Software Design of the MicroArray Explorer Data Mining Tool

Home: http://www.lecb.ncifcrf.gov/MAExplorer

Open Source: http://MAExplorer.SourceForge.net/

P.F. Lemkin
LECB, CCR, NCI/FCRDC
mail: lemkin@ncifcrf.gov

[This document is under construction]

Revised: 05-16-2002

Java MAExplorer program design issues

1. Overview
2. Design decisions, Client-centric vs. Server-Centric
3. MAExplorer GUI (graphical user interface)
4. Web database I/O
5. Genes and Gene lists
6. Gene data filter
7. Multiple plot windows
8. Other plot window implementations
9. Reports & access to other Web databases
10. Synchronizing windows
11. Dumping text and plot windows to .txt and .gif files
12. Saving and Restoring the MAExplorer state
13. Miscellaneous classes
14. MAEPlugin design
15. MaeJavaAPI design
16. Examples (links): MAExplorer menu organization

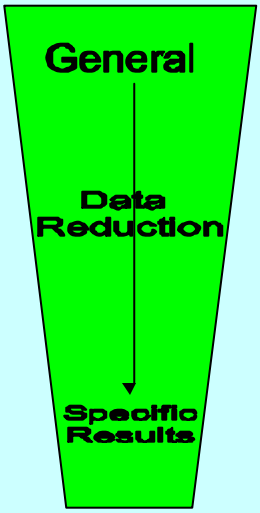
Contents of this document

- This document discusses the software design of MAExplorer
- MAExplorer is Open Source with the source code and a collaborative environment for improving the code available at http://MAExplorer.SourceForge.net/
- The first part contains the primary design discussion
- The second part contains Examples of computer screens for many of the windows illustrating these data structures and classes

1. Overview - MicroArray Explorer

- **MAExplorer is a flexible Java microarray data-mining tool**
- Handles multiple cDNA or oligo array samples
- Handles duplicate spots/array and replicate samples
- Handles intensity or ratio (Cy3/Cy5) quantified array data
- Data organized by 2- (X vs Y) and N-condition lists expression profiles, sample lists, data structures could support ordered lists of condition lists
- Gene data-filters gene set computed by statistics, clustering, gene sets
- Direct data manipulation in pseudoarray image, graphics, spreadsheets
- Access genomic Web servers from plots and reports
- Oriented toward mRNA data, could extend to protein arrays
- Stand-alone (off-line) or applet (Web-based)
- User data converted using Cvt2Mae "wizard" tool
- MAEPlugins allow users to add new analysis methods

1.1 Array data mining - finding patterns of genes



Quantified array spot data for multiple samples

Organize by: sample, gene expression, gene sets

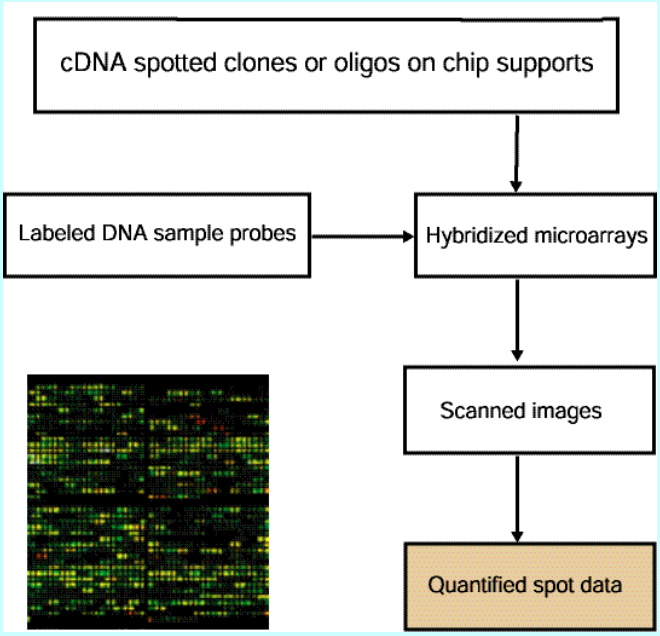
Change views: normalization, data filters

Visualize and query: plots, cluster, reports

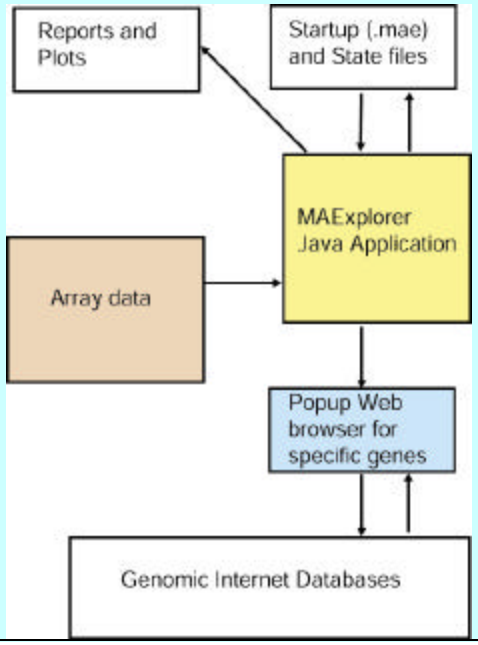
Explore: external genomic databases

Subset of genes for further analysis

1.2 Data preparation for MAExplorer

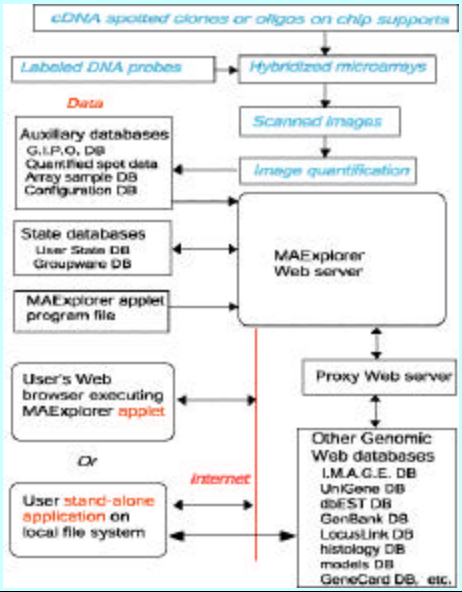


1.3 Paradigm: local & genomic-Web databases



1.4 Overview of MAExplorer database system

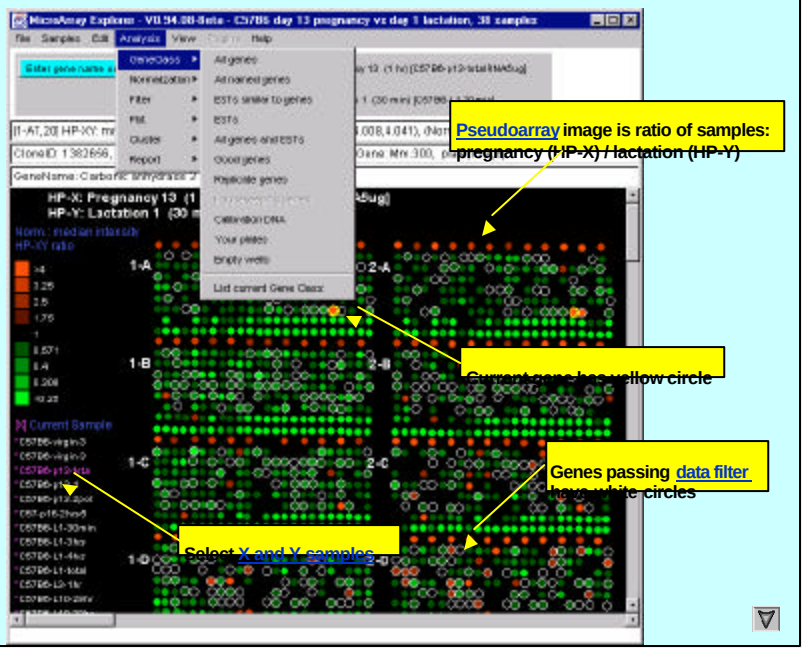
(Data preparation steps prior to MAExplorer analysis are in cyan)



1.5 MAExplorer analysis environment

- Stand-alone Java application for user data
- Download MAExplorer program from Web site
- Installers: Windows, MacOS/-X, Solaris, Linux, Unix, etc.
- Documentation, tutorials, MGAP demo database on Web site
- Cvt2Mae "wizard" tool converts user data for use with MAExplorer
- May download data from NCI/CIT mAdb microarray data server RDBMS for direct use with MAExplorer

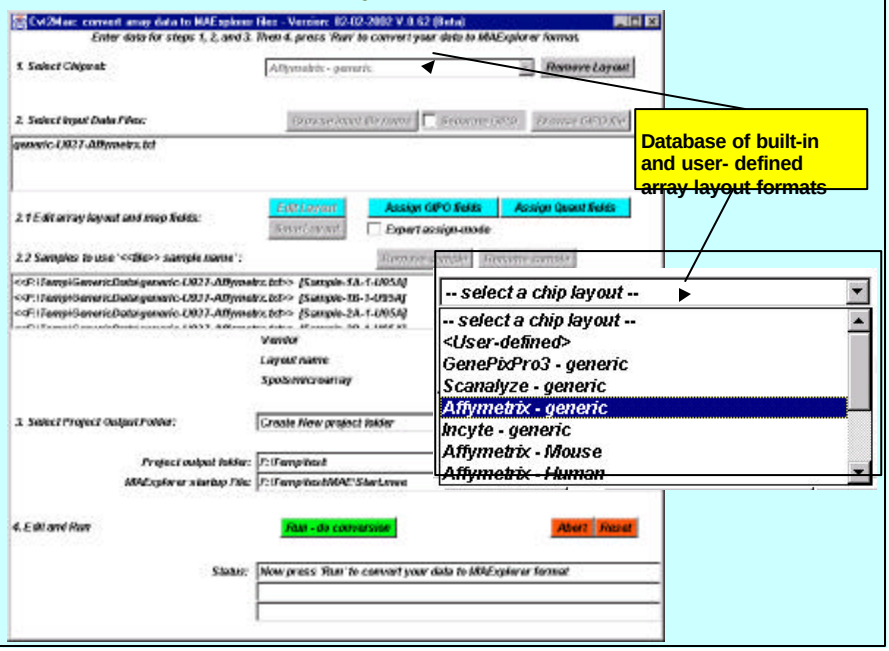
1.6 MAExplorer user interface - main Analysis menu



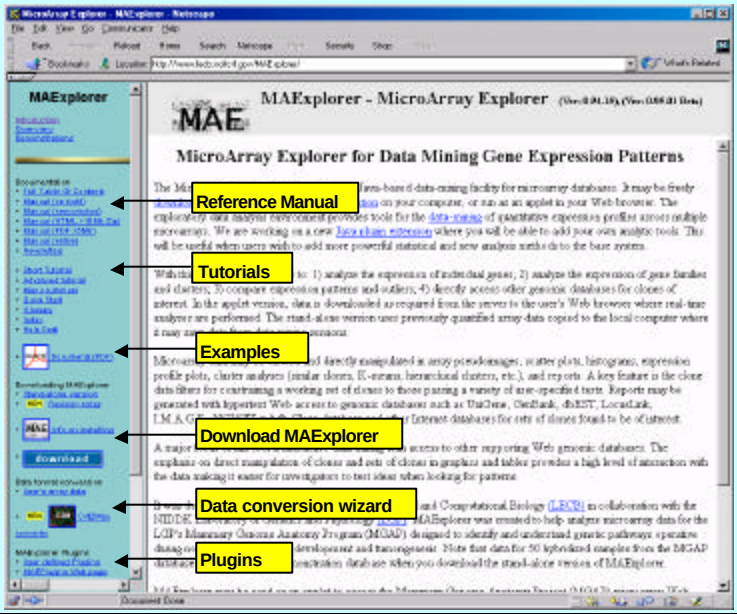
1.7 Data conversion "Wizard" for MAExplorer

- Cvt2Mae "Wizard" converts commercial and user-defined array formats (e.g. Affymetrix, GenePix, Scanalyze, etc)
- Users may create and save Array Layout descriptions for subsequent conversions
- Conversion generates a MAExplorer project directory tree of files that are ready to analyze
- After MAExplorer installed, click on project's *MAE/Start.mae* file to start analysis
- See Cvt2Mae home page for details on operation

1.7.1 Cvt2Mae array data conversion "wizard"



1.8 MAExplorer Home Page
http://www.lecb.ncifcrf.gov/MAExplorer



2. Our design decisions

- Computation is done in the Java application for better user interaction
- Initially used Java JDK1.1, since not all Web browsers handled JDK1.2/1.3. JDK1.1 is the least common denominator for applet portability. Not important with stand-alone version which uses JDK1.3. The JDK1.3 is packaged with the Installers
- All data files (except images) are tab-delimited ASCII files - Excel-compatible. The Cvt2Mae data converter translates user data files to MAExplorer formatted files
- Minimize amount of data that needs to be read initially
- Use object-oriented design with new base classes and extended classes as required. Write custom GUI classes to better control the user interface
- Use Integrated Development Environment such as Sun's free "Forte for Java" for rapid debugging
- Optimize code and garbage-collect data structures often …

2.1 Client-Centric computations - advantages/disadvantages

Client-centric approach uses stand-alone programs

+ 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 allows 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 & data downloaded when run
- difficult to build large stable Web-applets handling very large data sets
- for stand-alone application, must be installed on client's computer

2.2 Server-Centric (CGI or Applet) computations - advantages/disadvantages

Server-centric approach uses mix of HTML, CGI, Java Applets

+ may have better resources for very large data sets but with dependence on server
+ faster startup than full applet since minimal GUI required and little data is 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

