform CGELP2 database construction.
     3.1 ts3cgl.gdo - 'cgelp2 -file' batch input script.
     3.2 ts3cgl.log - 'cgelp2' output log file.
Three other script files are produced which may be used when segmenting
gels prior to landmarking.
    1'.  ts3seg.do - segment gels into .gsf files.
    2'.  ts3drm.do - (optionally) generate R-map plot files (plotted with
                     'dwrmap' and 'plotn').
    3'.  ts3lms2.do - do landmarking with GSFs, then start job ts3cmp.do.
    4'.  ts3cmp.do - do gel pairing into .gcf files, then start job ts3cgl.do.
The Concise Control List file of accession numbers for this
project is  ts3.ccl

I. Normally you landmark the set of gels interactively under X-windows by
typing
      ts3lms.do
After you finish landmarking the set of gels, ts3lms.do will automatically
submit ts3prc.do which in turn will submit ts3cgl.do

II. Alternatively you can segment the gels BEFORE you landmark them:
      ts3seg.do >& ts3seg.log&        # put into batch

After the gels are segmented you need to run 'landmark' to define
corresponding landmark spots for all of the gels. Because you have
segmented the gels, you can use the SPOT POSTION mode in the 'landmark'
program. As with the other method, you start the landmark process by typing
ts3lms2.do
After you finish landmarking the set of gels, ts3lms.do will automatically
submit ts3prc.do which in turn will submit ts3cgl.do. These will create log
files ts3prc.log and ts3cgl.log respectively.

Steps [1,2,3] are prefered to steps [1',3',4'] because you do not need
to wait for the sementation computations to finish. However, landmarking
may be more accurate if you have the GSF spot positions available during
landmarking (using the 'POSITION AT SPOT CENTER' rather than 'POSITION
EXACT' mode).

   *** ALL FINISHED CREATING BATCH SCRIPTS ***
You can start landmarking by typing:
      ts3lms.do
Real TIME =00:00:05  CPU TIME =00:00:00,  0.00%
```

EXAMPLE 2 - print selected accession entries when enter accession numbers with `makjob -fields`.

listing gel acc-file fields

```
%37  makjob -fields

MAKJOB: Version October 2, 1988
Today's date is 10/02/1988, 04:34:16PM
User: /home/lemkin
Written 1982-1988, P. Lemkin.
```

```
ACCESS. #/PATIENT/BIRTHDATE/RACE&SEX/EXP DATE/EXP #/CULTURE REAG/AMPH,GEL/
     1        2        3         4        5      6         7          8
INTRVL BEFR LBLNG/LBLNG ISOTOPE/DURTN LABEL/DURTN OF EXPSR/STUDY/ FILE #/
         9              10           11            12        13     14
TAPE #/OPT. BACKUP TAPE #/ CAMERA,LENS,DISTANCE/EXPRMNTR/
   15           16                17              18
```

Having selected fields 13 and 14, run **makjob** again as:

```
%38  makjob -Study:13:14

MAKJOB: Version October 2, 1988
Today's date is 10/02/1988, 04:46:10PM
User: /home/lemkin
Written 1982-1988, P. Lemkin.
Searching accession file /home/lemkin/gel.id for ACC#s
NOTE: Terminate answers with '@' to back up to previous question.
What is the Reference gel?: 324.1 <CR>
What is this project's prefix (3 char)[prj]?:hem <CR>
Default Project name to 'hem'
[1] ACC# to be processed?: 369.1 <CR>
   Study: /HEME MALIG-ALL,LYMPHOID/B00889
[2] ACC# to be processed?: 378.2 <CR>
   Study: /HEME MALIG-CLL,LYMPHOID (DUPL. SCAN)/B00945
[3] ACC# to be processed?: 384.1 <CR>
   Study: /HEME MALIG-HCL,LYMPHOID/B00981
[4] ACC# to be processed?: 324.1 <CR>
   Study: /HEME MALIG-AML,MYELOID/B00661
Ignoring Rgel since you already told me.
[4] ACC# to be processed?: 324.2 <CR>
   Study: /HEME MALIG-AML,MYELOID (2 OF 3 SCANS)/B00665
[5] ACC# to be processed?:  <CR>
What is the maximum OD (where saturation begin) [2.00]?: <CR>
```

EXAMPLE 3. Sample **makjob** output script files produced when running **makjob** on `makts3.do`. Note that in all '*.do*' files, comments begin with a '#'. In the `ts3cgl.gdo` file (used as input to **cgelp2**), comments begin in column 1 and are preceded by ';' or '#' while command input to **cgelp2** are prefaced with a '*'.

running from command line

```
39% makjob -rgel:0324.1 -class:AML:ALL:CLL:HCL:HL-60 -accF:ts3.ccl -prj:ts3 -study:2:13 -saturated:2.0
                # Complete command line which produced the following files.


# The following is the ts3.ccl CCL file:
0324.1
0369.1
0378.2
0384.1
0396.1
0497.1
0503.1
```

```
0511.1
0514.1
0515.1
0517.1
0393.2
```

running from command line

```
#!/bin/csh
# JOB ts3cgl.do - CGELP2 database generation 09/07/1992, 08:57:36PM
# The R-gel is: 0324.1

cgelp2 -f ts3cgl.gdo >& ts3cgl.log
date
# ------> THE END <------


#!/bin/csh
# JOB ts3cmp.do - Compare gels 07/02/1993, 07:09:08PM
# The R-gel is: 0324.1
# GELLAB-II version: V-1.3.52

date
cmpgl2 0324.1 0369.1   -COMPRESS  -CHangeThresholds:5,10
cmpgl2 0324.1 0378.2   -COMPRESS  -CHangeThresholds:5,10
cmpgl2 0324.1 0384.1   -COMPRESS  -CHangeThresholds:5,10
cmpgl2 0324.1 0396.1   -COMPRESS  -CHangeThresholds:5,10
cmpgl2 0324.1 0497.1   -COMPRESS  -CHangeThresholds:5,10
cmpgl2 0324.1 0503.1   -COMPRESS  -CHangeThresholds:5,10
cmpgl2 0324.1 0511.1   -COMPRESS  -CHangeThresholds:5,10
cmpgl2 0324.1 0514.1   -COMPRESS  -CHangeThresholds:5,10
cmpgl2 0324.1 0515.1   -COMPRESS  -CHangeThresholds:5,10
cmpgl2 0324.1 0517.1   -COMPRESS  -CHangeThresholds:5,10
cmpgl2 0324.1 0393.2   -COMPRESS  -CHangeThresholds:5,10
date
ts3cgl.do        # -- submit batch job
date
# ------> THE END <------



#!/bin/csh
# JOB ts3drm.do - generate R-map plots from .GSF files 07/02/1993, 07:09:08PM
# The R-gel is: 0324.1
# GELLAB-II version: V-1.3.52

date
dwrmap 0324.1  -NOWAIT -DARKESTSPOTS:45 -TABLE -DISPLAY:LASER
dwrmap 0369.1  -NOWAIT -DARKESTSPOTS:45 -TABLE -DISPLAY:LASER
dwrmap 0378.2  -NOWAIT -DARKESTSPOTS:45 -TABLE -DISPLAY:LASER
dwrmap 0384.1  -NOWAIT -DARKESTSPOTS:45 -TABLE -DISPLAY:LASER
dwrmap 0396.1  -NOWAIT -DARKESTSPOTS:45 -TABLE -DISPLAY:LASER
dwrmap 0497.1  -NOWAIT -DARKESTSPOTS:45 -TABLE -DISPLAY:LASER
dwrmap 0503.1  -NOWAIT -DARKESTSPOTS:45 -TABLE -DISPLAY:LASER
```

```
dwrmap 0511.1  -NOWAIT -DARKESTSPOTS:45 -TABLE -DISPLAY:LASER
dwrmap 0514.1  -NOWAIT -DARKESTSPOTS:45 -TABLE -DISPLAY:LASER
dwrmap 0515.1  -NOWAIT -DARKESTSPOTS:45 -TABLE -DISPLAY:LASER
dwrmap 0517.1  -NOWAIT -DARKESTSPOTS:45 -TABLE -DISPLAY:LASER
dwrmap 0393.2  -NOWAIT -DARKESTSPOTS:45 -TABLE -DISPLAY:LASER
# NOW - use PLOTN to plot R-maps and then manually landmark spots
# entering them into the current .LM file landmark database using
# 'landmark -lmsedit'.
date
# ------> THE END <------


#!/bin/csh
# JOB ts3lms.do - landmark gels 07/02/1993, 07:09:07PM
#       The R-gel is: 0324.1
# GELLAB-II version: V-1.3.52

date
landmark 0324.1 0369.1
landmark 0324.1 0378.2   -AUTOMODE:0369.1
landmark 0324.1 0384.1   -AUTOMODE:0369.1
landmark 0324.1 0396.1   -AUTOMODE:0369.1
landmark 0324.1 0497.1   -AUTOMODE:0369.1
landmark 0324.1 0503.1   -AUTOMODE:0369.1
landmark 0324.1 0511.1   -AUTOMODE:0369.1
landmark 0324.1 0514.1   -AUTOMODE:0369.1
landmark 0324.1 0515.1   -AUTOMODE:0369.1
landmark 0324.1 0517.1   -AUTOMODE:0369.1
landmark 0324.1 0393.2   -AUTOMODE:0369.1
date
ts3prc.do >& ts3prc.log&    # submit batch job

# ------> THE END <------


#!/bin/csh
# JOB ts3prc.do - Segment and compare gels 07/02/1993, 07:09:07PM
#       The R-gel is: 0324.1
# GELLAB-II version: V-1.3.52

date
#             Segment gels to quantitate spots.
sg2gii 0324.1   -COMPRESS
sg2gii 0369.1   -COMPRESS
sg2gii 0378.2   -COMPRESS
sg2gii 0384.1   -COMPRESS
sg2gii 0396.1   -COMPRESS
sg2gii 0497.1   -COMPRESS
sg2gii 0503.1   -COMPRESS
sg2gii 0511.1   -COMPRESS
sg2gii 0514.1   -COMPRESS
sg2gii 0515.1   -COMPRESS
sg2gii 0517.1   -COMPRESS
```

```
sg2gii 0393.2   -COMPRESS
#             Compare gels to pair spot lists.
cmpgl2 0324.1 0369.1   -COMPRESS  -CHangeThresholds:5,10
cmpgl2 0324.1 0378.2   -COMPRESS  -CHangeThresholds:5,10
cmpgl2 0324.1 0384.1   -COMPRESS  -CHangeThresholds:5,10
cmpgl2 0324.1 0396.1   -COMPRESS  -CHangeThresholds:5,10
cmpgl2 0324.1 0497.1   -COMPRESS  -CHangeThresholds:5,10
cmpgl2 0324.1 0503.1   -COMPRESS  -CHangeThresholds:5,10
cmpgl2 0324.1 0511.1   -COMPRESS  -CHangeThresholds:5,10
cmpgl2 0324.1 0514.1   -COMPRESS  -CHangeThresholds:5,10
cmpgl2 0324.1 0515.1   -COMPRESS  -CHangeThresholds:5,10
cmpgl2 0324.1 0517.1   -COMPRESS  -CHangeThresholds:5,10
cmpgl2 0324.1 0393.2   -COMPRESS  -CHangeThresholds:5,10
date
ts3cgl.do         # submit batch job
date

# ------> THE END <------


#!/bin/csh
# JOB ts3seg.do - Segment gels and generate R-map plots 09/07/1992, 08:57:35PM
# The R-gel is: 0324.1

date
sg2gii 0324.1   -COMPRESS
sg2gii 0369.1   -COMPRESS
sg2gii 0378.2   -COMPRESS
sg2gii 0384.1   -COMPRESS
sg2gii 0396.1   -COMPRESS
sg2gii 0497.1   -COMPRESS
sg2gii 0503.1   -COMPRESS
sg2gii 0511.1   -COMPRESS
sg2gii 0514.1   -COMPRESS
sg2gii 0515.1   -COMPRESS
sg2gii 0517.1   -COMPRESS
sg2gii 0393.2   -COMPRESS

# Now you must continue the analysis by
# ts3cmp.do >& ts3cmp.log&        # submit batch job
# after the landmarks have been defined. You might generate 'sg2gii'
# R-maps to do manual landmarking using 'landmark -lmsedit'. To plot
# these files, log into a Tektronix 4010 Type compatible terminal and
# then type
#       ts3drm.do
date

# ------> THE END <------


; JOB ts3cgl.gdo - CGELP2 database generation 07/02/1993, 07:04:51PM
; The R-gel is: 0324.1
;
```

```
; [1] List CGELP2 commands available
*HELP
;
*INQUIRE
*HELP
*
;
; [2] Declare the new paged database file and accession file
;     to be used.
*SYSTEM
*rm -f /home/opus/lemkin/gellab/demo/pcg/ts3pcg.pcg
;
*SET DATABASE FILE
*ts3pcg.pcg
;
*SET ACCESSION FILE
*
;
*SET RGEL//0324.1
*
;
; [3] Declare the gel record fields to be used (on data from
;    the accession file) for study titles.
*SET FIELDS
*2,13
;
; [4] Create a paged CGL (PCG) database using Sure, Possible as
;     well as Ambiguous pairs.
;
*SET REGION LIMITS
*0,512 - pIe limits
*0,512 - MW limits
;
;
; Set up area, density, OD diff and DP sizing for /eRspot DB.
; Set it wide open. Could set limits to only allow robust G2 US spots.
*SET DENSITY
*Absolute
;
*SET STATISTICS
*0,2000 - relative DX,DY
*0,1024 - DL (distance from spot pair to landmark)
*0,100 - DP (distance between spots in a pair)
*0,10000000 - area of spot
*0.0,10000000 - density of spot
*0,10.0 - COV area of spot
*0,10.0 - COV density of spot
*0,10.0 - peak OD seen in any pixel in spot
*.90 - p-Value/confidence limits
*0,1000 - # gels/class
;
*CREATE/eRspot
*0369.1
```

```
*0378.2
*0384.1
*0396.1
*0497.1
*0503.1
*0511.1
*0514.1
*0515.1
*0517.1
*0393.2
*
*USAPC
; The following null response indicates that the default value should
; be used. It is the maximum effective dP distance allowed for eRspot
; creation when adding a US spot from a non-Rgel gel to an existing eRspot -
; default is (30 pixels * 170 microns / gel resolution in microns/pixel).
*
;
*BACKUP  - Save the new PCG DB
;
; [4.1] List the valid landmarks for each gel.
*VALIDLANDMARKS/ListLMS
;
; [5] Declare gel class names for use in automatic partitioning
; the gels into classes by finding the keyword class name in
; the gel study title.
*SET CLASSES
*AUTO
*Y
*AML
*ALL
*CLL
*HCL
*<null>
*<null>
*<null>
*<null>
*<null>
;
;
; [5.1] List the classes set up
*SET CLASSES
*NO
;
; [5.2] Define GEL subsets
*SET GEL SUBSETS
*WORKING SET
*working set of gels
*
;
; [6] The following set of commands is used to find a set of R-spots
; found in all gels with mean uncorrected D prime. These spots
; are then used to normalize the database. Alternatively, the
```

```
;       Least-Squares density calibration is computed and used to reorder
;       it based on the new density measure.
*SET LABEL - only normalize with Rspot sets consisting of SP and PPs
*PSC
;
*SET DENSITY MODE
*Uncorrected
;
;
;       Relax limit to include all spots for normalization
*SET STATISTICS
*0,1024 - relative distance of (Dx,Dy) from Landmark
*0,1024 - mean DL
*0,1024 - mean DP
*0,10000000 - mean area
*0,10000000 - mean density
*0,10.0 - CV of area
*0,10.0 - CV of density
*0,2.000 - mean OD difference (do not use saturating spots)
*.90 - two class significance limit
*ALL - test only R-spot sets with all spots.
;
*INQUIRE - search for normalization R-spots
*Search
;
*SET SRL SUBSETS
*Assign
*normalization spots non-saturating and found in all gels
*
;
*SET RATIO LIST
*$
;
;
; [6.1] Compute Least Squares density normalization and reorder
;       PCG DB.  Tighten limits for robust spots for normalization.
*SET STATISTICS
*0,1024 - relative distance of (Dx,Dy) from Landmark
*0,1024 - mean DL
*0,1024 - mean DP
*25,10000000 - mean area
*3,10000000 - mean density
*0,10.0 - CV of area
*0,10.0 - CV of density
*0,2.000 - mean OD difference (do not use saturating spots)
*.90 - two class significance limit
*0,1000 - relax # spots/Class.
;
*SET LABEL - only normalize with Rspot sets consisting of SP and PPs
*PSC
;
*SET LEAST SQUARES CALIBRATION - Calibrate as Least Squares.
*Yes
```

```
;
*SET DENSITY MODE - change the density mode to LSQ.
*Least squares
;
;        Relax limit to include all spots for normalization.
*SET STATISTICS
*0,1024
*0,1024
*0,1024
*0,10000000
*0,10000000
*0,10.0
*0,10.0
*0,10.0
*.90
*0,1000 - all Rspot sets
;
*SET LABEL - reorder all spots in the database
*USAPCEX
;
*REORDER - database based on rank order of Least Squares normalization.
;
*SET LABEL - only search for SP, PP and US in Rspot and eRspot DBs.
*PSCUX
;
;
*SET SRL SUBSETS - create SPSS file and script to make mosaics
*SPSS/MOSAIC
*1
*
;
; Save the database in case of crash during searches
*BACKUP
*EXTRAPOLATE/QUIET
*BACKUP
;
;
; [7] List the gel names, # spots/gel and total D prime/gel and
; each gels least square density normalization parameters.
*GELS
;
;
; [8] Save the database in a file which may be printed.
;       IGnore Rspots based solely on AP pairing labels.
; [DISABLED - you may edit this back if you like].
;*DUMP CGL ascii database
;*ts3new.cgl
;
;
; [9] Find the list of landmarks in the database.
*SET WORKING GELS
*Define - to just the Rgel
*0324.1
```

```
;
*INQUIRE
*LANDMARKS
;
*SET WORKING GELS
*Define - to entire set of gels
*ALL
;
;
*SET SRL SUBSETS
*Assign
*landmarks set of spots
*SPSS/MOSAIC - create SPSS file and script to make mosaics
*<LAST>
*
;
; [9.1] Print the Ordered Expression-Profile table for landmark spots.
*SET SRL SUBSETS
*RESTORE
*Landmarks
*
*INQUIRE
*OExpression-Profile
*1.00    ;minimum Least Square Error threshold
;
*INQUIRE
*OExpression-Profile
*0.50    ;minimum Least Square Error threshold
;
*INQUIRE
*OExpression-Profile
*0.25    ;minimum Least Square Error threshold
;
*INQUIRE
*OExpression-Profile
*0.10    ;minimum Least Square Error threshold
;
;
;
;
;
; [10] Perform .90, .95 and .99 p-value F-test searches on all classes.
;
; [10.1] Perform a 90% p-value level F-test search on all classes.
*SET STATISTICS
*0,1024
*0,1024
*0,1024
*0,10000000
*0,10000000
*0,10.0
*0,10.0
*0,2.000 - mean OD difference (do not use saturating spots)
```

```
*.90
*0,1000 - test all Rspot sets
;
*INQUIRE
*F-test
*1,2,3,4
;
*SET SRL SUBSETS
*Assign
*F-test of all classes at p-value=0.90
*SPSS/MOSAIC - create SPSS file and script to make mosaics
*<LAST>
*
;
; [10.2] Perform a 95% p-value level F-test search on all classes.
*SET STATISTICS
*0,1024
*0,1024
*0,1024
*0,10000000
*0,10000000
*0,10.0
*0,10.0
*0,2.000 - mean OD difference (do not use saturating spots)
*.95
*0,1000 - test all Rspot sets
;
*INQUIRE
*F-test
*1,2,3,4
;
*SET SRL SUBSETS
*Assign
*F-test of all classes at p-value=0.95
*SPSS/MOSAIC - create SPSS file and script to make mosaics
*<LAST>
*
;
; [10.3] Perform a .99 p-value F-test search on all classes.
*SET STATISTICS
*0,1024
*0,1024
*0,1024
*0,10000000
*0,10000000
*0,10.0
*0,10.0
*0,2.000 - mean OD difference (do not use saturating spots)
*1%
*0,1000 - test all Rspot sets
;
*INQUIRE
*F-Test
```

```
*1,2,3,4
;
*SET SRL SUBSETS
*Assign
*F-test of all classes at p-value=0.99
*SPSS/MOSAIC - create SPSS file and script to make mosaics
*<LAST>
*
;
*BACKUP - checkpoint database
;
;
;
; [11] Perform a .90, .95 and .99 p-value t-test search on classes(1,2).
;
; [11.1] Perform a 90% p-value level t-test search on classes (1,2).
*SET STATISTICS
*0,1024
*0,1024
*0,1024
*0,10000000
*0,10000000
*0,10.0
*0,10.0
*0,2.000 - mean OD difference (do not use saturating spots)
*.90
*0,1000 - test all Rspot sets
;
*INQUIRE
*TB-test
*1,2
;
*SET SRL SUBSETS
*Assign
*TB-test of classes (1,2) at p-value=0.90
*SPSS/MOSAIC - create SPSS file and script to make mosaics
*<LAST>
*
;
; [11.2] Perform a 95% p-value level t-test search on classes (1,2).
*SET STATISTICS
*0,1024
*0,1024
*0,1024
*0,10000000
*0,10000000
*0,10.0
*0,10.0
*0,2.000 - mean OD difference (do not use saturating spots)
*.95
*0,1000 - test all Rspot sets
;
*INQUIRE
```

```
*TB-test
*1,2
;
*SET SRL SUBSETS
*Assign
*TB-test of classes (1,2) at p-value=0.95
*SPSS/MOSAIC - create SPSS file and script to make mosaics
*<LAST>
*
;
; [11.3] Perform a .99 p-value t-test search on classes(1,2).
*SET STATISTICS
*0,1024
*0,1024
*0,1024
*0,10000000
*0,10000000
*0,10.0
*0,10.0
*0,2.000 - mean OD difference (do not use saturating spots)
*1%
*0,1000 - test all Rspot sets
;
*INQUIRE
*TB-Test
*1,2
;
*SET SRL SUBSETS
*Assign
*TB-test of classes (1,2) at p-value=0.99
*SPSS/MOSAIC - create SPSS file and script to make mosaics
*<LAST>
*
;
*BACKUP - checkpoint database
;
;
; [12] Perform a .90, .95 and .99 p-value Wilcoxon-Mann-Whitney
;            Rank-Sum test search on classes 1 and 2.
;
; [12.1] Perform a 90% p-value level WMW-test search on classes 1 and 2.
*SET STATISTICS
*0,1024
*0,1024
*0,1024
*0,10000000
*0,10000000
*0,10.0
*0,10.0
*0,10.0
*.90
*0,1000 - test all Rspot sets
;
```

```
*INQUIRE
*WMW-test
*1,2
;
*SET SRL SUBSETS
*Assign
*WMW-test of classes (1,2) at p-value=0.90
*SPSS/MOSAIC - create SPSS file and script to make mosaics
*<LAST>
*
;
; [12.2] Perform a 95% p-value level WMW-test search on classes 1 and 2.
*SET STATISTICS
*0,1024
*0,1024
*0,1024
*0,10000000
*0,10000000
*0,10.0
*0,10.0
*0,10.0
*.95
*0,1000 - test all Rspot sets
;
*INQUIRE
*WMW-test
*1,2
;
*SET SRL SUBSETS
*Assign
*WMW-test of classes (1,2) at p-value=0.95
*SPSS/MOSAIC - create SPSS file and script to make mosaics
*<LAST>
*
;
; [12.3] Perform a .99 p-value WMW-test search on classes 1 and 2.
*SET STATISTICS
*0,1024
*0,1024
*0,1024
*0,10000000
*0,10000000
*0,10.0
*0,10.0
*0,10.0
*1%
*0,1000 - test all Rspot sets
;
*INQUIRE
*WMW-Test
*1,2
;
*SET SRL SUBSETS
```

```
*Assign
*WMW-test of classes (1,2) at p-value=0.99
*SPSS/MOSAIC - create SPSS file and script to make mosaics
*<LAST>
*
;
*BACKUP - checkpoint database
;
;
;
; [13] Perform a MISSING-SPOT-test search on classes 1 and 2.
;      First on gels present in class 1, then those present
;            in class 2.
*SET STATISTICS
*0,1024
*0,1024
*0,1024
*0,10000000
*0,10000000
*0,10.0
*0,10.0
*0,10.0
*1%
*3,1000 - minimum of 3 gels per Rspot sets to be counted as present
;
*INQUIRE
*MISSING-SPOT-test/MustBeIn1stClass
*1,2
;
;
*SET SRL SUBSETS
*Assign
*MISSING-CLASS-test of classes 1 (present) and 2 min #gels/class=3
*SPSS/MOSAIC - create SPSS file and script to make mosaics
*<LAST>
*
;
*INQUIRE
*MISSING-SPOT-test/MustBeIn1stClass
*2,1
;
*SET SRL SUBSETS
*Assign
*MISSING-CLASS-test of classes 1 and 2 (present) min #gels/class=3
*SPSS/MOSAIC - create SPSS file and script to make mosaics
*<LAST>
*
;
*BACKUP - checkpoint database
;
;
;      [14] Create an SPSS file with set of all Rspots which occur or
;           are extrapolated in the Rgel.
```

```
*SET WORKING SET//DEFINE//0324.1
;          Set prefilter limits wide open.
*SET STATISTICS
*0,1024
*0,1024
*0,1024
*0,10000000
*0,10000000
*0,10.0
*0,10.0
*0,10.0
*1%
*0,1000 - test all Rspot sets
;
*SET LABEL//PSEUX
*INQUIRE//SEARCH
*SET SRL SUBSETS//ASSIGN//all PSEUX Rspots for Rgel 0324.1////
*SET SRL SUBSET//SPSS//<LAST>////
*SET WORKING SET//DEFINE//<ALL>
*SET LABEL//PSUX
;
;
;       [15] Intersection of LANDMARK and NORMALIZATION spots.
*SET SRL SUBSETS//INTERSECTION//NORMALIZATION//LANDMARKS////
*SET SRL//ASSIGN//Intersection of LANDMARKS and NORMALIZATION spots////
;*SET SRL SUBSET//LIST/DIR////
*SET SRL SUBSET//DIRECTORY////
;
;
; [16] Compute global histograms of selected database features.
;            Set prefilter limits wide open.
*SET STATISTICS
*0,1024
*0,1024
*0,1024
*0,10000000
*0,10000000
*0,10.0
*0,10.0
*0,10.0
*1%
*0,1000
;
*SET DISPLAY//PLOT
*
;
*HISTOGRAM/QUIET
*Number of gels per Rset
;
*HISTOGRAM/QUIET
*Individual spot density
;
*HISTOGRAM/QUIET
```

```
*Set mean Rspot set density
;
*HISTOGRAM/QUIET
*Maximum spot density
;
*HISTOGRAM/QUIET
*Area of spots
;
*HISTOGRAM/QUIET
*P - DP values of spots
;
*HISTOGRAM/QUIET
*L - DL values of spots
;
*HISTOGRAM/QUIET
*Variation of std dev/mean Rset density
;
*HISTOGRAM/QUIET
*OD difference of spot densities
;
; [16.1] GEL-GEL and CLASS-CLASS density correlations.
;
*TABLE
*Correlation
*ts3mnv.tbl
;
; [16.2] Plot CLASS vs CLASS scatter plots for normalization,
; landmark, and .99 T-test SRLs.
*CCPLOT// /CLASSNAMES
;
*CCPLOT/SRL[1]// /CLASSNAMES
;
*CCPLOT/SRL[2]// /CLASSNAMES
;
*CCPLOT/SRL[8]// /CLASSNAMES

;
; [17] Save the database;
*GELS/FULL
;
*FEATURES
;
*SET GEL SUBSETS
*LIST
*<ALL>
*
*SET SRL SUBSETS
;*LIST/DIR
*DIRECTORY
*<ALL>
*
;
*LIMITS
```

```
;
;
; [18] Plot Rmaps and mosaics of landmark spots for all gels
; in the database;
*SET SRL SUBSETS//RESTORE//landmarks////
*
;
*RMAP//Allgels /noPPXplot/SrlLabel////
;
*MOSAIC//AllSRLspots/noPPXplot/SrlLabel////
;
;
;
; [18.1] Plot Expression Profile plots for normalization, landmark,
; and .99 T-test SRLs.
*SET SRL SUBSETS//RESTORE//1////
*EXPPLOT// /NOLINES
;
*SET SRL SUBSETS//RESTORE//2////
*EXPPLOT// /NOLINES
;
*SET SRL SUBSETS//RESTORE//8////
*EXPPLOT// /NOLINES

;
;       [19] Generate dendrograms for selected SRL subsets. These include
;            the NORMALIZATION  SRL[1], LANDMARKS SRL[2], and TB t-Test
;            (.95) of classes (1,2) SRL[7].  Cluster Gel objects as a
;            function of the set of Rspots in each SRL. Then cluster the
;            Rspots as objects as a function of their expression profiles
;            of the working set of gels for each Rspot and .95 T-test SRLs.

; Cluster Gels as a function of a set of Rspots.
*SET SRL SUBSETS//RESTORE//1////
*DENDROGRAM// $/ClusterGels
;
*SET SRL SUBSETS//RESTORE//2////
*DENDROGRAM// $ /ClusterGels
;
*SET SRL SUBSETS//RESTORE//7////
*DENDROGRAM// $ /ClusterGels

;           Cluster Rspots as a function of the expression profile of a
;           set of gels by invoking the -MEANCLASSES switch of the
;           dendrogram program.
*SET SRL SUBSETS//RESTORE//1////
*DENDROGRAM// $ /ClusterRspots
;
*SET SRL SUBSETS//RESTORE//2////
*DENDROGRAM// $ /ClusterRspots
;
*SET SRL SUBSETS//RESTORE//7////
*DENDROGRAM// $ /ClusterRspots
```

```
;
;  [20] Save PCG DB (checkpointing it) and exit.
*EXIT
# ------> THE END <------
;
```

3.10 markgel - Generate Rmaps of Rspots from .sps files

An *Rmap* is a derived gel image which illustrates particular spots of interest by labeling them in a copy of some particular gel. The list of spots is specified by an SPSS file which is created using the **cgelp2** database program.

uses SPSS data file

The **markgel** program takes an SPSS (*.sps*) input file and generates a Rmap image file. This derived image file is given the name `m`*xxxxx*`.ppx`. The *xxxxx* is the same gel picture file number as specified in the accession file entry for this gel. Normally the Rmap image is saved on the `gel.rc` specified `ppnp2x` auxiliary directory.

displaying Rmap

Alternatively, the `-name` switch can be used to force a particular output picture file name. This is useful when using the `-Xpix` option (invoke **Xpix** after computing the Rmap to display it) and you do not want to build up a large number of picture files on the disk (hint: use `-name:rmap.ppx`). If the picture file name is proceeded by '`./`' as in `-name:./rmap.ppx`, then the current working directory is used for the output image.

'+' marks the spot

The Rmap is generated for all of the Rspots found in the SPSS file for the specified gel. Each Rspot is denoted by a white '+' (3x3 pixels in size) drawn in its center followed by a white Rspot number. The name of the SPSS file and today's date is written at the bottom of the image. This is followed by the 'title' from the SPSS file and then the accession information for this gel. Generally, the image has an optimal background gray value *optionalValue* (default 50 - where 0 is 'white' and 255 is 'black') added to the image to enable the labels to be seen clearly. The `-GRAPHSCALE` option (the default), generates a **Xpix** pseudocolor *graphscale* image with the graphics drawn in red. Graphscale is a grayscale image with overlayed color graphics. Otherwise, with `-noGraphScale`, the labels are drawn in white which is better for photographing with black and white film.

options

The synthesized Rmap image may be flipped along one or both the pIe (`-HflipPIE`) or MW (`-VflipMW`) axes. The image may be zoomed by specifying `-Zoom:nX`. In this case, the center of the set of spots in the synthesized image is used as the center of the image. In addition, it may be zoomed about a particular spot, `-Zoom:nX:Rspot#`. If the SPSS file has AP and/or EP spots, the `-restrictLabel` switch can be used to tell markgel NOT TO DRAW the AP and EP spots if found in the SPSS file. If the SPSS file is not specified, it attempts to find the default file `rmap.sps` which is then used. The `-laser` <*opt. printer name*> switch can be used to dump the Rmap to the laser printer using `ppx2ps -height:7` *Rmap.ppx* | `lpr -Pprinter`.

If the `-Xpix` switch is specified, then display the Rmap with **Xpix** when finished computing the Rmap image. use the **Xpix** EXIT selection from the VIEW OPR

menu to terminate both **Xpix** and **markgel**. Figure 3.20 shows a derived Rmap gel image with selected Rspots labeled on a copy of one of the gels from a composite gel DB. *controlling image display*

Figure 3.20. Rmap gel image. Derived Rmap gel image with selected Rspots of interest labeled on a copy of one of the gels from a composite gel DB. The data used by **markgel** is derived from a SPSS data file which was generated from a search results list subset of spots generated by the **cgelp2** PCG DB program.

USAGE:

```
markgel <Rgel> <SPSS .sps-file> <opt. -Switches>
```

Type `markgel -info` for more information on **markgel**.

SWITCHES

-COMpress derived image file after they are created on the disk to save space.

-CONtrast enhance the image in the range above optimum background.

-CORrectbackground:optionalValue add value to the background to see labels (default value is 50). `-NOCorrectbackground` uses 0.

-Fill background of lettering with complementary color.

-GRaphScale generate a *GraphScale* pseudo color processed PPX image with default red labeling instead of the white label if use `-NOGRAPHSCALE`.

-HFlip pIe reverse output image about horizontal pIe axis.

-Info print more information on markgel.

-LARge labels use large (5x7) instead of small (4x6) spot labels. (Default).

-LASer:<opt. printer-name> use `ppx2ps | lpr -Plaser` to print the Rmap to the laser printer.

-NAme:outputfile force outputFile to be the output .ppx file name.

-NUmber draw Rspot numeric labels after '+' (default).

-Restrict label of spots to only US+SP+PP+CP spots - ignore EP and AP (default).

-SIlent do not print output on the terminal.

-SWlist:"<List of switches>" allows passing of a switch list to child **Xpix** process. For example: `-swList:"-ppx -rgb"` passes the switches `-ppx -rgb`.

-Title:'text' use the specified text in the title field.

-USAge print UNIX command level switch usage.

-USElandmarks instead of Rspot numbers where applicable.

-VErsion print the RCS version of the program.

-VFlip MW reverse output image about MW axis.

-WmWait When done, wait until do CLICK TO EXIT `mwwait` widget to exit.

-Xpix display the Rmap image using **Xpix** program when finished.

-Ytitle:newYposition position the 4 line title at the new Y-position (default is imageSize-62).

-Zoom:nX(Opt. Rspot:j) zoom nX where n is 1, 2, 4 or 8 else default to 1X. If the *Rspot[j]* additional switch is specified, then center the Rmap around that Rspot, otherwise center it around the centroid of the set of Rspots being displayed. (Default is `-Zoom:1X`).

EXAMPLES OF USAGE

```
markgel 324.1 ts3s02.sps
         # Default switches. This will generate the default Graphscale
         # image with labeling information in RED.

markgel 324.1 ts3s02.sps -noGraphscale
         # Same as above but labeling information in WHITE which
         # is better for photographing with black and white film.

markgel 324.1 ts3s02.sps -nonumber
         # Do not number labeled spots.

markgel 324.1 ts3s02.sps -non
         # Short form of switches.

markgel 324.1 ts3s02.sps -uselandmarks
         # If a spot is a landmark spot, then label it A, B, ...

markgel 324.1 ts3s01.sps -nolarge -zoom:4X
         # Zoom the image 4X about centroid of labeled spots.

markgel 324.1 ts3s01.sps -nolarge -zoom:4X:273
         # Zoom the image 4X about Rspot 273.

markgel 324.1 ls2s03.sps -vFlipMW -hFlippIe
         # Reverse image in both MW and pIe.

markgel 324.1 ts3s05.sps -Xpix
         # Compute and display the Rmap in an Xpix window.
         # pass switches to Xpix.

markgel 324.1 ts3s05.sps -laser:qms
         # Compute and print Rmap on laser printer called 'qms'.
```

EXAMPLE 1. Create a Rmap of SRL[23] which was previously saved in SPSS file `ts3s23.sps`. Zoom it by 2X around the centroid of the spots in the SPSS file. *composite Rmap*

```
25% markgel 0324.1 ts3s23.sps -zoom:2X -large

MARKGEL :Version October 30, 1988
Today's date: 01/28/1989, 12:59:15AM
Written 1982-1988, P. Lemkin.
```

```
Magnification is 2X.
Generating a map of R-spots for gel 0324.1 from file ts3s23.sps.
0324.1/.../?/?/1-18-82/#12/CULT #2/3:10,5-20%/
0 HRS/H3/2 HRS/96 HRS/HEME MALIG-AML,MYELOID/
A00661/-NONE-/-NONE-/VIDICON-AUTO,28MM,F8,76CM/LESTER*
 027 049 072 095 117 136 153 168 181 192 200 208 213 220 225 000 036 497 074 509
Creation date:  01/26/1989, 10:29:15AM
Title: Rspots from intersection of SRL subsets 17, 18, 19.
Estimating Centroid [29 Rspots] at (279,254)
Creating Rspot Image file: /home/joeUser/m00661.ppx
R-spot[7] at (304,184)
R-spot[9] at (302,191)
R-spot[25] at (300,146)
R-spot[61] at (317,199)
R-spot[71] at (324,227)
R-spot[93] at (406,218)
R-spot[97] at (357,221)
R-spot[102] at (354,244)
R-spot[150] at (321,283)
R-spot[166] at (342,320)
R-spot[193] at (261,261)
R-spot[194] at (247,266)
R-spot[233] at (246,173)
R-spot[236] at (246,181)
R-spot[237] at (240,183)
R-spot[281] at (258,178)
R-spot[282] at (270,179)
R-spot[283] at (263,187)
R-spot[393] at (239,214)
R-spot[445] at (210,267)
R-spot[505] at (244,329)
R-spot[533] at (280,359)
R-spot[594] at (355,381)
R-spot[649] at (262,413)
R-spot[660] at (245,448)
Real TIME =00:00:13  CPU TIME =00:00:10,  76.92%

Finished creating Rspot Image file: /home/joeUser/m00661.ppx
```

compute & display Rmap

EXAMPLE 2. Same as above but do not print anything on the display, just compute and put up the X-windows image.

```
26% markgel 0324.1 ts3s23.sps -zoom:2X -large -Xpix -silent
```

compute & display Rmap

EXAMPLE 3. Similar to above but dump the picture on the laser printer called `qms`.

```
27% markgel 0324.1 ts3s23.sps -Laser:qms
```

3.11 mosaic - Generate mosaics of Rspots from .sps files

A *mosaic* is a derived gel image which illustrates a particular spot of interest in a number of different gels by extracting local regions surrounding that spot in each of these gels. These panels are then spliced together in the derived image sorted by minimum spot integrated optical density as illustrated below. Each panel is labeled with corresponding gel identification information. The spot to be used in the mosaic must be in the list of spots is specified by a SPSS file (which is created using the **cgelp2** database program).

uses SPSS data file

The **mosaic** program takes an SPSS (*.sps*) input file produced by **cgelp2** and generates up to 36 mosaic image(s) named `wjrrrr.ppx` where *rrrr* is the Rspot number and *j* is the mosaic sequence number counting from 0 to 9 then *a* to *z*. For example, for Rspot `123`, then `w00123.ppx`, `w10123.ppx` $\cdots$ `wz0123.ppx`, etc. images would be generated. The mosaic image is saved on the `gel.rc` specified `ppnp2x` auxiliary directory.

Since each image holds 16 gel sub-images in a 4x4 grid, the additional images are generated only when needed to create up to 576 (i.e. 36x16) gels (where *j* ranges from 0 to 9 and *a* to *z*) in the mosaic. Normally, the panels are ordered lowest density in the upper left hand corner and highest in the lower right hand corner. Spot density increases going from left to right and then from top to bottom. It also increases with increasing panel number *j*. For example, these appear on the screen sorted as $G_1 < G_2 < \cdots G_n$.

gel ordering convention

```
<==lighest
       ---------------------
       | G1 | G2 | G3 | G4 |
       ---------------------
       | G5 | G6 | G7 | G8 |
       ---------------------
       | G9 | G10| G11| G12|
       ---------------------
       | G13| G14| G15| G16|
       ---------------------
                        ==> darkest
```

Alternatively, the `-name` switch can be used to force a particular output picture file name. This is useful when using the `-Xpix` option (invoke **Xpix** after computing the mosaic to display it) and you do not want to build up a large number of picture files on the disk (hint: use `-name:mosaic.ppx`). The `-laser` <*opt. printer name*> switch can be used to dump the mosaic images to the laser printer using `ppx2ps -height:7` *mosaic.ppx* `|` `lpr -Pprinter`. If the picture file name is proceeded by '`./`' as in `-name:./mosaic.ppx`, then the current working directory is used for the mosaic output image. Normally the mosaic image is saved on the `gel.rc` specified

image display

`ppnp2x` auxiliary directory. If more than one mosaic is generated, the names are of the form `mosaic0.ppx, mosaic1.ppx,` etc. Normally the mosaic images are saved on the `gel.rc` specified `ppnp2x` auxiliary directory.

Alternatively, instead of creating the mosaic images from a set of gels for the *same* spot it is possible to see several spots for the same gel or several spots for all gels. This is specified by the `-gel` switch. If you want to restrict it to a single gel (say `324.1`), use `-gel:324.1` and only Rspots for that gel will be included in the mosaic image.

labels

The image has the Rspot region from each corresponding image and its extracted (2X default) spot region inserted into 128x128 pixel subregions of the 512x512 pixel output image. At the bottom of each of these subregions is an accession number label. The name of the SPSS file, Rspot number of the mosaic and today's date is written at the top of the image. This is followed by the 'title' from the SPSS file. The Rspot has a 3x3 "+" drawn in its center. Each gel subregion is labeled by default with information specifying (*gel acc#, class#/name, density, pairing label*). Various switches can be use to reconfigure what is drawn or omitted in this label.

options

In order to view a set of different gels which may have different mean background densitites, it is necessary to adjust their background values to be the same. If this is not done, then at the current contrast range of the display some gel pannels will be visible and others not. Generally, the image has an optimal background grayvalue `optionalValue` (default 50 - where 0 is 'white' and 255 is 'black') added to the image to enable the labels to be seen clearly. A histogram of the top line of each gel pannel is computed and the maximum peak value assumed to be the background (`-quickBackground` switch). If this fails because there was a dark streak in this position, use the `-noQuickBackground` option to compute the histogram over the entire pannel to get a better estimate.

The `-GraphScale` option (the default), generates a **Xpix** pseudocolor *graph-scale* image with the graphics drawn in red. Graphscale is a grayscale image with overlayed color graphics. Otherwise, with `-noGraphScale`, the labels are drawn in white which is better for photographing with black and white film.

The synthesized images may be flipped along one or both the pIe (`-HflipPIE`) or MW (`-VflipMW`) axes. The images may be zoomed by specifying `-Zoom:nX` (2X default). If the SPSS file has AP and/or EP spots, the `-restrictLabel` switch can be used to tell mosaic NOT TO DRAW the AP and EP spots if found in the SPSS file. If the SPSS file is not specified, it attempts to find the default file `mosaic.sps` which is then used.

Figure 3.21. Mosaic gel image. Derived mosaic gel image with subregions from a set of gels which surround the selected Rspot from a composite gel DB. The data used by **mosaic** is derived from a SPSS data file which was generated from a search results list subset of spots generated by the **cgelp2** PCG DB program.

controlling image display

If the `-Xpix` switch is specified, then display the mosaic image(s) using the **Xpix** program when finished computing the mosaic(s). If there is more than one mosaic image generated, then they are fed to **Xpix** two at a time - starting from the lightest (i.e. least dense) to the darkest spots. That is, when you `EXIT` selection from the `VIEW OPR` menu in **Xpix**, it gets the next two mosaics and so on until you have gone though all of them. Figure 3.21 shows a derived mosaic gel image with subregions from a set of gels which surround the selected Rspot from a composite gel DB.

USAGE:

```
mosaic <Rspot#> <SPSS filename> [<Opt. -switches>]
```

Type `mosaic -info` to get more information.

SWITCHES

-CLass display the class name/number (default).

-COMpress derived image file(s) after they are created on the disk to save space.

-CONtrast enhance the image in the range above optimum background.

-DEnsity display density information for each spot (default).

-DIsplay name display class name rather than number (default).

-Fill background of lettering with complementary color.

-GEl:[optional ACC# to plot all Rspots in the mosaic for all gels or for a particular gel if the `acc#` argument is specified.

-GRaphScale generate a *GraphScale* pseudo color processed PPX image with default red labeling instead of the white label if use `-NOGRAPHSCALE`.

-HFlip pIe flip about y Axis (i.e. swap acid and base orientation).

-Info print additional information about MOSAIC.

-LABel print spot pairing information labels (default).

-LASer:<opt. printer-name> use `ppx2ps | lpr -Plaser` to print the mosaics to the laser printer.

-Name:outputfile force outputFile to be the output .ppx file name.

-Prefix:letter of image to use in mosaic.

-Quickbackground estimate background using min of 1st line of each window rather than the histogram peak of the window (default).

-Restrict Label of spots to only US+SP+PP+CP spots], ignore EP and AP (default).

-SIlent do not print output on the terminal.

-SEssion display entire session statistics (default). `-NOSESSION` causes a shorter version to be printed.

-SWlist:"<List of switches>" allows passing of a switch list to child **Xpix** process. For example: `-swList:"-mouse -rgb"` passes the switches `-mouse -rgb`.

-TExt don't display any of the spot specific information normally shown at the bottom of each tile.

-TItle:'text' use the specified text in the title field. `-notitle` causes no title to be displayed (default is SPSS title).

-Usage print unix command level switch usage.

-Version print the version of the program.

-Vflip mw flip image about x Axis (i.e. high and low MW orientation).

-WmWait When done, wait until do CLICK TO EXIT `mwwait` widget to exit.

-Xpix display the mosaic image using **Xpix** program when finished.

-Zoom:nX zoom nx where n is 1, 2, 4 or 8 else default to 2x.

EXAMPLES OF USAGE

```
mosaic 273 ts3s02.sps
         # Default (generates a graphscale image with RED labeling).

mosaic 273 ts3s02.sps -noGraphscale
         # Same as above, but with WHITE labeling.

mosaic 273 ts3s02.sps -NOquickbackground
         # Compute exact background correction for each gel.

mosaic 273 ts3s02.sps -NOq
         # Short form command.
```

```
mosaic 273 ts3s02.sps -uselandmarks
         # Draw mosaic with Landmark label if it is a LM spot.

mosaic 273 ts3s01.sps -nolarge -zoom:4X
         # Draw mosaic with small letters 4X zoom

mosaic 273 ts3s01.sps -vflipMW -hflippIe
         # Draw mosaic flipped in MW and pIe

mosaic 273
         # Draw mosaic, assume that mosaic.sps is the SPSS file.

mosaic 273 ts3s02.sps -Xpix
         # Compute mosaic and display in using an Xpix window

mosaic 273 ts3s02.sps -Xpix -SWlist:"-mouse -full"
         # Same as above, but display them with Xpix and
         # pass switches to Xpix.

mosaic 273 ts3s02.sps -laser:qms
         # Compute and print Rmap on laser printer called 'qms'.

mosaic ts3s02.sps -gels
         # Compute mosaic of all Rspots for all gels in SPSS file

mosaic ts3s02.sps -gels:384.1
         # Compute mosaic of all Rspots for just gel 0384.1  in SPSS file
```

EXAMPLE 1. Generate a mosaic of Rspot 166.

```
26% mosaic 166 ts3s09.sps -label

MOSAIC - Version November 30, 1988
Today's date: 01/29/1989, 07:20:12AM
Written 1982-1988, P. Lemkin.
Magnification of gel inserts is 2X.
Labeling R-spot[166] in SPSS file:ts3s04.sps
Creation date:01/29/1989, 07:20:04AM
Title: T-TEST CLASSES (1,3) AT P=.90
There are 12 gels.
These are divided into 1 pictures.
Creating Rspot Image file: /home/joeUser/w00166.ppx
Doing frames [1:9]
#1 Doing Gel 0566.1 index 315 at [290,258]
#2 Doing Gel 0324.1 index 491 at [342,320]
     .
     .
     .
#9 Doing Gel 0544.1 index 516 at [314,300]
Drawing labels in mosaic.
0366.1 L:PP D:0.37R
 -CLL
```

```
0324.1 L:PP D:0.38R
 -AML
      .
      .
      .
0544.1 L:PP D:2.78R
 -ALL
Real TIME =00:00:49  CPU TIME =00:00:33,  67.35%
```

EXAMPLE 2. Same as above but do not print anything on the terminal window. Just display mosaic image in an X-window.

```
27% mosaic 166 ts3s09.sps -Xpix -silent
```

EXAMPLE 3. Same as above but dump the mosaic images on the laser printer qms.

```
27% mosaic 166 ts3s09.sps -laser:qms
```

3.12 parg - Prompt and map UNIX command line arguments

The **parg** program is used only with the `twm` window manager with X-windows. It expands *'@'prompts* in the command line by user Q&A using a popup X-Window dialog widget and then evaluates it with a UNIX `system()` call. This is useful for building scripts where some of the arguments must be supplied by the user at run time. It is also useful for embedding in window manager menus such as `twm(1)` using the `.twmrc` startup file.

If the environment variable `USEPARGTTY` is not defined, it pops up an X-Window form widget for '`@`' fields which need to be supplied.

If `USEPARGTTY` is defined, then it prompts for the argument from terminal associated with the job or `/dev/console` if there is none.

If there are '`-`' characters in the *prompt*, they are changed to spaces prior to displaying the prompt.

If the user specifies '~' in any responses, it is mapped to the `HOME` environment variable if it exists else it is not mapped. If the user specifies '~*user-name*' in any responses, it is mapped to the full path for that user.

Prompt argument syntax

Arguments which need to be prompted are indicated by the following syntax

```
preface@prompt-msg=optional-default-value
```

The optional *default-value* after the '=' is optional and is used in the initial values in the dialog popup. Then the final *value* entered by the userwith the dialog prompt is inserted after the *preface'* to create the string

```
preface value
```

replacing *@prompt-msg=optional-default-value*.

You may concatenate multiple expressions in a single argument if they end in a '`:`' or '`,`' and do not contain any white space as for example

```
-changeParameters:@x1=0,@x2=1023,@y1=0,@y2=1023
```

Invoking parg

Finally, the mapped command line is evaluated by *exec*'ing with '`/bin/sh`'. The UNIX syntax to invoke **parg** is:

```
parg unix-cmd special-arguments-list
```

Execution Path

Execution Path

If we use **parg** from the `.twmrc` filw, **parg** tries to exec the program from the ~/ (users home) directory where `twm` was started. We need to have a "startup-state directory" global state that **parg** can read. This startup-state can be set from a **parg** dialog popup. This is done by saving the new project path in the ~/`parg-target.path` for later use.

```
parg @new-project-path > ~/parg-target.path
```

Then, when **parg** would read this file and do a change working directory if ~/`parg-target.path` exists. One implication is that you can only have one user logged into this account who is setting this path file. This may be a potential problem - but only if running interactive. Normally, only `twm` usually invokes **parg**.

`wmwait`

Process Termination

Normally, a process which is started with an `xterm(1)` will destroy the window when that process completes. If you want the `xterm` window to stay around so that you can scroll through it, use the `-WMWAIT` switch (if it exists in the user process). This runs program **wmwait** which prompts the user to `CLICK TO EXIT` popup when they are all done scrolling and finally wish to exit the **xterm** window. E.g.

```
parg xterm -e pgelrc -wmwait
```

Environment Variables

USEPARGTTY send messages to tty not console tty.

HOME used to expand '~/'.

Bugs

It does not always handle quoted args correctly if they enclose strings containing: "~ , : = " when invoked from `twm` because of interaction with `twm`.

USAGE:

```
          parg [-info] [-usage] [-version]
or
          parg -path @Enter-new-path
or
          parg pgm-to-eval [-switches-with-@-arg-prompts]
```

Type `parg -info` to get more information.

SWITCHES

-Info print more information on parg.

-Path new-path will save path in `$HOME/parg-target.path`.

-Usage print UNIX command level switch usage.

-VErsion print the version of the program.

EXAMPLES OF USAGE

```
parg -path @Enter-new-path
         # Take path response and save it in $HOME/parg-target.path

parg xterm -e cgelp2 -protect -d @PCG-database-file-name
         # Request switches then run cgelp2 in xterm window.

parg xterm -e cgelp2 -protect -d @PCG-database-file-name -wmwait
         # Request switches then run cgelp2 in xterm window.
         # Prompt CLICK TO EXIT popup when all done to allow
         # scrolling in the xterm window.

parg accppx @Left-image @Right-image -graphscale @Optional-switches
         # Request switches then run accppx on two images.

parg cmpgl2 @Gel-accession-nbr -ChangeThr:@Threshold-T1:@Threshold-T2
         # Request switches then run cmpgl2 with multiple args

parg sg2gii @Gel-Acc#=324.1 -ChangeParam:C:@x1=0,@x2,@y1=0,@y2\
                  -@Additional-optional-switches.
         # Request switches then run sg2gii with multiple args

parg Xpix11 @Image-file1 @Image-file2=~/mcrew.ppx -ppx -mark -mouse
         # Request switches then run Xpix11 with multiple args.
```

3.13 pgelrc - Print or define the gel.rc contents

The `gel.rc` file contains GELLAB-II state information required by most of the GELLAB-II programs. See Section 1.6.5 page 61 for additional details.

The **pgelrc** program provides an easy-to-read printout of the contents of the `gel.rc` file required by all GELLAB-II software. If this file does not exist, it will help you create it. The `gel.rc` file points to the current state of the GELLAB system as seen from your working directory. This means that you may have several `gel.rc` files scattered throughout your directory tree (perhaps in different gel projects for example).

If you do not have `gel.rc` in the current project directory you are in, running any GELLAB program will tell you that it is missing and suggest that you create it using **pgelrc**. Running **pgelrc** the first time in a new project directory prompts the user as they define the initial `gel.rc` file. Later, running **pgelrc** in the same directory prints a user-friendly form of `gel.rc`. If you wish to change the `gel.rc` resource file, you should either use **pgelrc -change** or a text editor.

You may also use **pgelrc** to find the current highest accession number and PPX file name as well as the next free accession number and PPX file name with the `-NEXT` switch. You can get a list of that subset of accession numbers corresponding to `ppnp1x` PPX files on the disk which are in the accession file by using the `-AccDir` switch. This puts the list of accession numbers in file `accdir.ccl`. If the `-AccDir:Study` option is specified, then it also appends the corresponding gel study information. These files are useful for further processing of the data using **makjob** *project status*
or determining which gels you wish to investigate which are already in the database. Similarly, the `-PPXnames` generates a file `ppxnam.ccl` of all PPX files listed in the accession file (independent of whether they are on the disk or not). This can be used to restore image files from a backup tape or run **ppxcvt** with the appropriate suboptions specified to `-PPXNAME:`*opt*.

Note that among the parameters which are set up are the pixel resolution in microns, the image size (rows and columns), and the names and positions (in the Rgel) of the up to 25 known proteins, eg *(x,y,pIe,mw,name)*.[16]

With no switches, **pgelrc** pretty-prints an existing `gel.rc` file if it exists. If no *creating* `gel.rc`
`gel.rc` file exists then it will prompt the user for the information necessary and create one in the current directory.

To change where PPX images, PCG DB files or any other files are actually stored, edit `gel.rc` with `pgelrc -change` , or delete it and run `pgelrc` to let GELLAB-II force you to redefine it The first method is to be preferred. Since *changing* `gel.rc`
`gel.rc` is an ASCII file, in addition you can change any of the parameters at any time by editing it with your favorite text editor (eg `vi` or `emacs`).

[16] We currently make no use of "known proteins" in the `gel.rc` file.

changing projects

It should also be noted that there is normally *no* `gel.rc` file in the user's home directory. However, each experiment project sub directory - each with its own database in that account - should have its own `gel.rc` file. You may have multiple databases in the same project directory, but that is normally discouraged since it complicates things. To access a different project, then just change your working directory to that project by doing a `cd` there and your current path will then pick up the `gel.rc` in the project directory.

Default gel.rc values and initial calibration

Because of the large amount of data generated on disks with limited space, it is sometimes desirable to partition the `gellab` subdirectories *.ppx*, the *.gsf, .gcf* files, *.pcg* paged database files etc. to the disk paths different from the that of the project's `gellab` directory. So one would edit the `gel.rc` file with `pgelrc -change` to reflect where the files should be stored.

default database names

Generally, the names of the accession, landmark database and (future) annotation database files (pointed to by your `gel.rc` state file) are called `gel.id`, `lms.lm` and `ann.ann` respectively.

neutral density step wedge

The neutral density (ND) step wedge optical density (OD) values (possibly scanned with the image) are entered during `gel.rc` state definition. Note that if other OD calibrated scan files are used, then *any* set of up to 15 monotonic OD values *covering the range* of the gels on the scanner can be used. The **getacc** program switch) will compute corresponding gray values for the wedge to match these OD values. For example, in the 0-3.0 OD range, such a calibration might be:

```
0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8, 3.0
```

For the 0-2.0 range, one might select 10 values:

```
0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0
```

USAGE:

```
pgelrc [<Optional -switches>]
```

Type `pgelrc -info` to get more information.

SWITCHES

-AccDir:<opt. keyword> Generate list of ACC#s which have existing PPX files. Write the list into file `accnbr.ccl`. If the `STUDY` is added, then append the accession file *study* field in output file. If keyword `ACC#` is specified, then append *:acc#* for each picture file name. If keywords `ADDHEADER, EDITHEADER, PRINTHEADER, REMOVEHEADER, RESETHEADER, SHRINKPPX` or `STD512PPXFILE` is used, then generate `ppxcvt -<option>:ACC# ppxFile` if the *ACC#* is needed or ppxcvt -<option> *ppxFile* if not.

-ChangeParameters `gel.rc` parameters of an existing `gel.rc` file.

-Fields get and print the accession file Data Dictionary fields.

-Info Display general information about this program.

-Next Find last and next free (ACC#, PPX-name).

-PPxNames:<opt.> Generate list of PPX file names which are listed in ACC file. Write the list into file `ppxnam.ccl`. If keyword `ACC#` is specified, then append *:acc#* for each picture file name. If keywords `ADDHEADER, EDITHEADER, PRINTHEADER, REMOVEHEADER, RESETHEADER, SHRINKPPX` or `STD512PPXFILE` is used, then generate `ppxcvt -<option>:ACC# ppxFile` if the *ACC#* is needed or ppxcvt -<option> *ppxFile* if not.

-PRefix:*c* Use the prefix letter *c* instead of `a` when searching with the `-NEXT` or `-ACCDIR` options.

-Study:<Opt field #'s f1,f2,...,fn> Get a listing of specified fields $(f1, f2, ...fn)$ from accession entries.

-Terminal When used with `-ACCDIR` or `-PPXNAMES` options, send output to the terminal instead of to a file.

-Usage Display command line format information.

-Version Print current version number of the program.

-WmWait When done, wait until do CLICK TO EXIT `mwwait` widget to exit.

EXAMPLES OF USAGE

```
pgelrc
       # Pretty print the gel.rc file.

pgelrc -AccDir
       # Create accdir.ccl list of ACC#s on PPX disk.

pgelrc -AccDir:Study
       # Same as above but add corresponding ACC's study.

pgelrc -AccDir -Terminal
       # Create list of ACC#s on PPX disk printed on terminal.

pgelrc -AccDir -Prefix:B
       # Do the search but look for PPX files which start with 'b'.

pgelrc -PPXnames
       # Create ppxnam.ccl list of PPX names of ACC#s in ACC file.
```

```
pgelrc -PPXnames:ACC#
       # Create ppxnam.ccl list of 'PPX names:ACC#' in ACC file.

pgelrc -PPXnames:RESETHEADER
       # Create ppxnam.ccl executable script for ppxcvt -RESETHEADER
       # for 'ppxcvt -RESETHEADER:ACC# ppxFile' in ACC file.

pgelrc -PPXnames -Terminal
       # Create list of PPX names of ACC#s in ACC file printed
       # on terminal.

pgelrc -Next
       # Find highest and next free ACC# and PPX file name.

parg xterm -e pgelrc -WmWait
       # Popup a scrollable xterm running pgelrc and CLICK-TO-EXIT
       # widget.
```

EXAMPLE 1. Create new project. This is done in several steps. First we create the project subdirectory and then run **pgelrc** to create the initial `gel.rc` and `gellab` directory tree files by answering questions about the files and parameters associated with this new project's database files. *new project*

```
30% cd ~/
                    # Go to your home directory.

