10 Microarray Analysis Using the MicroArray Explorer

Peter F. Lemkin, Gregory C. Thornwall, Jai Evans[3]

(1) Laboratory of Experimental and Computational Biology, CCR, NCI-Frederick, Frederick, MD 21702; (2) SAIC, NCI-Frederick; (3) DECA, CIT, NIH-Bethesda.
Revised: 9-25-2002

To appear in: The analysis of gene expression data: methods and software. Parmigiani G, Garrett ES, Irizarry R, Zeger SL (eds), Springer Press, 2003, pp. 229-251.

Abstract

The MicroArray Explorer (MAExplorer) is an open-source Java-based microarray data-mining tool that is available from the open source Web site as both ready-to-run program and source code from `http://maexplorer.sourceforge.net/`. MAExplorer helps analyze expression patterns of individual genes, gene families, and clusters of genes. It is used as a stand-alone Java application and may be used for both ratio and intensity quantified array data (e.g., Cy3/Cy5, Affymetrix, and others). Data-mining sessions may be saved for continuation at later times or shared with collaborators; significant gene subsets, plots, and reports may be saved on the local disk. Extensions, called MAEPlugins, enable users to add new analysis methods and access to new genomic databases as they become available. MAExplorer was implemented in Java so that the same software could run on many platforms (e.g., Windows, MacOS 8/9 and X, Solaris, Linux, and Unix).

10.1 Introduction

The MicroArray Explorer (MAExplorer) is a user-friendly data-mining tool written in Java (Lemkin et al., 2000, 2001). It is available from the open source Web site (`http://maexplorer.sourceforge.net/`) as well as the original site (`http://www.lecb.ncifcrf.gov/MAExplorer`). The site at SourceForge.net was created to encourage collaborative community development of MAExplorer. MAExplorer was initially created to help analyze gene expression patterns of mammary enriched cDNA microarrays developed by one of our collaborators (L. Hennighausen, NIDDK, NIH). Expression patterns are of individual genes, gene families, and clusters of genes. The tool was first developed as a Web applet for the Mammary Genome Anatomy Program (MGAP) Web site (`http://mammary.nih.gov/`) to publish ^{33}P labeled mouse clone expression data from early 1700 clones spotted membranes of normal and knockout breast tissue mouse models. The MAExplorer applet was subsequently converted for use as a stand-alone Java application and its capabilities greatly extended for use with ratio or intensity quantified array data (e.g., Cy3/Cy5, Affymetrix, and others). The stand-alone version has other advantages over the applet: it starts up faster, it makes use of more memory for clustering, data-mining sessions may be saved for continuation at later times or

sharing with collaborators, and significant gene subsets, plots, and reports may be saved on the local disk. MAEPlugins extensions enable addition of new analysis methods and access to new genomic databases as they become available. MAExplorer was implemented in Java so that the same software could run on many platforms (e.g., Windows, MacOS 8/9 and X, Solaris, Linux, and Unix). Figure 10.1 (see color insert) shows the MAExplorer graphical user interface. We welcome developers who would like to contribute to this Open Source project.

10.1.1 Need for the Methodology

Traditionally, researchers have studied the regulation of biochemical and genetic pathways as well as developmental programs on a gene-by-gene basis. The development of large-scale gene expression profiling using DNA microarray chip technology holds promise as a powerful tool to help in the discovery of new genes and biochemical pathways involved in the development and regulation of mammalian cells and diseases such as cancer (DeRisi et al. 1996; Weinstein et al., 1997; Alizadeh et al., 2000; Cooper 2001; Schulze and Downward, 2001; Sorlie et al., 2001). Although there is no simple 1:1 correspondence between mRNA expression and protein expression because of post transcriptional and post translational processing (Ideker et al., 2001), understanding which mRNAs are expressed will help in understanding biological pathways. During the last half decade, the DNA microarray chip has become a major biological research tool. “At one end, the investigator is interested only in finding the single change in gene expression that might be the key to a given alteration in phenotype … At the other extreme, the aim is to look at overall patterns of gene expression in order to understand the architecture of genetic regulatory networks, a global approach that could ultimately lead to complete description of the transcription-control mechanism of a cell” (Schulze and Downward, 2001).

In general, data mining is a collection of exploratory analysis methods for uncovering relevant patterns of interest in data from a particular problem domain (Tukey, 1977; Cleveland, 1985; Tufte, 1997) and through direct manipluation (Schneiderman, 1977). Typically, this involves using various methods, including database, statistical, data-filtering, graphical, and direct-manipulation user interface techniques. In the context of microarrays, the goal is to help identify genes having similar expression patterns associated with particular experimental conditions. Such subsets of genes and their associated patterns may be useful for helping uncover gene functions and genetic pathways (Schena et al., 1995; Strausberg and Austin,1997; Ermolaeva et al., 1998, Eisen et al., 1998; Hughes et al., 2000).

10.1.2 Basic Ideas Behind Approach

MAExplorer is an integrated program that includes a variety of data analysis methods for sample and gene-set management, data normalization, gene-set data filtering, direct-manipulation graphic displays (pseudoarray images, scatterplots, expression-level plots, histograms, cluster plots, clustergrams (i.e., heat maps), and dendrograms), and tabular reports for both genes and sample data. Gene

data in plots and reports are hyperlinked to pop-up Web pages on genomic Internet database servers. This extensive set of features is fully described in the MAExplorer Reference Manual available for online use or download from the Web site. Because of space limitations, we cannot show examples for most of the concepts described in this chapter and will only touch on some of this functionality. Instead, we refer you to the Reference Manual, which has many examples of screen captures illustrating the functionality described in this chapter.

A sample in a MAExplorer database is the quantified data from a hybridized array. A MAExplorer database consists of a directory tree of files that may be generated by various conversion tools. These include: (1) our Cvt2Mae data converter "wizard"; (2) the Web-based NCI–CIT Micro Array Database (mAdb, `http://nciarray.nci.nih.gov/)` at NIH; or (3) data edited manually (e.g., with Microsoft–Excel) into MAExplorer format. The detailed formats are described in Appendices C and D in the Reference Manual. Cvt2Mae translates commercial (Affymetrix, GenePix, Scanalyze, and others) or non-standard academic user-defined chip tab-delimited data files to ready-to-analyze MAExplorer format data files; this surmounts the data conversion problem. For most non-NIH users, using the wizard (1) is the simplest method.

All data files are tab-delimited ASCII text. A database consists of a directory with at least three subdirectories: `Config/, Quant/,` and `MAE/.` The `Config/` directory contains a Gene In Plate Order (GIPO) file mapping genes on the array with position of spots on the array as well as specifying genomic identifiers (such as GenBank, LocusLink, UniGene, and others) for each gene, a samples database listing the hybridized samples, and a configuration file describing the array layout and other information about the data. `Quant/` contains separate quantified `.quant` data files for each hybridized sample with intensity, background, array location and good spot flag data for each spot on the array. MAE/ contains startup or `.mae` files. Additional directories are created as needed and include State/ which holds named gene sets and named sample lists when you save a data-mining session, and Report/, which contains `.txt` or `.gif` files generated when you save tab-delimited text reports or pop-up graphics plots respectively.

New bioinformatic analysis methods may be added to MAExplorer by using a dynamically loadable Java MAEPlugins programming facility. Because new genomic analysis methods and databases are constantly becoming available, this facility lets researchers add new methods and access these databases without having to modify the kernel MAExplorer software.

MAExplorer, Cvt2Mae, MAEPlugins, documentation, and sample data are freely available for download over the Internet from our Web site and are easily installed on common operating systems. We encourage authors of new plug-in methods to make them freely available for the research community either on the `http://maexplorer.sourceforge.net/` Web site or their own site.

10.2 Methods – Statistical and Informatics Basis

MAExplorer consists of methods for manipulating sets of gene data across sets of hybridized experiment samples. The general paradigm is to use gene data filters to define a *working set of genes* — a subset of all genes in the database based on various filter criteria. Having found interesting subsets of genes, the user can then visualize these subsets and make reports of those sets of genes. We now discuss some of the concepts and methods used to implement this paradigm.