2.3 The MAExplorer project

- A database resides in a project directory which contains all samples the user may wish to analyze
- Multiple MAExplorer projects may exist on a local disk (or Web server) - each having a standard project directory tree (shown in next slide)
- A projects database file (maeProjects.txt) in the stand-alone installation directory tracks the names of these projects and disk location and last active database
- The (File | Database | Open file DB) menu command specifies a particular database startup file within a project directory
- MAExplorer is started by opening a .mae startup file that specifies the subset of samples to use (Note: .mae is the file extension - not the complete file name!)
- Clicking on a .mae file (e.g. in Windows) will start MAExplorer on that database

2.3.1 MAExplorer Project Directory Tree

/Cache - (optional) cached data saved from initial download from Web DB server

/Config - project databases for Configuration DB, GIPO DB, Samples DB files

/MAE - set of .mae stand-alone startup files for subsets of the project samples

/Plugins - (optional) set of MAEPlugins written by the user. [Normally, it checks the /Plugins directory in the MAExplorer installation directory]

/Quant - quantified spot data files for each hybridized sample

/Report - (optional) directory where text and GIF files are saved by user with SaveAs Report and SaveAs GIF commands

/State - **(optional)** the GeneBitSets (.cbs) and SampleSets (.hpl) "SaveAs DB" files

2.3.2 Required project directories

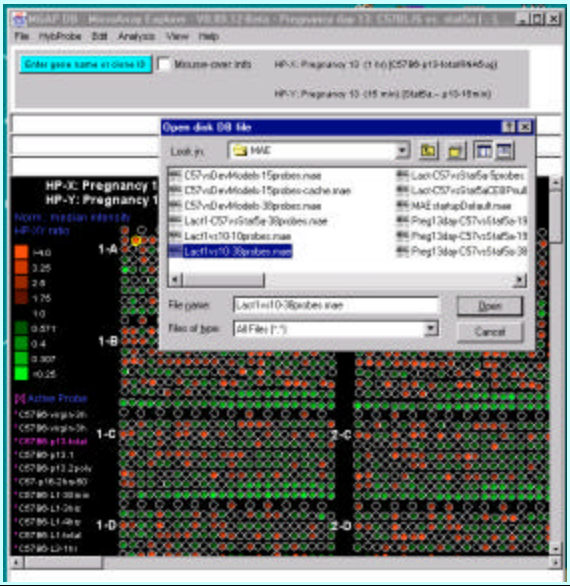
- All data files (except images) are tab-delimited files
- The .mae startup file is a physical file on a local disk or a Web server virtual CGI file in the MAE/ directory
- There may be any number of .mae startup files. They all end with ".mae"
- Each .mae startup file points may point to a specific configuration file
- The Config/ directory contains:
  1. Configuration database file describing the array architecture and other DB files
  2. Samples database file listing all of the samples available in the database
  3. GIPO (Gene In Plate Order) print table database file mapping spot position to geneomic information through a spot identifier
- The Quant/ directory contains quantified spot data files for each hybridized sample that includes channel(s) intensity and background, good spot flags (QualCheck), spot identifier

2.3.3 Optional/Generated project directories

- The Cache/ directory can be used to saving all data files downloaded from a Web server to avoid returning to the server when accessing that data in subsequent data mining sessions
- The State/ directory is used for saving the named gene sets (.cbs files) that are shared between all startup files.
- The State/ directory is also used for saving the named lists of hybridized samples (.hpl) files that are shared between all startup files
- When saving a data mining session using (File | Database | SaveAs … DB) menu command, the named gene sets and sample lists are written to the State/ directory as .cbs and .hpl files
- Saving the database also writes the complete state into a .mae startup file that overides the configuration file data the next time MAExplorer is started
- The Report/ directory is used for saving text reports as .txt files and all plots as .gif files

2.3.4 Opening a database from local disk

In stand-alone mode, you can browse a project database containing a set of startup databases.



2.4 Notation : JDK vs. new MAExplorer classes

- Classes which are part of the SUN JDK library are in **Green**, classes which we wrote specifically for MAExplorer are in **Red**.
- The stand-alone main() method is in MAExplorer
- There are about 140 classes including the MAEPlugins support
- The MAExplorer class contains instances of the single instantiations of all classes that may require global access (for speed) from any subsequent processing
- These global instances are created in *MAExlorer.init()* at startup

2.4.1 New MAExplorer base classes

- Guesser - scrollable text area for selecting one or more items using prefix (e.g. “carbonic”) or wild card (e.g. “*onco*) notation. Includes: PopupGeneGuesser, PopupHPmenuGuesser, PopupProjDirGuesser.
- Chooser - specialized Chooser that lets users move objects from a “remainder” list to a “selected List. Includes: PopupHPChooser for selecting a the HP-X, HP-Y sets and HP-E list.
- Table - tab-delimited table constructor and file reader. Includes: ConfigTable, GipoTable, MaHPquantTable, MaInfoTable, SamplesTable. Table is also used to compute intermediate data structures for computing reports

2.4.1 New MAExplorer base classes (continued)

- Table is extended from SimpleTable that can be used (and is used) to make short-lived temporary tables for various purposes including reports
- Draw2Dplot - create scrollable 2D plots. The DrawScatterPlot, ExprProfileOverlay extend Draw2Dplot
- Gene - create Gene instances. It is used in *GeneList.mList[]* that may contain ordered lists of Genes
- GeneBitSet - creates efficient 64-bit/word bit-sets representing Gene sets and operations on these sets. It is normally accessed through *GeneList.bitSet*

2.4.2 Data structures: genes, hybridized samples

- Any class instance that may need to be accessed from other classes has a class instance kept in MAExplorer. Most classes have an "initialization" constructor that captures the MAExplorer instance (typically called 'mae') for future use by that class. This lets methods in that class access any other classes required
- All variables and methods are private unless they need to be accessed from outside of the class
- State data is found primarily in two classes in **MAExplorer** and **Config**
- Fundamental objects are genes (**Gene**) and hybridized samples (**MaHybridSample**) which are then used in other classes and lists

2.4.3 Data structures: HPs, MIDs, GIDs, gang GIDs

- A HP is all of the array data for a hybridized sample. It contains data for multiple spot intensity channels (e.g. F1&F2, or Cy3&Cy5), background, QualCheck flags, etc.
- A GID is a Grid index ID and uniquely defines a spot in the array database. Corresponding spots in different samples have the same GIDs
- Replicate spotted grid in the array have Gang GIDs
- A MID is a Master gene ID and uniquely defines a gene in the database. All GIDs representing the same gene have the same MID
- There is one copy of a Gene instance in the database and it has all gene specific data (gene name, GenBank ID, Clone ID, etc.)

2.4.4 Data structures: Maps

- The **Map** class defines maps between MIDs, GIDs, GridCoords, and Genes
- The master gene list, *Map.midStaticCL*, has a list of all Genes instances indexed by MID as *Map.midStaticCL.mList[mid]*
- The *Map.gidStaticCL.mList[gid]* accesses corresponding Gene instances by GID
- The *Map.gid2mid[gid]* looks up the MID given the GID
- The *Map.mid2gid[mid]* looks up the GID given the MID
- The *Map.gidToGangGid[gid]* looks up the Gang GID given the GID
- There are other maps between lists of spot **GridCoord** (field,grid,row,column) and GIDs

2.4.5 Data structures: Gene

- The Gene Gene is the base class used to define a single gene (clone or oligo) data structure consisting of sample-specific data fields and sample independent genomic identifiers and name fields. The latter is represented by the Master Gene ID (MID) which is unique for any number of spots for that gene and the Grid coordinate ID (GID) which corresponds to a particular spot for that gene
- The *Gene.midList[0:nMid-1]* is a list of all other Gene instance MIDs that are the same gene (i.e. replicates)
- This identifies replicate genes on the array that are available for computing statistics
- Quantified data may be temporarily stored in (data, data1, data2, pValue, geneDist, etc.) variables
- Generally, F1 (Cy3) is data1, F2 (Cy5) is data2 and F1/F2 or Cy3/Cy5 is data

2.4.5 Data structures: Gene (continued)

- The Genomic IDs include: Clone_ID, GenBankAcc, GenBankAcc3, GenBankAcc5, Unigene_ID, dbEST3, dbEST5, SwissProt, RefSeqID, LocusID.
- The Master_ID is set to one of these.
- The arrays GenomicID[] and nGenomicID[] may be used for specifying external identifiers for particular user databases
- Gene names include: Gene_Name, UGclusterName. The MasterGeneName is set to one of these
- Additional identifiers include: Gene_Class, plate, plate_row, plate_col
- Each gene has various properties indicated the inclusive-or of C_xxxx constants

2.5 Algorithms: initialization and event handling

MAExplorer. init()
1. Read (name,value) parameters from Applet PARAMs or Config file
2. Read database files and set up database structures
3. Create GUI with scrollable pseudoarray image with
ArrayScroller, ScrollableImageCanvas, DrawPseudoImage
4. Create pull-down MenuBar with MenuBarFrame

ScrollableImageCanvas - pseudoarray direct manipulation event handling
1. Select spots invokes PopupRegistry current gene change
2. Select sample invokes PopupRegistry change current HP sample and Filter

EventMenu - pull-down menu event handling
1. Menu item command *EventMenu.handleActions()* - eval menu command
2. Menu checkbox item *EventMenu.handleItemStateChanged()* - eval menu checkbox command

2.6 Startup (name,value) Parameters: .mae file or Applet PARAMs

- The Config class contains many of the state variables (MAExplorer contains most of the rest)
- The parameters are set in the Config class using the GetParam class to get (Name,Value) data definitions if they exist. They are defined in an overide hierarchy:
- 1) Parameters are initially defined by reading the Config file using the ConfigTable class. If they are not defined, then either the variables are not defined or use hardwired values
- 2.a) These are overidden using <APPLET> PARAM values if they exist when using an applet, or
- 2.s) They are overidden using (Name,Value) data from the .mae startup file when used in stand-alone mode

2.7 Data structures: hybridized samples (HP)

- The MaHybridSample class contains the un-normalized data for a particular sample read by MAExplorer. It uses a one-time instance of MaHPquantTable class to read and parse the .quant data file
- The SampleSets class contains the working HP-X, HP-Y and HP-E lists of MaMybridSample instances used by the data Filter. It also contains the menu sample names (parsed from SamplesTable class by StageNames class)
- The HPxyData class contains the the current HP-X and HP-Y sample sets data for a particular gene MID including the statistics for each set for use in computations on a single gene across X-Y samples
- New set statistics are computed for a new MID using HPxyData.updateData()

2.7.1 Data structures: sample condition sets

- The Condition class contains named lists of named hybridized samples
- These may be copied to the SampleSets data structure working lists for the HP-X set, HP-Y set and HP-E list
- It also contains ordered lists of Conditions that could be used with expression lists of averaged samples

2.8 Data structures: spot data for genes

- The SpotData is a data-only class holds the raw and normalized data copied from a single Gene for a single sample
- The SpotData instance is loaded using the *MaHybridSample.get…Data()* methods. There are a number of different methods
- The SpotFeature class methods computes a pretty-print string summary line or 3 lines for data for a single Gene for one HP, HP-(X,-Y), HP-(X,Y) 'sets', or HP-E
- This summary line is modified for single channel (F1 or F2) and (Cy3 or Cy5) intensity data or ratio data (Cy3/Cy5), HP-X/HP-Y etc. taking the normalization mode into account

2.8.1 Data structures: lists of spot data for genes

- Arrays of a specific sample HP (F1,F2) (I.e. Cy3, Cy5) gene spot data passing the data Filter are loaded using the *MaHybridSample.getF1F2data()* method
- Arrays of single HP-(X,Y) samples data passing the data Filter are accessed using the *CompositeDatabase.getHP_XandYdata()* methods
- Arrays of sets of HP-(X,Y) data passing the data Filter are accessed using the *CompositeDatabase.getHP_XandYsetData()* method

2.9 Data structures: expression profile

- The ExprProfile class contains the expression profile data structure for a single gene
- Data may be changed using *ExprProfile.updateData()*
- The samples (HP1, …, HPn) used in the expression profile are defined by the HP-E list of samples with n = |HP-E|
- For intensity data from each sample j (j=1:n) for gene g (g=1:maxGenes) it computes: mean(j,g), stdDev(j,g), CV(j,g)
- Then ExprProfile instances serve as the data structure basis for other expression profile plot classes (ExprProfilePlot, ExprProfilePanel, ExprProfileScrollPane, ExprProfileCanvas, ExprProfileOverlay)

3. MAExplorer GUI: Graphical User Interface

- The initial GUI is created during startup using the MenuBarFrame class that is a separate popup frame
- It creates the pull-down menu created by the class that is an instance of a **MenuBar** may is attached. This lets us use cascading menus for a reasonably easy to use look and feel
- The EventMenu class handles all events from the pull-down menu. It is the main dispatching area for calling the actual functionality to service these events
- Note: The instances of menu checkboxes are defined in MenuBarFrame but are used in EventMenu
- The MenuBarFrame also creates a control panel for buttons and messages at the top
- Messages are written into status areas by the ***Util.showMsg()*** methods

3.1 MAExplorer GUI: status messages and logging

- Messages are written into status areas by the *Util.showMsg()* methods into the three status lines at the top of the GUI
- If command history logging is enabled, then commands (from both the menu selections and clicking on genes etc.) are logged into a popup log window by *Util.saveCmdHistory()*
- If message logging is enabled, then all messages that go to the status lines as are logged into a popup log window by *Util.saveMsgHistory()*
- The command history and message logging text may be saved into log files

3.2 MAExplorer GUI - the pseudoarray image

- The pseudoarray image is recomputed only as needed (e.g. samples makeup or normalization changed) using the DrawPseudoImage class
- The image may or may not represent the actual physical array layout depending on whether the data was available to the configuration database
- Spot color is either grayscale (spot intensity) or a pseudocolor representing the ratio of (F1/F2, Cy3/Cy5, HP-X/HP-Y, HP-X 'set'/HP-Y 'set', p-Value) or the sum of two channels (red+green)
- The pseudo microarray image is then painted in a ScrollableImageCanvas class contained as an instance of the ArrayScroller class in the main frame.
- Whenever the pseudoarray image is redrawn, various overlay graphics are added (e.g. current gene, E.G.L, and cluster boxes or circles, etc) in different colors

3.3 MAExplorer GUI: selecting genes and samples

- Users select genes using direct manipulation by clicking or by typing in pseudoarray image, plots or reports
- Users type sub-strings of names of things (e.g. genes, samples) into a pop up "guesser" window:

  1. The PopupGeneGuesser extends the Guesser base class to select a single gene or a set of genes (into the E.G.L) based on gene name, genomic ID, etc.