31% mkdir ts3
                    # Create 'project' subdirectory.

32% cd ~/ts3
                    # Go to this project subdirectory.

33% pgelrc
File: gel.rc does not exist. Creating substate file!
You can backup to the previous question by reponding with '@'.
Gel Accession Number database file
 (/home/joeUser/ts3/gellab/id/gel.id)?: gelts3.id<CR>
Landmark set database file
 (/home/joeUser/ts3/gellab/lms/lms.lm)?: lmsts3.lm<CR>
Annotation database file
 (/home/joeUser/ts3/gellab/ann/ann.ann):annts3.ann<CR>
Input 15 ND wedge values (NULL==std wedge, use 'std.wedge' file,
specific wedges 'AT20X15', 'BT30X10' or 'RTPP'.
Type ? to list these wedges)
 ?: 0.05 0.20 0.35 0.50 0.66 0.80 0.95 1.10 1.25 1.41 1.56 1.72 1.87 2.02 2.17<CR>
Processing non standard wedge.
Representitive gel, i.e. Rgel, (0000.0)?: 0324.1<CR>
Three character Project Prefix [prj]?: uri<CR>
Enter standard proteins - NULL entry to terminate list.
Name of standard protein [1]= <CR>
Scanner name
   (BIOIMAGE1K,CSPI,DATACOPY,MOLDYN1,MOLDYN2,MOLDYN4)
   []?: BIOIMAGE1K
Pixel-size [0] (in microns) ?: 168<CR>
# rows in PPX image [512](in pixels) ?: 1024<CR>
# columns in PPX image [512](in pixels) ?: 1024<CR>
Picture Disk PATH (/home/joeUser/ts3/gellab/ppx/)
 ?: <CR>
Auxiliary Picture Disk PATH
 (/home/joeUser/ts3/gellab/aux/)
 ?: <CR>
Temporary Picture PATH
  (/home/joeUser/ts3/gellab/tmp/)
 ?: <CR>
PCG Database PATH
 (/home/joeUser/ts3/gellab/pcg/)
 ?: <CR>
PCG generated file PATH
 (/home/joeUser/ts3/gellab/gen/)
 ?: <CR>
Original (non-ppx) gel image files PATH
```

```
 (/home/joeUser/ts3/gellab/org/)
 ?: <CR>
SG2GII spot area sizing range [10.00:2000.00] (in pixels**2)?: <CR>
SG2GII spot density sizing range [0.10:500.00] (in integrated OD)?: <CR>
SG2GII spot Optical Density sizing range [0.04:2.70] (in OD)?: <CR>

Creating GELLAB-II 'gellab' directory tree in this directory:
      gellab/ann - annotation database files
      gellab/aux - auxillary derived files (GSF, GCF, mosaic, Rmap images)
      gellab/gen - PCG DB derived files
      gellab/id  - accession database files
      gellab/lms - landmark database files
      gellab/org - original (non-ppx) gel image files
      gellab/pcg - Paged Composite Gel database files
      gellab/ppx - original gel picture files
      gellab/tmp - temporary images and files

Done creating GELLAB-II 'gellab' directory tree.

Creating empty accession file:
 /home/joeUser/ts3/gellab/id/gelts3.id
Creating empty landmark database file:
 /home/joeUser/ts3/gellab/lms/lmsts3.lm

34% pgelrc
PGELRC: Version June 17, 1989
Today's date is 06/18/1989, 09:32:29AM
Written 1980-1989, P. Lemkin.
Print the GELLAB <gel.rc> contents.
Gel Accession file is: /home/joeUser/ts3/gellab/id/gelts3.id
Landmark set data base file: /home/joeUser/ts3/gellab/lms/lmsts3.lm
Annotation database file: /home/joeUser/ts3/gellab/ann/annts3.ann
nd Wedge step[   1]=  0.05nd =   0 gray value
nd Wedge step[   2]=  0.20nd =   0 gray value
nd Wedge step[   3]=  0.35nd =   0 gray value
nd Wedge step[   4]=  0.50nd =   0 gray value
nd Wedge step[   5]=  0.66nd =   0 gray value
nd Wedge step[   6]=  0.80nd =   0 gray value
nd Wedge step[   7]=  0.95nd =   0 gray value
nd Wedge step[   8]=  1.10nd =   0 gray value
nd Wedge step[   9]=  1.25nd =   0 gray value
nd Wedge step[  10]=  1.41nd =   0 gray value
nd Wedge step[  11]=  1.56nd =   0 gray value
nd Wedge step[  12]=  1.72nd =   0 gray value
nd Wedge step[  13]=  1.87nd =   0 gray value
nd Wedge step[  14]=  2.02nd =   0 gray value
nd Wedge step[  15]=  2.17nd =   0 gray value
Project prefix: uri
Pixel-size (in microns) is 168
# rows in PPX file (in pixels) is 1024
# columns in PPX file (in pixels) is 1024
Representitive gel: 0324.1
Picture disk directory PATH is: /home/joeUser/ts3/gellab/ppx/
```

```
Auxillary Picture disk PATH is: /home/joeUser/ts3/gellab/aux/
Temporary Picture disk PATH is: /home/joeUser/ts3/gellab/tmp/
PCG Database PATH is: /home/joeUser/ts3/gellab/pcg/
PCG generated file PATH is: /home/joeUser/ts3/gellab/gen/
Original gel image files PATH is: /home/joeUser/ts3/gellab/org/
SG2GII spot area sizing range [10:2000]
SG2GII spot density sizing range [0.1:500.0]
SG2GII spot density range sizing range [0.04:2.70]
That's all folks!
```

EXAMPLE 2. Pretty-print existing `gel.rc` file. *print state*

```
36% cat -n gel.rc
     1 # gel.rc - gel state file 03/06/1992, 04:42:55PM
     2 Xpix.switches:
     3 Xpix11.switches:
     4 accppx.switches:
   . . .
    26 ppxcvt.switches:
    27 ppxodt.switches:
    28 sg2gii.switches: -3x3 -BUSSE:3:C -SAT:99.7 -BACK:64 -RESTOFGEL \
           -CCMIN:4 -DRAWSPOTS:PO -CH:A:25,10000,D:0.001,10000,O:0.001,4.5
    29 tek2psG.switches:
    30 wmwait.switches:
    31 #
    32 ##### End of Program Switches ######
    33 #
    34 gelFile=/home/joeUser/gellab/id/gel.id # Accession database file
    36 lmsFile=/home/joeUser/gellab/lms/lms.lm # Landmark set DB file
    37 annDBfile=/home/joeUser/gellab/ann/ann.ann #  Annotation DB file
    38 ndWedgeCal=1 0.050000 0  # [step#, OD-value, gray-value]
    39 ndWedgeCal=2 0.200000 0  # [step#, OD-value, gray-value]
    40 ndWedgeCal=3 0.350000 0  # [step#, OD-value, gray-value]
    41 ndWedgeCal=4 0.500000 0  # [step#, OD-value, gray-value]
    42 ndWedgeCal=5 0.660000 0  # [step#, OD-value, gray-value]
    43 ndWedgeCal=6 0.800000 0  # [step#, OD-value, gray-value]
    44 ndWedgeCal=7 0.950000 0  # [step#, OD-value, gray-value]
    45 ndWedgeCal=8 1.100000 0  # [step#, OD-value, gray-value]
    46 ndWedgeCal=9 1.250000 0  # [step#, OD-value, gray-value]
    47 ndWedgeCal=10 1.410000 0  # [step#, OD-value, gray-value]
    48 ndWedgeCal=11 1.560000 0  # [step#, OD-value, gray-value]
    49 ndWedgeCal=12 1.720000 0  # [step#, OD-value, gray-value]
    50 ndWedgeCal=13 1.870000 0  # [step#, OD-value, gray-value]
    51 ndWedgeCal=14 2.020000 0  # [step#, OD-value, gray-value]
    52 ndWedgeCal=15 2.170000 0  # [step#, OD-value, gray-value]
    55 projectPrefix=uri # 3 character project prefix
    56 Rgel=324.1  # Representative gel
    57 PixelSizeMicrons=168 # Pixel size in microns
    58 PpxNrows=512 # PPX image size in pixels
    59 PpxNcols=512 # PPX image size in pixels
    60 ppnP1X=/home/joeUser/gellab/ppx/ # Picture disk PATH
    61 ppnP2X=/home/joeUser/gellab/aux/ # Auxiliary Picture disk PATH
```

```
62 ppnP3X=/home/joeUser/gellab/tmp/ # Temporary Picture disk PATH
63 ppnP4X=/home/joeUser/gellab/pcg/ # PCG Database PATH
64 ppnP5X=/home/joeUser/gellab/gen/ # PCG generated file PATH
65  ppnP6X=/home/joeUser//gellab/org/       # Original (non-ppx) gel image files PATH
66 SG2areaLimits=10.000000 5000.000000  # SG2 area sizing in pixels
67 SG2totDensLimits=0.100000 10000.000000  # SG2 total density sizing in SUM OD
68 SG2odRange=0.040000 4.000000  # SG2 OD range sizing in OD
```

```
37% pgelrc
PGELRC V-1.3.35 - Version December 20, 1991
Today's date is 03/06/1992, 04:45:42PM
Written 1980-1991, P. Lemkin.
Printing the GELLAB <gel.rc> contents.
Gel Accession file is: /home/joeUser/gellab/id/gel.id
Working gel ACC# is: 0000.0
Landmark set data base file: /home/joeUser/gellab/lms/lms.lm
Annotation data base file: /home/joeUser/gellab/ann/ann.ann
ND Wedge step[   1]=  0.05OD =   0 gray value
ND Wedge step[   2]=  0.20OD =   0 gray value
ND Wedge step[   3]=  0.35OD =   0 gray value
ND Wedge step[   4]=  0.50OD =   0 gray value
ND Wedge step[   5]=  0.66OD =   0 gray value
ND Wedge step[   6]=  0.80OD =   0 gray value
ND Wedge step[   7]=  0.95OD =   0 gray value
ND Wedge step[   8]=  1.10OD =   0 gray value
ND Wedge step[   9]=  1.25OD =   0 gray value
ND Wedge step[  10]=  1.41OD =   0 gray value
ND Wedge step[  11]=  1.56OD =   0 gray value
ND Wedge step[  12]=  1.72OD =   0 gray value
ND Wedge step[  13]=  1.87OD =   0 gray value
ND Wedge step[  14]=  2.02OD =   0 gray value
ND Wedge step[  15]=  2.17OD =   0 gray value
Window [0:511,0:511]
Project prefix: uri
Representitive gel: 0324.1
Pixel-size (in microns) is 0
# rows in PPX file (in pixels) is 512
# columns in PPX file (in pixels) is 512
Picture disk directory PATH is: /home/joeUser/gellab/ppx/
Auxillary Picture disk PATH is: /home/joeUser/gellab/aux/
Temporary Picture disk PATH is: /home/joeUser/gellab/tmp/
PCG Database PATH is: /home/joeUser/gellab/pcg/
PCG generated file PATH is: /home/joeUser/gellab/gen/
SG2GII spot area sizing range [10:5000](pixels**2)
SG2GII integrated spot density sizing range [0.10:10000.00](OD)
SG2GII spot Optical Density extrema sizing range [0.040:4.00](OD)
That's all folks!
```

Note that you can set the temporary disk path to the be the same as the auxiliary disk path. In fact, all paths *could* be set to the same directory. Normally, however,

they are split up as shown.

EXAMPLE 3. Find next free image file and accession numbers.

find free gel acc# & file

```
31% pgelrc -nextFreeEntries
PGELRC: Version March 14, 1989
Today's date is 03/17/1989, 05:12:39PM
Written 1980-1989, P. Lemkin.

Analyzing Accession File /home/joeUser/gellab/id/gelpq1.id
for next free ACC# and PPX name.
Searching accession file /home/joeUser/gellab/id/gelqp1.id for ACC#s
Found 425 ACC# entries.
Highest PPXname is A01697 with associated ACC# 0326.4
Highest accession file ACC# is 0525.1 with PPX file name A01689

Next free ACC# is 0526.1
Next free PPX file name A01698

Analyzing Current PPXP1X Directory /home/joeUser/gellab/ppx/
Last PPNP1X file is A01697 with ACC# 0393.2
That's all folks!
```

EXAMPLE 4. generate file `accdir.ccl` listing all accession numbers for pictures which already exist on the disk. The original gel PPX file prefix is set to `b` instead of the default `a`.

what gels exist?

```
37% pgelrc -accDir -Prefix:b
PGELRC: Version March 18, 1989
Today's date is 03/19/1989, 12:42:11AM
Written 1980-1989, P. Lemkin.
Searching accession file /home/joeUser/gellab/id/gelqp1.id for ACC#s
Generated directory file accdir.ccl of existing 12 PPX files.
That's all folks!


38% cat -n accdir.ccl
     1 0324.1
     2 0369.1
     3 0378.2
     4 0384.1
     5 0396.1
     6 0497.1
     7 0503.1
     8 0511.1
     9 0514.1
    10 0515.1
    11 0517.1
    12 0393.2


39% pgelrc -accDir:Study -Prefix:b -Terminal
0324.1 HEME MALIG-AML,MYELOID
0369.1 HEME MALIG-ALL,LYMPHOID
0378.2 HEME MALIG-CLL,LYMPHOID (DUPL. SCAN)
0384.1 HEME MALIG-HCL,LYMPHOID
0396.1 HL-60 HUMAN MYELOID DIFFERENTIATION
0497.1 HEME MALIG-AML MYELOID
0503.1 HEME MALIG-AML MYELOID
0511.1 HEME MALIG-ALL,LYMPHOID
0514.1 HEME MALIG-ALL,LYMPHOID
0515.1 HEME MALIG-CLL,LYMPHOID
0517.1 HEME MALIG-CLL,LYMPHOID
0393.2 HEME MALIG-HCL,LYMPHOID
```

EXAMPLE 5. Generate a file `ppxnam.ccl` containing a list of all picture file names in the accession file.

what PPX files exist?

```
50% pgelrc -PPXnames
PGELRC: Version March 14, 1989
Today's date is 03/17/1989, 05:19:29PM
Written 1980-1989, P. Lemkin.
Searching accession file /home/joeUser/gellab/id/gelqp1.id for ACC#s
```

```
Generated ppxnam.ccl of 12 PPX files listed in ACC file.
That's all folks!


51% cat -n ppxnam.ccl
     1 b00661.ppx
     2 b00889.ppx
     3 b00945.ppx
     4 b00981.ppx
     5 b01045.ppx
     6 b01577.ppx
     7 b01601.ppx
     8 b01633.ppx
     9 b01645.ppx
    10 b01649.ppx
    11 b01657.ppx
    12 b01693.ppx

52% pgelrc -PPXnames -Terminal
b00661.ppx
b00889.ppx
b00945.ppx
b00981.ppx
b01045.ppx
b01577.ppx
b01601.ppx
b01633.ppx
b01645.ppx
b01649.ppx
b01657.ppx
b01693.ppx
```

EXAMPLE 6. Change parameters in `gel.rc` file for current project `abc`.

changing `gel.rc` *parameters*

```
50% cd ~/abc

51% pgelrc -changeparameters
PGELRC: Version July 1, 1989
Today's date is 07/31/1989, 03:26:52PM
Written 1980-1989, P. Lemkin.
You can backup to the previous question by reponding with '@'.
Gel Accession Number database file
 (/home/joeUser/abc/gellab/id/gel.id)
 ?: gelprj.id
Landmark set database file
 (/home/joeUser/abc/gellab/lms/lms.lm)
 ?:
Annotation database file
 (/home/joeUser/abc/gellab/ann/ann.ann)
 ?:
Using Standard Wedge:
```

```
 0.05 0.20 0.35 0.50 0.66 0.80 0.95 1.10 1.25 1.41 1.56 1.72 1.87 2.02 2.17
Input 15 ND wedge values (NULL==std wedge, use 'std.wedge' file,
specific wedges 'AT20X15', 'BT30X10' or 'RTPP'.
Type ? to list these wedges)
 ?:
Using Standard Wedge:
 0.05 0.20 0.35 0.50 0.66 0.80 0.95 1.10 1.25 1.41 1.56 1.72 1.87 2.02 2.17
Representitive gel, i.e. Rgel, (0000.0)?:
Three character Project Prefix [prj]?: uri <CR>
Enter standard proteins - NULL entry to terminate list.
Name of standard protein [1]?:
Pixel-size [168] (in microns) ?: 250
# rows in PPX image [1024](in pixels) ?:
# columns in PPX image [1024](in pixels) ?:
Picture Disk PATH
 (/home/joeUser/abc/gellab/ppx/)
 ?:
Auxiliary Picture Disk PATH
 (/home/joeUser/abc/gellab/aux/)
 ?:
Temporary Picture PATH
 (/home/joeUser/abc/gellab/tmp/)
 ?:
PCG Database PATH
 (/home/joeUser/abc/gellab/pcg/)
 ?:
PCG generated file PATH
 (/home/joeUser/abc/gellab/gen/)
 ?:
SG2GII spot area sizing range [10.00:2000.00](pixels**2)?:
SG2GII spot integrated density sizing range [0.30:500.00](OD)?:
SG2GII spot Optical Density sizing range [0.03:2.70](OD)?:

Printing the GELLAB <gel.rc> contents.
Gel Accession file is: /home/joeUser/abc/gellab/id/gelela.id
Landmark set data base file: /home/joeUser/abc/gellab/lms/lms.lm
Annotation data base file: /home/joeUser/abc/gellab/spot/annspt.ann
ND Wedge step[   1]=  0.05OD =  27 gray value
ND Wedge step[   2]=  0.20OD =  49 gray value
ND Wedge step[   3]=  0.35OD =  72 gray value
ND Wedge step[   4]=  0.50OD =  95 gray value
ND Wedge step[   5]=  0.66OD = 117 gray value
ND Wedge step[   6]=  0.80OD = 136 gray value
ND Wedge step[   7]=  0.95OD = 153 gray value
ND Wedge step[   8]=  1.10OD = 168 gray value
ND Wedge step[   9]=  1.25OD = 181 gray value
ND Wedge step[  10]=  1.41OD = 192 gray value
ND Wedge step[  11]=  1.56OD = 200 gray value
ND Wedge step[  12]=  1.72OD = 208 gray value
ND Wedge step[  13]=  1.87OD = 213 gray value
ND Wedge step[  14]=  2.02OD = 220 gray value
ND Wedge step[  15]=  2.17OD = 225 gray value
Pixel-size (in microns) is 250
```

```
# rows in PPX file (in pixels) is 1024
# columns in PPX file (in pixels) is 1024
Representitive gel: uri
Picture disk directory PATH is: /home/joeUser/abc/gellab/ppx/
Auxillary Picture disk PATH is: /home/joeUser/abc/gellab/aux/
Temporary Picture disk PATH is: /home/joeUser/abc/gellab/tmp/
PCG Database PATH is: /home/joeUser/abc/gellab/pcg/
PCG generated file PATH is: /home/joeUser/abc/gellab/gen/
Original gel image files PATH is: /home/joeUser/abc/gellab/gen/
SG2GII spot area sizing range [10:2000](pixels**2)
SG2GII integrated spot density sizing range [0.3:500.0](OD)
SG2GII spot Optical Density extrema sizing range [0.03:2.70](OD)
That's all folks!
```

3.14 plotn - Plot Universal Graphics Files (*.ugf*)

Program **plotn** is used to re-plot a (*.ugf*) file for a X-Windows or 4010-display terminal, or to create a *.tek, .ppx, .ps* file. The display is created using the `-DISPLAY:<type>` switch. The default display is `XWND` for X11 X-Windows.

If the `-DISPLAY:XWND` is specified, and you are under X11, it will popup a plot window. Press `Exit` to exit, `Print` to print the plot on your laser printer via `$LASERPRINTER` environment varible, and `Refresh` to refresh the plot window. Note that you may resize the plot window with the X-Windows window manager and the plot will be redrawn.

The `-DISPLAY:4010` display emulates a Tektronix 4010 display by sending the Tektronix codes to the user terminal. The `-DISPLAY:VT240` display emulates a Tektronix 4010 display in which case it puts the VT240 into and takes it out of 4010 mode.

If the display is `LASER`, then print the plot on the laser printer using the **tek2psG** program and the laser printer indicated by the environment variable `$LASERPRINTER`.

Multiple *.ugf* files may be displayed by specifying multiple input file names. A range of UGF files by *n1-n2* (eg. `123-137`) would specify `000123.ugf-000137.ugf`.

When printing on the laser printer when you specify the `-Display:laser` option, it recursively invokes the following shell command pipeline (say we are trying to print file `000123.ugf`):

```
plotn -display:4010  000123.ugf | tek2psG | lpr -h -P$LASERPRINTER
```

where `$LASERPRINTER` is typically defined by `setenv LASERPRINTER laser`.

USAGE:

```
plotn [<opt. -Switches>] <list of .ugf files>
```

Type `plotn -info` to get more information.

SWITCHES

-Color allow RGB plot, (single color plotting is `-NOCOLOR` default).

-DEbug:bits print debugging information information.

-DIsplay:<4010,VT240,TEK,XWND,PPX,PS,LASER> select display (4010 is the default).

-Info print more information on plotn.

-Usage print UNIX command level switch usage.

-VErsion print the version of the program.

-WAit after plotting, wait for `DONE` to be typed to erase the display (`-NOWAIT` default).

-WMwait When done, wait until do CLICK TO EXIT `mwwait` widget to exit.

EXAMPLES OF USAGE

```
plotn -Display:4010 000123.ugf
      # Display on Tektronix-4010 terminal.

plotn 000123.ugf
      # Default is display on Tektronix-4010 terminal.

plotn -Display:tek 000123.ugf
      # Create .tek (4010 code) file.

plotn -Display:ps 000123.ugf
      # Create PostScript .ps file.

plotn -Display:ppx 000123.ugf
      # Create PPX image file.

plotn *.ugf
      # Display all .ugf files on Tektronix-4010 terminal.

plotn -Display:ps *.ugf
      # Convert all .ugf files to PostScript files.

plotn 000012 -Display:4010 | tek2psG | lpr -Plaser
      # Convert to Tektronix 4010 which is then piped
      # into a tek to PostScript converter which is then
      # piped into a PostScript speaking laser printer
      # which is called 'Plaser' on our system.

plotn 000012 -Display:LASER
      # Does the same as the above command. The default
      # UNIX command to print a PostScript file is given
      # in the $LASERPRINTER environment variable.
```

3.15 ppx2ps - convert .ppx image file to PostScript .ps

Program **ppx2ps** converts a PPX image file to PostScript for printing on a PostScript printer. If no input file is specified, the UNIX "standard input" is used. The resulting PostScript file may be directed to the Unix "standard output" or to a named file.

The switches for PPX2PS provide some flexibility as the PostScript file is generated. For example, you can zoom in on a particular point in the image with the `-ZOOM`:nX,x,y switch. Also, you can invert the image with the `-BLACK` switch. See the SWITCHES section below to get a description of all the switches. Any case independent switch may be negated by preceeding it with a `NO` (eg. `-NOINFO`).

There are two important things to be aware of when using **ppx2ps**. First, because PostScript only allows images with a maximum depth of 8-bits, only 8-bit PPX images can be translated at this time.

8-bit images only

Second, since the PPX binary image data is converted to Postscript ascii hex image data, the file size will typically double. Since image file sizes are generally large to begin with, this could cause a problem with limited memory PostScript printers. It might also fill the printer's spool directory when multiple files are spooled (stored locally before being sent to the printer). To get around this, you may zoom into a particular area of an image with `-ZOOM`:nX,x,y (which converts only the area of interest to PostScript) or you may sample the image with `-SAMPLE`:n which will typically decrease the output file size by a factor of n^2.

This program was derived from the Public Domain Utah Raster Toolkit program *rlelaser(1)* by Rod Bogart and John W. Peterson.

Figure 3.22 shows the a typical gel image printed on a laser printer after converting the PPX file to Postscript.

Figure 3.22. Sample **ppx2ps** converted PPX file printed on a laser printer.

USAGE:

```
ppx2ps [<Opt. PPX input file>] [<opt. -Switches>]
```

Type `ppx2ps -info` to get more information.

SWITCHES

-Aspect:*n* Aspect ratio *n* of image (default 1.0)

-Black define black as 0, white is 255. Default: black is 255, white is 0.

-CEnter:*n* center output about *n* inches.

-Fullsize use full size of the image (default).

-HEIght:*n* Image height *n* in inches (default 8", width is found via aspect and height).

-Hflip Flip the image about the Y axis.

-Info print more information on **ppx2ps**.

-OutPSfile:*f* specify output file *f*, otherwise stdout is used.

-SAmple:*n* sample image by *n*. For example, *n*=2 maps a 512x512 image to 256x256 (assuming there is no zooming).

-SIlent don't be verbose (default)

-Title:*msg* display title under bottom of image.

-Usage print UNIX command level switch usage.

-VErsion print the version of the program.

-VFlip flip the image about the X axis.

-WmWait When done, wait until do CLICK TO EXIT `mwwait` widget to exit.

-Zoom:*nX,x,y* zoom image with magnification *nX* centered at pixel position *x,y* from upper left hand corner.

EXAMPLES OF USAGE

```
ppx2ps a00661.ppx > a00661.ps
      # Convert a00661.ppx to PostScript file.

ppx2ps a00661.ppx -sample:2 -out:a00661.ps
      # Will do the same thing as above, but resulting
      # PostScript file will be 1/4 the size.

ppx2ps a01234.ppx -zoom:4X,256,256 | lpr -Plaser
      # Convert the PPX file to Postscript format with
      # a magnification factor of 4 centered at pixel (256,256)
      # from upper left hand corner (ULHC) and send it directly
      # to a printer.

cat a01234.ppx | ppx2ps -zoom:4X,256,256 | lpr -Plaser
      # This does the same thing as above but demostrates
      # PPX2PS reading the PPX file from UNIX 'standard input
```

3.16 ppxcvt - Convert external image to .ppx files

Program **ppxcvt**[17] is an multi-purpose image manipulation program used to convert arbitrary sized image files into PPX format images for GELLAB-II. **Ppxcvt** currently assumes that the files to be converted are uncompressed. If not, use the UNIX *uncompress(1)* utility on each of the compressed files prior to running **ppxcvt**.

scanner files

Special Scanner Specific Switches

Because of the wide variety of input image formats from different scanners, there are a large number of command line switches that may be required to describe and correctly convert an input image file. To simplify using **ppxcvt** with popular image formats, special switches are available to preset other switches which would be required. Special switches include: `-BIOIMAGE`, `-CSPI`, `-ELSIE`, `-MOLECULARDYNAMICS` and `-TRUVAL`. Many of the other switches discussed in the following sections are automatically set using one of the above special switches. The use of these special switches instead of explicitly setting the other switches is strongly encouraged.

3.16.1 Image Conversion Model

conversion model

The best way to think of how images are converted is with the conversion model shown in Figure 3.23. The model consists of two images: an input image and an output image, each with its own subregion. **Ppxcvt** performs a number of operations to transform the input image into a PPX format output image. These operations will be described in detail in Section 3.16.5

Image geometry consists of an image width and height as well as a subregion within that image. The subregion is defined by an offset from the upper left hand corner of the image as well as the size (width and height) of the subregion. Note that all positions and sizes in the images are specified in pixels (picture elements) rather than cm. or inches.

input image geometry

The input image size is normally specified by the input file itself if that information is available in the file header or related files. BioImage has a default 1Kx1K input image file size. The default size for the Elsie images is approximately 1Kx1K and the Elsie file `geldata` associated with each gel image is read to determine its exact size. TIFF images (including Molecular Dynamics, CSPI, Truval, etc.) have the image size specified in the TIFF file header.

[17] Since the new **getacc11a** program can optionally downsample and make the stnadard **s** image, the **ppxcvt** program is not longer always required for GELLAB-II operation. But it is available for manual image conversions.

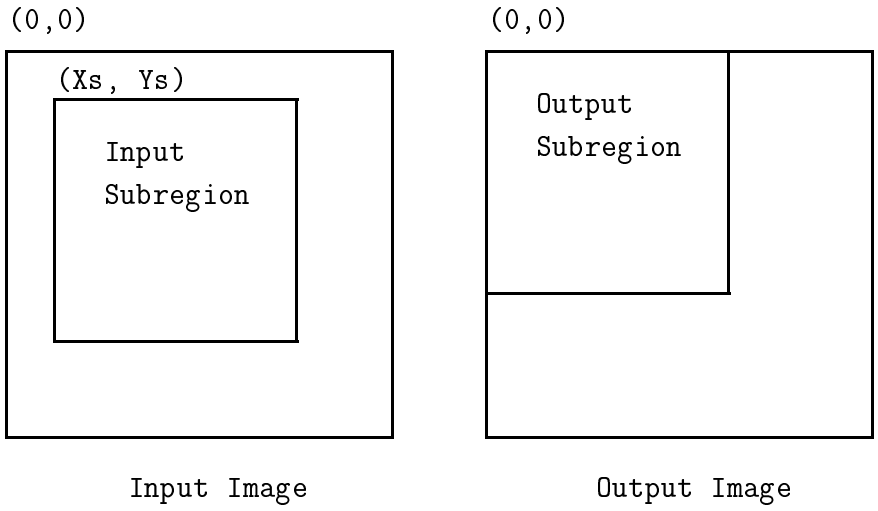


Figure 3.23. Image Conversion Model used by **ppxcvt**. The model consists of an input and output image, each with an associated subregion and pixel "depth" (bits/pixel reflecting its grayscale dynamic range). The input subregion's origin is at (X_s, Y_s) relative to the image origin while the output subregion's origin is at (0, 0). Data is mapped from the input subregion to the output subregion and from there to the output PPX file.

If input image size information isn't available from the input file header, an associated file or a special switch, then it must be explicitly specified by the command line switch `-INSIZE`:*w,h* where *w* and *h* define the input image's width and height in pixels. The subregion within the input image defaults to the full input image unless the command line switch `-SUBSIZE`:*w,h,x0,y0* is used to specify the width, height and origin of the input sub-region within the input image.

output image geometry

The output image size is specified by the `PpxNrows` and `PpxNcols` variables in the GELLAB-II `gel.rc` file or may be overridden from the command line with the `-OUTSIZE`:*w,h* switch. Additionally, the `-1Kx1K`, `-2Kx2K` and `-4Kx4K` switches force a specific output image size rather than using the more cumbersome `-OUTSIZE`:*w,h* switch. The output image subregion is always positioned at (0,0) in the upper left hand corner of the image.

image sampling

Often, the size of the input image subregion is larger than the size of the specified output image. When this occurs, the input image needs to be sampled. Every *n*'th pixel from every *n*'th line is used and the rest discarded where *n* refers to the sampling step. The output image subregion size is calculated to be the *largest* that will fit in the output image taking the input image subregion size and sampling into account. If the input subregion size is greater than the output image size, then the input subregion will be sampled so that it will fit into the output image's specified

width and height while maintaining the same aspect ratio (ratio of width to height) as the input subregion.

Image Bytes/Pixel And Bits/Pixel

The input image pixel depth is defined by the number of bits/pixel and the number of bytes/pixel. These are normally gotten from the input picture file header (the case with TIFF files). Otherwise, the special switches `-BIOIMAGE`, `-CSPI`, `-ELSIE`, `-TRUVAL`, `-MOLECULARDYNAMICS` set them accordingly. Alternatively, the `-BYTESIN`:*n* and `-BITSIN`:*b* may be used to explicitly define them for non-standard input files. The default is 8 bits/pixel and 1 byte/pixel.

The switch `-BITSOUT`:*b* may be used to explicitly define the number of bits per sample for PPX output files (the default is 8). The the number of bytes per sample in the output image is calculated from this. Images with a depth of 8 bits or less will have one byte/pixel; images with a depth between 9 and 16 bits will have two bytes/pixel.

3.16.2 Location of Image Files

Unlike other GELLAB-II programs that require input files to be in the project directory tree `gellab` in the current directory, **ppxcvt** assumes that the input images can be specified by an absolute or relative path name. If a relative path name is specified and the input file cannot be found, then the project directory tree `gellab` in the current directory (if one exists) is searched to locate the input file. Output *.ppx* files are *always* saved in `ppnP1X` as specified in the `gel.rc` file.

In addition, **ppxcvt** can convert many input image files at the same time with the `-DIRTREE`:*directory* switch. If the `-DIRTREE`:*directory* is specified with one of `-ELSIE` or `-BIOIMAGE`, the input picture files are located in the Elsie or BioImage directory tree specified by *directory*. **Ppxcvt** will search through that directory tree looking for the image files corresponding to the specified special switch.

special scanners

When the `-DIRTREE`:*directory* switch is used without the `-ELSIE` or `-BIOIMAGE` switches, then **ppxcvt** just looks in *directory* for input files to convert to PPX format. See example page 436.

3.16.3 Image File Names

When single files are converted, the output file name is the same as the input file name, but the file extension is *.ppx*. For example, `leukemia.gel` gets mapped to `leukemia.ppx`.

mapping file names

You can map the input image file names to standard output image names used by GELLAB-II. This is done using the `-MAPNAME`:*pxxxxx* switch where *xxxxx* is a 5 digit number and *p* is either the letter '`a`' or '`b`' (GELLAB-II standard gel input

image prefixes). The `-NEXTFREEPPXFILE` switch searches the accession file for the next free PPX file name *pxxxxx* and uses it to force `-MAPFILENAME:`*pxxxxx*.

automatically finding next free file name

If file name mapping is requested and multiple input images are being converted, then sequential numbers will be generated (and printed as they are converted). For example, if *xxxxx* is `00123`, then the first output image name is `a00123.ppx`, the second will be `a00124.ppx`, etc.

When converting BioImage or Elsie images, all gel image files in their directory trees have the same names (`gel` with BioImage, `geldata` with Elsie). Therefore, we need to map these file names to unique sequential PPX names since all output PPX files are saved in the same directory. This is done by having the `-BIOIMAGE` and `-ELSIE` switches force the `-NEXTFREEPPXFILE` switch. However, if the `-MAPFILENAME:`*Pxxxxx* switch is used instead, then it uses the starting picture file name *Pxxxxx*.

Ascii Input Image Files

The input image file may consist of a stream of ASCII encoded pixels. These may be either hex, decimal or octal. If they are hex encoded (`-HEX` switch), then two hex characters are required for each 8-bit pixel. ASCII pixel values specified by the `-DECIMAL` or `-OCTAL` switches will scan numbers in the input file delimited by any non-numeric character. In all three cases, whitespace (spaces, tabs, FF, CR or LF) are ignored.

Input Image File Name Extensions

If you have input files with *.hex*, *.dec*, *.oct*, or *.mdy* file extensions, then **ppxcvt** will default to `-HEX`, `-DECIMAL`, `-OCTAL`, or `-MOLECULARDYNAMICS`, respectively.

3.16.4 The Standard PPX Image

GELLAB-II requires that all gels have a 512x512 size image which is used by **markgel**, **mosaic**, and other programs. If the output gel image converted by **ppxcvt** is 512x512, then the requirement is satisfied. For larger output image sizes, we need an additional 'standard' PPX file which is automatically generated. This file has the same PPX number as the a*xxxxx*`.ppx` file, but has a '`s`' prefix instead of the '`a`'. Once the gel '`a`' images are segmented, they can be removed from the disk to save space as the '`s`' images are used when gel images are needed for further processing.

The `-STD512PPXFILE` switch forces the creation of an extra 512x512 PPX file with a '`s`' prefix. Normally, this switch is not required since the '`s`' image is automatically generated. The `-NOSTD512PPXFILE` option, will prevent the standard image from being created; however, this is not normally used.

3.16.5 Operations on the Input Image Subregion

The input image being converted can be modified by a number of operations as it is transformed into a PPX format output image. These operations work either on individual pixels within the input image subregion or on the subregion as a whole. The collection of operations is referred to as the image pipeline. Since the composition of these operations is not symmetric, the order in which they occur is important. The pipeline operations occur in the same order as they occur in the following descriptions. If any pipeline switches are enabled, they are performed in this pre-set order - otherwise, they do not occur.

order of pipeline operations

Pipeline Operations at the Pixel Level

For the following pipeline operations we refer to the current pixel gray value as g, the transformed pixel gray value as g' and the maximum gray value of a pixel as *Gmax*.

If the `-SIGNEDPIPE`:b switch is used, it indicates that an input pixel is to be interpreted as a b bit 2's complement (signed) integer and will be scaled to an positive (unsigned) integer.

If the `-SCALEPIPE`:m,b switch is specified, both m and b are signed floating point numbers and

$$g' = mg + b\,.$$

If the `-CPMPIPE`:*maxCPM* switch is specified, the data is transformed by

$$g' = \frac{\mathrm{Gmax}}{\mathrm{maxCPM}}\, g\,.$$

If the `-LOGPIPE` switch is set it is transformed again by

$$g' = \frac{\mathrm{Gmax} * \log(g + 1.0)}{\log(\mathrm{Gmax})}\,.$$

If the `-COMPLEMENTPIPE` switch is set

$$g' = (\mathrm{Gmax} - g)\,.$$

Pipeline Operations at the Image Level

The following pipeline switches operate on the image subregion as a whole after the pixel level pipeline operations have completed. However, they do not modify the image geometry; the subregion's width, height and origin within the image is unchanged.

If the `-HFLIPPIPE` switch is set, the subregion is flipped horizontally. The left and right sides of the subregion are exchanged.

If the `-VFLIPPIPE` switch is set, the subregion is flipped vertically. The top and bottom sides of the subregion are exchanged.

If the `-ROTATEPIPE:`*degrees* is set, the subregion is rotated *degrees*° counter-clockwise. Note that *degrees* can only be one of `90`, `180` or `270`.

3.16.6 OD (Optical Density) Calibration

The gray scale to OD calibration is available for certain types of input images. In all cases, this is mapped to a piecewise linear ND wedge calibration and saved in the PPX output image file header. These include:

- in the header of the Bioimage file,
- in the Elsie file `scanlines` associated with each image data file `gel`,
- precalibrated in 0:4.095 OD in the 12-bit data of the Molecular Dynamics image data as 0:2.56 OD *or* scaled to a subset of 0:4.095 OD,
- precalibrated in 0:2.55 OD in the 8-bit data of the Truval image data,
- CSPI ...

The ND wedge is specified in the `gel.rc` state file and need not correspond to an actual scanned ND wedge. This allows us to encode an image's known grayscale-to-OD calibration in terms of this pseudo wedge. The calibration consists of up to 15 two-tuples of the form (OD value, gray value); each ND wedge OD value has a corresponding gray value. The set of these values can be used to synthesize a piecewise-linear grayscale-to-OD lookup table used by GELLAB-II in mapping grayscale pixels values to OD.

In the case of `-MOLECULARDYNAMICS`, there are 12-bits of OD in the range of [0:4.095], so we need to map the data to 8-bits as used by GELLAB-II. To do this we need to know the maximum OD in the input image. The `-DYNAMICRANGE:`*ODmax* switch specifies this and defaults to a *ODmax* of 2.55 OD. If the switch is not used, the mapping of 12-bits to 8-bits is as follows. Let:

$$\begin{aligned} g_8 &= \text{the 8 bit resluting grayvalue,} \\ g_{12} &= \text{the 12 bit input grayvalue,} \\ \text{ODmaxint} &= 1000 * \text{ODmax}. \end{aligned}$$

Then, if ODmax ≤ 2.55:

$$g_8 = \begin{cases} 255 & \text{for } g_{12} > 2550 \\ g_{12}/10 & \text{for } g_{12} \leq 2550, \end{cases}$$

otherwise,

$$g_8 = \begin{cases} 255 & \text{for } g_{12} > \text{ODmaxint} \\ (g_{12} * 255)/\text{ODmaxint} & \text{for } g_{12} \leq \text{ODmaxint} \end{cases}$$

3.16.7 Special PPX File Operations

There are several PPX header editing options for manipulating existing PPX files rather than converting an input file to a PPX file. These must be typed in their entirety since they might potentially destroy the PPX file: `-ADDHEADER`, `-EDITHEADER`, `-REMOVEHEADER` and `-RESETHEADER`. The `-PRINTHEADER` option just lists the current PPX file header. These operations will only work on one file specified on the command line. These are useful for making minor changes in an existing header. No image file conversion is performed, other than manipulating the PPX file header.

PPX header editing

USAGE:

```
ppxcvt <input-image> [<opt. -Switches>]
```

Type `ppxcvt -info` to get more information on **ppxcvt**.

SWITCHES

-ADDHEADER:<opt. ACC#> insert 512 byte PPX header into front of existing *.ppx* file. If the *ACC#* is specified, use its accession file ND and CW calibration.

-BIOimage supply defaults to read Bioimage 1KX1K gel image.

-BITSIn:b bits/pixel b bits/input-pixel. (Default $b = 12$ for `-BYTES:2` and $b = 8$ for `-BYTES:1`).

-BITSOut:b bits/pixel b bits/output-pixel.

-BYTESIn:n bytes/pixel (1 or 2) (default is 1).

-BYTESOut:n bytes/pixel (1 or 2) (default is 1).

-COMPLementPipe invert grayscale values in image i.e. black$\Longrightarrow$white.

-COMPRess derived images after they are created on the disk to save space.

-CPMpipe:maxCPM grayscale $g' = (255 * g/maxCPM)$ (default $g' = g$).

-CSpi supply defaults to read CSPI image.

-DEBug:0nnn Set the debug level to octal value *0nnn*.

-DEcimal input file is ascii decimal encoded pixels.

-DIrectoryTree:path specify directory of gel images to be converted.

-DYnamicRange:ODmax specify maximum OD of pixel data for Molecular Dynamics files.

-EDITHEADER edit 512 byte PPX header in existing .ppx file.

-ELsie supply defaults to read Elsie 1Kx1K gel image.

-HEAder:n skip n byte header at front of file.

-HEX input file is ascii hex encoded pixels.

-HFlip flip image about horizontal axis.

-Info Display general information about this program.

-INSize:w,h width and height of entire input image.

-LogPipe compute grayscale $g' = (255 * \log(g+1))/\log(256)$ (default $g' = g$).

-MApFileName:pxxxxx map PPX input file names to sequential output file names of the form *pxxxxx*`.ppx` where *xxxxx* is the first decimal number (eg. `a00001.ppx`). This switch is typically used with Molecular dynamics, Elsie or BioImage files.

-MOleculardynamics supply defaults to read Molecular Dynamics scanner gel image. Dynamic rangs is 0 to 2.56 OD. Use the `-DYNAMICRANGE` switch to change this to up to 4.095.

-NExtFreePPXfile search accession file for next free PPX file number *xxxxx* and use it to force `-MAPFILENAME:a`*xxxxx*. This defaults for `-BIOIMAGE` or `-ELSIE`.

-OCtal input file is ascii octal encoded pixels.

-OUTsize:w,h width and height of entire output image.

-PRintHeader print 512 byte PPX header in existing *.ppx* file.

-Quiet Don't print run-time messages during execution.

-REMOVEHEADER remove the 512 byte PPX header in existing .ppx file.

-RESETHEADER:<opt. ACC#> reset 512 byte PPX header in existing *.ppx* file. If the *ACC#* is specified, use its ND and CW calibration.

-REVerseBytes reverse hex characters bytes in input pixel.

-ROtatePipe:<degrees> rotate the image by -<degrees> (counter clockwise). Only the values 90, 180, 270 are allowed.

-SCalePipe:m,b linearly scale each pixel g by multiplying by m and adding b.

-SIGnedPipe:b interpret pixels as b bit 2's complement (signed) integers and scale them to positive (unsigned) integers.

-STd512PPXfile create an extra 512x512 size PPX file with a `s` file name prefix if the output image is not 512x512 in size (default).

-SUBsize:w,h,x0,y0 Define a *w*idth by *h*eight subregion located at (*x0,y0*) in the input image.

-Truval Supply defaults to read Truval scanner Tiff image file. Dynamic range is 0 to 2.56 OD.

-Usage Display command line format information.

-VErsion Print current version number of the program.

-VFlipPipe flip image about vertical axis.

-WmWait When done, wait until do CLICK TO EXIT `mwwait` widget to exit.

-1Kx1K create a 1Kx1K pixel PPX file (default is 1024x1024).

-2Kx2K create a 2Kx2K pixel PPX file (default is 2048x2048).

-4Kx4K create a 4Kx4K pixel PPX file (default is 4096x4096).

EXAMPLES OF USAGE

Molecular Dynamics Examples

```
ppxcvt mdgel.gel -moleculardynamics
      # Convert Molecular Dynamics scanned file using TIFF header
      # information into a PPX output file. The output image size
      # is specified in the gel.rc file by the PpxNrows and
      # PpxNcols variables (presumably 1024x1024).
      # Keep data in 0.0 to 2.55 (default) OD dynamic range
      # and synthesize a pseudo ND wedge in the PPX file
      # containing the OD calibration.
      # Make a standard PPX 's' image (by default).

ppxcvt mdgel.gel -moleculardynamics -dynamicRange:3.7 -1kx1k
      # Convert Molecular Dynamics scanned file using TIFF header
      # information into a 1024x1024 PPX output file.
      # Same as above, but keeps data in 0.0 to 3.7 OD dynamic range.

ppxcvt -dirtree:mdgel -moleculardynamics -dynamicRange:3.7 -2kx2k
      # Same as above but convert all the files in the directory
      # 'mdgel' into PPX files with image sizes of 2048x2048.
```

BioImage Examples

```
ppxcvt -bioimage -dirTree:proj2001 -mapName:a00101 -1kx1k
      # Convert all the image files in a Bioimage directory tree
      # mapping the output PPX image names to a00101, a00102, etc.
      # The PPX files are 1024x1024.  A standard PPX 's' image
      # file is created with each 1024x1024 PPX image file.
      # Also, extract ND wedge calibration from input images
      # and save as a pseudo ND wedge calibrations in the PPX
      # file headers.
```

Elsie Examples

```
ppxcvt -elsie -dirTree:proj2010 -mapName:a00201 -1kx1k
      # Same as BioImage example above, but for a Elsie
      # directory tree mapping the gel images to a00201,
      # a00202, etc. The output PPX images are 1024x1024.
      # Also, make a standard 512x512 PPX 's'
      # image with each larger image (by default).
      # Extract the CPM (OD) calibration or default a
      # linear silver stain gel calibration and save
      # as a pseudo ND wedge in each PPX file header.

ppxcvt -elsie -dirTree:proj2010 -1kx1k
```

```
      # Same as above, but uses -NextFreePpx to get the mapped
      # file name since this is the default when -Elsie or
      # -BioImage is used without -MapName.
```

BioGen Examples

```
ppxcvt betaGel.193 -header:256 -insize:144,144 -Bytes:2 -CPM:4000 -log
      # Skip past the 256 byte header and extract
      # 144x144 2-byte pixels from file 'betaGel.193' into
      # upper left hand corner of PPX image. Scale the CPM
      # data by 4000 before taking the log.
```

Signed Image Examples

```
ppxcvt rnadd.dat -header:3642 -insize:200,200 -BytesIn:2 -BitsIn:12
                 -Signed:12 -ReverseByteOrder -Complement
      # Skip past 3642 byte header of a 200x200 image.
      # The pixels in the input image are 2 bytes deep of which
      # only 12 bits are valid.  The 2 bytes in each pixel are swapped
      # as each 12-bit pixel is treated as a signed integer
      # and converted to a 12-bit unsigned integer.  The image
      # is then complemented before being written as a PPX
      # file.
```

Ascii Image Examples

```
ppxcvt a01234.hex -hex -insize:512x512
      # Convert a 512x512 hex input file with no header.

ppxcvt a01234.hex -insize:512x512
      # Default to -HEX (because of filename suffix)
      # and convert as in previous example.

ppxcvt a02010.hex -insize:512x512 -1kx1k
      # Same as above, but the PPX output file is 1024x1024.

ppxcvt a01234.data -decimal -insize:512x512
      # Convert a 512x512 decimal text input file with no header.

ppxcvt a01235.hex -hex -complement -1kx1k
      # Convert a 1024x1024 hex input file
```

Other Misc. Conversion Examples

```
ppxcvt -dirtree:imageDir -outsize:512x512
      # Convert all the image files in the directory 'imageDir'
      # into 512x512 PPX images.  Each input image should
      # have necessary input size information in its header.

ppxcvt gel.1k -header:590 -complement -insize:1026,1024  \
       -subsize:1024,1024,2,0 -outsize:512,512
      # Skip past the the header of gel image gel.1k which is
      # 590 bytes.  Copy a 1024x1024 subregion with origin at
      # (2,0) from within the input image.  Sample it down
      # to fit in a 512x512 image and complement it before
      # saving it as a PPX output image. The PPX file is 512x512.
      # No 's' image is created since output image is already
      # 512x512.

ppxcvt gel.1k -header:590 -complement -insize:1026,1024  \
       -subsize:1024,1024,2,0 -1kx1k
      # Same as above, but save as a 1kx1k image. Also make
      # a standard 512x512 image.
```

PPX File Header Manipulation Examples

```
ppxcvt a01235.ppx -printHeader
      # Print header of existing PPX file.

ppxcvt a01235.ppx -ADDHEADER
      # Add PPX header to PPX file which does NOT HAVE THE HEADER!.

ppxcvt a01235.ppx -RESETHEADER
      # Reset header of existing PPX file to default values.

ppxcvt a02010.ppx -RESETHEADER -insize:1024,1024
      # Same as above, but set image size to 1024X1024 in header.

ppxcvt a01235.ppx -EDITHEADER
      # Edit header of existing PPX file.
```

convert BioImage data

EXAMPLE 1. Create a GELLAB project directory called 'expt' and read a set of BioImage picture files from a "tar" tape. Convert them to 1Kx1K GELLAB-II PPX files and read their ND wedge calibration to create the PPX ND wedge calibration saved in the PPX header file by generating corresponding grayvalues as if the wedge existed. The PPX file created is (1KX1K) pixels. Users who already have the BioImage data on their disk can skip to command line 30 to see how to convert the data.

```
18% pwd
/home/joeUser

19% mkdir expt

20% cd expt

21% mkdir data

22% cd data

23% tar xvf /dev/rst8
x epl_340.1/gel, 1051214 bytes, 2054 tape blocks
x epl_340.1/bound, 23184 bytes, 46 tape blocks
x epl_340.1/bitmap, 131104 bytes, 257 tape blocks
x epl_340.1/list, 34578 bytes, 68 tape blocks
x epl_341.1/gel, 1051214 bytes, 2054 tape blocks
x epl_341.1/bitmap, 131104 bytes, 257 tape blocks
x epl_341.1/list, 42586 bytes, 84 tape blocks
x epl_343.1/gel, 1051214 bytes, 2054 tape blocks
x epl_343.1/bitmap, 131104 bytes, 257 tape blocks
x epl_343.1/list, 30002 bytes, 59 tape blocks
x epl_345.1/gel, 1051214 bytes, 2054 tape blocks

24% cd ..

25% pwd
/home/joeUser/expt

26% ls
data

27% ls data
epl_340.1 epl_341.1 epl_343.1 epl_345.1

28% pgelrc
PGELRC V-1.3.33 - Version January 24, 1992
Today's date is 01/23/1992, 02:55:26PM
Written 1980-1992, P. Lemkin.
File gel.rc does not exist. Creating substate file!
You can backup to the previous question by reponding with '@'.
Gel Accession Number database file
 (/home/joeUser/expt/gellab/id/gel.id)
 ?: gelepl.id
Landmark set database file
 (/home/joeUser/expt/gellab/lms/lms.lm)
 ?: lmsepl.lm
Spot list annotation database file
 (/home/joeUser/expt/gellab/ann/ann.ann)
 ?:annepl.ann
Input 15 ND wedge values (NULL==std wedge, use 'std.wedge' file,
specific wedges 'AT20X15', 'BT30X10' or 'RTPP'.
Type ? to list these wedges)
```

```
 ?:
Using Standard Wedge:
 0.05 0.20 0.35 0.50 0.66 0.80 0.95 1.10 1.25 1.41 1.56 1.72 1.87 2.02 2.17
Representative gel, i.e. Rgel, (0000.0)?:
Enter standard proteins - NULL entry to terminate list.
Name of standard protein [1]?:
Pixel-size [0] (in microns) ?: 162
# rows in PPX image [512](in pixels) ?: 1024
# columns in PPX image [512](in pixels) ?: 1024
Picture Disk PATH
 (/home/joeUser/expt/gellab/ppx/)
 ?:
Auxiliary Picture Disk PATH
 (/home/joeUser/expt/gellab/aux/)
 ?:
Temporary Picture PATH
 (/home/joeUser/expt/gellab/tmp/)
 ?:
PCG Database PATH
 (/home/joeUser/expt/gellab/pcg/)
 ?:
PCG generated file PATH
 (/home/joeUser/expt/gellab/gen/)
 ?:
SG2GII spot area sizing range [10.00:5000.00](pixels**2)?:
SG2GII spot integrated density sizing range [0.10:10000.00](OD)?:
SG2GII spot Optical Density sizing range [0.04:4.00](OD)?:

Creating GELLAB-II directory tree in this directory:
    gellab/ann - annotation database files
    gellab/aux - auxiliary derived files (GSF, GCF, mosaic, Rmap images)
    gellab/gen - PCG DB derived files
    gellab/id  - accession database files
    gellab/lms - landmark database files
    gellab/pcg - Paged Composite Gel database files
    gellab/ppx - original gel picture files
    gellab/tmp - temporary images and files


Done creating GELLAB-II 'gellab' directory tree.
Creating empty accession file:
 /home/joeUser/expt/gellab/id/gelepl.id
Creating empty landmark database file:
 /home/joeUser/expt/gellab/lms/lmsepl.lm

29% ls
data    gel.rc  gellab

30% ppxcvt -bioimage -dirtree:data -mapName:a02001
PPXCVT: Version January 16, 1992
Today's date is 01/23/1992, 03:07:56PM
User: /home/joeUser/gellab/src/ppxcvt/
Written 1981-1992, P. Lemkin.
```

```
Estimated nd Wedge Step[ 1] = 0.05 =   1 gray value
Estimated nd Wedge Step[ 2] = 0.20 =  29 gray value
Estimated nd Wedge Step[ 3] = 0.35 =  93 gray value
Estimated nd Wedge Step[ 4] = 0.50 = 141 gray value
Estimated nd Wedge Step[ 5] = 0.66 = 180 gray value
Estimated nd Wedge Step[ 6] = 0.80 = 200 gray value
Estimated nd Wedge Step[ 7] = 0.95 = 216 gray value
Estimated nd Wedge Step[ 8] = 1.10 = 229 gray value
Estimated nd Wedge Step[ 9] = 1.25 = 237 gray value
Estimated nd Wedge Step[10] = 1.41 = 243 gray value
Estimated nd Wedge Step[11] = 1.56 = 247 gray value
Estimated nd Wedge Step[12] = 1.72 = 250 gray value
Estimated nd Wedge Step[13] = 1.87 = 252 gray value
Estimated nd Wedge Step[14] = 2.02 = 254 gray value
Estimated nd Wedge Step[15] = 2.17 =   0 gray value

Converting Gel #1
/home/joeUser/expt/data/epl_340.1/gel
to
/home/joeUser/expt/gellab/ppx/a02001.ppx
Creating standard (512x512) PPX file:
  /home/joeUser/expt/gellab/ppx/s02001.ppx

Estimated nd Wedge Step[ 1] = 0.05 =   1 gray value
Estimated nd Wedge Step[ 2] = 0.20 =  29 gray value
Estimated nd Wedge Step[ 3] = 0.35 =  93 gray value
Estimated nd Wedge Step[ 4] = 0.50 = 141 gray value
Estimated nd Wedge Step[ 5] = 0.66 = 180 gray value
Estimated nd Wedge Step[ 6] = 0.80 = 200 gray value
Estimated nd Wedge Step[ 7] = 0.95 = 216 gray value
Estimated nd Wedge Step[ 8] = 1.10 = 229 gray value
Estimated nd Wedge Step[ 9] = 1.25 = 237 gray value
Estimated nd Wedge Step[10] = 1.41 = 243 gray value
Estimated nd Wedge Step[11] = 1.56 = 247 gray value
Estimated nd Wedge Step[12] = 1.72 = 250 gray value
Estimated nd Wedge Step[13] = 1.87 = 252 gray value
Estimated nd Wedge Step[14] = 2.02 = 254 gray value
Estimated nd Wedge Step[15] = 2.17 =   0 gray value

Converting Gel #2
/home/joeUser/expt/data/epl_341.1/gel
to
/home/joeUser/expt/gellab/ppx/a02002.ppx
Creating standard (512x512) PPX file:
  /home/joeUser/bioimage/gellab/ppx/s02002.ppx

Estimated nd Wedge Step[ 1] = 0.05 =   1 gray value
Estimated nd Wedge Step[ 2] = 0.20 =  29 gray value
Estimated nd Wedge Step[ 3] = 0.35 =  93 gray value
Estimated nd Wedge Step[ 4] = 0.50 = 141 gray value
Estimated nd Wedge Step[ 5] = 0.66 = 180 gray value
Estimated nd Wedge Step[ 6] = 0.80 = 200 gray value
```

```
Estimated nd Wedge Step[ 7] = 0.95 = 216 gray value
Estimated nd Wedge Step[ 8] = 1.10 = 229 gray value
Estimated nd Wedge Step[ 9] = 1.25 = 237 gray value
Estimated nd Wedge Step[10] = 1.41 = 243 gray value
Estimated nd Wedge Step[11] = 1.56 = 247 gray value
Estimated nd Wedge Step[12] = 1.72 = 250 gray value
Estimated nd Wedge Step[13] = 1.87 = 252 gray value
Estimated nd Wedge Step[14] = 2.02 = 254 gray value
Estimated nd Wedge Step[15] = 2.17 =   0 gray value

Converting Gel #3
/home/joeUser/expt/data/epl_343.1/gel
to
/home/joeUser/expt/gellab/ppx/a02003.ppx
Creating standard (512x512) PPX file:
  /home/joeUser/bioimage/gellab/ppx/s02003.ppx

Estimated nd Wedge Step[ 1] = 0.05 =   1 gray value
Estimated nd Wedge Step[ 2] = 0.20 =  29 gray value
Estimated nd Wedge Step[ 3] = 0.35 =  93 gray value
Estimated nd Wedge Step[ 4] = 0.50 = 141 gray value
Estimated nd Wedge Step[ 5] = 0.66 = 180 gray value
Estimated nd Wedge Step[ 6] = 0.80 = 200 gray value
Estimated nd Wedge Step[ 7] = 0.95 = 216 gray value
Estimated nd Wedge Step[ 8] = 1.10 = 229 gray value
Estimated nd Wedge Step[ 9] = 1.25 = 237 gray value
Estimated nd Wedge Step[10] = 1.41 = 243 gray value
Estimated nd Wedge Step[11] = 1.56 = 247 gray value
Estimated nd Wedge Step[12] = 1.72 = 250 gray value
Estimated nd Wedge Step[13] = 1.87 = 252 gray value
Estimated nd Wedge Step[14] = 2.02 = 254 gray value
Estimated nd Wedge Step[15] = 2.17 =   0 gray value

Converting Gel #4
/home/joeUser/expt/data/epl_345.1/gel
to
/home/joeUser/expt/gellab/ppx/a02004.ppx
Creating standard (512x512) PPX file:
  /home/joeUser/expt/gellab/ppx/s02004.ppx
Real TIME =00:01:19  CPU TIME =00:00:43,  54.43%

32% ls -l gellab/ppx
total 5248
-rw-------  1 joeUser    1049088 Jan 23 15:07 a02001.ppx
-rw-------  1 joeUser    1049088 Jan 23 15:07 a02002.ppx
-rw-------  1 joeUser    1049088 Jan 23 15:08 a02003.ppx
-rw-------  1 joeUser    1049088 Jan 23 15:08 a02004.ppx
-rw-------  1 joeUser     262656 Jan 23 15:07 s02001.ppx
-rw-------  1 joeUser     262656 Jan 23 15:07 s02002.ppx
-rw-------  1 joeUser     262656 Jan 23 15:08 s02003.ppx
-rw-------  1 joeUser     262656 Jan 23 15:08 s02004.ppx
```

EXAMPLE 2. Convert Molecular Dynamics 300A gel TIFF format image. Create the PPX ND wedge calibration saved in the PPX header file by generating corresponding grayvalues as if the wedge existed of the dyanmic range (0 to 2.8 OD) specified. The PPX file created is (1KX1K) pixels.

convert Molecular Dynamics data

```
34% ppxcvt -molecularDynamics -dynamicRange:2.8 -1kx1k a01234.gel
PPXCVT: Version January 16, 1992
Today's date is 01/23/1992, 03:54:24PM
User: /home/joeUser/expt
Written 1981-1992, P. Lemkin.