Users first select a pre defined database of multiple *hybridized samples* abbreviated as HPs, representing different sample conditions from either local files or Web databases. *Individual samples* may be assigned to separate X and Y variables referred to as HP-X and HP-Y. These are often used for comparing two samples, as, for example, comparing two samples in an HP-X vs. HP-Y scatterplot. For Cy3/Cy5 ratio data, one can compare any two channels between separate HP-X and HP-Y samples or compare Cy3 vs. Cy5 of the same sample. We defined the *current sample* as the last sample HP-X or HP-Y sample selected.

The user may also assign any number of *replicate samples* to *sets* called HP-X and HP-Y "sets." An *ordered list* of samples used for gene expression profiling is called the HP-E "list." and may be assigned by the user. Subsequent analyses then operate on samples in these sets and list. The X and Y sets may be used for pair-wise analyses comparing mean values of replicate samples. For example, if one had replicate controls and replicate tumors, then one could compare the mean values in scatterplots, perform *t*-test gene data filtering, or apply other data filters using mean data.

The simplest database consists of a single sample assigned to HP-X. MAExplorer can report on these data to some extent, although more samples allow additional types of analysis. If that single sample contains Cy3/Cy5 data, you can do some limited analysis such as plotting Cy3 vs. Cy5, data filtering the Cy3/Cy5 ratio, viewing an intensity or ratio histogram, or interrogating individual genes for their ratios, and more. The default startup mode is a single HP-X sample and a single HP-Y sample, even if sample sets were assigned. You can toggle between single and replicate HP-X and HP-Y "sets" of samples using the (Samples | Use HP-X & HP-Y "sets"' else single samples) menu command.

The HP-E ordered list of individual samples is generally assigned using the ordered conditions of a particular project experiment. Typical ordered conditions might be a time series, cell cycle, developmental stages, drug dose-response, and others. The HP-E samples list is used for expression profile plots (EP plots), clustering, and reports. Additional data structures and methods to manipulate ordered lists of condition sets of replicate samples are being added.

The hybridized arrays are scanned and the images quantified into tab-delimited intensity data files using programs such as Axon's GenePix™, Scanalyze, Molecular Dynamics ImageQuant™, Research Genetics' Pathways™, and others as well as Affymetrix software that generates tab-delimited data. Specific microarray image quantification characteristics are

determined by the particular image analysis program being used to process the images into quantified spot data files. These quantified data files must then be converted to a specific set of MAExplorer tab-delimited data files using the Cvt2Mae data conversion wizard tool. Most spot intensity data should be able to be converted for use with MAExplorer.

When comparing data between samples, it is necessary to normalize the data between samples — even those generated under supposedly exactly the same conditions. This is critical because of differences in amount of sample, labeling efficiency, variations in scanner operation, including gain and baseline settings, and other systematic errors. There are a variety of data normalization methods built into MAExplorer. Other methods could be added using a normalization plug-in with the MAEPlugin facility.

All analyses operate on a subset of genes called the *working gene set* defined by the data filter and computed in real-time by the intersection of the results of gene data filter tests selected by the user. The working gene set is then used for plotting, clustering, or reporting. There are a number of predefined gene sets that are fixed for a particular array and include named genes, ESTs, and so forth. There are three special gene sets, including a user-defined *Edited Gene List* (EGL) that may be created or edited manually or that capture the current cluster(s) during clustering. The two other special gene sets are the normalization and user data filter gene sets. Gene sets may be saved in named gene sets and used for data filtering, reports, normalization, or to synthesize new gene sets using Boolean operations. Similarly, the HP-X, -Y, and -E sets and lists of sample conditions may be saved and restored as named sample lists as well as manipulated with Boolean list operations. The current data-mining session state is the collection of all of these named gene sets and sample sets, normalization options, data filter thresholds and options, and other settings. The state may be saved to the disk as a named startup file when the database is saved. Starting MAExplorer on this file at a later time restores the session state at the time it was saved.

Sets of genes or sample condition lists are useful for tracking intermediate results in complex data-mining sequences of analysis operations. For example, derived named gene sets may be used in successive data filters and for reports.

One could do the following experiment given four different types of samples (e.g., virgin, pregnancy, lactation, and involution). First, compare two HP sample sets using a statistical test such as a *t*-test. Then save the resulting set of genes under the name “virgin vs. pregnancy.” Then, compare the next two HP sample sets and save the resulting genes under the name "lactation vs. involution". Finally, compute the difference of genes found in "virgin vs. pregnancy" that are not found in “lactation vs. involution.” This resulting gene set could then be saved (e.g., with the name “Genes found in virgin vs. pregnancy but not in lactation vs. involution”). Similarly, taking the intersection of these two named sets shows genes that are common between the two sets. Taking the union shows genes found in either of the two named sets.

Data reports of sets of genes or lists of samples may be exported to Excel spreadsheets as tab-delimited data by cut & paste or by saving to a text file or as

dynamic spreadsheets with clickable links to external databases such as genomic databases. Clicking on a link displays information for that gene in the related genomic database in a pop-up Web browser. Full resolution plots can be saved as GIF files for documenting results or for publication purposes.

The main MAExplorer window displays a pseudoarray image of the currently selected HP sample, individual HP-X and HP-Y samples, or sets of samples. It generates either an intensity image or a ratio image of various types of data (e.g., Cy3/Cy5, Cy3/Cy3, Cy5/Cy5 from the same sample or from different samples, HP-X/HP-Y of single samples, mean HP-X "set"/mean HP-Y "set" or others). Depending on the pseudoarray mode, different numeric data values are presented when one clicks on a spot in the image. This has the side effect of defining the corresponding gene as the *current gene*. Similarly, clicking on a point that corresponds to a gene in a scatterplot or other type of plot also defines that gene as the current gene. You may alternatively specify the current gene using a pop-up *gene name guesser* window where you can type a gene name or partial name. Changing the current gene in any display causes all of the active displays to be updated with the current data for that gene. Similarly, changing the normalization method or data filters selected or data filter parameters via threshold sliders also causes all active displays to be updated and the working set of genes defined by the data filter to be recomputed. A SaveAs button is available on the various plot and report windows that lets you save the plots as full-resolution GIF files and reports as tab-delimited text files.

Scatterplots are useful for comparing two samples, two mean samples composed of HP-X and HP-Y "sets", and channels from the same ratio (e.g. Cy3, Cy5) data or channels from different HP samples or duplicate spots (denoted F1, F2 as duplicate grid fields) in the sample. Changing the normalization automatically rescales and redraws the scatterplot. You may use scrollbars to zoom in on particular regions of the plot. Histogram plots of ratio and intensity data for individual samples or sets of samples can be generated. Then, a histogram bin can be selected and used to define a ratio range, intensity range, or functions of these ranges for the data filter.

We define the expression profile EP_j of a given gene j as an ordered list of normalized intensity or ratio data values for each sample for that gene. This ordered list of samples used is specified by the HP-E list of samples. Then, the expression profile plot (EP plot) of a given gene is the intensity on the vertical axis and the sample number on the horizontal axis. You may generate EP plots for any selected gene, a scrollable list of EP plots for a set of genes, or an overlay plot of expression profiles for a set of genes. The scrollable list and overlay EP plots are useful for comparing expression profiles in gene sets.

MAExplorer clustering methods all work by clustering genes based on their expression profiles across an HP-E ordered list of samples. Only genes within the working set of genes are clustered. They use the concept of cluster distance d_{ij} between gene i and gene j computed according to a particular distance metric on the expression profile space. Gene–gene similarity s_{ij} is defined as (1.0 - d'_{ij}), where d'_{ij} is d_{ij} scaled to have a range of 0.0 to 1.0. The metrics currently

available are Euclidean distance and Pearson correlation coefficient. The three cluster methods include: (1) clustering by expression profiles most similar to that of the currently selected gene (an example is shown in Figure 10.2; see color insert). This finds the subset of genes whose cluster distance is less than the adjustable cluster distance threshold; (2) *K*-means or *K*-median clustering with a user-adjustable number of clusters *K*; and (3) hierarchical clustering of genes as a function of the HP-E sample expression-list. The latter method results in a clustergram heat map (one gene per row, one HP sample per column), with an optional zoomable dendrogram. Dynamic cluster reports are generated for all of these methods. Gene subsets may be captured from all of these methods for subsequent operations. Sneath and Sokol (1973) describe these and other clustering methods.