  2. The PopupHPChooser uses collections of the ChooserGUI class to select the HP-X, and HP-Y 'sets' and HP-E 'list' from the set of all samples. These sample sets are stored in SampleSet lists of samples

3.5 MAExplorer GUI: state sliders to set thresholds

- The StateScroller class implements a popup window containing the threshold state scrollers
- Only threshold scrollers that are being actively used (i.e. selected in the data filter) are normally presented. This can be overridden to force all threshold sliders to be presented
- Changing a scroller invokes the *PopupRegistry.updateSlider()* that calls all active windows that have registered for a callback when the scroller value changed
- Changing the normalization method changes the state of the scroller since the dynamic range of some threshold will also change.
- The popup window is different from all other MAExplorer popup windows in that you may not close it by clicking on the close window button. Rather, you must deselect all active filters to make it close. However, you may minimize it

3.6 MAExplorer GUI: dialog popups

- There are two types of text dialog popups: a single variable popup PopupDialogQuery class and a triple operand PopupBinOprDialogQuery class
- The PopupDialogQuery class has both a pull-down list of items to select and a text area. Selecting a pull-down item enters it into the text area
- The PopupBinOprDialogQuery class has three instances of the PopupDialogQuery class in its GUI. It is desiged for doing Boolean set or list operations (e.g. dstOperand3= srcOperand1 OPR srcOperand2)
- Gene set operations are done in the GeneClass class
- Hybrized sample set operations are done in the Condition class

3.7 MAExplorer GUI: Menu organization

•**Note: The Analysis Menu organized as a first approximation of an analysis**

1. **File** - database, file access operations
2. **Samples** - select lists of hybridized sample conditions
3. **Edit** - edit gene and condition subsets, E.G.L. and preferences

4. **Analysis** - primary analysis menus
   - 4.1 **GeneClass** - select gene subset for gene class data Filter
   - 4.2 **Normalization** - select gene intensity normalization mode
   - 4.3 **Filter** - select data filters to compute gene subset of interest
   - 4.4 **Plot** - pseudoarray image, scattter, histograms, expression profile popup plots
   - 4.5 **Cluster** - perform cluster analysis on data filtered genes
   - 4.6 **Report** - popup spreadsheet reports of genes & samples

5. **View** - setup genomic gene data views preferences
6. **Plugins** - add and execute new MAEPlugin methods
7. **Help** - popup documentation on MAExplorer and specific DB

4. Database I/O: file and Web I/O

- All I/O goes through a single FileIO class that determines if local disk **file://** or Web **http://** I/O needs is needed
- Startup using either <APPLET> HTML code or .mae startup files
- Web **http://** I/O is read using the JavaCGIbridge class
- All data files are tab-delimited tables read by a Table base class
- Different types of specialized tables are extended from the Table base class (e.g. ConfigTable, SamplesTable, GipoTable, MaHPquantTable, etc.)
- The MaHybridSample base class contains quantitative data used only for input
- Then, a particular project DB subset, is a set of MaHybridSample instances
- Used as an applet, protected projects require a user login validation by a CGI server program - modifying the JavaCGIbridge class

4.1 Expression Data Used in MAExplorer

- **Database configuration data table** for specific array layout and content **
- **Hybridized array samples table** describing their experimental conditions **
- **Gene-In-Plate-Order table** listing Clone Ids, gene names, genomic DB Ids, spot and source plate coordinates **
- **Quantified array spot data table** for samples from quantification software such as GenePix, ScanAnalyze, Research Genetics' Pathways™, Molecular Dynamics' ImageQuant™, etc. **
- **Data is optionally cached** from a microarray Web database server data to the local computer. Future analysis of this data is then independent of the Web database server
- **External Web genomic databases** corresponding to probes and Clone IDs are accessed as needed: I.M.A.G.E, GeneBank, dbEST, UniGene, LocusLink, NCI/CIT mAdb Clone DB, GeneCard, MGAP histology and model DBs, etc.

** Required tab-delimited data files for MAExplorer are indicated in blue

5. Data structures: Genes and gene lists

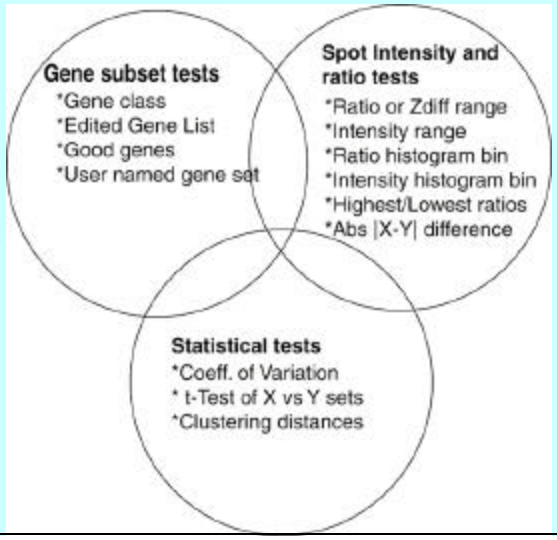
- The basic unit of discourse is the gene. This is defined by the Gene base class. Sets or lists of genes are defined by the GeneList class for named gene lists
- Named gene sets may be saved and used in subsequent set operations
- The GeneBitSet class is a specialized efficient implementation of the Java BitSet class (that it does not use)
- The GeneList always implements the GeneBitSet class - an unordered set
- If an ordered list of genes is required, then an explicit list of Gene[s] is defined in the *GeneList.mList[]*
- Set operations (Union, Intersection, Difference) are implemented using logical 64-bit operations (that are VERY FAST) on GeneBitSet[s].

6.1 The gene data "Filter"

- The gene data "Filter" is the intersection of various tests selected from the menu by the user and is implemented as a Filter class
- If a particular test requires direct manipulation of parameter values, then scrollers are added to a StateScroller class pop up window and removed when the filter is disabled
- The StateScroller window contains a variable number of active adjustment scrollers for all active data Filter test parameters
- The data Filter is run whenever parameters change and resulting gene sets for the active tests are intersected (GeneBitSets) to create the working gene list
- The working gene list is displayed as colored overlays in the pseudo image and scatter plots (and is the *Filter*. *workingCL* GeneList)

6.1 Gene data Filter is intersection of tests

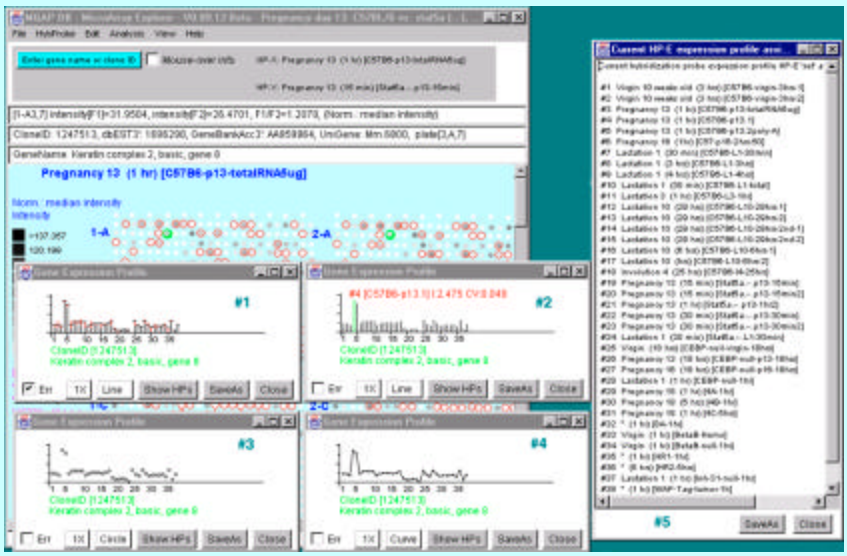
- Working set of genes is **intersection** of gene sets each passing selected filter tests
- Filtered gene subset is used as **pre-filter** for subsequent clustering, plots, and tables
- Changing any filter parameters causes the data filter to be re-computed



7. Multiple pop up windows

- Multiple pop up windows (e.g. Expression profiles, scatter plots in different modalities, etc.) are sometimes allowed so that different interpretations of the data would be available at the same time
- All windows are registered with the PopupRegistry that tracks refreshing (see 10. Synchronizing windows)
- Plots use data from the working gene list. If it changes, then the plots will change.
- The user may select genes or sets of genes from plots
- All plots track both genes and their x,y positions in the plots. This helps the event handlers quickly determine which gene was selected

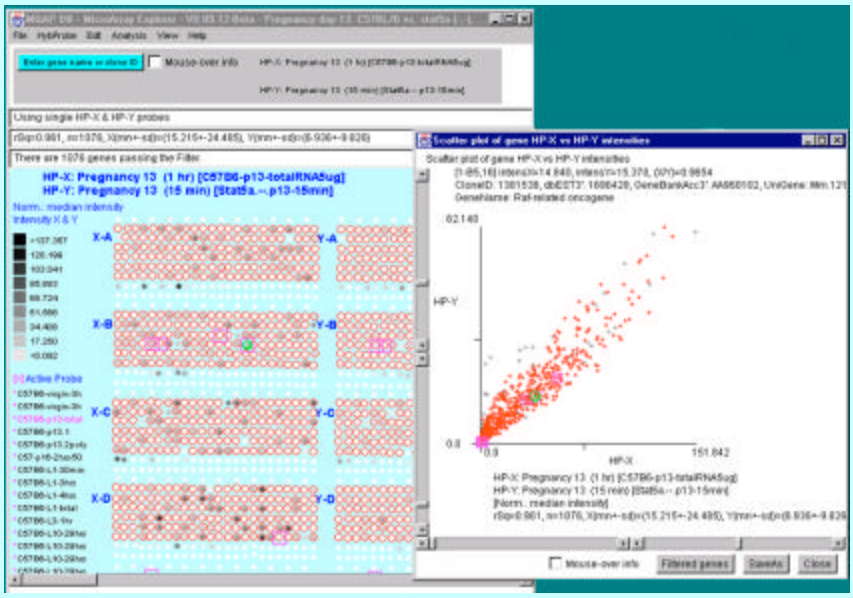
7.1 E.g. Multiple Expression Profile plots



7.2 Two-dimensional plots: scatter plots and expression profile overlays

- The Draw2Dplot class extends Canvas
- The DrawScatterPlot and ExprProfileOverlay classes are extended from Draw2Dplot
- Draw2Dplot class zoom scroll bars compute scale factors used to clip data and scale it to the actual size of the plot window
- Tracks (x,y,gene) for drawn points so event handler can track points clicked on to determine corresponding gene

7.3 E.g. Scatter Plots of Two Conditions

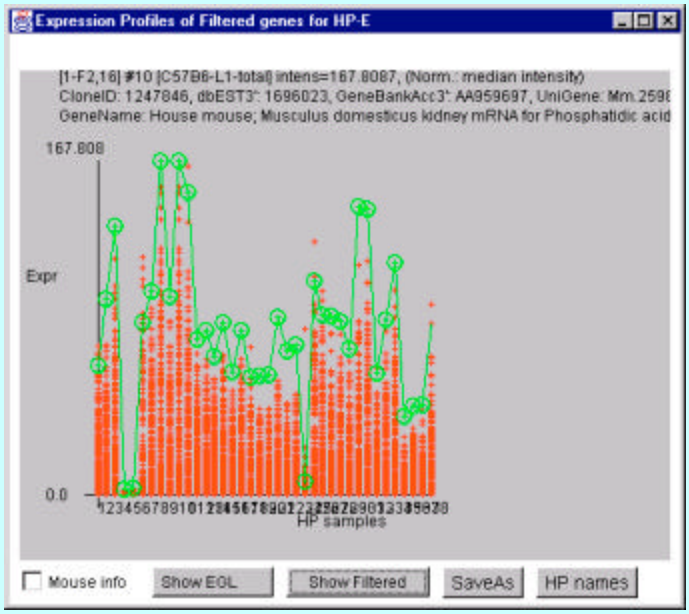


8. Other plot window classes

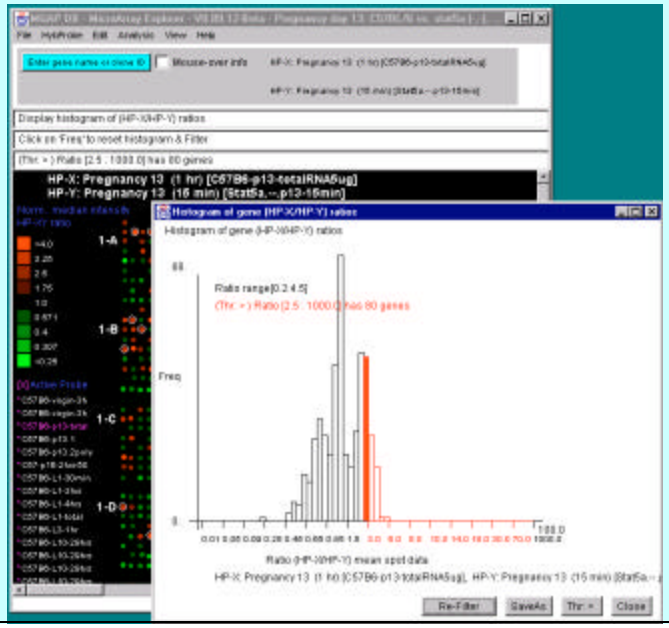
- A scrollable list of expression profile instances is implemented by creating a grid-layout in a ScrollPane that was contained in a pop up Frame
- Then, individual ExprProfilePanel plots are inserted in the ScrollPane grid elements
- The functionality of the ExprProfilePopup, DrawRatioHistogram, DrawHistogram, and ClusterGram (includes dendrograms) classes were so specialized that we created new classes instead of extending the Draw2Dplot class (used for DrawScatterPlot, ExprProfileOverlay)
- The event handlers for the histogram plots track histogram bins and may be used for setting additional tests in the data Filter
- The histogram bin Filter tests track the range of gene intensities or ratios selected for a particular histogram bin