Estimated nd Wedge Step[ 1] = 0.05 =   4 gray value
Estimated nd Wedge Step[ 2] = 0.20 =  18 gray value
Estimated nd Wedge Step[ 3] = 0.35 =  31 gray value
Estimated nd Wedge Step[ 4] = 0.50 =  45 gray value
Estimated nd Wedge Step[ 5] = 0.66 =  60 gray value
Estimated nd Wedge Step[ 6] = 0.80 =  73 gray value
Estimated nd Wedge Step[ 7] = 0.95 =  86 gray value
Estimated nd Wedge Step[ 8] = 1.10 = 100 gray value
Estimated nd Wedge Step[ 9] = 1.25 = 114 gray value
Estimated nd Wedge Step[10] = 1.41 = 128 gray value
Estimated nd Wedge Step[11] = 1.56 = 142 gray value
Estimated nd Wedge Step[12] = 1.72 = 157 gray value
Estimated nd Wedge Step[13] = 1.87 = 170 gray value
Estimated nd Wedge Step[14] = 2.02 = 184 gray value
Estimated nd Wedge Step[15] = 2.17 = 198 gray value

Converting Gel #1
  a01234.gel
to
  gellab/ppx/a01234.ppx
Created standard (512x512) PPX file:
  gellab/ppx/s01234.ppx
Real TIME =00:00:42  CPU TIME =00:00:38,  90.48%
```

EXAMPLE 3. Convert a directory of Molecular Dynamics files.

convert directory of images

```
ppxcvt2 -moldyn -dirtree:tiff-data
PPXCVT: Version January 16, 1992
Today's date is 01/23/1992, 04:07:22PM
User: /home/joeUser/expt
Written 1981-1992, P. Lemkin.

Estimated nd Wedge Step[ 1]= 0.05=   5 gray value
Estimated nd Wedge Step[ 2]= 0.20=  20 gray value
Estimated nd Wedge Step[ 3]= 0.35=  34 gray value
Estimated nd Wedge Step[ 4]= 0.50=  50 gray value
Estimated nd Wedge Step[ 5]= 0.66=  66 gray value
Estimated nd Wedge Step[ 6]= 0.80=  80 gray value
Estimated nd Wedge Step[ 7]= 0.95=  94 gray value
```

```
Estimated nd Wedge Step[ 8]= 1.10= 110 gray value
Estimated nd Wedge Step[ 9]= 1.25= 125 gray value
Estimated nd Wedge Step[10]= 1.41= 140 gray value
Estimated nd Wedge Step[11]= 1.56= 155 gray value
Estimated nd Wedge Step[12]= 1.72= 172 gray value
Estimated nd Wedge Step[13]= 1.87= 187 gray value
Estimated nd Wedge Step[14]= 2.02= 201 gray value
Estimated nd Wedge Step[15]= 2.17= 217 gray value

Converting Gel #1
/home/joeUser/expt/tiff-data/b03243.gel
to
/home/joeUser/expt/gellab/ppx/b03243.ppx
Created standard (512x512) PPX file:
  /home/joeUser/expt/gellab/ppx/s03243.ppx

Estimated nd Wedge Step[ 1]= 0.05=   5 gray value
Estimated nd Wedge Step[ 2]= 0.20=  20 gray value
Estimated nd Wedge Step[ 3]= 0.35=  34 gray value
Estimated nd Wedge Step[ 4]= 0.50=  50 gray value
Estimated nd Wedge Step[ 5]= 0.66=  66 gray value
Estimated nd Wedge Step[ 6]= 0.80=  80 gray value
Estimated nd Wedge Step[ 7]= 0.95=  94 gray value
Estimated nd Wedge Step[ 8]= 1.10= 110 gray value
Estimated nd Wedge Step[ 9]= 1.25= 125 gray value
Estimated nd Wedge Step[10]= 1.41= 140 gray value
Estimated nd Wedge Step[11]= 1.56= 155 gray value
Estimated nd Wedge Step[12]= 1.72= 172 gray value
Estimated nd Wedge Step[13]= 1.87= 187 gray value
Estimated nd Wedge Step[14]= 2.02= 201 gray value
Estimated nd Wedge Step[15]= 2.17= 217 gray value

Converting Gel #2
/home/joeUser/expt/tiff-data/b03244.gel
to
/home/joeUser/expt/gellab/ppx/b03244.ppx
Created standard (512x512) PPX file:
  /home/joeUser/expt/gellab/ppx/s03244.ppx

Estimated nd Wedge Step[ 1]= 0.05=   5 gray value
Estimated nd Wedge Step[ 2]= 0.20=  20 gray value
Estimated nd Wedge Step[ 3]= 0.35=  34 gray value
Estimated nd Wedge Step[ 4]= 0.50=  50 gray value
Estimated nd Wedge Step[ 5]= 0.66=  66 gray value
Estimated nd Wedge Step[ 6]= 0.80=  80 gray value
Estimated nd Wedge Step[ 7]= 0.95=  94 gray value
Estimated nd Wedge Step[ 8]= 1.10= 110 gray value
Estimated nd Wedge Step[ 9]= 1.25= 125 gray value
Estimated nd Wedge Step[10]= 1.41= 140 gray value
Estimated nd Wedge Step[11]= 1.56= 155 gray value
Estimated nd Wedge Step[12]= 1.72= 172 gray value
Estimated nd Wedge Step[13]= 1.87= 187 gray value
Estimated nd Wedge Step[14]= 2.02= 201 gray value
```

```
Estimated nd Wedge Step[15]= 2.17= 217 gray value

Converting Gel #3
/home/joeUser/expt/tiff-data/b03253.gel
to
/home/joeUser/expt/gellab/ppx/b03253.ppx
Created standard (512x512) PPX file:
  /home/joeUser/expt/gellab/ppx/s03253.ppx
Real TIME =00:00:56  CPU TIME =00:00:26,  46.43%
```

EXAMPLE 4. Convert 1026x1024 image with 590 byte header and 1024 pixels per line where we ignore first 2 pixels per line.

specify conversion sizes

```
42% ppxcvt gel.1k -header:590 -complement insize:1026,1024  \
          -subsize:1024,1024,2,0 -outsize:512,512
PPXCVT: Version January 23, 1992
Today's date is 01/23/1992, 04:28:47PM
User: /home/joeUser/expt
Written 1981-1992, P. Lemkin.

Converting Gel #1
gel.1k
to
gellab/ppx/gel.ppx
Real TIME =00:00:02  CPU TIME =00:00:01,  50.00%
```

EXAMPLE 5. Convert ASCII hex encoded gel image (512X512).

convert ASCII data

Hex data consists of a sequential string of 8-bit bytes ASCII Hex character encoded (eg. AA AB AA ...). The *.hex* input file is a byte stream of 8-bit bytes coded in hex (i.e. 2 hex characters (each coding 4-bits) to represent the 8-bit byte). The number of hex coded pixels/line is arbitrary. This means that if line width portability is a problem, you can have short lines (i.e < 80) by just inserting a linefeed wherever needed. Note: since hex is less efficient than binary, expect these files to be larger. To store them when they are not being used, consider using the UNIX *compress(1)* utility program to save space (See page C).

```
61% more a02099.hex
E8E8E7E5E8E8E8E7E7E6E6E8E8E4E6E8E6E4E6E7E4E3E4E3
E9E7E2E7E5E5E7E5E6E6E5E5E5E6E4E6E5E5E3E5E5E1E2E2
E2E3E4E3E2E3E1E3E4E1E1E2E1E1E2E1E1E3E2E4E1E3DEE5
E3E1E2E4DDDEDFDDDFDDDCDEDFDEE0E0DEDCDFDEDEDFDBDD
DFDFDBE1DCDCE0DDDDE0DBDAD9D9D9D8D9DBD7DBDAD9DBD7
D9DAD7DAD9DAD9DCD8D8DBD9D7D9D9D7D8D3D2D2D7D1D4D4
```

```
62% ppxcvt -hex a02099.hex
PPXCVT: Version January 23, 1992
Today's date is 01/23/1992, 04:37:33PM
User: /home/joeUser/expt
Written 1981-1992, P. Lemkin.

Converting Gel #1
a02099.hex
to
gellab/ppx/a02099.ppx
Real TIME =00:00:04  CPU TIME =00:00:02,  50.00%
```

print PPX header

EXAMPLE 6. Print header of an existing PPX file.

```
72% ppxcvt -Printheader gel.ppx
PPXCVT: Version January 23, 1992
Today's date is 01/23/1992, 04:40:03PM
User: /home/joeUser/expt
Written 1982-1992, P. Lemkin.
The old header is:
fversn=35
nWedgeSteps=24
wedgeVal[0]=0.050  grayCalWedge[0]=0
wedgeVal[1]=0.199  grayCalWedge[1]=32
wedgeVal[2]=0.350  grayCalWedge[2]=94
wedgeVal[3]=0.500  grayCalWedge[3]=138
wedgeVal[4]=0.659  grayCalWedge[4]=172
wedgeVal[5]=0.800  grayCalWedge[5]=197
wedgeVal[6]=0.949  grayCalWedge[6]=213
wedgeVal[7]=1.100  grayCalWedge[7]=227
wedgeVal[8]=1.250  grayCalWedge[8]=235
wedgeVal[9]=1.409  grayCalWedge[9]=242
wedgeVal[10]=1.560  grayCalWedge[10]=247
wedgeVal[11]=1.720  grayCalWedge[11]=250
wedgeVal[12]=1.869  grayCalWedge[12]=252
wedgeVal[13]=2.020  grayCalWedge[13]=253
wedgeVal[14]=2.170  grayCalWedge[14]=253
nrows=512
ncols=512
bitpp=8
bytpp=1
nbands=1
odmn=0
odmx=255
filtyp=1
name=gel.ppx
rix(cwx1)=0
riy(cwy1)=0
cwx2=511
cwy2=511
```

```
x0=0
y0=0
isptsz=0
istpx=0
istpy=0
domain=2
cMapFlag=0
imOrientation=0X9
imEncode=0
initl=PFL
```

EXAMPLE 7. Edit PPX header of an existing PPX file. *edit PPX header*

```
62% ppxcvt -EDITHEADER gel.ppx
PPXCVT: Version January 23, 1992
Today's date is /23/1992, 04:40:03PM
User: /home/joeUser/expt
Written 1982-1992, P. Lemkin.
You will edit a PPX header in front of .ppx file.
Are you sure(YES/NO)[No]?: yes
fversn=[35]?:
nWedgeSteps=[24]?: 15<CR>        <---- Note change.
wedgeVal[0] [0.050]?:
wedgeVal[1] [0.199]?:
wedgeVal[2] [0.350]?:
wedgeVal[3] [0.500]?:
wedgeVal[4] [0.659]?:
wedgeVal[5] [0.800]?:
wedgeVal[6] [0.949]?:
wedgeVal[7] [1.100]?:
wedgeVal[8] [1.250]?:
wedgeVal[9] [1.409]?:
wedgeVal[10] [1.560]?:
wedgeVal[11] [1.720]?:
wedgeVal[12] [1.869]?:
wedgeVal[13] [2.020]?:
wedgeVal[14] [2.170]?:
grayCalWedge[0] [0]?:
grayCalWedge[1] [32]?:
grayCalWedge[2] [94]?:
grayCalWedge[3] [138]?:
grayCalWedge[4] [172]?:
grayCalWedge[5] [197]?:
grayCalWedge[6] [213]?:
grayCalWedge[7] [227]?:
grayCalWedge[8] [235]?:
grayCalWedge[9] [242]?:
grayCalWedge[10] [247]?:
grayCalWedge[11] [250]?:
grayCalWedge[12] [252]?:
```

```
grayCalWedge[13] [253]?:
grayCalWedge[14] [253]?:
nrows=[512]?:
ncols=[512]?:
bitpp=[8]?:
bytpp=[1]?:
nbands=[1]?:
odmn=[0]?:
odmx=[255]?:
filtyp=[1]?:
name=gel.ppx

rix(cwx1)=[0]?:
riy(cwy1)=[0]?:
cwx2=[511]?:
cwy2=[511]?:
x0=[0]?:
y0=[0]?:
isptsz=[0]?:
istpx=[0]?:
istpy=[0]?:
domain=[2]?:
cMapFlag=[0]?:
imOrientation=0X[9]?:
imEncode=[0]?:
initl=[PFL]?:
```

3.17 ppxodt - Picture debugger for checking image data

Program **ppxodt** is a picture debugger for GELLAB-II Portable PiXture (PPX) image files. It provides debugging functions for easy access to the numeric contents of *.ppx* picture files. It is available to aid in the debugging of picture output from any of the GELLAB programs by detailed examination of final or intermediate images which may have been generated. Its list of commands and sample output are illustrated below. It enables the user to interactively examine and change individual picture elements (pixels) of an image on the user's disk area. It requests a picture file to look at and then requests an (x, y) position which may be changed on subsequent commands to **ppxodt**. That is both the (x, y) value can be changed or the pixel value *at* that position can be changed. It looks for the PPX image file first in the current directory and then in `ppxp1x`, `ppxp2x` and `ppxp3x` paths.

numeric
PPX
windows

USAGE:

```
ppxodt[Opt. .ppx file name] [<Opt. -switches>]
```

Type `ppxodt -info` to get more information.

SWITCHES

-Info Display general information about this program.

-Usage Display command line format information.

-Version Print current version number of the program.

-WmWait When done, wait until do CLICK TO EXIT `mwwait` widget to exit.

EXAMPLES OF USAGE

```
ppxodt b00661.ppx
       # Start ppxodt with specific file.

ppxodt b00661
       # Start ppxodt with specific file, but don't bother
       # with the file extension.

ppxodt
       # Start ppxodt with no specific file.
```

EXAMPLE 1. look at windows around pixel coordinates (195,250) for three related image files `b00661` (original image), `c00661` (propagated central core spot image), and `z00661` (segmented spot image).

```
39% ppxodt b00661.ppx
PPXODT : Version May 30, 1989
Written 1980-1989, P. Lemkin.
Edit user image file: b00661.ppx

Commands
--------
<number> <CR> replace old contents.
/ reopen current location.
+ or R or ^F open (right) pixel address.
       . . .
H or ? print this list of commands
BYE  exit back to operating system

 x,y picture address: 195,250
/ 26  w

WINDMP [ 186: 203, 240: 260]
ppxodt
       186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203
       --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- ---
 240 :  16   9  10  13  13   8  10  10   7  13   9   9   7   9   9   8  13  12
 241 :  10  14   6   6  11  11   8  11  10  15  13  12   9  11  11  10  10  13
 242 :   6  11  13   8   9  12   9  10  12  11   7   9  10  13  11   8  12  17
 243 :  10  13  16  12  13  13   9  11   4  15  15  14  14  11  12  13  12   6
 244 :  18  15  18  18  15  12  13  16  15  20  15  15  13  19  17  14  17  12
 245 :  16  23  27  26  27  27  21  18  24  30  23  22  23  19  19  18  16  16
 246 :  28  37  41  42  42  31  33  32  32  34  34  33  32  24  25  18  14  14
 247 :  40  49  58  57  46  42  42  43  40  43  42  38  32  25  19  14  16  13
 248 :  37  55  52  51  49  44  33  45  38  41  39  32  29  27  16  11  20  14
 249 :  31  40  38  36  40  37  35  29  28  31  29  30  23  24  20  20  15  23
 250 :  21  28  29  30  23  27  24  25  23  26  22  21  21  16  16  16  18  21
 251 :  19  24  24  21  20  21  23  23  19  20  21  17  16  12  10  16  19  11
 252 :  15  26  16  23  21  19  24  22  18  24  24  18  13  13  10  13  13  17
 253 :  16  18  14  14  17  17  18  20  20  19  20  14   9   9  15   8  93  12
 254 :  13  10  13  19  15  21  16  21  23  23  22  16  11   8   9   7   5  12
 255 :   8  11   8   8  15  16  16  16  16  18  12  12  12  12  16   4  10  10
 256 :   7  14  15  10  14  15  12  20  18  16  15  17  12  13  14  12  12  12
 257 :  11  16  11  12  11  11  13  14  14  14  16  17  23  14  16  18  22  17
 258 :  13  15   8   8   8  11  17  19  10  13  19  21  23  28  25  27  30  26
 259 :  10  15   7  12  18  18  16  21  19  20  29  26  34  48  54  51  44  37
 260 :   7  12  14  12  14  20  19  21  23  34  33  39  49  68  74  72  60  42


/ 26   x,y picture address:
/ 26  bye


40% ppxodt c00661.ppx
```

```
PPXODT : Version May 30, 1989
Written 1980-1989, P. Lemkin.
Edit user image file: c00661.ppx

Commands
--------
<number> <CR> replace old contents.
/ reopen current location.
+ or R or ^F open (right) pixel address.
- or L or ^B open previous (left) pixel address.
^P open pixel address up 1 line.
^N open pixel address down 1 line.
N print 3x3 neighborhood of previous (x,y) address.
W print 18x20 pixel window centered at previous (x,y) address.
C print 18x20 pixel window with ULH corner at previous (x,y) address.
D toggle print out mode to decimal (default)
O toggle print out mode to octal
X toggle print out mode to hexadecimal
E exit saving changed image.
P toggle the Write-enable protection of the image (initially protected)
H or ? print this list of commands
BYE  exit back to operating system

 x,y picture address: 195,250
/ 105  w

WINDMP [ 186: 203, 240: 260]
ppxodt
       186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203
       --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- ---
 240 : 254 254 254 254   0   0   0   0   0   0   0 255   0   0   0   0   0   0
 241 : 254   0 255   0   0   0   0   0 255   0 255   0   0   0   0   0 255   0
 242 : 254   0   0   0   0   0   0   0   0 255   0   0   0   0   0   0   0   0
 243 : 255   0   0   0   0 104   0   0   0   0 105   0   0   0   0   0 255   0
 244 :   0   0 104 104 104 104   0   0   0   0 105   0   0   0   0   0   0   0
 245 :   0   0 104 104 104 104   0   0   0   0 105   0   0   0   0 255   0   0
 246 :   0 104 104 104 104 104 104   0   0   0 105 105 105 106 106   0   0   0
 247 : 104 104   4   4   4   4 104 104   0 105   5 105   6 106   0 255   0   0
 248 : 104   4   4   4   4 104 255   0 105 105   5 105   6 106 106   0   0   0
 249 : 104 104   4   4   4 104   0   0 105   5   5 105   6   6 106   0   0   0
 250 :   0 104   4   4   4 104   0   0 105 105   5 105 105 106 106   0   0   0
 251 :   0 104 104 104 104 104   0   0   0 105 105 105   0   0   0 120 120 120
 252 :   0   0   0 104 104   0   0   0 254 254 105   0   0   0 120 120  20  20
 253 :   0   0 255 104 104   0   0   0 254 254 254 254 254   0 120 255  20  20
 254 :   0   0   0   0   0 254 254   0 254 254 254 254 255   0 120  20  20  20
 255 : 255   0   0   0   0 254 254   0 254 254 254 254 254 255 120  20  20  20
 256 :   0   0   0   0 255 254 254   0 254 254 254 254 254   0 120 120  20 120
 257 :   0 255   0   0   0   0   0   0   0 254 254 254   0   0   0 120 120 120
 258 :   0   0   0   0   0   0   0   0   0   0 254   0   0   0   0 141   0   0
 259 :   0   0   0   0   0   0   0   0   0   0   0   0   0 141 141 141 141 141
 260 : 255   0   0   0   0   0   0   0 147   0   0   0 141 141  41  41  41  41

/ 105   x,y picture address:
```

```
/ 105  bye
```

```
41% ppxodt z00661.ppx
PPXODT : Version May 30, 1989
Written 1980-1989, P. Lemkin.
Edit user image file: z00661.ppx

Commands
--------
<number> <CR> replace old contents.
/ reopen current location.
          . . .
H or ? print this list of commands
BYE  exit back to operating system

 x,y picture address: 195,250
/ 26  w

WINDMP [ 186: 203, 240: 260]
ppxodt
       186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203
       --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- ---
 240 :   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 241 :   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 242 :   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 243 :   0   0   0   0   0  13   0   0   0   0  15   0   0   0   0   0   0   0
 244 :   0   0  18  18  15  12   0   0   0   0  15   0   0   0   0   0   0   0
 245 :   0   0  27  26  27  27   0   0   0   0  23   0   0   0   0   0   0   0
 246 :   0  37  41  42  42  31  33   0   0  34  34  33  32  24   0   0   0   0
 247 :  40  49  58  57  46  42  42  43   0  43  42  38  32  25   0   0   0   0
 248 :  37  55  52  51  49  44   0   0  38  41  39  32  29  27  16   0   0   0
 249 :  31  40  38  36  40  37   0   0  28  31  29  30  23  24  20   0   0   0
 250 :   0  28  29  30  23  27   0   0  23  26  22  21  21  16  16   0   0   0
 251 :   0  24  24  21  20  21   0   0   0  20  21  17   0   0   0  16  19  11
 252 :   0   0   0  23  21   0   0   0   0   0  24   0   0   0  10  13  13  17
 253 :   0   0   0  14  17   0   0   0   0   0   0   0   0   0  15   0  93  12
 254 :   0   0   0   0   0   0   0   0   0   0   0   0   0   0   9   7   5  12
 255 :   0   0   0   0   0   0   0   0   0   0   0   0   0   0  16   4  10  10
 256 :   0   0   0   0   0   0   0   0   0   0   0   0   0   0  14  12  12  12
 257 :   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  18  22  17
 258 :   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  27   0   0
 259 :   0   0   0   0   0   0   0   0   0   0   0   0   0  48  54  51  44  37
 260 :   0   0   0   0   0   0   0   0  23   0   0   0  49  68  74  72  60  42

/ 26   x,y picture address:
/ 26  bye
```

3.18 sg2gii - 2D gel spot segmentation and quantitation

Program **sg2gii**[18] performs a 2D gel segmentation of a gel image to find and quantitate the integrated density and position of all spots in that gel. It performs a gel segmentation and spot quantitation on a computing window of a PPX formated 2D gel image file using information contained in the second derivative magnitude and direction as well as neighborhood connectivity properties. Very large regions which are potentially conglomerations of several large saturated or near saturated spots are analyzed by various methods in order to split them into their smaller component spots.

The gel to be segmented is specified by its accession number `XXXX.E` (e.g. `0324.1`, abbreviated `324.1` or `324` (the `.1` is assumed)). The algorithm is described in ([LipL80a], [LemP81a], [LesE81b], [LemP82a], [LemP83a]) and in ALGORITHM SG2GII in Appendix G. The default **sg2gii** sizing parameter limits: spot area, integrated density corrected for background D′, and OD range within the spot are specified in the `gel.rc` file (see Section 1.6.5 for sample values).

sizing limits

`INPUT GEL INFORMATION:`

The gel input image, neutral density wedge calibration and computing window (valid region in image) required for gel segmentation are associated with the accession number in the accession file entry for that gel. If the wedge calibration is all 0's, then get both the ND wedge values and the corresponding grayscale to OD calibration from the *.ppx* input picture file header. This is equivalent to manually setting the switch `-CALibPPXheader`. If the ND wedge calibration is all 0's, then get both the ND wedge OD values and the corresponding calibration from the *.ppx* picture file header. [This is equivalent to setting the `-CALibPPXhdr` switch. The accession file is searched for this information. If the gel image name associated with some gel `XXXX.E` is *a0NNNN* then other images will be derived from the *0NNNN* part of the name. Note that the alternative `b` prefix may be used. The input image file would be *b01493.ppx* and resides in the `ppx` directory `ppnp1x` path specifed by `gel.rc`.

OD calibration

`OUTPUT FILES:`

The ASCII data output file is called the Gel Segmentation File (GSF), with generated name `p`*EXXXX*`.gsf` and is suitable for further analysis by other programs (e.g. for *XXXX.E* of `0123.4`, the GSF file will be `p40123.gsf`. These include the gel comparison programs **cmpgl2** and **autopair**, and the draw Rmap program

GSF

[18] Note that **sg5gii** is the same as **sg2gii** but is able to handle original image data of higher bits/pixel such as the Molecular Dynamics laser scanner (12-bits) or PhosphorImager (16-bits) whereas the original segmenter only handles 8-bits/pixel data.

dwrmap and **annotate**. The *.gsf* file will be placed in the `gel.rc` specified `aux` directory `ppnp2x`.

The output image for gel `XXXX.E` is `z`*0NNNN*`.ppx` and is located on the `gel.rc` specified `tmp` directory `ppnp3x` (see `gel.rc` discussion in Section 1.6.5 page 61). If the central core image is to be saved (by specifying the `-CTLcore` switch), then it will be put into picture file `c`*0NNNN*`.ppx` in `gel.rc` specified directory `ppnp3x`. In addition, the average image is written out into `j`*0NNNN*`.ppx`, background notch filter image in `n`*0NNNN*`.ppx`, the Laplacian magnitude in `k`*0NNNN*`.ppx`. All temporary pictures are stored in `gel.rc` specified `tmp` directory `ppnp3x`. If `-RESTofPPX` is specified, then an additional output image `y`*0NNNN.ppx* is created which is the original gel less the segmented spots (i.e. `z`*0NNNN.ppx*). If the default `-FULL` is specified, then derived output images will be at the same resolution as the input image. If the `-COMPRESS` switch is also specified, then compress derived images.

derived images

SPOT OVERLAYS IN OUTPUT IMAGE:

Spots may be indicated in the `z` or `y` output images in several ways. The `-DRAWSPOT:`<*options* `O`*riginal or* `Z`*image*> switch will draw one or more of the options (`D` for a dot, `P` for a plus (+) and `B` for a boundary. It will draw it in the default `Z`image rather than in a copy of the `O`riginal image.

SCANNER SWITCHES:

User defined default switches may be specified as a resource string `sg2gii.switches: ...` in the `gel.rc` state file. See the PGELRC program for defaults. For example:

```
-3x3 -BUSSE:3:C -SAT:99.7 -BACK:64 -RESTOFGEL -CCMIN:4 -DRAWSPOTS:PO
```

Figure 3.24 shows the original gel image which will be segmented. Figure 3.25 shows the propagated connected component derived gel image of segmented gel. Figure 3.26 shows the notch filter background density derived gel image of segmented gel. Figure 3.27 shows the rest-of-image derived gel image of segmented gel. This image is computed by subtracting the spots which were found from the original image. Figure 3.28 shows the final segmented spots derived gel image of segmented gel.

Figure 3.24. Sample **sg2gii** of original gel image which will be segmented.

Figure 3.25. Sample **sg2gii** of propagated central core image.

Figure 3.26. Sample **sg2gii** gel background image computed using the notch filter on the rest-of-gel image.

Figure 3.27. Sample **sg2gii** of a rest-of-gel image computed by subtracting spots which were found from the original image. This is the effective input used by the background image used to estimate a spots background OD.

Figure 3.28. Sample **sg2gii** of final spots segmented by this program.

IMAGE SMOOTHING AND LAPLACIAN OPERATIONS:

smoothing filter

Two image convolution operations are performed when doing the gel segmentation: Gaussian smoothing and a Laplacian. The Gaussian smoothing is specified by one of the following switches `-3X3`, `-5X5`, `-7X7` or `-13X17:`*size*. For a 512x512 image, the `-7X7` filter is optimal while the `-13X17` one is for a 1024x1024 image. If you do not specify the filter, then it is defaulted from the image size. Similarly, there are three Laplacians: `-LAPLACE5X5`, `-BUSSELAPLACIAN:`*n:mode* (*n*`X`*n* and a 3x3 if neither of the other two are specified. For mode being `P`, pixel weights are used, for `C` 3x3 averaged pixel weights are used. Since the default Laplacian is `-LAPLACE5X5`, one would specify `-NOLAPLACE5X5` to use the 3x3 filter.

The `-AVERAGEOD` and the `-DIFFERENCEOD` switches do the image averaging and Laplacian computations respectively in OD space rather than grayscale space.

HANDLING SATURATED SPOTS:

The `-SATURATESPOTS:`*percentThreshold* option will try to find saturated spots greater than *percentThreshold* of the darkest pixel (default 0.95). Note that integrated density values for these spots will not be accurate. The `-SPLITSPOTS:`*(*B*oundary,* R*egion or* T*hreshold), minCCsize* switch invokes Miller's spot splitting algorithm [OlsA89] which is useful in some cases where spots are not resolved which should have been. The `-CCMINSIZE:#pixels` ignores spots or subspots less than this size. Normally these switches are not used.

saturated and splitting spots

`IMAGE SIZE AND DEFAULT PARAMETERS:`

These disk directory path names are kept in the gel substate file `gel.rc` (which is defined by the user the first time **sg2gii** is invoked if gel.rc does not exist). `gel.rc` may also be edited using a text editor. If `-PPXimageSize` is specified, then it is changed from the default 512x512 to the (*ncols,nrows*) is obtained from the header of the PPX file The default computing window is set to `[2:nCols-2:2:nRows-102]` if it is not defined. In any case it is clipped to `[2:nCols-2,2:nRows-2]`. The ND wedge values from the accession file are updated in the gel.rc file. If the `-CALibPPXhdr` switch is specified, then get the ND wedge calibration from the picture file header. This switch is automatically set if the accession file ND wedge calibration for the gel being segmented is all 0's.

USAGE:

```
sg2gii <ACC#> [<optional -switches>]
```

Type `sg2gii -info` to get more information.

SWITCHES

-AllowTouchingEdges allow spots which touch an edge to be counted rather than deleted (the default).

-AVerageOD instead of grayvalue in Gaussian filters (default).

-BAckGroundSize:n compute the background zonal notch filter using a nxn window rather than the 32x32 default for n (legal range is 2 to 64).

-BUsseLaplacian:<gridSize><opt. :Cone> use Busse's variable grid Laplacian with *gridSize* sampling. If the *C*one option is used, it increases weights in the Laplacian filter extrema. The default gridSize is 2 (equivalent to `-LAPLACE5x5`). The default is `-NOBUsse:3:C`.

-CALibPPXhdr get the ND wedge calibration from the PPX file header.

-CAShеLines:*n* use n lines if running cache line buffer version of the segmenter (default 40).

-CCminSize:<min # pixels> minimum number of pixels a spot's central core has to be before it is considered a spot (default 2).

-CHangeparameters:<new-parameters> modify segmentation parameters from those specified in `gel.rc` file. The parameters which can be changed are:

1. computing window C:x1,x2,y1,y2,
2. spot area in pixels sizing range A:a1,a2.
3. spot total density (in OD) sizing range D:d1,d2].
4. density range within the spot (in ND) sizing range O:o1,o2. Example: C:100,200,100,200,A:20,500,D:25,200,O:0.05,2.1

-COmpress derived images and GSF file after they are created on the disk to save space.

-CTlcore to save the central core image on the picture disk with a `c0=`*nnnn*`.ppx` picture file instead of central core `c-.ppx`. Also save Laplacian magnitude `k-.ppx`, averaged `j-.ppx` and background `n-.ppx` images.

-DEBug:bits,dbCW,wss dump various conditional debugging parameters onto the screen and output GSF file. The *bits* are the debug bits specified as either octal or decimal and enable particular debugging output if the program was compiled with debugging enabled. The optional *dbCW* is *dbx1,dbx2,dby1,dby2* and is used when printing debug information about the 'debug' window. The optional dump window step size *wss* (default 1) samples the the image being dumped. [Only available in EXPERIMENTAL version.]

-DELete set deleted spots (i.e. those which do not meet sizing criteria) to the value 255 (the default) rather than the 0 in the central core image. A value of 0 indicates there are NO deleted spots in the central core image.

-DIfferenceOD use OD instead of gray value when computing central core Laplacian (default).

-DRawSpots:<*options***>** and **O** *or* **Z** indicate spots in the `y` and `z` output images by a dot (`D`), a '+' (`P`) or a boundary (`B`). One or more of (`D P B`) can be used. Draw this in a copy of the `O`riginal or segmented `Z` image (default `-DRAWPLUS:PZ` default).

-EstimateBackground if it dMin for a spot is $<$ *mnBkgrdDens* for that spot, then use *dMin* for *mnBkgrdDens* in that spot's calculation of D' as $D' = D - A * mnBkgrdDens$ This is useful if the *mnBkgrdDens* is inaccurate because of poorly segmented adjacent spots or other artifacts. (Default is `-ESTIMATEBACKGROUND`).

-Full output derived images at same image resolution as input image (the default), else use standard 512x512 size.

-Histfeatures:<opt. ADPOXNBQE>,<opt. SimpleOutput> compute histograms of specified individual spot features: A=area, D=density, P=D', O=Od Diff (dmax - dmin), X=dmax, N=dmin, B=meanbknd, Q=(dmin - meanbknd), E= 100%*(dPrimeM-dPrimeB)/dPrimeM. It generates files with the name <*feature-letter*>*EXXXX.hst*. If *SimpleOutput* is added, then make output suitable for plot.

-IGnorebackground ignore background density subtraction correction so that $D' = D$. Otherwise $D' = D - AmnBackground$ for each spot.

-IMageSize:nrows,ncols change image size to nrowsXncols (or PPX to get size from the PPX file header).

-INfo print additional information on **sg2gii**.

-LAplacian5x5 use a 5x5 Laplacian to compute the central core instead of the default 3x3 or Busse Laplacian of size nXn (default `-LAPLACE5x5`).

-LOngGSF Output GSF file with extended feature labeling (the default is `-NOLongGSF` which is equivalent to `-SHortGSF`).

-OPtimizeFBL optimize the FBL processesing (default off).

-OUtputpix save the output image in *z0NNNN.ppx* (the default).

-PIxdump:w,x,t,wss dump the pictures generated during the segmentation algorithm defined by the computing window (with maximum width w of 18 pixels for an 80 column screen (the default) and up to 30 for a 132 column screen) on the screen and output GSF. Optional (x, y) to specify ULHC of dump window (else default CW). Optional window step size *wss* (default 1).

-PPximagesize use image size specified in the PPX file header. Image size greater than 512x512 may be used (eg. 1024x1024).

-QUICkestimate of D$'$ using only central core and gaussian estimate.

-QUIEtswitch print stage of the analysis and line numbers as the gel is being processed, (default is `-QUIET`).

-RAte print the line I-O paging rate for lines rather than the line 'number' during processing.

-REstofpix create the output image `y0`*NNNN*`.ppx` be defined as the original image less the `z0`*NNNN*`.ppx` segmented spots. This is useful in finding spots that were fragmented or were missed.

-SAturatedSpotPropagate:PercentThreshold propagate saturated spots from central cores to adjecent pixels with similar high gray values. Only apply this to spots > threshold percent of maximum spot OD seen in gel.

-SHortGSF Output the GSF without as much feature labeling. (This GSF format is the default). Note that `-NOSHortGSF` is equivalent to `-LongGSF`.

-SIzeD'remove delete spots in CC and `z0`*NNNN*`.ppx` based on D' (the default) instead of just D (`-NOSIZED'REMOVE`).

-SPlitSpots:(B *or* **R** *or T*)**,***minCCsize* split large spots with connected component area > *minCCsize* (default 15). `B` uses a boundary analysis method, `R` analyzes the Run Length Map of the spot while while `T` uses a method based on finding multiple subspot clusters using Miller and Olson's thresholded Laplacian test.

-STd512ppxFile use the standard '`s`' 512x512 image even if higher resolution image exists.

-Usage print UNIX command level switch usage.

-Version print the version of the program

-WmWait When done, wait until do CLICK TO EXIT `mwwait` widget to exit.

-Ytitle:newYposition position the 4 line title at the new Y-position (default is imagesize-62).

-3x3lowpass use the 3x3 pixel low pass averaging filter (`-NO3x3LOWPASS` is the defaults).

-5x5lowpass use the 5x5 pixel low pass averaging filter (`-NO5x5LOWPASS` is the default).

-7x7lowpass user Miller's [VoKP81] 7x7 low pass Gaussian filter (this filter is the default).

-13x17lowpass:<opt. size> use 13x17 pixel Gaussian low pass averaging filter with *size* σ (`-NO13x17LOWPASS` and *size*=1.00 is the default). It uses a convolution filter of the form $\exp^{\frac{-(x^2+y^2)}{2\pi\sigma^2}}$, where σ is 6.0.

EXAMPLES OF USAGE

```
sg2gii 324.1
     # Default.

sg2gii 324.1 -shortgsf -allowtouch -7x7 -background:32 -avgOD -diffOD
     # This is the above default.

sg2gii 324.1 -7x7Laplacian -noAllowTouchingEdges
     # Disallow spots which touch an edge.

sg2gii 324.1 -7x7 -noA
     # short form of switch.

sg2gii 324.1 -longGSF
     # Long form has labled data.

sg2gii 324.1 -3x3
     # Average using a 3x3 Gaussian filter.
sg2gii 324.1 -5x5
     # Average using a 5x5 Gaussian filter.

sg2gii 324.1 -pixdmp:18 -ch:C:313,330,195,212,A:4,100,D:0,1000,O:0,2.7
     # print numeric window trace for small computing window.

sg2gii 324.1 -pixdmp:18 -ch:A:4,100,D:0,1000,O:0,2.7
     # Change some of the sizing parameters.

sg2gii 24h.ppx -7x7 -noGel
     # Segment an image which is not a gel.

sg2gii 2001.1 -imagesize:1024,1024
     # Specify non-default image size.

sg2gii 2010.1 -ppx -calibNDppxHdr
     # Use input image size and ND wedge determined by PPX
     # file header.

sg2gii 324.1 -drawSpots:DZ
     # Draw '.' dots in center of each segmented spot Z image.

sg2gii 324.1 -drawSpots:PO
     # Draw '+' in center of each segmented spot overlayed in
     # the original gel.

sg2gii 324.1 -drawSpots:BPO
     # Draw '+' and boundaries for each segmented spot overlayed in
     # the original gel.

sg2gii 324.1 -stdPPX
     # Use low resolution gel even if high resolution gel exists.

sg2gii 324.1 -13x17
```

```
        # Use larger filter and 5x5 Laplacian for high resolution gels.

    sg2gii 324.1 -13x17:1.0 -laplacian5x5
        # Use larger filter and laplacian for high resolution gels.

    sg2gii 324.1 -13x17:1.0 -busseLaplacian:3:C
        # Use larger filter and 7x7 (2*3+1) Laplacian for high
        # resolution gels.

    sg2gii 324.1 -noLaplacian5x5
        # Use 3x3 Laplacian.

    sg2gii 324.1 -Back:64 -CH:A:30,100000,D:0.0005,100000,O:0.001,4.5 \
-3X3 -CCmin:4 -Busse:3:c -Sat:99.7 -full -drawSpots:BPO -RestOf
    accppx 324.1.1 -p1:y -p2:z -laser:laser5
        # For 1K BioImage gel. Dump the segmented gel and the rest-of
        # images on laser printer laser5.

    sg2gii 324.1 -full -7x7 -ch:A:45,8000,D:0.5:10000,O:0.040:4.0 -Restof
        # For 1:2 sampled Molecular Dynamics gel.
```

image smoothing

The standard set of switches generally used with **sg2gii** are `-shortgsf`, `-allowtouchingedges` and `-7x7lowpass`. So the short form is the default.

3.18.1 Smoothing filters used in sg2gii

There are three possible smoothing convolution filters available in **sg2gii**. One of these must be used to remove image noise prior to further processing.

The 3x3 high pass filter

```
1 2 1
2 4 2  divided by 16
1 2 1
```

The 5x5 low pass filter

```
1 1 2 1 1
1 2 4 2 1
2 4 8 4 2   divided by 52
1 2 4 2 1
1 1 2 1 1
```

7x7 low pass filter (from [VoKP81])

```
 4  -6  -12  -14  -12   -6  4
-6   9   18   21   18    9 -6
-12 18   36   42   36   18 -12
-14 21   42   49   42   21 -14   divided by 441
-12 18   36   42   36   18 -12
-6   9   18   21   18    9 -6
 4  -6  -12  -14  -12   -6  4
```

13x17:σ low pass filter Is a symmetric filter using the Gaussian $\exp \frac{-(x^2+y^2)}{2\pi\sigma^2}$. The default of σ is 10.

CENTRAL CORE IMAGE CODING SCHEME

As described in ([LipL80a], [LemP81a], [LemP83a]), pixels in the propagated central core image file `c-.ppx` have the following semantics.

0 background pixel.

1 unlabeled central core pixel of potential spot region.

2:99 labeled central core pixel of spot region.

102:199 labeled propagated central core pixel of spot region.

254 deleted spot in central core if fails sizing (0 if `-NODELETE`).

255 isolated pixel or non 4-neighbor connected pixel.

You can use **ppxodt** window mode or `Xpix` `print region` mode to look at numeric windows of the resultant images. The **sg2gii** `-PIXDUMP:w` switch can print debug windows of size wxw (18 for 80 columns and 30f for 132 columns). The `-DEBUG:0,dbCW` can be used to change the default debug CW from the standard CW. Several examples of some of these numeric windows are given in the **ppxodt** discussion in Section 3.17.

3.18.2 Format of Gel Segmentation File *.gsf*

A typical example from a short format **sg2gii** GSF file is illustrated below.

`.gsf` *file format*

```
SG2GII: Version December 19, 1988
Today's date is 01/19/1989, 02:30:48PM
User: /users/joeUser
Written 1980-1988, P. Lemkin.
Gel Segmentation File is: p10369.gsf
in directory /users/joeUser/gellab/aux/
0369.1/..../?/?/3-10-82/#2/CULT #2/3:10,5-20%/
0 HRS/H3/2 HRS/96 HRS/HEME MALIG-ALL,LYMPHOID/
B00889/-NONE-/-NONE-/VIDICON-AUTO,28MM,F8,76CM/LESTER*
 028 051 075 098 118 138 155 169 183 192 200 208 215 225 229 000 061 505 068 503

Switches:
Window [61:505,68:503]
Spot Area sizing limits (10:2000)
Integrated Density sizing limits (0.30:500.00)
Density difference sizing limits (0.03:2.70)
Zonal Notch filter background window 32x32
```

```
Saving output image in /home/joeUser/gellab/aux/z00889.ppx

Background range [0.00:0.66] OD
 CC#1[140:145, 69:72][0.00:0.13]0.06,0.00
  [142.95,70.33]20,1.15,1.15
  0.38,0.23,0.15,0.08
 CC#2[149:152, 70:72][0.00:0.13]0.04,0.00
  [149.91,70.37]10,0.41,0.41
  0.20,0.00,0.00,0.00
 CC#3[158:164, 69:72][0.00:0.24]0.12,0.03
  [161.37,70.09]17,2.00,1.49
  0.49,0.19,0.20,0.15
        .
        .
        .
 CC#813[357:363, 499:503][0.00:0.06]0.01,0.00
  [361.00,501.68]24,0.31,0.31
  0.00,0.00,0.00,0.00
Total of 879 accepted D spots accumulated density=21571.9, area=46621
Total of 813 accepted D' spots accumulated density=16302.4, area=45044
Total of 3502 omitted D spots accumulated density=472.1, area=40652
Omitted/Accepted density =2%
Total # spots failed Area sizing[areaT1,areaT2]=[2014, 0]
Total # spots failed Density sizing[densityT1,densityT2]=[3077, 0]
Total # spots failed ODrange sizing[ODrangeT1,ODrangeT2]=[5218, 0]

FINISHED! The Gel Segmentation File, GSF, is p10369.gsf
Real TIME =00:03:34  CPU TIME =00:03:16,  91.59%
```

Compute spot moments S_x, S_y, S_{xy}, and the Gaussian estimate of density volume $V = 4.0\sqrt{\pi} D_{max} S_x S_y$. The long output format for a typical spot entry from another gel is illustrated as follows.

```
CC# 427 M.E.R[119:122,150:153] D.R.=[.58:.66] D/A= .613 MnB= .556
1st MOM[120.89, 151.58] A= 12 D= 7.36 D'= .68
Sx= .96 Sy= 1.11 Sxy= .81 V= 4.98
```

Where:

```
CC is the connected component or unique spot index.
M.E.R. is the minimum enclosing rectangle [x1:x2,y1,y2].
D.R. is the range in pixel OD seen within the spot.
D/A is the density divided by area (mean density).
MnB is the mean background density at this spot.
1st MOM(x,y) is the centroid of the spot in picture space coordinates.
D is the total spot density in OD without background correction.
D' is the total spot density in OD with background correction.
A is the total spot area in pixels.
Sx, Sy and Sxy are the density weighted gaussian size estimates of
   the spot.
V is the density 'volume' approximated from Sx, Sy and
   maximum pixel Density-MnB.
```

3.18.3 C-gel′ synthetic *.gsf* files produced by the C-GEL′ command in cgelp2

In the case where a synthetic Cgel′ is produced from a composite of gels using **cgelp2** - *not* **sg2gii**, a *.gsf* file is produced which has the `.9` accession number extension reserved for such estimates of the Cgel′. It is only generated in `-SHORTGSF` format and has four additional fields in the third line of the record. These are in order, the standard deviation of the centroid values (D', $area$, X, Y, $\#gels$). The #gels were those used to compute the means and centroid of the Rspot set used to generate the C-spot′ estimate. The following is a short form example of a CC entry. *Cgel′*

```
CC#2[213:215, 241:245][ .32: .57] .138, .000
 [ 214.29, 242.57]27, 3.77, 3.77
  .96, 1.11, .81, 3.90 2.39, 12.27, 1.28, .90, 7
```

The last line in this case is interpreted as:

```
Sx= .96 Sy= 1.11 Sxy= .81 V= 4.9 sD'=2.39 sA=12.27 sXc=1.28 sYc=.90 #g=7
```

Where:

```
D' is the mean total spot density in OD with background correction.
A is the mean total spot area in pixels.
1st MOM(x,y) is the mean centroid of the spot..
sD' is the total spot density variance in OD with background correction.
sA is the total spot area variance in pixels.
sXc,sYc are the mean centroid variance of the spot.
#g is the number of gels used to compute these means.
```

3.18.4 Tuning the parameters of the segmenter

Although the default segmenter parameters are fairly robust, you can optimize the segmentation by adjusting some of the parameters. We suggest you create a batch script to titrate some of the parameters and generate segmented output images which you either print or rename and review with **Xpix** after the job is finished. The following is a sample script which adjusts minimum area and prints each segmented image on the laser printer.

```
#!/bin/csh
# JOB testPrint.do - Segment gels 8-27-91. Titration of Parameters
# First titrate area, then density, OD range, etc.

date

# [1] minimum area = 5
sg2gii 8003.1 -ch:A:5,10000,D:0.001,10000,O:0.001,4.5 -restofPPX
ppx2ps -title:"1. A5D0.001N64 y00002" gellab/tmp/y00002.ppx | lpr
ppx2ps -title:"1. A5D0.001N64 z00002" gellab/tmp/z00002.ppx | lpr
```

```
# [2] minimum area = 10
sg2gii 8003.1 -ch:A:10,10000,D:0.001,10000,O:0.001,4.5 -restofPPX
ppx2ps -title:"2. A10D0.001N64 y00002" gellab/tmp/y00002.ppx | lpr
ppx2ps -title:"2. A10D0.001N64 z00002" gellab/tmp/z00002.ppx | lpr

# [3] minimum area = 15
sg2gii 8003.1 -ch:A:15,10000,D:0.001,10000,O:0.001,4.5 -restofPPX
ppx2ps -title:"3. A15D0.001N64 y00002" gellab/tmp/y00002.ppx | lpr
ppx2ps -title:"3. A15D0.001N64 z00002" gellab/tmp/z00002.ppx | lpr

# [4] minimum area = 20
sg2gii 8003.1 -ch:A:20,10000,D:0.001,10000,O:0.001,4.5 -restofPPX
ppx2ps -title:"4. A20D0.001N64 y00002" gellab/tmp/y00002.ppx | lpr
ppx2ps -title:"4. A20D0.001N64 z00002" gellab/tmp/z00002.ppx | lpr

# [5] minimum area = 25
sg2gii 8003.1 -ch:A:25,10000,D:0.001,10000,O:0.001,4.5 -restofPPX
ppx2ps -title:"5. A25D0.001N64 y00002" gellab/tmp/y00002.ppx | lpr
ppx2ps -title:"5. A25D0.001N64 z00002" gellab/tmp/z00002.ppx | lpr

# ... Then titrate density and then OD range, etc.

#--------- the end ---------
```

Alternatively, we could look at the gels with **Xpix**:

```
#!/bin/csh
# JOB testDisplay.do - Segment gels 8-27-91. Titration of Parameters
# First titrate area, then density, OD range, etc.

date

# [1] minimum area = 5
sg2gii 8003.1 -ch:A:5,10000,D:0.001,10000,O:0.001,4.5 -restofPPX
accppx 8003.1 8003.1 -p1:z -p2:y

# [2] minimum area = 10
sg2gii 8003.1 -ch:A:10,10000,D:0.001,10000,O:0.001,4.5 -restofPPX
accppx 8003.1 8003.1 -p1:z -p2:y

# [3] minimum area = 15
sg2gii 8003.1 -ch:A:15,10000,D:0.001,10000,O:0.001,4.5 -restofPPX
accppx 8003.1 8003.1 -p1:z -p2:y

# [4] minimum area = 20
sg2gii 8003.1 -ch:A:20,10000,D:0.001,10000,O:0.001,4.5 -restofPPX
accppx 8003.1 8003.1 -p1:z -p2:y

# [5] minimum area = 25
sg2gii 8003.1 -ch:A:25,10000,D:0.001,10000,O:0.001,4.5 -restofPPX
```

```
accppx 8003.1 8003.1 -p1:z -p2:y

# ... Then titrate density and then OD range, etc.

#--------- the end ---------
```

3.19 tek2psG - Tektronix to PostScript plot file conversion

printing 4010 graphics

tek2psG converts Tektronix 4015 alphanumerics and graphics sequences into PostScript. The Tektronix 4010 is a subset of the 4010. GELLAB-II generates 4010 style Tektronixs output. **tek2psG** is primarily used with **plotn** `-display:4010` to convert GELLAB-II UGF graphics files to PostScript which is more suitable for printing on a laser printer.

tek2psG was originally designed to be used with the COPY function in the 4015 emulation of *xterm(1)*, but should be general purpose enough to be used with any 4015 output. If no files are specified, then **tek2psG** reads its standard input. The output of **tek2psG** is sent to the standard output, so that it can be redirected to a file or piped to a PostScript printer via *lpr(1)*. By default, the output is drawn in portrait mode, centered on the page and scaled to fit. The `-LANDSCAPE` option causes the output to be drawn in landscape mode, again centered and scaled to fit (the default landscape drawing scale is larger than the portrait drawing scale). The `-XORIGIN #` and `-YORIGIN #` options specify, in inches, where to position the origin (lower left-hand corner) of the drawing relative to the lower left-hand corner of the page (in either portrait or landscape modes). Not specifying the origin in either one or both dimensions causes the drawing to be centered within the page in that dimension. The `-WIDTH #` and `-HEIGHT #` options specify the size of the drawing, in inches. If only one of these is specified, the other is adjusted to preserve the correct aspect ratio. By specifying both width and height, the aspect ratio can be changed arbitrarily.

Normally, the PAGE escape sequence causes the page to be erased. If the `-PAGE` option is specified, **tek2psG** prints a copy of the current screen on the printer, then erases the page and continues drawing the rest of the input file. Certain programs that output Tektronix codes often clear the screen before drawing anything. If the `-P` option is specified when processing such an input file, an extra blank page is produced. Use the `-FIRST` option to ignore the first PAGE escape sequence found in the input file. **tek2ps** was originally written by Edward Moy, UC Berkeley. Further hacks `-FIRST` and `-PAGE` modifications by Mic Kaczmarczik, U.T. Austin Computation Center. When integrated with GELLAB-II, it was renamed to **tek2psG**.

USAGE:

```
tek2psG [<Opt. -switches>] file(s)
```

Type `tek2psG -info` to get more information.

SWITCHES

-First Ignore first PAGE escape found in Tek input.

-Height # specifies the height of the drawing in inches.

-Landscape Plot in landscape mode (i.e. rotate 90 degrees).

-Info Display general information about this program.

-Page Print copy of current screen and erase (disabled).

-Usage Display command line format information.

-Version Print current version number of the program.

-Width # specifies the width of the drawing in inches.

-WmWait When done, wait until do CLICK TO EXIT `mwwait` widget to exit.

-Xorigin # specifies the Lower Left corner of the drawing in inches.

-Yorigin # specifies the Lower Left corner of the drawing in inches.

EXAMPLES OF USAGE

```
tek2psG -p < 000001.tek > 000001.ps
       # Convert Tektronix file to Postscript file.

plotn -Display:4010 000001.ugf | tek2psG -p > 000001.ps
       # Convert GELLAB UGF file to Postscript file.

plotn -Display:4010 000001.ugf | tek2psG -p | lpr -Plaser
       # Convert GELLAB UGF file to Postscript and print it.
```

Chapter 4

Xpix Image Display Program

Xpix[1] is a UNIX X-Window System client program to display and process gray scale pictures. Upon startup, one or more PPX-formated (Portable PiXture) files containing optional header information and 8-bit picture data are read into memory and presented for display enhancement as well as operations on and between pictures. A number of image transforms are then available which can operate on pictures kept in memory. The resultant transformed pictures can then be displayed or saved in picture files. The operator uses the mouse and cursor to directly control these interactive functions. A number of built-in unary and binary picture transforms are available as well as some neighborhood picture functions ([RosA77], [LemP79a], [LemP80a], [BalD82]). Provision is made to dynamically add external picture functions (i.e. UNIX programs) to **Xpix** either through the startup file `.Xdefaults` or interactively. See Appendix D, page 603.

image display

Multiple pictures reside in memory at the same time. This then permits the results of picture transforms to be saved and reused as source pictures for successive transforms. Image data may be calibrated in x, y and grayscale user-specified units using either a default linear calibration or externally supplied calibration lookup tables which can be computed and/or read from files. This lets measurements of picture data be made in the appropriate system of units – and with non-linear scales if necessary. An interactively defined rectangular *computing window* [x1:y1,x2:y2] (initially the size of the full image) can be used to restrict picture transforms to a particular region of a picture.

other functions

[1] There are two X11 image display programs: **Xpix11** derived from the original X10 **Xpix** and the newer **Xpix2** which is more portable and runs under OpenLook and Motif better than **Xpix11**. Most GELLAB-II programs that want to display a derived image invoke **Xpix** so the GELLAB-II maintainer should make a symbolic link in ~`gelmgr/gellab/bin/sun4/` from either **Xpix11** or **Xpix2** to **Xpix**. The **disp11** program is able to flciker whole gels rather than zoom regions.

4.1 Xpix11 and Xpix2 versions of Xpix

Xpix was derived from the public domain *showimg(1)* X client program distributed with X10R4 by William F. Wyatt, Smithsonian Astrophysical Observatory and Jim Gettys of DEC/Athena. The X11R4 version is called **Xpix11** and although it is functionally similar was a major rewrite of the original X10 code. Since the X10 and X11 versions are functionally similar, we use this same discussion here for both - but be aware that there are differences and not all X10 functionality is implemented in the X11 version. The **Xpix2** program is a completely new image viewer with different functionality, but it may be substituted for **Xpix** in looking at derived images. It is much more menu driven and is fairly self explanatory.

Although we don't go into the details of **Xpix2** here, we briefly list the main menus.

FILE File related commands (pull down menu).

- **Add** Add a picture file to list of displayable pictures.
- **Quit** Quit the application.

IMAGE1 Display a picture in image window 1 (popup directory of files).

IMAGE2 Display a picture in image window 2 (popup directory of files).

HELP Popup a window describing keyboard/mouse bindings actions in images.

LOGGING Toggle logging measurements into a logging file xpix2-pid.log.

MEASURE Measure regions (pull down menu).

- **Background 1** Draw region to set mean background for image 1.
- **Quantitate 1** Draw region and measure region using mnBkgrd for image 1.
- **Background 2** Draw region to set mean background for image 2.
- **Quantitate 2** Draw region and measure region using mnBkgrd for image 2.

FUNCTIONS Image 1 functions ==¿ image 2 (pull down menu).

- **Threshold** Image 1 by slicing gray values [g1:g2].
- **8-neighbor average** of Image 1.
- **4-neighbor gradient** of Image 1.
- **(NA) 8-neighbor gradient** of Image 1.
- **Laplacian** of Image 1.
- **(NA)Busse Laplacian** of Image 1.
- **Complement** Image 1 into Image 2.
- **(NA) Rotate 90** Image 1 +90 degrees into Image 2.
- **(NA) Rotate -90** Image 1 -90 degrees into Image 2.
- **(NA) Rotate 180** Image 1 +180 degrees into Image 2.

ANNOTATION Annotate Image 2 functions (pull down menu).

- **(NA) Draw annotation** Draw annotation in image 2.

(NA) Delete annotation Delete annotation in image 2.

(NA) Move Annotation Move annotation in image 2.

(NA) Erase region 2 Draw region then Erase it in Image 2.

FLICKER Flicker images functions (pull down menu)

Flicker Flicker images 1 and 2.

The key and mouse bindings are listed as follows and depend on the mode you are in.

```
Normal Mode
===========
Keyboard
--------
     p  - (Popup)  Display the magnifier window in its own popup.
     n  - (None)   Dismiss the magnifier window.
 CTL-z  - Cycle through magnification factors (when magnification
          window is displayed).
     c  - (Crop)   Crop the image to the region of interest within ts pressed,
                   cropping area. (See Mouse - Left Button below.)
     u  - (Uncrop) Uncrop the (previously cropped) image back to its
                   initial size.
     b  - (Quant)  Enter spot quantitation mode to define background.
     q  - (Quant)  Enter spot quantitation mode.
     a  - (Adjust Contrast)  Toggle the contrast adjustment method:
                   No Contast Adj vs. Histogram Contrast Stretching.