10.2.1 Analysis Paradigm

Generally, an investigator has a number of objectives when analyzing a set of data. The types of analyses performed and how useful they are depends on what one wishes to get out of the analyses as well as the type and quality of the data.

Data mining is an exploratory data analysis for uncovering of relevant patterns of interest in data from a particular problem domain (Tukey, 1977). Typically, this activity involves using various statistical techniques to identify the patterns, including cluster analysis. Researchers across a wide range of fields have suggested that a major aspect of this problem is finding the correct means of graphical presentation to allow humans to be a part of the pattern recognition process (Cleveland, 1985; Tufte, 1997). Tufte (1997) argues that the proper display of quantitative data in the context of the problem domain can aid in understanding complex sets of data. This carries over to the analysis of microarrays using statistical and graphical methods as well as quantitative and genomic knowledge databases. Jagota describes a number of these methods and applications for microarray data analysis and visualization (Jagota, 2001).

To avoid biasing the results, one should approach the analysis of a set of data with minimal expectations of what the patterns might be. However, some idea of what genes you might be interested in helps focus the search. Beware of the trap of mining the data until you find the patterns you hope for since they may have occurred by chance.

Obviously, this introduction to data mining is a first approximation to what is eventually required to successfully explore one's data. However, it does capture the flavor of the data-mining process. Typically, user's would refine their search using variations of the data filters and might contrast results using Boolean operations on gene sets and hybridized sample condition lists found under one set of conditions with those found under other sets of conditions. Figure 10.3 illustrates this iterative process.

The user makes some initial decisions on the experimental design such as the hybridized samples to compare as well as the types and numbers of replicates. A guess can then be made as to the normalization method to use and the gene subset to concentrate on when setting the initial data filter. Additional data filters may be used to remove bad or noisy data. Data are viewed in various

modalities to get a feeling for their inherent dynamic range and where interesting outliers might appear. Clustering and plots may help bring these differences into view. The results are then evaluated and either the process is stopped or the views are refined by adjusting data normalization, filter parameters, data subsets to be investigated, clustering methods, plots and so forth, and the process repeated until the user is able to see the differences between gene subsets more clearly or no significant differences appear to be found.

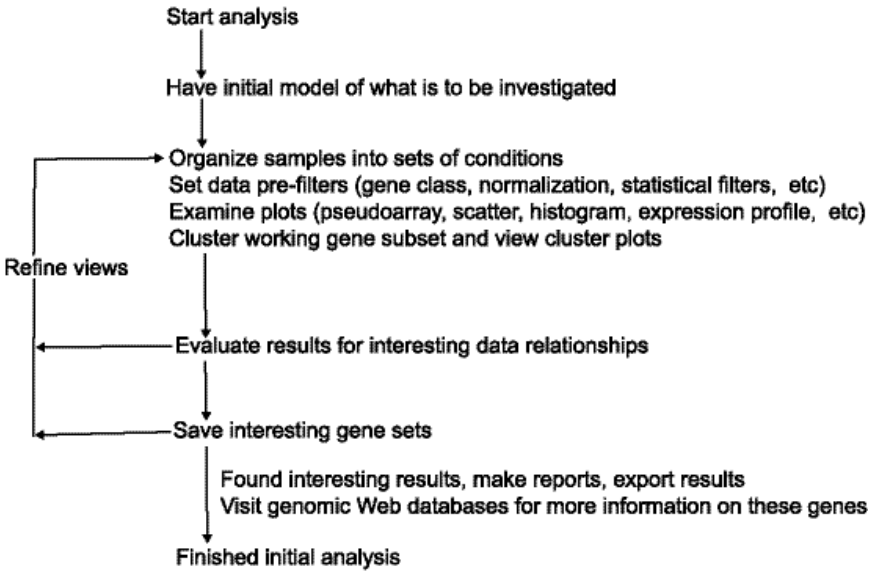

Figure 10.3 Flow chart of a simplified view of the iterative data mining process.

Because data mining is a pattern discovery activity, the researcher should try to make use of all the tools available. It is open-ended because of the variety of ways data may be partitioned, normalized, prefiltered, clustered, and viewed. Patterns that are apparent in one view may not be apparent in another. When data-mining microarray data, look at correlated genes from the point of view of what relationships might be interesting to a biologist when characterizing genes that cluster together. Additional information, such as knowledge of genes belonging to known pathways, may also be useful. Further investigate your results using various NCBI and PubMed database searches on the resulting genes, by designing other lab experiments to better uncover the causation, and other methods. Correlation does not imply cause and effect.

The process above must utilize an appropriate experimental design. Proper experimental design is critical for resolving significant differences in gene expression between experimental conditions and in making valid statistical inferences. Experimental design issues common to microarray experiments are

discussed in Dudoit et al. (2000), Jagota (2001), Kerr and Churchill (2001), and Simon et al. (2001).

If users compared two different sample conditions, the type of analysis used would be different than if they were comparing an ordered sequence of sample conditions (e.g., time series, cell cycle, dose-response, tumor-stage, and so on). MAExplorer gives users the ability to:

1. Organize experiments by sample conditions, where each condition may have one or more replicate samples. If the database has samples with many different conditions, this allows users to perform pairwise comparisons of all of these condition sets.
2. Reduce the set of all genes on the array to a working set of genes that pass the selected data filters. This process may be thought of as a data pre-filter. There are three main categories of filters: set membership, data ranges, and statistical tests.
3. Cluster the working gene set using various clustering methods, possibly generating additional gene subsets corresponding to particular clusters.
4. Explore, compare, and record these gene sets using direct manipulation graphics and spreadsheet reports to gain different insights into the underlying relationships of the data.
5. Save various gene sets during this analysis for subsequent analysis, documentation, and the accessing of public Internet genomic databases.

Because of the iterative nature of this analysis process, it is important to keep a record of the analysis steps you used or the measurements made during this process. MAExplorer lets you record command history and measurements in log files.

10.2.2 Particular Analysis Methods

The primary way that the user invokes analysis steps is by selecting commands from pull-down menus. Since all of the menus are described in detail in Section 2 of the Reference Manual, we will not repeat that level of detail here but rather summarize their functionality.

10.2.3 Data conversion

One of the problems in analyzing data from a variety of systems is the need to handle different formats. When the microarray community adopts the final MGED-ML MIAME standard data format, we will adopt it as well. Until then, to use MAExplorer with your array data, you must convert your data files into MAExplorer input format. Although you can do this by manually editing your data files into the required format (see Reference Manual), it is a nontrivial process. Therefore, we developed a Java "wizard" conversion tool called Cvt2Mae to automate data conversion. It handles most commercial and nonstandard array data formats. For user-defined chips or special exceptions to common formats, the wizard Q&A lets you describe or edit the specification for

your chip and experiment project. We call this description the *array layout*. Part of this process is the specification of which fields of interest are defined in the GIPO (Gene-In-Plate-Order or Print) file, and which fields contain the quantified intensity data. After you have defined the chip layout, you can save it for use in future conversions or share it with a collaborator. After you have created the array layout and specified the input files to be converted, you run the converter and it generates the proper set of converted data files for MAExplorer.

10.3 Software

We give an overview of MAExplorer functionality by summarizing the menus. Menus contain commands (such as to pop-up a scatterplot) as well as checkboxes and ratio buttons to toggle the status of a command modifier (e.g., use sets of X, Y samples rather than individual samples in computations). Menus can also contain submenus where the organization requires it.

