Extending MicroArray Explorer with R Language Scripts

Peter F. Lemkin(1), Gregory Thornwall(2), W. Gregory Alvord(3), Mariusz Lubomirski(3), Shyam Sundaram(4)

[1]LECB/NCI-Frederick, lemkin@ncifcrf.gov;
[2]SAIC-Frederick; [3]CSS/DMS-Frederick; [4]SRA/BIMAS/CIT-NIH

http://maexplorer.sourceforge.net/

Poster: Frederick Bioinformatics Forum June 24, 2003 (Revised: 06-14-2003)

Abstract

The Microarray Explorer (MAExplorer) is a versatile open-source Java-based data mining bioinformatic tool for analyzing quantitative DNA expression profiles across multiple microarray platforms and DNA labeling systems **http://maexplorer.sourceforge.net/**

MAExplorer analyses may now be extended by writing R scripts in the powerful R data, statistics, and graphics language (www.r-project.org).

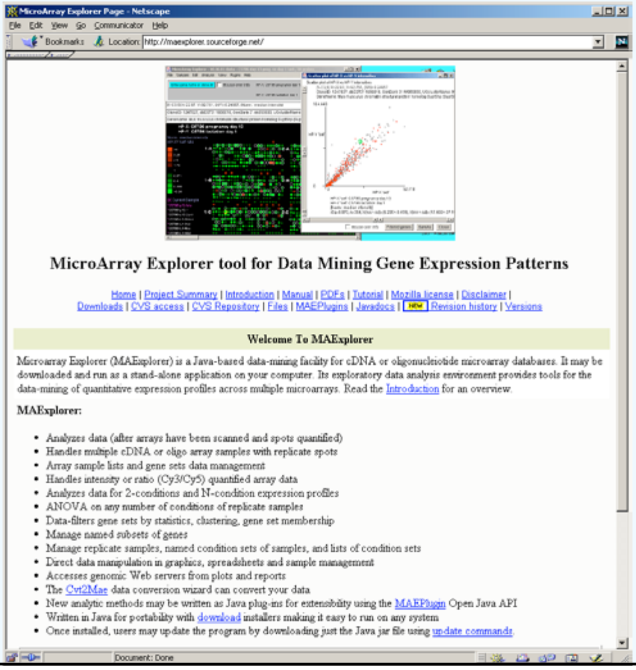
The RtestPlugin is a Java plugin for use with MAExplorer to extend its statistical data analysis capabilities using the R program. Scripts in the R language may invoke statistical and other methods in R.

Abstract (continued)

Users may create and edit R scripts and associated R LayOut (RLO) descriptions of the interface between MAExplorer and R. The RLO describes data exported from MAExplorer into R (R input files), output files from R after the R script has been evaluated, and actions to import R results back into MAExplorer.

Once a RLO and its associated R script has been created and tested, it may be saved as a special purpose "permanent" plugin so it automatically appears as a plugin extension to MAExplorer.

Home Page: http://maexplorer.sourceforge.net/



MicroArray Explorer tool for Data Mining Gene Expression Patterns

Home | Project Summary | Introduction | Manual | PDFs | Tutorial | Mozilla license | Disclaimer |
Downloads | CVS access | CVS Repository | Files | MAEPlugins | Javadocs | NEW Revision history | Versions

Welcome To MAExplorer

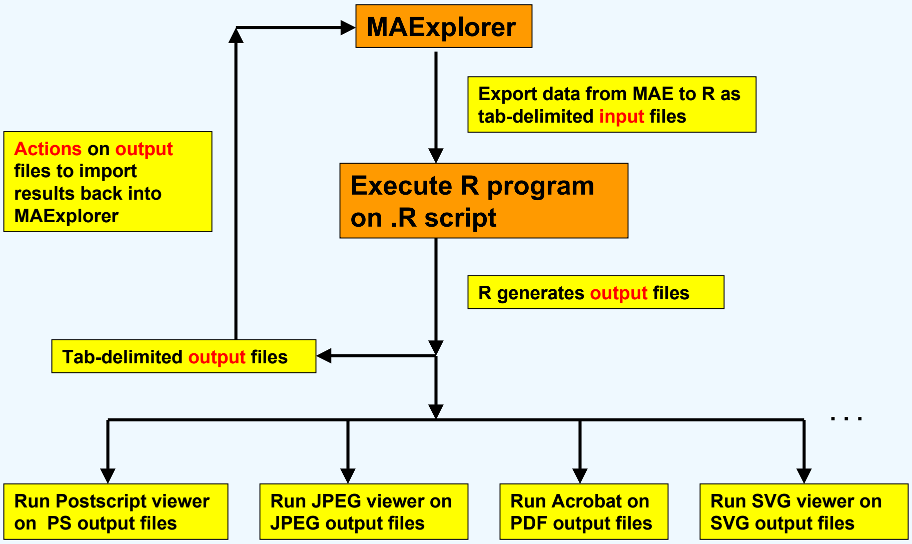
Microarray Explorer (MAExplorer) is a Java-based data-mining facility for cDNA or oligonucleiotide microarray databases. It may be downloaded and run as a stand-alone application on your computer. Its exploratory data analysis environment provides tools for the data-mining of quantitative expression profiles across multiple microarrays. Read the Introduction for an overview.

MAExplorer:

- Analyzes data (after arrays have been scanned and spots quantified)
- Handles multiple cDNA or oligo array samples with replicate spots
- Array sample lists and gene sets data management
- Handles intensity or ratio (Cy3/Cy5) quantified array data
- Analyzes data for 2-conditions and N-condition expression profiles
- ANOVA on any number of conditions of replicate samples
- Data-filters gene sets by statistics, clustering, gene set membership
- Manage named subsets of genes
- Manage replicate samples, named condition sets of samples, and lists of condition sets
- Direct data manipulation in graphics, spreadsheets and sample management
- Accesses genomic Web servers from plots and reports
- The Cvt2Mae data conversion wizard can convert your data
- New analytic methods may be written as Java plug-ins for extensibility using the MAEPlugin Open Java API
- Written in Java for portability with download installers making it easy to run on any system
- Once installed, users may update the program by downloading just the Java jar file using update commands.

Evaluating R Scripts on MAExplorer Data

The "R Layout" (RLO) specifies the R script, menu name, input/output files, actions, etc.



MAExplorer

Export data from MAE to R as tab-delimited input files

Execute R program on .R script

R generates output files

Tab-delimited output files

Actions on output files to import results back into MAExplorer

Run Postscript viewer on PS output files

Run JPEG viewer on JPEG output files

Run Acrobat on PDF output files

Run SVG viewer on SVG output files

. . .

Functionality of RtestPlugin

- Create/edit R scripts and associated R LayOut (RLO) descriptions of the interface between MAExplorer and R
- Evaluate R scripts using data exported from MAExplorer and generate R output files (text, PDF, image, etc.)
- Display PDF or image files generated by R after the script has been evaluated
- Import data from R output back into MAExplorer after the script has been evaluated
- Create user-defined RLOs using data templates to generate R code
- Use to learn the R interface to MAExplorer with editable demonstration examples for various data MAExplorer data types

R Methods: User Defined Analyses

- Use Templates to generate data-dependent R code
- Allows totally general analyses
- May import R script code files to serve as a basis for a new RLO which may then be edited
- Invoke an editing wizard when create a "New RLO" or use "Edit RLO" on a previously saved RLO
- Permanently saved with "Save RLO" command
- When "saved", the RLO menu name will automatically be added to MAExplorer menu

---

R Methods: Demonstrations

- Data-specific demos allow selecting 'canned' R analysis methods specific for various MAExplorer data
- Simple methods are intended for demonstration purposes to illustrate how to access MAExplorer from R scripts
- User may modify the generated code by specifying which R analyses (e.g., histogram, plots, t-test, summaries, etc.) should be added to the R script
- Editing is invoked using the "Edit RLO" command.
- RLO's may be saved for invoking from MAExplorer (Plugins | RLO Methods) menu

R LayOut (RLO) Schema

- **R script name** (e.g., ”SomeAnalysis.R")
- Analysis method **menu name** (e.g., ”Perform some analysis on the data")
- **Menu stub name** to place this RLO method in the MAExplorer menu tree
- Optional list of exported MAExplorer data in **R input files** (file names, data-types)
- Optional list of **R output files** written by R during script execution (file names, data-types)
- Optional list of **actions** to be taken on R output files used to import the R results back into MAExplorer (specified by: output-file-name, MAE-object-name, MAE-object-action)
- **Processing options** for MAExplorer data to be exported to R (e.g., raw vs. normalized data, filtered vs. unfiltered, etc., horizontal vs. vertical data stacking, etc.)
- Optional **R command line options** (e.g., increase memory, etc).
- (Demo mode only) **generate demonstration R script methods** to let you configure different types of R analyses in your script.

R Data Types Exported from MAExplorer as Input data to R

- Single HP-sample expression data
- HP-sample (Cy3,Cy5) or (F1,F2) channels expression data
- HP-X and HP-Y samples expression data
- HP-X 'set' and HP-Y 'set' samples expression data

- HP-X ’set' samples expression data
- HP-Y ’set' samples expression data
- HP-E 'list' expression profile samples expression data
- Current Condition list of samples expression data
- Multiple Conditions List (OCL) of samples expression data

- Gene annotation data (e.g., gene ids, gene names)
- MAExplorer state thresholds (e.g, p-value & other sliders)

R Output Data Types

These are files that may be specified to be created by R

- Stdout is the normal text output of R
- Tab-delimited table (to import data to MAExplorer from R)
- XML table (to importing data to programs from R)
- Report (free form)
- PDF file
- JPEG image file
- PNG image file
- Postscript file
- SVG file

MAExplorer Actions on R Output Data Types

Create named data structures as part of the MAExplorer state by importing R output data back into MAExplorer

- Gene set
- Gene set with p-Value
- Gene set with cluster data (#, cluster distance)
- Filter gene set
- Normalization gene set
- Condition List of (replicate or similar) samples
- Multiple Conditions List (OCL) of condition lists
- Named normalization map [Future]
- 2D plot overlay map [Future]
- Normalization polynomial [Future]

Resources Required

- Must install MAExplorer and the R program
- Both are open source and free
  http://maexplorer.sourceforge.net/
  http://www.r-project.org/
- MAExplorer and R run on Windows, Linux, Solaris, MacOS-X
- To view PDF generated plots, you could use Adobe Acrobat
- No other software is required

Disk directories used

- *<Current MAExplorer project directory>*
  - **/Temp** - holds all files that communicate with the R program from MAExplorer
  - **/Report** - holds all files that are generated by your R script as specified by the RLO for that script
- *<MAExplorer installation directory>*
  - **/RLO** - holds all permanent **.rlo** RLO specification files in your RLO database. Read by MAExplorer when it starts if the **.rlo** files exist
  - **/R** - holds all permanent **.R** script files for the RLO database
  - **/MAERlibr** - is R style project library to support MAExplorer-R interface