8.1 Scrollable list of Expression Profile plots

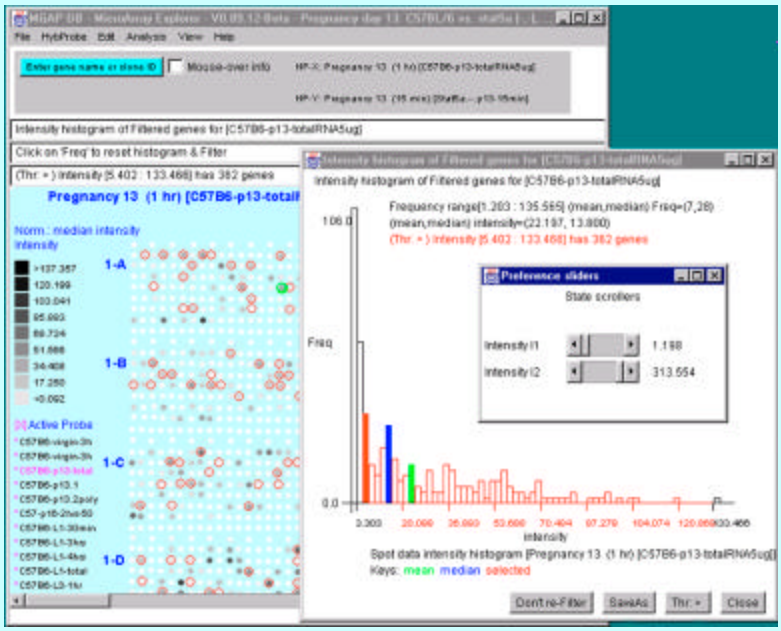
8.2 Expression Profile Overlay plots



8.3 Ratio Histogram plot and data filter



8.4 Intensity Histogram plot and data filter



8.5 ClusterGram and DendroGram plots

9. Reports and access to other Web databases

- Sample and gene Reports are available in Report class instances as either
  1) dynamic selectable spreadsheets or
  2) tab-delimited TextArea's for export via cut & paste to Excel

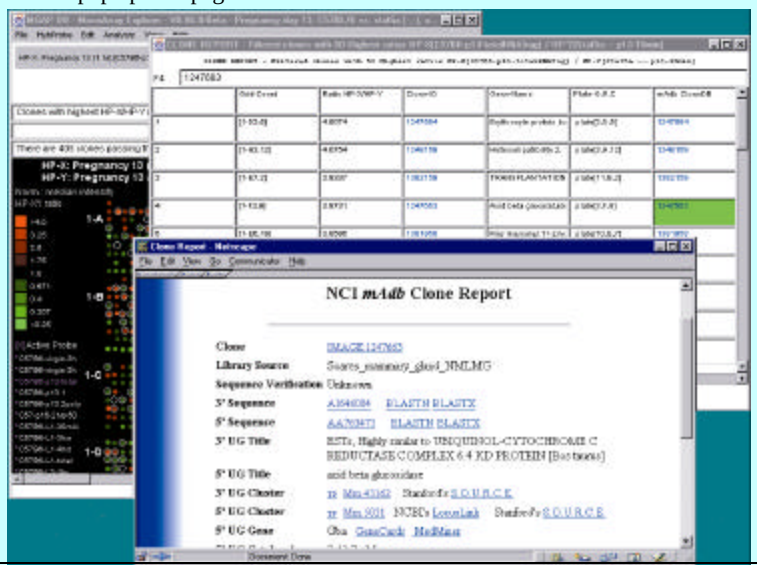
- The dynamic scrollable spreadsheet uses a SpreadSheet class with separate event handlers that enable
  1) scrolling cells,
  2) Web access if enabled using popupViewer() method in Util class,
  3) sorting rows by data in a particular column

- Applet access to other Web genomic DBs uses a proxy server residing on the same Web server as array data. Stand-alone access is done directly

- Web data is requested by selecting a spot in the array, a point in a scatter plot, or a cell in a report. Data is displayed in a new browser window

9.1 Scrollable Dynamic Gene Reports

Scrollable gene report of highest ratio genes & NCI mAdb pop up Web browser page (foreground) of particular gene. Clicking on blue hypertext cell in gene report (middle) invokes pop up web pages



9.2 Tab-delimited report for cut & paste to Excel

Clones clustered with similar expression profiles

9.3 Saving Reports of Results

- Plot Results saved as: GIF image files
- Table Report Results saved as: tab-delimited text files
- Reports:
  1. Web-accessible dynamic spreadsheets or
  2. tab-delimited text exportable to Excel
- Gene set reports - linked to pop-up Web browser reports from genomic Web databases
- Array sample reports
  1. information on samples
  2. pop-up Web browser reports linked to histology and mouse model Web pages,
  3. data filtered sample vs sample correlation coefficients

10. Synchronizing Windows

- Direct manipulation of data in plots and reports is synchronized to highlight the same current gene and Filtered genes if they are changed in any of these windows
- E.g. selecting (clicking on) a gene in the pseudo array image, scatter plots and gene report windows, will change it in the other visible windows
- The PopupRegistry class is used to register all windows when they are created, refresh them when the system state changes (normalization, Filter, current clone, E.G.L, etc.), and remove them when the user is finished with them
- The system state changes if the current gene, current cluster, or Filter changes. The popup registry is called and in turn notifies all relevant plots and reports that the state has changed so they can update themselves if needed

10.1 PopupRegistry plot types and callbacks

- **Types of windows that may be registered with the PopupRegistry:**

1. ShowPlotPopup for plot popups
2. ShowSpreadsheetPopup for dynamic spreadsheet report popups
3. ShowStringPopup for String report popups
4. ShowExprProfilesPopup for expression profile Expression Profile popups
5. MAEPluginUpdateListener for active MAEPlugins
6. Object also for active Popup Window Object's Object

- **Types of callback methods available for registered popup windows**

1. *updateCurGene(mid)* if the current gene has changed to mid
2. *updateFilter(gene list)* if data filter has changed the working gene list
3. *updateSlider()* if any threshold slider has changed
4. *updateLabels()* if any label names (eg. Class names,etc) have changed

11. Saving text & plot windows to .txt & .gif files

- Text popup windows are written out in stand-alone mode as tab-delimited .txt files when servicing SaveAs commands. Data is written using *FileIO.writeFileToDisk()*
- Later, tab-delimited text files may be read into Excel
- The WriteGifEncoder class is used to convert repainted Images to Gif encoded .gif files in stand-alone mode when servicing "SaveAs" commands in plot windows
- All .txt and .gif files saved by the user doing a "SaveAs" command are saved in the **/Report** directory

12. MAExplorer state: Saving and Restoring

- The UpdateState class is used to read and write state information files including: .mae state startup files, .cbs GeneBitSet files, .hpl SampleSet files
- All .mae files are saved in the **/MAE** directory
- All .cbs and .hpl files are saved in the **/State** directory
- The UpdateState.*readMAEstartupFile()* method is used to read the specified .mae startup file. This overides (Name,Value) pairs setup initially through the Config class
- When (File | Databases | SaveAs … DB) is invoked, it writes out the current state with UpdateState.*writeMAEstartupFile()* method. This puts the names of the current .cbs and .hpl files in the .mae file - not the data itself. Then it writes out the .cbs and .hpl files

13. Miscellaneous Classes

- The MathMAE class contains various specialized math functions
- The Statistics class contains various specialized statistics and probability functions (t-test, etc)
- The SortMAE class contains various sorting methods specific for MAExplorer
- The Util class contains various utility methods uses in multiple modules for data structure conversion, display, string conversion, colormap conversion, etc.

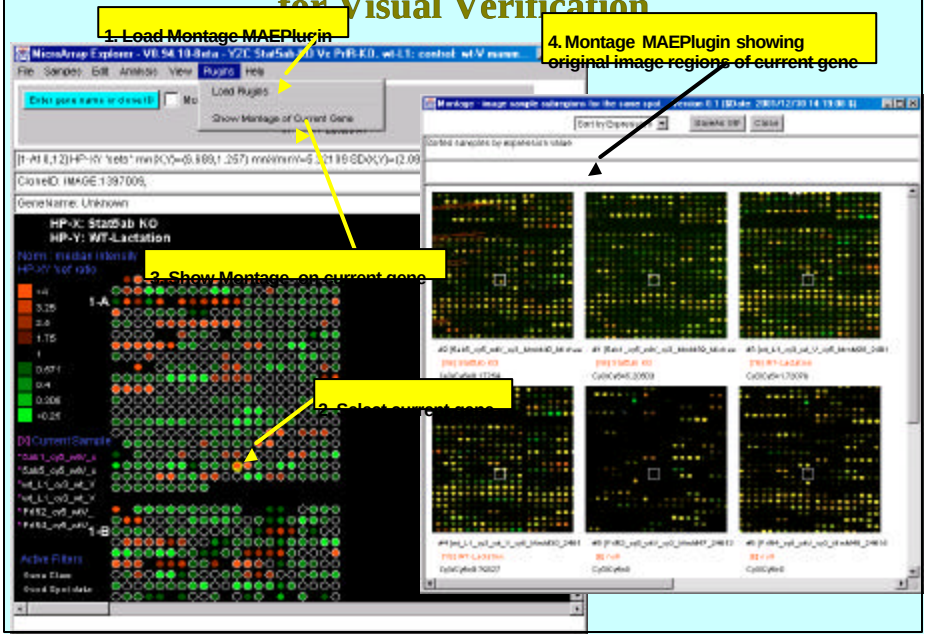
14. MAEPlugin Paradigm

- Plugins are installed and dynamically loaded when MAExplorer starts or when it is running using the MAEPlugin class and package
- Plugins are invoked from pull-down menus
- Client-server plugins may invoke MAExplorer from other programs or vice versa
- Plugins may access any MAExplorer data structures, but in a portable structured way using the MaeJavaAPI and MJAxxxx classes (Open Java API)
- Plugins may provide their own GUI interfaces or save data back into MAExplorer and use it's plot and report capabilities

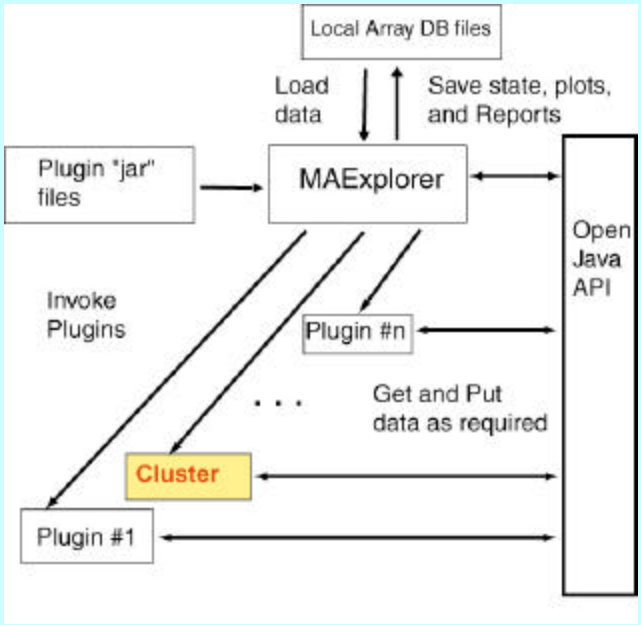
14.1 MAEPlugin Extensions for MAExplorer

- Java plugins allow investigators to extend capabilities of core MAExplorer program to new analysis methods
- Web site contents: Open Java API, Java open-source examples, donated plugins & links will be published and freely available
- MAEPlugin types: normalization, metrics, Filters, PCA, clustering, client-server, functional genomic analysis of cluster results, etc.
- MAEPlugins will have three types of implementations:
  1. Using 100% Java code
  2. Access local programs written in any language (e.g. 'R')
  3. Web-CGI or client-server to specialized genomic DBs

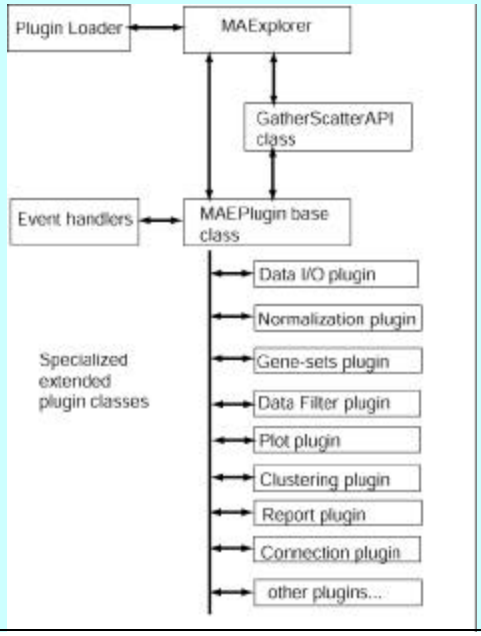
14.2 Montage MAEPlugin of Current Gene for Visual Verification