Mouse
-----
  Left Button - Draw a cropping area in image.
Middle Button - Change image contrast.
 Right Button - Display image coordinate and gray-value under mouse
                    pointer in upper left hand corner of image.
                If <Ctrl> or <Shift> key is pressed, display gray-values
                    in region around mouse pointer in a popup window.
                    If <Ctrl> key is pressed, display in hexidecimal.
                    If <Shift> key is pressed, display in decimal.


Spot Quantitation Mode - (cursor changes to pencil icon)
=======================
Keyboard
--------
     x  - (Delete) Delete last segment or point from the spot boundary.

Mouse
-----
 Left Button - Add segment (clicking) or point (dragging) to spot boundary.
 Middle Button - Change image contrast.
 Right Button - Close spot boundary and print spot quantitation.
 CTRL/Right Button - Close spot boundary and Erase region in image 2.
```

Unlike **Xpix11** there are no switch options in **Xpix2**. It may be invoked with a list of image file names or they may be added after it is started. If the image file has an OD calibration, then the values reported by the right button are in OD otherwise they are in gray-scale.

4.2 Xpix capabilities

Xpix is capable of performing composition of picture functions and measurements on images in addition to being used as a display vehicle. Many additional operations were added including: a single picture (PPX) file format, new picture operators including binary and unary operators (which operate on so-called image registers I1, I2 and I3), scalar and neighborhood operators, multiple images with up to two simultaneous display windows (with associated simultaneous micro zoom windows), read and write picture files, histograms, line (horizontal and vertical) profiles, image measurement, calibration of (x,y,grayscale), image comparison, status inquiry, "GraphScale" pseudocolor images, help facility etc.

color

Pseudo color can be invoked using the standard *hexcone* model [SprR85]. We have added a special type of processed image called *GraphScale* which is a standard grayscale image which has all gray values above 245 clipped to 245 (i.e. 0 is white and 255 is black). Then the color map is changed so that gray values 246 through 255 are pseudo colors red, orange, yellow, etc. It is possible in **Xpix** to then add color line graphics to a gray scale image resulting in a "GraphScale" image. The merged picture can now be saved as a PPX file and viewed later using the `graphscale` option.

Pictures as "register" variables I1, I2 and I3

picture variables

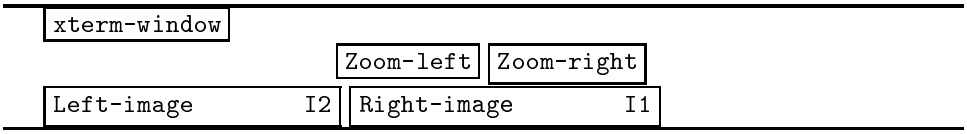
In the following discussion, we use the term *picture* to refer to the original picture file data and the term *image* to its visual transform as seen on the display. Unary and binary picture functions operate on picture objects - not image objects. The latter only exist when a picture is being displayed. In particular, the picture functions operate on I1 and I2 input pictures and generate an I3 output picture. One needs to assign pictures to these image registers before applying picture functions. This will be elaborated later. Images picture files larger than 512x512 may be read, but are sampled and computed within **Xpix** as 512x512 size images.

X-Window environment

Normally, **Xpix** figures out what display resources are available and does what is required to use them. It should be started from an **xterm** 80x25 character window in the upper left hand corner, i.e `=+0+0`. This is because prompts and tabular output are normally directed toward that window and requires that size and geometry. It is also convenient to run **Xpix** under the UNIX `script` program in that X-windows `xterm` window in order to capture image measurements made during a session. Alternatively, measurement data can optionally be appended to a

specific data logging file (see `-append` switch). Note that you *must* have a least one image (I1). A second image (I2) is optiona. The default positions of the windows are:

or alternatively using the `-ZoomRight` switch,



compressed image files

If pictures are stored compressed in the UNIX in *compress(1)* format (indicated by a `.Z` file extension (e.g. `abc.ppx` becomes `abc.ppx.Z`)), then **Xpix** will first *uncompress* them by making a copy of the picture file in a temporary file, run *uncompress*, read the uncompressed temporary file, and finally delete the temporary file. Doing this will typically save 30% to 40% or more of the picture file space at the cost of taking little more than a few seconds time in loading images.

Xpix is still under development and some of the commands discussed below are not completely debugged - although code exists. The sections on command options (Section 4.4 on page 491) and examples (Section 4.5 on page 495) illustrate the UNIX interface. The section entitled **Current Xpix Caveats** lists what is not working.

4.3 Mouse control of menus

contrast & brightness

Xpix is controlled primarily by interaction with the mouse. By default the picture(s) come up with a linear–intensity contrast–enhanced gray scale. Pressing any *mouse button* (without pressing any *key*) causes the intensities displayed to change. Moving the mouse in the *horizontal* direction while pressing a mouse button changes the background level and contrast displayed. Moving in the *vertical* direction with a mouse button pressed changes the *contrast*. The center of the window is considered the default position for background and contrast. Simply moving the mouse *without* pressing any buttons or keys presents a 2X zoom of the cursor region in a small zoom window located above the corresponding main window.

popup menus

Holding down the `CONTROL` key while pressing a mouse button causes a set of five pop-up menus to appear from which you select the operation you wish to perform. One switches between menu panes by moving the cursor, using the mouse, to any area of the desired menu. Then one moves the mouse to the desired selection within the menu. *Releasing* the mouse button while pointing to a menu entry selects that entry. If you change your mind while in the menu, move the cursor off the

menu before releasing the button and it will select nothing. The five popup menus, (VIEW, CURSOR, SCALING, UNARY OPR, BINARY OPR), are discussed below in more detail. Note the keyboard *SHIFT* key is used in conjunction with the mouse buttons for some of the commands to be discussed. The *CONTROL* key is only used to request the popup menus. Figure 4.2 shows **Xpix** started with one image and the menu brought up on the screen. The five menus are listed here to provide a context. The details are specified later in this Section.

MENU: 1. VIEW
Help
Grayscale
RGB
Graphscale
Inverse
Initialize
Status
Cycle image
Post image
Restore image
Define I1 I2 I3
DEBUG toggle
Display Large Image
EXIT

MENU: 2. CURSOR
Print region
Zoom 1X
Zoom 2X
Zoom 4X
Compare images
Print pixel
Measure region
Line profile
CW profile
Define CW
Draw text
Cursor Off

MENU: 3. SCALING
Full range
Linear
Sqrt
Log
Grid
Calibrate X Y G
Histogram
Add Img to PPX

MENU: 4. UNARY OPR
Read file $\Rightarrow I_j$
Write $I_j \Rightarrow$ *file*
Laser $I_j \Rightarrow$ *.ps file*
Copy $I_j \Rightarrow I_k$
Complement I1 $\Rightarrow$ *I3*
Threshold I1 $\Rightarrow$ *I3*
Grad4 I1 $\Rightarrow$ *I3*
Grad8 I1 $\Rightarrow$ *I3*
Laplacian I1 $\Rightarrow$ *I3*
Avg8 I1 $\Rightarrow$ *I3*
Smooth3x3 I1 $\Rightarrow$ *I3*
Median I1 $\Rightarrow$ *I3*
Def external Unary Opr
Eval UOP I1 $\Rightarrow$ *I3*

MENU: 5. BINARY OPR
Add(I1,I2) $\Rightarrow$ *I3*
Sub(I1,I2) $\Rightarrow$ *I3*
Max(I1,I2) $\Rightarrow$ *I3*
Min(I1,I2) $\Rightarrow$ *I3*
And(I1,I2) $\Rightarrow$ *I3*
Or(I1,I2) $\Rightarrow$ *I3*
Xor(I1,I2) $\Rightarrow$ *I3*
Def external Binary Opr
Eval I1 BOP I2 $\Rightarrow$ *I3*

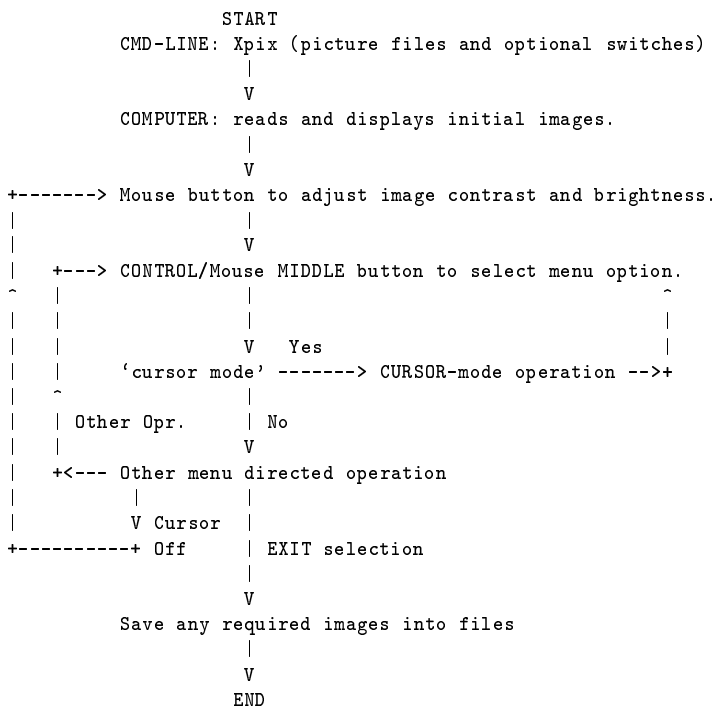


Figure 4.1. Basic user interaction loop for **Xpix**. After the initially specified image(s) are read and displayed, the user can manipulate them using the mouse to select options from the menus.

Figure 4.2. **Xpix** with popup menus. The popup menus are brought up on the screen by holding the `CONTROL` key while pressing (and holding) the mouse `MIDDLE` button when the cursor is over the window. Without releasing the button, move the menu you wish by moving the cursor to a piece of that window and then select the menu entry you desire. If you change your mind, move the cursor away from the menus before releasing the mouse button.

4.3.1 VIEW operators menu

The **first** pop-up menu selects functions dealing with display viewing. It can re–`initialize` the intensity map, display the image in `rgb` (red-green-blue "hexcone" model), `grayscale` or `graphscale`. In addition, the image may be inverted in gray value using the `inverse` toggle so that black is white etc. The `help` option reads this manuscript (file Xpix.dvi) by putting up a temporary *xdvi* window which you can then page through using any of the *xdvi* commands. *xdvi* is a TEX output file (*.dvi* file) previewer program available with X-Windows. If *xdvi* is not available, then a simple *more*–like facility on a `detex`'ed version of this file is used in the root

`xterm` window.

If the `rgb` option is selected, then the left, middle, and right mouse buttons are used to control the full hexcone mapping (0 is magenta and 255 is red with the other colors in between being blue, cyan, green, and yellow), adjustable brightness (X direction) and saturation (Y direction), and non-standard mapping in the rightmost three quadrants of (X,Y) region respectively.

The `graphscale` option sets up the GraphScale color mapping described above. The picture needs to be a GraphScale picture or it will display garbage.

The `status` operator prints the current status of **Xpix** as shown below. It lists which pictures are in memory, which are mapped to active image variables I1, I2 and I3, and which menu options are active, etc. When multiple pictures for unary and binary image operations are needed, then command `define I1 I2 I3` is used to assign pictures to working image variables I1, I2 and I3. These in turn are used by the Unary and Binary picture operators to specify which pictures are to be used as the argument(s) when picture transforms are performed. If an I3 image was not assigned when you try to do an image operation or was assigned to a picture which does not exist, **Xpix** will then create a white (i.e blank) picture in memory - not on the disk. See `read` and `write` operators in the `UNARY` operators menu to move picture files to or from the disk. When you `write` pictures, it is possible to change the output file name so as to avoid overwriting an original picture which was modified in memory.

There are up to two active full image display windows called *Right* and *Left*. The `post image` is used to change the image being displayed in the window where the command is invoked. `cycle image` is used to cycle through the existing pictures to post the "next one" in the window you have selected. By repeatedly clicking on this, you can go through all of the images. You will be asked which picture you want to post. The `display large image` displays the selected image such that it covers the entire screen - suitable for photographing. The `restore image` command restores the picture data in the image through the current color map with the *Left* and *Right* buttons restoring the left and right windows respectively. At startup the main image is displayed on the right and the auxiliary image (if `-two` *pix2File* was specified) on the left. If there is to be no auxiliary startup image, then let *pix2File* be `0` and a white image will be displayed for the second image. Although you can change the images displayed in the window(s), currently you must start **Xpix** with at least one image.

```
------------P A R A M E T E R S ----------
Posted Right Image I1 [mcrew.ppx] size nrows=512 by ncols=512
Posted Left  Image I2 [boys.ppx]
Threshold1=0 threshold2=255
External Unary UOP =  [ ] sw = [ ]
External Binary BOP = [ ] sw = [ ]
```

```
Right Computing window[0:511,0:511] Pixel range=[127:247] mode=210 max peak=247
Left  Computing window[0:511,0:511] Pixel range=[ 37:255] mode=185 max peak=255
  ----- ACTIVE PICTURES -----
  Picture #1 [mcrew.ppx]
  Picture #2 [boys.ppx]
  Picture #3 [i3.tmp ]
  Picture #4 [i4.tmp]
  Picture #5 [ ]
  Picture #6 [ ]
  Picture #7 [i7.tmp]
  Picture #8 [boys.ppx]
Assigned I1 to Picture #1 [mcrew.ppx]
Assigned I2 to Picture #2 [boys.ppx]
Assigned I3 to Picture #4 [i4.tmp]
Display{Planes, Cells}={8,256} cycleColor=3
colorinfo{nplanes, ncolors, shift,pixels[0]}={7,128,0,128}
VIEW menu       = 0X4A = {GrayScale Status }
CURSOR menu     = 0X0 = {}
SCALING menu    = 0X2 = {Linear }
UNARY OPR menu  = 0X0 = {}
BINARY OPR menu = 0X0 = {}
```

MENU: 1. VIEW operators

`help` enter help facility.

`grayscale` set color map to grayscale (from RGB) (default).

`RGB` set color map to color (from Grayscale).

`graphscale` set color map to display graycale with pseudocolor graphics.

`inverse` set $minGray$ = black and $maxGray$ = white (toggle).

`initialize` reset the color map using current SCALER mode.

`status` print status of **Xpix**.

`cycle image` cycle to next image I_j to display in current window.

`post image` select image I_j to display in current window.

`restore image` restore display window image from picture data.

`define I1 I2 I3` define mappings of pictures to working images for UOP/BOP.

`DEBUG toggle` toggle debugging switch.

`display large image` display current image (1024x900 on SUN).

`EXIT` back to UNIX.

4.3.2 CURSOR operators menu

The second menu selects other cursor-activated options. These include: printing, panning, zooming, comparing two different images, printing pixel data and displaying X or Y line or region intensity profiles. The profile region is generally made a narrow rectangular region defined by the `computing window` command. As each of these is selected, the mouse cursor is changed to reflect the intended operation. `cursor off` is used to take the mouse out of *any* cursor mode and return it to the default mode for adjusting the background and contrast levels. The position of the cursor determines on which window the operation will be performed. For example, if the cursor is in the left window, then the operation is performed on the image in that window when the mouse button is clicked.

Clicking a mouse button with the `print-pixel` option prints the (x,y,grayvalue) for that pixel as well as the distance to the previous measurement (in the current calibration units). It also records an incrementing measurement number for each picture. This may be useful when performing a sequence of measurements on images. The previous measurement must of course be made in the same window. If the `-mark` switch was specified, then a '+' is marked in the image of the pixel. The color of the '+' is determined by the key pressed: (left, middle, right) map to (WHITE, RED, BLACK). (If `-cycle` was specified on startup, the middle key draws cyclicly in RED, YELLOW, GREEN and BLUE). Pressing the *Middle/Shift* will erase the last measurement made and backup the point counter. Typical output appears as follows where each image has its own associated measurement counter.

```
#1 [b00981.ppx] (267.0,260.0) = 27.0, dist from (0.0,0.0)=0.0
#2 [b00981.ppx] (240.0,268.0) = 47.0, dist from (267.0,260.0)=28.2
#1 [b00661.ppx] (320.0,280.0) = 49.0, dist from (240.0,268.0)=0.0
                .
                .
                .
```

The `define cw` option is used to (re)define the computing window which is an image subregion used as the active area for all picture transforms. Its default value is the size of the image. Invoking this overlays the computing window on top of the image. One defines this window by marking its upper left hand corner with the *left* mouse button and the lower right hand corner with the *right* mouse button. When the computing window is changed, this outline representing it is redrawn. When you are finished with the definition, press the *middle* button to terminate the definition and remove the computing window overlay - if you wish.

Clicking a mouse button with the `print-region` option selected prints the 8-bit pixel value under that point, and the pixels around it in an 18x18 subarray of picture data. The data mode is determined by the mouse button pressed. *Left* for

hex, *Right* for octal and *Middle* for decimal. The following illustrates the subarray output.

```
[mcrew.ppx] Row 97, Col 243:
       234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251
      ------------------------------------------------------------------------
   88 |231 228 230 232 230 229 224 224 228 225 227 227 227 226 223 222 223 223
   89 |225 223 225 228 227 224 223 221 224 225 222 224 225 224 226 225 222 222
   90 |226 225 222 221 222 223 220 220 218 218 221 224 223 221 222 228 223 220
   91 |224 221 222 221 221 222 216 219 216 215 218 220 221 221 220 223 223 225
   92 |219 223 223 221 223 223 225 228 224 223 220 223 223 223 220 223 226 219
   93 |229 228 228 235 234 235 238 235 234 234 232 228 218 223 220 220 222 221
   94 |232 234 236 237 241 242 240 237 235 237 238 234 233 230 226 220 221 219
   95 |236 235 237 241 241 241 240 238 238 234 232 233 235 230 225 225 221 220
   96 |233 235 234 235 237 241 240 241 237 231 228 230 230 226 225 226 219 217
   97 |230 229 228 226 232 234 234 232 234 231 221 223 221 223 224 221 218 219
   98 |233 228 226 224 228 231 227 226 226 222 221 219 218 219 220 219 221 219
   99 |220 220 221 224 220 221 220 225 222 219 219 214 216 219 216 215 215 216
  100 |219 220 223 221 219 218 215 215 217 216 213 216 219 219 218 215 214 211
  101 |221 224 225 225 221 223 222 221 215 218 219 220 221 218 213 210 213 210
  102 |219 221 220 221 223 226 224 225 226 222 220 221 218 214 209 207 206 205
  103 |214 214 213 215 217 216 217 220 221 219 219 216 211 207 206 208 206 202
  104 |210 210 208 211 207 210 209 212 211 212 212 211 208 204 204 202 203 201
  105 |200 207 203 205 206 204 207 207 205 206 207 206 205 204 199 200 205 200
```

Under the `zoom 1X` and the two other zoom `zoom 2X`, `zoom4X` modes, the picture is redisplayed centered around the point where the cursor is clicked. When in a 2 or 4X zoom mode, by clicking repeatedly, it is possible to move to regions near the edges of the window which are not currently visible. Both main images are zoomed at the same cursor position if either one is zoomed. If the point is too close to the edge of the display window, the point may not actually be redisplayed in the center when the new zoom image is redrawn. For speed, the zoom options (`zoom2x` and `zoom 4x`) magnify the picture using pixel replication. Note that the small additional 200x200 pixel active zoom region is 2X that of the current window – so that it is possible to see a small 8X region while in `zoom4x`. Since zoom1X, zoom 2X and zoom 4X are cursor modes, it is not possible to use other cursor modes with them (such as `print-region` or `print-pixel`). You may not save zoomed images - they are only a display option.

`Compare images` will flicker–compare [LemP79a] two zoom windows in the right zoom window. The *Middle* button corresponds to setting the zoom position for the window you are in. The *Left* button increases the display–time/frame (1 to 16) while the *Right* button decreases it. Pressing *Middle/Shift* will print the coordinate pairs of the corresponding points in the two images which were 'visited' previously with the *Middle* key while flickering. If the `-mark` option was selected, the two points are marked in both images with a colored '+'. Pressing *Right/Shift* key deletes the last point measured if any. The prompt area will notify you that you need to mark

the LEFT and/or RIGHT image if you forgot to when you try to flicker. Because **Xpix** is rewriting the display continuously when flickering - it should be run on the X display server system rather than over the network on a separate client system to get better response. Sample measurement (x, y) coordinate pair output is:

```
#3 [b00661.ppx](353,214) [b00981.ppx](338,185)
#4 [b00661.ppx](357,380) [b00981.ppx](347,376)
                  .
                  .
                  .
```

The `measure region` command places you in boundary drawing mode where the mouse is used to clear, enter, erase or complete–and–measure a maximum convex hull outline of a region of the image. The algorithm is described in [LemP79a]. The *Left* button clears any boundary drawn so far. The *Middle* button when pressed draws the convex boundary. A maximum convex hull is a boundary which may be concave on the sides but must be convex or flat on the top and bottom. Furthermore, it may not contain any invaginations. The *Right/Shift* button erases the boundary backwards from the last point drawn. Hint: since it is very slow, draw the region using a polygon set of points to enclose the object of interest. The *Left/Shift* button closes the boundary back to the first point drawn and measures features of the region, then undraws the boundary. Features measured currently include: area, perimeter, sum gray values (or calibrated sum of gray values if specified by the header), and centroid. A typical measurement output report when grayscale is calibrated in OD units (for example, if it is indicated in the PPX image file header) is:

```
#19 area=192 pixels**2 perim=38 pixels density=38.2 +/-1.5 OD dens'=30.5 +/-1.5 OD
mnBkgrdDens= 0.04 OD mnDens=0.19 OD (P*P)/A=7.92
centroid=(257.1,357.6) +/-(2.3,3.7)
M.E.R.=(250:261,354:360) [minD:maxD]=[0.04,1.32] OD
% slice [10:200] region: 18% below, 57% within, 25% above
```

The `define cw` command defines a rectangular region called the *computing window* within an image under which a transform may be applied. The default computing window is the size of the image. Pressing the *Left* button defines the upper left hand corner while the lower right hand corner is defined by the *Right* button. Pressing the *Middle* button will redraw the current computing window. *Middle/Shift* clears the computing window from the image. Pressing *Left/Shift* moves the CW upper left hand corner to the cursor position (preserving its size). Pressing *Right/Shift* moves the CW lower right hand corner to the cursor position (preserving its size).

`Line profile` computes and plots the current horizontal (*Left* button) and vertical (*Right* button) line(s) of picture data at the cursor position as a profile display

in that image. The line of the image where the data was obtained is also indicated in the image by drawing a line over it. Multiple lines may be plotted by performing the operation repeatedly. Pressing the *Middle* button will clear the profile(s). Pressing *Left/Shift* or *Right/Shift* will print the profile data instead of plotting it.

`CW profile` computes and plots the current horizontal (*Left* button) and vertical (*Right* button) computing window region picture data at the cursor position as a profile display in that image. The computing window where the image data was obtained is also indicated in the image by drawing a rectangle over it. Multiple regions may be plotted by performing the operation repeatedly. Pressing the *Middle* button will clear the profile(s). Pressing *Left/Shift* or */Right/Shift* as well will print the profile instead of plotting it. The CW profile data is similar to line profile but instead of data taken from a line, the corresponding CW width (X direction) or height (Y direction) is summed. The upper left hand corner of the CW, previously defined with `define cw`, is moved to the position defined by the mouse.

`Draw text` prompts for a text string in the main X-windows `xterm` window and then draws it in the image. The Left button is WHITE, middle is RED (or cycle colors if set), and the right is BLACK. `Shift/RIGHT` erases the last text drawn. This is useful for labeling images and can be saved as a GraphScale image (see `Add Im2 to PPX` command.

MENU: 2. CURSOR operations

`print region` print 18x18 pixel region around the cursor.

`zoom 1X` zoom a 512x512 window 1X image at the cursor position in main image.

`zoom 2X` zoom a 2X image at the cursor position in main image.

`zoom 4X` zoom a 4X image at the cursor position in main image.

`compare images` compare two zoomed images.

`print pixel` print cursor position (x,y,grayvalue,distance) using current calibration.

`measure region` draw convex boundary region and measure features using current calibration.

`line profile` draw X or Y line intensity-profile at cursor.

`CW profile` draw X or Y Computing Window intensity-profile at cursor.

`define CW` define Computing Window used in picture transforms.

`draw text` add text to image at cursor position.

`cursor Off` reset the cursor to brightness/contrast control from previous cursor mode.

4.3.3 SCALING operators menu

The third menu selects one of several image display scaling functions used to map the 8-bit intensities of the picture data into the n-bit intensities, $n < 8$, loaded into the X display server memory. The `full` range linear mapping forces a linear mapping over the complete gray scale range of [*minGray*:*maxGray*] ([0:255] for 8-bit data). Alternatively `linear` mode contrast enhances the image based on a sample histogram of five small subregions of the picture (it may miss some pixels which are then mapped to the X window background color). The `sqrt` and `log` are useful to de-emphasize contrast between background and bright pixels. After recalculating the mapping, the entire picture will be redrawn with the new pixels in the color display. `grid` superimposes a 20x20 pixel grid on the image to allow you to make rough measurements.

`Calibrate X Y G` lets you assign a calibration independently to x, y and grayvalue which is used in data printouts and feature calculations. You may assign a *name, slope* and *intercept* to each of the three quantities or read explicit calibration data in from separate files. The file data format is a one line calibration unit name (e.g. "inches"). This is followed by a list of values: *ncol* values for x (left to right), *nrow* values for y (top to bottom) and $maxGray - minGray$ values for g. The values may be integer or real numbers.

`Histogram` recomputes the gray scale histogram of the current picture under the computing window and displays it as an overlay in the same image. The (min, max, mode, max-peak) histogram values are also reset to the new values which may be checked using the `status` option.

`Add Img to PPX` adds the current image color graphics to picture making GraphScale picture in memory. You must explicitly `write Ij` the picture to put it into a file. It processes the picture as follows: `(a)` clip all pixels > 245 in the "picture" to 245, `(b)` map color pixels in the associated "image" to 246 (RED), 247 (ORANGE), etc. It also notes that this is now a GraphScale picture so that if you write it out, it will be indicated in the PPX file header as such so that other programs can detect it.

MENU: 3. SCALING operations

`full range` use full *minGray* to *maxGray* dynamic range - no enhancement.

`linear` use contrast enhanced linear scaling (default).

`sqrt` use square-root (pixel value) contrast enhanced scaling.

`log` use Log (pixel value) contrast enhance scaling.

`grid` draw 20x20 pixel calibration grid over the image.

`calibrate X Y G` calibrate x, y and grayscale in problem domain specific units.

`histogram` draw histogram of picture as an overlay on the image.

`Add Img to PPX` add current image color graphics to picture making GraphScale picture.

4.3.4 UNARY operators menu

The fourth menu consists of unary picture operators to be applied to the original picture I1 and the result stored in output picture I3. Picture transforms are applied only under the *computing window*. `Threshold` sets picture display thresholds *Thresh1* and *Thresh2* which are then used to map gray values outside of the range *[Thresh1:Thresh2]* to *minGray*. When writing an image I_j, it will ask you if you want to change the name of the file if it already exists - to protect destroying an original image file. The `laser` operator dumps the picture into a Postscript file (with a generated file name and *.ps* file extension). It also will print this file on a laser printer connected to the system if set up to do so when **Xpix** is compiled.

MENU: 4. UNARY operators

`read file⇒`I_j read a new picture file into I_j.

`write `I_j`⇒file` write I_j into a new picture file.

`laser `I_j`⇒ .ps file` write I_j into a Postscript file and print it.

`copy `I_j`⇒`I_k copy Picture I_j into I_k.

`complement `$I1$`⇒`$I3$ complement picture.

`threshold I1⇒I3` request *Thresh1*, *Thresh2* then slice pixels out of range to *minGray*.

`grad4 I1⇒I3` compute 3x3 4-neighbor gradient.

`grad8 I1⇒I3` compute 3x3 8-neighbor gradient.

`laplacian I1⇒I3` compute the 3x3 Laplacian.

`avg8 I1⇒I3` compute 3x3 neighbor average.

`smooth3x3 I1⇒I3` compute 3x3 smoothing convolution.

`median I1⇒I3` compute 3x3 median.

`def external Unary Opr` define external unary operator UOP.

`eval UOP I1⇒I3` eval external unary operator.

4.3.5 BINARY operators menu

The fifth menu consists of binary picture operators which operate on pictures I1 and I2 and save the result in I3. Picture transforms are applied only under the *computing window*. If I2 is a constant, specified when `I2` is set to 0, then the binary operation is performed using that constant value for the second picture as if it were a constant valued picture.

MENU: 5. BINARY operators

`add(I1,I2)⇒I3`	Compute sum of two pictures.
`sub(I1,I2)⇒I3`	Compute difference of two pictures.
`max(I1,I2)⇒I3`	Compute Max of two pictures.
`min(I1,I2)⇒I3`	Compute Min of two pictures.
`and(I1,I2)⇒I3`	Compute And of two pictures.
`or(I1,I2)⇒I3`	Compute Or of two pictures.
`xor(I1,I2)⇒I3`	Compute Xor of two pictures.
`def external Binary Opr`	define external binary operator BOP.
`eval I1 BOP I2⇒I3`	eval external binary operator.

4.4 UNIX command line options

Besides the options listed below, the usual menu and X-Window options as documented in the **X** and **XMenu** manual pages are available, either as inputs to the command line or in the `.Xdefaults` file. The following are startup options available when **Xpix** is invoked from the UNIX shell.

`-append` *logFile*

or

`-a` *logFile*

Append all measurement printout to the specified *logFile* instead of *stdout*. All errors go to *stderr* and all menu prompts to *stdout*.

`-background` *n*

or

`-bg` *n*

Set the default window background color.

`-bd` *n*
or
`-color` *n*
Set the default window border color.

`-border` *n*
or
`-bw` *n*
Set the default window border width.

`-calibrate` *fileX fileY fileG*
or
`-cal` *fileX fileY fileG*
Set the default calibration from the three calibration files. See `calibrate` command for file format details. Otherwise, you must invoke the `calibrate` command interactively.

`-calibrate` *ppx*
or
`-cal` *ppx*
If `-cal` is specified along with the `ppx` suboption for standard PPX files, then the calibration wedge in the PPX file header is used to set up a standard calibration.

`-cycle`
or
`-cyc`
Turn on the graphic color cycling so that each time a graphic gets generated, it cycles through the colors RED, YELLOW, GREEN and BLUE instead of just RED. Switching cursors resets it to RED.

`-dbug`
Turn on the *debug* toggle switch.

`-dolargeimage top` *n*
or
`-dolargeimage bottom` *n*
Set the initial window size to a large window displaying either the `top` or `bottom`.

`-full`
Use the full image dynamic range when doing the contrast enhancement rather than the estimated range. This is useful if you expect the estimation to be incorrect.

`-gra`
or
`-graphscale`
Start with GraphScale mode rather than grayscale.

`-hf`
or
`-hflip`
or
`-horizflip`
Flip input images about the horizontal axis when they are read.

`-las` *lpr-printer-name*
or
`-laserprinter` *lpr-printer-name*
Define the default PostScript laser printer which is used for dumping images.

`-log`
or
`-ln`
Default to a logarithmic mapping of grayscale on startup instead of linear enhanced contrast.

`-mark`
or
`-m`
or
`-mark` *letter*
Default the use of `print pixel` to also mark a '+' in the image. This is useful for indicating which points were measured in the image. If `-mark` *letter* is specified, then the measurement numbers are printed as sequential letters A, B, C, ...Z (modulo 26) instead of numbers (good of course only up to 26).

`-mouse`
Default to continuously print the (*fileName,x,y,g*) value at the same line in the *stdout* terminal window each time the mouse is moved. This is useful for checking different regions of an image.

`-neg`
or
`-n`
Default to inverse video (the negative image) instead of normal video on startup. Normal video interprets pixel data as *minGray* for white and *maxGray* for black.

`-planes` n
Allocate n planes to the display image which will then use 2^n colors or gray levels. The default is to let X windows determine this (which works fine in general).

`-postI3`
or

`-po`
Displays I3 in the left image if you do any UNARY or BINARY operations.

`-ppx`
Use Portable PiXture format, PPX, (defined by the *ppxHdr_t* typedef struct in file `ppxfmt.h`). [Currently, the PPX format defaults to a 512x512x8-bit data section with no header. *This will change to arbitrary size images with a header as the default later.*] The default is a 512x512x8-bit gray scale images (0=white, 255=black) raster image without a header. This may be overridden with the `-skip` option (see below). Using the `-ppx` option will cause the image size to be obtained from the PPX file header so that images larger than 512x512 can be read. They will be sampled down to a 512x512. This means that if the picture is written out again, the new image will have a 512x512 pixel size.

`-rgb`
Default to red-green-blue color selection instead of gray scale on startup.

`-skip` *hbytes* `-nrows` *nr*`-ncols` *nc*`-nbits` *nb*
Ignore the picture file header. Don't read image size information from the header and skip *hbytes* bytes of information at the beginning of the file instead. The image is then defined to *nc* columns by *nr* rows of *nb*-bit pixels. This option can be used to read and display images in 'foreign' formats. The *nb* can be used for formats different from the default 8-bits. Note: alternate switch forms are: `-sk, -nr, -nc,` and `-nb`.

`-sqrt`
or
`-sq`
Default to square root grayscale mapping on startup *instead* of linear.

`-threshold` *thresh1 thresh2*
Use [*thresh1:thresh2*] as the threshold levels for slicing the picture data instead of determining them interactively when using the `threshold` operation. Pixels outside of this range are set to *minGray* when the `threshold` menu command is invoked.

`-silent`
do not print output on the terminal.

`-two 0`
or
`-two optional-picture-file`
Create a second display window on the left. The optional picture file is specified, then load it into I3. Otherwise, if the optional picture file name is `0` (zero) then create and display a white image in I2. This window can then be used to view I2 or I3 (default is I2) while the default right display window is normally assigned to I1. If two image file names are specifies, the `-two` switch is automatically forced on.

`-vf`
or
`-vflip`
or
`-vertflip`
Flip input images about the vertical axis when they are read.

`-w`
or
`-whitegraphscale`
Start with GraphScale mode rather than grayscale, but map *all* colored pixels to white. This is useful if you wish to photograph the graphscale image with black and white film..

`-zoomRight`
or
`-z`
Position of the two little zoom windows to be on the right so as to not interfere with the `xterm` window in the upper left hand corner of the screen.

`=`*Xsize*`x`*Ysize*`+`*Xorig*`+`*Yorg*
Set the geometry for the Right X window to be created. The default is "=512x512+513+340" which is optimized for a sun with a 900 lines screen. For a system with fewer lines N, specify the geometry as "=512x512+513+M" where M=(340+900-N). Eg. A microVax-II/GPX is 1024x860, so the geometry would be "=512x512+512+300". The (0,0) origin is in the upper left hand corner. Note that the size specifications can be omitted and just the origin specified eg. (upper left hand corner) '=+0+0'. Alternatively, the size alone could be specified '=512x512'.

hostName`:`*displayNumber*
Set the display and host to receive the X windows. Eg. *opus:0* - is host *opus* and it's display number set to *0*. Note that the host must have previously run `xhost +your-host-name` to allow it to accept window data from your machine. Also, if you are using `compare images` across a network - response will probably be too slow to be useful.

displaying elsewhere

4.5 Examples for invoking Xpix

We give several examples here to help you get started.

```
Xpix mcrew.ppx
```
Display the image of *mcrew.ppx* in the standard geometry which puts in the lower right hand corner of the screen so that it does not overlap the Xterm window which invoked it. The Xterm window is used to dump the 18x18 grid, other numeric data and status reports and so should be completely visible.

```
Xpix -two mcrew.ppx boys.ppx
```
or
```
Xpix -two mcrew boys
```
or
```
Xpix mcrew boys
```
The first example defaults to the second if you forget to type the *.ppx* file extensions. The `-two` is optional. Display two images *boys.ppx* in the right window and *mcrew.ppx* in the left window.

```
Xpix -postI3 mcrew.ppx
```
Display two images. The one you requested and a blank image for I3 which will be computed.

```
Xpix -full -zoomRight -two b00661 b00981
```
Display two images, but place the zoom windows to the right - out of harms way of the `xterm` window which is in the upper left hand corner.

```
Xpix -mark -append data.log -full -two b00661 b00981
```
Display the two images, append all measurement output in the *data.log* file, mark measured point with '+' in the images. Use full contrast when the images come up on the screen.

```
Xpix -mark -cycle 1Dgel.ppx
```
Cycle through the four colors RED, YELLOW, GREEN and BLUE when marking or drawing line or CW profiles in the 1D bands of the image.

```
Xpix -mouse -two -zoomRight -postI3 b00661.ppx b00981.ppx
```
Continuously update and print the (x,y) position and gray value of the picture data at the cursor position as the mouse is moved. Enable I3 to be displayed if you do any UNARY or BINARY operations. Also move the micro zoom windows to the right.

```
Xpix -skip 128 -nr 512 - nc 256 -nb 8 orion:0 b71556
```
Display image of *b71556* on X-Window display 0 of host *orion*. Skip the 128 byte header. The image is organized as 512 rows of 256 columns.

```
Xpix =256x256+0+0 a01022.ppx
```
Display the image of *a01022.ppx* in a 256x256 window in the upper left hand corner (0,0).

```
Xpix b00661.ppx -ppx
```
Display the image of *b00661.ppx* reading the PPX header which includes gray scale calibration wedge data as well as a conversion units specification. This is then used to preset the grayscale calibration so that `measure region` will integrate pixels in the appropriate units.

```
Xpix b00661.ppx -ppx
```
Display the image of *b00661.ppx* reading the PPX header which includes gray scale calibration wedge data as well as a conversion units specification. This is then used

to preset the grayscale calibration so that `measure region` will integrate pixels in the appropriate units.

```
Xpix g00661.ppx -ppx -graphscale
```

Display the image of *b00661.ppx* in `graphscale` mode.

4.6 Using the menu to control Xpix

typical user interaction

A typical interactive session consists of starting **Xpix** with one or two images. When you first start up, you may decide to increase the contrast/brightness of the image by pressing the mosue and moving it until you are satisfied with the image. Then you might performs a series of image operations. These operations are either Unary or Binary operations which redefine the value of the picture assigned to I3. Eventually, you write the computed image out to the disk to save it. You can also make measurements on the images or their transforms and optionally output the measurement data to a log file.

4.7 Startup file options

Xpix startup options can be placed in the universal `.Xdefaults` startup file in your home directory. The options include:

Xpix.Append=`logFile`
Xpix.Background=`nnn`
Xpix.BorderWidth=`nnn`
Xpix.BorderColor=`nnn`
Xpix.CalibrateFiles=`fileX,fileY,fileG`
Xpix.CycleColor
Xpix.FullRange
Xpix.HelpDVI=`/user/joeUser/bin`
Xpix.Hflip
Xpix.Log
Xpix.Mark=*optional* `letter`
Xpix.MousePosition
Xpix.Neg
Xpix.Nbits=`nnn`
Xpix.Nplanes=`n`
Xpix.Ncols=`nnn`
Xpix.Nrows=`nnn`
Xpix.PostI3
Xpix.Ppx
Xpix.PrintLaser=`laser`

Xpix.Rgb
Xpix.Skip=nnn
Xpix.Sqrt
Xpix.Threshold=nnn,nnn
Xpix.TwoImages=image2
Xpix.Vflip
Xpix.ZoomRight

4.8 Current Xpix Caveats

The following caveat list reflect the current **Xpix** status.

- The `def external` `Unary/Binary Opr` and `eval` `UOP/BOP` are not fully implemented.
- Because of conflicts with the CONTROL-KEY/Mouse-buttons, it is difficult to run **Xpix** successfully with `uwm` window manager. The X10R4 version only runs smoothly with `menuwm` (a simpler human interface). The X11R4 version works nicely with `twm` window manager. It needs to be generalized for any window manager.
- Image *display* size is currently restricted to 512x512x8-bit - although it can read and sample larger PPX input image files into a 512x512 image.
- The PPX file header is currently ignored while reading with 512x512 headerless image files unless the `-ppx` switch i specified. Also the `-skip -nrows -ncols -nbits` are not fully functional. Using the `-ppx` switch forces the PPX file header to be read. Although larger images (eg 1024x1024) may be read, they are sampled down to a 512x512.
- `measure region` works but is very slow because of the way we are currently posting image changes. Hint: define the object using a minimum sided polygon approximation.
- Calibration prompt and data lookup conversion do not work yet.
- The `calibrate X Y G` command to calibrate x, y and grayscale in problem domain specific units does not work. Either of course does the switch option '`-cal` *fileX fileY fileG*' does not work.
- `print-pixel`, `compare images` have a cursor jitter with long delay when one make a mark from the left window but not the right. [This seems to be an X-Window bug since is found in other applications as well.]

- Continuous flicker should be available in `compare images`. The problem is that flicker does not start until the button is released because of the overall double-click event model. So you must click it repeatedly to continue the flicker.
- There is currently no UNIX *man* file documentation. Use *xdvi(1)* instead.
- `Xpix -postI3` *PPX-file-name* does not work for a single image. You need the a second image for now.

4.9 Other associated files used by Xpix

The following run time files are used with **Xpix**.

twm - X11 window manager (optional).

.twmrc - startup file for X11 *twm* window manager.

xdvi - X windows dvi (from TeX) file previewer (optional).

Chapter 5

Exploratory Data Analysis Methods

As mentioned in the Introduction, GELLAB-II is an integrated collection of computer programs for the exploratory data analysis of multiple 2D electrophoretic gels. This chapter discusses specific areas for data exploration within the context of 2D gel analysis. As Tukey [TukJ86] and others have suggested, the use of visual feedback is essential in exploratory data analysis. We have available a number of tools, both visual and numeric, with which to dissect a set of data and we will be discussing some of them here. The following material on exploratory data analysis is derived from [LemP89a] which discusses common paradigms for exploratory data analysis on 2D gel data. The discussion later in this Chapter dealing with search strategies in GELLAB-II is derived from [LemP83b]. It is keyed to tutorial examples from Chapter 2. Section 5.1, page 503 discusses search strategies using **cgelp2**. Finally, Section 5.2.1, page 533 lists some rules for helping run GELLAB-II.

exploration tools

As noted in Section 1.3 on page 25, by cleverly comparing the data inherent in 2D gels from different but related samples, we can begin to discern the organization of genetic expression - which genes are expressed coordinately during a biologic process and the sequence of changes in their expression. Two problems arise however:

what do I do?

1. The large quantity and complexity of the data ("it's too complicated, I want a simple problem!"), and

2. The lack of a relationship between the 2D gel data and traditional biochemistry ("What are all those spots anyway?").

The solution suggested was to create and analyze a computerized database. For this all of the computerized 2D gel analysis systems face a common set of problems:

1. Record study data and digitize corresponding 2D gel images.
2. Locate and quantify the component spots in each image.
3. Map the data from one image to the locations of homologous data in one or more similar images in an experimental series.
4. Create a unified database suitable for analysis. Such a database has, of necessity, a structure analogous to a 3D stack of 2D gels, thus implying certain practical constraints on data manipulation as well as the possibility of analysis of the stack from various points of view (by gels, by individual spots, by groups of spots, by external ordering of the stack, etc.).
5. Develop an analytic strategy for disecting and finding patterns in the composite gel database.
6. Display results of analysis with derived (marked) images, graphs and tables. This would constitute a "report" of one "view" of the gel database. Additional view amy also be investigated.

In the course of this, problems such as integrating data external to the gel image (e.g. sample source, time sequence in an experiment, etc.), correction of digitization noise and other errors, and normalization of quantitative data must also be addressed. While important and potentially instructive differences between various systems for analyzing 2D gels exist, we believe that the *fundamental similarity imposed by this hierarchy of data structures* is more significant.

Analytic strategies likewise have a *common domain of methods.* Fisher et al. [FisD86] suggest a definition that is appropriate for 2D gel databases: "Exploratory data analysis can be characterized as a search for *regularity* or *structure* among objects in an environment, and the subsequent *interpretation* of discovered regularity."

As discussed in Section 1.3, computerized gel databases permit tracking and correlating a number of biological relationships including: 1) identifying individual *protein markers* for a biologic process; 2) identifying a set of quantitative *protein markers* for a biologic process; 3) comparison of *sets of proteins* to infer putative *key gene products*; 4) development of an annotated *catalog of proteins* for use with other biologic data; and 5) cluster analysis of gel patterns wtih *other* biologic data.

sets of proteins

The larger problem alluded to earlier of relating 2D gel data to other forms of biologic data is more difficult. Ideally the protein databases can provide a scaffolding upon which more traditional biochemical information can be arranged, thus relating function to the underlying structure (proteins involved). The most obvious example of this sort is the capacity of 2D gel database analysis to identify sets of proteins which are coordinately regulated during differentiation or other biological processes. In order to achieve this coordination, proteins in such sets must share

common regulatory mechanisms. The recent explosion of knowledge about classes of DNA binding proteins provides examples of such common regulatory mechanisms operative upon groups of genes. The problem is that most 'spots' on 2D gels are unidentified and most proteins which have been isolated and studied functionally have not been located on a 2D gel map. Furthermore, there is as yet no universal 2D gel map of proteins, transferable from lab to lab, and allowing meaningful communication based on the location of proteins in the map. Such a solution to the current 'Tower of Babel' in the 2D gel world is clearly feasible but technically non-trivial.

Therefore, we need to apply these methods in ways which will clarify the inherent patterns expected to be present in the data. The next two subsections discuss some of the details for generating these different views of the database with the hope that some of these patterns will emerge.

5.1 Search strategies using cgelp2

The following material is derived from the paper "2D Electophoresis gel database analysis: Aspects of data structures and search strategies in GELLAB" [LemP83b].

Search strategies for finding spot differences among multiple two dimensional polyacrylamide electrophoresis gels are discussed in the context of the GELLAB spot database management system. A 2D gel experiment should have a well defined biological experimental and preparation protocol reflecting the hypotheses of the problem. So too should the analysis of its corresponding 2D gel computer spot data base have a protocol. This protocol is heavily influenced by the nature of the biological experiment as well as 2D gel preparation considerations including the realities of artifactual and systematic noise. It is further influenced by constraints due to computational considerations. The search strategy is that part of the analysis protocol in which an investigator iteratively defines tests to find significant spot differences. One goal of designing a well thought out search protocol is to reduce the number of search iterations required. Aspects of some requirements and constraints for useful search strategies are discussed.

iterate finding spot diffs.

5.1.1 Basis for 2D gel DB search

When performing an experiment entailing the use of two-dimensional polyacrylamide electrophoretic gels (PAGE) as a tool , care is taken in designing the biological and 2D gel preparation protocols. Similarly, a carefully defined gel database *analysis* protocol is also essential, especially in the case of large multidimensional databases for effective analysis.

Because of the great analytical power of GELLAB-II and similar 2D gel analysis systems, particular attention needs to be focused on the biological model(s) for spot

changes suggested by investigators. Another way of thinking about this process is in terms of defining a *search protocol* that will result in significant spot differences being discovered.

search protocol

Many of the GELLAB search protocols were derived and implemented as a result of the requirements of biological protocols brought to our attention through the collaboration of GELLAB users. Figure 5.1 illustrates the general analysis process.

Given a set of gel images (derived from autoradiographs, photographs or stained gels themselves) one can construct a composite gel database (CGL DB) using the procedure illustrated in Figure 5.2 and discussed in Section 3.3.

Briefly reiterating the definitions given in Section 1.3, a representative gel or *Rgel* (used as a reference gel) is selected from the set of gels in the experiment for inter-gel alignment purposes. Each of the other gels is aligned with the Rgel at a set of manually defined spots called *landmark* spots. All of the previously segmented spots in each gel are then automatically paired between the Rgel and other gels. The **cgelp2** program is then used to construct and manipulate the Paged Composite Gel (PCG) DB which groups corresponding spots from different gels together in sets called *Rspot* sets. Spots missing in the Rgel but present in other gels are extrapolated into the Rgel and denoted as *eRspot* sets so that all spots found in all gels are included in the composite gel DB. Spots missing from any gel can also be extrapolated so that *all* spot positions can be found in the PCG DB for all gels. The term *paged* means that the PCG DB is brought into and out of computer main memory from a very large disk file in small chunks called 'pages'. This permits constructing and maintaining a very large multiple-gel DB consisting of many gels with many spots. Typical PCG databases can consist of 10 to over 100 gels of over 1000 spots/gel).

Finally, spot differences are *visualized* and then manually *verified* by creating derived images called *Rmaps* and *mosaics* using the **markgel** and **mosaic** programs respectively. A Rmap is an image of one of the gels in the database overlayed with selected Rspot labels. A Mosaic image is composed of panels of subregions of all of the gel images surrounding a particular Rspot and are ordered by increasing spot total optical density.[1] Every 16 panels are grouped into one image as a 4x4 array and therefore multiple images are generated for databases with more than 16 gels. Alternatively, Rmap and mosaic plots can be generated and viewed using the `RMAP` and `MOSAIC` <CMD>>s in **cgelp2**. Many examples of these mosaics have appeared in previous GELLAB papers.

visualize spot differences

In order to better understand the computational implications of the search strategies to be discussed, some understanding of the PCG DB structure and mechanisms to manipulate it is helpful. In addition, some of the key data structures used

[1] That is, the gel panel in the upper left hand corner has the lowest protein concentration for that Rspot while the panel in the lower right hand corner has the highest.

in **cgelp2** are mentioned. Finally, we concentrate on the basis for and derivation of search strategies for a 2D gel database analysis.

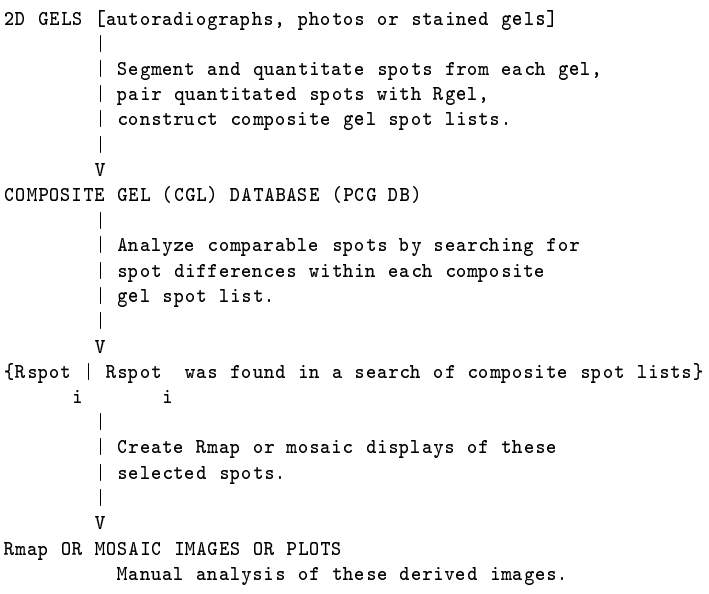


Figure 5.1. General overview of 2D gel analysis process. Gels are scanned by the computer after which a composite spot database of corresponding spots across gels is constructed. The CGL DB is searched for spot differences. These are then visualized on derived Rmap and mosaic images for manual verification for true and false positive differences.

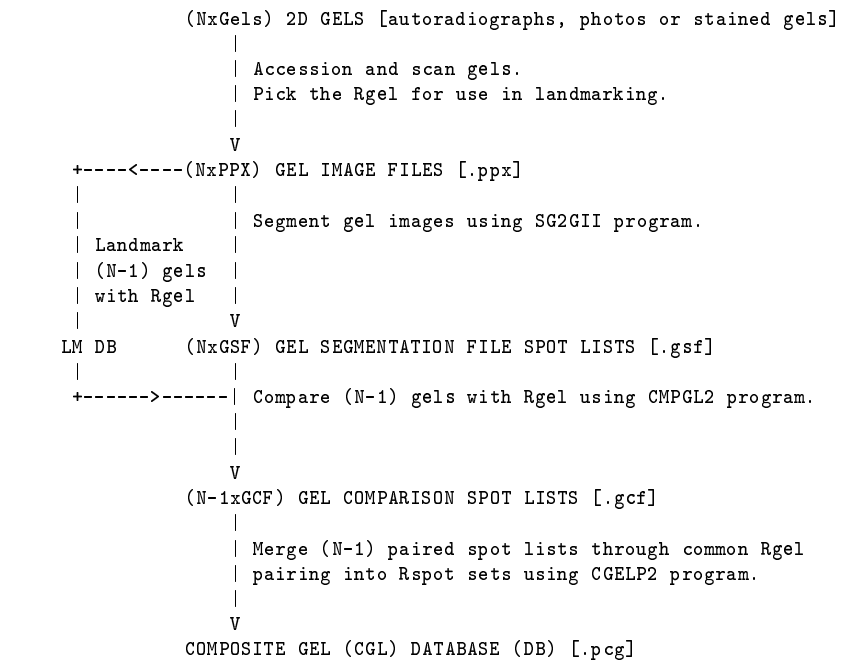


Figure 5.2. Overview of the process for constructing a composite gel database (CGL). The practical computer version is a "paged" CGL or PCG DB. The algorithms for these processing stages are described in detail in the papers for **sg2gii**, **cmpgl2** and **cgelp2**. The instructions for running these programs are given in the Chapter 3 descriptions.

5.1.2 GELLAB-II as a DataBase Management System

GELLAB-II is a complete database management system (DBMS) for 2D electrophoretic gel analysis within a given laboratory. It is designed primarily to compare gels in a single set of gel experiments. However, it can be used for comparing 2D gel data among gel sets of different gel experiments in the same or different laboratories using the estimate of the canonical gel or Cgel′ concept discussed in [LemP82a] and on page 192 or alternatively using spot mapping files (see `SET MAPPING FILE` page **??**). The overall utility of the latter depends to a great extent on the reproducibility of sets of gels over long periods of time and between laboratories.

It is useful to have a consistent PCG DB spot numbering (which we term a *Rspot number*) on a particular type of material which is used in several different databases. This can be implemented several ways. In the first method, the same Rgel GSF file can be used. In the second, the Rgel can be selected to be a previously computed Cgel′′s GSF. Then all landmarking and spot pairing is done using this synthetic gel.

spot numbering

When the PCG DB is to be searched, the Cgel′ may be then be removed from the working set of gels so as not to enter into any of the searches. However, its presence imposes a specific Rspot numbering on the set of gels. The other method involves using foreign spot mapping files (again, see `SET MAPPING FILE` command) which impose a logical spot numbering on any PCG DB. The issue then turns to one of whether to use Cgel′ numbered databases or mapping files. Cgel′ and Foreign spot mapping will not be discussed here. Rather, we will concern ourselves with the problem of searching a large multiple gel spot DB in an efficient manner.

searching composite gel DB

Internal DBMS

In the literature, formalisms for coherently manipulating large quantities of data are called database management systems or DBMS. Baker [BakR82] defines a DBMS as a software facility used by one or more programs to access the same DB (possibly in different ways) as well as to protect it from unauthorized access or changes. When the quantity of data becomes too large to keep completely in the computer main memory or the program too complex, then a *central* data control facility greatly facilitates system design and implementation. This is *not* to say that the entire process could not be handled by a program kludge (i.e. an ad-hoc creation which is not recommended as good programming practice).

As pointed out by Baker, there are two classes of DBMS: internal (IDBM) and external (EDBM). **Cgelp2**, the DBMS of GELLAB-II, is an internal DBMS. Internal DBMS are characterized by a technique and not by a separate program (although this technique may be implemented as a set of procedures included in application software as is the case with **cgelp2**).

In such a system, data resides in: (1) disk files, (2) in procedure areas of core memory, and (3) global areas of core (i.e. Random Access Memory (RAM)) memory and "windows" into the disk files. All of these areas are under control of the systems' programmer.

An IDBM passes data directly from one procedure to another in a *global* manner through the intermediary of global *state* variables, thus encouraging the *consistent* use of these global variables. In **cgelp2** this is implemented by the use of a consistent set of global *state* variables and a set of disk data base paging procedures completely transparent to the user.

The actual disk PCG DB file in GELLAB-II logically consists of two parts. The first part being the global state variables header (which functionally may be thought of as the **cgelp2** user state). It is updated or checkpointed on the disk *only* when directed to be done so by the user. It is read from the disk *only* when first instantiating a particular database with **cgelp2** in a particular user interactive session.

PCG DB layout

The second part of the DB file is the actual Rspot sets (a set of corresponding spots in multiple gels) feature data. Each Rspot set is allocated a *fixed* amount

of sequential disk space such that for a given number of gels, there is sufficient space to store a spot for each gel and also up to 2 to 3 ambiguous pair (AP label) spots. (Ambiguous spots are additional spots or spot fragments associated with corresponding spots - see ([LemP81b], [LemP83b], [LemP82a]) and Appendix G for a more detailed discussion.) By being able to compute a Rspot's disk address, the Rspot set data can be retrieved directly. Furthermore, by storing Rspot sets sequentially in the PCG DB file, sequential searches on all Rspot sets can be performed at maximum efficiency.

If in constructing or expanding a particular PCG DB it is found that there is insufficient space in which to store a particular Rspot set, then **cgelp2** will automatically expand this space for *all* Rspot sets by shuffling the data to an extension of the file. Each Rspot set is reallocated as a larger contiguous subregion in the file. The shuffling of the file is done prior to continuation of the process which caused the problem. Because this is a time consuming process that is hopefully rarely invoked, we have empirically selected a rather large AP factor of 2 to 3 spots/gel which insures that this expansion procedure will almost never occur under usual operation. [2] If a large number of gels is added at a later time to an existing PCG DB, then the automatic expansion algorithm might be automatically invoked in order to make room for the additional gels.

expanding PCG DB

Since the offset of the start of each Rspot set can be quickly computed, we can then "page" any Rspot set in or out of RAM memory. In practice, because consecutively numbered Rspots sets will generally be needed during a sequential search through the PCG DB, a number of these Rspots sets are paged at one time (i.e. as many as fit into a large core memory buffer). The number of Rspots paged at one time is a function of the number of gels in the database. This is a reasonably efficient procedure since the cost of moving the disk head is much greater in terms of in time than actually performing the data transfer. An example of this is shown in Section 2.3 Example 7, page 109. For an example of creating, saving, reloading and checkpointing the new PCG DB see Section 2.3 Examples 1, page 106, and 2, page 106.

"paging" the DB

Data Structures in CGELP2 as seen by the user

Although not essential to performing an exploratory data analysis, a basic understanding of some of the underlying "tools" will help in suggesting areas to explore in the data analysis. Some of the special software data structures which support high level record keeping and the search aspects of the PCG DB are: sets, inverted lists, records, Fspot maps, spot annotation, strings and associative table lookup. Because of lack of space we will not go into the details of these data structures and their use.

[2] For our 115 gel leukemia PCG DB, it has occurred two times when adding gels and once when extrapolating eRspots. The longest time it took was on the order of an hour on the slower DEC-2020.

sets

Sets are implemented as bit arrays and are used for manipulating subsets of gels and Rspots. In the current 32-bit/word or larger generation of computers, it is possible to store 32 potential elements of the set in one word. A '0' bit value indicates no membership and a '1' bit value indicates membership. In many cases, intersection, union and set difference between 32 possible pre-assigned subset elements may be performed in one computer hardware instruction. That is, set intersection is a logical AND instruction, union is an inclusive OR and subtraction is the logical AND of one set with the 1's complement of the set. In GELLAB, to implement a **cgelp2** set of 3072 *possible* Rspots requires only 96 32-bit words! Thus many sets can be kept in core memory for efficient and rapid processing.

For examples of *SRL subset operations* see Section 2.3 Examples 31, page 121; 32, page 121; 47, page 126; 48, page 126; 49, page 127; and 50, page 127;

For examples of *gel subset operations* see Section 2.3 Examples 11, page 110; 18, page 114; 19, page 115; and 41, page 124.

associative processing

Inverted lists are used internally in **cgelp2** for mapping (x,y) positions in Rgel coordinate space to Rspot sets numbers. This associative mapping is used both during PCG DB creation and in order to map non-Rgel unresolved spots to extrapolated or eRspot sets using a landmark transformation [LemP82a]. This latter algorithm permits finding the Rspot set closest to a specified x,y coordinate pair.

list of sets

List of sets is a list of SRL subset names indicated by their corresponding subset numbers. Such a List of SRL subsets is called a `LOS` and is an efficient way for merging the results of many PCG DB searches and/or SRL subset operations.

records

Records are collections of data, called fields, associated with an object which for **cgelp2** is a particular spot in a given gel for a Rspot set. Records are implemented as sub-arrays of the paged PCG DB and contain bit and sub-word fields. Within a particular Rspot set, records are ordered by a 'successor' linked list pointer field of each record. Thus the Rspot set can be maintained as a rank ordered (by spot total integrated density) list.

foreign spots

Fspot map is a spot number mapping from (or to) a foreign spot numbering (See `SET MAPPING FILE` page **??**) to (or from) the internal **cgelp2** Rspot numbering. It can be used to map spot numbers between databases of different 2D gel analysis systems as well as between different GELLAB-II PCG DBs.

annotation

Annotation lists are bit-sets associated with each Rspot set (See `SET ANNOTATION` page 263). Currently, they are of size 64 (with 32 Rspot and 32 Fspot possible annotations). Each of the annotations has an associated annotation string. These annotation sets can be exchanged with the SRL subsets for performing additional operations. Each SRL subset and gel subset also has an associated annotation title. This title can be searched associatively for SRL subsets. For examples of SRL subset titles using the `LIST/DIR` subcommand, see Section 2.3 Examples 3, page 107; 26, page 118; 27, page 119; 48, page 126; 49, page 127; and 50, page 127.

Strings and string operations (e.g. substrings, substring searching, number conversion to and from strings, composition of strings, etc.) are used extensively for table and formating presentation, auxiliary gel information, and some data processing.

Associative Table Lookup (i.e. hash functions) are used for mapping gel Accession numbers (which are five digit numbers XXXX.E) into one of MAXGEL internal gel numbers for a given PCG DB. MAXGEL is currently defined as 128 but may be redefined as 512, 1024, etc. when GELLAB-II is recompiled - it can not be changed by the user. This associative table lookup is used extensively in checking to see whether a gel is in a set of gels or not (e.g. the working set of gels, gel subset or gel class.) For a large number of gels, it is much more efficient than checking a linear list of gels. This is critical since every spot (i.e. gel) in every Rspot set must be checked to see whether it is (1) in the working set of gels (see discussion on prefilter to follow), and (2) in a particular gel class if a multiple class test is being performed.

5.1.3 Search strategies under GELLAB-II

In light of the implementation considerations previously discussed, we will discuss some practical search strategies developed under GELLAB and their rationale. We first review the entire 2D gel DB computer analysis process at a high level and then at successively more detailed levels as in Figure 5.2.

Finding and g Removing Marginal Gels from the PCG DB

Marginal gels may be discovered in the PCG DB early in the analysis and temporarily removed from the working set of gels (WS) during statistical searches. For an example, see Section 2.3 Example 11, page 110 for removing a specific gel. Various measures of performance for spot segmentation and gel pairing are available to help make these decisions. This is necessary so as not to incorrectly bias the searches. Later, prior to making mosaic or Rmap derived images, the marginal gels may be added back into the WS for visualization. This permits checking marginal gels on the basis of results found under more robust constraints. Care must be taken if this removal procedure is used so as not to unduly bias which gels are removed (i.e. a particular experimental class of gels may not run well in the PAGE process).

strategies for marginal gels

A marginal gel is one which exhibits one or more of the following characteristics:

- Looks bad visually with very dark background, gross distortions of spots, very heavy clustering, artifacts, etc.
- Poorly segmented as evidenced by one of:
    1. Poorly segmented image (see [LemP81d] and [LemP82a]). Compare the original image with the segmented "`z`" image using '`accppx acc#1 acc#2 -P2:z`' (page 3.1).

2. Very small or very large total number of segmented spots in relation to other gels in the DB. (These total numbers of spots/gel are listed with the `GELS` command (page 216) in **cgelp2** in the PCG DB.) For an example, see Section 2.3 Example 3, page 107. These numbers should be approximately the same for all gels in the database. The following gel segmentation data is computed by **sg2gii** in the GSF file. It is also saved as part of the PCG DB header state.