Running RtestPlugin

- First load the RtestPlugin
  1. In MAExplorer, select the (Plugins menu | Load plugin) command.
  2. Specify "RtestPlugin.jar". This installs the plugin in the MAExplorer Plugins menu with the new menu entry "R test.
  3. To run RtestPlugin, select (Plugins menu | R test)

- Select demo or user-created RLO to edit or evaluate

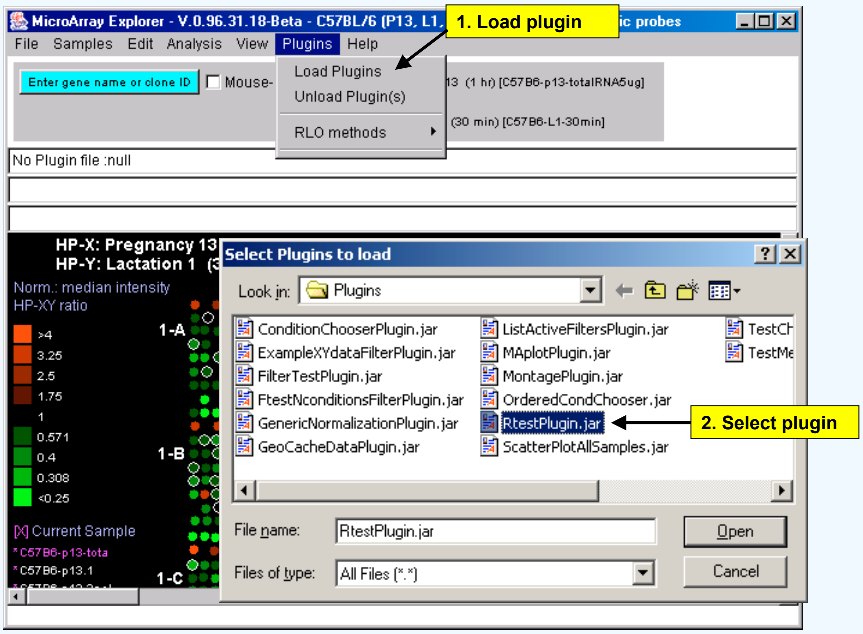
- If you want, edit or create new RLOs and save them using the “Save RLO” button in the permanent RLO database

- Press “Evaluate” to invoke R on the selected RLO script

- Or, after saving RLOs, they may be run created directly from MAExplorer (Plugins | RLO methods) submenu

---

Loading RtestPlugin in MAExplorer



MicroArray Explorer - V.0.96.31.18-Beta - C57BL/6 (P13, L1, ... ic probes

No Plugin file :null

Look in: Plugins

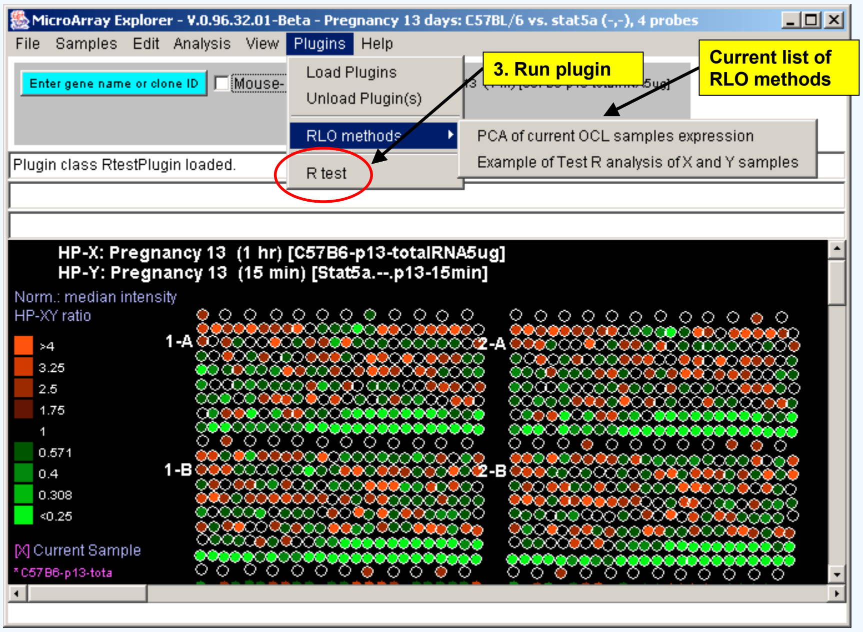
File name: RtestPlugin.jar

Files of type: All Files (*.*)

1. Load plugin

2. Select plugin

Running RtestPlugin


MicroArray Explorer - V.0.96.32.01-Beta - Pregnancy 13 days: C57BL/6 vs. stat5a (-,-), 4 probes

Plugin class RtestPlugin loaded.

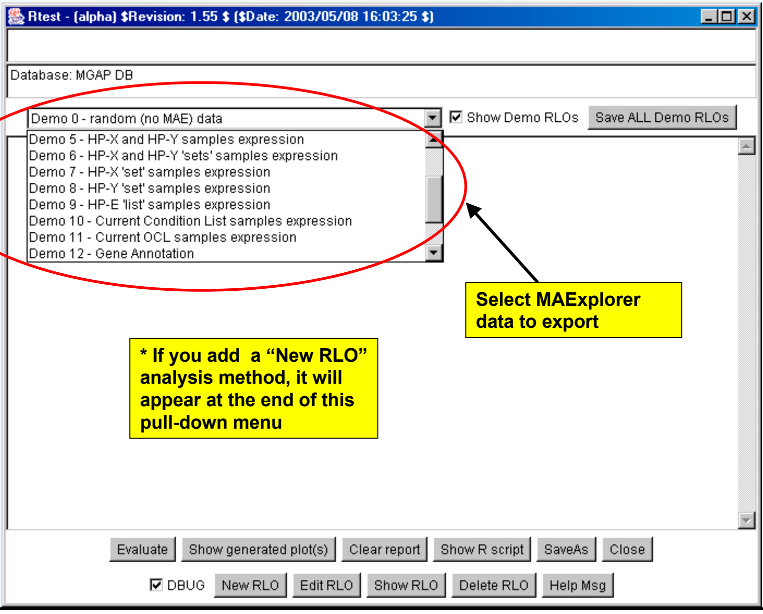
HP-X: Pregnancy 13 (1 hr) [C57B6-p13-totalRNA5ug]
HP-Y: Pregnancy 13 (15 min) [Stat5a.--.p13-15min]

Norm.: median intensity

3. Run plugin

Current list of RLO methods

Select a Data Analysis and Its RLO for Analyzing Current MAExplorer Data*


Rtest - (alpha) $Revision: 1.55 $ ($Date: 2003/05/08 16:03:25 $)

Database: MGAP DB

Demo 0 - random (no MAE) data

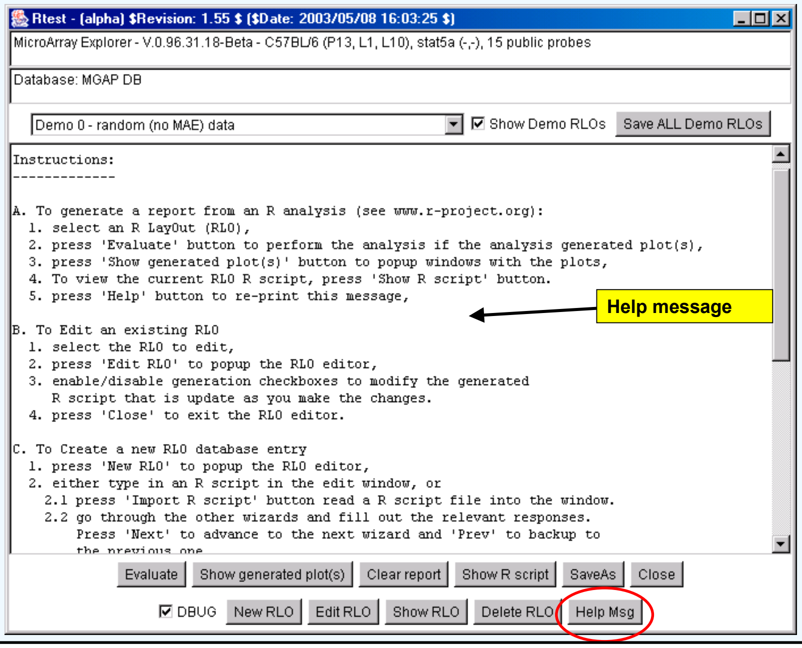
☑ Show Demo RLOs

- Demo 5 - HP-X and HP-Y samples expression
- Demo 6 - HP-X and HP-Y 'sets' samples expression
- Demo 7 - HP-X 'set' samples expression
- Demo 8 - HP-Y 'set' samples expression
- Demo 9 - HP-E 'list' samples expression
- Demo 10 - Current Condition List samples expression
- Demo 11 - Current OCL samples expression
- Demo 12 - Gene Annotation

Select MAExplorer data to export

*** If you add a “New RLO” analysis method, it will appear at the end of this pull-down menu**

Help Message - RtestPlugin



Rtest - (alpha) $Revision: 1.55 $ ($Date: 2003/05/08 16:03:25 $)

MicroArray Explorer - V.0.96.31.18-Beta - C57BL/6 (P13, L1, L10), stat5a (-,-), 15 public probes

Database: MGAP DB

Demo 0 - random (no MAE) data

☑ Show Demo RLOs

```
Instructions:
-------------

A. To generate a report from an R analysis (see www.r-project.org):
  1. select an R LayOut (RLO),
  2. press 'Evaluate' button to perform the analysis if the analysis generated plot(s),
  3. press 'Show generated plot(s)' button to popup windows with the plots,
  4. To view the current RLO R script, press 'Show R script' button.
  5. press 'Help' button to re-print this message,

B. To Edit an existing RLO
  1. select the RLO to edit,
  2. press 'Edit RLO' to popup the RLO editor,
  3. enable/disable generation checkboxes to modify the generated
     R script that is update as you make the changes.
  4. press 'Close' to exit the RLO editor.

C. To Create a new RLO database entry
  1. press 'New RLO' to popup the RLO editor,
  2. either type in an R script in the edit window, or
    2.1 press 'Import R script' button read a R script file into the window.
    2.2 go through the other wizards and fill out the relevant responses.
        Press 'Next' to advance to the next wizard and 'Prev' to backup to
        the previous one
```

☑ DBUG

Help message

An R Script Contains Three Parts

1. **Prologue R code** reads the exported MAExplorer data as R input tab-delimited data files using R read.table() or scan() methods to fixed tables.

2. **Computation R code** performs the analysis of the data read in prologue.