Files Menu — Select Database to Be Analyzed

The File menu includes commands for providing access to database data from disk and Web servers as well as for saving the data-mining session. The user may select the database subset to be loaded from either a Web server or a local file system. When you open an existing database, you do so by specifying a `.mae` startup file. As was mentioned in the introduction, the initial database includes a `.mae` startup file created prior to running MAExplorer. If you are on a system where programs are invoked by clicking on a data file (e.g., in MS Windows, MAExplorer is associated with files ending in `.mae`), then you may start MAExplorer directly by just clicking on the startup file. If not, then you can open the `.mae` file using the (File | Databases | Open disk DB) menu command.

Samples Menu — Selecting Sets of Data Sample Conditions

The Samples menu includes commands to select the current hybridized sample or samples to be analyzed. There is a pop-up sample chooser that lets you graphically assign samples to the HP-X “set”, HP-Y “set”, and HP-E “list” of hybridized samples. You may also manipulate these sample sets and lists using menu lists of samples or through a wild-card name-guesser interface. Cy3 and Cy5 channel data may be swapped on an individual sample basis. This menu contains the preference for treating X and Y sample data as individual samples or as sets for computation and graphic display purposes.

Edit Menu — Edit EGL, Gene Sets, Condition Lists, and Preferences

The Edit menu includes operations to modify the Edited Gene List (EGL). You may also define and edit named gene sets and named sample condition lists using Boolean operations to create new sets and lists. Various user preferences such as the database title, X and Y sample set names, and others may be set in this menu. It is sometimes useful to preset threshold parameters used in data filters and clustering. Therefore, a window of threshold scrollbars may be

popped up to adjust these preferences to their desired range. A user-defined named gene set may be used as an additional data filter or normalization gene set or be assigned to the EGL. You may define named gene sets for particular gene set ontologies using the pop-up *gene name guesser* and gene set Boolean operations in the Edit menu.

Analysis Menu — Organized as First Approximation of an Analysis

The Analysis menus is an ordered list of six primary menus that may be used sequentially to perform an initial analysis. In practice, all analyses will be much more complex, but it is a useful way to think about the tasks that would be required. In more complex analyses, the sequence of operations will vary and include commands selected from other menus. The analysis tasks are as follows:

1. *GeneClass* — define the primary subset of genes to be analyzed
2. *Normalization* — select initial normalization method
3. *Filter* — select gene data filters and adjust parameters for these filters
4. *Plot* — view various graphic representations of the data
5. *Cluster* — cluster genes by expression profile data
6. *Report* — generate dynamic or exportable reports

Gene Class Menu — Restrict Database to a Primary Subset of Genes

This menu lets you specify the current gene class from a set of built-in gene sets that are characteristic of your array and are extracted from the GIPO data. The data filter can be enabled to test for gene class membership. The *gene class* is a specific subset of genes (e.g., all genes, all named genes, ESTs, ESTs similar to named genes, all genes and ESTs, and so on). Some of the other built-in gene sets are good genes, replicate genes, housekeeping genes, calibration DNA, your plates, or empty wells, depending on their availability in a specific database.

Normalization Menu — Scale Data so Samples Are Comparable

This menu includes operations to normalize raw gene channel intensity data between hybridized samples by applying the transform to all spots within each sample in the database. Currently, each sample is normalized by itself, not taking other samples into account. The built-in methods do not currently take the possibly nonlinear dye response to intensity or gene sequence into account. These could be handled by creating MAEPlugin normalization methods to take these factors into account.

The current normalization methods that transform raw intensity include *Z*-score of intensity, median intensity, $\log_{10}$ median intensity, *Z*-score $\log_{10}$ intensity using either standard deviation or mean absolute deviation in its calcuation, by the sum of calibration DNA genes (if any), by the sum of genes in a user-defined normalization gene set, by scaling to 65K/maximum intensity, and unnormalized. In addition to the normalization method, you may adjust the raw data in other ways prior to performing the normalization. If background data are included in your database, you may subtract the background value.

Filter Menu — Find a Subset of Genes Meeting Data Filter Criteria

The working set of genes computed by the data filter is the intersection of gene sets passing the selected tests. Tests are grouped into three main types with any number of tests being allowed for any of these groups. They include: 1) membership in particular sets of genes, 2) testing whether data are in specified intensity or ratio ranges, and 3) statistical or clustering tests. Many of these data filters require a threshold parameter, and will automatically pop-up parameter sliders for you to adjust if required. You then interactively set the thresholds using visual feedback from the active displays and reports. This process of interactively adjusting threshold parameters is one of the tools used in direct manipulation.

Gene set filters include filtering by membership one or more of the following sets: gene class, user defined gene set, EGL, global *good genes list*, genes with replicates on the same array. If you created ratio or intensity histograms, you can select bins to automatically generate ranges of data to be filtered. Data may be restricted to only positive data (after background subtraction). Several filters let you set channel intensity, sample intensity, and ratio ranges by threshold sliders on a various sublists of samples. The coefficient of variation of intensity may be used in a thresholded filter computed over various sublists of samples. The *p*-value computed on a *t*-test between X and Y sets, and the absolute difference between mean X and Y samples are other filters. You can also filter by the highest or lowest X/Y ratios of individual samples or sets.

Plot Menus — Display Graphical Representations of the Data

The Plot menu lets you specify the pseudoarray image in the main MAExplorer window. Scatterplots, ratio and intensity histograms, and expression profile plots are generated in separate pop-up windows. Depending on the particular plot, multiple instances may be allowed.

The Show Microarray submenu sets the pseudoarray for the current samples presented as either a grayscale or pseudo-color image. The grayscale image represents spot intensity. Several types of pseudo-color images may be used, including a *sum* of HP-X (red) and HP-Y (green) that generates a red-yellow-green image. A similar display may be generated for Cy3 (green) + Cy5 (red). An alternate display generates a ratio X/Y or Cy3/Cy5 image by scaling values >1.0 as red, values near 1.0 as black, and values <1.0 as green. Where appropriate, the *p*-Value for all spots can be displayed as a pseudo-color image when comparing HP-X and HP-Y sets of samples.

Depending on the origin of the array data, a pseudoarray may have the same geometric verisimilitude as the original arrays. If there is no grid, row, or column data associated with array data, they are displayed as a generated pseudoarray image containing grids, rows, and columns computed to fit the window for visualization purposes. It will not have the same geometry as the original array image. However, the pseudoarrays may be useful to getting a rough idea of the global changes in the data between arrays and how many genes

pass the data filter. When performing gene clustering, clusters are reported as blue circles or squares drawn as overlays on the pseudoarray with the size of the overlay being proportional to similarity. If you are doing *K*-means clustering, the cluster currently selected is displayed in the scatterplot with the number of genes in the cluster proportional to the circle size, and so forth.

Scatterplots may be generated for X vs. Y data, mean X vs. Y "set" data, or Cy3 vs. Cy5 data. You may compare different permutations of Cy3 and Cy5 channel data between different X and Y samples. Cy3/Cy5 plots are available if the data exist in your particular database. That might be the case with replicate spots or with Cy3/Cy5 data. You may zoom into any area of the scatterplot using the horizontal and vertical scrollbars on the left and right edges of the scatterplot. If there are duplicate spots (F1,F2) for each gene on a sample, you may plot F1 vs. F2 intensity. For each scatterplot, the correlation coefficient for the filtered data is displayed in the plot. Data plotted are the intensity values of channels, sample, or mean samples using the current normalization method. The plot scales are changed when the normalization method changes. Clicking on spots in an array image or points in scatterplots sets the current gene and will bring up data on the gene or (optionally) access corresponding data from a genomic database in a pop-up Web browser.

Intensity histograms may be generated for the current sample, and ratio histograms may be generated for the mean X/Y or X/Y "set" data, or Cy3/Cy5 data. The histogram plots are active, letting you select a histogram bin and then use it to define data filters for all genes as a symmetric function (around a ratio of 1.0) of the range specified by that bin (=, <, >, <>, ><).