```
Total of 723 accepted D spots accumulated density=7566.7, area=29367
Total of 676 accepted D' spots accumulated density=5951.6, area=28560
Total of 4261 omitted D spots accumulated density=845.1, area=46771
Omitted/Accepted density =11%
```

3. The ratio of *Total Accepted D'/Total Omitted D'* is large. These numbers and the ratio is computed by **sg2gii** (page 466) at the time the gel is segmented. A low ratio indicates a noisy gel.
4. The gel is very light or very dark in relation to other gels in the DB.

- Has a very large root mean square landmark deviation (using the `VALIDLANDMARKS` command (page 313) in **cgelp2** as well as being computing during landmarking). For an example, see Section 2.3 Example 15, page 112.

- Another more subtle estimate of gel quality is the average deviation of spot position when pairing two gels. This is one of the measures of pairing accuracy. The following calculation is biased to some extent by the LM selection and does not take mispairing into account. Let: N_{G1}^P be the number of spots paired in the Rgel (i.e. G1); N_{G1} and N_{G2} be the number of spots present in gel G1 and G2. Then $\overline{dP}$ is the average deviation of spot position and $\overline{dP}'$ is the normalize average deviation taking the total number of spots into account.

$$\overline{dP} = \frac{\sum_{i \in N_{G1}^P} dP_i}{N_{G1}^P} \tag{5.1}$$

$$\overline{dP}' = \frac{\overline{dP}(N_{G1} + N_{G2})}{2N_{G1}^P} \tag{5.2}$$

- The gel comparison of two gels produces pairing statistics (which includes $\overline{dP}$ and $\overline{dP}'$) as well as the spot pairings. The (SP+PP) pairings should be as high as possible considering the types of gels being paired (i.e. similar numbers of spots or not depending on the type of sample material). Look at the number of US spots to total spots in that gel. High numbers of AP spots are also indicative of either gels with fuzzy spots (causing spot fragmentation) or gels with widely different numbers of spots. The statistics output from

cmpgl2 (illustrated below) include the fraction of (SP+PP) spots as well as the number of AP spots and $\overline{dP}'$.

```
PAIRING STATISTICS
------------------
 After Initial pairing:
US 229
SP 224
PP 488
AP 319
CP 0
0.5(SP+PP)/(|G1| MIN |G2|)=52.7%

 After secondary pairing:
US 212
SP 224
PP 520
AP 304
CP 0
0.5(SP+PP)/(|G1| MIN |G2|)=55.0%

mean dP(SP+PP)=5.28 mean dP'((|G1|+|G2|)/(SP+PP))=8.05
```

- One measure of gel sensitivity is the least quantifiable spot (suggested by Jim Garrels) in the middle molecular weight range of the gel. Doing a **cgelp2** `HISTOGRAM` of Rspot set integrated density will show the distribution from which this type of information can be estimated. For an example, see Section 2.3 Example 40, page 124.

ALGORITHM: RMS List of Landmarks Deviation

global gel distortion

The root mean square landmark deviation is a crude global estimate of gel "quality" and is estimated as using the following algorithm:

[1] Compute the centroid C_g of the landmark set of spots for gel g (actually existing as pairs corresponding of spots in both the Rgel and gel g).

[2] Map the landmark spots in gel g to the domain of the Rgel by the vector transformation: $\mathit{Offset} = C_r - C_g$.

[3] Compute the root mean square sum of the deviations for the set of landmark spots between the Rgel and the transformed gel g landmark spot coordinates.

A sample of `VALIDLANDMARKS` generated data in Table 5.1 illustrates the root mean square landmark deviation values for a typical gel database (72 hour P388D1 cell line [Lip80a]). Note that two of the gels have high values. High values are indicative of either faulty landmarking or greatly distorted gels (with respect to the

Rgel). In the case of faulty landmarking, the relevant gels are then re-landmarked and the database rebuilt. This consistency check is also available at the time the manual interactive landmarking is performed so that corrections can be made immediately.

TABLE 5.1. Global gel similarity estimated by landmark root mean square deviation of a gel. Two estimates of landmark deviation in a gel computed by GELLAB are the valid landmark table and the landmark root mean square deviation estimate. The valid landmark table is used to indicate that a landmark was within the specified distance (typically of the centroid within 3 to 5 pixels) of a spot. If it is not valid, then the user interactively-specified coordinates are used rather than a spot's segmenter computed centroid. At the bottom of the table the number of landmarks as a function of the number of gels is given indicating landmarks which were difficult to landmark. The global estimate of gel deviation is given in the second table as the LM root mean square deviation. Note that the validity table and root mean square deviation metric can be used to indicate that something is wrong with the landmarking of a particular gel or that it is geometrically quite different from the Rgel. In this table, the case where gels 271.2 and 282.1 have very high values (> 100) was traced to faulty landmarking by the operator. Gels 278.1, 279.1, 280.1 and 283.1 had somewhat larger global distortion (~ 15) than the other gels (~ 6) when compared to the Rgel.

```
 Valid landmarks [T is OK (spot exists for landmark), F is NG
 (spot does not exist for landmark) or SM (same spot for several landmarks)
 in set of landmarks validity check]

 GEL   | A B C D E F G H I J K L M N O P Q R S T U V W X Y
-------------------------------------------------------
0269.1| T T T T T F T T T T T T T T F F F F T T
0266.1| T T T T T T T T T T T T T T T F T F T T
0267.1| T T T T T T T T T T T T T T T T T T T T
0268.1| T T T T T T T T T T T T T T T T T T T F
0270.1| T T T T T T T T T T T T T T T F T T T T
0270.2| T T T T T T T T T T T T T T T T T T T T
0271.2| T T T T T T T T T T T T T T T T T T T
0272.2| T T T T T T T T T T T T T T T T T T T T
0273.1| T T T T T T T T T T T T T T T T T T T T
0273.2| T T T T T F T T T T F T T T T F F F F T T
0274.1| T T T T T T T T T T T T T T T T T T T T
0275.1| T T T T T T T T T T T T T T T T T T T T
0276.1| T T T T T T T T T T T T T T T T T T T T
0277.1| T T T T T T T T T T T T T T T T T T T T
0278.1| T T T T T T T T T F T T T T T T T F T T
0279.1| T T T T T T T F T F T T T T T T F T F T F
0280.1| T T T T T F T F T F F T T T T T F F F T F
0281.1| T T T T T T T T T T T T T T T T T T T T
0282.1| T T T T T F T T T T T T T T F F T F T T
0283.1| T T T T T F T T T T T T T T T F F F F T T

Percentage of gels in which landmark is present (T=100%)
-------------------------------------------------------------------------
  A  B  C  D  E  F  G  H  I  J  K  L  M  N  O  P  Q  R  S  T  U  V  W  X  Y
-------------------------------------------------------------------------
  T  T  T  T  T 75  T 90  T 85 90  T  T  T 95 80 65 75 65  T 75 %

Global estimate of LM centroid of gel and RMS deviation from Rgel
-----------------------------------------------------------------
[0269.1] Mean LM centroid (276,167), LM RtMnSqDev from Rgel= .0
[0266.1] Mean LM centroid (258,186), LM RtMnSqDev from Rgel= 4.8
[0267.1] Mean LM centroid (280,195), LM RtMnSqDev from Rgel= 7.3
```

```
[0268.1] Mean LM centroid (263,185), LM RtMnSqDev from Rgel= 8.2
[0270.1] Mean LM centroid (269,185), LM RtMnSqDev from Rgel= 5.9
[0270.2] Mean LM centroid (257,179), LM RtMnSqDev from Rgel= 4.8
[0271.2] Mean LM centroid (262,186), LM RtMnSqDev from Rgel= 112.7
[0272.2] Mean LM centroid (286,185), LM RtMnSqDev from Rgel= 7.0
[0273.1] Mean LM centroid (263,168), LM RtMnSqDev from Rgel= 7.0
[0273.2] Mean LM centroid (301,177), LM RtMnSqDev from Rgel= 4.5
[0274.1] Mean LM centroid (264,186), LM RtMnSqDev from Rgel= 6.7
[0275.1] Mean LM centroid (267,161), LM RtMnSqDev from Rgel= 5.5
[0276.1] Mean LM centroid (260,188), LM RtMnSqDev from Rgel= 6.8
[0277.1] Mean LM centroid (260,180), LM RtMnSqDev from Rgel= 6.2
[0278.1] Mean LM centroid (274,249), LM RtMnSqDev from Rgel= 12.6
[0279.1] Mean LM centroid (266,264), LM RtMnSqDev from Rgel= 15.6
[0280.1] Mean LM centroid (266,255), LM RtMnSqDev from Rgel= 16.3
[0281.1] Mean LM centroid (257,253), LM RtMnSqDev from Rgel= 9.6
[0282.1] Mean LM centroid (262,243), LM RtMnSqDev from Rgel= 110.1
[0283.1] Mean LM centroid (268,264), LM RtMnSqDev from Rgel= 14.5
```

Variability of a PCG DB

PCG DB quality

As just discussed, there are several measures of gel variability which can be investigated in the PCG DB. These also include the coefficient of variation of spot density and area (CVD and CVA) for the Rspot set which can be investigated with the prefilter and the `INQUIRE` command searches. For examples, see Section 2.3 Example 52, page 128. The plot commands `DDPLOT, PLOT` and `HISTOGRAM` allow the global comparisons of the CV distributions. For an example of using histograms, see Section 2.3 Example 40, page 124. The `TABLE` command allows gel-gel and SRL-SRL correlations. For examples, see Section 2.3 Examples 42, page 125 and 43, page 125. If you analyze the gel-gel correlations by within class and outside of class, then estimates of gel variability may be obtained.

5.1.4 Normalization of a PCG DB

density normalization

It is essential that the images comprising a PCG DB be properly calibrated with respect to protein concentration (integrated density) prior to performing a quantitative spot difference search ([LemP81c], [LemP82a], [LemP83a], [LemP83b]) . The calibration is performed using one of several normalization schemes (see `SET DENSITY MODE` page 270 in **cgelp2** to select a mode). For examples in changing the density mode and the effect it has on the view of Rspot set data, see Section 2.3 Examples 13, page 111; 20, page 115; and 21, page 116. Spot data must be calibrated before it can be compared between gels. This is done in three steps, the first two are performed in **sg2gii** and the last in **cgelp2** using one of the following normalization methods.

1. Image gray scale is calibrated in optical density (OD) prior to summing pixel values belonging to the same spot (integrated density).

2. Gel background integrated density is subtracted $D'_i = D_i - A_i D_{background}$.

3. The D'_i is normalized by the particular normalization scheme α function $\mathcal{F}_\alpha$, so $D^\alpha_i = \mathcal{F}_\alpha(D'_i)$.

Some of the possible methods available in **cgelp2** are presented here. For high quality gels with many similar spots, the ratio method is preferred as an estimate of total gel density. In those cases of very different gels where it can not be practically applied, the modified least squares method is used. For gels with very low background, the percent-of-total-density method may be usable. In cases where an exposure decay correction factor (EDF) can be estimated from whole cell extract counts, then the D' correction for exposure might be considered.

Ratio method

1. Set the Rspot feature limits to select robust spots found in all gels in the working set (WS) of gels (i.e. bad gels and gels not of interest are removed from the WS). Typically the feature limits are constrained for [Area, OD range (i.e. $maximumOD - minimumOD$ within a spot), CV area, minimum total integrated density D', and the #gels/Rspot set = size of the WS of gels. The pairing labels are: SP+PP+US (including eRspot DB)]. *mean of list of spots*

2. Do a *index* type search of the PCG DB to compute the mean normalization factor N_g, for all gels g in the WS to find a set of consistent Rspots. D'_{ig} is the background corrected total integrated Optical Density (OD) for Rspot i in gel g. The density mode is forced to `Absolute`, i.e. D', for the computation.

$$N_g = \{(\sum_{Rspot_i \in PREFILTER} D'_{ig})/n_g\} \tag{5.3}$$

where n_g is the number of spots found by the prefilter for gel g, and the normalized density D^r_{ig} for spot i in gel g is computed by,

$$D^r_{ig} = (\frac{D'_{ig}}{N_g}) * 100\%. \tag{5.4}$$

For an example in computing the `Ratio-sum` normalization using the **cgelp2** `SET RATIO NORMALIZATION <CMD>`, see Section 2.3 Example 21, page 116.

Modified least square method

1. Set the Rspot set feature limits to select robust spots found in any gels in the working set of gels and the Rgel (all gels may be used - even if they are marginal gels or not of immediate interest). Typically the limits are constrained for [Area, OD range, CV area, minimum total integrated density D'. The pairing labels used are: PP+SP]. *map density to Rgel*

2. Do a least squares type search of the PCG DB to compute a regression line (M_g, B_g) mapping D'_g to the density domain of the Rgel. This is done for all pairs of spots in Rspot sets meeting the feature limits criteria (i.e. prefilter concept to be discussed in Section 5.1.10, page 524) and for which the Rgel as well as gel g are present. The density mode is forced to `Absolute` D'. A piecewise linear function is then used to map D'_g to D^L_g as

$$D^L_{ig} = D'_{ig} M_g + B_g \quad \text{if } |D'_{ig}| > |B_g|, \tag{5.5}$$

Otherwise it is modeled by

$$D^L_{ig} = D'_{ig}(M_g/2) \quad \text{if } B_g > 0, \tag{5.6}$$

and

$$D^L_{ig} = D'_{ig}(2M_g) \quad \text{if } B_g < 0. \tag{5.7}$$

This model minimizes the error for the set of spots in the range of D' used to estimate the line. It better forces the intercept to go through zero (thus better estimating low values of D' which have a large error if a piecewise estimate of the curve is not used).

For an example in computing the `least square` normalization using the **cgelp2** `SET LEAST SQUARES CALIBRATION <CMD>`, see Section 2.3 Example 20, page 115.

Absolute density corrected for background (D') method

1. First compute spot integrated density D estimate and then correct for background to D' in the spot segmentation **sg2gii**. *raw density*

2. This mode is only appropriate when the gel loading, labeling or staining and scanner introduce insignificant errors (almost never!). However, it is useful for looking at raw data - especially if you want to process it outside of the existing GELLAB programs.

For an example in using the `absolute` density mode, see Section 2.3 Example 13, page 111.

Density D' corrected for exposure decay factor (EDF) method

exposure decay factor

1. First compute spot integrated density D' as above.

2. Then correct with *exposure factor* F^e which is specified as one of the accession file field entries as `FACTOR=3.12` etc. Corrected D' is then used in all of the density modes including percent and volume. This `FACTOR=`F^e must appear as one of the fields used to compose the gel *study* using the `SET FIELDS` command. If it does not exist for gel g then the F^e_g defaults to 1.0. Then D^{eF}_{ig} is defined as:

$$D^{eF}_{ig} = {D'}_{ig} F^e_g. \tag{5.8}$$

Note: assuming that the exposure factors have been set up using the `SET FIELDS <CMD>`, then do `SET DENSITY MODE` to `Absolute` uses the exposure factor. Setting the density mode to `Uncorrected` does not use it.

Percent of total gel density method

percent total density

1. First compute D'_{ig} estimates in the spot segmentation **sg2gii**. Also compute the total "accepted" spot density $D^{'accepted}_g$ and $D^{'omitted}_g$ in **sg2gii**.

$$D^{'accepted}_g = \sum_{j \in G} D'_{jg}, \tag{5.9}$$

2. Then compute $D^{\%}_{ig}$. If the set of gels have low background noise so that $D^{'omitted}_g / D^{'accepted}_g$ is low then this method might be considered.

$$D^{\%}_{ig} = \frac{D'_{ig}}{D^{'accepted}_g}. \tag{5.10}$$

For an example in using percent of total density, see Section 2.3 Example 13, page 111.

5.1.5 Types of Changes Expected & Corresponding Searches

qualitative & quantitative

There are basically three types of real changes found between two or more 2D gels: (1) *qualitative* (i.e. spots missing in one gel which are present in the other), (2) *quantitative* (i.e. varying amounts of protein in one gel relative to the other when the amount of material in one gel with respect to the other is taken into account for normalization), and (3) *shifts in MW or pIe*. The third type of change can be handled by GELLAB-II for small deviations whereas large changes can not at this time - except for the coordinate-pair test mode (see Section 3.5, page 245)..

Of course one must realize that since 2D gels are a finite precision detection system, qualitative changes may in reality be quantitative changes. So care must

be taken in interpretation - possibly by verification with a darker exposure (in the case of an autoradiograph). Shift changes need to be separated from changes due to the position variance of simply running gels.

F^+ & F^- events

Results obtained by the computer must be further checked. We will be discussing this in the context of: false positive (F^+, spots mispaired), true positive (T^+, spots correctly paired), and false negative (F^-, spot not segmented) events. Proteins suspected of extensive pIe-MW migration need to be detected using other methods - e.g. gel flickering or possibly protein extraction techniques which are in common use today. For an example of gel flickering, see Section 2.2 Examples 15, page 103 and 16, page 103.

using SRL subsets

Qualitative changes can be succinctly defined as follows for the 2-class problem. Let C_1 and C_2 be the two classes. Let r be a Rspot set. Then, a *Search Results List* (SRL) of *missing gel class* spots (using $\sim$ to indicate a logical NOT condition, $\wedge$ to indicate a logical AND condition, and $\vee$ to indicate a logical OR) is defined by

$$SRL = \{r | (\sim (r \in C_1) \wedge (r \in C_2)) \vee ((r \in C_1) \wedge \sim (r \in C_2))\}. \tag{5.11}$$

That is - a spot appears in one experimental class or the other. The test can be made more robust by requiring, that in order for (r in C_i) to be true, the subset of spots in class C_i meet the statistical limits criteria for the subset of gels belonging to class i for Rspot set r. The use of this prefilter will be discussed later in Section 5.1.10, page 524.

Quantitative changes are generally found using standard parametric (T^- or F^-) test or non-parametric (%change-, missing-spot- or Wilcoxon-Mann-Whitney-) test of total integrated density (per spot) distributions for each Rspot set of gels.

Sometimes the researcher has indications of the types and quantity of changes to expect which are based on external biological evidence or experimental conditions. Often, however, this is not the case and so both types of searches should be performed. The `SET SRL SUBSETS` and `SET GEL SUBSETS` commands are used to manipulate SRL subsets and gel subsets respectively. For examples of set operations, see Section 2.3 Examples 48, page 126; 49, page 127; and 50, page 127; for SRL subsets and Examples 18, page 114; and 41, page 124; for gels.

5.1.6 Multiple Gel Analysis Problems

The term multidimensional is used in this paper to refer to a set of data which can be partitioned into two or more classes of gels. For example, experimental condition-1 `X` $\cdots$ `X` experimental condition-n.

partitioning PCG DB

A set of gels from a particular experiment suggests by its biological protocol a *partitioning* into a n-dimensional abstract problem space. An experiment may be

described as an n-class problem. Some typical examples might be:

1. 2-class problem: (control Vs. experiment).
2. n-class problem: (control Vs. exper[1] Vs. $\cdots$ Vs. exper[n-1]).
3. n-class `X` m-condition problem: (control Vs. exper[1] Vs. $\cdots$ Vs. exper[n-1]) `X` m-time samples.
4. n-class `X` m-condition problem: (control Vs. exper[1] Vs. $\cdots$ Vs. exper[n-1]) `X` m-dose samples.
5. n-class `X` m1-condition `X` m2-condition problem: (control Vs. exper[1] Vs. $\cdots$ Vs. exper[n-1]) `X` m1-dose samples X m2-time samples.

For examples of manipulating the experimental gel classes, see Section 2.3 Examples 16, page 112 and 17, page 113. For examples of changing the working set of gels, see Section 2.3 Examples 10, page 110; 11, page 110; 16, page 112; 18, page 114; 19, page 115; and 41, page 124.

For all parametric and some non-parametric (e.g. rank-order) tests one needs at *least* two gel samples/class. This implies replicate gels of the sample specimen or duplicate cultures.

5.1.7 Types of statistical tests and when they are used

There are a number of different statistical tests which can be applied to a set of data to detect differences. In GELLAB-II, we have available parametric (the t-Tests and F-test), the non-parametric tests (Wilcoxon-Mann-Whitney and Kruska-Wallis rank order tests), and a number of constraint-based tests. The selection of a particular tests depends on which biological model you might bring to bear on this data as well as the constraints of the tests themselves. Constraints include: minimum number of samples (i.e. gels) per experimental class, assumptions about normality of the distribution, assumptions about equal or unequal variance of the distribution. For example, the parametric tests required a minimum of 2 samples per experimental class. See Section 5.2.1, page 533 for a first approximation of this type of selection criteria.

standard tests

The other constraint based tests such as the `Missing-class`, `%-search-above-threshold`, `Expression-profile`, `least-squares-fit`, and `coordinate-pair` each depend on a particular model. The biological assumptions of the model should be evaluated to see if the test might apply. For example, the `Missing-class` test might be used when one would expect the change to be qualitative - that is the spot(s) are completely missing from one of two experimental class. The `%-search-above-threshold` simply compares the ratios of spots from

non-standard tests

two experimental classes and is used where there are not enough gels per experimental class to use the parametric or non-parametric tests. The `Expression-profile` search is generally more useful when there are more than two experimental classes and you are trying to find spots which exhibit protein expression profiles similar to those of a known spot in order to suggest spots with similar coordinated expression. The `least-squares-fit` is useful if the set of gels is organized as an increasing function of some dependent variable such as time, dose, concentration, etc. The `coordinate-pair` is a particular constraint based model for precursor-products.

handling F^+ events

When applying any statistical test to a large number of seemingly unrelated distributions from the same domain (such as Rspot sets from the same gels), it can be shown that the false negative and positive rates increase ([MiilR81], [ParR88]). One solution to this problem is to treat the SRL spots found as potential true positives and then evaluate them more carefully with additional tests and with Rmap and mosaic images.

5.1.8 Correlation and Cluster analysis techniques

There are several different ways of clustering data in GELLAB-II: a) correlation tables using the **cgelp2** `TABLE` command, b) correlation plots using the **cgelp2** `DDPLOT` command, c) clustering of the subset of Rspots contained in the SRL using the **cgelp2** `INQUIRE` subcommands (`OEXPRESSION PROFILE TABLE`, `ORDER RSPOTS TABLE`, `CHANGE HISTOGRAM TABLE`, `COORDINATE PAIRS TABLE`) d) dendrogram cluster analysis using the **dendrogram** program on subsets of PCG DB data.

correlation and clustering

Correlation tables can be generated which do gel-gel correlation as a function of Rspot set density data. This is useful to determine the global difference between gels within the same class and between classes. Similarly, Rspots can be correlated with one-another using density data for different Rspots from the same gel (for all gels which have both Rspots). A density-density scatter plot for any two gels is a graphic method for visualizing the gel-gel correlation. Cluster analysis using the **dendrogram** program performs a complete-linkage cluster analysis of features belonging to a set of objects. It then plots the results in a *dendrogram tree* - a plot of a set of objects as a function of a set of feature properities of each object. It can cluster a set of $\{gels\}$ as function of list of $\{Rspot\}$ objects or cluster $\{Rspots\}$ objects as a function of spot expression profiles in a list of gels.

Cluster analysis can used to find sets of gels which are similar in some sense and detection of outlier gels in an experimental class. It can also be used to find sets of marker proteins which can be used to discriminate between two or more classes of gels. We have used it for the separation of marker spots into expression profile-groups which are putatively functionally related [SonP86]. Cluster analysis is discussed in terms of numerical taxonomy ([SneP73], [And84]).

5.1.9 Data reduction using set operations

Data reduction can also be performed using set operations. We present an example summarizing a partial analysis of a 4-class problem (four different types of leukemia) by breaking it down into several 2-class sub-problems. Thus 2D gels of one class of leukemic cells are sequentially compared with a series of different classes of leukemic cells. Manipulation of subsets of spots is used to perform this problem decomposition. Leukemic cells from AML (acute myeloid leukemia), ALL (acute lymphoblastic leukemia), CLL (chronic lymphocytic leukemia) and HCL (hairy cell leukemia) patients were prepared and run as H3 fluorograph 2D gels as described by Lester etal. in [LesE82b] where a full discussion and analysis of this data is detailed. There are three possible combinations of 2-class difference searches where AML is one of the classes. We denote lists of Rspots found in a search i by S_i in the following equations.

set operations

$$S1 = AML\ versus\ ALL, \tag{5.12}$$
$$S2 = AML\ versus\ CLL, \tag{5.13}$$
$$S3 = AML\ versus\ HCL. \tag{5.14}$$

A possible biological question might be *which polypeptides are possibly correlated with differences between AML and all of the lymphoid leukemias (i.e. ALL, CLL and HCL)?* Differences correlating with the AML class are found in the S1, S2 and S3 searches. Therefore a *potential* list of Rspots which meet the initial constraints of the question might be computed as S4 using the following set operations in the next Equation where INTERSECTION ($\bigcap$) is the standard set theoretic operator [KorR66]. If the missing class-test is used, then Rspots in S4 will be those either missing from AML while present in ALL, CLL and HCL or present in AML while missing from the other three classes.

sets of sets

$$S4 = \bigcap(S1, S2, S3). \tag{5.15}$$

The results of any search is a search results list of which S1, S2, and S3 are examples. GELLAB-II **cgelp2** has SRL subset operations in the `SET SRL SUBSETS` command for saving and manipulating (using set theoretic operators) up to 88 SRL subsets. Sets may be written to or read from disk files so 88 sets is not really a limitation. The user would typically: (1) do the three searches required to generate the $\{S_i\}$ saving the SRL in subsets called S1, S2, S3, (2) use the SRL subset `UNION` ($\bigcup$), `INTERSECTION` ($\bigcap$) and `DIFFERENCE` ($-$) operations to compute the final additional SRL subsets. The `SEQUENTIAL SET OPR` can same time by doing the intersection of a sequence of sets. In addition, any or all of the SRL subsets may be written into files and selectively restored at a later date. The set of Rspots in S4

sequential set operations

is a *potential* list of significant spots which however *must be verified* with manual observation of Rmap and mosaic images as well as further wet-lab experiments ([EckC88], [StoE89]).

For examples of using set operations to perform these complex search operations, see Section 2.3 Examples 18, page 114 and 50, page 127.

Alternatively, the *list of SRL subsets* can be used with the `SET SRL SUBSETS` command to achieve the same effect. A list of SRL (LOS) subsets is gathered from various *existing* SRL subsets using the `FINDKEYWORDS`, `QUERYRSPOT`, or `EXPLICIT/LISTOFSRLS` subcommands. The LOS can then be used to supply arguments to other `SET SRL SUBSETS` commands including `UNION`, `INTERSECTION`, `SUBTRACT` as well as `LIST/DIR/LISTOFSRLS`. For example, let the LOS be a list of n SRL subsets $(p, q, \cdots, t)$ such that $f(1) = p$, $f(2) = q$, $\cdots$ $f(n) = z$. Then, UNION, INTERSECTION and SUBTRACT would be:

$$\bigcup_{i=1}^{n} SRL(f(i)),$$

$$\bigcap_{i=1}^{n} SRL(f(i)),$$

and

$$SRL(f(1)) - SRL(f(2)) - SRL(f(3)) \cdots - SRL(f(n)).$$

respectively.

5.1.10 Selection of Search Constraints - Prefilter & Subsequent Test

the prefilter

The search procedure may be thought of as a two stage process as illustrated in Figure 5.3. First, gel density must be normalized between gels. This calibration is performed using one of the methods listed in Section 5.1.4. The first stage is called the *prefilter*, detailed in Figure 5.4, and attempts to select those Rspot sets which would give a robust response to the second part of the procedure. The second part is the parametric or non-parametric test.

For examples of changing the working set of gels and its effect on a search, see Section 2.3 Example 19, page 115. For examples of changing gel classes, see Section 2.3 Example 16, page 112. For examples of changing the (pIe,MW) region of the gel to search, see Section 2.3 Example 28, page 119. For examples of adjusting the Rspot features statistical limits, see Section 2.3 Examples 27, page 119; 30, page 120; 31, page 121; 51, page 128; and 52, page 128. For examples of adjusting the spot pairing labels, see Section 2.3 Examples 28, page 119; 29, page 120;

and 39, page 123. For examples of restricting search by prefiltering Rspots which are members of a SRL subset, see Section 2.3 Examples 25, page 118; and 26, page 118.

F^+/F^- and prefilter

The use of the prefilter stage is critical in reducing the number of false positive estimates of significant Rspot set differences found when applying any test. Of course, any decrease in the false positive rate will result in an increase in the false negative rate. (A false positive is an event which is called significant by a test which, on further analysis, is determined not to be a real event. A false negative is an event which is real but was missed by the test. The rate is computed as the ratio of the number of specific events to the total number of all events.)

The prefilter consists of a conjunction of tests - each of which must be satisfied for it to consider a Rspot set as "passed". If any element of the prefilter fails, then the prefilter aborts checking the rest of the prefilter tests and considers that particular Rspot set as failed - then going on to test the next one. Using the `/EXPLAIN` switch in **cgelp2** with the `INQUIRE` search command, it is possible to see which parts of the prefilter failed (see Section 3.3.4, page 164). For an example, see Section 2.3 Example 22, page 117

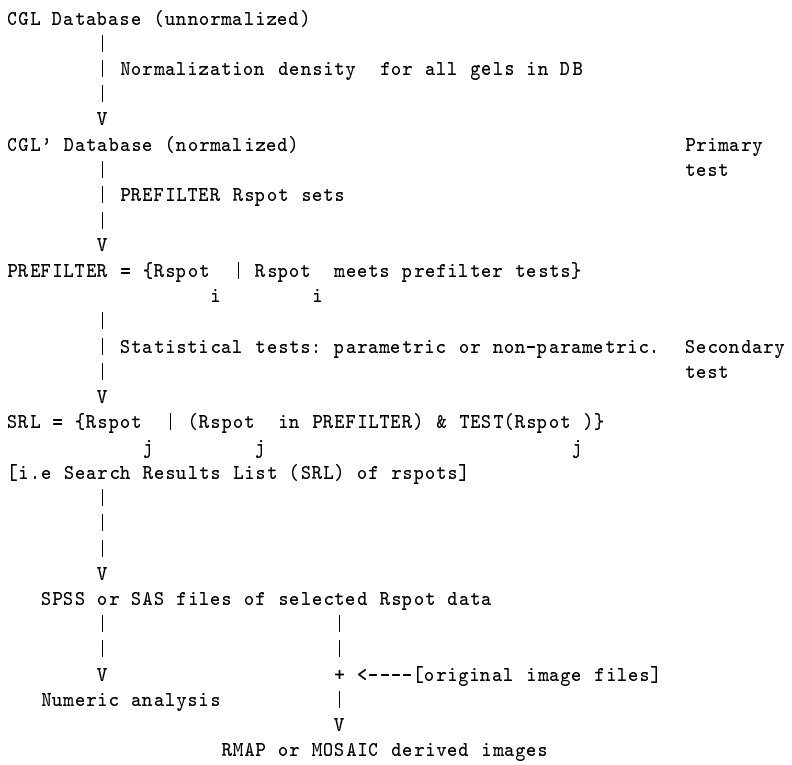


Figure 5.3. Performing a search of the PCG DB. The PCG DB is searched as a two stage process. Each normalized Rspot set is first tested with a prefilter to determine whether it is a candidate for further testing. If it is, then the statistical test selected is applied. Spots passing this second test are saved in the search results list, SRL, which is then used to make Rmap and or mosaic images or plots as well as for other SRL operations.

```
{CGL DB}
  |
  | Partition WORKING SET of GELS from set of all gels in DB.
  |
  V
{CGL DB | WS}
  |
  | Partition gels into up to 9 EXPERIMENTAL CLASSES.
  |
  V
{CGL DB | WS & Classes}
  |
  | Partition individual Rspot sets by (pIe,MW) sub-REGION.
  |
  V
{CGL DB | WS & Classes & Region}
  |
  | Partition individual spots by Spot-Pairing LABEL.
  |
  V
{CGL DB | WS & Classes & Region & Label}
  |
  | Partition DB by Rspot set or Rspot set/class meeting
  | statistical spot FEATURE LIMITS of Rspot set features.
  |
  V
{CGL DB | WS & Classes & Region & Label & Feature-Limits}
  |
  | (Optional) Restrict Rspot membership in SRL[n] SUBSET of Rspots.
  |
  V
{CGL DB | WS & Classes & Region & Label & Feature-Limits & SRL[n]}
```

Figure 5.4. Prefilter: partitioning the PCG DB. The prefilter test consists of five (or six) Rspot set specific tests which determine whether or not an Rspot set is to be visible for further processing. Any of the parameters of the five tests may be changed by the user. The Rspot set statistical limits are applied on such Rspot set features as: relative distance from the landmark, mean DL, mean DP, mean area, mean density, CV Rset area, CV Rset density, OD difference within spot, significance level, # gels in Rspot set (see `LIMITS` command on page 250). For example, restricting the CV area, OD difference, and # gels/Rspot set limits would be useful in finding robust stable spots possibly suitable for normalization. Note that prior to performing a search, any or all of the above parameters may be adjusted by the experimenter.

5.1.11 Adjustment of Search Parameters in the prefilter

adjusting the prefilter

By adjusting search parameters, the false negative rate of a **cgelp2** search could be set closer to zero in order to pick up all real results. However, the side effect of doing this is to greatly increase the false positive rate up to near 100%. Thus every spot would be called significant! As in all practical statistically-based systems, a

careful balance of the false positive and false negative rates must be found for each application. It is this issue of how one tunes the parameters for the search process which we will be addressing.

checking visually

In the initial gel image data reduction to spot lists, the true positive (spot properly segmented) and false negative (spot not segmented) rates can be computed as follows. Generate an image which is the original image with the segmented image subtracted from it (cf. Section 3.18). An example of this type of image can be seen in [LipL80a] and can be generated by **sg2gii**. For an example, see Example 6, Step 9%, page 97. Spots which were segmented appear as if they were cut out of the gel and are white. Spots which are were not segmented are still in the image. A photograph is then taken and used to manually record spots that were not segmented by placing a mark through remaining spots. Counting marks yields the number of false negative events.

In the spot list pairing stage of the analysis, true positive (good spot pairing) and false positive (incorrect pairing) rates can be computed for a given set of spots paired by GELLAB-II. The **Xpix** `compare images` flicker mode[3] invoked with **accppx** can be used to manually check each spot pairing on derived images marked with the same sure pair, SP or possible pair, PP, pairing label. For an example of visualizing paired labels, see Section 2.2 Example 8, page 99.

replicate gels

If replicate gels are used in the PCG DB with the constraint that prefiltered spots must occur in the replicate gels, then the false positive rate in the PCG DB can be greatly reduced. This is because it is very unlikely for a false positive spot to occur in all of the replicate gels.

The end product of a PCG DB search is a search results list and is a list of the Rspot set numbers which passed both the prefilter and the particular statistical test. Figure 5.5 shows the various options available in GELLAB for manipulating the SRL or SRL subset in continuing an analysis.

[3] Also available with **disp11**.

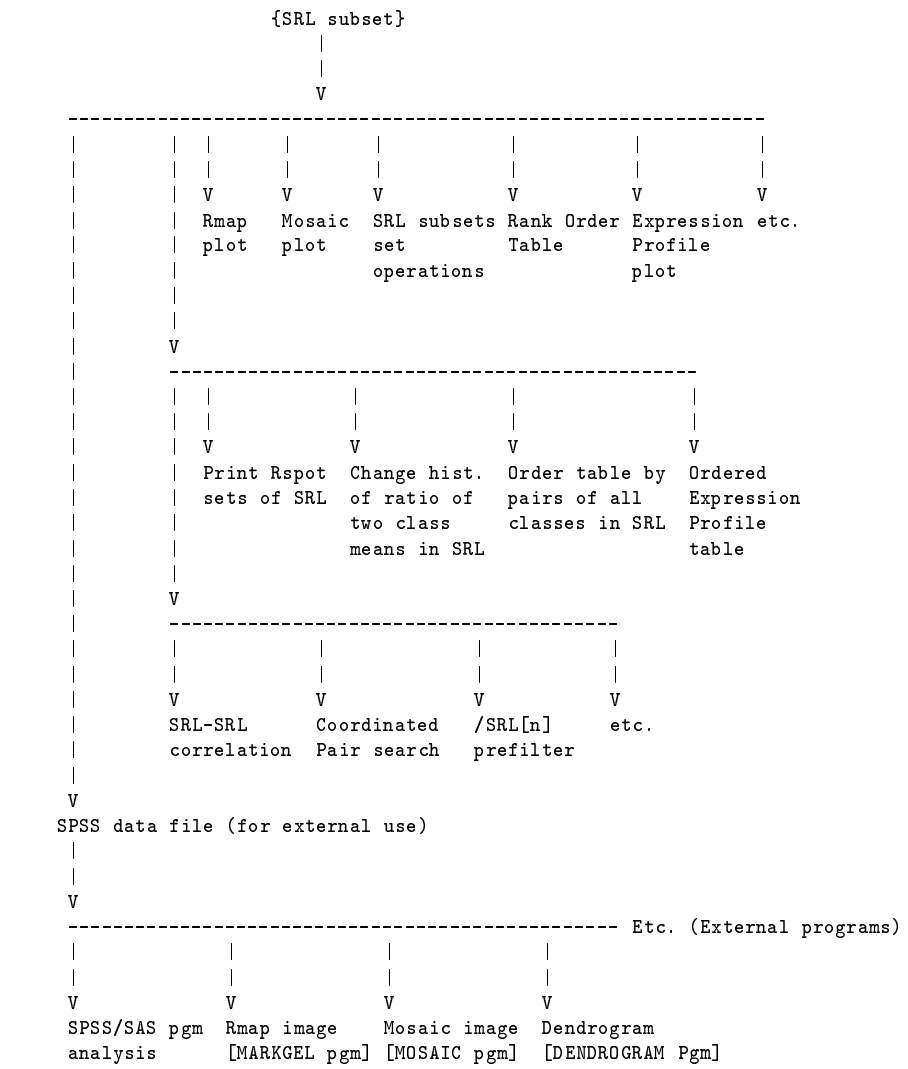


Figure 5.5. Different uses of the Search Results Lists to check the analysis. A number of independent additional **cgelp2** processing options are available to check and manipulate these search results lists. Used judiciously, these can aid the refinement or termination of the search process.

5.1.12 Evaluation of Search Results using Rmaps and Mosaics

A typical 2D gel PCG DB analysis protocol is shown in Figure 5.6. For the case where a good guess has been made on the initial prefilter parameters and only two classes of gels exist, a single search through the PCG DB might be adequate. Each SRL search to be viewed in an Rmap or mosaic image is saved in an SPSS data file (suitable for further analysis using the SPSS statistical package [NieH75]). When the user is finished making a number of SPSS files he would leave the **cgelp2** program and proceed to make Rmap and mosaic images using the two GELLAB-II programs **markgel** and **mosaic**. For examples of computing and displaying Rmaps, see Section 2.2 Example 13, page 101 and Section 2.3 Examples 6, page 108; and Example 51, page 128. For examples of computing and displaying mosaics, see Section 2.2 Example 14, page 102 and Section 2.3 Example 6, page 108; and Section 2.3 Example 51, page 128. Particular Rspots can then be manually evaluated as true or false positive events. If more information is desired, **cgelp2** is reentered and the process repeated until those outstanding questions (which *can* be answered by such a 2D gel DB system) are answered.

using Rmaps & mosaics

It is more difficult to realize those questions which *cannot* be answered by probing a given PCG DB. If spot changes are subtle, the normalization marginal or noisy, inadequate numbers of replicate gels used, the statistical results not particularly robust, etc. may all suggest that other experiments should be performed to establish the tentative results. This may include adding more replicate gels or changing the experimental conditions under which the gels are run.

The creation of mosaic images is a computationally expensive procedure as well as possibly requiring a large amount of temporary disk storage. In the case where the SRL contains just a few spots, mosaic images of all of them can be generated and evaluated manually. In the case where there are many spots (greater than say 10 or 20), a brief manual pre-analysis of the Rmap(s) of these spots can be used to eliminate some of the false positive spots *in advance of making mosaics* thus reducing the number of mosaic images to be actually computed or reviewed. For examples of this review process, see Section 2.3 Examples 6, page 108; and 51, page 128; Mosaic images which start with `w` can be reviewed using `accppx` (e.g. `accppx w00273.ppx`. Most of the false positive spot differences can be quickly resolved by flicker (using **accppx**, e.g. `accppx 524.1 535.1 -P1:m -P2:m`) comparing Rmaps (derived images may be flickered as well as original images) from each of the two classes used in the search. Figure **??** illustrates this iterative parameter adjustment process. For examples of this iterative adjustment of p-value and CVD, see Section 2.3 Examples 51, page 128; and 52, page 128.

false positive evaluation

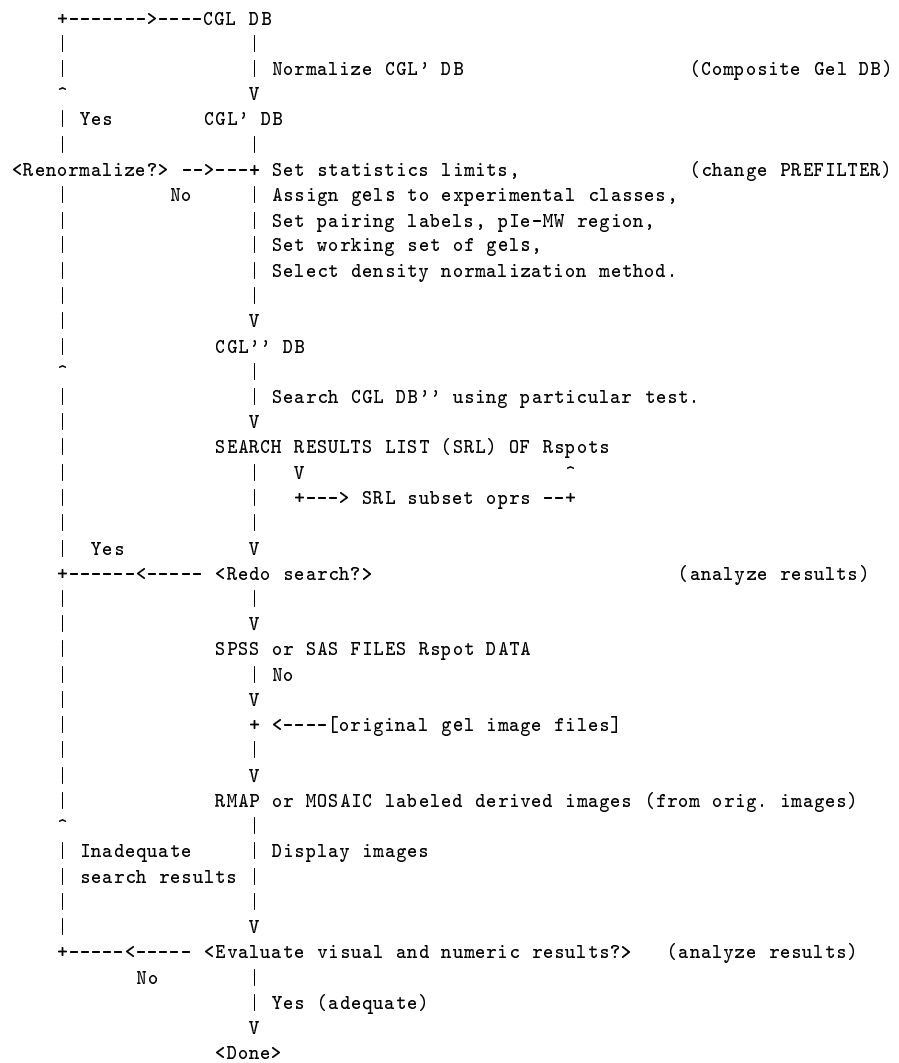


Figure 5.6. Changing search parameters. This shows the order in which one would modify parameters for search based on SRL analysis in iterating an analysis. A typical 2D gel PCG DB analysis protocol. Modification of prefilter parameters or tests may be indicated for iterative search based on an analysis of the SRL found in the current search. Set operators on SRL subsets found in a series of tests may also be necessary to detect multi-dimensional class differences. Iterative adjustment of prefilter search parameters modifies the TRUE/FALSE positive rate of spot differences detected. Decreasing the false negative rate (to find *more* real spot differences) will result in increasing the false positive rate (finding *more* incorrect spot differences) and vice-

versa.

5.1.13 Other Types of Searches and PCG DB views

We have presented techniques for increasing the effectiveness of the 2D gel analytic process for finding polypeptide spot differences. The use of such tools is properly a reflection of the hypotheses brought to bear on the entire biological experimental design and execution, and 2D gel preparation. By carefully taking these factors into account, the analytical process can be enhanced in effectiveness and shortened in time by taking advantage of the powerful logical and statistical techniques described in this paper which can lead to zeroing in on results if properly applied.

Obviously checking the T^+, F^+, F^- rates for one pair of gels will not give us the actual rates for any arbitrary set of gels. However, it does give an indication of the *types of errors* the GELLAB system may make thus aiding in interpreting results. These include: edge region or streak mis-pairing, mis-pairing due to inadequate number of landmarks in a highly variable region, hazy and very light spots are not always detected or are sometimes fragmented. Most of these errors occur with marginally detected spots or regions. As a result, robust spot changes found by GELLAB are given more credence.

session logs

It is necessary to keep track of what occurred during an interactive user session with GELLAB in order to verify parameters specified and procedures performed in obtaining search results. A UNIX program called *script(1)* allows all video terminal "traffic" to be saved in a file which may be printed at the end of the session. We have found these session files to be invaluable for both teaching the GELLAB-II system and for keeping track of complicated sequences of operations performed by the operator during an analysis without slowing down the user by requiring them to use a hard copy (i.e. printing) terminal.

command scripts

Particularly useful is the grouping of short often-used sequences of operations in so called *command files* which can be instantiated as a high level *command* with particular arguments at execution time. These UNIX script command files may be run as either background (batch–like) processes or interactively. For example, two frequently used UNIX scripts or aliases are the high level commands are `mark, ppxmosaic` (see Section B, page 585). The former computes a Rmap image given the gel accession number and the SPSS Rspot file name of a SRL subset. The latter constructs a mosaic image(s) given the specified Rspot number and the SPSS SRL-subset list file which contains it. The command file `mark (ppxmosaic)` passes the user specified arguments plus default program switches to the **markgel (mosaic)** program(s) mentioned previously. Using the '`SET SRL SUBSET//SPSS/MOSAIC`', **cgelp2** command invokes generation of several files (i.e. SPSS data files and batch scripts to create mosaic images from these SPSS data files). For an example, see Section 2.3 Example 6, page 108. These are SPSS file(s)

and UNIX script command file(s) to generate mosaic images contained in a specified SRL subset(s). Using these high level command forms permits one to concentrate on the problem and not get bogged down in details.

A practical problem, difficult to decide is the choice between 1) assembling a very large database, 2) constructing several smaller ones. A 100 gel PCG DB not only takes a very large amount of disk space, but more important, with all gels in the working set, it takes a long time to search. One alternative is to reduce the working set of gels in such a large DB to a still statistically-significant smaller-number of gels for probing purposes. After finding interesting spots using the smaller working set of gels, the working set is expanding back to the full set of gels for generating Rmaps and mosaics for backchecking search results by manually viewing and checking spots GELLAB-II had previously indicated to be significant. Another option is to construct a Cgel′.

large gel DBs

5.2 Rules to help run GELLAB-II

We have been experimenting using expert-system rules to aid in performing different aspects of the gel analysis. ([HayF83] is a good introductory text on expert-systems in general). One application is as an aid in running software such as the `cgelp2` exploratory data analysis part of GELLAB. We are investigating rule sets in statistical test selection, parameter selection by analyzing *failCode* histogram data (cf. Section 3.3.4 page 164) as well as normalization method selection, detection of correlated spots, constraint analysis, gel classification, assistance in batch script generation, and the use of auxiliary databases in the context of 2D gels. Some of the early rule sets being developed are listed here and can be used manually to help decide on courses of action in your exploratory data analysis.

help is on the way

Tukey discusses some of the uses of expert-systems for exploratory data analysis [TukJ86]. In particular, expert-systems can be especially useful for helping occasional or beginning users. Because of the complexity of any 2D gel analysis system, a system which worries about forgotten or unknown details (and prompts the user where appropriate) will be much more usable by users who can't and don't want to remember all of the idiosyncrasies of performing an analysis.

5.2.1 Rules for selecting cgelp2 test to search PCG DB

It is possible to suggest which statistical test might be used in a PCG DB search using an expert-system rule-based algorithm. The set of rules can be evaluated manually given a question phrased as set of facts. Check each rule to determine if all preconditions (antecedents in the "`if`" part of the rule) are true and if so to "`then assert`" the corresponding (consequent) facts. Do this repeatedly, starting

at rule R1 each time, until one of the hypotheses is true. The rules are given below in Table 5.2.

The notation expresses facts or hypotheses as UPPER-CASE-WORDS, lower case words and '@' prefixed words are rule connectives and data evaluation operators respectively. For example '@val CLASSES' will return the current number of experimental classes currently visible in the PCG database and which we are using to compare gels. Then, the predicate '@val GELS/CLASS @> 1' is true if it evaluates to true - i.e. there is more than one gel in a class. The objective of using the expert-system with the above rule set is to select one of the hypotheses which makes sense given the current state of the GELLAB-II database. Information which is not found in the database will be prompted for from the user in the future implementation using an expert-system. The rules can be applied manually in the interim. We illustrate this will several scenarios.

TABLE 5.2. Rules to select a statistical test. Example of expert-system rule set for helping select statistical test.

```
TITLE: STATISTICAL-TEST-TO-USE.RULES,

@rules: STATISTICAL-TEST-TO-USE-RULES,
R1:  if CHANGES-EXPECTED
         then assert: DO-TESTS
R2:  if @not CHANGES-EXPECTED
         then assert: IS-DON'T-DO-TESTS
R3:  if DO-TESTS and @not QUANTITATIVE
         then assert: QUALITATIVE
R4:  if DO-TESTS and QUALITATIVE
         then assert: IS-MISSING-CLASS-TEST
R5:  if DO-TESTS and QUANTITATIVE and EXPECT-GAUSSIAN-DISTRIBUTIONS
         then assert: PARAMETRIC-TESTS
R6:  if DO-TESTS and QUANTITATIVE and @not EXPECT-GAUSSIAN-DISTRIBUTIONS
         then assert: NON-PARAMETRIC-TESTS
R7:  if PARAMETRIC-TESTS and @val NBR-CLASSES @== 2 and @val GELS/CLASS @> 1
         then assert: T-TESTS
R8:  if PARAMETRIC-TESTS and @val NBR-CLASSES @>= 2 and @val GELS/CLASS @> 1
         then assert: IS-F-TEST
R9:  if T-TESTS and EQUAL-VARIANCE
         then assert: IS-STANDARD-T-TEST, IS-T-TEST
R10:  if T-TESTS and @not EQUAL-VARIANCE
         then assert: IS-BEHRENS-FISHER-T-TEST, IS-T-TEST
R11:  if T-TESTS and @don't-know-EQUAL-VARIANCE
         then assert: IS-TB-TEST, IS-T-TEST
R12: if NON-PARAMETRIC-TESTS and 2-CLASSES and @val GELS/CLASS @== 1
         then assert: IS-%CHANGE-TEST
R13: if QUANTITATIVE and NON-PARAMETRIC-TESTS and @val CLASSES @== 2 and
                 @val GELS/CLASS @>= 3 and @val P-VALUE @>= 0.80
         then assert: IS-WILCOXON-MANN-WHITNEY-TEST
R14: if QUANTITATIVE and NON-PARAMETRIC-TESTS and @val CLASSES @>= 2
                 @val GELS/CLASS @>= 5
         then assert: IS-KRUSKA-WALLIS-TEST
```

```
@hypotheses:
IS-F-TEST
IS-T-TEST
IS-STANDARD-T-TEST
IS-BEHRENS-FISHER-T-TEST
IS-TB-TEST
IS-WILCOXON-MANN-WHITNEY-TEST
IS-KRUSKA-WALLIS-TEST
IS-%CHANGE-TEST
IS-MISSING-CLASS-TEST
IS-DON'T-DO-TESTS
@end
```

The following illustrates an example of manually evaluating these rules. Given the following facts for 12 gels, attempt a quantitative test with unequal variance in the Rspot distributions which may be Gaussian. Run through the rules R1 through R12 until a rule applies. Then apply it. Then start at R1 again and repeat this procedure until one of the assertions is in the hypothesis list.

```
STEP  RULE  ASSERTED-FACTS
----  ----  --------------
 1     R1:  DO-TESTS
 2     R5:  PARAMETRIC-TESTS
 3     R7:  T-TESTS
 4     R10:  IS-BEHRENS-FISHER; IS-T-TEST
 5  Hypothesis-matched DONE.
```

In the case where you don't know if the variance is equal,

```
STEP  RULE  ASSERTED-FACTS
----  ----  --------------
 1     R1:  DO-TESTS
 2     R5:  PARAMETRIC-TESTS
 3     R7:  T-TESTS
 4     R11:  IS-TB; IS-T-TEST
 5  Hypothesis-matched DONE.
```

In the case where you want a quantitative non-parametric 2-class test,

```
STEP  RULE  ASSERTED-FACTS
----  ----  --------------
 1     R1:  DO-TESTS
 2     R6:  NON-PARAMETRIC-TESTS
 3     R13: IS-WILCOXON-MANN-WHITNEY
 4  Hypothesis-matched DONE.
```

5.2.2 Rules for adjusting cgelp2 prefilter parameters

parameter selection

The key to successful use of **cgelp2** is in quickly adjusting the prefilter and then zeroing in on potential interesting spot differences. A problem with this type of data exploration is that there are a large number of different prefilter parameters - some more important thatn others. The following simple rules attempt to illustrate one orientation for prefilter parameter selection of the *more critical* parameters. See the discussion on the failCode histogram which can be used to help drive this rule selection.

TABLE 5.3. Example of simple setup rules for statistical parameters in prefilter.