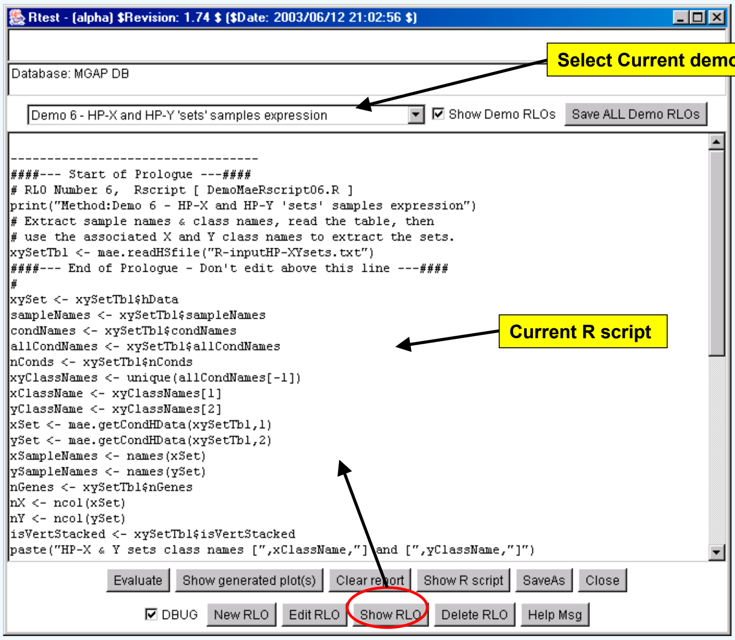
3. **Epilogue R code** to write data to output files using plot files, or text tab-delimited table files using R write.table() methods.

Simple Example of Three Part R Script

```
####--- Start of Prologue ---####
# RLO Number 13, Rscript [ DemoMaeRscript13.R ]
print("Method:Demo 13 - MAExplorer State-Scroller thresholds")
maeStateThr <- read.table("R-inputStateThresholds.txt",
                          header=TRUE, sep="\t")
####--- End of Prologue  - Don't edit above this line ---####

print("MAExplorer state-scroller threshold values")
maeStateThr

####--- Start of Epilogue - Don't edit below this line ---####
####--- End of Epilogue ---####
```

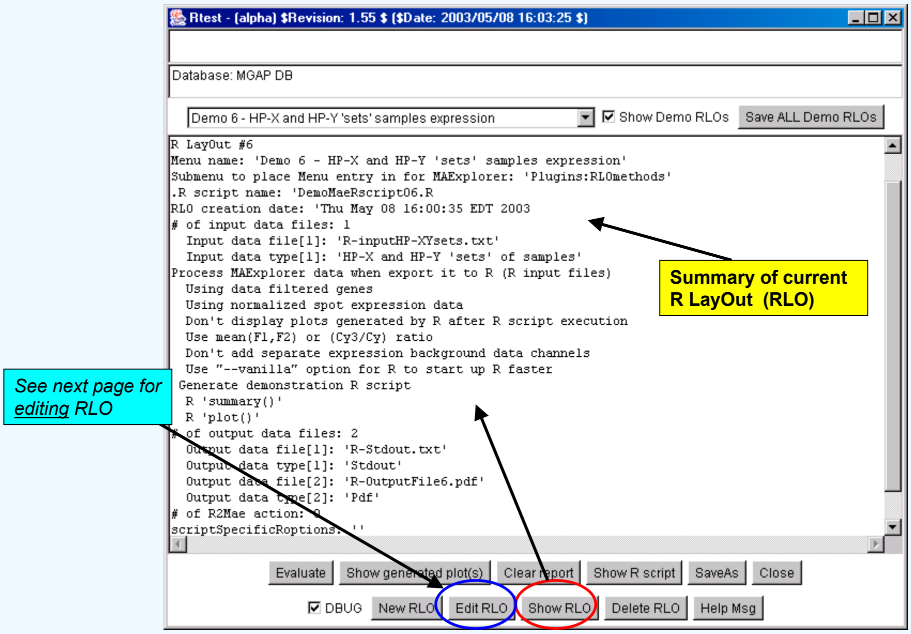
Note: there is no epilogue R code in this example

Viewing R Script of Currently Selected RLO



```
------------------------------------
####--- Start of Prologue ---####
# RLO Number 6,  Rscript [ DemoMaeRscript06.R ]
print("Method:Demo 6 - HP-X and HP-Y 'sets' samples expression")
# Extract sample names & class names, read the table, then
# use the associated X and Y class names to extract the sets.
xySetTbl <- mae.readHSfile("R-inputHP-XYsets.txt")
####--- End of Prologue - Don't edit above this line ---####
#
xySet <- xySetTbl$hData
sampleNames <- xySetTbl$sampleNames
condNames <- xySetTbl$condNames
allCondNames <- xySetTbl$allCondNames
nConds <- xySetTbl$nConds
xyClassNames <- unique(allCondNames[-1])
xClassName <- xyClassNames[1]
yClassName <- xyClassNames[2]
xSet <- mae.getCondHData(xySetTbl,1)
ySet <- mae.getCondHData(xySetTbl,2)
xSampleNames <- names(xSet)
ySampleNames <- names(ySet)
nGenes <- xySetTbl$nGenes
nX <- ncol(xSet)
nY <- ncol(ySet)
isVertStacked <- xySetTbl$isVertStacked
paste("HP-X & Y sets class names [",xClassName,"] and [",yClassName,"]")
```

Viewing the Currently Selected <u>RLO</u>



Rtest - (alpha) $Revision: 1.55 $ ($Date: 2003/05/08 16:03:25 $)

Database: MGAP DB

Demo 6 - HP-X and HP-Y 'sets' samples expression

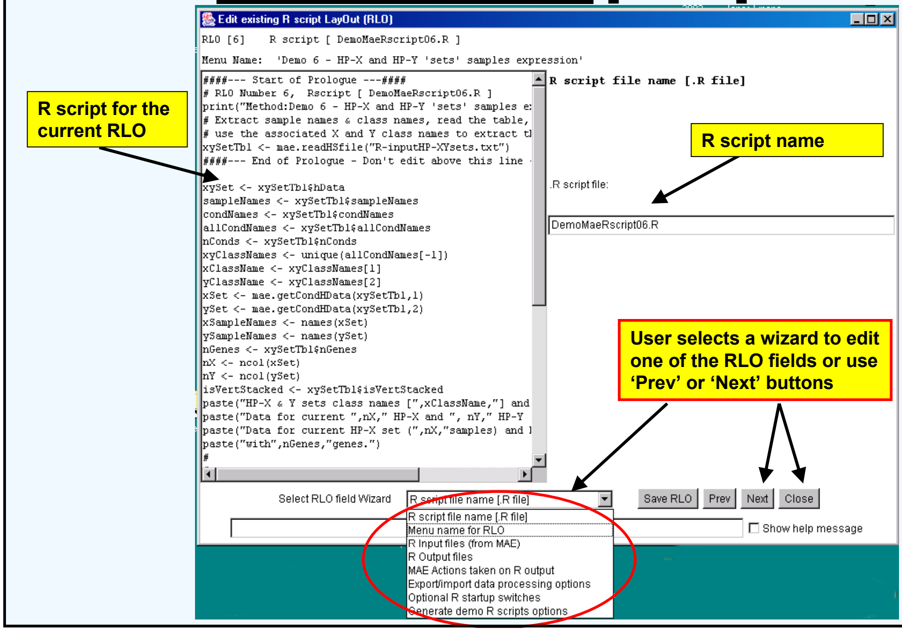
☑ Show Demo RLOs

```
R LayOut #6
Menu name: 'Demo 6 - HP-X and HP-Y 'sets' samples expression'
Submenu to place Menu entry in for MAExplorer: 'Plugins:RLOmethods'
.R script name: 'DemoMaeRscript06.R
RLO creation date: 'Thu May 08 16:00:35 EDT 2003
# of input data files: 1
  Input data file[1]: 'R-inputHP-XYsets.txt'
  Input data type[1]: 'HP-X and HP-Y 'sets' of samples'
Process MAExplorer data when export it to R (R input files)
  Using data filtered genes
  Using normalized spot expression data
  Don't display plots generated by R after R script execution
  Use mean(F1,F2) or (Cy3/Cy) ratio
  Don't add separate expression background data channels
  Use "--vanilla" option for R to start up R faster
 Generate demonstration R script
  R 'summary()'
  R 'plot()'
# of output data files: 2
  Output data file[1]: 'R-Stdout.txt'
  Output data type[1]: 'Stdout'
  Output data file[2]: 'R-OutputFile6.pdf'
  Output data type[2]: 'Pdf'
# of R2Mae action: 0
scriptSpecificRoptions: ''
```

Summary of current R LayOut (RLO)

See next page for <u>editing</u> RLO

1. Edit Wizard for Current RLO: <u>R Script File Name</u> [.R file]



Edit existing R script LayOut (RLO)

RLO [6]   R script [DemoMaeRscript06.R]

Menu Name: 'Demo 6 - HP-X and HP-Y 'sets' samples expression'

```
####--- Start of Prologue ---####
# RLO Number 6,  Rscript [ DemoMaeRscript06.R ]
print("Method:Demo 6 - HP-X and HP-Y 'sets' samples e
# Extract sample names & class names, read the table,
# use the associated X and Y class names to extract t
xySetTbl <- mae.readHSfile("R-inputHP-XYsets.txt")
####--- End of Prologue - Don't edit above this line

xySet <- xySetTbl$hData
sampleNames <- xySetTbl$sampleNames
condNames <- xySetTbl$condNames
allCondNames <- xySetTbl$allCondNames
nConds <- xySetTbl$nConds
xyClassNames <- unique(allCondNames[-1])
xClassName <- xyClassNames[1]
yClassName <- xyClassNames[2]
xSet <- mae.getCondHData(xySetTbl,1)
ySet <- mae.getCondHData(xySetTbl,2)
xSampleNames <- names(xSet)
ySampleNames <- names(ySet)
nGenes <- xySetTbl$nGenes
nX <- ncol(xSet)
nY <- ncol(ySet)
isVertStacked <- xySetTbl$isVertStacked
paste("HP-X & Y sets class names [",xClassName,"] and
paste("Data for current ",nX," HP-X and ", nY," HP-Y
paste("Data for current HP-X set (",nX,"samples) and I
paste("with",nGenes,"genes.")
#
```

R script file name [.R file]