Expression profile plots (*EP plots*) may be generated for either an individual gene or a list of genes using the ordered HP-E list of samples. In the latter case, it generates a scrollable list of EP plots that may alternatively be presented as an overlay plot. These plots are active and may be zoomed and interrogated for the expression data of a particular gene and sample.

By clicking on a spot (i.e., gene) in the pseudoarray image or on a point (i.e., gene) in the scatterplot, that gene is defined as the *current gene* that is used in other operations. The current gene is indicated by a yellow circle in the pseudoarray image and a green circle in the scatterplot. Similarly, you may add (remove) genes from the EGL from either the pseudoarray image or the scatterplot by clicking on a gene with the Control (Shift) key press. When viewing is enabled, it overlays those genes in the EGL with magenta squares.

Cluster Menu — Cluster Genes Into Similar Groups

Cluster analysis finds a subset or subsets of similar genes based on expression-profile similarity measures across the HP-E list of samples. The three methods perform various types of gene clustering operations on the working gene set. When you invoke a clustering operation, it will pop-up a *cluster report* window and may modify the pseudoarray image using overlay graphics indicating which genes are part of the cluster(s). Direct-manipulation sliders let you specify the number of clusters (*K*-means or *K*-median) or a similarity distance. Changing these parameters recomputes the cluster(s) each time they are changed. The

current gene is used as the *seed* gene for the similar genes and *K*-means methods. The center of the clusters is replaced by the gene closest to the centroid and the entire data set is then reclustered. Silhouette plots are generated for similar gene and *K*-means clustering. When doing *K*-means clustering, selecting the current gene defines all genes in its cluster as the *current cluster*. The gene clusters may be saved as named gene sets. The clustergram (i.e., heat map) and dendrogram analysis dynamic plots are used with the hierarchical clustering and may be interrogated, and gene subsets may be selected. These graphic plots show genes belonging to particular clusters or genes that cluster well with specified genes. Additional lists of EP plots, mean EP plots, and reports including statistics on clusters may be generated. For all of these methods, you may select the intergene distance metric as either the Euclidean distance or correlation coefficient. Various weighting, scaling, and normalization options are available for the hierarchical clustering method.

Reports Menu — Generate Dynamic or Exportable Summary Reports

Tabular reports summarizing gene or sample data may be generated and appear in pop-up windows. These include: data on all of the hybridized samples in the database, which may include links to related Web databases; hybridized sample calibration; Samples vs. Samples correlation coefficient of filtered genes, and mean and variance tables. You can generate reports of all named genes, genes in the EGL, genes in the current gene class, genes passing the data filter, or the N genes with the highest or lowest ratios (X/Y, Cy3/Cy5, or F1/F2) of the filtered genes. Additional features may be added to the data filtered gene reports, including expression-profile data values, X vs.Y "set" statistics, and correlation coefficient statistics. Report tables are presented as either (a) an active *dynamic* (i.e., clickable) spreadsheet that may contain links to genomic Web sites and that pops up a Web browser window to that site or (b) a scrollable tab-delimited text window that may be cut and then pasted into an Microsoft Excel-type spreadsheet for further analysis. Clicking on a column name in the dynamic report will sort the report by data in the report in ascending and descending order. If using the tab-delimited format, you may save the table into a text file.

View Menu — Adjust Views in Plots, Reports, and Genomic DB Access

The View menu options are used to modify the view of genes visible in the pseudoarray image and other displays. Genes may be displayed with additional properties or capabilities, including access to Web-based genomic database entries for specific genes if those database identifiers are available in your data and you are connected to the Internet.

Plugins Menu — Add New User-Defined Analysis Methods

Users can add new analysis methods to MAExplorer using Java plug-ins we call MAEPlugins. These plug-ins can include those written by us, collaborators, the research community, or commercial groups. If you have created a Java `.jar` MAEPlugin file, you may load it at run time using a (Plugin | Load Plugin) menu command that adds the plug-in command to the appropriate menu in the

MAExplorer menu tree. You may also remove unwanted plug-ins if they are no longer needed to save memory or when debugging a MAEPlugin. When you save a data-mining session, the names of plug-ins you were using are saved and are reloaded when you restart MAExplorer. Figure 10.4 shows the MAEPlugin design used to add new analysis methods to MAExplorer. Additional analysis methods are written as Java MAEPlugins and can be 100% portable Java, Java stubs to connect to other local programs such as the R statistics package, or Java stubs to perform client–server access to genomic Internet databases. For example, a new cluster method plug-in called *Cluster* might be added and then invoked from the Cluster menu. MAEPlugins may be loaded at startup time or during an analysis as needed. Once loaded and started, they access MAExplorer data using the Open Java API (Application Programming Interface described on the Web site under MAEPlugins and Javadocs) to request data be fetched or saved.

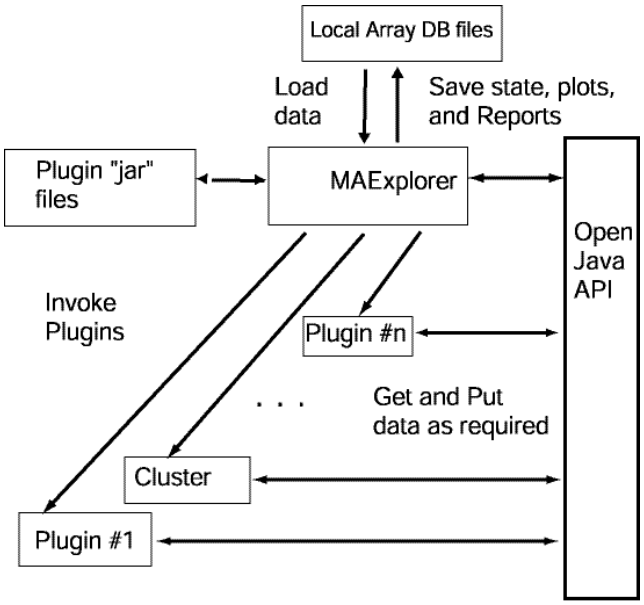

Figure 10.4 The MAEPlugin design is enables adding new analysis methods to MAExplorer. Plug-ins access MAExplorer data through an Open Java API (Application Programming Interface). For example, the box marked Cluster might be a new cluster method plug-in. It accesses data through the Open Java API which in turn accesses data in MAExplorer.

Help Menu — Online Help in Pop-Up Web Browsers

Online documentation is available if you are connected to the Internet. This appears in a separate pop-up Web browser window so you may view it while working with MAExplorer. The documents include the Reference Manual, tutorials, menu overview, index, glossary, and other information. If you are setting up a database, you may include links to other Web pages describing key information on your databases.

10.3.1 System Design — Software Implementation

MAExplorer was written in Java to make it available as a Web applet, although its use as an applet has been deprecated. Java as an implementation language has an advantage over many other languages in that it is highly portable between operating systems as well as being reasonably efficient. For this reason, the stand-alone application was also written in Java. Users have the option of including a Java Virtual Machine (JVM) with the downloaded program. Since we know that the JVM is current, we are able to ensure that MAExplorer runs correctly.

The primary design concept is that the base program should contain the minimum set of core functionalies needed to do a basic analysis. Since no software can contain all possible analysis methods, which is not desirable in any case because of the complexity of learning and maintaining such a system, we added Java plug-ins to allow users to intelligently extend MAExplorer's functionality.

MAExplorer was constructed using a number of fundamental data objects, including genes, hybridized samples (arrays), tables, plots, and so on organized using an object-oriented methodology enforced by Java. Sets of genes are implemented as bit sets for efficiency in both storage and set-theoretic operations. With a set being implemented as 64 bits/word, a set intersection, union or difference can be performed on 64 genes in parallel in one logical (i.e., AND, OR, XOR) computer instruction. This makes the data filter quite efficient when computing the intersection of many gene sets. When ordered gene lists are required, memory and computationally intensive lists are used, but only when needed. Tab-delimited ASCII is used as the basic I/O file type for all types of data. This simplifies I/O and allows data to be prepared with a variety of systems, including Excel, array quantification programs, and relational database systems.