14.2 MAEPlugin Design



14.3 Open Java API for Writing MAEPlugins



15. MaeJavaAPI Open Java API

- The MaeJavaAPI class implements methods for MAEPlugin.MAEstub class
- Typical user written plugins can adopt the following convention:

1. E.g. DemoPlugin.java implements the MAEstub methods DemoPlugin.pluginMain(), DemoPlugin.updateCurGene(), DemoPlugin.updateFilter(), DemoPlugin.updateSliders(), DemoPlugin.updateLabels()
2. It creates an instance of a Demo class which actually implements the plugin analysis
3. DemoPlugin passes down the instance of *MAExplorer.mja* to the Demo() constructor so it can access all of the MJAxxxx classes

- Data is accessed through MJAxxxx class methods
- See sample plugins and MAExplorer javadocs for more details (on Web site)

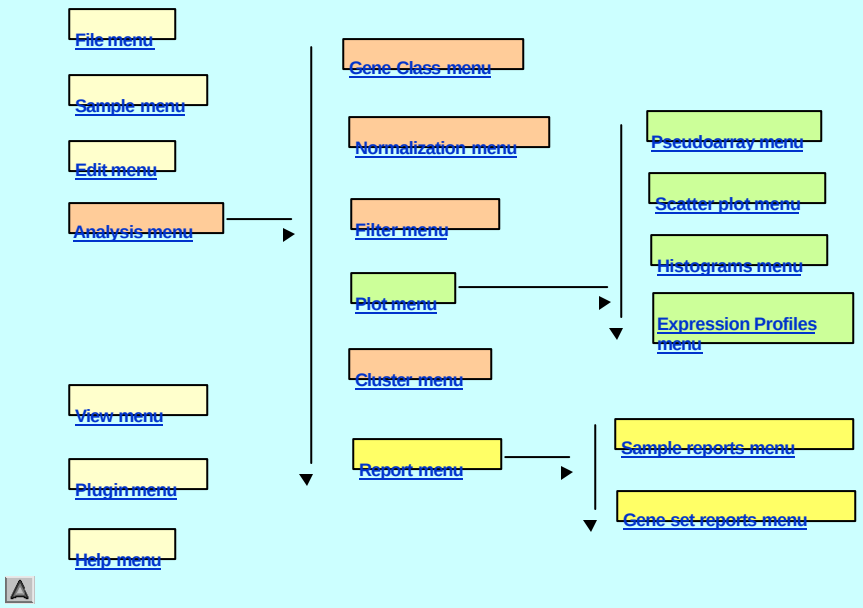
15.1 MaeJavaAPI MJAxxxx classes

- MJAbase constants used by other MJAxxxx classes
- MJAcluster cluster data structures and methods
- MJAcondition condition lists of samples & ordered lists of condition lists
- MJAeval command interpreter for use with client-server access
- MJAexprProfile expression profiles data
- MJAfilter gene data filters
- MJAgene single gene data
- MJAgeneList lists of genes and get sets
- MJAgenomicDB genomic databases on the Internet
- MJAgeometry array geometry, spot to gene maps, etc.
- MJAhelp popup browser help methods
- MJAhistogram histograms and histogram plots
- MJAmath built-in math functions
- MJAnormalization normalization data and methods

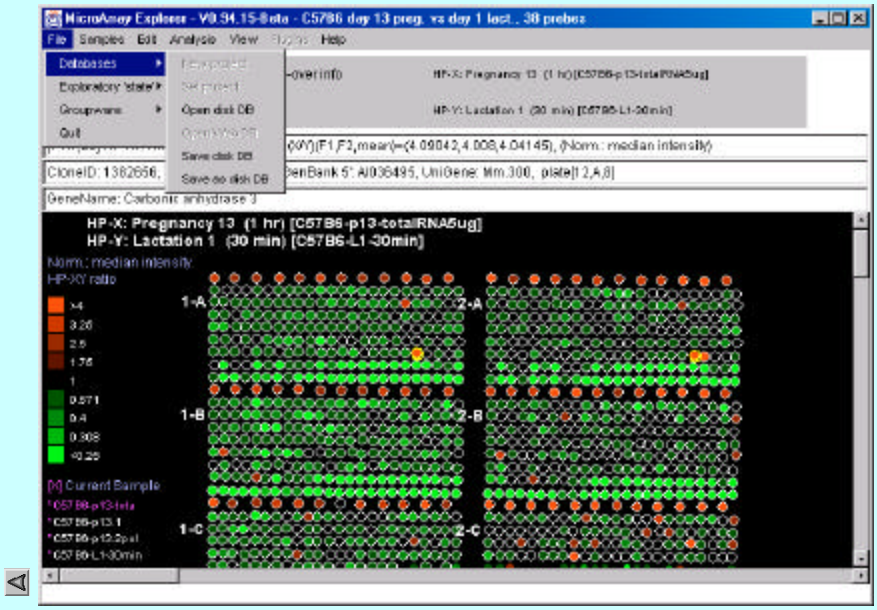
15.2 MaeJavaAPI MJAxxxx classes

- MJAproperty get and put individual MAExplorer state properties
- MJApropList get lists of properties
- MJAsample get and put single sample top-level data
- MJAsampleList get lists of samples top-level data
- MJAsort built-in sort methods oriented toward MAExplorer
- MJAstatistics built-in statistics methods
- MJAstate get and save state, get additional state info
- MJAutil built-in utility methods

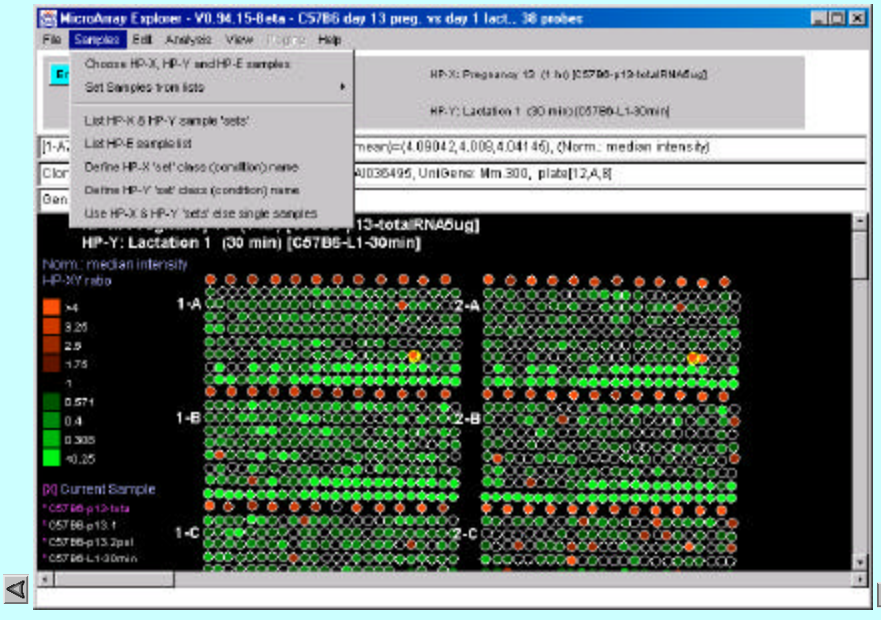
16. Examples (links): MAExplorer menu organization



15.1 Files - Open and Save Exploration State



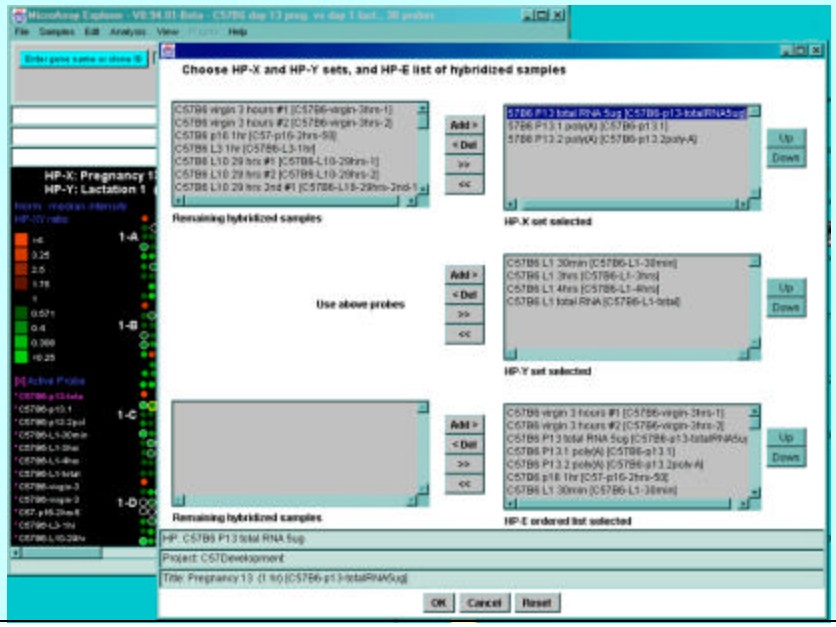
15.2 Samples - Select Condition Lists



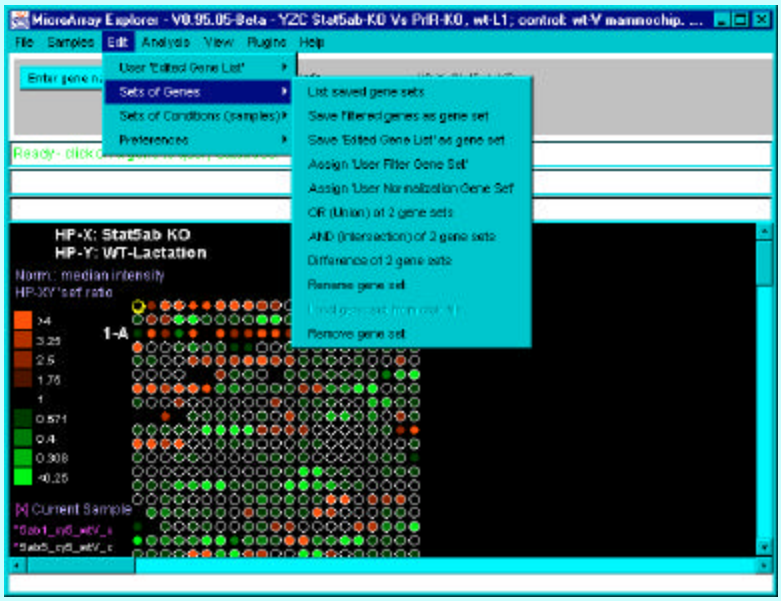
15.2.1 Lists of Hybridized Sample Conditions

- Database may consist of multiple conditions with replicates
- Organize as lists of conditions:
  1. X and Y pairs of individual samples
  2. Sets of X and Y replicates (**X-set**, **Y-set**)
  3. Ordered expression profile list (**E-list**)
- Manipulate named sample lists with AND, OR, DIFFERENCE
- Save and restore named sample lists during analysis and between sessions

15.2.2 Sample Condition HP-X &-Y Sets, HP-E Lists



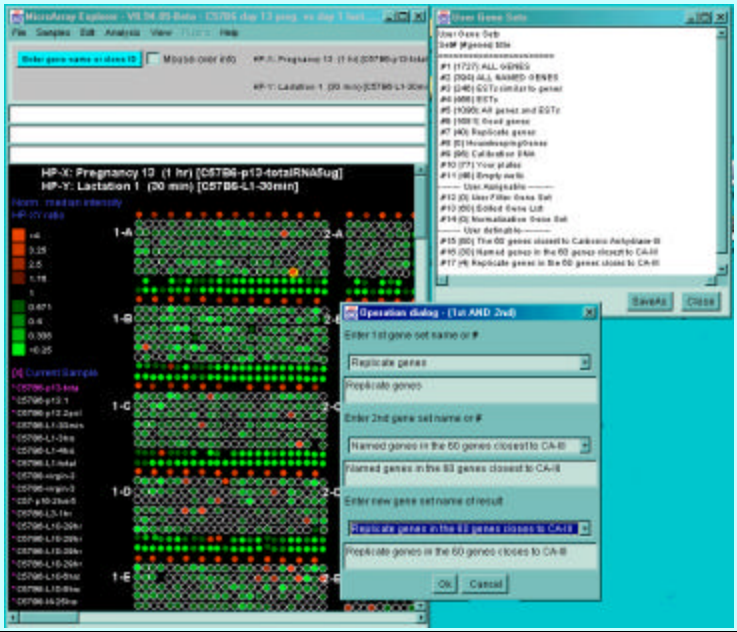
15.3 Edit Menu - Gene Sets & Condition Lists



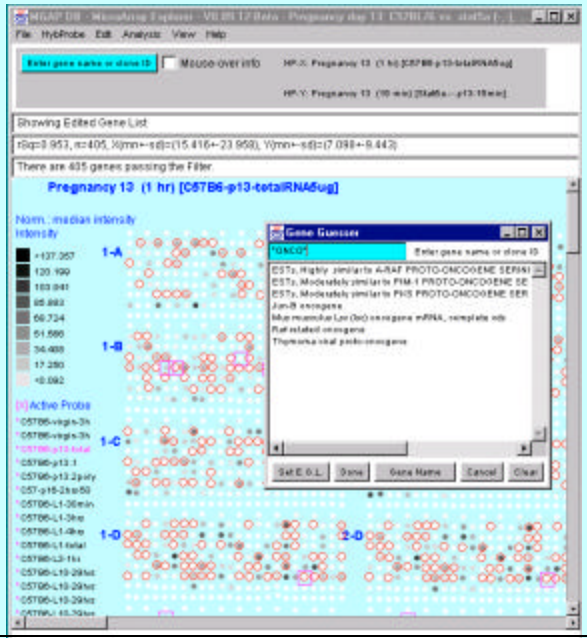
15.3.1 Gene Set Operations help manage data and search results