.R script file: DemoMaeRscript06.R

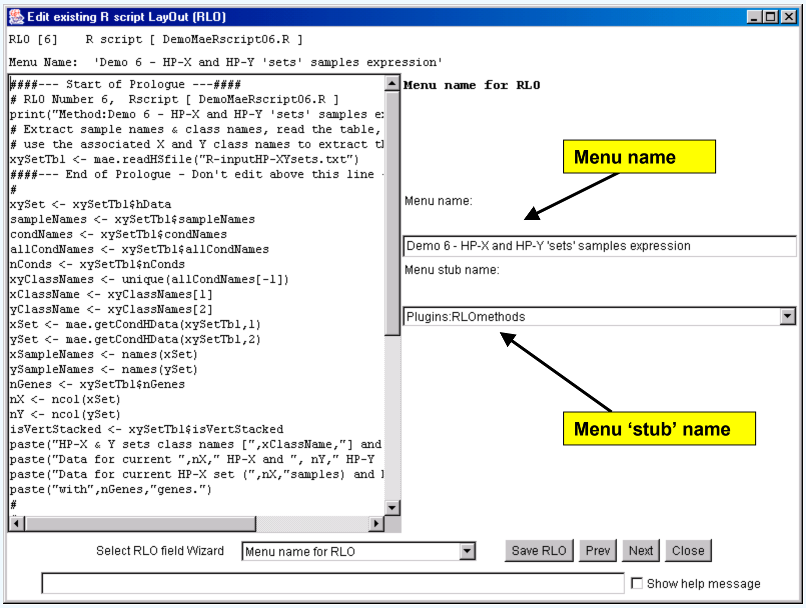
Select RLO field Wizard: R script file name [.R file]

R script for the current RLO

R script name

User selects a wizard to edit one of the RLO fields or use 'Prev' or 'Next' buttons

2. Edit Wizard for Current RLO: Menu name - for RtestPlugin or MAExplorer



RLO [6]   R script [DemoMaeRscript06.R]

Menu Name: 'Demo 6 - HP-X and HP-Y 'sets' samples expression'

```
####--- Start of Prologue ---####
# RLO Number 6,  Rscript [ DemoMaeRscript06.R ]
print("Method:Demo 6 - HP-X and HP-Y 'sets' samples e:
# Extract sample names & class names, read the table,
# use the associated X and Y class names to extract t]
xySetTbl <- mae.readHSfile("R-inputHP-XYsets.txt")
####--- End of Prologue - Don't edit above this line
#
xySet <- xySetTbl$hData
sampleNames <- xySetTbl$sampleNames
condNames <- xySetTbl$condNames
allCondNames <- xySetTbl$allCondNames
nConds <- xySetTbl$nConds
xyClassNames <- unique(allCondNames[-1])
xClassName <- xyClassNames[1]
yClassName <- xyClassNames[2]
xSet <- mae.getCondHData(xySetTbl,1)
ySet <- mae.getCondHData(xySetTbl,2)
xSampleNames <- names(xSet)
ySampleNames <- names(ySet)
nGenes <- xySetTbl$nGenes
nX <- ncol(xSet)
nY <- ncol(ySet)
isVertStacked <- xySetTbl$isVertStacked
paste("HP-X & Y sets class names [",xClassName,"] and
paste("Data for current ",nX," HP-X and ", nY," HP-Y
paste("Data for current HP-X set (",nX,"samples) and ]
paste("with",nGenes,"genes.")
#
```

Menu name for RLO

Menu name

Menu name: Demo 6 - HP-X and HP-Y 'sets' samples expression

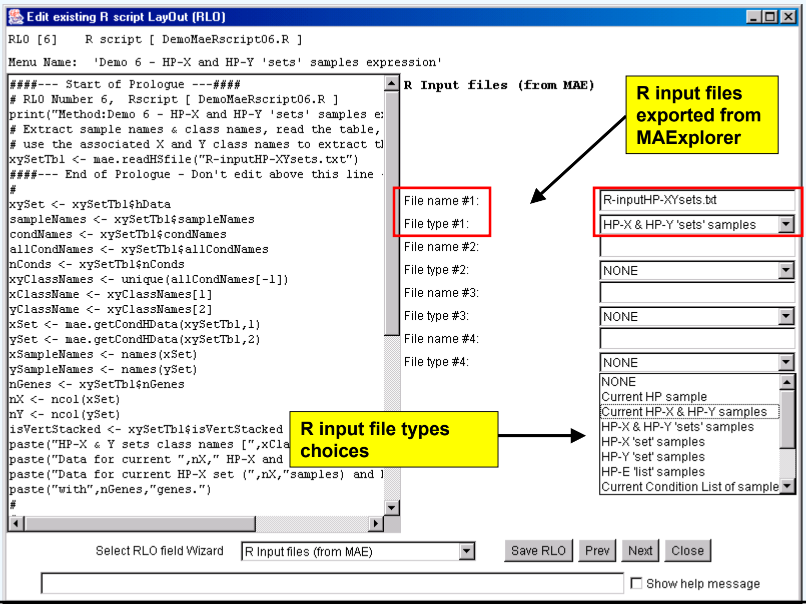
Menu stub name: Plugins:RLOmethods

Menu 'stub' name

Select RLO field Wizard: Menu name for RLO

☐ Show help message

3. Edit Wizard for Current RLO: R Input Files - from MAExplorer



RLO [6]   R script [DemoMaeRscript06.R]

Menu Name: 'Demo 6 - HP-X and HP-Y 'sets' samples expression'

```
####--- Start of Prologue ---####
# RLO Number 6,  Rscript [ DemoMaeRscript06.R ]
print("Method:Demo 6 - HP-X and HP-Y 'sets' samples e:
# Extract sample names & class names, read the table,
# use the associated X and Y class names to extract t]
xySetTbl <- mae.readHSfile("R-inputHP-XYsets.txt")
####--- End of Prologue - Don't edit above this line
#
xySet <- xySetTbl$hData
sampleNames <- xySetTbl$sampleNames
condNames <- xySetTbl$condNames
allCondNames <- xySetTbl$allCondNames
nConds <- xySetTbl$nConds
xyClassNames <- unique(allCondNames[-1])
xClassName <- xyClassNames[1]
yClassName <- xyClassNames[2]
xSet <- mae.getCondHData(xySetTbl,1)
ySet <- mae.getCondHData(xySetTbl,2)
xSampleNames <- names(xSet)
ySampleNames <- names(ySet)
nGenes <- xySetTbl$nGenes
nX <- ncol(xSet)
nY <- ncol(ySet)
isVertStacked <- xySetTbl$isVertStacked
paste("HP-X & Y sets class names [",xCla
paste("Data for current ",nX," HP-X and
paste("Data for current HP-X set (",nX,"samples) and ]
paste("with",nGenes,"genes.")
#
```

R Input files (from MAE)

R input files exported from MAExplorer

File name #1: R-inputHP-XYsets.txt

File type #1: HP-X & HP-Y 'sets' samples

File name #2:

File type #2: NONE

File name #3:

File type #3: NONE

File name #4:

File type #4: NONE

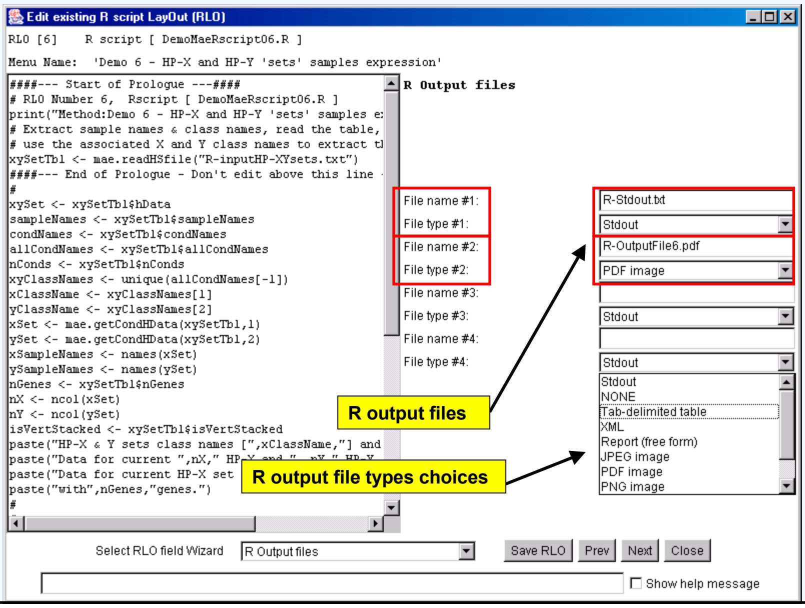
- NONE
- Current HP sample
- Current HP-X & HP-Y samples
- HP-X & HP-Y 'sets' samples
- HP-X 'set' samples
- HP-Y 'set' samples
- HP-E 'list' samples
- Current Condition List of sample

R input file types choices

Select RLO field Wizard: R Input files (from MAE)

☐ Show help message

4. Edit Wizard for Current RLO: R <u>Output</u> Files



```
RLO [6]    R script [ DemoMaeRscript06.R ]

Menu Name:  'Demo 6 - HP-X and HP-Y 'sets' samples expression'
```

```
####--- Start of Prologue ---####
# RLO Number 6,  Rscript [ DemoMaeRscript06.R ]
print("Method:Demo 6 - HP-X and HP-Y 'sets' samples e
# Extract sample names & class names, read the table,
# use the associated X and Y class names to extract t
xySetTbl <- mae.readHSfile("R-inputHP-XYsets.txt")
####--- End of Prologue - Don't edit above this line
#
xySet <- xySetTbl$hData
sampleNames <- xySetTbl$sampleNames
condNames <- xySetTbl$condNames
allCondNames <- xySetTbl$allCondNames
nConds <- xySetTbl$nConds
xyClassNames <- unique(allCondNames[-1])
xClassName <- xyClassNames[1]
yClassName <- xyClassNames[2]
xSet <- mae.getCondHData(xySetTbl,1)
ySet <- mae.getCondHData(xySetTbl,2)
xSampleNames <- names(xSet)
ySampleNames <- names(ySet)
nGenes <- xySetTbl$nGenes
nX <- ncol(xSet)
nY <- ncol(ySet)
isVertStacked <- xySetTbl$isVertStacked
paste("HP-X & Y sets class names [",xClassName,"] and
paste("Data for current ",nX," HP
paste("Data for current HP-X set
paste("with",nGenes,"genes.")
#
```

R Output files

File name #1: R-Stdout.txt
File type #1: Stdout
File name #2: R-OutputFile6.pdf
File type #2: PDF image
File name #3:
File type #3: Stdout
File name #4:
File type #4: Stdout