Another major decision was to use multiple pop-up windows for 2D plots, histograms, expression profiles, clustergrams, reports, and dialog boxes rather than sharing a single window. These windows are maintained by a special pop-up registry class that handles many of the bookkeeping chores involved with tracking and updating multiple windows viewing the same underlying data. Whenever an event occurs that may change the set of data-filtered genes, the current gene, or the current cluster set of genes, the registry is notified. Some of the events are: the current gene changed, the data filter was recomputed, the filter parameters changed, the sample labels changed, and the normalization method changed. The pop-up registry in turn invokes methods in all registered

active windows (plots, tables, reports, MAEPlugins) to make the actual updates for windows that requested notification for particular events. This object-oriented design greatly simplifies the process of synchronizing the various data presentations with changes in the database and makes it easier to implement a direct manipulation.

We use direct manipulation methodology rather than a strictly command or menu-driven paradigm because of the intuitive way users interact with their data — they want to "grab" it and manipulate it. This meant that users should be able to select genes and samples by clicking on their representations in various ways. Scrollbars are used for adjusting zoom views as well as setting parameters for data filtering. These interactions let users select the data visually rather than having to enter specific values without the visual feedback.

Because data mining is a complex process, possibly extending over multiple sessions, the use of *a data-mining state* was introduced to help users keep track of intermediate results. We associated the state with the startup file so restarting the database on a startup file restores the state to where it was when it was last saved to that file.

Extendable Analyses via MAEPlugins Modules

More information on MAEPlugins is available on the MAExplorer Web site which includes documentation on the Open Java API, open-source Java code examples to serve as a basis for you to create your own plug-ins, our plug-ins and donated plug-ins, and links to plug-ins at other Web sites. Typical plug-ins may include new methods for normalization, data filters, PCA, clustering, client–server, Web-server functional analysis of cluster results, and more. The MAExplorer Open Java API allows users to access all data structures without having to understand the low-level details of the system. Specialized application interfacing classes called MJA classes were written for the plug-in developer to call. These and a special MaeJavaAPI Java class let plug-in developers access all of the internal MAExplorer data structures in a protected manner. This has the added advantage of allowing us to improve and change the internal data structures without causing future problems with user-written plug-ins using those internal data structures directly. Figure 10.4 shows the top-level plug-in design.

10.3.2 How to Download the Software

The distribution of executable and source files on our MAExplorer home page Web site may be freely downloaded for use by academic or commercial interests and may be redistributed without restriction under the Mozilla1.1 license (available on the Web site). The ready-to-run installation download includes the `MAExplorer.jar` file, a set of 50 hybridized sample MGAP data for use as a demonstration database, an optional JVM, and other startup support files. You may also download a "slimmed-down" version that does not include the JVM. The latter may be downloaded separately and should be installed in the same directory where you installed MAExplorer. These files are packaged using the commercial InstallAnywhere™ packaging software by ZeroG which provides

simple download installations for a wide variety of computers. If you do not have a recent JVM or are having problems with the program on your computer, you might want to initially download the full version with the JVM to install the JVM. You can later download slimmed-down versions for subsequent releases and omit the JVM. Note that the downloaded JVM is used only with MAExplorer and does not overwrite your existing JVM if it exists (except for MacOS 8/9). InstallAnywhere generates a program called `MAExplorer.exe` (Windows), `MAExplorer.bin` (MacOS, Linux, Solaris, Unix, and others) that may be used to start MAExplorer. For some systems, it will make files ending with `.mae` (the startup files) clickable so it starts MAExplorer on that particular file.

Open Source Web Site `http://maexplorer.sourceforge.net/`

The Open Source Web site gives full access to downloads for MAExplorer, examples of MAEPlugins, and the Cvt2Mae data converter wizard. You may download ready-to-run installers, Java source files, or Java `.jar` files (for subsequent updates). In addition, many documents are available, including the hyperlinked Reference Manual (both online and downloadable versions as well as Adobe PDF format), tutorials, and PDF and Microsoft PowerPoint versions of training slide shows. A history of revisions reflecting ongoing changes in the status of MAExplorer is constantly updated. To help elucidate the Open Java API for MAExplorer, "javadocs" are available online showing various levels of detail from writing plug-ins to the entire core system. Information on and examples of writing plug-ins are also available. We encourage the research community to write MAEPlugins implementing functionality that is missing that they would like to see and, if possible, make that plug-in available on the Open Source Web site or provide a link to their Web site. The original site (`http://www.lecb.ncifcrf.gov/MAExplorer`) contains a mirror of most of the materials (but no source code) as well as older documentation and the MGAP database. You can reach all of these data through either the SourceForge Web site or the LECB/NCI Web site.

10.3.3 Strengths and Weaknesses of the Approach

There are a range of approaches for performing data mining of microarray data over the Internet between 100% servercentric and 100% clientcentric methods. However, both methods assume rapid access to underlying databases and the ability to transform data from one presentation mode to another where differences might be easily observed. One extreme is the servercentric model using CGI, servlets, or lightweight applets in a Web browser (see Vilo et al., Chapter 6, this volume) as an example of a server-based system with minimal browser requirements). This assumes that all data search and analysis is performed on a back-end server and graphic or tabular results from the server are sent back to the researcher over the Internet. The servercentric model has the advantage of keeping all user data up-to-date but the disadvantage of performing all computations and graphics generation on the back-end server. Relying on the server for major computations and graphics generation may result in significant

delays if the networks or servers are heavily loaded. The other extreme is the clientcentric model, which may include stand-alone programs or data-intensive applets. Here all of the data being analyzed are copied to a user's computer and then computationally expensive analyses are done there. This has the disadvantage for the user of possibly not having the most up-to-date data to analyze unless the copy is kept up-to-date. However, it does distribute the computational load, allowing more effective data-mining with many alternate views and avoiding excessive delays during a data mining session (see Gentleman and Carey, Chapter 2, this volume) for an example of a client-based system using methods written for the R statistical package in a stand-alone environment). In both the stand-alone application and the Web browser applet, data need to be downloaded to MAExplorer. A major difference between these paradigms is being able to read and write data locally. Such data might include array database caching, state and report saving, and the data-mining state.

Table 10.1 Comparison of clientcentric versus servercentric methods for data mining comparing some of the features of clientcentric and servercentric data-mining analysis methods. The clientcentric approach presented here primarily uses Java with data downloaded to the client's computer. A servercentric approach might use a mix of HTML, CGI programs, servlets and Java. However, even a clientcentric approach may take advantage of server support for additional functionality (e.g., accessing genomic servers to gain additional information about specific genes or sets of genes). Advantages and disadvantages are indicated with a + and - respectively.

Clientcentric	
+	Java runs on all operating systems as either stand-alone or browser applets
+	handles rapid response required for direct manipulation on desktop computers
+	stand-alone version may be restarted quickly from local or cached data
+	size limitations are not a problem with stand-alone Java applications
+	Java plug-ins allow prototyping new analysis methods by any group of users
+	easy to build large, stable stand-alone programs handling very large data sets
-	for applet version, slow startup since program and data downloaded when run
-	difficult to build large stable Web applets handling very large datasets
-	for stand-alone application, must be installed on client's computer
Servercentric	
+	may have better resources for very large datasets but with dependence on server
+	faster startup than full applet since minimal GUI required and little data downloaded
+	easier to prototype and distribute new functionality using centralized CGI or servlets
-	susceptible to Internet traffic bandwidth problems for large numbers of users
-	susceptible to server-load dependencies for large numbers of users
-	difficult to get very rapid response for direct manipulation for data mining

A good intersection of the servercentric and clientcentric methods is to distribute the computation and data to the systems where they can be handled most

effectively. Because Java enables computation in a Web browser, PCs currently available have enormous power, large memory and high-speed Internet connections are readily available, and it is now possible to distribute some of the data and computations to the desktop. If high-speed direct manipulation methodology is to be made available on the Internet for microarray data mining, then it must be brought to the user's desktop browser or local computer rather than residing solely on the back-end server. This is the approach taken in designing the MAExplorer. Table 10.1 summarizes this comparison of the two approaches.