- All gene sets are "named" - view directory of current sets
- Set operations (AND, OR, DIFFERENCE) may be used to create new derived named sets
- Special sets:
  1. Filtered genes set holds genes passing the data filter
  2. Edited Gene List holds results of clustering or editing
  3. Normalization set may be used as normalization method
  4. User data filter set may be used as a data filter
- Genes sets are saved when the session is saved, restored when MAExplorer restarted

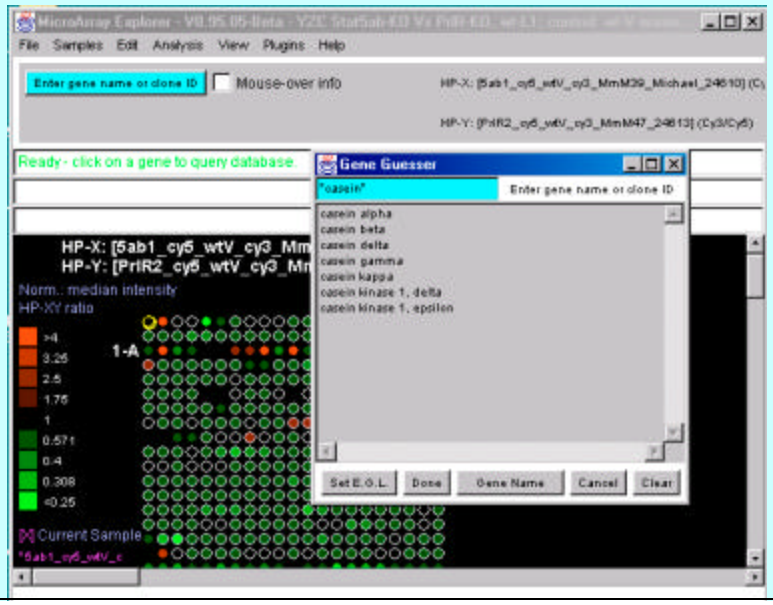
15.3.2 Gene Set Operations - e.g. 'AND' of Two Sets



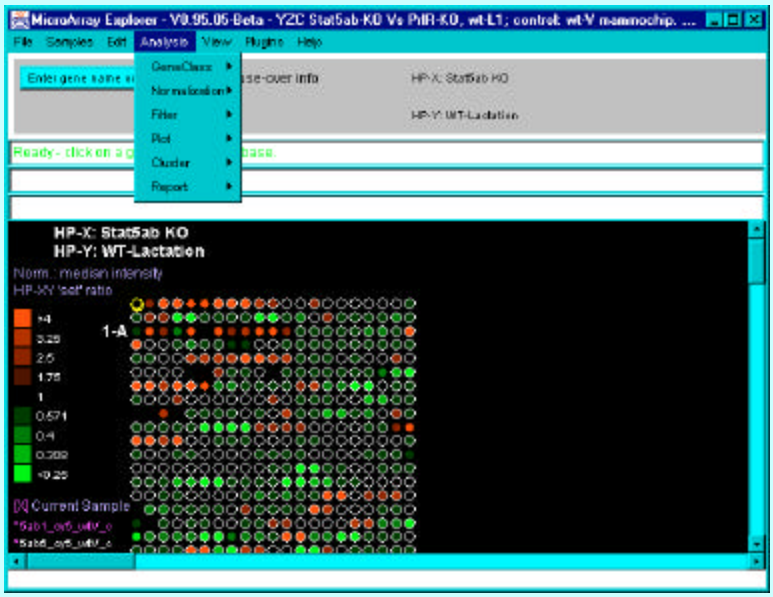
15.3.3 Find Gene - Enter gene name, then press Done



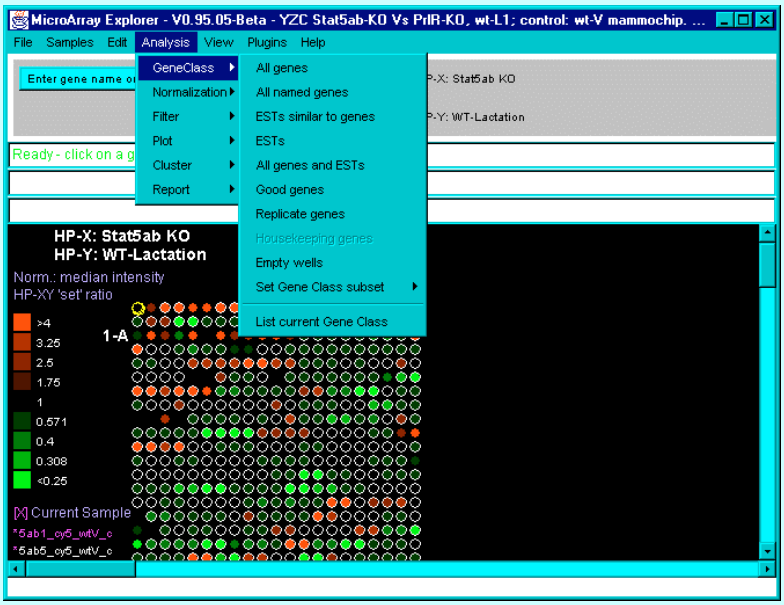
15.3.4 Find Gene Subset by Enter gene name, then Press Set E.G.L.



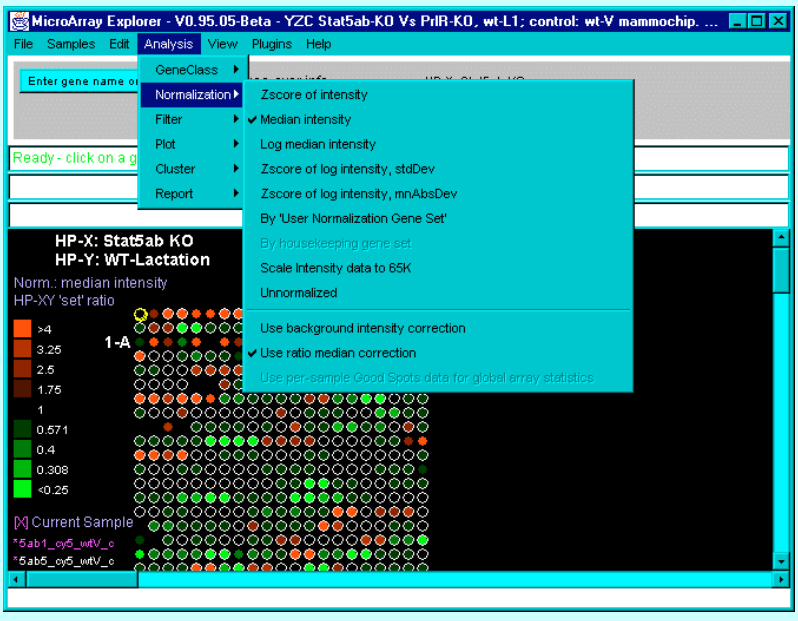
15.4 Analysis Menu - Organized by Task



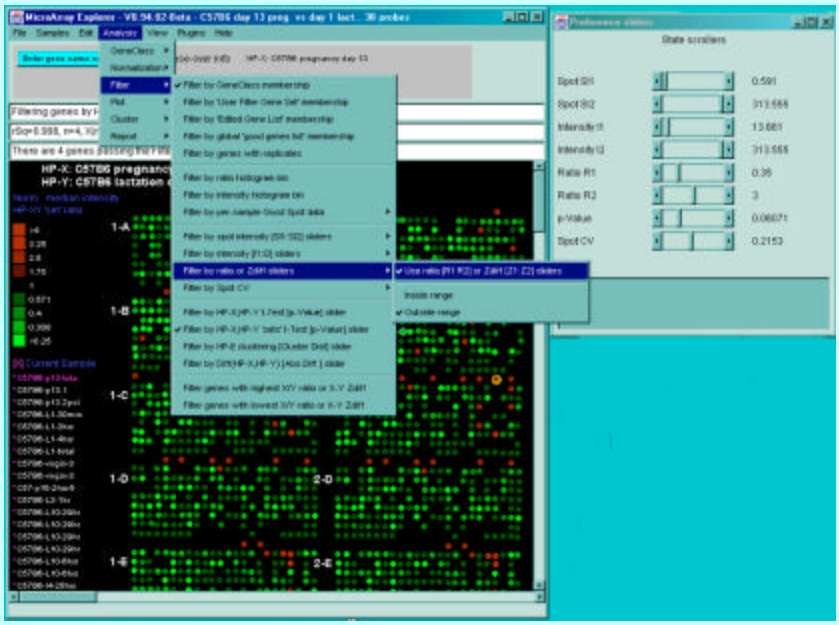
15.4.1 Gene Class Menu



15.4.2 Normalization Menu



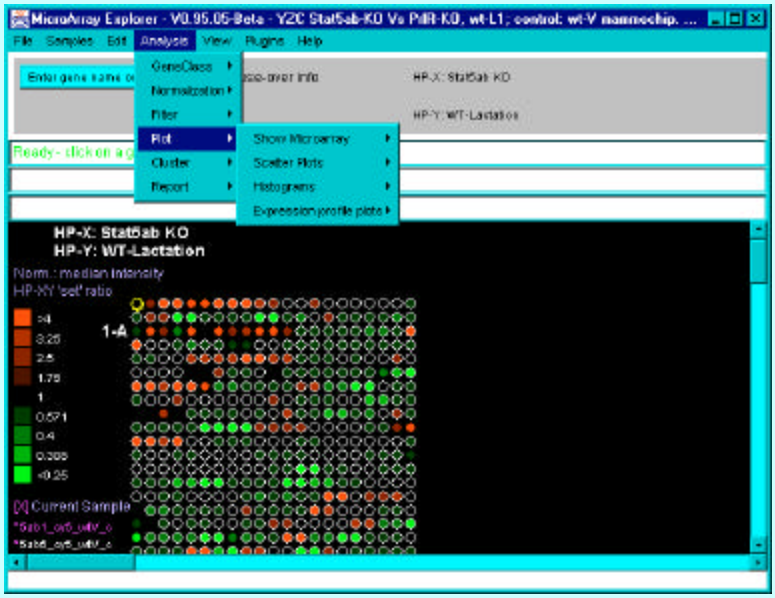
15.4.3 Filter gene data Operations By Multiple Tests



15.4.3.1 Gene data filters - finding a gene subset

- The selected data-filters tests are applied to all genes
- Creates working subset of genes used for subsequent clustering, plots, and reports
- Data filters compute intersection (AND) of sets from tests:
  1. Gene subsets from previous operations
  2. Spot intensity and ratio ranges
  3. Statistics: CV, t-Test
  4. Clustering: similar expression profiles, K-means, hierarchical
- Changing filter parameters recomputes active data filter and updates active plot or cluster displays

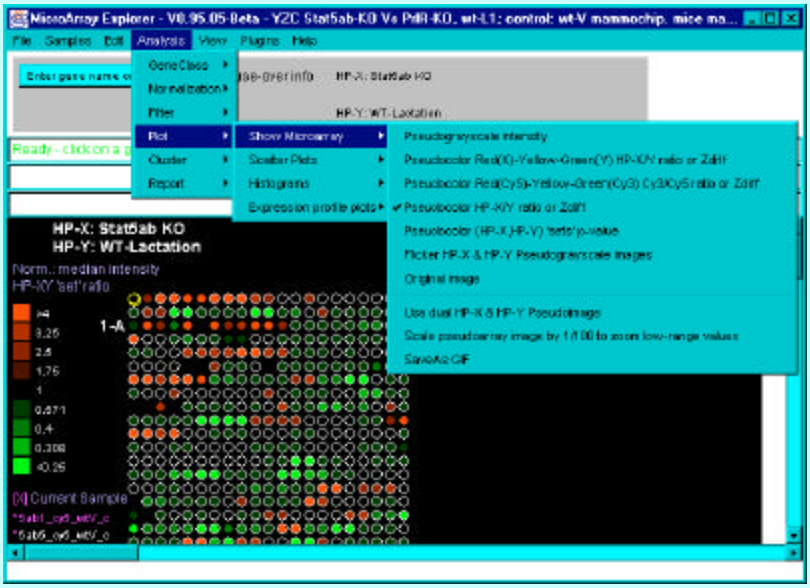
15.4.4 Plot menu - pseudoarrays, zoomable scatter plots, histograms, expression profile plots



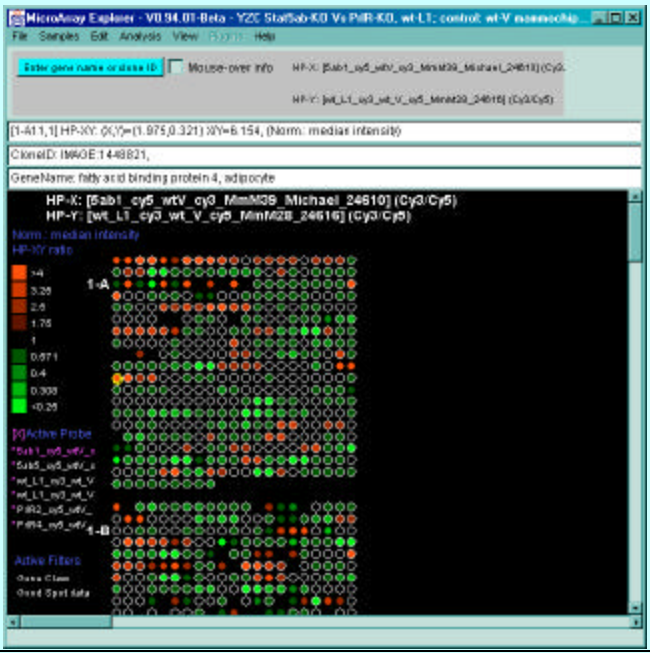
15.4.4.1 Direct manipulation of gene data in plots