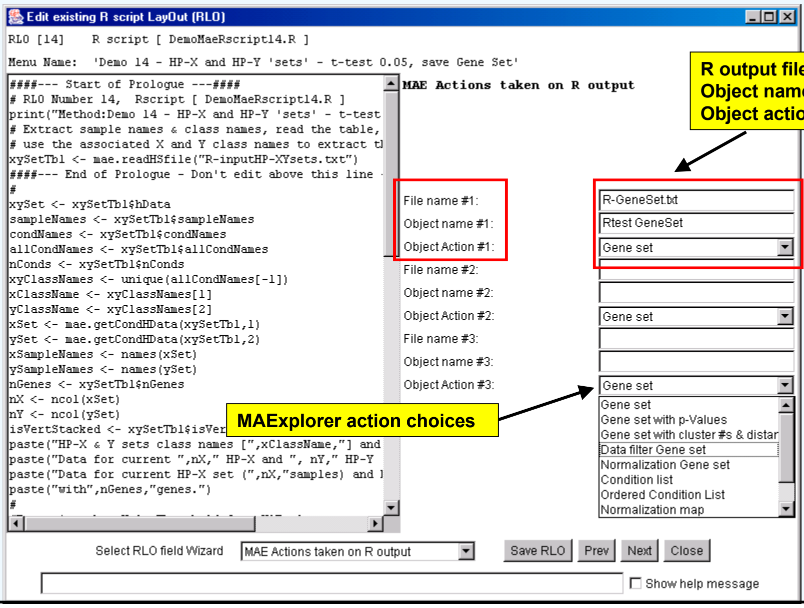
R output files

R output file types choices

Select RLO field Wizard: R Output files

☐ Show help message

5. Edit Wizard for Current RLO: MAExplorer <u>Actions</u> Applied to R Output Files



```
RLO [14]    R script [ DemoMaeRscript14.R ]

Menu Name:  'Demo 14 - HP-X and HP-Y 'sets' - t-test 0.05, save Gene Set'
```

```
####--- Start of Prologue ---####
# RLO Number 14,  Rscript [ DemoMaeRscript14.R ]
print("Method:Demo 14 - HP-X and HP-Y 'sets' - t-test
# Extract sample names & class names, read the table,
# use the associated X and Y class names to extract t
xySetTbl <- mae.readHSfile("R-inputHP-XYsets.txt")
####--- End of Prologue - Don't edit above this line
#
xySet <- xySetTbl$hData
sampleNames <- xySetTbl$sampleNames
condNames <- xySetTbl$condNames
allCondNames <- xySetTbl$allCondNames
nConds <- xySetTbl$nConds
xyClassNames <- unique(allCondNames[-1])
xClassName <- xyClassNames[1]
yClassName <- xyClassNames[2]
xSet <- mae.getCondHData(xySetTbl,1)
ySet <- mae.getCondHData(xySetTbl,2)
xSampleNames <- names(xSet)
ySampleNames <- names(ySet)
nGenes <- xySetTbl$nGenes
nX <- ncol(xSet)
nY <- ncol(ySet)
isVertStacked <- xySetTbl$isVer
paste("HP-X & Y sets class names [",xClassName,"] and
paste("Data for current ",nX," HP-X and ", nY," HP-Y
paste("Data for current HP-X set (",nX,"samples) and
paste("with",nGenes,"genes.")
#
```

MAE Actions taken on R output

R output file
Object name
Object action

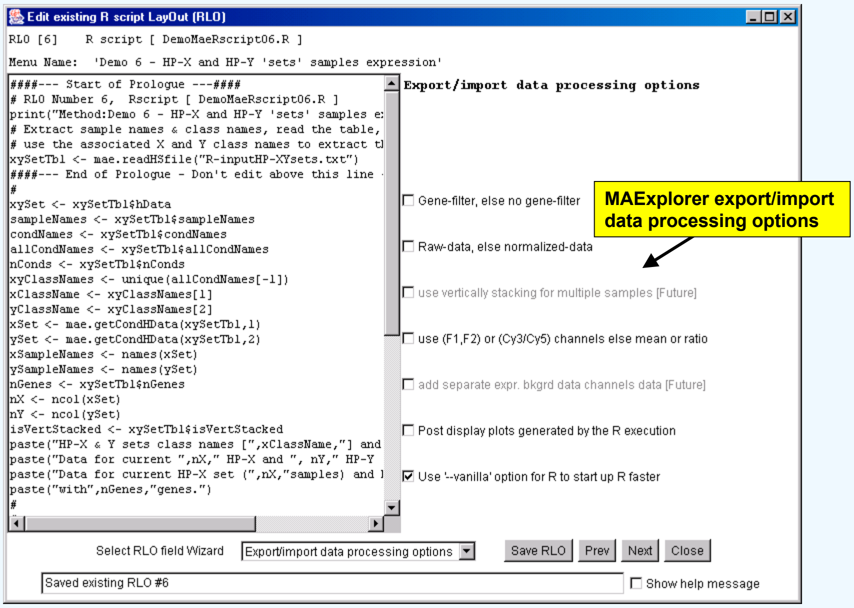
File name #1: R-GeneSet.txt
Object name #1: Rtest GeneSet
Object Action #1: Gene set
File name #2:
Object name #2:
Object Action #2: Gene set
File name #3:
Object name #3:
Object Action #3: Gene set

MAExplorer action choices

Select RLO field Wizard: MAE Actions taken on R output

☐ Show help message

6. Edit Wizard for Current RLO: MAExplorer Export/Import Data Processing



```
RLO [6]    R script [ DemoMaeRscript06.R ]
Menu Name:  'Demo 6 - HP-X and HP-Y 'sets' samples expression'
```

```
####--- Start of Prologue ---####
# RLO Number 6,  Rscript [ DemoMaeRscript06.R ]
print("Method:Demo 6 - HP-X and HP-Y 'sets' samples e
# Extract sample names & class names, read the table,
# use the associated X and Y class names to extract t
xySetTbl <- mae.readHSfile("R-inputHP-XYsets.txt")
####--- End of Prologue - Don't edit above this line
#
xySet <- xySetTbl$hData
sampleNames <- xySetTbl$sampleNames
condNames <- xySetTbl$condNames
allCondNames <- xySetTbl$allCondNames
nConds <- xySetTbl$nConds
xyClassNames <- unique(allCondNames[-1])
xClassName <- xyClassNames[1]
yClassName <- xyClassNames[2]
xSet <- mae.getCondHData(xySetTbl,1)
ySet <- mae.getCondHData(xySetTbl,2)
xSampleNames <- names(xSet)
ySampleNames <- names(ySet)
nGenes <- xySetTbl$nGenes
nX <- ncol(xSet)
nY <- ncol(ySet)
isVertStacked <- xySetTbl$isVertStacked
paste("HP-X & Y sets class names [",xClassName,"] and
paste("Data for current ",nX," HP-X and ", nY," HP-Y
paste("Data for current HP-X set (",nX,"samples) and
paste("with",nGenes,"genes.")
#
```

Export/import data processing options

- ☐ Gene-filter, else no gene-filter
- ☐ Raw-data, else normalized-data
- ☐ use vertically stacking for multiple samples [Future]
- ☐ use (F1,F2) or (Cy3/Cy5) channels else mean or ratio
- ☐ add separate expr. bkgrd data channels data [Future]
- ☐ Post display plots generated by the R execution
- ☑ Use '--vanilla' option for R to start up R faster

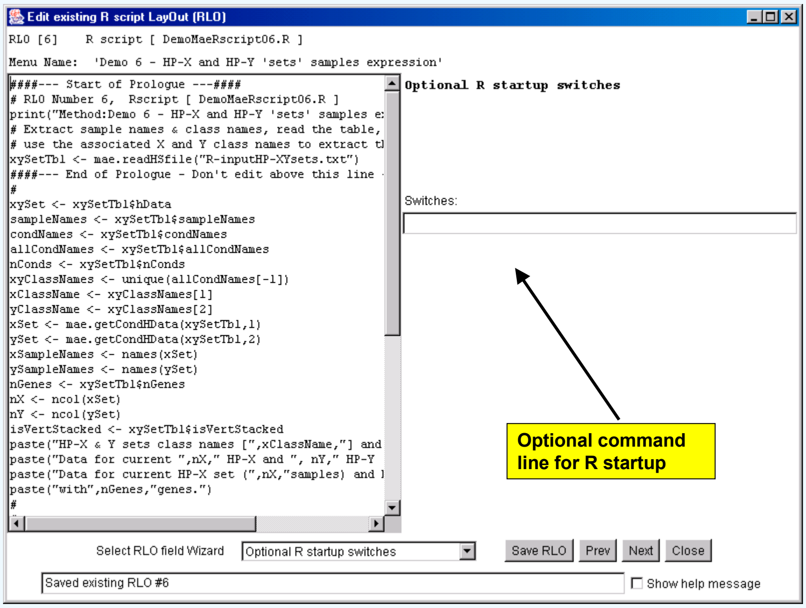
Select RLO field Wizard: Export/import data processing options

Saved existing RLO #6

☐ Show help message

MAExplorer export/import data processing options

7. Edit Wizard for Current RLO: Optional R Startup Switches



```
RLO [6]    R script [ DemoMaeRscript06.R ]
Menu Name:  'Demo 6 - HP-X and HP-Y 'sets' samples expression'
```

```
####--- Start of Prologue ---####
# RLO Number 6,  Rscript [ DemoMaeRscript06.R ]
print("Method:Demo 6 - HP-X and HP-Y 'sets' samples e
# Extract sample names & class names, read the table,
# use the associated X and Y class names to extract t
xySetTbl <- mae.readHSfile("R-inputHP-XYsets.txt")
####--- End of Prologue - Don't edit above this line
#
xySet <- xySetTbl$hData
sampleNames <- xySetTbl$sampleNames
condNames <- xySetTbl$condNames
allCondNames <- xySetTbl$allCondNames
nConds <- xySetTbl$nConds
xyClassNames <- unique(allCondNames[-1])
xClassName <- xyClassNames[1]
yClassName <- xyClassNames[2]
xSet <- mae.getCondHData(xySetTbl,1)
ySet <- mae.getCondHData(xySetTbl,2)
xSampleNames <- names(xSet)
ySampleNames <- names(ySet)
nGenes <- xySetTbl$nGenes
nX <- ncol(xSet)
nY <- ncol(ySet)
isVertStacked <- xySetTbl$isVertStacked
paste("HP-X & Y sets class names [",xClassName,"] and
paste("Data for current ",nX," HP-X and ", nY," HP-Y
paste("Data for current HP-X set (",nX,"samples) and
paste("with",nGenes,"genes.")
#
```

Optional R startup switches

Switches:

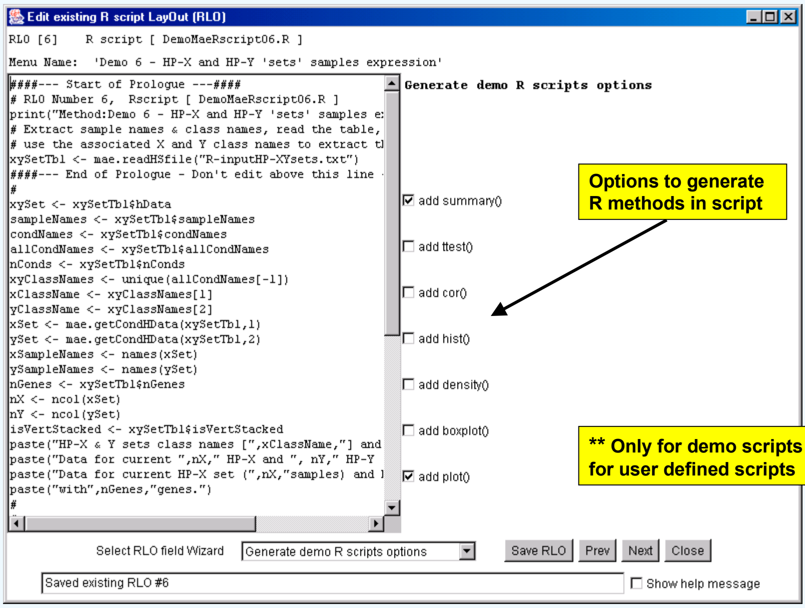
Select RLO field Wizard: Optional R startup switches