10.4 Applications

Tutorials for Learning How to Use MAExplorer

The Reference Manual has a short tutorial in Appendix A and a more advanced tutorial in Appendix B. There are also many examples and screen captures illustrating the commands in the Reference Manual and in PDF slide shows on the Web site. The tutorials show the sequences of menu and other commands used to get a rudimentary understanding of MAExplorer in performing various aspects of an analysis. They also suggest examples of different analysis methods to try. The tutorials were written in a generic way to be used with any database. When you install the full version of MAExplorer, it includes a subset of the MGAP database data set that could be used with the tutorials. The MGAP data are also available as a separate download from the MAExplorer Web site. If you have access to other data, you can use those data. As with all tutorials, they are only starting points for getting you started — in this case into understanding the MAExplorer data mining analysis environment.

A MAExplorer database is started using a `.mae` startup file generated when the database is created using the Cvt2Mae converter or other method. If your operating system lets you start programs by clicking on a particular type of file (e.g., a `.doc` file in Windows will start Microsoft Word), then clicking on a `.mae` file will start MAExplorer on that database. Alternatively, you can start MAExplorer with no data and then specify the `.mae` startup file from File menu. Note that the MGAP database array data distributed with MAExplorer includes a number of `.mae` startup files described in Appendix D.6 of the Reference Manual. In the following example, we show how to start this for MS Windows. For other operating systems, see the Reference Manual.

1. To start MAExplorer after you have installed it, go to the Windows Start Menu and click on MAExplorer or click on the MAExplorer startup icon. If it is not in your Start Menu, you can go to where you installed it (typically `C:\Program Files\MAExplorer`) and click on `MAExplorer.exe`.
2. Then, after it starts, go to the File menu, select the Databases submenu, and finally select the Open disk DB command. It will pop-up a file browser to let you select the startup file you want. Alternatively, you may

launch the file directly by going to the list of startup files in the `C:\Program Files\MAExplorer\MAE` folder and selecting a startup file.

When MAExplorer starts, the main pseudoarray window will pop up. When it is ready for you to begin interaction, the menu bar becomes active and it displays a green *Ready – click on a gene to query database* message. You might print the tutorial and then read the instructions from the printout rather than trying to keep this window visible.

Tutorial Notation

The following example is from the self-guided tutorial (you issue the commands). The notation go to A:B:C means go pull-down menu A, then submenu B, and then make selection C. Selecting a gene from the microarray image or scatterplot means clicking on a spot in the pseudoarray image or a point in any of the plots. We show a few examples from the tutorial to illustrate this notation.

A.3.1.5 Filter by expression ratio between two conditions X and Y

Step 1: Go to Analysis : Plot : Histograms : HP-X/HP-Y.
The histogram shows the ratios of the data passing the data filter.
Step 2: Move the pop-up plot so you can see it and the array simultaneously.
Step 3: Choose (click on) a ratio bin. Genes filtered by the ratio range of the bin will have white circles as they passed the data filter.
Step 4: Click on a different bin in the histogram to select another bin.
Step 5: Click on the word "Freq" in the histogram to remove the histogram bin filter.

Note: if the signal is close to background the X/Y ratio is probably incorrect. You can filter out low intensity genes by filtering by spot intensity.

A.3.1.6 Filter by spot intensity range

Step 1: Go to Analysis: Filter : Filter by spot intensity [SI1: SI2] sliders : Use spot intensity [SI1: SI2] sliders.
Step 2: Adjust intensity lower bound (SI1) to remove low ratio genes.
Step 3: When done, remove the “Filter by intensity sliders” by toggling it off (redo step 1 to toggle it off).
Step 4: Repeat steps 1–3, but this time use Filter : Filter by [I1:I2] sliders : Use spot intensity (or Cy3/Cy5) [I1:I2] sliders.

A.3.1.7 Multiple conditions — expression profile plots of HP-E data

Step 1: Go to Analysis : Plot: Expression profile : Display a gene's expression profile for HP-E.
Step 2: After the expression profile window pops up, click on a gene in the array to see its profile .
Step 3: Click on a line in the profile plot to see its intensity.

Step 4: Click on a different gene in the array to see its profile.
Step 5: Press "Show HPs" button to see the list of samples used.
Step 6: Press "Close" button to remove pop-up windows.

10.5 Discussion

MAExplorer is a flexible microarray data-mining tool running on the user's computer. It uses direct manipulation, data filtering, built-in graphics, statistics, clustering, and gene and sample set operations, with results presented as reports. It has tools for managing multiple samples, replicates, gene sets, and expression profile lists. The data-mining exploration state may be saved any time during a session and restored at a subsequent session. Genomic Internet databases may be accessed for genes of interest in pop-up Web browsers. The Cvt2Mae "wizard" tool may be used to convert commercial and academic chip array data for use with MAExplorer. Extensibility for new analytic methods may be provided using MAEPlugins extensions.

Acknowledgments. We wish to thank the members of Lothar Hennighausen's Laboratory of Genetics and Physiology (NIDDK), who inspired the initial development of MAExplorer and its continued development. Thanks also to many others for useful discussions and suggestions that have helped improve the MAExplorer's capabilities and usability. We also want to thank Tom Schneider, Eric Shen, and Ellen Burchill for useful comments on this chapter.

References

Alizadeh A, Eisen MB, Davis RE, Ma C, Lossos IS, Rosenwald A, Boldrick JC, Sabet H, Tran T, Yu X, Powell JI, Yang L, Marti GE, Moore T, Hudson Jr J, Lu L, Lewis DB, Tibshirani R, Sherlock G, Chan WC, Greiner TC, Weissenburger DD, Armitage JO, Warnke R, Levy R, Wilson W, Grever MR, Byrd JHC, Botstein D, Brown PO, Staudt LM (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511.

Cleveland WS (1985). *The Elements of Graphing Data.* Wadsworth Press: Monterey, CA.

Cooper CS (2001). Applications of microarray technology in breast cancer research - Review. *Breast Cancer Research*, 3:158–175.

DeRisi J, Penland L, Brown PO, Bittner ML, Meltzer PS, Ray M, Chen Y, Su YA, Trent JM. (1996). Use of a cDNA microarray to analyze gene expression patterns in human cancer. *Nature Genetics* 14:457–460.

Dudoit S, Yang YH, Callow MJ, Speed TP (2000). *Statistical methods for identifying differentially expressed genes in replicated microarray experiments.*

Technical Report No. TR-578 (August 2000). Department of Statistics, Stanford University: Stanford, CA.

Eisen MB, Spellman PT, Brown, PO, Botstein D (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences USA*, 95:14863–14868.

Ermolaeva O, Rastogi M, Pruitt KD, Schuler GD, Bittner, Chen Y, Simon R, Meltzer P, Trent JM, Boguski MS (1998). Data management and analysis for gene expression arrays. *Nature Genetics*, 20:19–23.

Hughes TR, Marton MJ, Jones AR, Roberts CJ, Stoughton R, Armour CD, Bennett HA, Coffey E, Dai H, He YD, Kidd MJ, King AM, Meyer MR, SladeD, Lum PY (2000). Functional discovery via a compendium of expression profiles. *Cell*, 102:109–126.

Ideker T, Thorsson V, Ranish JA, Christmas R, Buhler J, Eng JK, Bumgarner R, Goodlet DR, Aebersold R, Hood L (2001). Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science*, 292:929–934.

Jagota A (2001). *Microarray Data Analysis and Visualization*. Bioinformatics By The Bay Press, Santa Cruz, CA.

Kerr MK, Churchill GA (2001). Statistical design and the analysis of gene expression microarray data. *Genetical Research*, 77:123–128.

Lemkin PF, Thornwall GC, Walton K, Hennighausen L (2000). The Microarray Explorer tool for data mining of cDNA microarrays — application for the mammary gland. *Nucleic Acids Research*, 20:4452–4459.