1. Pseudoarray image - intensity, ratio (X/Y) or (Cy3/Cy5), sums (red+green) of (Cy5+Cy5) or (Y+X), etc.
2. Zoomable scatter plots - X vs Y, Cy3 vs Cy5, duplicate spots, labeled K-means clusters
3. Histograms - ratio and intensity, select bins for filtering
4. Expression profiles - individual genes, lists, overlay plots

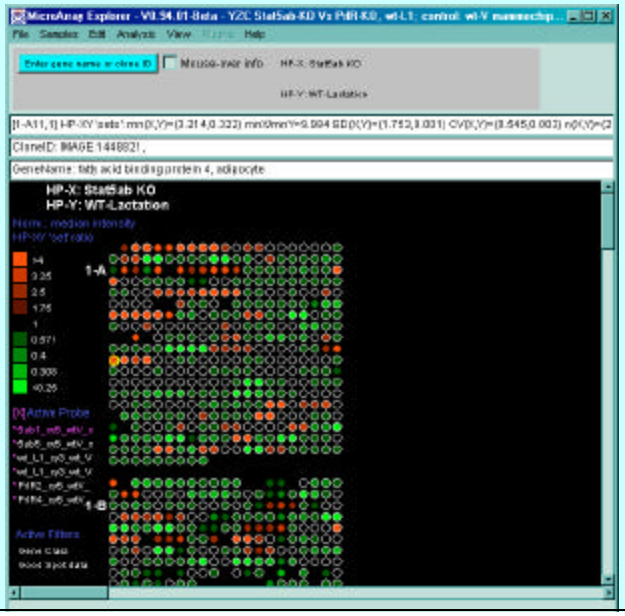
15.4.4.1 Plots - Pseudoarrays of functions of samples (Show Microarray)



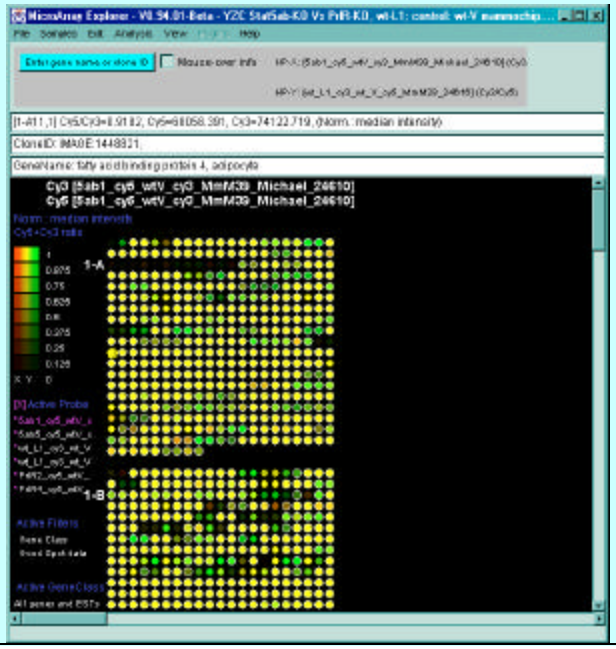
15.4.4.1.1 Pseudoarray of individual X/Y samples



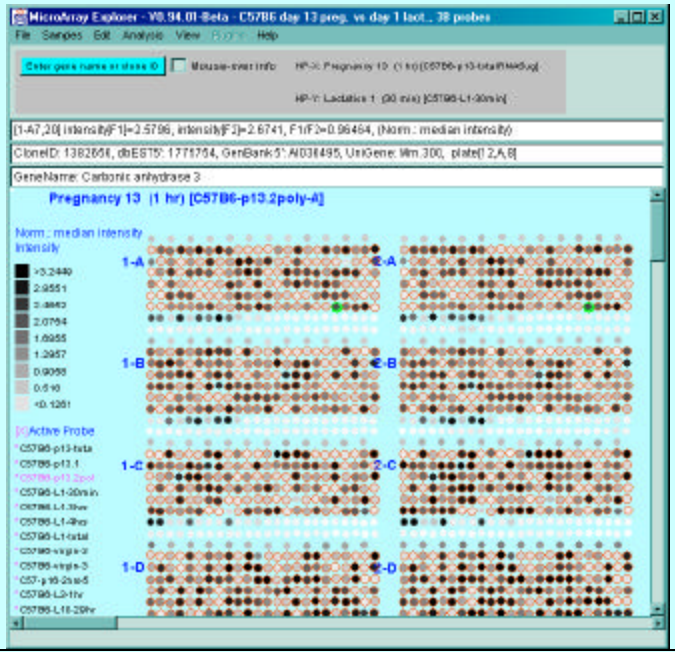
15.4.4.1.2 Pseudoarray of (mean X / meanY) sample sets



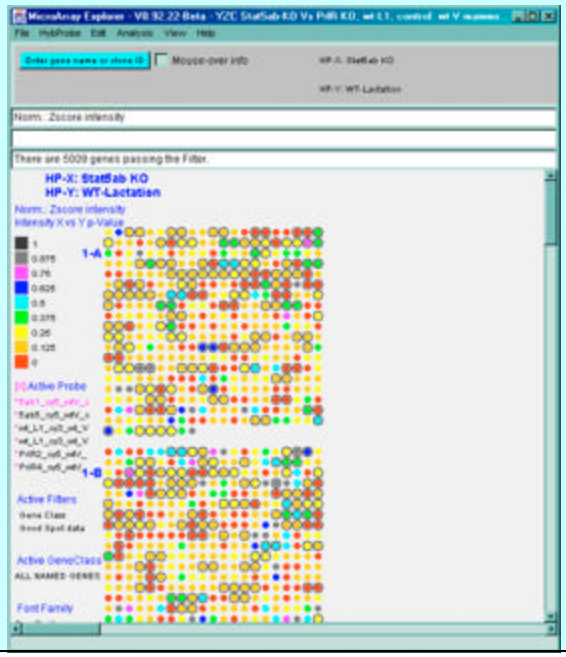
15.4.4.1.3 Pseudoarray sum Cy3+Cy5 per sample



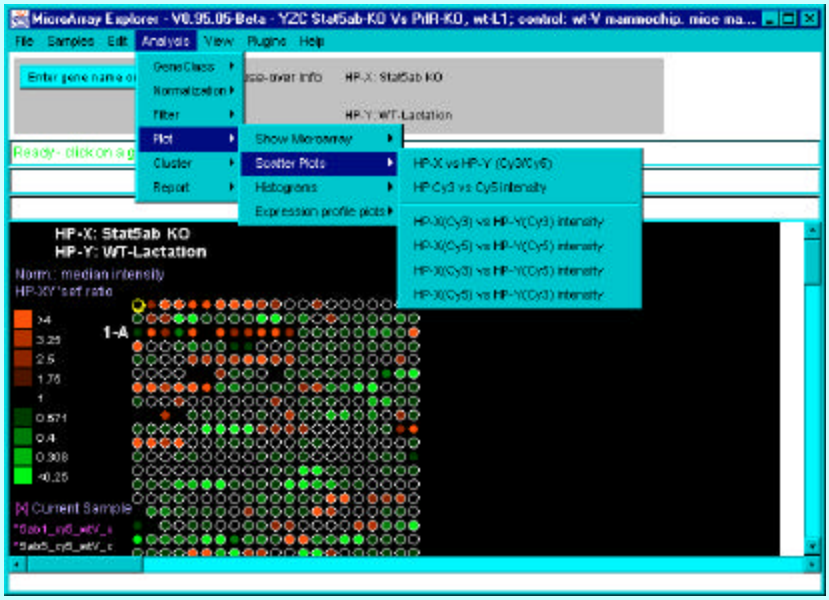
15.4.4.1.4 Pseudoarray of intensity of one sample



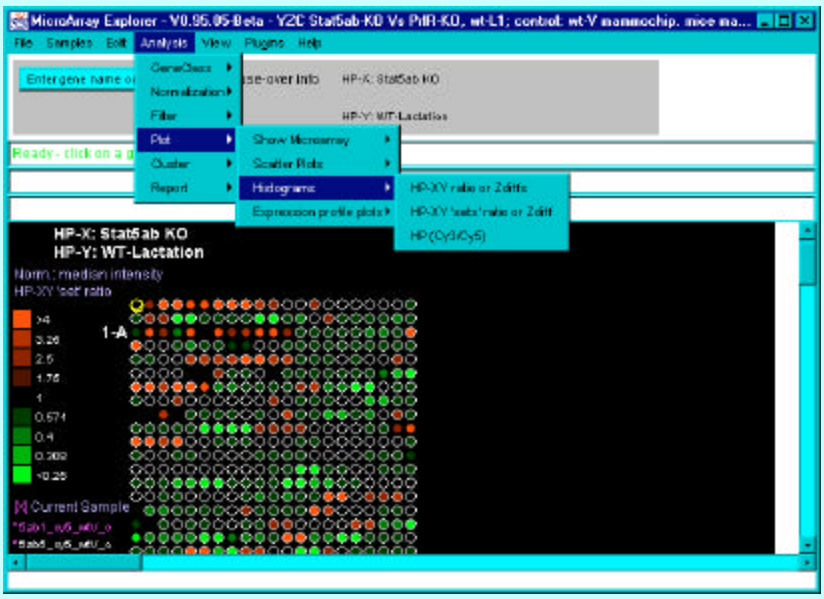
15.4.4.1.5 Pseudoarray of X- vs Y-sets p-value (t-test)



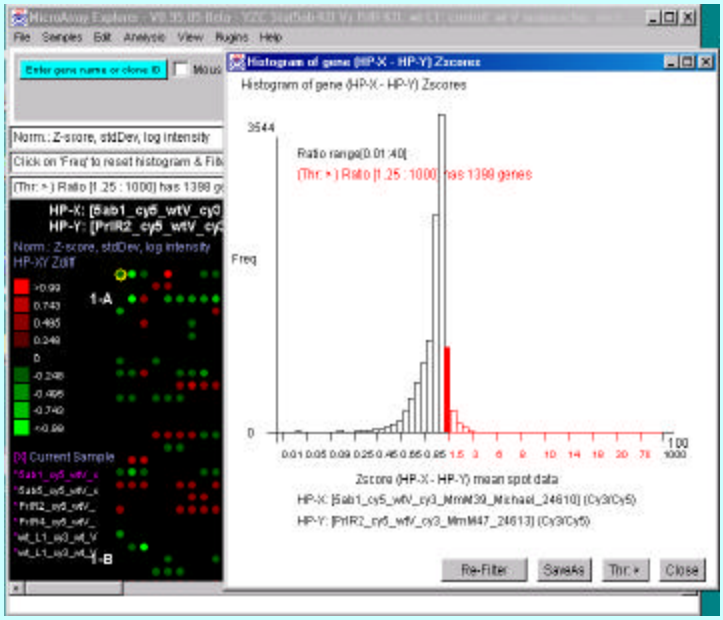
15.4.4.2 Zoomable Scatter Plots



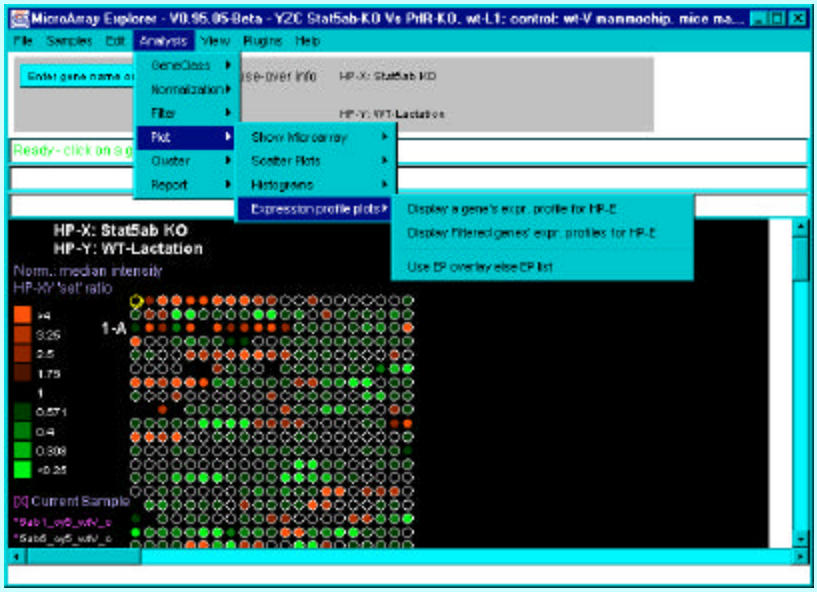
15.4.4.3 Plots - Histograms



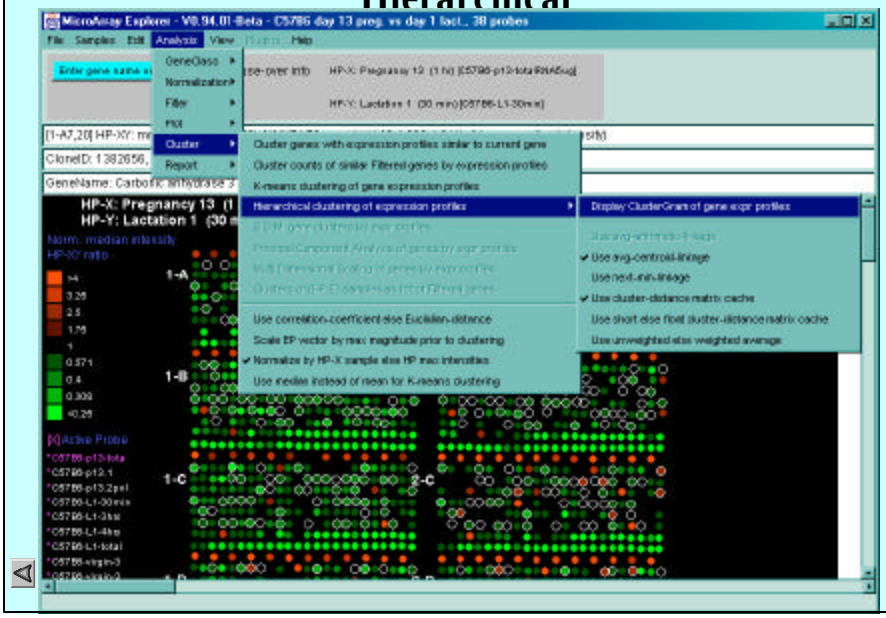
15.4.4.3.1 Plots - Ratio Histogram HP-X/HP-Y