Saved existing RLO #6

☐ Show help message

Optional command line for R startup

8. Edit Wizard for Current RLO: Generate Demo R Script Methods**



RLO [6]   R script [DemoMaeRscript06.R]

Menu Name: 'Demo 6 - HP-X and HP-Y 'sets' samples expression'

```
####--- Start of Prologue ---####
# RLO Number 6,  Rscript [ DemoMaeRscript06.R ]
print("Method:Demo 6 - HP-X and HP-Y 'sets' samples e
# Extract sample names & class names, read the table,
# use the associated X and Y class names to extract t
xySetTbl <- mae.readHSfile("R-inputHP-XYsets.txt")
####--- End of Prologue - Don't edit above this line
#
xySet <- xySetTbl$hData
sampleNames <- xySetTbl$sampleNames
condNames <- xySetTbl$condNames
allCondNames <- xySetTbl$allCondNames
nConds <- xySetTbl$nConds
xyClassNames <- unique(allCondNames[-1])
xClassName <- xyClassNames[1]
yClassName <- xyClassNames[2]
xSet <- mae.getCondHData(xySetTbl,1)
ySet <- mae.getCondHData(xySetTbl,2)
xSampleNames <- names(xSet)
ySampleNames <- names(ySet)
nGenes <- xySetTbl$nGenes
nX <- ncol(xSet)
nY <- ncol(ySet)
isVertStacked <- xySetTbl$isVertStacked
paste("HP-X & Y sets class names [",xClassName,"] and
paste("Data for current ",nX," HP-X and ", nY," HP-Y
paste("Data for current HP-X set (",nX,"samples) and
paste("with",nGenes,"genes.")
#
```

Generate demo R scripts options

☑ add summary()

☐ add ttest()

☐ add cor()

☐ add hist()

☐ add density()

☐ add boxplot()

☑ add plot()

Select RLO field Wizard: Generate demo R scripts options

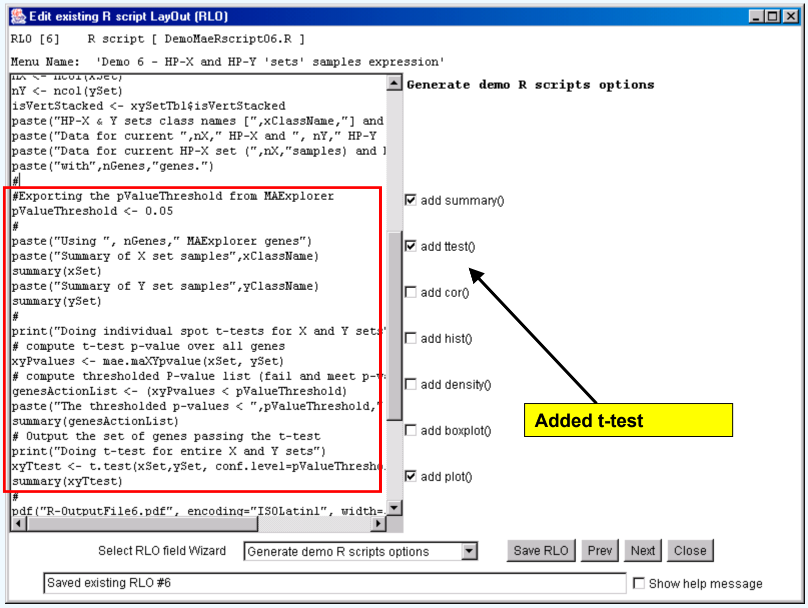
Saved existing RLO #6

☐ Show help message

Options to generate R methods in script

**** Only for demo scripts - not for user defined scripts**

Example: Edit Current Demo RLO Generate Demo Methods - Add t-test



RLO [6]   R script [DemoMaeRscript06.R]

Menu Name: 'Demo 6 - HP-X and HP-Y 'sets' samples expression'

```
nY <- ncol(ySet)
isVertStacked <- xySetTbl$isVertStacked
paste("HP-X & Y sets class names [",xClassName,"] and
paste("Data for current ",nX," HP-X and ", nY," HP-Y
paste("Data for current HP-X set (",nX,"samples) and
paste("with",nGenes,"genes.")
#
#Exporting the pValueThreshold from MAExplorer
pValueThreshold <- 0.05
#
paste("Using ", nGenes," MAExplorer genes")
paste("Summary of X set samples",xClassName)
summary(xSet)
paste("Summary of Y set samples",yClassName)
summary(ySet)
#
print("Doing individual spot t-tests for X and Y sets
# compute t-test p-value over all genes
xyPvalues <- mae.maXYpvalue(xSet, ySet)
# compute thresholded P-value list (fail and meet p-v
genesActionList <- (xyPvalues < pValueThreshold)
paste("The thresholded p-values < ",pValueThreshold,"
summary(genesActionList)
# Output the set of genes passing the t-test
print("Doing t-test for entire X and Y sets")
xyTtest <- t.test(xSet,ySet, conf.level=pValueThresho
summary(xyTtest)
#
pdf("R-OutputFile6.pdf", encoding="ISOLatin1", width=
```

Generate demo R scripts options

☑ add summary()

☑ add ttest()

☐ add cor()

☐ add hist()

☐ add density()

☐ add boxplot()

☑ add plot()

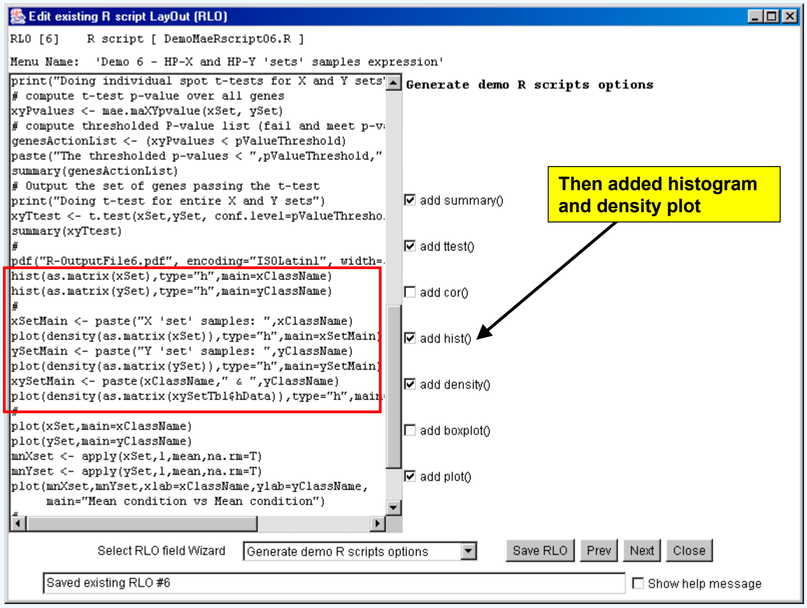
Select RLO field Wizard: Generate demo R scripts options

Saved existing RLO #6

☐ Show help message

Added t-test

Example: Edit Current Demo RLO Generate Demo Methods - Also Add histogram



```
RLO [6]    R script [ DemoMaeRscript06.R ]

Menu Name:  'Demo 6 - HP-X and HP-Y 'sets' samples expression'
```

```
print("Doing individual spot t-tests for X and Y sets'
# compute t-test p-value over all genes
xyPvalues <- mae.maXYpvalue(xSet, ySet)
# compute thresholded P-value list (fail and meet p-v
genesActionList <- (xyPvalues < pValueThreshold)
paste("The thresholded p-values < ",pValueThreshold,"
summary(genesActionList)
# Output the set of genes passing the t-test
print("Doing t-test for entire X and Y sets")
xyTtest <- t.test(xSet,ySet, conf.level=pValueThresho
summary(xyTtest)
#
pdf("R-OutputFile6.pdf", encoding="ISOLatin1", width=
hist(as.matrix(xSet),type="h",main=xClassName)
hist(as.matrix(ySet),type="h",main=yClassName)
#
xSetMain <- paste("X 'set' samples: ",xClassName)
plot(density(as.matrix(xSet)),type="h",main=xSetMain)
ySetMain <- paste("Y 'set' samples: ",yClassName)
plot(density(as.matrix(ySet)),type="h",main=ySetMain)
xySetMain <- paste(xClassName," & ",yClassName)
plot(density(as.matrix(xySetTbl$hData)),type="h",main
#
plot(xSet,main=xClassName)
plot(ySet,main=yClassName)
mnXset <- apply(xSet,1,mean,na.rm=T)
mnYset <- apply(ySet,1,mean,na.rm=T)
plot(mnXset,mnYset,xlab=xClassName,ylab=yClassName,
     main="Mean condition vs Mean condition")
```

Generate demo R scripts options

☑ add summary()
☑ add ttest()
☐ add cor()
☑ add hist()
☑ add density()
☐ add boxplot()
☑ add plot()

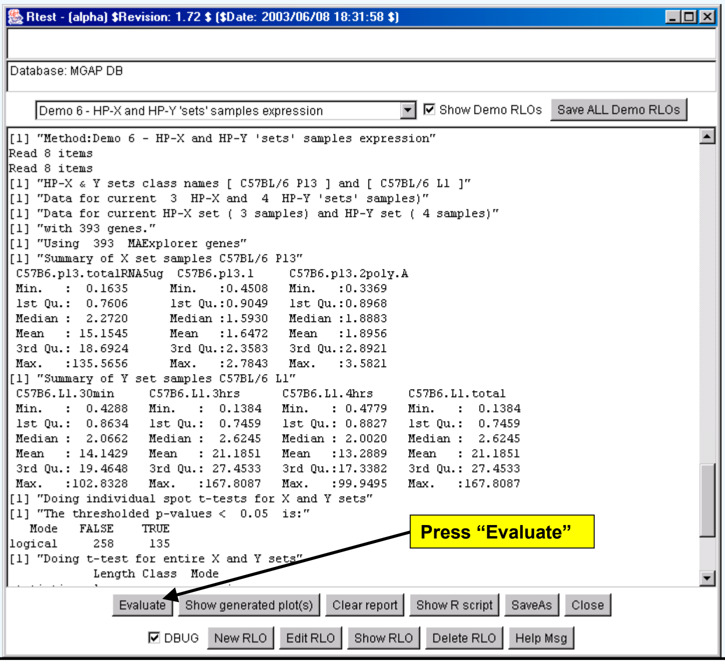
Select RLO field Wizard: Generate demo R scripts options