Lemkin PF, Thornwall GC, Hennighausen L (2001). MicroArray Explorer — A Java-based tool for data mining microarrays. AMS-IMS-SIAM Summer Conference on Statistics in Functional Genomics, June 10–14, 2001. (`http://www.lecb.ncifcrf.gov/MAExplorer/PDF/AMS-IMS-SIAM-Web-9-28-2001.pdf`).

Schena M, Shalon D, Davis RW, Brown PO (1995). Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270:467–470.

Schneiderman B (1997). *Designing the Human Interface*, 3rd ed. Addison–Wesley: New York.

Schulze A, Downward J (2001). Navigating gene expression using microarrays — a technology review. *Nature Cell Biology*, 3: E190–E195.

Simon R, Radmacher MD, Dobbin K (2001*). Design of Studies Using DNA Microarrays*. Technical Report #4.NCI, BRB: Rockville, MD.

Sneath PHA, Sokol RR (1973). *Numerical Taxonomy*. W.H. Freeman and Co: San Francisco.

Sorlie T, Perou CM, Tibshirani R, Aas T, Geisler S, Johnsen H, Hastie T, Eisen MB,van de Rijn M, Jeffrey SS, Thorsen T, Quist H, Matesse JC, Brown PO, Botstein D, Lonning PE, Borresen-Dale A-L (2001). Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proceedings of the National Academy of Sciences USA*, 98:10869–10874.

Strausberg RL, Austin MJF (1997). Functional genomics: Technological challenges and opportunities. *Physiological Genomics*, 1:25–32.

Tufte E (1997). *Visual Explanations. Images and Quantities, Evidence and Narrative*. Graphics Press: Cheshire, CT.

Tukey J (1977). *Exploratory Data Analysis.* Addison–Wesley: Reading, MA.

Weinstein JN, Myers TG , O'Conner PM, Friend SH, Fornace AJ, Kohn KW, Fojo T, Bates SE, Rubinstein LV, Anderson NL, Buolamwini JK, van Osdol WW, Monks AP, Scudiero DA, Sausville EA, Zaharevitz DW, Bunow B, Viswanadhan VN, Johynson GS, Wittes RE, Paull KD (1997). An Information-Intensive Approach to the Molecular Pharmacology of Cancer. *Science*, 275:343–349.

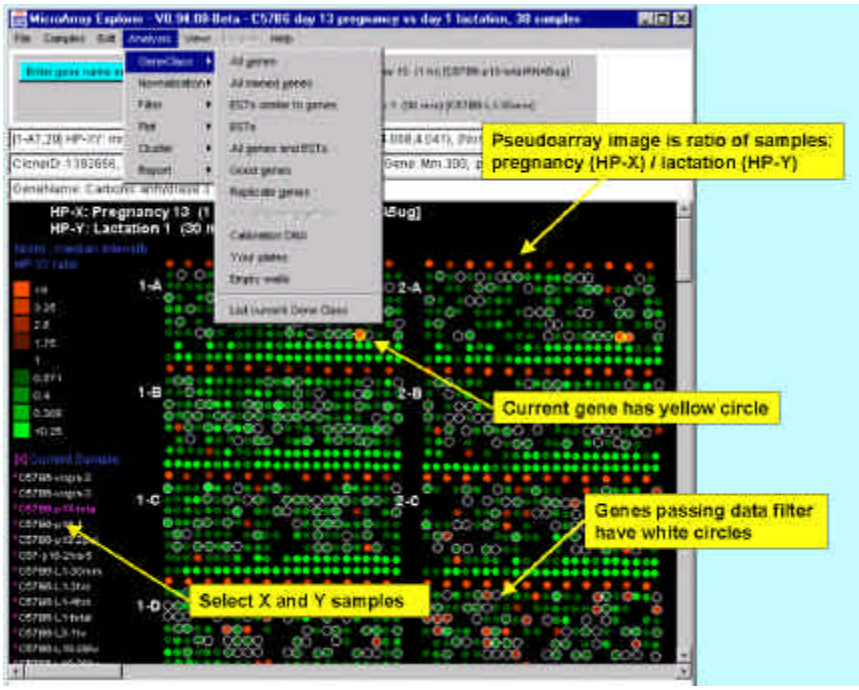

Figure 10.1 The main MAExplorer graphical user interface window showing some of the pull down menus, the pseudoarray image (from MGAP duplicate spotted membranes i.e. Grid 1-A is duplicated in Grid 2-A), and selectable individual samples on the left. The current gene is selected by clicking on spots in the pseudoarray, points in plots, or reports. Genes passing the data filter are indicated with white circles. The user interacts directly with the program through pull-down menus, by adjusting parameter sliders, and by clicking on spots in the pseudoarray, points in plots, cells in reports.Various pop-up scatterplots, histograms, expression plots, cluster plots, dendrograms and clustergrams, and reports may be generated. The set of all genes is restricted to a *working-genes* subset computed by the data filter. Adjusting any of the *threshold state* scroll bars causes the data filter to be recomputed. Any number of orthogonal data filters may be combined to isolate a potentially significant subset of genes.

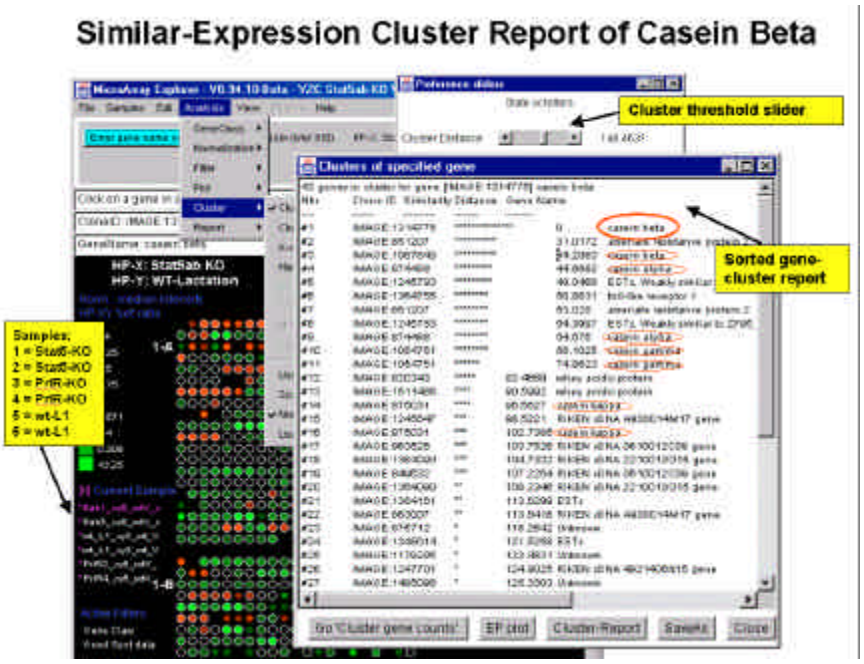

Figure 10.2 Example of expression profile similarity clustering of genes similar to Casein Beta (red circle) in a MAExplorer data-mining session on mouse mammary data. The data consists of 6 Cy5-labeled hybridizations (transplanted Stat5ab-KO, transplanted PrlR-KO, and wt-Lactation day 1) using wt-Virgin day 1 as the Cy3-labeled reference sample (unpublished, Hennighausen etal). The cluster distance threshold slider was adjusted to limit the number of genes passing the similar cluster filter resulting in 43 genes. Passing genes are also saved in the Edited Gene List set that could be saved in a named gene set or further analyzed. Genes failing the cluster threshold distance test are not reported. These are presented in a dynamic cluster report of these genes sorted by similarity to the "seed" gene being tested. Selecting a new seed gene or adjusting the cluster threshold will re-compute the cluster. Other useful options (not shown) include generating a cluster report (Cluster-Report), creating a popup list of expression profile plots (EP plot), and saving the cluster report in a tab-delimited text file (SaveAs).