15.4.4.4 Plots - Expression Profile plots



15.4.4.5 Clustering Similar Genes, K-means, Hierarchical



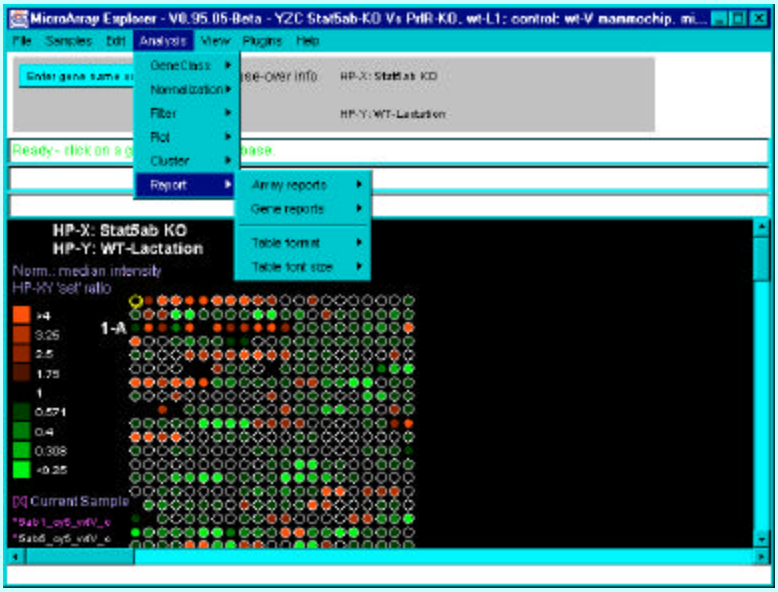
15.4.4.5.1 Cluster operations on filtered genes

1. Find genes similar to specified gene - sorted list and silhouette plots, gene reports, save cluster as set

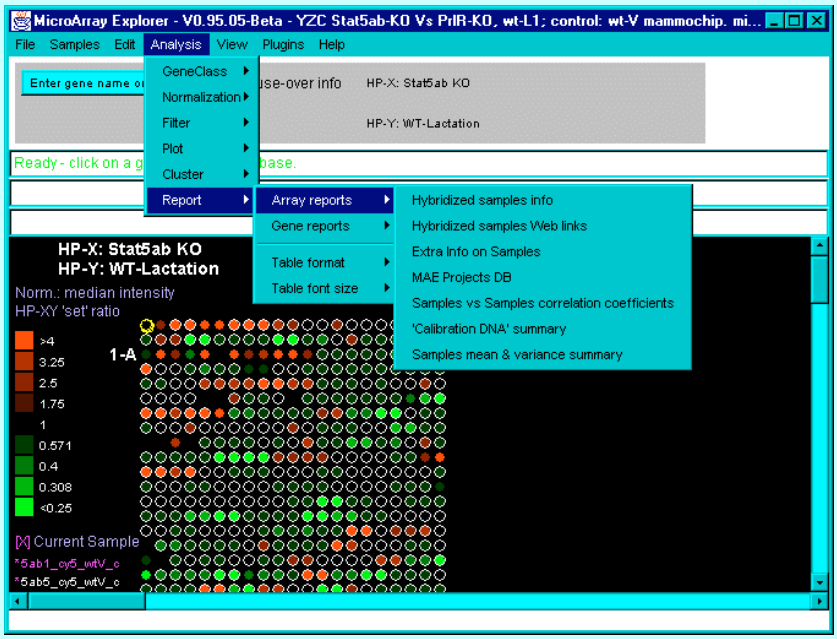
2. K-means clustering given # (K) of clusters - sorted lists and Expression Profile (EP) plots, silhouette plots, gene reports, save clusters as sets

3. Hierarchical clustering of genes - clustergram, dendrogram, list of EP plots, gene reports

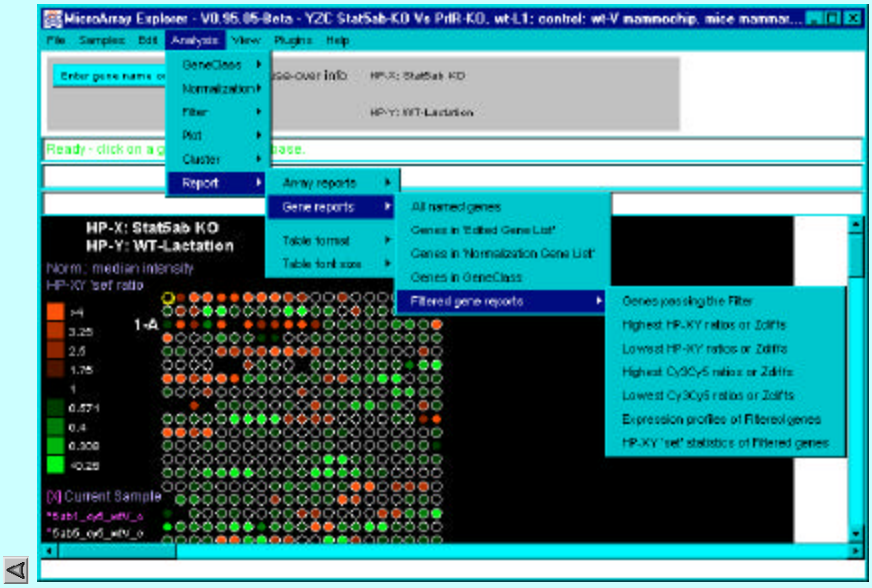
15.4.4.6 Reports - Samples and Gene Subsets



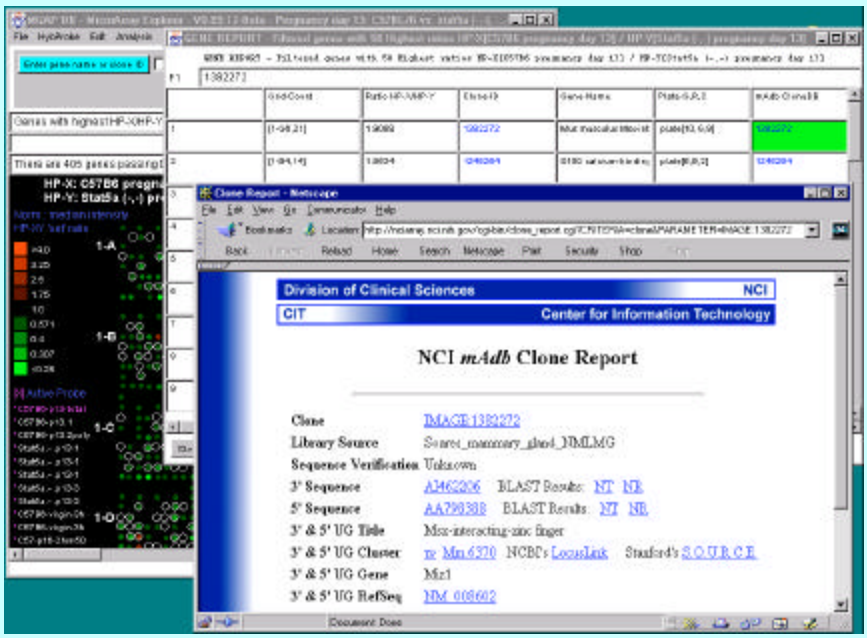
15.4.4.6.1 Report Menu - Samples Reports



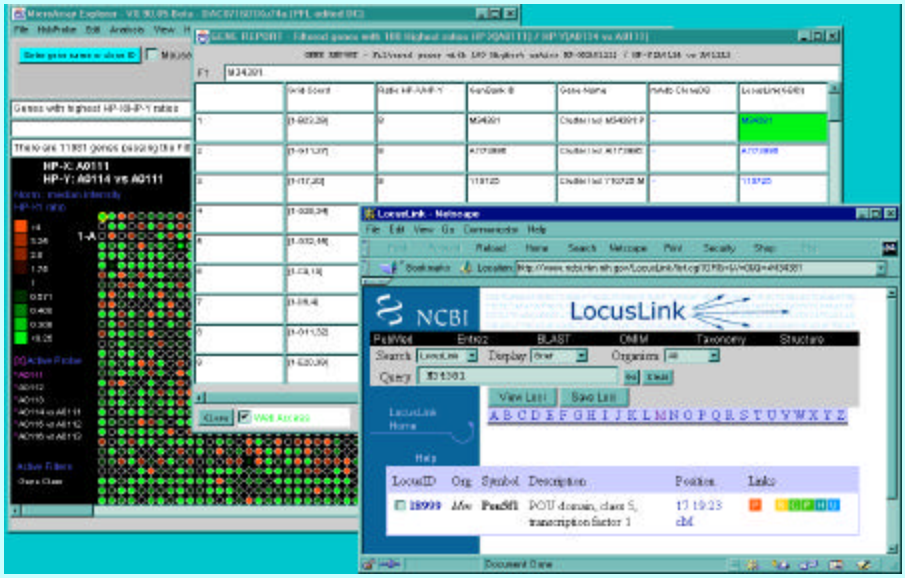
15.4.4.6.2 Report Menu - Gene Reports



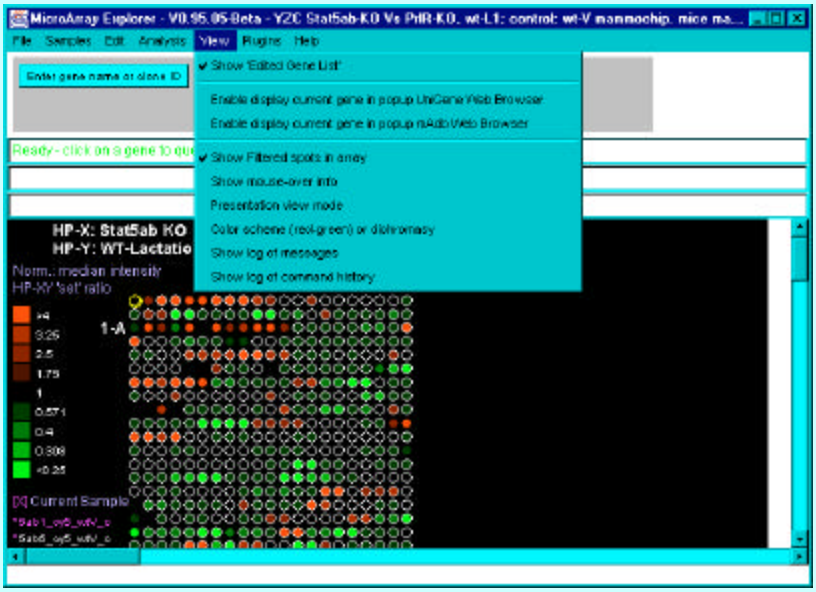
15.4.4.6.3 Scrollable Dynamic Gene Reports



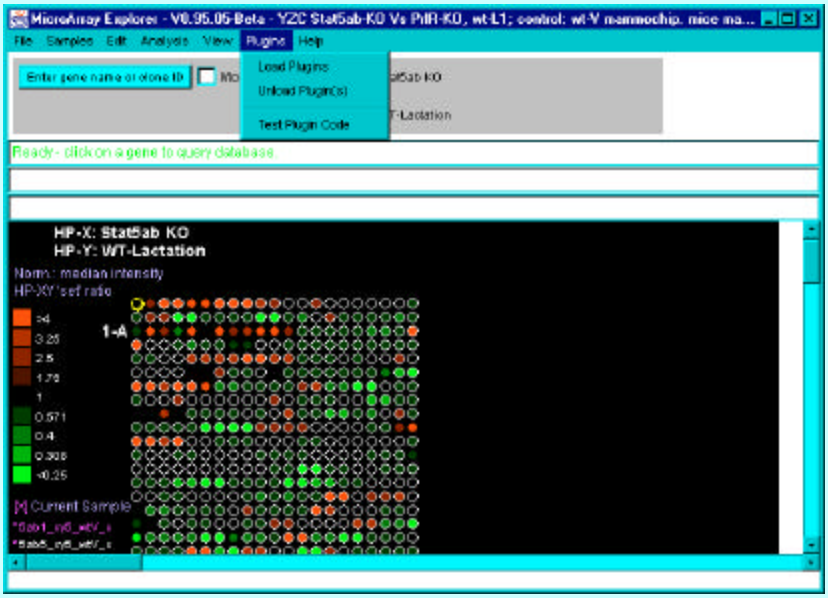
15.4.4.6.4 Scrollable Dynamic Gene Reports



15.4.5 View Menu



15.6 Plugins Menu



15.7 Help Menu