Saved existing RLO #6

☐ Show help message

Then added histogram and density plot

Evaluate R Script for Current RLO



Database: MGAP DB

Demo 6 - HP-X and HP-Y 'sets' samples expression

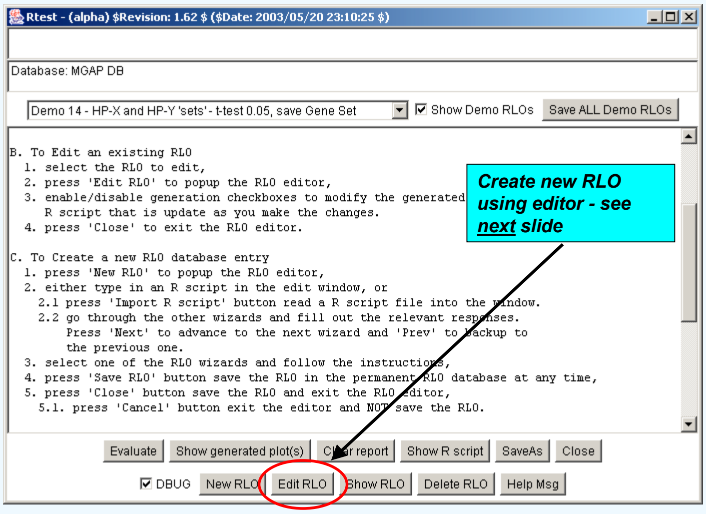
☑ Show Demo RLOs

```
[1] "Method:Demo 6 - HP-X and HP-Y 'sets' samples expression"
Read 8 items
Read 8 items
[1] "HP-X & Y sets class names [ C57BL/6 P13 ] and [ C57BL/6 L1 ]"
[1] "Data for current  3  HP-X and  4  HP-Y 'sets' samples)"
[1] "Data for current HP-X set ( 3 samples) and HP-Y set ( 4 samples)"
[1] "with 393 genes."
[1] "Using  393  MAExplorer genes"
[1] "Summary of X set samples C57BL/6 P13"
 C57B6.p13.totalRNA5ug  C57B6.p13.1     C57B6.p13.2poly.A
 Min.   :  0.1635      Min.   :0.4508   Min.   :0.3369
 1st Qu.:  0.7606      1st Qu.:0.9049   1st Qu.:0.8968
 Median :  2.2720      Median :1.5930   Median :1.8883
 Mean   : 15.1545      Mean   :1.6472   Mean   :1.8956
 3rd Qu.: 18.6924      3rd Qu.:2.3583   3rd Qu.:2.8921
 Max.   :135.5656      Max.   :2.7843   Max.   :3.5821
[1] "Summary of Y set samples C57BL/6 L1"
 C57B6.L1.30min     C57B6.L1.3hrs      C57B6.L1.4hrs      C57B6.L1.total
 Min.   :  0.4288   Min.   :  0.1384   Min.   : 0.4779   Min.   :  0.1384
 1st Qu.:  0.8634   1st Qu.:  0.7459   1st Qu.: 0.8827   1st Qu.:  0.7459
 Median :  2.0662   Median :  2.6245   Median : 2.0020   Median :  2.6245
 Mean   : 14.1429   Mean   : 21.1851   Mean   :13.2889   Mean   : 21.1851
 3rd Qu.: 19.4648   3rd Qu.: 27.4533   3rd Qu.:17.3382   3rd Qu.: 27.4533
 Max.   :102.8328   Max.   :167.8087   Max.   :99.9495   Max.   :167.8087
[1] "Doing individual spot t-tests for X and Y sets"
[1] "The thresholded p-values <  0.05  is:"
   Mode   FALSE    TRUE
logical     258     135
[1] "Doing t-test for entire X and Y sets"
            Length Class  Mode
```

☑ DBUG

Press “Evaluate”

Creating a New User-defined RLO

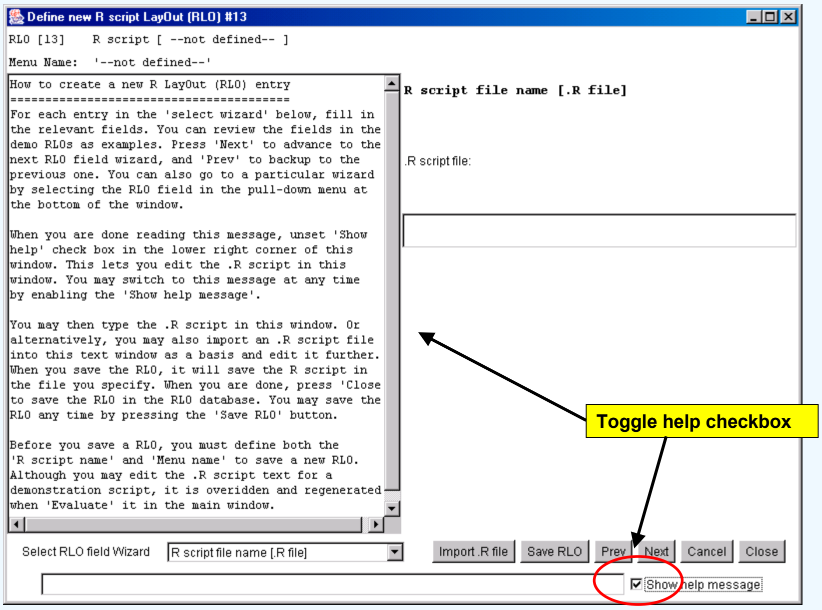


Database: MGAP DB

```
B. To Edit an existing RLO
  1. select the RLO to edit,
  2. press 'Edit RLO' to popup the RLO editor,
  3. enable/disable generation checkboxes to modify the generated
     R script that is update as you make the changes.
  4. press 'Close' to exit the RLO editor.

C. To Create a new RLO database entry
  1. press 'New RLO' to popup the RLO editor,
  2. either type in an R script in the edit window, or
    2.1 press 'Import R script' button read a R script file into the window.
    2.2 go through the other wizards and fill out the relevant responses.
        Press 'Next' to advance to the next wizard and 'Prev' to backup to
        the previous one.
  3. select one of the RLO wizards and follow the instructions,
  4. press 'Save RLO' button save the RLO in the permanent RLO database at any time,
  5. press 'Close' button save the RLO and exit the RLO editor,
    5.1. press 'Cancel' button exit the editor and NOT save the RLO.
```

Create new RLO using editor - see <u>next</u> slide

View Help Message for RLO Wizard Editor



```
RLO [13]    R script [ --not defined-- ]

Menu Name:  '--not defined--'
```

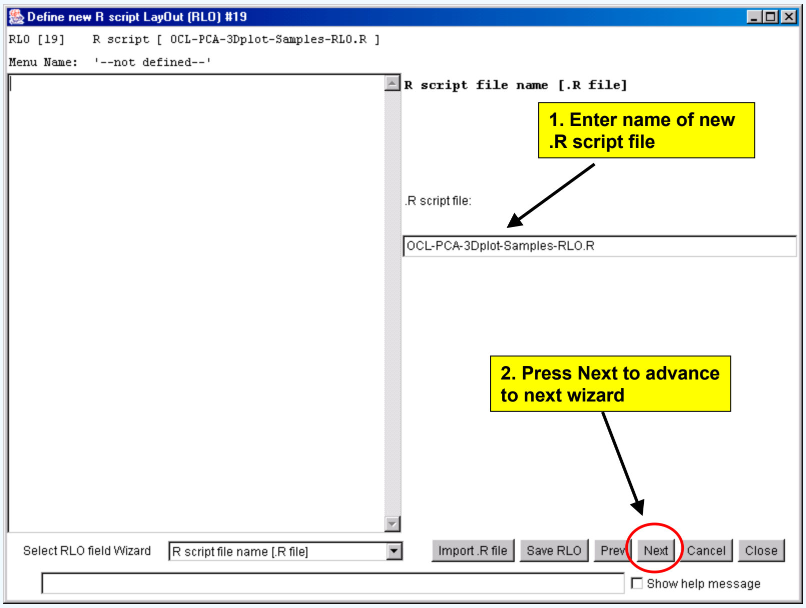
```
How to create a new R LayOut (RLO) entry
==========================================
For each entry in the 'select wizard' below, fill in
the relevant fields. You can review the fields in the
demo RLOs as examples. Press 'Next' to advance to the
next RLO field wizard, and 'Prev' to backup to the
previous one. You can also go to a particular wizard
by selecting the RLO field in the pull-down menu at
the bottom of the window.

When you are done reading this message, unset 'Show
help' check box in the lower right corner of this
window. This lets you edit the .R script in this
window. You may switch to this message at any time
by enabling the 'Show help message'.

You may then type the .R script in this window. Or
alternatively, you may also import an .R script file
into this text window as a basis and edit it further.
When you save the RLO, it will save the R script in
the file you specify. When you are done, press 'Close
to save the RLO in the RLO database. You may save the
RLO any time by pressing the 'Save RLO' button.

Before you save a RLO, you must define both the
'R script name' and 'Menu name' to save a new RLO.
Although you may edit the .R script text for a
demonstration script, it is overidden and regenerated
when 'Evaluate' it in the main window.
```

R script file name [.R file]

.R script file:

Toggle help checkbox

1. Define New R script name Using RLO Wizard



Define new R script LayOut (RLO) #19

```
RLO [19]    R script [ OCL-PCA-3Dplot-Samples-RLO.R ]
Menu Name:  '--not defined--'
```

R script file name [.R file]

1. Enter name of new .R script file

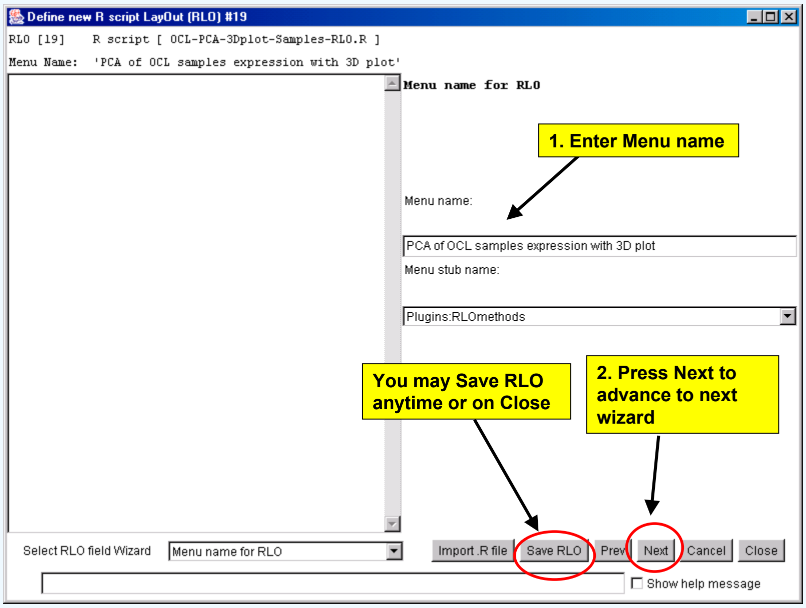
.R script file: OCL-PCA-3Dplot-Samples-RLO.R