```
TITLE: STATISTICAL PARAMETERS-SETUP.RULES(4-29-88)

@rules: STATISTICAL PARAMETERS-SETUP.RULES
R1: if @val EXPECTED-nbr-CHANGES @< @val SMALL-nbr and PARAMETRIC-CONSTRAINTS and
       @val nbr-CHANGES-FOUND @> @val LARGE-nbr
        then assert: TIGHTEN-PREFILTER, DECREASE-P-VALUE,
                     @del IS NEW-PARAMETER-SETTING

R2: if @val EXPECTED-nbr-CHANGES @> @val LARGE-nbr and PARAMETRIC-CONSTRAINTS and
       @val nbr-CHANGES-FOUND @< @val SMALL-nbr
        then assert: LOOSEN-PREFILTER, INCREASE-P-VALUE,
                     @del IS NEW-PARAMETER-SETTING

R3: if IS TIGHTEN-PREFILTER and DECREASE-P-VALUE and @val P-VALUE @== 0.90 And
       @not IS NEW-PARAMETER-SETTING
        then assert: @set P-VALUE @= .90, IS NEW-PARAMETER-SETTING

R4: if IS TIGHTEN-PREFILTER and DECREASE-P-VALUE and @val P-VALUE @== 0.95 And
       @not IS NEW-PARAMETER-SETTING
        then assert: @set P-VALUE @= .95, IS NEW-PARAMETER-SETTING

R5: if IS TIGHTEN-PREFILTER and DECREASE-P-VALUE and @val P-VALUE @== 0.99 And
       @not IS NEW-PARAMETER-SETTING
        then assert: @set P-VALUE @= .99, IS NEW-PARAMETER-SETTING

R6: if IS LOOSEN-PREFILTER and INCREASE-P-VALUE and @val P-VALUE @== 0.99 and
       @not IS NEW-PARAMETER-SETTING
        then assert: @set P-VALUE @= .95, IS NEW-PARAMETER-SETTING

R7: if IS LOOSEN-PREFILTER and INCREASE-P-VALUE and @val P-VALUE @== 0.95 and
       @not IS NEW-PARAMETER-SETTING
        then assert: @set P-VALUE @= .90, IS NEW-PARAMETER-SETTING

R8: if IS LOOSEN-PREFILTER and INCREASE-P-VALUE and @val P-VALUE @== 0.90 and
       @not IS NEW-PARAMETER-SETTING
        then assert: @set P-VALUE @= .80, IS NEW-PARAMETER-SETTING

R9: if TIGHTEN-PREFILTER
        then assert: @set COV-DENSITY @-= .50,
```

```
                @set COV-AREA @-= .50,
                @set MIN-D' @+= 2.0,
                @set MIN-nbr-GELS-PER-CLASS @+= 1,
                IS TIGHTEN-PREFILTER

R10: if LOOSEN-PREFILTER
        then assert: @set COV-DENSITY @+= .50,
                @set COV-AREA @+= .50,
                @set MIN-D' @-= 2.0,
                @set MIN-nbr-GELS-PER-CLASS @-= 1,
                IS LOOSEN-PREFILTER

R11: if PARAMETRIC-CONSTRAINTS and IS T-TEST
        then assert: @set MIN-nbr-GELS-PER-CLASS @= 2
                @set MAX-OD-PER-PIXEL @= 1.80

R12: if PARAMETRIC-CONSTRAINTS and IS F-TEST
        then assert: @set MIN-nbr-GELS-PER-CLASS @= 2,
                @set MAX-OD-PER-PIXEL @= 1.80

R13: if NON-PARAMETRIC-CONSTRAINTS and IS %CHANGE-TEST
        then assert: @set MIN-nbr-GELS-PER-CLASS @= 1,
                @set MAX-OD-PER-PIXEL @= 2.7

R14: if NON-PARAMETRIC-CONSTRAINTS and IS MISSING CLASS-TEST
        then assert: @set MIN-nbr-GELS-PER-CLASS @= 2,
                @set MAX-OD-PER-PIXEL @= 2.7

R15: if NON-PARAMETRIC-CONSTRAINTS and IS WMW-TEST
        then assert: @set MIN-nbr-GELS-PER-CLASS @= 2,
                @set MAX-OD-PER-PIXEL @= 2.7

R16: if NON-PARAMETRIC-CONSTRAINTS and IS KRUSKA WALLIS-TEST
        then assert: @set MIN-nbr-GELS-PER-CLASS @= 2,
                @set MAX-OD-PER-PIXEL @= 2.7

@hypotheses:
IS NEW-PARAMETER-SETTING
@end
```

5.2.3 Rules for adjusting cgelp2 normalization method

As pointed out in Section 5.1.4, page 516, the normalization of density data between gels is essential prior to comparing data between gels. Given the variety of normalization methods, it is useful to be able to suggest which one to try under different experimental and exploratory data analysis conditions. The following rules show one way of deciding which normalization method to select.

normalization selection

TABLE 5.4. Example of simple setup rules for PCG DB normalization.

```
TITLE: NORMALIZATION-TO-USE.RULES,

@rules: NORMALIZATION-TO-USE.RULES,
R1:  if @val nbr-well-formed-spots-present-in-all-gels @>
        0.25 @* @val total-nbr-spots-per-gel
         then assert: IS-RATIO-LIST-NORMALIZATION, IS-METHOD

R2:  if @val Domitted/Daccepted-per-gel @< 0.25
         then assert: IS-%-NORMALIZATION, IS-METHOD

R3:  if @val nbr-well-formed-spots-present-in-all-gels @<=
        0.25 @* @val total-nbr-spots-per-gel
         then assert: IS-LEAST-SQUARES-NORMALIZATION, IS-METHOD

R4:  if @val nbr-spots-per-gel < @val small-nbr-spots and
        @val variance(nbr-spots-per-gel) @> @val high-nbr and
        @val Domitted/Daccepted-per-gel @>= 0.25
         then assert: IS-EXPOSURE-DENSITY-FACTOR-NORMALIZATION,
                              IS-METHOD

R5:  @not IS-METHOD
         then assert: IS-D'-NORMALIZATION, IS-METHOD

@hypotheses:
IS-RATIO-LIST-NORMALIZATION
IS-LEAST-SQUARES-NORMALIZATION
IS-%-NORMALIZATION
IS-EXPOSURE-DENSITY-FACTOR-NORMALIZATION
IS-D'-NORMALIZATION
@end
```

Chapter 6

Export version of GELLAB-II

This Chapter discusses the set of files which constitutes the GELLAB-II system currently exported (see MTA in Appendix ?? page ??. A list of the files and what they are used for is also given in this section. Files for the programs listed without †'s in Table 1.1, page 49 are included in the distribution.

Section 6.1 discusses what hardware, software, and personnel are required to setup, run and maintain GELLAB-II in other laboratories. Section 6.2 discusses the contents of the export *tar(1)* formated magtape, Section 6.3 page 544 discuss how to read the tape, Section 6.4 page 545 how to install GELLAB-II files into the computer, and Section 6.6 page 547 tests the newly installed system. Section 6.7 page 549 shows how to add new users to GELLAB-II.

6.1 Resource requirements for running GELLAB-II in other laboratories

The following minimum requirements for computer hardware, software, and personnel and software (vendor) support are discussed in order to indicate what is required to install and maintain an ongoing GELLAB-II system. GELLAB-II has been ported to several machines including the SUN3 and microVAX-II workstations, and the CONVEX super-computer.

Color display GELLAB-II will work without a high resolution color display. However, you want (i.e. need) a color display to see what you are doing. An 8-bit color display to view a grayscale or pseudocolor gel image. GELLAB-II will work without using line drawing plots of gel outlines, but currently all effort

is going into the color display version. The SUN SparcStation series are (today) the most cost effective color SUN computers which can handle this. We suggest running GELLAB-II with a minimum of 16 Mbytes for reasonable response when running interactive graphics - 32 Mbytes is ideal.

Disk storage To be comfortable in running GELLAB-II, you should have at *least* 200 Mbytes (preferably 600 Mbytes or more) of available *user* disk space. For exampl,e high resolution scanners such as the Molecular Dynamics laser scanners generate scan images files of about 10 Mbytes each! So 20 of these gel images would take about 200 Mbytes - just for the raw data!

This free disk space *over* and above any space needed for the system and other users. If higher resolution images (larger than 512x512 pixels) are used, even more disk space is required. See Section 1.6, page 60 for more information on disk space requirements based on numbers of gels. GELLAB-II only requires the full resolution images during image segmentation. Later, if the `-std512ppx` option is used with 1Kx1K or larger scans, then storage needs can be reduced during processing if the high resolution images are removed from the disk after the initial gel segmentation step. Then GELLAB goes to the standard '`s`' images when making Rmaps or mosaic derived images since they are used visually – not quantitatively.

Camera/Scanner A fairly good camera/scanner is required. Our Datacopy 612F CCD camera is connected to a SUN3/50 SCSI (Small Computer System Interface) bus via the Datacopy GPII interface with a UNIX driver, but is no longer sold. Since GELLAB-II can convert other types of scanned image data, other existing hardware such as Molecular Dynamics laser densitometer and BioImage scan data can be used. In general, it is difficult to connect a TV camera or scanner to a computer without special hardware and software. The SUN VideoPix(tm) board can capture NTSC or PAL TV camera images in a 640x480 format for NTCS and 525x512 for PAL. However, TV cameras have limited resolution as well as lens flair and image shading problems.

If another camera/scanner is used, then some software development would be required to interface it to the general purpose scanner library procedures. The **ppxcvt** program is available to convert a number of different input image formats such as ASCII hex-coded image file, Molecular Dynamics, BioImage, Elsie, general TIFF scanned image files to standard GELLAB-II PPX formated image files.

Software and support personnel GELLAB-II (and other similar 2D gel analysis systems) are complex enough to install and to maintain that some computer specialist aid would be almost essential. Once GELLAB-II is installed and one

understands its operation, it is not that difficult to use and so less support would be required. There is also some routine maintenance that should be performed with respect to disk file backups as well as installation of new UNIX releases from the vendor. Hardware needs to be maintained and fixed by the computer and camera vendors as required.

6.2 GELLAB-II export tar tape contents

The rest of the chapter discusses what is on the export distribution tape, how to read the tape and how to install it on your system so that users of the `gelusr` user group (i.e. members of the `gelusr` group specified in the UNIX system file `/etc/group`) can access it. Note: some familiarity with UNIX is required to do this installation. **DON'T DO IT OTHERWISE**.

READ CAREFULLY

Note that we will be referring throughout this Section to the user account `gelmgr`. If you do not want to put GELLAB there, then substitute the home directory name of the account where it will live and be managed. Other GELLAB-II users will access this directory from their accounts since all GELLAB-II accounts are members of UNIX group `gelusr`. In addition, the `gelmgr` account also is a member of UNIX group `gellab`.

GELLAB manager account

There is currently one SUN tar file (device `/dev/rst8` or `/dev/rst0`) SUN3 DC600A (60 Mb), SUN4 DC6150 (150 Mb) cartridge tape or EXOBYTE 8mm type cassette tape in the GELLAB-II software distribution. The tape is meant to be restored into the `gelmgr` login account home directory with primary UNIX user group 'gellab and secondary user group `gelusr`.

tape format

The main GELLAB-II distribution directories are defined as follows:

```
~/
DIRECTORY         CONTENTS
---------         --------
home-dir.files    files to be copied to your home directory
gellab            main GELLAB-II distribution
mkGelMgrAcct.do   script to setup the GELLAB manager's account.


Going into these directories,

home-dir.files    files to be copied to your home directory (i.e. cd ~/)
DIRECTORY         CONTENTS
---------         --------
.cshrc.gellab     GELLAB specific .cshrc file, copy to .cshrc and edit
.emacs.gellab     GNU EMACS (GEMACS) startup file defaults
.login.gellab     GELLAB specific .login file, copy to .cshrc and edit
.twmrc.gellab     X11 twm window manager defaults
.xinitrc.gellab   X11 MIT window manager startup for xinit
gel.rc            GELLAB-II state file for user's HOME pointing to ~gelmgr/gellab/demo
```

```
gel.rc.DEMO       GELLAB-II state file for ~gelmgr/gellab/demo

gellab            main GELLAB-II distribution
DIRECTORY         CONTENTS
---------         --------
aux               auxiliary runtime directory of *.gsf *.gcf *.ppx etc
ann               annotation database directory
bin           +++ executable files
demo              demonstration script files and logs of these scripts
doc           +++ GELLAB-II general documentation
gen               CGELP2 generated derived files from analysis
id                accession database files
lms               landmark set database files
pcg               Paged Composite Gel database files
ppx               original gel scanned image *.ppx files
src           +++ directory of source code for all GELLAB-II programs
tmp               temporary *.ppx image directory

+++ NOTE: When making accounts and directories for other GELLAB users
on the same file system, these directories are NOT created in the
users directory space.  These directories are not included in the
general binary distribution. See ADDING OTHER GELLAB-II USERS below
for more details.


gellab/src        directory of source code for all GELLAB-II programs
DIRECTORY         CONTENTS
---------         --------
Makefile          Create whatever... for all GELLAB-II programs
Xpix              Xpix X-window image viewer (X11 only)
accppx            display image(s) given accession number(s)
annotate          spot annotation database (X10 only)
autopair          pair two GSFs to create GCF paired spot file w/o landmarks
camera            Datacopy camera (only for NCI-FCRDC camera)
cgelp2            interactive composite gel PCG database program
cmpgl2            pair two GSFs to create GCF paired spot file
disp11            flicker compare two 512x512 images.
dwrmap            draw Rmap plot from GSF file
getacc            get accession image and info for a set of gels
landmark          landmark two gels for LM database
libr              libGLAB.a GELLAB-II universal library
libXGLAB          libXGLAB.a X-window library for use with GELLAB-II.
libW              X11 widget object library based on Xt/Xaw.
libPDW            Public Domain X11 widget object library based on Xt/Xaw.
makjob            create UNIX batch scripts to run GELLAB-II
markgel           create Rmap .ppx image file for set of Rspots
mosaic            create mosaic .ppx image file of Rspot for set of gels
parg              X11 window manager interface to GELLAB-II programs
pgelrc            "pretty-print" the gel.rc GELLAB-II state file
plotn             draw plot from UGF file
ppx               (place to store original images)
ppx2ps            PPX to POSTSCRIPT converter
ppxcvt            general purpose conversion PPX from other image formats
```

```
ppxodt            PPX image debugger
sg2gii            gel spot segmenter to generate Gel Segmentation File
tek2psG           convert Tektronix 4010 codes to Postscript for GELLAB


(Only non-executable files are listed here - executables are the same
names as those listed in the 'src' directory above.)
gellab/bin        executable script files
DIRECTORY         CONTENTS
---------         --------
acc2patppx.awk    AWK program used by acclookup.do
acc2ppx.awk       AWK program used by acclookup.do
acclookup.do      match patterns  in gel.id accession file
cgelp2.hlp        on-line help file for cgelp2
gellab.1          GELLAB-II man pages (nroff format)
gellab.man        GELLAB-II man pages (nroff -man gellab.1) for printing
mapcar            map & eval each element of CCL list to UNIX command line to
mapcar2           map & eval each element of CCL list to UNIX command line to

gellab/bin/sun3   executable script files
DIRECTORY         CONTENTS
---------         --------
<executable binaries for a SUN3>

gellab/bin/sun4   executable script files
DIRECTORY         CONTENTS
---------         --------
<executable binaries for a SUN4>

gellab/demo       (selected demo. script files & logs of these scripts.)
DIRECTORY         CONTENTS
---------         --------
aux               demonstration aux directory
ann               demonstration ann directory
cgelp2.hrc        last history when used CGELP2
cmpgl2-demo.do    demonstrate cmpgl2.
demo11.do         run selected GELLAB-II acquisition and precomputed demo.
                  (select Xpix EXIT or Xpix2 QUIT to advance to next one).
gel.rc            GELLAB state file for this directory. Edit for your system.
gel.rc.DEMO       edited gel.rc plate for demo directory.
gen               demonstration gen directory
getacc-demo.do    demonstrate getacc.
id                demonstration id directory
landmark-demo.do  demonstrate landmark.
lms               demonstration lms directory
make-slides.do    create images for use by slides.do.
makts3-demo.do    demonstrate batch script generation for ts3???.do files.
pcg               demonstration pcg directory
ppx               demonstration ppx directory
print-slides.do   print images used in slide show on PostScript laser printer.
sg2gii-demo.do    demonstrate gel spot segmentation.
slides11.do       run images through a series of Xpix interactions.
                  (select Xpix EXIT or Xpix2 QUIT to advance to next one).
```

```
tmp              demonstration tmp directory
ts3.ccl          list of gels used to create ts3 database.
ts3cgl.do        script to build ts3pcg.pcg PCG DB
ts3cgl.gdo       script used by CGELP2 to build ts3pcg.pcg PCG DB
ts3cgl.log       log file for ts3cgl.do
ts3cmp.do        script to pair GSF files into GCF files
ts3drm.do        script to draw Rmap plots from GSF files
ts3lms.do        script to landmark gels for ts3 database.
ts3prc.do        script to segment gels to GSF and then pair GSFs to GCFs.
ts3prc.log       log file for ts3prc.do
ts3s01.do        generated ts3pcg.pcg script to make mosaics of SRL[1].
ts3s02.do        generated ts3pcg.pcg script to make mosaics of SRL[2].
                  .
                  .
                  .
ts3seg.do        script to segment gels to GSF files.
```

6.2.1 Other Public Domain UNIX programs used with GELLAB-II

Although GELLAB-II is relatively self contained, there are several public domain UNIX programs which make using GELLAB easier. These include *compress(1)/uncompress(1)* for compressing files and saving disk space. The X-Windows System [SchR86] is used to supply a portable windowing environment. GNU *emacs(1)* is our text editor of choice [StaR86]. *xdvi(1)* is a *TeX(1)* or *LaTeX(1)* *.dvi* file previewer, used with X-windows, which is used to display on-line documentation from some of the programs. The *xless(1)*, *xv(1)*, *xwd2ps(1)* programs from the MIT distribution is also included. We have modified it to handle GELLAB-II PPX files.

6.3 Reading the export tar tape

To install GELLAB you must have system manager privileges. While logged in as UNIX `root` create a GELLAB-II manager's login directory account:[1]

```
/home/gelmgr
```

etc - or whatever the convention is for your system. This is referenced in UNIX as ~`gelmgr` and each user will have an "environment" variable `$GELLABMANAGER` defined in the `.cshrc` file. `$GELLABMANAGER` points to this (see Section 6.7, page 6.7).

UNIX accounts needed

Create UNIX file system groups called `gelusr` (for both `gelmgr` and future GELLAB user accounts), and `gellab` for exclusive use by `gelmgr`. This must be

[1] If you don't know what `root` is - you probably should not login to it as you can easily destroy your file system - like a bull in a china shop.

setup by the UNIX system administrator in the `/etc/group` file. The ~`gelmgr` account has `gellab` as the major group and `gelusr` as the minor group.

Then login to the `gelmgr` account - you *must not* be in `root`! You can use the default `.login` and `.cshrc` files for the system until you overide these with those on the tape (although this is not necessary since you can get those on the tape soon enough). If you have a tyrant as a system manager who will not let you make a new account, then just put it in your own account.

The GELLAB-II tar file distribution tape should be read into the ~`gelmgr` home directory while logged in as ~`gelmgr` as follows:

```
cd ~gelmgr
tar xvfp /dev/rst8
```

which will restore the above `gellab` directory tree with the UNIX protections the same as on the tape. Depending on how your system was configure, the drive might be rst0, rst1, rst8, rst12, etc. Check with your system administrator. Watch the restore operation output on the screen (or put it into a file by instead typing `tar xvf /dev/rst8 >& restore.log`) to check for tape parity errors, disk overflow, etc..

6.4 Installing GELLAB-II for `gelusr` group

1. In order for users to access GELLAB-II `bin` and `demo` files (which is all that they need to do), do the following to force these directories and files into the UNIX `gelmgr` group. As mentioned, you need to create a UNIX group called `gelusr`. This is done (under `root` login) by adding the following line to the system file `/etc/group` where the name of the group is the *first* entry in bold-face and the members of the group (GELLAB-II users in this case) are in the remainder.

gelusr `gelmgr terry peter eric lemkin jem trygve`

For example, to add `joeUser` to the `gelusr` group, it would then look like:

gelusr `gelmgr terry peter eric lemkin jem trygve joeUser`

The `gellab` group is the primary group for the `gelmgr` account should be setup with just one entry as:

gellab `gelmgr`

6.5 Setting up proper access to `gelmgr` account

Then, login to the `gelmgr` account and execute the script `mkGelMgrAcct.do`. This will set automatically set up this account. Then you are done.

ALTERNATIVELY, you could do it by hand as follows (not recommended). The following material is mostly for your information - it is not really meant to be typed.

1. Setup correct protections and file groups so users can execute the files.

```
cd ~gelmgr
chgrp gellab  .*  * home-dir.files
chmod 700 * .*
chmod 750 gellab
cd gellab
chmod 700 *
chmod 750 bin demo
chmod 710 bin/sun*/*
chmod 640 bin/{gellab.1,gellab.man,*.dvi,*.hlp}
cd demo
chmod 750 *
chmod 640 *.gdo *.log
chmod 640 ./{ann,aux,gen,id,lms,pcg,ppx,tmp}/*
chmod 750 home-dir.files
chmod 640 home-dir.files/{.[a-z]*,.[A-Z]*,gel*}
chmod 750 home-dir.files/{.login*,.cshrc*,*.do}
```

2. If you *did not* execute the `mkGelMgrAcct.do` script, then set up your ~`gelmgr` home directory startup files.

```
cp ~gelmgr/home-dir.files/*  .
```

3. If you *do not* have `.login` and `.cshrc` defined in your home directory then just make a copy of the default ones,

```
cd ~gelmgr
cp .login.gellab .login
cp .cshrc.gellab .cshrc
cp .xinitrc.gellab .xinitrc
cp .twmrc.gellab .twmrc
cp .emacs.gellab .emacs
```

This last step running **pgelrc** initializes the `gel.rc` file in your home directory of your account.

4. If you *do* have these `.login` and `.cshrc` "dot" files, then you need to merge them using a text editor (such as *vi(1)* or *emacs(1)*). Merge `.login.gellab` into `.login` and similarly for `.cshrc.gellab` into `.cshrc`. You may also want to check them and make sure that the paths contained in these files are correctly for your system. Make sure that the `$GELLABMANAGER` environment variable is correct in `.cshrc`.

5. Re-evaluate the startup files

```
cd ~/
source .login
source .cshrc
```

You are now ready to run GELLAB-II in the ~`gelmgr` account.

6.6 Testing GELLAB-II - getting on the air

The following examples can be used to make sure that GELLAB-II is installed correctly. It is assumed that you are logged into the `gelmgr` or another GELLAB user account (*not* `root`)!

[1] Try to put up some gel images specified by their accession numbers.

[1.a] Display original gel images.

```
cd ~gelmgr/gellab/demo
cp gel.rc.DEMO gel.rc
pgelrc
accppx 324.1 378.2
```

(Select `EXIT` menu entry to exit the **Xpix** program which is called from **accppx**).

[1.b] Display segmented (extracted spot) gel images.

```
accppx 324.1 378.2 -prefix:z
```

(Select `EXIT` menu entry to exit).

[2] Run a "slide show" (see Section 2.1.2 page 76 by (2.a) creating a set of images, (2.b) displaying them. NOTE: since (2.a) was done previously and the files are on the distribution tape, you might want to do (2.b) first, see what they look like, then run (2.a) and (2.b) again to see how the images were created.

[2.a] Create images for use by `slides.do`,

```
make-slides.do
```

[2.b] Run images through series of **Xpix** interactions (select **Xpix** `EXIT` menu entry to advance to next one - or if using **Xpix2** select `QUIT` in the **FILE** menu.)

```
slides11.do
```

[2.c] Similar to `slides11.do`, but also run some acquisition demonstrations (select **Xpix** `EXIT` menu entry to advance to next one - or if using **Xpix2** select `QUIT` in the **FILE** menu.)

```
demo11.do
```

[3] Process the set of demonstration gels (optional at this point). Run UNIX script to run makjob on `ts3.ccl` to create a set of `ts3`*xxx*`.do` scripts

```
makts3-demo.do
```

[3.a] Run the created script to segment, pair-gels and create database

```
ts3prc.do
```

[3.b] Run the created **cgelp2** script to rebuild the database. Note that if you did step [3.a], it will automatically do this step [3.b].

```
cgelp2 -f ts3cgl.gdo
```

[4] Create and display Rmap from SPSS file previously created using the **cgelp2** program.

[4.a] Generate a Rmap 'm' image

```
markgel 324.1 ts3s02.sps
accppx 324.1 -prefix:m
```

[4.b] Generate Rmap 'm' image and display it

```
markgel 324.1 ts3s02.sps -Xpix
```

[4.c] Generate Zoomed Rmap 'm' image around Rspot[54] and display it. [Note: the spot number may change if the segmenter is improved.]

```
markgel 324.1 ts3s02.sps -Zoom:2X:54 -Xpix
```

[5] Create and display mosaic of LM C spot (Rspot[87]) from SPSS file previously created using the **cgelp2** program. [Note: the spot number may change if the segmenter is improved.]

[5.a] Generate mosaic 'w' image

```
mosaic 87 ts3s02.sps
accppx w00087.ppx
```

[5.b] Generate mosaic 'w' image and display it

```
mosaic 87 ts3s02.sps -Xpix
```

See documentation in Chapter 3 and in Tutorial Chapter 2 for other examples of particular programs.

6.7 Adding other GELLAB-II users

Once GELLAB-II is installed, you may want to add other users to your system and have them be able to access the GELLAB-II files. You can do this by: [1] logging into the new user account, and then [2] updating their `.cshrc` file etc. as discussed below. You may need to adjust some of the paths in `~/gellab/demo/gel.rc` to reflect the paths in your system.

1. First do the standard UNIX procedures to add another UNIX user while logged in as the super user (e.g. add the user to the `/etc/passwd` file and create a user directory with the same name). Have your system manager do this part if you are squimish about destroying your system! Then add that user (`joeUser` for purposes of discussion), to the `/etc/group` file for group `gelusr`. Logout of super user mode.

2. Log into the new user account (`~joeUser`). You need to change your UNIX start up files to reflect additions from GELLAB. If they exist, you might want to edit `.login` and `.cshrc` files in the user's home directory. Otherwise you can just copy the default GELLAB-II startup files. The following is already in the default GELLAB `.cshrc` file. However, if you do not use the default `.cshrc` but use your own instead, it is necessary to add an entry for the user's search path to GELLAB-II executables in `$GELLABMANAGER/gellab/bin`. This is typically done in their `.cshrc` file by editing in the following line (if it is not already there). Put this entry at the *end* of the `set $path= ...` entries.

```
        set $path = ($path $GELLABMANAGER/gellab/bin)
        set $path = ($path $GELLABMANAGER/gellab/bin/‘arch‘)
or
        set $path = ($path ~gelmgr/gellab/bin)
        set $path = ($path ~gelmgr/gellab/bin/‘arch‘)
or
        set $path = ($path ...to wherever you keep these files...)
```

The ‘`arch`‘ code is evaluated to be the architecture of your system (i.e. `sun3`, `sun4`, etc). This lets us put GELLAB on an NFS-mount network with different systems. Edit your `.cshrc` file to relect the proper UNIX shell environmental variable `setenv GELLABMANAGER /home/gelmgr` as well.

3. Then run the following program.

```
        cd ~/
        # Create top level gellab tree.
        pgelrc
        mkdir ~/gellab/demo

        # Create dummy demo directory pointing into that for gelmgr.
```

```
cd ~/gellab/demo
cp ~gelmgr/gellab/demo/gel.rc.DEMO gel.rc
pgelrc -change
cd ~/
```

You may enter different answers to the questions or you may just press the RETURN key for each question until the dialog is finished. If you will be generating files in the directory tree then they should be in your directory - not the ~gelmgr/gellab/demo/ path. This creates a gellab project tree in your home directory.
To reset or create a `gel.rc` for the GELLAB demonstration directory files, first `cd` to the directory you wish to operate from, then:

At this point everything should be setup. Note that in any routine user (not `gelmgr`) account, you may delete all of the demonstration subdirectory files. This is recommended when the user has worked through the demonstration and no longer needs it. To delete the users demonstration directory, *carefully* type

`rm -f -r ~/gellab/demo`.

6.8 GELLAB-II gellab(1) on-line man file documentation

GELLAB-II is briefly described in a single UNIX man pages documentation file `gellab(1)`. The gellab.1 file is distributed in *sim*`gelmgr/bin/gellab.1`. This may be viewed by typing

```
man gellab
```

If the `gellab.1` file is installed in `/usr/man/manl` or by typing

```
nroff -man ~gelmgr/gellab/doc/gellab.1 | more
```

or copied to a file which may be printed (if your printer can handle it) by

```
nroff -man ~gelmgr/gellab/doc/gellab.1 > gellab.man
```

6.9 User disk file system maintenance

Because no one has invented a disk file system with infinite capacity, and 2D gel analysis generates many large files, it is necessary to recover space on the disk to make room for further or future analyses. This recovery operation is called disk "maintenance". Some of the 2D gel analysis files are used only in intermediate parts of the analysis and so may be deleted before other more important files.

6.9.1 When to remove files from the disk

If there is plenty of room on the disk (found using the UNIX "`df .`" or "`du .`" command which gives the percentage of disk which is used), then one need not bother removing files. It is when space gets tight (if you are over say 95% full) or a project is finished that this issue must be addressed. Files can be deleted without previously saving them onto tape (or other media), in which case you can never recover them. In general that is *not* a good idea. Better, is to save them first and *then* delete the files. The concept of saving a set of files on some semi-permanent media such as magnetic tape is called "backing up files" or a *backup*.

6.9.2 Backup strategies

Normally when initially building the composite gel database, one keeps all image files on the system. After satisfactory image segmentation quantitation results have been obtained some of the image and other files can be removed. This can be done either of two ways: a) simply deleting files which can be recomputed quickly (i.e. within a reasonable time) if needed, or b) archiving files to magnetic tape prior to deleting them. The latter involves all primary data. Primary data is information obtained from interfaces to the computer (such as camera, accession information (on the experiment) supplied by the investigator, landmarking, etc.) and includes: the original gel images, gel accession database file, landmark database file, annotation database file, batch scripts and PCG gel database files as a minimum. In addition, GSF and GCF spot files, and additional batch scripts might also be considered to be primary data. Secondary data includes Rmaps and mosaics images and plots as well as SPSS files and other CGELP2 derived files - since all of these can be regenerated from the PCG DB. For that matter, the PCG DB can be regenerated, but may take the 1/2 hour or more and so might be better considered to be primary data.

6.9.3 What can go and what must stay

If one is continuing a project, but must make room on the disk, then some files can be removed (with the suggested backups being performed first). Files which can be deleted include any files which are easily regenerated.

The files which can be *deleted* include: *.gsf, .gcf* gel segmentation and gel comparison files; derived images including segmented images `z*`*.ppx*; central core images `c*`*.ppx*, Rmaps `m*`*.ppx*; mosaics `w*`*.ppx*; SPSS files *.sps*; SRL files *.srl*. In addition, the following generated files might also be removed: PCG `INQUIRE` search *.inq*, PCG tables *.tbl*, universal graphics plot files *.ugf*.

Files which should *stay* include: original gel images `a*`*.ppx*, accession database file *.id*, landmark database files *.lm*, annotation database files *.ann* and paged com-

posite gel database files *.pcg*. One may wish to keep some of the PCG DB derived SPSS and SRL files on the disk since regenerating them may be time consuming in some cases.

6.9.4 Backing up and restoring user project files

A simple way to backup (i.e. save) and restore files is using the UNIX *tar(1)* tape utility. It is important to restore the files the same way they were saved. By using the following convention the average user can perform this maintenance with minimum trouble. For the following scheme, the tar device must be on the same computer you are logged into. The tape devices are called: a) cartridge tape `/dev/rst0`, `/dev/rst8` or `/dev/rst12`; b) 9-track magtape `/dev/rmt0` or `/dev/rmt8`, etc. The `/dev/r...` indicates that the tape should rewind before writing, so that `/dev/n...` indicates the same drive is to be used but without initial rewinding. The following UNIX commands may be used to save a set of *user* files which are not needed for continuing the exploratory data analysis. The first script assumes that the original gel image files are high resolution gel scanned image data (>512x512) and that corresponding sampled lower-resolution original images (equal to 512x512) with `s` prefixes exist (use script `save-major-user-files.do`). If not, then *do not* remove the `a*.ppx` files (use `save-minor-user-files.do`) with the second script. The third script, `save-user-project.do`, saves and removes the entire project's `gellab` directory.[2][3]

```
#! /bin/csh
# File: save-major-user-files.do
# Save files not needed for further exploratory data analysis and
# then delete them in a user account.
cd ~/gellab
if(-e s*.ppx) tar cvf /dev/rst8 ./ppx/{a*}.ppx
tar cvf /dev/nst8 ./tmp/{c*,y*,z*}.ppx ./aux/{m*,w*,*.gsf,*.gcf}
tar cvf /dev/nst8 ./gen/{*.sps,*.inq,*.tbl,*.srl}
mt -f /dev/rst8 rewind
if(-e s*.ppx) rm ./ppx/a*.ppx
rm ./aux/{c*,m*,w*,z*}.ppx ./aux/{*.gsf,*.gcf}
rm ./gen/{*.sps,*.inq,*.tbl,*.srl,*.ugf}

#! /bin/csh
# File: save-minor-user-files.do
# Save files not needed for further exploratory data analysis and
# then delete them in a user account. Preserve original gel images.
cd ~/gellab
tar cvf /dev/rst8 ./tmp/{c*,y*,z*}.ppx ./aux/{m*,w*,*.gsf,*.gcf}
tar cvf /dev/nst8 ./gen/{*.sps,*.inq,*.tbl,*.srl,*.ugf}
mt -f /dev/rst8 rewind
```

[2] If you are compressing data files so that have a `.Z` file extension, then you must change these scripts to reflect the file name changes accordingly.

[3] Note that you should customize these scripts for your site.

```
rm ./aux/{c*,m*,w*,z*}.ppx ./aux/{*.gsf,*.gcf}
rm ./gen/{*.sps,*.inq,*.tbl,*.srl}
```

```
#! /bin/csh
# File: save-user-project.do
# Save all files in a project when the user is completely finished
# with the project and then deletes all files in that project.
cd ~/gellab
tar cvf /dev/rst8 .
rm -r -f ./
```

To *restore files* from the tape, mount the tape and do essentially the inverse operation as shown in the following scripts.

```
#! /bin/csh
# File: restore-major-user-files.do
# Save files not needed for further exploratory data analysis and
# then delete them in a user account.
cd ~/gellab
tar xvf /dev/rst8 ./ppx/{a*}.ppx
tar xvf /dev/nst8 ./tmp/{c*,y*,z*}.ppx ./aux/{m*,w*,*.gsf,*.gcf}
tar xvf /dev/nst8 ./gen/{*.sps,*.inq,*.tbl,*.srl}
mt -f /dev/rst8 rewind
```

```
#! /bin/csh
# File: restore-minor-user-files.do
# Restore files previously saved with 'save-minor-user-files.do'
cd ~/gellab
tar xvf /dev/rst8 ./tmp/{c*,y*,z*}.ppx ./aux/{m*,w*,*.gsf,*.gcf}
tar xvf /dev/nst8 ./gen/{*.sps,*.inq,*.tbl,*.srl,*.ugf}
mt -f /dev/rst8 rewind
```

To *restore a project*, first move to the project directory in the user account, and then execute the following script.

```
#! /bin/csh
# File: restore-user-project.do
# Restore all files in a project.
cd ~/gellab
tar xvf /dev/rst8 .
```

6.10 Installing the X11 X-Windows System on a SUN

*****NEEDS WORK*****

This chapter discusses how to install the logical X-Windows System version `X11` on your SUN system. This discussion only has been tested on a SUN running SUNOS-4.1.1 and may be different on other systems.

The public domain X-Windows System [SchR86] is used by many GELLAB-II programs for visualizing original and derived 2D gel images and graphs. GELLAB-II Version*beta*-level 1.3.52 uses X-Windows System version X11. For your convenience, the subset of the X11 binarys required for a SUN3 and SUN4 system may be included in the GELLAB-II distribution tape.

Three subdirectories are included in `X11-runtimes`: `bin init man`. The contents of `X11-runtimes/bin` must be copied to `/usr/local/bin`. The contents of `X11-runtimes/man` must be copied to `/usr/man/manl`. Files `README-FIRST` discusses this and tells you how to run `INSTALL.do`.

REDO

You must install the X11 binary files in `/usr/local/bin`. This can be done by hand or using the script `INSTALL.do`. Pay attention to: (1) logging in as `root` and (2) you must execute `INSTALL.do` in the same directory in which it appears. Do a `cd` to this directory before executing `INSTALL.do`. You should check the `README-FIRST`

REDO

directory as there may be a few things you might have to do by hand.

```
File: README-FIRST

SUBJECT: INSTALLING BINARY X11 for SUN
DATE: 12-11-89

You must install the X11 binary files in /usr/local/bin.
You must install the X11 fonts in /usr/local/lib/X/fonts.
To save space on the distribution tape, the fonts/* are compressed.
You must uncompress them to use it.


This can be done by hand or using the script INSTALL.do
Pay attention to
   1. logging in as root.
   2. you must execute INSTALL.do or do the commands in the same directory
      in which you have read the tape and INSTALL.do appears. Do a cd to this
      directory before executing INSTALL.do.


\smallskip
The {\tt INSTALL.do} file is:

\marginpar{\footnotesize\it\fbox{REDO}}   % -------------------------

{\footnotesize
\begin{verbatim}
    #!/bin/csh -v
    # File: INSTALL.do
    #
    # PURPOSE: copy files from the ./X11-runtimes/{bin,man,init} to /usr/local.
    # NOTES:
    #     1. You MUST be logged in as super user.
    #     2. Some of the X files MUST reside in /usr/local/bin since
    #        they are hard coded for that directory ... sorry...
```

```
#
# P.Lemkin - 12-11-89
#
# Create /usr/local if it does not exist.
if(!-e /usr/local) mkdir /usr/local

# Move the selected X11 and a few other runtimes into the /usr/local
# directory.
pwd

chown -R root bin man init font

if(! -e /usr/local) mkdir /usr/local
chmod 755 /usr/local
if(! -e /usr/local/bin) mkdir /usr/local/bin
chmod 755 /usr/local/bin
rcp -r bin/* /usr/local/bin
chmod -R 755 /usr/local/bin

if(! -e /usr/local/lib) mkdir /usr/local/lib
chmod 755 /usr/local/lib
if(! -e /usr/local/lib/X) mkdir /usr/local/lib/X
chmod 755 /usr/local/lib/X
if(! -e /usr/local/lib/X/fonts) mkdir /usr/local/lib/X/fonts
chmod 755 /usr/local/lib/X/fonts
rcp -r fonts/* /usr/local/lib/X/fonts
if(-e 6X11.onx.Z) uncompress -f /usr/local/lib/X/fonts/*.Z
chmod -R 644 /usr/local/lib/X/fonts

if(! -e /usr/man/manl) mkdir /usr/man/manl
chmod 755 /usr/man/manl
rcp man/* /usr/man/manl
chmod 644 /usr/man/manl/*

#
echo "That is all folks."
# -- the end ---
```

Chapter 7

REFERENCES

[AebR90] Aebersold, R., Leavitt, R. Sequence analysis of proteins separated by polyacrylamide gel electrophoresis: Towards an integrated protein database, *Electrophoresis* 11: 517-527, 1990.

[AmbA91] Amberger, A., Tontsch, U., Lemkin, P., Gabbiani, G., Bauer, H.C., Two cloned cerebral endothelial cell phenotypes: An in Vitro model for angiogenesis, in Proceedings on Angiogenesis, 1991.

[AndG86] Anderson, G., Anderson, P., `The UNIX C Shell Field Guide`, Prentice-Hall, Englewood Cliffs, NJ, 1986. [ISBN-0-13-937468-X]

[AndL84] Anderson, L., Anderson, N., Some perspectives on two-dimensional protein mapping, *Clin.Chem* 30(12): 1898-1905, 1984.

[AndL88] Anderson, N.L., Comprehensive Databases Constructed from 2D electrophoretic Measurements, in: Schafer-Nielson, C. (Ed.) `Electrophoresis '88`, VCH Publ., Weinheim, 313-321, 1988.

[AndL88a] Anderson, N.L., `Two-Dimensional Electrophoresis. Operation of the ISO-DALT System`, Large Scale Biology Press, Rockville, MD., pp 162. [ISBN-0-945532-00-8]

[AndN79] Anderson, N.G., Anderson, N.L., Molecular anatomy, *Behring Inst.Mitt* 63: 169-210 ,1979.

[AndN81] Anderson, N.L., Taylor, J., Scandora, A.E., Coulter, B.P., Anderson, N.L., The TYCHO System for Computer Analysis of Two-Dimensional Gel Electrophoresis Patterns, *Clin. Chem.* 27: 1807-1821, 1981.

[AppR87] Appel, R.D., MELANIE un systeme d'analyse et d'interpretation automatique d'image de gels d'electrophorese bidimensional: systemès experts et apprentissage automatique, Thesis #2241, Dept. D'informatique, Univ. Geneva, 1987.

[ArmP71] Armitage, P., `Statistical Methods in Medical Research`, Blackwell Sci. Pub., Oxford, 1971. [ISBN-0-632-05430-1]

[BakR82] Baker, R.A., *Datamation*, May, 139-142, 1982.

[BalD82] Ballard, D.H., Brown, C.M., `Computer Vision`, Prentice Hall, Englewood Cliffs, NJ, 1982. [ISBN-0-13-165316-4]

[BauG90] Bauw, G., Rasmussen, H.H., van Den Buicke, M., van Damme, J., Puype, M., Gesser, B., Celis, J.E., Vanderkerckhove, J., *Electrophoresis* 11: 528-537, 1990.

[BerT90] Bergman, T., Jörnvall, H., Electropblotting of polypeptides onto glass fiber filters for direct sequence analysis after sodium dodeculsulfate-polyacrylamide electrophoresis, *Electrophoresis* 11: 569-572, 1990.

[BloS86] Blose, S.H., The mouse NIH/3T3 cell line protein database developed form computer analyzed two dimensional gels: key protein identification by experiments and amino acid ratios, pp 552-555, In M. J. Dunn, `Electrophoresis '86`, VCH Publ. Deerfield Beach, FL, 1986. [ISBN-3-527-26566-X]

[BraR84] Bravo, R., Two-dimensional gel electrophoresis: a guide for the beginner, pp 4-34, in `Two-Dimensional Gel Electrophoresis of Proteins`, J.E. Celis, R.Bravo (Eds), Academic Press, NY, 1984. [ISBN-0-12-164720-X]

[CarG74] Carman, G., Lemkin, P., Lipkin, L., Shapiro, B., Schultz, M., Kaiser, P. A Real Time Picture Processor for Use in Biological Cell Identification - II Hardware Implementation, *J. Histochem. Cytochem.* 22: 732-740, 1974.

[CelJ89a] Celis, J.E., Ratz, G.P., Madsen, P., Gesser, B., Lauridsen, J.B., Brogarrd Hansen, K.P., Kwee, S., Holm Rassmussen, H., Nielsen, H.V., Cruger, Basse, B., Leffers, H., Honore, B., Moller, Celis, A., Computerized, comprehensive databases of cellular and secreted proteins from normal human embryonic lung MRC-5 fibroblasts: Identification of transformation and/or proliferation sensitive proteins, *Electrophoresis* 10(2): 76-115, 1989.

[CelJ92] Celis, J.E., Rassmussen, H., Madsen, P., Leffers, H., Honore, Dejgaard, K., Gesser, B., Olsen, E., Gromov, P., Hoffmann, H.J., Nielsen, M., Celis, A., Basse, B., Lauridsen, J.B., Ratz, G.P., Nielsen, H., Andersen, A.H., Walbun, E., Kajaergaard, I., Puype, M., Van Damme, J., Vanderkerckhove, J., The

human keratinocyte two-dimensional gel protein database (update 1992): Towards an integrated apoproach to the stuidy of cell proliferation, differentiaon and skin diseases. *Electrophoresis* 13(12) 893-959, 1992.

[ChaL90] Chang, L.,-H., Hsieh, J.-C., Chen, W.-L., Tam, M.F., Identification of rat liver glutathione S-transferase Yb subunits by partioal N-terminal sequencing after electropblotting of proteins onto a polyvinylidene difluoride membrane from an analytical isoelectric focusing gel, *Electrophoresis* 11: 589-593, 1990.

[ChoT90] Choli, T., Wittmann-Liebold, B., Protein blotting followed by microsequencing, *Electrophoresis* 11: 562-568, 1990.

[DelC88] Delisi, C., Computers in molecular biology: current applications and emerging trends, *Science* 240:, 47-52, April 1, 1988.

[DunM83] Dunn, M.J., Burghes, M.H.M., High resolution two-dimensional polyacrylamide electrophoresis. II. Analysis and applications, *Electrophoresis* 4: 173-189, 1983.

[DunM87] Dunn, M.J., Two-dimensional polyacrylamide gel electrophoresis, In A. Chrambach, M.J. Dunn, B.J. Radola (eds), `Advances in Electrophoresis`, VCH Pub., NY, 1987, 1-109. [ISBN-0932-3031]

[DCRT79] DCRT, DECsystem-10 Omnigraph Display Manual, CCB, DCRT, NIH, Bethesda, Md. 20205, Oct, 1979.

[EckC88] Eckerskorn, C., Jungblut, P., Mewes, W., Klose, J., Lottspeich, F., Identification of mouse brain proteins after two-dimensional electrophoresis and electroblotting by microsequencing analysis and amino acid composition, *Electrophoresis* 9: 830-838, 1988.

[EckC90] Eckerskorn, C., Lottspeich, F., Combination of two-dimensional gel electrophoresis with microsequencing and amino acid composition analysis: Improvement of speed and sensitivity in protein characterization, *Electrophoresis* 11: 554-561, 1990.

[FisD86] Fisher, D., Langley, P., in Gale, W.A., (Ed) *Artificial Intelligence and Statistics*, Addison-Wesley, Reading, MA, 1986, pp. 77-116.

[FunM87] Funk, M., Appel, R.D., Roch, C., Hochstrasser, D., Pellegrini, C., Muller, A.F., Knowledge acquisition in expert system assisteded diagnosis: a machine learning approach, pp 97-103, in `Lecture Notes in Medical Information: Proc. AIME 87`, Marseille, J. Fox, M. Fieschi, R. Engelbrech (Eds), 1987.

[GarJ79] Garrels, J.I., Two dimensional gel electrophoresis and computer analysis of proteins synthesized by clonal cell lines, *J.Biol.Chem* 254: 7961-7977, 1979.

[GarJ84] Garrels, J.I., Farrar, J.T., Burwell IV, C.B., The QUEST System for Computer-Analyzed Two Dimensional Electrophoresis of Proteins, pp 37-91, in `Two-Dimensional Gel Electrophoresis of Proteins`, J.E. Celis, R.Bravo (Eds), Academic Press, NY, 1984. ISBN-0-12-164720-X.

[GioD92] Giometti D.F., Taylor, J., Tollaksen, S.L., Mouse liver protein database: A catalog of proteins detected by two-dimensional gel electrrophoresis, *Electrophoresis* 13(12): 970-991 (1992).

[HayF83] Hayes-Roth, F., Waterman, D.A., Lenat, D.B., (Eds) *Building Expert Systems*, Addison-Wesley, Reading, MA, 1983. [ISBN-0-201-10686-8]

[HocD88] Hochstrasser, D.F., Harrington, M.G., Hochstrasser, A.C., Miller, M.J., Merril, C.R., Methods for increasing the resolution of two-dimensional protein electrophoresis, *Anal. Biochem.* 173: 424-435, 1988.

[HocD92] Hochstrasser D.F., Frutiger, S., Paquet, N., Bairoch, A., Ravier, F., Pasquali, C., Sanchez, J.C., Tissoe, J.D., Bjellqvist, B., Vargas, R., Appel, R.D., Hughes, G.J., Human liver protein map: A reference database establuished by microsequencing and gel comparison, *Electrophoresis* 13(12): 992-1001 (1992).

[HirH90] Hirano, H., Watanabe, T., Microsequencing of proteins electrotransferred onto immobilizing matrices from polyacrylamide gel electrophoresis: Application to an insoluble protein, *Electrophoresis* 11: 573-580, 1990.

[HooL87] Hood, L., Smith, L., Genome sequencing: how to proceed, *Issues Sci. Tech.* Spring: 36-47, 1987.

[HowR83] Howard, R.J., Aley, S.B., Lemkin, P.F., High resolution comparison of Plasmodium Knowlesi clones of different variant antigen phenotypes by 2D gel electrophoresis and computer analysis, *Electrophoresis* 4: 420-427, 1983.

[HurP78] Hurley, P.M., Catsimpoolas, N., Wogan, G.N., Bidimensional electrophoresis of nuclear chromosomal proteins from aflatoxin-induced liver cancer. In `Electrophoresis'78`, N. Catsimpoolas (Ed), Elsevier North-Holland, Amsterdam, 283-296, 1978.

[JunP90] Jungblut, P., Eckerskorn, C., Lottspeich, F., Klose, J., Blotting efficiency investigated by using two-dimensional electrophoresis, hydrophobic membranes and proteins from different sources, *Electrophoresis* 11: 581-588, 1990.

[KenT88] Kennedy, T.E., Gawinowicz, Barzilai, A., Kandel, E.R., Sweat, J.D., Sequencing of proteins from two-dimensional gels by using *in situ* digestion and transfer of peptides to polyvinylidene difluoride membranes: Applictation to proteins associated with sensitization in *Aplysia*, *Proc. Nat. Acad. Sci. USA* 85: 7008-7012, 1988.

[KloJ84] Klose, J., Zeindl, E., An attempt to resolve all of the various proteins in a single human cell type by two dimensional electrophoresis: I. Extraction of all cell proteins, *Clin.Chem.* 30(12): 2014-2021, 1984.

[KloJ89a] Klose, J., Systematic analysis of the total proteins of a mammalian organism: Principles, problems and implications for sequencing the human genome, *Electrophoresis* 10(2): 140-152, 1989.

[KocS88] Kochen, S.G., Wood, P.H., `UNIX Shell Programming`, Hayden Books, Indianapolis, 1988. [ISBN-0-8104-6309-1]

[KuiR86] Kuick, R., Sing, C., Hanash, C., An interactive multigel matching strategy. pp 91-96 In Galteau, M.M., Siest, G. (Eds), `Recent Progress in Two-Dimensional Electrophoresis`, Univ. Press Nancy, France, 1986. [ISBN-2-86480-248-1]

[LemP74] Lemkin, P., Carman, G., Lipkin, L., Shapiro, B., Schultz, M., Kaiser, P. A Real Time Picture Processor for Use in Biological Cell Identification - I System Design, *J. Histochem. Cytochem.* 22: 725-730, 1974.

[LemP77] Lemkin, P., Carman, G., Lipkin, L., Shapiro, B., Schultz, M. Real Time Picture Processor - Description and Specification. NCI/IP Technical Report #7a, Nat. Tech. Info. Serv. PB269600/AS, 1977.

[LemP78] Lemkin, P. Buffer Memory Monitor System for Interactive Image Processing. NCI/IP Technical Report #21b, Nat. Tech. Info. Serv. PB278789 (listing PB278790), 1978.

[LemP79a] Lemkin,P., Merril, C., Lipkin, L., Van Keuren, M., Oertel, W., Shapiro, B., Wade, M., Schultz, M., Smith, E., Software aids for the analysis of 2D gel electrophoresis images, *Computers and Biomedical Research* 12: 517-544, 1979.

[LemP79b] Lemkin, P., Shapiro, B., Lipkin, L., Sklansky, J., Schultz, M., Preprocessing of electron micrographs of nucleic acid molecules for automatic analysis by computer. II Noise removal and gap filling, *Comp. Biomed. Res.* 12: 615-630, 1979.

[LemP80a] Lemkin, P., Lipkin, L., BMON2 - A distributed monitor system for biological image processing, *Computer Programs in Biomedicine* 11: 21-42, 1980.

[LemP80b] Lemkin, P., Lipkin, L., Merril, C., Shiffrin, S., Protein abnormalities in macrophages bearing asbestos, *Environmental Health Perspectives* 34: 75-89, 1980. NIEHS Conf. Medical Aspects of Mineral Fibers, 1979.

[LemP81a] Lemkin, P., Lipkin, L., GELLAB: A computer system for 2D gel electrophoresis analysis. I. Segmentation and preliminaries., *Computers in Biomedical Research* 14: 272-297, 1981.

[LemP81b] Lemkin, P., Lipkin, L., GELLAB: A computer system for 2D gel electrophoresis analysis. II. Spot pairing, *Computers in Biomedical Research* 14: 355-380, 1981.

[LemP81c] Lemkin, P., Lipkin, L., GELLAB: A computer system for 2D gel electrophoresis analysis. III. Multiple gel analysis, *Computers in Biomedical Research* 14: 407-446, 1981.

[LemP81d] Lemkin, P.F., Lipkin, L.E., GELLAB: Multiple 2D Electrophoretic Gel Analysis, in `Electrophoresis '81`, R. Allen, Arnaud (eds), W. De Gruyter, New York. 401-411, 1981.

[LemP82a] Lemkin, P.F., Lipkin, L.E., Lester, E.P., Extensions to the GELLAB 2D Electrophoresis Gel Analysis System. Paper given at "Clinical Applications of 2D Electrophoresis", Mayo Clinic, Nov. 15-18, 1981, *Clinical Chemistry* 28: 840-849, 1982.

[LemP83a] Lemkin, P.F., Lipkin, L.E., Database Techniques for 2D Electrophoretic Gel Analysis, in `Computing in Biological Science`, Elsevier/North-Holland, M. Geisow, A. Barrett (eds), 181-226, 1983.

[LemP83b] Lemkin, P.F., Lipkin, L.E., 2D Electophoresis gel database analysis: Aspects of data structures and search strategies in GELLAB, *Electrophoresis* 4: 71-81, 1983. Presented at Argonne Workshop on Technical advances in 2D electrophoresis and clinical applications of the technique", Aug.29-Sep.1, 1982.

[LemP84b] Lemkin, P., Sonderegger, P., Lipkin, L., Identification of coordinate pairs of polypeptides: A techniques for screening of putative precursor product pairs in 2D gels, *Clinical Chemistry* 30: 1965-1971, 1984.

[LemP85] Lemkin, P., PSAIL - A portable SAIL compiled translator for C environments, *Computer Language* 2: 39-45, 1985.

[LemP88a] Lemkin, P.F., Xpix - An image processing system for X windows, In preparation, 1-16, March, 1988.

[LemP88b] Lemkin, P. F., The GELLAB Papers - A Collection of Papers Describing the GELLAB-I System, NCI/FCRF, April 12, 1988.

[LemP88c] Lemkin, P.F., PSAIL: A Portable SAIL to C Compiler - Description and Tutorial, *SIGPLAN Notices* 23(10): 149-171, 1988.

[LemP89a] Lemkin, P.F., Lester, E.P., Database and Search Techniques for 2D Gel Protein Data: A Comparison of Paradigms For Exploratory Data Analysis and Prospects for Biological Modeling, *Electrophoresis*, 10(2): 122-140, 1989.

[LemP89b] Lemkin, P.F., GELLAB-II, A workstation based 2D electrophoresis gel analysis system, in *Two-Dimensional Electrophoresis*, T. Endler, S. Hanash (Eds), Vienna Austria, Nov 8-11, 1988, VCH Press, W.Germany. 53-57, 1989.

[LemP91] Lemkin, P., Rogan, P., Automatic detection of noisy spots in two-dimensional Southern Blots, *App. Theoretical Electrophoresis* **2**, 141-149, 1991.

[LemP92] Lemkin, P., Representations of protein patterns from 2D gel electrophoresis databases, In `The Visual Display Of Biological Information`, C. Pickover (ed), World Publishers, Teaneck, NJ, in press, 1992.

[LemP93] Lemkin, P.F., J.E., Kyle M. Upton, K.M., Splitting merged spots in 2D PAGE gel images, *Applied and Theoretical Electrophoresis*, **3**, 163-172, 1993.

[LesE80] Lester, E.P., Lemkin, P., Cooper, H.L., Lipkin, L.E., Computer-Assisted Analysis of Two-Dimensional Electrophoresis of Human Peripheral Blood Lymphocytes, *Clinical Chemistry* 26: 1392-1402, 1980.

[LesE81a] Lester, E.P., Lemkin, P., Lipkin, L.E., Cooper, H.L., Two-Dimensional Electrophoretic Analysis of Protein Synthesis in Resting and Growing Lymphocytes in Vitro, *J. Immunology* 126: 1428-1434, 1981.

[LesE81b] Lester, E.P., Lemkin, P.F., Lipkin, L.E., New Dimensions in Protein Analysis - 2D Gels Coming of Age Through Image Processing, Invited paper, *Analytical Chemistry* 53: 390A-397A, 1981.

[LesE82a] Lester, E.P., Lemkin, P.F., Lipkin, L.E., A two-dimensional Gel Analysis of Autologous T and B lymphoblastoid Cell lines, *Clinical Chemistry* 28: 828-839, 1982.

[LesE82b] Lester, E.P., Lemkin, P.F., Lowery, J.F., Lipkin, L.E., Human leukemias: A preliminary 2D electrophoretic analysis, *Electrophoresis* 3: 364-375, 1982.

[LesE83] Lester, E.P., Lemkin, P.F., Lipkin, L.E., States of differentiation in leukemias: A 2D gel analysis. In `Chromosomes and Cancer: From Molecules to Man`, Proceedings of 5th Annual Bristol Myers Symposium on Cancer Research. Academic Press, 226-245, 1983.

[LesE84a] Lester, E.P., Lemkin, P.F., Lipkin, L.E., Protein indexing in leukemias and lymphomas, *NY Acad. Sci.* 428: 158-172, 1984.

[LesE84b] Lester, E.P., Lemkin, P.F., A 'GELLAB' computer assisted 2D gel analysis of states of differentiation in hematopoietic cells, In Neuhoff, V. (Ed): In `Electrophoresis '84`, 1984. Basel, Switzerland, Springer-Verlag Chemie, pp. 309-311.

[LevR90] Levenson, R.M., Anderson, G.M., Cohn, J.A., Blackshear, P.J., Giant two-dimensional gel electrophoresis: Methodological update and comparison with intermediate format gel systems. *Electrophoresis* 11(3): 269-279, 1990.

[LipL79] Lipkin, L., Lemkin, P., Shapiro, B., Sklansky, J., Preprocessing of Electron Micrographs of Nucleic Acid Molecules for Automatic Analysis by Computer, *Comp. Biomed. Res.* 12: 279-289, 1979.

[LipL80a] Lipkin, L.E., Lemkin, P.F., Database techniques for multiple PAGE (2D gel) analysis, *Clinical Chemistry* 26: 1403-1413, 1980.

[LipL80b] Lipkin, L.E.,Cellular effects of asbestos and other fibers, *Environ. Health Perspect. 34*, 91-102, 1980.

[LevR90] Levenson, R.M., Anderson, G.M., Cohn, J.A., Blackshear, P.J., Giant two-dimensional gel electrophoresis: Methodological update and comparison with intermediate-format gel systems, *Electrophoresis* **11**, 269-279, 1990.

[LutW78] Lutin, W.A., Kyle, C.F., Freeman,J.A., in `Electrophoresis '78`, N. Catsimpoolas (Ed)), Elsevier/North-Holland, New York. 93-106, 1978.

[ManR88] Mann, R.C., Mansfield, B.K., Selkirk, J.K., Two dimensional gel electrophoresis of cytoplasmic proteins from friend cell erythroleukemia cells chemically induced to undergo terminal erythroid differentiation, *Cancer. Res.* 48: 1110-1118, 1988.

[MerC79] Merril, C.R., Switzer, R.C., Van Keuren, M.L, Trace polypeptices in cellular extracts and human body fluids detected by two-dimensional electrophoresis and a highly sensitive silver stain, *Proc.Natl.Acad.Sci.USA* 76: 4335-4339, 1979.

[MerC87] Merril, C.R., Detection of proteins separated by electrophoresis, In A. Chrambach, M.J. Dunn, B.J. Radola (eds), `Advances in Electrophoresis`, VCH Pub., NY, 1987, 111-139.

[MilR81] Miller, R.G, `Simultaneous Statistical Inference`, Springer-Verlag, NY, 1981.

[MilM84] Miller, M.J., Olson, A.D., Thorgeirsson, S.S., Computer analysis of two dimensional gels: automatic matching, *Electrophoresis* 5: 297-303, 1984.

[MilM86] Miller, M.J., Quantitative analysis of two-dimensional gel electrophoretograms: strategies and requirements for computerized analysis, *Exp. Bio. and Med.* 19: 235-260, 1986.

[NatM66] Natrella, M.G., `Experimental Statistics - NBS Handbook 91`, August, 1963 (1966 Edition), U.S. Govt. Printing Office, Stock # 003-003-00135-0.

[MyrJ93] Myrick, J.E., Lemkin, P.F., Robinson, M.K., Upton, K.M., Comparison of the Bio Image Visage^{TM} 2000 and the GELLAB-II Two-Dimensional Electrophoretic Analysis Systems, submitted, 1993.

[NeiF89a] Neidhardt, F.C., Appleby, D.B., Sankar, P., Hutton, M.E., Phillips, T.A., Genomically linked cellular protein databases derived from two-dimensional polyacrylamide gel electrophoresis, *Electrophoresis* 10(2): 116-122, 1989.

[NieH75] Nie, H.H., Hull, C.H., Jenkins, J.G., et al., `SPSS - statistical package for the Social Sciences`, McGraw Hill, New York. 1975.

[OlsA88] Olson, A.D., Miller, M.J., Elsie 4: Quantitative computer analysis of sets of two dimensional gel electrophoretograms, *Anal. Biochem.* 169: 49-70, 1988.

[OFarP75] O'Farrell, P.H., High resolution two-dimensional electrophoresis of proteins, *J.Biol.Chem* 250: 4007-4021, 1975.

[ParR88] Parker, R.A., Rothenberg, R.B., Identifying important results from multiple statistical tests, *Statistics in Medicine* 7: 1031-1043, 1988.

[RasH92] Rasmussen, H.H., Van Damme, J., Puype, M., Gesser, B., Celis, J.E., Vandekerckkhove, J., Microsequences of 145 proteins recorded in two-dimensional gel protein diatabase of normal human epidermal keratinocytes, *Electrophoresis* 13(12): 960-969, (1992).

[Reis76] Reiser, J.F., SAIL, Stanford University Artificial Intelligence Laboratory memo AIM-289, August, 1976. Also available from U.S. Dept. Commerce. Nat. Tech. Inform. Serv. No. AD-A045-102, Springfield, Va., 1976.

[RidG84] Ridder, G., VonBargen, E., Burgard, D., Pickrun, H., Williams, E., Quantitative analysis and pattern recognition of two dimensional electrophoresis gels, *Clin.Chem* 30(12): 1919-1924, 1984.

[RogP91] Rogan, P., Lemkin, P., Klar, A., Singh, J., Strathern, J.N., Two-dimensional agarose gel electrophoresis of restriction-digested genomic DNA., *Methods: A Companion to Methods in Enzymology* **3**(2) 91-97, 1991.

[RosA77] Rosenfeld, A., Kak, A. `Digital Image Processing`, Academic Press, New York, 1977.

[SAS85] SAS Institute Inc., `SAS - User's Guide: Statistics`, SAS Inst. Inc., Cary, NC. 1985.

[SchA77] Schwartz, A. A., Soha, J. M., Variable Threshold Zonal Filtering, *Applied Optics* 16: 1779-1781, 1977.

[SchR86] Scheifler, R.W., Gettys, J., The X window system, *ACM.Trans.Graph.* 5(2): #63, 1986.

[SchR88] Scheifler, R.W., Gettys, J., Newman, R., `X Window System`, Digital Press, Bedford, MA, pp701, 1988.

[SkoM86a] Skolnick, M.M., Application of morphological transformations to the analysis of two-dimensional electrophoretic gels of biological materials, *Comp.Vis.Graph.Img.Proces.* 35: 306-332, 1986.

[SkoM86b] Skolnick, M.M., Neel, J.V.,in Harris, H., Hirschhorn, K. (Eds), *Advances in Human Genetics*, Vol 15, Plenum Pub., NY, pp. 55-160, 1986.

[SinJ82] Sinclair, J., Rickwood, D., Two-dimensional gel electrophoresis, In B.D. Hames and D. Rickwood, `Gel Electrophoresis of Proteins: a practical approach`, IRL Press Ltd, London, 1982.

[SmiK86] Smith, K.A., Dunn, M.J., Computer analysis of 2D gels using statistical methods and parallel processing, pp 560-562, In M. J. Dunn, `Electrophoresis '86`, VCH Publ. Deerfield Beach, FL, 1986.

[SneG80] Snedecor, G.W., Cochran, W.G., `Statistical Methods`, Iowa State Univ. Press, Ames, Iowa, 7th edit., 1980.

[SneP73] Sneath, P.H.A., Sokal, R.R., `Numerical Taxonomy`, W.H. Freeman and Company, 1973.

[SonP85] Sonderegger, P., Lemkin, P., Lipkin, L., Nelson, P., Differential modulation of the expression of axonal proteins by non-neuronal cells and the peripheral and central nervous system, *EMBO J.* 4: 1395-1401, 1985.

[SonP86] Sonderegger, P., Lemkin, P.F., Lipkin, L.E., Nelson, P.G., Coordinate regulation of the expression of axonal proteins by the micro-environment, *Developmental Biology* 118: 222-232, 1986.

[SpeR84] Spencer, R., Cluster Analysis, *Byte*, 129-130 and 423-426, Sept., 1984.

[SprR85] Sproull, R.F., Sutherland, W.R., Ullner, M.K., "Device-independent Graphics", McGraw Hill, NY, 1985.

[StaR86] Stallman, R., GNU Emacs Manual, Version 18 for UNIX users, Free Software Foundation, Cambridge, MA, December 1986.

[StoE89] Stoeckli, E.T., Lemkin, P.F., Kuhn, T.B., Ruegg, M.A., Heller, M., Sonderegger, P., Axonally Secreted Proteins: I. Identification of Proteins Secreted from Axons of Embryonic Dorsal Root Ganglia Neurons, *Eur.J.Biochem.*, 180: 249-258, 1989.

[SweJ89] Sweatt, J.D., Kennedy, T.E., Wagner-Smith, K., Gawinowicz, M.A., Barzilai, Karl, K., Kangel, E.R., Development of a database of amino acid sequences for proteins identified and isolated on two-dimensional a]polyacrylamide gels, *Electrophoresis* 10(2): 152-157, 1989.

[TarP87] Tarroux P., Vincens, P., Rabilloud, T., HERMeS: A second generation approach to the automatic analysis of two-dimensional electrophoresis gels. Part V: database organization and management, *Electrophoresis* 8: 187-199, 1987.

[TarP89] Tarroux P., Vincens, P., Meyer, J.A., Processing of two-dimensional gel electrophoresis data, in *Two-Dimensional Electrophoresis*, T. Endler, S. Hanash (Eds), Vienna Austria, Nov 8-11, 1988, VCH Press, W.Germany. 68-71, 1989.

[TayJ82] Taylor, J., Anderson, N.L., Scandora, A.E., Willard, K.E., Anderson, N., Design and implementation of a prototype human protein index, *Clin.Chem.* 28(4): 861-866, 1982.

[TayJ83] Taylor, J., Anderson, N.L., Anderson, N.G., Numerical measures of two dimensional gel resolution and positional reproducibility, *Electrophoresis* 4: 338-346, 1983.

[TeiD89] Teichroew, D., Hanash, S., Kuick, R., Ting, J., Beaver, A., A strategy for the design of a laboratory information processing system for 2-D gel studies, in *Two-Dimensional Electrophoresis*, T. Endler, S. Hanash (Eds), Vienna Austria, Nov 8-11, 1988, VCH Press, W.Germany. 37-52, 1989.

[TemP90] Tempst, P., Link, A.J., Riviere, L.R., Fleming, M., Elicone, C., Internal sequence analysis of proteins separated on polyacrylamide gels at the submicrogram level: improved methods, applications and gene cloning strategies, *Electrophoresis* 11: 537-553, 1990.

[TukJ86] Tukey, J., in Gale, W.A., (Ed) *Artificial Intelligence and Statistics*, Addison-Wesley, Reading, MA, 1986, pp. 401-409.

[VanB92] VanBogelen R.A., Sankar, P., Clark, R.L, Bogan, J.A., Neidhardt, F.C., The gene-protein databse of *Escherichia Coli*: Edition 5, *Electrophoresis* 13(12): 1014-1054 (1992).

[VinP87] Vincens, P., Tarroux P., HERMeS: A second generation approach to the automatic analysis of two-dimensional electrophoresis gels. Part IV: database organization and management, *Electrophoresis* 8: 173-186, 1987.

[VinP89] Vincens, P., Tarroux, P.,2D-gel image evaluation and processing with HERMeS, in *Two-Dimensional Electrophoresis*, T. Endler, S. Hanash (Eds), Vienna Austria, Nov 8-11, 1988, VCH Press, W.Germany. 62-71, 1989.

[VoKP81] Vo, K-P, Miller, M.J., Geiduschek, E.P., Nielsen, C., Xuong, N.H., Computer Analysis of Two-Dimensional Gels., *Anal. Biochem.* 112: 258-271, 1981.

[WanP88] Wang, P., `An Introduction to Berkeley UNIX`, Wadsworth Publ. Co., Belmont, Calif., 1988.

[YouDA84] Young, D.A., Advantages of separations on 'giant' two-dimensional gels for detection of physiologically relevant changes in the expression of protein-gene products, *Clinical Chemistry* 30: 2104-2108, 1984.

[YunM92] Yun, M., Hood, L, Harrington, M., Human cerebrospinal fluid protein databse: Edition 1992, *Electrophoresis* 13(12): 1002-1013(1992).

Appendix A

Glossary

A glossary of some of the terms used in this book is given here as an aid in understanding the use of computer systems for analyzing 2D gels. Although it is particularly applicable to GELLAB-II, the general concepts and some of the terms are generally applicable to the field of computer analysis of 2D gels.

accession is a verb meaning to enter data, as in "to accession a set of gels". In the context of GELLAB-II, it means to enter information about a gel (or gels) and its image into the GELLAB-II database for the project dealing with those gels. Information includes: image accession number (see below), image file name, experiment related information, grayscale to OD calibration, image computing window and image size.

accession file is a database file (*.id* extension) which contains all of the accession information for the user's database. It is used by many GELLAB programs and is generated by **getacc** and may be edited with **getacc** `-edit -ask:q` or a text editor.

accession information is non-image data associated with a gel at the time it is scanned with **getacc.** Some typical accession information is:

```
ACCESSION-# / PATIENT / B.DAY / RACE&SEX / EXPER-DATE/ EXPER-# /
CULTURE-REAG / AMPH,GEL / INTRVL-BEFR-LBLNG /LBLNG-ISOTOPE /
DURTN-LABEL / DURTN-OF-EXPSR / STUDY / PPX-FILE# /TAPE-# /
OPT.-BACKUP-TAPE-# / CAMERA,LENS,DISTANCE / EXPRMENTER*
```

accession number is a 4 digit number XXXX with a 1 place fraction E denoted XXXX.E assigned by the user to a particular exposure of a gel at the time it is entered (accessioned) into the system. If different exposures are made of the

same gel the XXXX part would be the same but the E part would differ. Eg. 01234.1, 01234.2, etc. The E value of 9 is reserved for synthetic Cgel′ data files generated by **cgelp2.** The default value of E is '1' (e.g. `524` is `0524.1`).

ambiguous pair (AP) is a spot pairing label assigned to two spots from different gels such that one of the spots pairs better with another spot from the other gel. Therefore the spot with the worse pairing is *ambiguous* as defined by the **cmpgl2** program.

annotation is additional on-line documentation associated with gels, spots, sets of spots or sets of gels. It can be searched on a relational basis to retrieve either the annotation as a function of object, spot(s) or gel(s), or objects as a function of annotation fragment.

background image is the **sg2gii** intermediate image computed by first subtracting the segmented spots from the original gel image. Then, the zonal notch filter [LemP82a] is applied to estimate gel background which is estimated from that region of the gel not containing spots.

backup is the process using one of the magnetic tape utilities, such as the UNIX *tar(1)* program, to save subsets of files on the tape so that they can be recovered (again using `tar`) at a later date. After files are backed up on tape, they can be deleted from the disk.

batch script is a file which can be executed by UNIX. These can be generated automatic BATCH job by **getacc** or **makjob.** The script is a means in which the user can run programs in a background mode even if the user is not logged into the computer. For example, if a user batch job `ts3prc.do` is to be started, the user would type: `ts3prc.do&`. See Section 1.6.1, page 56 and UNIX documentation on job control in *csh(1)* for more information.

binary is a base 2 number system. It is sometimes referred to when discussing how data is encoded inside of the computer.

bit is a binary digit in a base 2 number system generally denoted by 0 or 1.

CCD camera is a Charge Coupled Device which is part of optical scanner camera. At NCI, this camera is connected to a SUN computer for scanning gels. It has a useful dynamic range of about 0 to 2.3 OD. Programs **camera** and **getacc** access the camera.

central core image is the **sg2gii** intermediate image computed from the smoothed second-derivative of the original gel image. Pixels which are part of the central portion of spots are indicated with 1's while the rest of the image is 0. See *propagated central core* for later stages of processing.

Cgel′ is the estimate of the canonical gel produced by the **cgelp2** program from the average of a set of replicate gels. See [LemP82a] for a discussion on the canonical gel.

composite gel (CGL) is a database consisting of Rspot sets derived from spots from a set of gels. The PCG DB is a disk based CGL used by the **cgelp2** program.

command is a string of characters typed by the user, usually terminated with a <CR> (return key), which directs UNIX or the GELLAB program being run to perform some action.

composite pair (CP) is a spot pairing label assigned to two composite spots from different gels as defined by the **cmpgl2** program with the `-cgfile` switch. A composite spot and its features are defined as the sum of a small set of spots which are treated as one spot.

computing window is a rectangular region interactively defined on the gel which specifies that part of the gel image which is to be analyzed by the **sg2gii** spot segmenter program ([LemP81a], [LemP83a]). It is defined for each gel by **getacc** or **calndw** and saved in that gel's accession file entry by those programs. The **Xpix** program also has a computing window but it is not related to that used in GELLAB-II.

clustering operations are operations performed on subsets of gels and/or SRL subsets can result in the definition of additional relationships between gels or Rspot sets. For example, spots can be clustered into coordinately regulated spots as a function of protein expression profiles; gels can be clustered into groups which can be predicted by marker proteins.

CPU time is the total amount of time the computer's central processing unit has spent on the user's program(s) processing data. This time will be less than the RUN time since the computer is doing other things than just processing your data.

C language is a computer programming language associated with UNIX. GELLAB-II is written in C.

C shell (csh) is one of several shell programs available in interface between the underlying UNIX system and the user. *csh(1)* is generally associated with UNIX systems derived from Berkeley UNIX.

D' **density** is a measure of spot quantitation being integrated optical density (OD units) for the spot corrected for gel background in the region of the spot. It is computed as $D_i^{'} = D_i - A_i D_i^{background}$ where D_i is the uncorrected spot

density for spot i, A_i is the spot Area, $D_i^{background}$ is the gel background density in the region of spot i.

derived gel image is a synthetic image such as an Rmap or mosaic image which encodes some original gel image data overlayed with additional gel database information.

directory is a list of the user's files. A directory can contain other sub directories in an inverted tree shaped hierarchy of directories. To see what the current files are. Eg. type: '`ls`' or '`ls gellab`' to see what directories are in `gellab`, or '`ls gellab/gsf`' to see what GSF files exist in particular sub directories, etc.

disk space is the amount of disk file storage used by a particular directory tree. Eg. Type: `du gellab` or `du gellab/ppx` or `du gellab/pcg` etc., to see what each of these directories or subdirectories has used. Type: `df .` to find out how much space is left on the entire disk. Disk space is one of those things of which you never have enough. See the UNIX *compress(1), uncompress(1)* utilities which can help you save disk space.

display is the user's terminal which may have the ability to perform line drawing graphics. It may also be a window in the X-window system. If the user's terminal can not display graphics then no interactive plotting or image display may be performed. However, other GELLAB functions may be invoked of a tabular or numerical nature.

environment variable is the name of a symbol the user can define, such as GELLABMANAGER, which programs can access when needed. See UNIX references for more information.

EP spot is an instance of an extrapolated spot for a particular gel in the PCG DB. It is created either of two ways. If the spot was not found in the Rgel during <`CMD`> `CREATE/ERSPOT`, then it is created for the Rgel. Later, it may be extrapolated for any gel missing from the PCG DB - i.e. there is no SP, PP, US, AP spot for that gel. the <`CMD`> `EXTRAPOLATE` command is used for this latter form of extrapolation in **cgelp2.** ([LemP82a], [LemP83a])

eRspot is an extrapolated Rspot in the PCG DB which was created from US (Unresolved Spots) spots not found in the Rgel. This operation is performed by **cgelp2.** ([LemP82a], [LemP83a])

experimental class of gels is a subset of gels in the working set of gels which are all of the same experimental class. Eg. lymphoid vs. myeloid leukemia gels. The same gels may simultaneously belong to other classes: eg. acute vs chronic Leukemia.

extended Rspot (eRspot) is a Rspot for any spot found in gels *other* than the Rgel and *not* found in the Rgel. Therefore the set of all Rspots (extended and unextended) includes *all* spots found on *any* gel.

extrapolated pair (EP) is a spot label given to a synthetic spot with density defined to be zero mapped to the position where the spot should be in that gel. ([LemP82a], [LemP83a])

file is the name of a particular set of sequential data stored on the computer's disk.

file extension is a sequence of 0 to 3 characters used in defining the file name suffix. The following file extensions are used in GELLAB-II:

.ann (not currently used) associated annotation DB file,

.cal the **cgelp2** (pIe,MW) calibration file,

.ccl Concise Control List file of accession numbers,

.cgl the **cgelp2** printable Rspot sets data from PCG DB,

.dgm the **dendrogram** cluster analysis tabular data,

.drm the **dwrmap** sorted (by density) spot data file,

.do executable UNIX script file,

.gdo executable **cgelp2** input<CMD> script file,

.gcf the **cmpgl2** gel comparison file,

.gsf the **sg2gii** gel segmentation file,

.hex (obsolete) portable ASCII hex coded image file,

.id accession database file,

.inq the **cgelp2** inquire search/table results file,

.lm the **landmark** LandMark set database file,

.map foreign spot map and annotation DB file,

.pcg the **cgelp2** Paged Composite Database file,

.ps PostScript graphics file,

.ppx Portable PiXture image file,

.rc startup file. i.e. GELLAB-II state file `gel.rc`,

.sas the **cgelp2** SAS generated intermediate data file,

.sps the **cgelp2** SPSS generated intermediate data file,

.srl the **cgelp2** SRL subsets intermediate data file,

.tbl the **cgelp2** generated table file,

.ugf Universal Graphics File which can be plotted.

All files are ASCII format except for *.pcg, .ppx* and *.ugf* which are binary.

file name is the name of a particular disk file. It is sometimes used to denote the whole file name (*prefix-file-name . file-extension name*) or just the prefix-file-name. E.g. `p20250.gsf` and `p20250` respectively. A full path file name has the UNIX path included, eg. `/home/joeUser/gellab/p20250.gsf`.

flicker comparison of two gels is the repeated alternate presentation in time by either optical or video display technique of two images. One of which is moved relative to the other in order to line up morphologically similar regions. In GELLAB-I, flickering gel images was done using the RTPP hardware and the FLICKER program [LemP79a]. In GELLAB-II, flickering is implemented using compare-mode in the **Xpix** program which is invoked for two gels by the **accppx** program. [Alternatively, the **disp11** program can be used to flicker compare two whole gel images.] **Xpix** can flicker two zoomed gel images where the regions being compared are interactively determined using the mouse position. The **landmark** program also uses flickering as an aid in performing landmark definition.

gamma in the context of photography and image processing generally refers to the characteristic response shown by a detector. The detector can be a stain, photographic film or camera. Each has its own characteristic gamma function which are often non-linear. The gamma function may be thought of as the exposure (output) as a function of light intensity (input). Therefore when using several of these mediums, such as when scanning an autoradiograph, the effective gamma response function is the convolution of the intermediate gamma response functions.

gel as used here refers to a 2D PAGE gel produced using either autoradiography, silver stain or some other stain to detect protein spots. The phosphor imaging direct particle counting scanners are also used to detect proteins. We often refer to a gel in this book (and associated papers) when we releally mean the detected gel image obtained by one of the above methods. In any case we are really specifying the quantitative information contained in the gel.

gel comparison file (GCF) is a spot pair list data file produced by pairing spots between the Rgel and another gel. It contains the features found in the GSF spot list files. These ASCII files have a *.gcf* file extension and are created by **cmpgl2.**

gel segmentation file (GSF) is a spot list data file containing (x,y,density,etc.) information about all spots in a segmented gel image. These ASCII files have a *.gsf* file extension and are created by **sg2gii.**

gel state file (gel.rc) is a disk file in the users home directory. It is used to pass *state* information concerning various directories, files and parameters required by different GELLAB programs between these programs. Type **pgelrc** to find out the status of an existing file or to be prompted for information to create the file if it does not exist.

gel subset is a subset of the gels used to construct a **cgelp2** PCG database. It is defined for the users convenience in manipulating lists of gels (see **cgelp2** <CMD>'s `SET GEL SUBSETS, SET CLASSES, SET WORKING SET` commands). A gel subset is defined by one of several criteria: gels in the same experimental class, gels in the working set, explicit definition, set operations on other sets of gels, etc. Although there many be gels in a subset which are not in the working set of gels, only those which are in *both* sets are visible during an analysis.

graphscale image is a pseudo color mapping used by several of the GELLAB programs to generate derived images. The **Xpix** program is able to read and display these images. The mapping combines an 8-bit grayscale image with up to 10 colors of line graphics overlays. Grayscale values in the range of [0:255] (i.e white to black) are mapped to [0:245] with gray values > 245 being set to 245. Then gray scale 246 is RED, 247 is ORANGE, etc.

group in UNIX, a way of further partitioning the file system so that members of the same group can access files with the same group "protection" while others may not. GELLAB-II has two groups: `gellab` for use only by the ~`gelmgr` account, and `gelusr` for use by all GELLAB-II user accounts.

image is the computer digital representation of a stained gel or its autoradiograph after it is scanned by and/or processed by various methods. Images in GELLAB have *at least* a 512x512 matrix of sample points (pixel or picture element). These are stored in Portable PiXture *.ppx* files. The Datacopy 612F CCD camera has a maximum resolution of over 2300x1700 8-bit pixels. However, the scan size of 1024x1024 should be adequate for most gels. The GELLAB software will be able to handle an image up to 4096x4096 pixels and the Datacopy model 620 camera is available (as are other scanners) with that resolution. Each pixel representing the density by a value 0 being white and 255 being the maximum density obtained with the scanner. Using auxiliary neutral density wedge calibration information in the accession file, these pixel values can be mapped to OD (optical density) for subsequent integration where required. Images scanned using a linear OD scale (such as the Optronics scanner) do not require this calibration and the `-OD:`*upperODvalue* switch is specified to the **sg2gii** program at the time the gels are segmented.

intersection is a set operation which selects set elements common to two sets. E.g. the set of spots found to be statistically significantly different between two different experimental classes under two different statistical tests.

inquire file is a logging file which lists the results of performing a search or SRL post-processing operations.

job an independent execution of UNIX commands as a separate user process. For example, execution of a GELLAB script produced by **makjob** or **getacc**.

kernel system or program is the primary program or programs supplied by the computer manufacturer (in this case the UNIX software system) which facilitates the easy operation of the computer hardware. It consists of a set of programs and commands which the user invokes in the process of using the computer. The user does not need to know how this works - merely some of the commands to make it do what they want.

landmark list (LM) is a list of landmark spots (typically 10 to 50) defined between the Rgel and another gel. A LMS DB file has a *.lm* file extension and is used by all programs needed local gel region alignment including: **cmpgl2, mapgel** etc.

landmark set of spots (LMS) is a set of segmented spots which are assigned to a particular landmark which it is closest. This assignment is performed by the **cmpgl2** program.

landmark spot is a spot selected and identified to be the same in two gels. It is usually defined between the Rgel and another gel.

landmarking is the process of generating a landmark set between the Rgel and another gel. The **landmark** program is used to interactively define this landmark set. The **autopair** program will combine the landmark function with the gel spot pairing functions of **cmpgl2.**

list of SRL subsets is a list of the *numbers* of selected SRL subsets. GELLAB-II has set operations which can be performed across a list of SRL subsets in addition to just between two. This is denoted by `/ListOfSRLs` or `LOS` in the `SET SRL SUBSETS` operation in **cgelp2**.

logging into the computer gains one access to a particular user account. You are required to be logged in, in order to run programs. In the process of logging into the computer you need to know your "login" name and your password.

logging off of the computer removes ones current (i.e. logged in) access to the computer. Typing `CONTROL/D` or the word `exit` to the UNIX shell will log you off.

mosaic image or plot of a Rspot for a set of gels is a composite image or graphic formed from panels from each gel arranged in a regular checkerboard pattern. The panels are taken from a subregion of each gel surrounding a particular Rspot. The synthetic images `wirrrr.ppx` generated by the `mosaic` program have the following notation. A `w` picture file prefix, the `i'th` image of the set of mosaics for spot `rrrr`. Eg. For Rspot 123 with 23 gels we would generate `w00123.ppx` and `w10123.ppx`. The first image having 16 panels and the second 7 (i.e. $23 - 16$). Mosaic images are create by the **mosaic** program using SPSS files and mosaic plots by using the `MOSAIC <CMD>` in **cgelp2**.

neutral density step wedge (for OD calibration of gel) is a narrow wedge of photographic film consisting of increasingly darker rectangular panels which have been calibrated in OD. It in turn is used to calibrate each gel image in terms of this OD in order to remove variation from camera, setup or light source. A ND wedge calibration is a set of (10 to 15) 2-tuples {(ND-wedge-OD-values, corresponding-gray-values)} for all usable steps in the wedge. This is used to generate a piecewise linear interpolation of gray value to OD. This conversion is performed during spot segmentation with **sg2gii** when making the GSF spot lists and is not required again.

normalizing gel density is a process performed in **cgelp2** for calibrating the density scale of each gel in order that density measurements performed in one gel may be compared with measurements from another gel. A calibration should map the spot concentration (integrated density) in one gel to a number such that a calibrated replicate gel should have a very similar value for the same spot. The `SET DENSITY MODE` command in **cgelp2** changes the normalization method. Before you can change to some normalizations, you must compute them. This is done using `SET RATIO NORMALIZATION` and `SET LEAST SQUARES NORMALIZATION`.

OMNIGRAPH is the name of the DECsystem-10 line graphics software written by Bob Sproull and obtainable from DCRT, NIH [DCRT79]. It was used throughout GELLAB-I for line graphics to generate interactive displays or plot files (*.plx*). Plot files could then be later redisplayed or plotted with **plotn** (a derivative of the OMNIGRAPH PLOTX program). GELLAB-II emulates much of the OMNIGRAPH functionality using the `smdisp.c` package but for a restricted set of devices: Tektronix 4010, X-windows, PostScript and *.ppx* files. The 36-bit/word .plx files are replaced with the more portable 32-bit/word *.ugf* files using the SMDISP system in GELLAB-II.

optical density (OD) is a unit of measurement of light absorption. It is the log of (input light intensity divided by output light intensity) where the output light intensity is measured on the other side of the gel from the transmitted

light source. A calibrated neutral density (ND) photographic step wedge is used to calibrate camera image data from grayscale to OD units. A calibration consists of finding corresponding values (grayscale,OD) for panels of the ND step wedge. The **sg2gii** and other programs can then extrapolate a piecewise linear calibration curve to map gray scale image pixel values to OD pixel values. The latter OD values can then be summed to compute integrated density over a region. The calibration is done in **getacc** for images being scanned during data acquisition. It can also be done using **calndw** for previously scanned images which contain a ND wedge strip. In both cases the accession file entry is updated with this calibration (as well as the computing window active region of the gel image).

PAGE is PolyAcrylamide Gel Electrophoresis.

paging is a computer technique whereby data from a file is brought in and out of the fast computer memory so that it appears to a program as if the memory were the size of the file. The **cgelp2** composite gel database (PCG DB) is paged as are *.ppx* images in all programs doing image I/O using the GELLAB library.

Paged Composite Gel (PCG) DB is a paged composite gel spot DB file constructed by merging a set of GCF files. This binary file is constructed by **cgelp2** which can checkpoint it and reload it at a later time.

pairing label Each spot from some gel G_j paired with corresponding spots from the Rgel gel gets a pairing label. These are Sure-Pair (SP), Possible-Pair (PP), Ambiguous-Pair (AP), Unresolved-Spot (US), Extrapolated-Pair (EP), and Garbage-Spot (GS). The SP, PP, AP and US are assigned during spot pairing with **cmpgl2** or **autopair**. The EP and GS are assigned in **cgelp2** during database analysis.

path (of a directory of files) specifies where in the directory tree in which files may reside. The `gel.rc` file in each experiment project directory defines paths for the rest of the GELLAB-II data files.

PCG DB is Paged Composite Gel DataBase.

plot is the drawing of a graphic display Universal Graphics File (*.ugf*) on paper using a plot program such as **plotn.** See OMNIGRAPH.

plot file is a Universal Graphics File (*.ugf*) disk file produced by a GELLAB program which may be plotted at a later time.

Portable PiXture File (PPX) is a disk file (with Portable PiXture File *(.ppx)* extension) used to store a gel image or derived image. The initial 512 bytes

of the file define a descriptive header which includes things like image size, #bits/pixel, white/black, OD range etc. A C struct definition of this region is given in `ppxfmt.h` for other programs desiring to use *.ppx* image files. The following one character file name prefixes are used by particular programs or functions in GELLAB (note that other letters may be used when assigning new gel image names (c.f. **getacc** program):

a,b original gel images,

c propagated central core image in **sg2gii**,

f flipped image (pIe or MW reversed from original) in **markgel**,

g GraphScale pseudocolor image generated by various programs,book.tex

j averaged image in **sg2gii**,

k maagnitude of 2nd derivataive image in **sg2gii**,

l Rmap landmark image in **cmpgl2**,

m Rmap image from SPSS file in **markgel,**

n notch filtered image,

s standard 512x512 PPX size copy of original gel images,

t change vector Rgel labeled paired-spot image in **autopair**,

u marked Rgel labeled paired-spot image in **cmpgl2**&**autopair**,

v marked gel paired-spot image in **cmpgl2**&**autopair**,

w mosaic image in **mosaic**,

y (original less the segmented spot) image in **sg2gii**.

z segmented image in **sg2gii.**

possible pair (PP) is a spot pairing label assigned to two spots from different gels such that it meets less strict pairing criteria as defined by the **cmpgl2** program.

prefilter is a primary statistical and logical test applied to a Rspot set prior to it's being used in the specified operation. It checks: a) if the gels in the Rspot set are in the Working Set of gels; b) if the gels are in the desired experimental classes; c) if the Rspot sets are in the desired (pIe,MW) sub-region; d) if individual spots meet the spot-pairing label; e) if the Rspot set meets the current statistical limits for Rspot set features. In addition, you can prefilter an operation by the spots in a particular SRL subset n using, `/SRL:`n.

prefix file name is a sequence of 1 to 6 characters (numbers, letters or both) used in defining a generated file name. Generated file names are used in many of the GELLAB-II programs. When generating a file name in the **cgelp2** database program, if no name is specified, sequential numeric names `000001`, `000002`, etc. are used.

process an independent execution of UNIX command(s) or program. The UNIX command `ps` lists your active process while `ps -aux` lists all user processes.

program is a sequence of instructions in computer machine language used to perform a particular function. For example gel spot segmentation, gel spot comparison, etc.

project is the set of original and derived data for a set of gels in one experiment. Several experiments can be put into one project and merged into a single database. They could alternatively be put into separate project directories.

project prefix is a 3 character alphabetic prefix to all **makjob** and **getacc** batch scripts for GELLAB. These scripts all deal with the same database.

propagated central core image is the **sg2gii** intermediate image computed from the central-core gel image. Pixels which are part of the central core image are re-labeled with values in the range of [2:99], propagated pixels for the same spot have 100 added. Background pixels are 0, deleted spots are 254 and isolated pixels are 255. See *central core*.

raster defines a way of traversing an image. Typically one starts in the upper left-hand corner and go line by line (from left to right) to the bottom of the image. Images in GELLAB-II are processed in raster order.

raw gel image is the original computer readable scanned image of a gel which can be calibrated in optical density units or counts per minute.

Rgel is the representative gel used to geometrically pair all of the other gels in a set of data. This is the common tie point used to merge a set of GCFs in **cgelp2**. All spots in an experiment need not be present in the gel for it to be used as the Rgel since they will be extrapolated into the Rgel when the database is built.

Rmap image or plot of a set of spots for a given gel. If it is an image, the selected spots are denoted by '+' with optional superimposed white Rspot labels. The labels can be Rspot numbers or landmark set alphabetic names if the spot is a landmark spot. Rmaps are created by **markgel** for images and by using the `RMAP <CMD>` for **cgelp2** plots.

Real Time Picture Processor (RTPP) was an interactive gray scale image processor at the IPS in NCI (1974-1987) which had 16 256x256x8-bit pictures which could be reconfigured as 512x512 etc, a TV video frame grabber, a DMA channel between the picture memory and a DEC2020, and display as well as a PDP8e controller. It was controlled by the BMON2 program [LemP80a] which ran on the PDP8e. It was used for distributed processing in data acquisition in GELLAB-I as well as flicker comparison and interactive landmarking of gels. This functionality has been replace by the **getacc**, **camera** and **Xpix** programs.

replicate gels are a set of gels created from split samples, parallel tissue cultures, or samples of the same experimental class which are used to improve the statistics when doing an analysis. They are useful for finding the variance in gel sample reproducibility.

Rspot number is a number which uniquely defines corresponding spots in different gels of a particular composite gel (PCG) DB. This Rspot number is an arbitrary sequential number which is assigned in the database. Corresponding spots in any gel can be retrieved by knowing the Rspot number and the gel accession number.

Rspot set is a set of spots from different gels in a **cgelp2** PCG database consisting of a set of corresponding spots for all gels currently under consideration. So more spots from different gels will be included in more gels are added to the working set of gels. (See spot).

RUN time is the total time-of-day clock-time (wall-time) amount of time the computer's central processing unit was running when it processed the users program(s). As UNIX can process multiple programs at the same time, not all of the time is spent on the user's program(s) processing data. The CPU/RUN times gives the percent of time the system was working for you!

SAIL is the programming language used in the construction of all GELLAB-I programs. It stands for Stanford Artificial Intelligence Language (Stanford University). It is a ALGOL-like language with many additional language features. GELLAB-I was translated from SAIL to C using the PSAIL compiler, so that GELLAB-II is written in C.

Search Results List (SRL) is a list of Rspots found or denoted in the **cgelp2** program. This list of spots may be used to generate various types of output. It may be saved as a SRL subset and subsequent operations performed on these subsets.

SRL subset (SRL:n) is a SRL saved as a separate subset n or Rspot numbers with its own user-assigned *annotation title*. The spot subset may be derived from a

variety of DB search and/or post-search operations. The SRL subset may be referred to by either its assigned set number (eg. SRL:5) or by its title. The SRL subset may also be retrieved by specifying relational expressions of key words found in the title - or by querying which SRL subsets contain which spots. SRL subsets are created and manipulated using the `SET SRL SUBSETS` command in **cgelp2**.

SRL subsets file is a list of Search Results Lists subsets, each of which consists of a title and list of Rspot numbers identified in the PCG DB analysis.

SRL subset operations include (Union, Intersection, Difference) They are considered a *Post-filter* operation performed using the results of previous statistical tests or *SRL subset operations* resulting in additional SRL subsets (defined above). It is used to compare the members of various SRLs.

segmentation of spots in a gel image is the extraction of (x,y) centroid, background area, spot integrated density as well as other features of spots. These results are tabulated in an ASCII Gel Segmentation File (GSF) of these features and an image of the extracted spots. The program **sg2gii** is used to perform spot segmentation.

set is a group of objects each of which appears only once in the group. Sets of spots or gels are used in **cgelp2**. Sets may be manipulated to generate new sets using the union, intersection and set difference operations which are available in GELLABII.

shell is a program which can run other programs. The shell is an integral part of UNIX. You interact with UNIX through the shell. There are several shells *csh(1), sh(1), ksh(1)*, etc. [KocS88]. Many of the commands from different shells are similar but some are not. We use *csh* [AndG68] in this book.

spot generally refers to a particular polypeptide spot in a particular gel. (See Rspot)

spot pairing is the process of matching spots between an Rgel and another gel. The program **cmpgl2** used to perform spot pairing. The results of which is the Gel Comparison File (GCF).

spot pairing label is a reliability measure describing the pairing between the Rgel and another gel (SP-sure pair, PP-possible pair, US-unresolved spot, AP-ambiguous pair, EP-extrapolated pair, CP-composite spot pair).

SPSS (SAS) data file is a list of Rspot set data for selected spots in the PCG database. Alternatively, it can consist of all protein concentration data for all working set gels by all spots visible to the prefilter with missing spots

defaulting to 0. This numeric data file is suitable for input to commercially available statistical analysis programs such as SPSS or SAS as well as other data analysis programs.

statistical test is a subsequent secondary Rspot set filter applied to all successfully (primary) *prefiltered* Rspot sets to produce a SRL. These univariate tests ([NatM66], [SneG80]) include parametric tests: such as a t-Test (both standard and Behrens-Fisher based on F-statistic for equal/unequal variance, F-test, etc.; and non-parametric tests such as the Wilcoxon-Mann-Whitney, Kruska-Wallis, etc. The data in each Rspot set is tested for significant differences using one of the these tests comparing two or more classes of gels eg. (CONTROL vs. EXPERIMENTAL), or DOSE, TIME or TITRATION.

sure pair (SP) is a spot pairing label assigned to two spots from different gels such that it meets strict pairing criteria as defined by the **cmpgl2** program.

switch is a software command modifier construction which is generally part of a command to alter the performance of that command. UNIX level switches are generally proceeded by a "-". An exception is the **cgelp2** program where <CMD> interpreter level switches are generally proceeded by a "/".

table file is a logging file which lists the results of performing statistical operations resulting in various types of global summary statistics tables (eg. correlation, expression-profile tables, etc).

terminal is the device (either a video TV like display or a printer device) with a typewriter like keyboard used for communicating with a computer.

union is a set operation combining elements which occur in either of two sets into a new set. Eg. the set of Spots found using either of two tests in **cgelp2.**

UNIX is the software operating system under which the user can log in/out of the computer and run system as well as their own programs.[1]

unresolved spot (US) is a spot pairing label assigned to one spot which is found in one of two different gels suring spot pairing such that it did *not* meet the minimum pairing criteria as defined by the **cmpgl2** program.

window system is an computer graphics program which allows the user to interactively put gray scale, color and plot information in local regions called windows. Many windows can be active at the same time - each running a different UNIX process. The user interacts with a window using the "mouse". The X-Window System is one such window system and is used with GELLAB-II.

[1] UNIX is a Trademark of ATT.

word in the context of computers denotes the basic unit of storage in the computer. It is generally described as being n-bits long. For the SUN and VAX computers, n=32. For the CRAY, n=64.

working set of gels is a subset of gels selected by the user from the database set of gels in the **cgelp2** program. Only gels in the working set of gels are considered for **cgelp2** operations. The investigator can redefine the working set of gels at any time.

X-windows is a specific window system developed at MIT project Athena [SchR86]. It is the window system used with GELLAB-II.

Appendix B

Some useful GELLAB-II UNIX shell scripts & aliases

The following C shell script (*csh(1)*) programs were found to be useful when using the GELLAB-II system on our computer under UNIX at IPS, NCI. They are invoked by the name of the program 'script' or 'alias' (such as `ppx` without the `.do`) followed by arguments (if any) separated by spaces between arguments. An *alias* is a shorthand form a UNIX shell command. Aliases can be added to your `.cshrc` file so they are installed when you log in. The aliases mentioned here are already included in the distributed `.cshrc` file.

EXAMPLE 1. `ppx` - display a particular gel or derived image(s) image in up to two **Xpix** window(s).

display gel images

```
     alias ppx 'accppx \!*'              # display originals or .ppx'

USE:
     ppx 324.1
or
     ppx 324
or
     ppx 324.1 369.1
or
     ppx 324.1 w00064.ppx
or
     ppx 324 w00064
or
     ppx mcrew.ppx boys.ppx
```

EXAMPLE 2. `ppxc` - display the original gel and its propagated central core image if it was generated by `-sg2gii -ctlImages`. Display them for a particular gel image in the **Xpix** windows.

```
alias ppxc 'accppx \!* \!* -p2:c'

USE:  pixc 324.1
```

EXAMPLE 3. `ppxg` - display the graphScale image(s) if they were generated by various GELLAB-II programs. Display it for a particular gel image in the **Xpix** window.

```
alias ppxg 'accppx \!* -graph -pr:g'

USE:  ppxg 324.1
```

EXAMPLE 4. `ppxj` - display the original gel and its Gaussian smoothed image if it was generated by `-sg2gii -ctlImages`. Display them for a particular gel image in the **Xpix** windows.

```
alias ppxj 'accppx \!* \!* -p2:n'        # display orig & background

USE:  ppxj 324.1
```

EXAMPLE 5. `ppxk` - display the original gel and its Laplacian magnitude image if it was generated by `-sg2gii -ctlImages`. Display them for a particular gel image in the **Xpix** windows.

```
alias ppxk 'accppx \!* \!* -p2:n'        # display orig & background

USE:  ppxk 324.1
```

EXAMPLE 6. `ppxl` - display the original gel and its landmark image if it was generated by `cmpgl2 -onlyMarkLMS`. Display them for a particular gel image in the **Xpix** windows.

```
alias ppxl 'accppx \!* \!* -p2:l'

USE:  ppxl 324.1
```

EXAMPLE 7. `ppxm` - display the original gel and last computed Rmap if it was generated by **markgel** for that gel. Display it for a particular gel image in the **Xpix** windows.

```
     alias ppxm 'accppx \!* \!* -p2:m'

USE:  ppxm 324.1
```

EXAMPLE 8. `ppxn` - display the original gel and its background density image if it was generated by `-sg2gii -ctlImages`. Display it for a particular gel image in the **Xpix** windows.

```
     alias ppxn 'accppx \!* \!* -p2:n'        # display orig & background

USE:  ppxn 324.1
```

EXAMPLE 9. `ppxv` - display the pairing label gels for the G2 gel and the associated Rgel if they were generated by `-cmpgl2 -markPairings`. Display them in the **Xpix** windows.

```
     alias ppxv 'accppx \!* \!* -p1:u -p2:v'

USE:  ppxv 324.1
```

EXAMPLE 10. `ppxw` - display the mosaic image(s) if they were generated by **mosaic**. Display them in the **Xpix** window(s).

```
     alias ppxw 'accppx \!*'

USE:  ppxw w00123
or
      ppxw w00123 w10123
```

EXAMPLE 11. `ppxy` - display the original and original less the segmented image if it was generated `-sg2gii -restOfImage`. Display them in the **Xpix** window.

```
     alias ppxy 'accppx \!* \!* -p2:y'

USE:  ppxy 324.1
```

EXAMPLE 12. `ppxz` - display the original and segmented image if it was generated `-sg2gii`. Display them in the **Xpix** window.

```
alias ppxz 'accppx \!* \!* -p2:z'

USE:  ppxy 324.1
```

EXAMPLE 13. `ppxG` - display the two different derived images of the same gel specified by the image prefixes. These are displayed them for a particular gel image in the **Xpix** windows.

```
alias ppxG 'accppx \!1 \!1 -p1:\!2 -p2:\!3'

USE:  ppxG 324.1 c z
```

plot UGF

EXAMPLE 14. `plotx` - plot a particular UGF plot file on the PostScript laser printer.

```
alias plotx 'plotn -display:laser'

USE:
     plotx 000002
or
     plotx 000002.ugf
```

make a Rmap

EXAMPLE 15. `mark` - create Rmap with white labels. The default for **markgel** is to generate color *graphScale* labels which do not photograph well with black and white film.

```
alias mark 'markgel \!* -nographscale -large -Xpix'

USE:
     mark 324.1 ts3s02.sps
or
     mark 324.1 ts3s02.sps -Zoom:2x
```

display all mosaics

EXAMPLE 16. `ppxmosaic` - sequentially display all mosaic files with Xpix. It finds all w*.ppx* files in gellab/aux and then displays them.

```
#!/bin/csh
# Script ppxmosaic - sequentially display all mosaic files with Xpix.
# It finds all w*.ppx* files in gellab/aux and then displays them.
#
#  E.g.
#         ppxmosaic
#
# September, 1989
echo "The following mosaics are available:"
cd gellab/aux
set d = `ls w*.ppx*`
cd ../..
echo " " $d
echo ""
foreach x ($d)
    echo "MOSAIC:    $x"
    accppx -graphscale $x
end
echo "----End of displaying mosaics---"

USE:  ppxmosaic
```

EXAMPLE 17. `mapcar` - apply a program to a list of accession numbers one by one. Each accession number is repeated once in the mapped command.

```
#!/bin/csh
# Script mapcar - apply function with switches listed from
# arg #2 to the end of the UNIX command line to the list
# referred to in arg #1 of the command line.
#
#         mapcar CCL_file executable_sequence
# e.g.
#         mapcar ts3 accppx -silent
#
#
# will apply 'accppx -silent'  to all of the accession
# numbers found in 'ts3.ccl'.
# E.g. if an element of file ts3.ccl was 324.1, then it would expand to:
#         accppx -silent 324.1
#
# September, 1989
if($#argv <2) then
 echo" you need two args:"
 echo "         mapcar CCL_file executable_sequence "
 echo " e.g. "
 echo "         mapcar ts3 accppx -silent "
 exit 1
endif
foreach x (`cat $1.ccl`)
    echo "DOING:   $argv[2-$#argv] $x"
```

```
            $argv[2-$#argv] $x
        end

USE:  mapcar ts3 accppx
or
      mapcar ts3 accppx -prefix:z
or
      mapcar ts3 sg2gii
```

EXAMPLE 18. `mapcar2` - apply a program to a list of accession numbers one by one. Each accession number is repeated once in the mapped command.

```
#!/bin/csh
# Script mapcar2 - apply function with switches listed from
# arg #2 to the end of the UNIX command line to the list
# referred to in arg #1 of the command line.
#
#         mapcar2 CCL_file executable_sequence
# e.g.
#         mapcar2 ts3 accppx -silent
#
#
# will apply 'accppx -silent -p2:z'  to all of the accession
# numbers found in 'ts3.ccl'.
# E.g. if an element of file ts3.ccl was 324.1, then it would expand to:
#         accppx -silent -p2:z 324.1 324.1
#
# September, 1989
if($#argv <2) then
 echo" you need two args:"
 echo "         mapcar2 CCL_file executable_sequence "
 echo " e.g. "
 echo "         mapcar2 ts3 accppx -silent "
 exit 1
endif
foreach x (`cat $1.ccl`)
    echo "DOING:   $argv[2-$#argv] $x $x"
    $argv[2-$#argv] $x
 end

USE:  mapcar2 ts3 accppx -prefix:z
or
      mapcar2 ts3 sg2gii -p1:c -p2:z
```

EXAMPLE 19. `ppxall` - display a sequence of original gel images specified by a Concise Control List (CCL) file. Display them in the **Xpix window**. Exit **Xpix** to continue to the next gel in the list. See `mapcar` for more details.

display all images

```
alias ppxall 'mapcar \!* accppx'

USE:  ppxall ts3
```

EXAMPLE 20. `ppxallc` - display a sequence of original gel and propagated central core imagesspecified by a Concise Control List (CCL) file. Display them in the **Xpix window**. Exit **Xpix** to continue to the next gel in the list. See `mapcar2` for more details.

```
alias ppxallc 'mapcar2 \!* accppx -p2:c'

USE:  ppxallc ts3
```

EXAMPLE 21. `ppxallg` - display a sequence of original gel and the graphScale images specified by a Concise Control List (CCL) file. Display them in the **Xpix window**. Exit **Xpix** to continue to the next gel in the list. See `mapcar` for more details.

```
alias ppxallg 'mapcar2 \!* accppx -p2:g'

USE:  ppxallg ts3
```

EXAMPLE 22. `ppxallm` - display a sequence of gel Rmap images specified by a Concise Control List (CCL) file. Display them in the **Xpix window**. Exit **Xpix** to continue to the next gel in the list. See `mapcar` for more details.

```
alias ppxallm 'mapcar \!* accppx -p1:m'

USE:  ppxallm ts3
```

EXAMPLE 23. `ppxalln` - display a sequence of original gel and notch filter background specified by a Concise Control List (CCL) file. Display them in the **Xpix window**. Exit **Xpix** to continue to the next gel in the list. See `mapcar` for more details.

```
alias ppxalln 'mapcar2 \!* accppx -p2:n'

USE:  ppxalln ts3
```

EXAMPLE 24. `ppxallcz` - display a sequence of propagated cental core and segmented gel images specified by a Concise Control List (CCL) file. Display them in the **Xpix window**. Exit **Xpix** to continue to the next gel in the list. See `mapcar` for more details.

```
alias ppxallcz 'mapcar2 \!* accppx -p1:z -p2:c'

USE:  ppxallcz ts3
```

Appendix C

Some useful UNIX commands

It is not necessary for you to be an expert UNIX "hacker" to run GELLAB-II. However, learning a little UNIX certainly can't hurt. You could read a number of tomes on using UNIX (which *is* suggested you do - ([WanP88], [AndG86], [KocS88]) are minimally painful) or learn a little here by example. Obviously, you will not be a UNIX expert by reading these few pages - but they will at least give you an orientation to UNIX especially if you have used another computer system. No examples of text editors are offered since they are in a "different ball park" in terms of complexity and difficulty to learn. GNU EMACS [StaR86] is our editor of choice, although any editor will do. EMACS, which edits regular ASCII files, is not trivial to learn but is well worth the effort because of its extensive capabilities.

the UNIX prompt

When you log into UNIX, you will get the shell prompt *number*%. [This command history reference *number* is incremented by the C shell for every command you type. We will not discuss UNIX C shell history here (see any of the above references).] At this point, you can run programs by typing their name. In addition, there are a few special characters which you may wish to know about.

- CONTROL/C - kill current process get back to UNIX shell level.
- CONTROL/D - kill current process and log off of UNIX.
- CONTROL/R - retype a cleaned up line.
- CONTROL/U - erase line typed so far.
- CONTROL/Z - stop process (put it in UNIX background with `bg` C shell cmd).

- DELETE or BACKSPACE - erase typed character to left of cursor.

basic UNIX commands

Some of the standard UNIX shell level (i.e. `control/C` or '.' level) commands or programs are useful when analyzing gels. These include: `cat, lpr, ls, pwd, cd, more, script, rm`, etc. Information can be obtained on-line from your UNIX system for any command *cmd* by typing `man` *cmd*. The following UNIX commands are included:

TABLE C.1. UNIX commands - see following subsection references for examples.

`at`	start batch script at specified time. (cf. C.13)
`atq`	list entries in batch job queue. (cf. C.13)
`atrm`	kill entry in batch job queue. (cf. C.13)
`cat`	concatenate file(s) to standard output. (cf. C.4)
`cd`	change working directory. (cf. C.9)
`chgrp`	change file group membership. (cf. C.10)
`chmod`	change file protection(s). (cf. C.10)
`compress`	compress file(s) to take less space. (cf. C.5)
`cp`	copy file(s). (cf. C.8)
`exit`	log user off of UNIX. (cf. C.1)
`login`	log user into UNIX. (cf. C.1)
`grep`	search for string in file(s). (cf. C.11)
`lpr`	print file on local line printer. (cf. C.4)
`lpq`	show entries in printer queue. (cf. C.4)
`lprm`	kill entry in printer queue. (cf. C.4)
`ls`	list file(s) in directory (cf. C.6)
`man`	print manual pages. (cf. C.3)
`mkdir`	create directory. (cf. C.7)
`more`	print text in viewable pages. (cf. C.4)
`mv`	rename a file. (cf. C.8)
`pr`	print file with header and date line. (cf. C.4)
`ps`	print status of user/system processes. (cf. C.2)
`pwd`	print working directory (cf. C.2)
`rm`	remove (delete) file(s) in directory. (cf. C.8)
`rmdir`	remove (delete) directory. (cf. C.7)
`script`	create session log file. (cf. C.13)
`tar`	save (restore) files to(from) tape. (cf. C.12)
`uncompress`	uncompress file(s) back to original. (cf. C.5)
`who`	who is logged in on the system. (cf. C.2)

C.1 Login on and off UNIX: `login, exit`

To log onto UNIX, answer the *login* prompt when you enter the computer (when no one else is logged on). Type `exit` to log off.

```
login joeUser
passwd:       # When you type password it doesn't echo.
 . . .
exit (or CONTROL/D)
```

C.2 Where am I with: pwd, who, ps

To find out where you are, print the working directory with pwd.

```
pwd            # Print current working directory.
who            # Print users on the system.
ps             # Print my processes status.
ps -aux        # Print all users' processes status.
pwd; who; ps   # Execute all 3 commands one after another.
```

C.3 Get information of UNIX commands: man

To get information on any UNIX command, use the man function.

```
man ls          # Get manual page information on 'ls'.
man cat         # Get manual page information on 'cat'.
man -k dir      # Get list of all commands which have to
                # do with 'dir' (as in DIRectories).
```

C.4 Print a file: cat, more, pr, lpr, lpq, lprm, print

To print onto your terminal any file which is contained in your directory (eg gel.rc), you *concatenate* it onto your terminal. For example, type:

```
cat gel.rc
cat -n gel.rc
```

The -n causes each line to be numbered on the left.

```
more gel.rc           # Print one page of file at a time.
cat -n gel.rc | more  # Same as above but add leading line #'s.
```

The second example causes the output of the cat to be *piped"* (indicated by the '|' into the more processor which prints one line at a time. In response to the prompt from more, type q key to quit, SPACE key to print the next full page.

```
pr gel.rc
pr -f -h GELLAB-STATE-FILE gel.rc
cat -n gel.rc | pr
cat -n gel.rc | pr | more
```

The pr prints the input with a form feed and header every line printer page. The -f says to preserve any form feeds in the original file and the -h says use the new name in the header printout instead of the file name. This example causes the output of the cat to be "piped" into the pr processor which prints one line at a time. The -v of the cat causes form feeds in the file to be preserved.

```
lpr gel.rc
cat -nv gel.rc | pr -f -h GELLAB-STATE-FILE | lpr
lpq
lprm 247          # kill print job number 247
```

The `lpr` prints the input onto the lineprinter by putting it into the lineprinter queue. The `lpq` gives a status report of what print jobs are in the printer queue. The name of this command may vary depending on your local site. For instance, at NCI/FCRDC it is called `vmsprint` and transmits the data to a VMS/VAX to be printed on its print queues.

```
alias print `cat -nv \$* | pr -f -h \$* | lpr`
print gel.rc
```

Defines a `print` command as an alias for the above sequence of commands. It may be used as if it were a new program.

For printing GELLAB-II UGF plot files, GELLAB-II asumes that there is a specific PostScript laser printer `Plaser` print queue. It is accessed by doing `lpr -Plaser`. See Section ??, page?? for a dicsussion on installing this print queue.

C.5 Compressing files to save space: `compress`, `uncompress`

The `compress` utility compresses a file or list of files saving from 2 to 10 times in the amount of file space used. It is especially useful for compressing image files.

```
ls -l a00123.ppx         # List the original file size.
compress a00123.ppx      # Compress it making .Z file.
ls -l a00123.ppx.Z       # List the new file size.
uncompress a00123.ppx.Z  # Uncompress back to .ppx file.
ls -l a00123.ppx         # List the original file size.
compress *.hex           # Compress all hex image files.
compress a05???.ppx      # Compress all 1000 level PPX files.
```

C.6 Checking files in your directory: `ls`

The user's directory may be checked in various ways using the `ls` command. To find all of the user's files type:

```
ls               # List all files in current directory.
ls *.*           # List files which match anything.anything.
ls ts3???.sps    # List all files matching prefix/extension.
ls ts3*.*        # List all files matching prefix.
```

The * indicates wild card or don't care about either the file name prefix or file extension. The ? indicates a don't care in a particular position of the file name. For example, all picture files beginning with w may be checked.

```
ls ~/gellab/aux/w*.ppx
```

Notice that we have also specified that the files are on ~/gellab/aux/. By adding a $-l$ (letter L) switch, the directory is printed with the dates. To get a hardcopy listing on the lineprinter rather than on the user's terminal, do the following:

```
ls -l ~/gellab/aux/w*.ppx | print
```

To find out which UNIX group a file is use the -ldg switches.

```
ls -ldg ~/gellab/aux/w*.ppx | print
```

C.7 Creating a subdirectory: mkdir, rmdir

It is often convenient to save data files in sub directories contained in some other directory. If you are in some directory called fruit, then the mkdir command can be used to create subdirectories apples and oranges. To remove a directory, use rmdir.

```
mkdir apples             # Make directories.
mkdir oranges
rmdir oranges            # Delete empty directory.
```

C.8 Delete, rename, move and copy file(s): rm, mv, cp

To delete a file use rm, a directory tree rm -r. To rename a file (eg abc to xyz) use mv.

```
rm bad.data               # Delete a file.
rm aaa bbb ccc ddd        # Delete four files.
rm ts3*.log               # Delete all ts3 log files.
rm -r abc.directory       # Delete all files in directory.
mv abc xyz                # Rename abc to xyz.
mv xyz xyz.directory      # Move xyz into directory.
mv ts3*.ppx ~/gellab/ppx  # Move files into directory.
cp xyz xyz.directory      # Copy xyz into directory.
cp ts3*.ppx ~/gellab/ppx  # Copy files into directory.
```

One of the complaints from those nervous among us is that UNIX can do irreversible damage without the user really trying. [See 3rd example below where you have an extra space between '*' and '.data' - so it interprets it as two different file specifications!] Well, the following is an example of this. If you count to 3 before you type Return whenever doing a `rm` you will probably be ok.

```
rm *.data                # Delete all files ending with '.data'.
rm *                     # DELETE ALL OF YOUR FILES!!!
rm * .data               # USER ERROR: THIS TOO WILL DELETE ALL FILES!!!
```

C.9 Changing directories with: `cd`

Often when data is saved in different directories it is useful to be able to switch from one directory to another.

```
cd apples                      # Go down into apples.
cd /home/joeUser/fruit/apples # Go to absolute path.
cd ./apples                    # Go to apples in current.
cd ~                           # Go to your home directory.
cd ~/                          # Go to your home directory.
cd ~joeUser                    # Go to specific home directory.
cd apples/../oranges           # Go down, up then down again.
cd ..                          # Go up to previous directory.
cd ../../footballs             # Go up 2 then down one directory.
```

The first switches to the subdirectory apples. The second does it by the absolute path relative to `/` which is called the system *root* directory. The third does it *relative* to the current directory (called `./` or `.`). The fourth through sixth switches to home directories (in this case yours or `/home/joeUser`). The seventh goes down to apples, then back up to fruit, then down another path to oranges. The eight example goes up one directory level. The last example goes up two levels of directory before descending to directory `footballs`.

C.10 Changing file protection and group membership: `chmod`, `chgrp`

The file protection may be changed with `chmod` and UNIX file system group membership with `chgrp`. A files has several different *access rights* which can be granted to yourself (i.e User $\times$ 0100), the UNIX Group ($\times$ 010) you are a member of , and the rest of the users on the system (Other $\times$ 01). These include: `r` (read) = `04`, `w` (write) = `02`, `x` (eXecute) = `01`.

```
ls -ldg abc    # List protection and group status.
chmod 751 abc  # User: rwx, Group: rx, Other: x.
```

```
chmod 710 abc  # User: rwx, Group: x, Other: none.
chmod 110 abc  # User: x, Group: x, Other: none.
chmod 444 abc  # User: r, Group: r, Other: r.
chgrp gelusr gellab/*.ppx  # change all PPX files to 'gelusr' group.
```

Normally, one would protect your files from anyone else writing it (755) and for patient proprietary information - from anyone reading it (711).

C.11 Search for text string in file(s): grep

Look for a text string phrase which may occur in one or more files.

```
grep apple *.txt        # Find 'apple' in files.
grep -n apple *.txt     # Print line #'s if find any.
grep -i apple *.txt     # Search independent of case.
grep "apple\-pie" *.txt # Find 'apple-pie' in files.
```

C.12 Saving or restoring tape files: tar

It is useful to save you gel data sets, either some or all of the data. Since you may not need some of your primary data or when changing gel data sets, you can archive it to tape. This includes: images, accession file, landmark file, annotation file, GSF, GCF, PCG DB, etc. Doing this is essential if you have limited disk space and have several different gel projects using the system at different times and so must delete and later retrieve files from tape. [1]

```
cd ~/gellab; tar cvf /dev/rst8 .  # Save (create) relative gellab
                                  # directory tree on SUN cartridge tape.
cd ~/gellab; tar cvf /dev/rmt8 .  # Same as above, but 9-track tape.
rm -r ../gellab                   # DESTROY your gellab directory.
tar tvf /dev/rst8                 # LIST directory of tape.
cd ~/gellab; tar xvf /dev/rst8    # RESTORE your gellab directory from tape.
cd ~/gellab/ppx; tar cvf /dev/rst8 .
                                  # Save all original image files
cd ~/gellab/aux; tar cvf /dev/rst8 m*.ppx w*.ppx
                                  # Save all Rmap and mosaic image files
cd ~/gellab/aux; tar xvf /dev/rst8 m00661.ppx
                                  # Restore a particular file to aux dir.
```

C.13 Batch and Session logging: script, at, atq, atrm

It is possible to get a logging file on your interactive session even if it is performed on a video terminal. This is done using the **script** program. You start the script

[1] Use only with extreme care.

program with an optional script log file. Then type `control/D` to terminate `script`. At this point the log file exists and may be printed.

```
script mar19a.ses
    ... run GELLAB and analyze data ...
^D
```

At this point the file `mar19a.ses` is on your disk and may be printed:

```
lpr mar19a.ses
```

```
at 2359 ts3prc.do    # start batch job to process gels at midnight
atq                  # list jobs in batch queue and job #s
atrm 237             # kill batch queue job 237
```

C.14 Shell programming: `csh, foreach, while`

To process a set of gels with sequential numbers one may use the *csh(1)* `foreach` and `while` loop construction (see [WanP88] for discussion on UNIX shells and Appendix C page 593 which lists some of the more often used UNIX commands).

The following example generates Rmaps as well as displaying them X-windows for gels `471.1` through `479.1` for a SPSS file `p07s05.sps` (see **cgelp2** and **markgel** for details).

```
set d='471.1 472.1 473.1 474.1 475.1 476.1 477.1 478.1 479.1 479.1'
foreach x ($d)
     echo ``Creating Rmap of gel'' $x
     markgel $x p07s05.sps
     accppx $x -Prefix:m
 end
```

This may be expressed as a `while` loop for gels `471.1` to `479.1`.

```
set x='471.1'
set d='479.1'
while ($x <= $d)
     echo ``Creating Rmap of gel'' $x
     markgel $x p07s05.sps -Uselms -Correctbackgrd
     accppx $x -Prefix:m
     @ $x += 1
end
```

Alternatively, the accession number *extension* may be incremented for gels `471.1` to `471.6`:

```
set x='1'
set d='6'
while ($x <= $d)
     accppx 471\.($x) -Prefix:m
     @ $x += 1
end
```

C.15 Shell scripts format

To process a program consisting of a of list of UNIX commands, then one would construct a *shell script* file. This file should have *execute* file protection set with *chmod(1)* by doing '`chmod 700` *script-file*'. All scripts mention the name of the UNIX shell (eg `/bin/csh`) to be used in the first line followed by the shell commands. For example,

```
#!/bin/csh
first cmd
  ...
last cmd
```

The following example script segments three gels, pairs them and builds a database (assuming that **cgelp2** script *tst3gels.gdo* exists) be:

```
#!/bin/csh
sg2gii 524.1
sg2gii 525.1
sg2gii 526.1
cmpgl2 524.1 525.1
cmpgl2 524.1 526.1
cgelp2 -f tst3gels.gdo   # Need CGELP2 input script to do it...
```

Appendix D

Starting Xwindows when using GELLAB-II

This section describes how the GELLAB-II user starts up the public domain X-Window System[SchR88] in their area prior to running any of the GELLAB programs. If the user's account has not been set up by the system administrator to run X-Windows, see Appendix D.3 on how to set this up. Section 6.10, page 6.10 discusses installing the X11 version of X-windows if it does not already exist.

D.1 Starting X-Window System with *xinit*

starting X-Windows

Starting X-Windows is actually rather simple. After logging in, just type `xinit` to startup a single window or double window X environment. Once, you get the prompt `%` in the window, you can run any of the GELLAB programs as discussed in the rest of the book. To get out of X-Windows so that you can log off the system, press the middle mouse button over the background window and select "Exit X Windows". This will stop X-Windows and return you to UNIX. You must type CONTROL/D again to log off of UNIX.

the window manager

When the window system starts up, it also starts a so-called "window manager" program, in this case `twm`, to interact with the user. They can use the mouse to interact with existing windows or create new ones. Window operations include: `Move` to reposition a window on the screen, `Resize` to change the size of a window, `Lower` to put a window below other overlapping windows, `Raise` to bring a window to the top, and `(de)Iconify` to turn a window into a small icon and vice versa. The `New Window` is used to create a new terminal window. Briefly, to get the option menu, move the cursor by moving the mouse to a region of the screen where there are no windows. Press the middle mouse button (keep holding it) and move the

mouse to the menu and then the select the option within the menu you desire. At that point, release the mouse button and the operation will be performed. The menu options are fairly self explanatory. Try them out. It is difficult to cause any real damage.

D.2 Some Tools available under X-Windows

other X-window programs

You can invoke any of the X11 utilities from a terminal emulator `xterm` window by just running them. There are man pages available for most of these programs (type `man xterm` for example). In general, the GELLAB-II user need not be aware of these X-Windows tools, but they are available.

emacs (or GNU emacs) is a version of emacs which uses the mouse to position the editing cursor.

twm is the window manager we use with X11R4 and with GELLAB-II. Press the mouse button when the cursor is over the background to get the menu. Holding the button down, move the cursor to the desired menu and then the desired menu selection. At this point release the mouse to select that action.

xcalc is a desk calculator. To use it either click the mouse on the keys or better still position the mouse over the calculator and just type the request (e.g. 123+456=, and it will display the result).

xclock is a time of day clock which is available in either analog or digital form.

xperfmon is performance monitor to visualize the status of your machine.

xterm is a standard terminal emulator for a DEC VT102 and Tektronix 4015 graphics display. Press the mouse (left and middle buttons are different) over the title bar to select additional options. Type CONTROL/D to delete the window. Click on the scroll bar to scroll the window. Position the cursor on the ⇳ and click ⇑ to scroll up and click ⇓ to scroll down.

D.3 User files required for X-Windows X11R4

your default files

In order to use X-Windows, the user, must have the following files in their home directory area. A change may also have to made to the `.cshrc` file to include some UNIX aliases listed below. The following files should be in their home directory. If they are not, copy them from the ~`gelmgr` account into the home directory.

.twmrc X11 `twm` window manager startup file defaults (used with GELLAB-II).

.xinitrc X11 startup file for `xinit`.

Appendix E

MicroEMACS commands used in GELLAB-II

When an X-Windows dialog window comes up on the screen in any of the interactive GELLAB-II programs, the following editing commands are active in the command line. You must Click mouse button on the text line before editing. Many of these microEMACS commands are the same as for the full blown GNU EMACS editor [StaR86].

Ctrl-A move to beginning of line

Ctrl-B move back one character

Ctrl-D delete next character

Ctrl-E move to end of line

Ctrl-F move forward one character

Ctrl-G ignore previous command

Ctrl-K kill rest of line (to right of cursor)

Ctrl-L redraw buffer

Ctrl-N move to next line

Ctrl-P move to previous line

Ctrl-Y unkill last delete

DELETE delete previous character

ESC-< move to beginning of buffer

ESC-> move to end of buffer

ESC-B move back one word

ESC-D delete next word

ESC-F move forward one word

ESC-N move to next buffer

ESC-P move to previous buffer

ESC-DELETE delete previous word

Appendix F

History of GELLAB

The current UNIX based GELLAB-II system was derived directly from the GELLAB-I system. A set of papers was collected together which describes the GELLAB-I system [LemP88b]. These discuss the underlying design and reasons for the design as well as some of the major biological investigations undertaken using GELLAB. Needless to say, most of the concepts used in GELLAB evolved during the many discussions with these collaborators and users of GELLAB - especially Eric Lester ([LesE80], [LesE81a], [LesE81b], [LemP82a], [LesE82a], [LesE82b], [LesE83], [LesE84a], [LesE84b], [LemP89b]) and Peter Sonderegger ([LemP84], [SonP85], [SonP86], [StoE89]), Peter Rogan ([LemP91], [RogP91]), James Myrick ([LemP92], [MyrJ93]). Carl Merril and his group were instrumental in introducing me to the domain of 2D gels and posing the initial problems which got us involved. Special acknowledgement must be made to Lewis Lipkin without whose encouragement, insights and suggestions GELLAB would never have been built.

GELLAB had started out on a DEC PDP8e controlled special-purpose image processor called the *Real Time Picture Processor* (RTPP) constructed by the Image Processing Section under Lewis Lipkin ([CarG84], [LemP74], [LemP77])) and its software was written in FORTRAN ([LemP78], [LemP79a], [LemP80a]). Users of the initial PDP8e based FLICKER program [LemP79a] performed and manually recorded all measurements using the manual interactive system - one pair of gels and one spot at a time.

RTPP

As this was extremely tedious, a decision was made to automate aspects of the measurement and recording procedures which were being done manually - and so the beginnings of GELLAB were implemented on the PDP8e/RTPP. It quickly outstripped the computational capacity of the PDP8e and was then rewritten in the SAIL programming language [ReiJ76] to run on a small DECsystem-10 (model 2020) connected to the PDP8e/RTPP. The RTPP was then used as an image I/O device - being able to both acquire and display processed images using the BMON2 system

SAIL language

[LemP80a]. The SAIL version of GELLAB ([LipL80a], [LemP81a], [LemP81b], [LemP81c], [LemP83a]) has been successfully exported to several sites which had DECsystem-10 or -20 computers. The RTPP front end was replaced at these other sites with various image I/O facilities. Line graphics terminal or plotter realizations of Rmap and mosaics derived images was added to facilitate user interaction when direct gray scale image I/O was not available or the user was dialing in to GELLAB from a remote terminal.

In 1984, a decision was made to export GELLAB to other less expensive and more readily available (i.e. portable) computing environments. Because of the extensive amount of SAIL code involved (about 70,000 lines) a machine translation of the SAIL code to the target portable language was required. C was selected as the target computer language. The PSAIL, Portable SAIL, compiler ([LemP85], [LemP88c]) is a portable SAIL to C compiler capable of translating SAIL source code programs into portable C programs. PSAIL has been running on the DECsystem-10 since 1985 and has been used to convert the GELLAB programs to C. This C code has been modified to ensure a closer fit with the UNIX and X-Window System environment. Currently, GELLAB-II is about 250,000 lines of C code. The original GELLAB-I reference manual was written in RUNOFF on the DEC10. Much of that, as well as new material, was used as the basis for this GELLAB-II book.

PSAIL compiler

UNIX and X-windows

The new C/UNIX based GELLAB is called GELLAB-II, and the older SAIL/DECsystem-10(-20) based system, GELLAB-I. GELLAB-II has been reimplemented in a UNIX windowing environment, although it runs on UNIX systems without graphics. The windowing system selected is the X-Window System [SchR86] which is a public domain windowing system from Project Athena at MIT. Initially, GELLAB-II has been brought up on a SUN and microVAX-II (under ULTRIX) and later on will be brought up on other UNIX machines and other systems. The current exported version is SUN4 (Sparc) based.

In September, 1992, NCI entered in a CRADA with CSPI, Inc. of Billerica, Mass to develope a commerical version of GELLAB-II. The final system will run on a PC-compatible computer under the Windows-NT operating system and using their array processor board to give it the type of power required for 2D gel analysis as well as being relatively inexpensive.

Software emulations

As some of the capability of the SAIL environment on the DECsystem-10 which is required for GELLAB did not exist in C/UNIX, it had to be emulated. Two major subsystems which are used extensively in GELLAB are the SAIL LEAP associative language ([ReiJ76], [LemP88c]), and the OMNIGRAPH 2D and 3D graphics library developed on the DEC10 at DCRT/NIH [DCRT79]. These are emulated in two C packages LEPCGL and SMDISP. LEPCGL implements a major subset of LEAP including items, sets, lists and all of the operations on them. SMDISP emulates a

major subset of DECsystem-10 OMNIGRAPH required for GELLAB. Unlike the original OMNIGRAPH which handled a variety of graphics terminals, SMDISP generates only Tektronix-4010 type terminal graphics. It will later generate X-Window calls directly. However, it also can generate raster Portable PiXture (PPX) images of the graphics as well as PostScript files for printing on laser printers. PPX or Portable PiXture file format images are variable size grayscale or pseudo-color images with a 512-byte format header description block in the front of the file. The header is defined by a C *typedef* in file `ppxfmt.h` (listed in Appendix H).

The RTPP image processor has been replaced by two programs. Image display and manipulation is now done using the **Xpix** program [LemP88a] which runs under the X-Window System. **Xpix** runs on 8-bit (or greater) high resolution color graphics UNIX workstations. Gel scanning data acquisition is done using a high resolution CCD camera (Datacopy autofocus model 612F) connected to the Small Computer System Interface (SCSI) interface of the SUN and uses the **getacc** program. Program **camera**, is used to take a picture with the Datacopy camera into an image file. Then program **getacc** uses **camera** to actually scan the gels. GELLAB-II is able to convert a variety of TIFF and other image file formats (Molecular Dynamics, BioImage, Elsie, etc.) into our standard PPX format.

In **getacc**, a calibrated neutral density wedge strip is scanned along with the gel autoradiograph or silver stained gel and is used to then calibrate the gray scale gel image data in terms of optical density. The camera data acquisition library package `libdcc.c` which **camera** uses has been designed so that other camera systems could be easily substituted - provided that they have equivalent functionality. [SCSI hardware interfaces are either supplied with most UNIX workstations or can often be easily added.] As a result, the graphics and data acquisition subsystems of GELLAB-II should be relatively easy to port.

In the process of converting GELLAB from SAIL to C, many of the programs were renamed to reflect enhancements. These include: SG2DRV ⇒ `sg2gii`, CMPGEL ⇒ `cmpgl2`, CGELP ⇒ `cgelp2`, SEERSPOT ⇒ `mosaic`, PIXRTPP ⇒ `accppx`, PGEL ⇒ `pgelrc`, PIXODT ⇒ `ppxodt`, PIXCVT ⇒ `ppxcvt`.

Appendix G

GELLAB-II - Selected Algorithms

This Chapter discusses the major GELLAB-II algorithms in more detail. Some of this material is taken from Lemkin and Lipkin [LemP83a].

G.1 Gel Segmentation: SG2GII

G.1.1 Gel segmentation: Nature of the image

Gel image accessioning and acquisition is only the first stage in making spot information available for automated processing. Provision must be made for separating out "pictorial information of interest" (in this case the spots) from "noise" and "background" before spot positions and spot properties can be compared. Consequently, a spot extraction algorithm must be capable, under a wide variety of actual gel image conditions, of 1) detecting, 2) defining the extent of, and 3) measuring the density of a spot.

The segmentation problem is one of the more important and ubiquitous, in the general field of image processing. Almost all real images resist simple gray scale thresholding as a solution to pictorial partitioning or segmentation and despite the simplicity of spot morphology, 2D gels are no exception. The thresholding operation applied to an image retains all values higher than a certain gray value while all others are set to white.

G.1.2 Spot morphology

The vagueness of spot morphology and inhomogeneity of gel background complicate these images. Spots often touch each other resulting in overlapping spots. The "tails" of spots may extend for a considerable distance into an overlapping region. Spots have no distinct boundary, but occur most often as an effectively continuous Gaussian-like distribution which Lutin [LutB78] asserts that this distribution tends to be symmetric in isoelectric point (pIe) but is skewed in molecular weight (MW) [GarJ79]. In practice, this holds only for ideal non-conglomerate spots. In addition, spots may appear round, oblong or take on various continuous shapes particularly when there is excessive loading of material in the gel. In all cases, except in the case of extreme overloading, however, the center of a spot is its darkest part. Spots may sometimes be obscured in certain regions of the gel which are susceptible to streaking in both MW and isoelectric axes. Spots which may be of interest can occur within these streaks and so we must be able to extract those spots from streaks.

Polypeptides in the gel are not visible by themselves and must therefore be visualized in order to perform an analysis. At least five methods of spot detection are currently used which include: Coomassie blue staining, autoradiography (on radioactively labeled proteins produced by growing the tissue culture in radiolabeled amino acids), silver staining, fluorescent dyes, and interference optics. These spot detection methods have widely different dynamic ranges and stochiometry, as well as application for different types of biological material.

Care must be taken to insure that autoradiographic film is used in the linear portion of the density versus log (exposure) "gamma" curve otherwise saturation of some spots will occur. The dynamic range of spot detection may be covered using a series of increasingly long autoradiographic exposures of the same gel. The Vidicon, CCD, photodiode imaging detectors are subject to similar saturation problems. The more expensive laser scanners do not suffer from this problem and may go up to 4.0 OD - higher than the linearity of most films. Although the Dupont type Cronex Wide Dynamic Range (WDR) medical X-ray film is nearly linear over this range (0 to about 3.5 OD) and obviating the need for multiple exposures.

Because some spots will be recorded as saturated, it is useful to know which ones and furthermore to be able to track these spots throughout the entire analysis process. This is done in GELLAB-II. Spots saturating in one gel might not do so in another so that alternative measurements could be made as for example in the case of multiply exposed autoradiographs.

G.1.3 A segmentation model

In any locally determined (e.g. non-thresholding) feature extraction process, some explicit or implicit model of the pictorial objects is necessary. The algorithm presented here embodies some of the ideas of the underlying spot model. A first order

model is the triple (x, y, d) consisting of the spot's centroid (x, y) in Cartesian space and its total integrated density d (a measure of polypeptide concentration). This triple appears adequate for many types of multiple gel analyses where the object of the analyses is to measure the amounts of polypeptides present. Segmentation is a method of spot extraction which results in obtaining this triple as well as other features. Our spot segmentation algorithm is based on a shape and density independent model and takes into account the realities of touching and overlapping spots.

G.1.4 Role of segmenter in overall gel analysis

As shown in Figure 1.3 in the **Introduction** page 43, the segmenter is applied following gel image acquisition. It is important that the segmentation procedure be made as automatic as possible with minimum manual intervention because of the large number of spots on a gel.

We present here a specific spot extractor which is able to handle a wide variety of spot shapes, density and cluster morphology. However, any spot extractor generating an ordered list of spot triples (x, y, d) could be used in the first stage of the GELLAB-II analysis. Parameterization of the segmentation algorithm permits a wide variety of gel stains to be handled, and produces different types of output which can be put to varied uses.

G.1.5 The Segmentation Algorithm

The segmentation algorithm is a sequence of procedures applied to a locally averaged image. The first of these is the digital analog of the spatial second derivative; it is used to construct an image called the central core image consisting of the centers of spots. Second derivative information delimits the extent of outward propagation resulting in an algorithmic limit on individual spot extent. Initial spot candidate generation is parameter independent. The decision function which later separates noise from valid spots is adjusted by user defined parameters. Auxillary information required by the segmenter, such as picture file name, and ND wedge calibration and computing window is obtained from the accession file. The algorithm is:

ALGORITHM: SG2GII - GEL SPOT SEGMENTATION

```
[1] FOR each line in the image DO
   [1.1] Compute NxN pixel Gaussian average of original picture.
   [1.2] Compute second derivative (magnitude,direction) images. Set
         central core (CC) image pixels to 1 if direction<0 in dX
         and dY else 0.
[2] FOR each non-zero central core image pixel DO
   [2.1] If isolated pixel (not 8-neighbor connected) remove by setting
         to 255 code.
   [2.2] Find spot list L[1:TOP] at current (x,y) in central core image.
      [2.2.1] If enabled, extend saturated (i.e. g > P% maxOD) CCs to
              adjacent regions with high density.
      [2.2.2] If enabled, split CCs with contain multiple sub-spots as
              defined by Miller and Olson [AdoA89].
      [2.2.3] Remove single pixel wide concavities by adding to CC image.
      [2.2.4] Number the CC pixel image from list L with next free value
              in [2:99] Mod 99.
      [2.2.5] Heuristically propagate out from CC with pixel value
              CC+100 extending the list L[1:TOP].
      [2.2.6] Fill remaining empty corners of spot adjacent to CC part
              of spot extending the list L[1:TOP].
      [2.2.7] Fill holes in the central core part of the image extending
              the list L[1:TOP].
      [2.2.8] Fill corners in propagated central core of image extending
              the list L[1:TOP].
      [2.2.9] Round corners in propagated central core of image extending
              the list L[1:TOP].
      [2.2.10] Compute spot features and delete spots failing sizing
              criteria otherwise save spot in SPOTLIST[N].
[3] FOR each spot i in SPOTLIST[N] DO
   [3.1] Compute restOfImage = (original image - spot[i]).
[4] FOR each restOfImage pixel DO
   [4.1] Compute zonal notch filter image mnBkgrd (estimate of gel
         background).
[5] FOR each spot i in SPOTLIST[N] DO
   [5.1] Recompute spot features D'=(D-A*mnBkgrd) and delete spots failing
         sizing criteria.
   [5.2] Dump passing spot and its features meeting sizing criteria into
         Gel Segmentation File.
```

Principle

Let g be a image gray scale point function whose mode, median and mean are all more or less central with respect to the extrema and let its second derivative be g''. The central region of a spot has a negative g'' direction and a g'' magnitude maximum. Beyond the mid-region where the direction of g'' changes sign, there is a second smaller peak in the magnitude of g''. Our segmentation procedure is based on finding these two maxima in 2-dimensions. The approximation to the boundary is operationally defined by the second maxima in the g'' magnitude function.

Smoothing

The original image is first smoothed to remove some of the high spatial frequency image noise. This is illustrated using a 3x3 convolution filter. Let matrix M be defined as:

$$M_{3x3} = \begin{matrix} m_{-1,+1} & m_{0,+1} & m_{+1,+1} \\ m_{-1,0} & m_{0,0} & m_{+1,0} \\ m_{-1,-1} & m_{0,-1} & m_{+1,-1} \end{matrix} \tag{G.1}$$

Then,

$$M_{3x3} = \begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix} \tag{G.2}$$

Then, for a center pixel (x, y), each smoothed pixel $f(x, y)$ is defined as:

$$f(x,y) = (1/16) \sum_{i=-1}^{+1} \sum_{j=-1}^{+1} m_{ij} g_{x+i,y+j}. \tag{G.3}$$

It is applied over the entire picture, pixel by pixel in a top to bottom, left to right fashion (called a *raster scan*). Each pixel in the 3x3 pixel neighborhood (defined by the center pixel) (after begin mapped from gray scale to OD) is multiplied by the corresponding 3x3 filter (pixel for pixel) and the total divided by 16. The result (after being mapped back to gray scale from OD) is saved in an *averaged* image. This filter removes enough of the high spatial frequency noise so the second derivative analysis algorithm may be more successfully applied. Spot shapes are not distorted to any noticeable degree. The actual spot density measurements are made on the original image data. Larger filter matrices M, a 5x5, and a 7x7 (from [VoKP81]), are also used with the 7x7 being the default for the 512x512 image. A 13x17 size filter is also available with the coefficients being the values of a circular Gaussian $\exp \frac{-x^2+y^2}{2\pi\sigma^2}$ whose size is determined by an additional parameter σ (default value of 6).

$$M_{5x5} = (1/52) \begin{matrix} 1 & 1 & 2 & 1 & 1 \\ 1 & 2 & 4 & 2 & 1 \\ 2 & 4 & 8 & 4 & 2 \\ 1 & 2 & 4 & 2 & 1 \\ 1 & 1 & 2 & 1 & 1 \end{matrix} \tag{G.4}$$

$$M_{7x7} = (1/441) \begin{matrix} 4 & -6 & -12 & -14 & -12 & -6 & 4 \\ -6 & 9 & 18 & 21 & 18 & 9 & -6 \\ -12 & 18 & 36 & 42 & 36 & 18 & -12 \\ -14 & 21 & 42 & 49 & 42 & 21 & -14 \\ -12 & 18 & 36 & 42 & 36 & 18 & -12 \\ -6 & 9 & 18 & 21 & 18 & 9 & -6 \\ 4 & -6 & -12 & -14 & -12 & -6 & 4 \end{matrix} \quad \text{(G.5)}$$

Central core and magnitude second derivative images

The second derivative is computed as the vector (d^2x, d^2y) using the following 3x3 difference formulae (in a similar manner to the convolution filter of discussed above. These filters are applied to the averaged image that was just computed. If the 5x5 option is used, then it uses the second set of filters.

$$d^2x = \begin{matrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{matrix} \quad \text{(G.6)}$$

$$d^2y = \begin{matrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{matrix} \quad \text{(G.7)}$$

5x5 Laplacian:

$$d^2x = \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & -4 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \quad \text{(G.8)}$$

$$d^2y = \begin{matrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -4 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{matrix} \quad \text{(G.9)}$$

The *magnitude* image of g'' is approximated by the city block distance - the sum of the absolute values of d^2x and d^2y. The direction image is not actually computed. Instead a *central core* image pixel is defined as having a "1" where both $(d^2x < 0)$ and $(d^2y < 0)$, and being "0" everywhere else. Both the average and central core images are computed during the first raster scan through the image. (A raster scan

traverses an image line by line from top to bottom as it goes left to right on each line.)

Extracting a spot

In a second raster pass through the image only those pixels coded as "1" are processed. Isolated pixels, defined as being 4-neighbor unconnected, are marked for deletion by setting them to a 255 code. 4-neighbor connected pixels are those adjacent to another pixel on either the horizontal or vertical axis. Otherwise, each time a "1" code is encountered, a spot pixel list (SPL) is computed as in step [2.2]. A push down stack is used to keep track of all unexpanded pixels. An unexpanded pixel is defined as one found to be a neighbor of an expanded pixel but which has not been checked (i.e. is not on the SPL or push down stack). Each unexpanded pixel is expanded and checked to determine whether any of its 4-neighbor pixels have a "1" code and are not already in the SPL. Unexpanded pixels so identified are put into the push down stack while the pixel being investigated is saved in the SPL. The algorithm keeps processing the push down stack until it is empty. The spot will then be processed to completion using the SPL which will grow as the spot is propagated to the region approximated by the second derivative magnitude function's second local maximum.

Removing spot concavities

Single pixel wide artifactual concavities which occasionally occur and are removed by checking each SPL pixel C for the following four neighborhood conditions. (By neighborhood we mean here the central pixel in question and its 8 adjacent neighboring pixels.) If a condition is found to be true, the "0" valued pixel in the central core image is changed to a "1". The SPL is also updated. In each of the following cases, a "0" and "1" *must* occur and a "-" means "don't care".

```
1 0 1          1 1 -          - - -          - 1 1
1 C 1   or     0 C -   or     1 C 1   or     - C 0
- - -          1 1 -          1 0 1          - 1 1.
```

Numbering the central core image

In the central core image, the spot is then assigned the next sequential number in the range of [2:99] modulo 99. All SPL pixels in the central core image for that spot get that number. It is very unlikely but in an extremely densely populated spot image, it is possible for two adjacent spots to have the same value for successive lines. This notation conflict problem could be solved by alternative coding schemes using larger numbers. A spot editor such as could be used to correct these types of errors should they ever occur.

Propagating the central core

The numbered spot is then propagated with the value (C+100) from the central core (C) value of the spot to the *propagated central core* region. This propagation from a central core edge point is performed in each of the 4-neighbor directions until it is terminated based on various constraints. (Whereas the 8-neighbor definition of a neighborhood included the corner pixels, the 4-neighbor definition does not.) The SPL is updated with the new pixels. The heuristic conditions for propagation termination are:

1. The second derivative magnitude is increasing (starting 1 pixel out from the central core), outward from the central core indicating a second local maxima.
2. The second derivative magnitude outward from the central core has the same value twice in a row indicating a noisy edge.
3. The propagation would impinge on another central core pixel.
4. The propagation would extend beyond the computing window.
5. The propagation would impinge on an isolated pixel.
6. The gray value outward from the central core is increasing instead of decreasing indicating that the spot is overlapping a much larger spot.

Corner filling

This type of heuristic propagation sometimes forms small rectangular empty corner regions in the four corners of the spot. Such corners can be filled with propagated central core values. Both 0 and 255 (isolated pixel) corner values are candidates for filling (by changing to N, i.e. $C + 100$) if the central pixel is its central core C. The four corner cases are expressed as neighborhood conditions as follows. In the corresponding positions of the neighborhood surrounding a pixel in question, C is the central core value, N is the propagated central core value, E is either 0 or 255, and '$-$' meaning "don't care".

```
E N -          - N E          - - -          - - -
N C -    or    - C N    or    - C N    or    N C -
- - -          - - -          - N E          E N -.
```

Hole filling the central core

In very large saturating spots, the center of the spot may not be detected as such and thus not segmented. The spot will have a doughnut topology. This problem is repaired by filling any artifactual holes in the central core region. The leftmost

and rightmost horizontal coordinates for each line of the central core are found and saved as run length codes. Then any 0's in the central core image between these points are changed to central core values.

Splitting merged saturated spots

We split large near-saturated merged spots by using a robust boundary analysis algorithm after the initial Laplacian of Gaussian (LOG) spot detection phase. It finds matching concavities on the border of the initial central core image after the merging algorithm has been applied to a saturated spot. Then, we find strong concavities and apply other constraints such as that it must have a high aspect ratio for splitting to occur. We then pass the spot back to the segmenter to resegment the pieces.

Concavity filling of propagated central core

Occasionally, concavities may appear in the propagated central core image. These are filled by applying the same hole filling algorithm as in step [2.1.5] but for *all* spot pixels.

Rounding spot corners

The propagation algorithms applied above tend to leave the corners rather sharp. These are rounded out by applying the following neighborhood conditions (as in step [2.1.8]) to each central core pixel. If an exact match is made, then the 0 pixel on the diagonal is propagated and thus rounds out the corner (value N being C+100).

```
0 N -       - N 0       - - -       - - -
N C -       - C N       N C -       - C N
- - -       - - -       0 N -       - N 0.
```

Spot features and initial sizing

After the final SPL is computed, it consists of the pixels in the central core and propagated central core. Several features are computed using density values mapped from the average image. A preliminary spot sizing is performed to remove most of the background noise spots in step [2.1.8] where a 254 code is also placed in each deleted spot pixel in the propagated central core image.

Background correction

A background density correction is performed during a third pass through the image using a zonal notch filter algorithm. The notch filter is described in [SchA77]. A running average of the averaged image is computed (see ([LipL79], [LemP79b], [LemP82a]) for algorithm description) for a $n \times n$ movable averaging window masked by the *complement* of the central core image. That is, an image consisting of

background pixels which are not isolated pixels, deleted spots, central cores or propagated central cores. The resultant image constitutes the background image and may be saved in the second derivative magnitude scratch image since it is now no longer needed. The mean background for a spot is then estimated by simply reading the background image at the spot centroid of the spot.

Computing corrected density D′ and secondary sizing

The features presently used to determine acceptance of a spot include: spot area (in square pixels), total integrated corrected spot density and range of pixel OD seen in the spot. The last is useful for eliminating small noise spots from the image. The corrected spot density D' is computed from D taking the background estimate into account. For spot i, $D'_i = D_i - D_i^{bkgrd} A_i$, where D_i = uncorrected integrated spot density, D_i^{bkgrd} = mean background density/unit area, and A_i = total spot area. Those spots which meet the criteria are saved in the Gel Segmentation File (GSF) (see page 465 for example). Spots failing the final density sizing criteria are deleted as before by having 254 placed in the central core image of each pixel. The gray value numeric data for the final set of spots may be optionally saved in an output image.

It is possible for the spot feature sizing parameter limits to be made more restrictive to eliminate some of the smaller noise objects.

G.2 Gel Pairing: CMPGL2

We have treated the problem of spot extraction within a single gel using the **sg2gii** program. Now we consider the first step in locating a particular spot in a set of gels - i.e., pairwise matching of the spot in two gels. This can be done by shifting one of the gels until the spot overlaps and recording the Cartesian coordinates of the spot in each gel. This spot by spot pairing is a prerequisite to detecting whether individual polypeptides change with respect to experimental conditions. Furthermore, pairing of spots within a set of gels taken two at a time is the means whereby a multiple gel database is gradually constructed.

Referring back to Figure 1.3 in the **Introduction** page 43, gels are first acquired, then spots are segmented using the **sg2gii** program. This resulted in a gel segmentation file (GSF) consisting of a list of spot (x, y, d) triples (see example in Section 3.18 page 452. Note that in the GSF each spot has a spot index (which can be used to refer to the spot), a (x, y) centroid and a density measurement given in several formats. An algorithm for spot pairing between two gels using a small set of landmark spots to locally align subregions is presented which uses the GSF spot list files as data. The **cmpgl2** program implements this algorithm producing a gel comparison file (GCF) (see page 325 for example).

In the analysis of 2D gels the argument for automation does not rest on problem complexity, though the problems are biologically complex, nor on image processing brilliance though the algorithms are efficient and appear sufficient for the task. A gel analysis system is necessary because most gels contain a very large number of spots, spots which are at best only locally congruent from gel to gel, which can not be counted on to maintain "shape" or optical density and contain little infrastructure on which to build a characterization. There is at best a congruence in local regions between two gels related by some a priori undetermined affine transformation.

G.2.1 Partitioned search in pairing spots

A major problem complicating spot localization is the local distortions in the gel such that neighboring spots in one gel will likely be neighbors in another gel while the intervening distances between them vary to some degree. The GELLAB-II semiautomated method for aligning corresponding spots in two gels is discussed.

divide and conquer

G.2.2 A landmark driven spot pairing

We present an alternative view of the primary pairing algorithm. This involves in effect constructing a projected image composed from the two members of the gel pair. In the actual matching, computations are performed only on (x, y, d) data in a single plane - the representative gel plane. Central to the algorithm is the establishment of landmark spots that serve to "anchor" the other spots in its vicinity. Essentially, landmark spots are manually aligned in the two gels at which point the computer automatically aligns all other spots with the corresponding spots in the other gel. The procedure is simple and is easily extended to align any number of gels.

Such partitioning by landmark region increases the efficiency of intergel spot matching by providing an empirical basis for the partitioning of a gel image into tractable corresponding subregions.

selecting landmarks

A landmark spot should be selected according to particular criteria. It is a morphologically distinctive spot present in all gels such that neighboring spots and the landmark spot form a consistent morphologic structure. Moreover, this morphologic structure should be easily recognized across the set of gels. The landmark spot should not be a touching spot. The set of landmark spots are selected to fairly easily cover the regions of interest of the gel fairly. From 10 to 25 landmarks are generally selected depending on the quality of the gel with fewer required for better gels. This set of spots is called the landmark set. In practice, the operator aligns the landmark spots in the two gel images using the flicker algorithm [LemP79a] which is incorporated in the **landmark** program, this permits the left gel to move while keeping the right one's position constant. Viewing time for each of the images may be independently set and varied until the user is satisfied that the two images of the

same spot are locally "superimposed". The superimposed spot of interest is noted to the computer and the next landmark spot processed in the same fashion. The flicker procedure is described in [LemP79a] and Section 3.1 page 152 and landmark acquisition in Section 3.8 page 362.

GELLAB-II also offers alternative facilities for generating landmark spot data without using the **landmark** interactive program (which requires the X-windows-system) see Section 3.8 page 362. Program **dwrmap** draws a labeled outline-plot of the segmented gel from the GSF file with the darkest spots (sorted by density) labeled in the plot with a table also given on the side of the plot. Spots are represented by an oval proportional to spot density. By manually comparing such Rmap plots, corresponding lists of landmark spots can be defined using the CC#s of each gel. The `-lmsedit` option in the **landmark** program then allows the manual entry of a landmark set as a list of pairs of CC#s from the two gels.

The landmark region surrounding a landmark spot is defined as a polygonal region having higher pairing certainty for spots closer to the landmark spot. The half-radius of certainty R_i for landmark i is a distance defined to be half the distance from the nearest landmark spot landmark i. Spots within the half-radius of a landmark set have a higher probability of being aligned (since the landmarks have "perfect" inter-gel alignment) than if the spot were outside of this radius. The landmark spots in each gel are compared with the two GSF spot lists and the best segmented spot's centroid is used rather than the coordinates manually produced. If no spot can be found for a landmark within specified error bounds (currently the dT_2 distance - see below), then the manual landmark coordinates are used.

valid landmarks

The **cgelp2** command `VALIDLANDMARKS` computes a table and statistics of all valid landmarks for all gels in a multiple gel data base. It is possible to ascertain how reliable the particular pairing actually was by backchecking paired spots in a set of pairing-labeled images to be discussed.

Partitioned search has the added advantage that landmark regions contain an order of magnitude fewer spots than the total gel space. Therefore the combinatorics of performing the spot matching is greatly decreased as well.

G.2.3 Algorithm landmark-oriented spot pairing between two gels

The spot pairing algorithm is illustrated below.It is implemented as the **cmpgel** program. Pairing is performed in two passes through the landmark sets data using the primary and secondary pairing procedures. Each procedure operates on one landmark set at a time, in both gels. The algorithm is:

ALGORITHM: CMPGL2 - GEL SPOT LIST PAIRING

```
[1] Read landmarks for the two gels from landmark database.
[2] FOR each non-zero central core image pixel DO
       Read the two gel segmentation files (GSF)s into lists L1 and L2.
[3] If -CGfile for composite spots
       Read in cgfile and construct additional L1 and L2 spots with CP label.
[4] FOR all landmarks DO
       Compute minimum radii, nearest and next nearest neighbor landmarks.
[5] FOR all spots in L1 and L2 DO
       Assign spots in L1 and L2 to landmark spot sets.
[6] FOR all landmark sets DO
       Perform 'initial' SP, PP, US, AP spot pairing.
[7] FOR all landmark sets DO
       Perform 'secondary spot pairing of AP and US to SP, PP, US.
[8] FOR all landmark sets DO
       If generating 'marked' image put spot label for each spot into
       copy of original gel image.
[9] FOR all landmark sets DO
       Ouput L1: SP, PP, AP, US, CP labeled spots into Gel Comparison File.
       Ouput L2: AP, US labeled spots into Gel Comparison File.
```

In the primary pairing algorithm, spots are first mapped to the Cartesian coordinate system defined by shifting the landmark spot to (0,0) relative to the origin in the two gels G1 and G2. Each spot in G1 is provisionaly paired to the spot that is its nearest neighbor (by minimum Euclidean distance) in the projected image of G2. Because of possible asymmetry of the two landmark regions, the reverse comparison is also performed so that each spot in G2 is provisionally paired with its nearest neighbor spot in G1. This nearest neighbor distance is denoted dP (pair distance). The distance from the landmark spot to the mean locus of the two spots in the provisional pair is denoted dL. Two user specified parameter distances are empirically defined: dT_1 and dT_2. Spots closer than dT_1 to the landmark spot are relatively well paired. Spots greater than dT_2 are poorly paired and possibly should not be paired. The current values of dT_1 and dT_2 (5 and 10 pixels respectively) were determined empirically, by examination of the nearest neighbor values of many sets of paired gels under gel resolution range of 170 to 250 microns/pixel.

Five types of pairing labels can be defined. During primary spot pairing labeling assignment, each potential nearest neighbor spot pair in a landmark set is assigned one of four labels: SP - sure pair, PP - possible pair, AP - ambiguous pair, US - unresolved spot, CP - composite group pair. The labelings are defined by the *pair label rules*

following cases:

```
[1] US - unresolved spot (no dP);
[2] SP - dL < Ra and dP < dT1;
[3] PP - dL > Ra and dP < dT2;
[4] PP - dL < Ra and dP > dT1 and dP < dT2;
[5] PP - dL > Ra and dP < dT1;
[6] PP - dL > Ra and dP < dT2. For the other spot
    AP' - dL' > Ra and dP' < dT2 and dP' > dP (i.e. better match);
[7] US - unresolved spot (no dP).
```

primary & secondary pairing

The primary pairing algorithm is a simple first order model not taking some spots on the periphery of the landmark region into account. These spots may be misclassified as an AP or US whereas they would be a SP and PP classification in another adjacent landmark region. To correct these few misclassification errors, a secondary pairing algorithm is applied in order to possibly re-pair AP and US spots in the next-nearest landmark set using AP and US spots from those sets. The resultant re-paired spot pair (either a SP or PP if it meets the threshold criteria) is then placed in the landmark set with the smallest dL value.

The CP spot is a synthetic spot consisting of a user defined group of spots. This group is treat as if it were one spot with $D'_g = \sum_{i \epsilon g} D'_i$ with its centroid being the weighted centroid of the spots in group g. By definition, CP spots in two different gels are automatically paired by virtue of having the same name.

CMPGL2 output

Finally, after spot pairing, the program can optionally draw the labels into copies of the original images. The paired spot data, (x, y, d) sorted by landmark sets, are then output into the gel comparison file (GCF) (see page 331 for an example). Other information regarding the identity of the two gels and gel segmentation files as well as the manually defined landmarks is part of the permanent preface to the GCF. The estimated landmark spots from the GSF found in the GSFs are also reported as is the Euclidian distance from them to those manually defined by the user. If this distance dL from a landmark is greater than dT_1 for either $G1$ or $G2$, than that landmark spot is so marked and the GSF spots are partitioned using the manually defined coordinates. landmark spot sets. At the end of the GCF is a statistics summary for both the primary and secondary pairing regarding the number of each of the four pairing assignments.

G.2.4 Pairing biases due to landmark selection

The partition of the plane into variable sized polygonal landmark regions is based essentially on the local spread of landmarks. A priori one would think that pairing

would be more likely to be correct in regions of high landmark concentration. However, a small radius of confidence may have undesirable effects on pairing i.e. spots that would otherwise be matched as sure pairs might be entered into the probable pair category. It is possible, for a spot to be paired, to be found in the *next*-to-next-nearest neighbor landmark set rather than the landmark or next-nearest landmark sets. In digital space, the problem of a possible shift of a spot from one landmark region to another, such that pairing would be affected, as a result of increasing the concentration of landmarks is obscure and does not seem easily treated.

The consideration of correctness and completeness of the primary pairing algorithm is not simple although the algorithm in itself is quite straightforward. Performance should not be gauged exclusively on the results when gels of widely different spot numbers are compared. On the other hand, comparisons of closely similar gels should yield good results.

The current pairing algorithm defines a sure pair (SP) as being within the landmark radius R_i for a given landmark i. We have found that most of the possible pairs (PP) are actually paired and should be pooled with the sure pairs as well-matched spots. Large numbers of ambiguous pairs and unresolved spots result when comparing two widely different or noisy gels. Currently, nothing is done with these (AP and US) spots. Although they are tracked through the data base. A possible extension to GELLAB processing would be to incorporate additional procedures to further process the AP and US spots such as merging AP fragments with the spots they belong with. Conglomerates of spots sometimes appear as single spots and other times as several spots, e.g. actin complex, so that merging spots is an attractive idea under the right conditions.

more on selecting landmarks

We have found that highly populated spot regions should have somewhat more landmarks, however landmarks should not be "on top of" each other. Other criteria in landmark selection include using fewer landmarks if the regions have little distortion and line up fairly well. A landmark spot should be well defined morphologically and non-touching and optimally being part of a locally consistent pattern in all of the gels to be compared.

The ability to pair most of the spots in a set of gels enables examination of larger gel databases where subtle shifts and correlations in the spot data can be more easily detected. These can be checked using log density vs. log density scatter plot of two normalized paired (SP and PP labels only) gels (such as would be computed with the `DDPLOT` **cgelp2** command). Most of the spot pairs are close to the 45 degree line. Some of the outliers are real and some are due to noise in the entire gel-image processing system. We will now consider techniques for further resolving noisy data using multiple gels and means for facilitating the checking of outliers.

G.3 Creation of PCG DB using CGELP2/CREATE command

The **cgelp2** program is used to construct and manipulate the PCG DB. The following algorithms show how the PCG DB is constructed from a set of GCF files and how extrapolated Rspots (eRspot) spots for gels with missing spots are generated.

G.3.1 Generation of initial Rspot set PCG DB: CREATE-CGL-DB

The first step in the construction of the PCG database is the generation of the list of Rspot sets. This is invoked by the `CREATE` command. The set of $n-1$ GCFs (Gel Comparison Files) are read one spot pair at a time for each gel pair where one of the spots is a Rgel spot. Each gel pair, referenced by a "key" for the Rgel spot, is formed for this pair. The database is then tested to determine whether a Rspot set currently exists for that Rgel spot. If it does not, then a new Rspot set is created and both spots are put into that new Rspot set. (The Rspot set is actually stored in the PCG DB as a linked list.) If it does, then the other spot in the pair is inserted into that set. In either case, the Rspot set is initially rank ordered by density, darkest first. If the `CREATE` command is invoked with the `/ERspot` switch, then pass 2 is performed to create the eRspots. Alternate Rspot set orderings may then be routinely performed as part of the analysis by changing the density normalization method and sorting each Rspot set's linked-list using the `REORDER` command. The algorithm is:

ALGORITHM: CREATE-CGL-DB - CONSTRUCTION FROM GCF'S

```
[1] FOR each GCF file i DO              /* PASS 1 */
    [1.1] Read GCF paired gel data for (Rgel,Gi) as [Gr,Gi].
    [1.2] If (([Gr,Gi] is SP, PP, CP, AP) Or (Rgel is US) And
              (Not Gr in PCG DB)
       [1.2.1] If Rgel spot j not found Make Rspot node Gr.
       [1.2.2] Make Rspot node for Gi.
       [1.2.3] If Gr spot j found And Old[Gr,Gi] is AP And
                 Old[Gr,Gi] is (SP, PP) Change Old[Gr,Gi]
                 pairing label to new (SP, PP) one.
        [1.2.4] If Gr spot j not found,
           [1.2.4.1] Create new Rspot set j as next free.
           [1.2.4.2] Insert Gr into Rspot set j.
           [1.2.4.3] If (Rgel Not US) insert Gi into Rspot set j.
                     Else Insert Gi into Rspot set j.

[2] FOR each GCF file i DO              /* PASS 2 */
    [2.1] Read GCF paired gel data for (Rgel,Gi) as [Gr,Gi].
    [2.2] If (Gi is US)
       [2.2.1] If Rgel spot j not found Make Rspot node Gr (EP).
       [2.2.2] Make Rspot node for Gi with US label.
       [2.2.3] If Gr spot j not found,
          [2.2.3.1] Create new eRspot set j as next free.
          [2.2.3.2] Insert Gi into eRspot set j.
          [2.2.3.3] If (new Gr spot) insert Gi into eRspot set j.
                    Else Insert Gi into eRspot set j.
```

G.3.2 Generation of extrapolated Rspots in PCG DB: EXTRAPOLATE

An extrapolated Rspot (denoted by a Extrapolated Pair (EP) label) spot in a Rspot set is a synthetic spot with zero density and area which indicates the position of where a spot would occur if it did exist. EP spots in Rspot sets are not created during the initial construction of the PCG DB. After the PCG DB is created, one can then estimate the mean $(\overline{dx}, \overline{dy})$ for a missing spot in some gel g, given the landmark for that gel which is *near* a missing spot. Using the EXTRAPOLATE command, this is done for each Rspot set j by first finding the mean $(dx, dy)_j$ for gels in that set which pass the prefilter, and then for each missing gel g estimate the new EP spot position $(X, Y)_{abs\,g,j}$

$$(X, Y)_{abs\,g,j} = (X, Y)_{lm_{g,k}} + (\overline{dx}, \overline{dy})_j. \tag{G.10}$$

The landmark k is arbitrarily selected as the landmark for this Rspot set for which the Rgel occurs. The newly created EP spot is then inserted in the Rspot set. The algorithm is:

ALGORITHM: EXTRAPLOLATE - EXTRAPOLATING SPOTS

```
[1] FOR each Rspot set j DO
    [1.1] FOR each gel g meeting prefilter DO
          [1.1.1] Compute: mean (dx,dy)j of the spot j to the Rgel's LM k.
    [1.2] FOR each gel g missing from Rspot set j DO
          [1.2.1] Estimate: (X,Y)g,j =(X,Y)lm(k,g)+(dx,dy)j.
          [1.2.2] Create eRspot with EP label, (X,Y)gj position, zero
                  density, area values and insert into Rspot set j.
```

Appendix H

GELLAB-II Portable PiXture (PPX) file header format

```
/* ************************************************************** */
/*     ppxfmt.h - portable picture format header specification      */
/*                     D R A F T                                    */
/* ************************************************************** */
/*
 * Original VAX/FORTRAN design by (VRSION=3.3, 24-AUG-1987 16:28:19.22)
 * John Taylor
 * Argonne National Labs, Bld 202
 * Argonne, Ill. 60439
 * Net Address: "TAYLOR%ANBIPM"@ANL.MFENET
 *
 * Converted to C struct by:
 * P. Lemkin
 * National Cancer Institute/FCRDC, Bld 469
 * Frederick Md. 21702
 * Net Address:
 * uucp: uunet.uu.net!ncifcrf!lemkin
 * ARPA: lemkin@ncifcrf.gov
 * BITNET: lemkin%ncifcrf.gov@cunyvm.bitnet
 *
 *
 * Revision Date: March 10, 1988          - PFL
 * Revision Date: November 24, 1987       - PFL
 * Revision Date: November 12, 1987       - PFL
 * Revision Date: October 4, 1987         - PFL
 * Revision Date: September 12, 1987      - PFL
 * Revision Date: September 2, 1987       - PFL
 * Revision Date: August 31, 1987         - PFL
 * Revision Date: August 28, 1987         - PFL
 * Revision Date: August 27, 1987         - PFL
 */

/* The image file header is a single block of 512 bytes [in VAX byte
 * order], which is followed by blocks of image data.  Each line
 * starts on a block boundary.
*/

/*  ============================================================ */
/*                              I M                              */
/*  ============================================================ */

 /* user setable parameters if the header changes */
#define PPX_VRSION 35      /* Code version 3.5 as 35 */
#define WEDGE_PPX 24       /* Max number of wedge steps if applicable*/

typedef struct ppxB24 {unsigned var : 24;} ppxB24_t; /* to make bit arrays */
typedef struct ppxB16 {unsigned var : 16;} ppxB16_t;

typedef union ppx_Node_ {
        struct ppx_sHdr_ {

                    /* (float fversn) image file format version # */
        unsigned fversn : 16;     /* Set to PPX_VRSION by program
```

```
                         * creating header */


unsigned filtyp : 8;      /* type of file.
                           *  = 0 for unknown.
                           *  = 1 for raw image.
                           *  = 2 for processed image.
                           *  = 3 for synthetic image.
                           *  = 4 for spot file.
                           *  = 5 for exception list file.
                           *  = ... 6 to 255 are free.
                           */

unsigned nrows : 16;      /* full image size in pixels */
unsigned ncols : 16;

char name[32];            /* name of the picture - was [16] */
char sid[12];             /* further identification -
                           * (sample ID)
                          */
char vism[12];            /* visualization method */
char sdate[8];            /* date of scan */
char stime[8];            /* time of scan */
char initl[4];            /* initials of person doing scan */
char scsys[20];           /* ID of the scanning system */
char scprog[12];          /* name of scanning program */
char scpvrs[4];           /* version of scanning program */

unsigned nbands : 8;      /* # of scanning bands (colors) */

unsigned bitpp : 8;       /* bits per pixel (one band) */
unsigned bytpp : 8;       /* bytes per pixel (one band) */

          /* (float x0) top left corner of image in cm */
unsigned x0 : 16;         /* TLC of image in cm*1024 */
          /* (float y0) top left corner of image in cm */
unsigned y0 : 16;         /* TLC of image in cm*1024 */


          /* (float sptsz) scanning spot diameter in microns*/
unsigned isptsz : 16;     /* s.s.d in microns*1024 */


          /* (float stpx) step sizes in microns */
unsigned istpx : 16;      /* step sizes in microns*1024 */
          /* (float stpy) step sizes in microns */
unsigned istpy : 16;      /* step sizes in microns*1024 */


unsigned tessl : 32;      /* tesselation code ??? bits??? */

unsigned domain : 8;      /* 0 = optical density,
                           * 1 = transmittance
```

```
                                * 2 = gray value
                                * ... 3 to 255 are free.
                               */

        unsigned rix : 16;      /* top left corner of region
                                * of interest (0,0) is U.L.H.C.
                               */
        unsigned riy : 16;

        unsigned nx : 16;       /* size of region of interest */
        unsigned ny : 16;       /* size of region of interest */

        unsigned blkaln : 8;    /* block alignment flag
                                * (0 = aligned)
                               */
        unsigned trnsl : 8;     /* translation flag
                                * (0 = raw, 1 = done)
                               */
        unsigned nexcpt : 32;   /*  number of exception entries */

                   /* (float odmn) Min optical densities */
        unsigned odmn : 16;     /* Min Pixel value */
                    /* (float odmx) Max optical densities */
        unsigned odmx : 16;     /* Max Pixel value */


/* NOTE: 11-12-87 PFL new fields ---------------------------------- */
        unsigned blackIsZeroFlag : 8; /* 1 if BLACK is gray value 0, else
                                * 0 if WHITE is gray value 0
                                */
        unsigned wedgeType: 8;  /* Type of step wedge if any in the image:
                                * 0 = none
                                * 1 = ND, neutral density
                                * 2 = CPM, counts per minute
                                * 3 = DPM, disintegrations per minute
                                * 4  to 255 = <free>
                               */
        unsigned nWedgeSteps : 8; /* will contain WEDGE_PPX to be set by user*/
        ppxB24_t wedgeVal[WEDGE_PPX]; /* ND*1024 or CPM wedge step values*/
        ppxB16_t grayCalWedge[WEDGE_PPX]; /* gray scale peaks
                                        * calibration corresponding to
                                        * step wedge values.
                                       */

        unsigned cMapFlag : 8;  /* color map flag.
                                * 0 = no color map
                                * 1 = color map is the first 1024
                                *     bytes of data after header
                                *     where 3x256 bytes are Red,
                                *     Green,Blue maps stored as
                                *     sequential arrays.
                                * 2 = same as 1 but image data is
                                *     true-color image consisting of
```

```
                                *     three sequential images (R,G,B).
                                * 3 = same as 2 but pixels are 24-bits
                                *     specifying 8-bits each of (R,G,B).
                                * 4 = color map is first 12,298 bytes
                                *     of data after the header where
                                *     3x4096 (12-bit lookup table entries)
                                *     are stored as sequential arrays.
                               */

        unsigned imOrientation : 8;   /* Image orientation bits
                                * where: Default 00==>011:
                                * 01 = left to right
                                * 02 = bottom to top
                                * 04 = right to left
                                * 010= top to bottom
                               */

        unsigned imEncode : 8;        /* Image encoding method

                                * 0 = none (just raw raster data)
                                * 1 = UNIX 'compress/uncompress'
                                * 2 = run length
                                * 3 = 1D modified Huffman
                                * 4 = 2D modified Huffman
                                * 5 = delta coding
                                * 6 = run length with delta coding
                                * ... 7 to 255 are free.
                               */

        unsigned startDataDict : 32;  /* if non-zero, then byte # of file
                                * where data dictionary starts. This
                                * should be multiple of 512 bytes.
                                */

        unsigned endDataDict : 32;    /* if non-zero, then byte # of file
                                * where data dictionary ends. This
                                * should be multiple of 512 bytes -1.
                                */

        unsigned startImageData : 32; /* if non-zero, then byte # of file
                                * where image data starts. This
                                * should be multiple of 512 bytes.
                                */


/*       NOTE: The last bytes of this header block are used for
 *             certain scanner specific parameters. See the appropriate
 *             scanner include files for details.
 *       NOTE: @8-31-87: The next free byte(s) is im[294]
*/
        unsigned char free[1];    /* next free byte */

        } sHdr;
```

```
        unsigned char im[512];    /* fill out block to 512 bytes */

    } ppxHdr_t;


/* NOTES: [PFL] on possible problems and suggested changes
   -------------------------------------------------------
1. Initially all VAX/FORTRAN REAL*4 and INTEGER*4 were changed to C's
   float and long. Upper case names were changed to lower as is the
   UNIX convention. But that is not the end of the story... see below.

2. 'Float' is not portable across different types of hardware. Eg 32-bit
   VAX float is different from IEEE format and CRAY's 64-bit float is
   yet another case. What is more portable is to recode all floats as
   scaled integers where required - with an adequate # bits for the
   required precision. If the values have fractional parts in
   [0.000-0.999] than can use scaled integers by packing with
   x 1024 and truncating it to an integer. Unpacking is performed by
   multiplying by  x 1.0/1024. 'floats' which were replaced with these
   packed integers are noted as comments.

3. 'Long' (or 'int') may be > 32 bits. Eg. on CRAY int is 64-bits. In
   addition, most fields do not need 32-bits.  'Struct bit-fields' which
   are portable across most C compilers were used. Also note that many
   fields do not require 32-bits and so additional space in the header
   can be recovered.

4. There is a need to include a ND or CPM wedge and its equivalent
   gray scale calibration. For a vidicon WEDGE_PIX=16 is adequate but
   24 would be much better:
       wedgeVal[WEDGE_PPX] holds ND*1024 or CPM wedge step values.
       grayCalWedge[WEDGE_PPX] holds the gray scale peaks calibration
   corresponding to the wedge values. The nWedgeSteps field holds the
   number of active wedge steps (or WEDGE_PPX). The wedgeType field
   describes the type of wedge used.

5. The generic header access 'INTEGER*4 im[128]' in the Fortran header
   was changed to 'unsigned char im[512]' because of the variable size
   of long. Extra header data is thus accessed as bytes not words.

6. Note: the DEC VAX bit and byte order (low byte first) is used in the
   header rather than the 68000 etc byte order (high byte first).

7. Image encoding field 'imEncode' is used to describe the image
   compression method used if any.

8. Added optional color map flag cMapflag which indicates that a RGB
   color map (3x256 bytes) is in the first 1024 bytes of the image data.
   The 1024 is used rather than the actual number to make the image start
   on an even 512 byte boundary. Note that case 2 permits the specification
   of true-color 24-bit images. Case three is where RGB 24-bit pixels are
   sequential rather than three sequential images (as in case 2). Case
```

```
   4 is for three 12-bit function memories for 12-bit gray values.

9. An image orientation code 'imOrientFlag' to describe mirror-image
   or inverted image data.

10. All bit-fields are a multiple of an 8-bit byte to make it more
   portable.

11. The changed size of the image name was changed from name[16] to
   name[32] to make life easier.

12. Note that in C, to access a field of 'hStr' one would use the
   following code:
           ppxHdr_t header;
           ppxHdr_t *ptrToHdr;
           ptrToHdr = (ppxHdr_t *)malloc(sizeof(ppxHdr_t));
            ... header.hStr.nx
or
            ... ptrToHdr->hStr.nx
and to access the entire header by any byte j, use
            ... header.im[j]
or
            ... ptrToHdr->im[j]
To access the jth wedge entry, use the more complicated code since
they are struct arrays
            ... header.sHdr.wedgeVal[j].var
or
            ... ptrToHdr->sHdr.wedgeVal[j].var

The first free byte at the end of the header can be used for storing
other data (such as strings, etc). The # bytes free can also be found.

           sizeFree = (&header.im[511] - &header.sHdr.free[0])
  and      strcpy((char *)&header.hStr.free[0],"add some more stuff");
*/

/* End of ppxfmt.h */
```

Index