2. Press Next to advance to next wizard

Select RLO field Wizard: R script file name [.R file]

☐ Show help message

2. Define New RLO Menu name Using Wizard



Define new R script LayOut (RLO) #19

```
RLO [19]    R script [ OCL-PCA-3Dplot-Samples-RLO.R ]
Menu Name:  'PCA of OCL samples expression with 3D plot'
```

Menu name for RLO

1. Enter Menu name

Menu name: PCA of OCL samples expression with 3D plot

Menu stub name: Plugins:RLOmethods

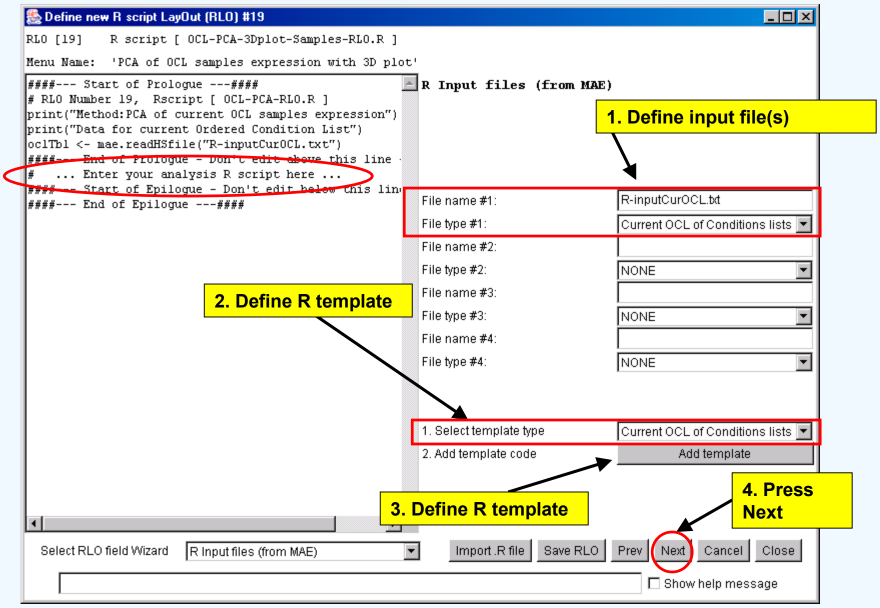
You may Save RLO anytime or on Close

2. Press Next to advance to next wizard

Select RLO field Wizard: Menu name for RLO

☐ Show help message

3. Define New RLO R Input files and R template Using Wizard



Define new R script LayOut (RLO) #19

```
RLO [19]    R script [ OCL-PCA-3Dplot-Samples-RLO.R ]

Menu Name:  'PCA of OCL samples expression with 3D plot'
```

```
####--- Start of Prologue ---####
# RLO Number 19,  Rscript [ OCL-PCA-RLO.R ]
print("Method:PCA of current OCL samples expression")
print("Data for current Ordered Condition List")
oclTbl <- mae.readHSfile("R-inputCurOCL.txt")
####--- End of Prologue - Don't edit above this line -
#   ... Enter your analysis R script here ...
####--- Start of Epilogue - Don't edit below this lin
####--- End of Epilogue ---####
```

R Input files (from MAE)

File name #1: R-inputCurOCL.txt
File type #1: Current OCL of Conditions lists
File name #2:
File type #2: NONE
File name #3:
File type #3: NONE
File name #4:
File type #4: NONE

1. Select template type: Current OCL of Conditions lists
2. Add template code: Add template

Select RLO field Wizard: R Input files (from MAE)

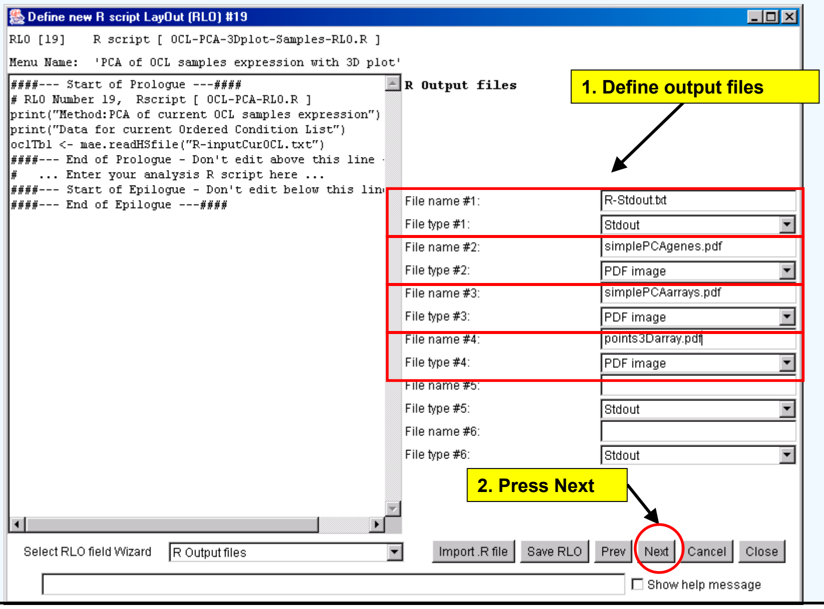
1. Define input file(s)

2. Define R template

3. Define R template

4. Press Next

4. Define New RLO R Output files Using Wizard



Define new R script LayOut (RLO) #19

```
RLO [19]    R script [ OCL-PCA-3Dplot-Samples-RLO.R ]

Menu Name:  'PCA of OCL samples expression with 3D plot'
```

```
####--- Start of Prologue ---####
# RLO Number 19,  Rscript [ OCL-PCA-RLO.R ]
print("Method:PCA of current OCL samples expression")
print("Data for current Ordered Condition List")
oclTbl <- mae.readHSfile("R-inputCurOCL.txt")
####--- End of Prologue - Don't edit above this line -
#   ... Enter your analysis R script here ...
####--- Start of Epilogue - Don't edit below this lin
####--- End of Epilogue ---####
```

R Output files

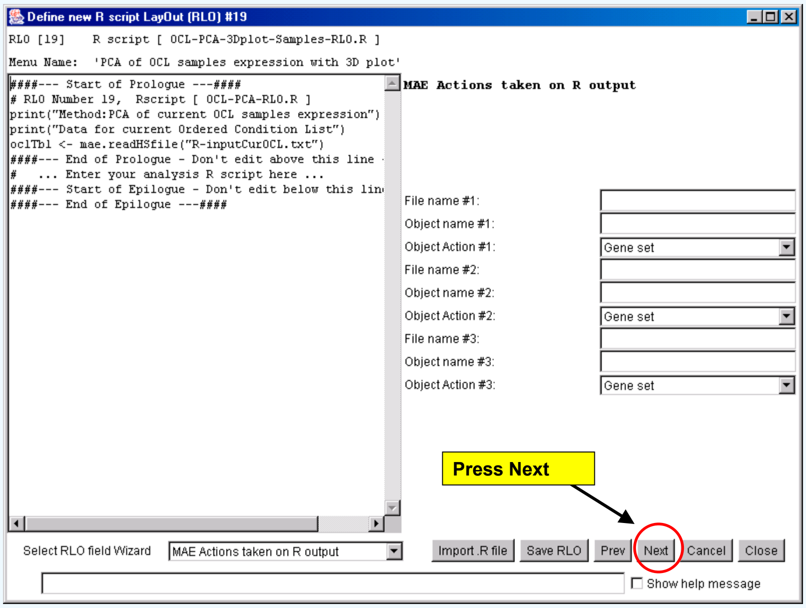
File name #1: R-Stdout.txt
File type #1: Stdout
File name #2: simplePCAgenes.pdf
File type #2: PDF image
File name #3: simplePCAarrays.pdf
File type #3: PDF image
File name #4: points3Darray.pdf
File type #4: PDF image
File name #5:
File type #5: Stdout
File name #6:
File type #6: Stdout

Select RLO field Wizard: R Output files

1. Define output files

2. Press Next

5. Define New RLO Actions Using Wizard - If Any



RLO [19]   R script [OCL-PCA-3Dplot-Samples-RLO.R]

Menu Name: 'PCA of OCL samples expression with 3D plot'

```
####--- Start of Prologue ---####
# RLO Number 19,  Rscript [ OCL-PCA-RLO.R ]
print("Method:PCA of current OCL samples expression")
print("Data for current Ordered Condition List")
oclTbl <- mae.readHSfile("R-inputCurOCL.txt")
####--- End of Prologue - Don't edit above this line
#    ... Enter your analysis R script here ...
####--- Start of Epilogue - Don't edit below this lin
####--- End of Epilogue ---####
```

MAE Actions taken on R output

File name #1:

Object name #1:

Object Action #1: Gene set

File name #2:

Object name #2:

Object Action #2: Gene set

File name #3:

Object name #3:

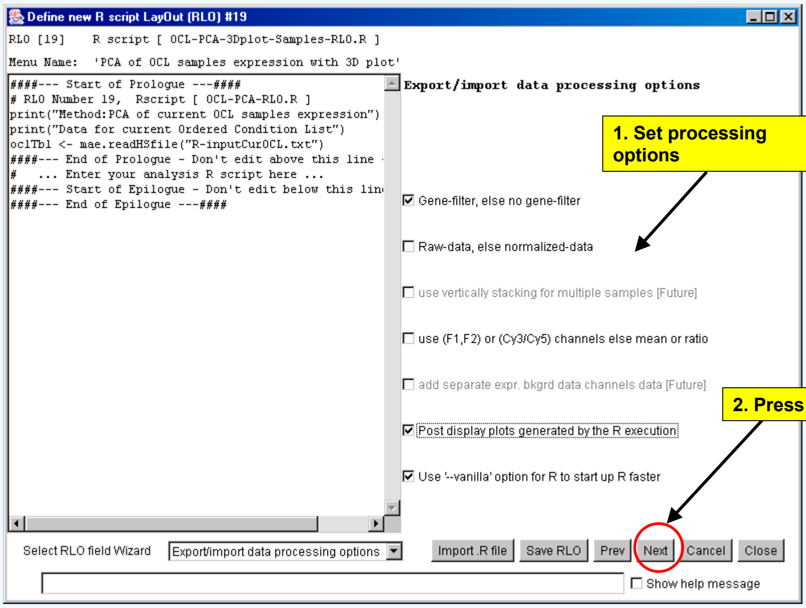
Object Action #3: Gene set

Select RLO field Wizard: MAE Actions taken on R output

☐ Show help message

Press Next

6. Define New RLO Export/Import Data Processing Options Using Wizard



RLO [19]   R script [OCL-PCA-3Dplot-Samples-RLO.R]

Menu Name: 'PCA of OCL samples expression with 3D plot'

```
####--- Start of Prologue ---####
# RLO Number 19,  Rscript [ OCL-PCA-RLO.R ]
print("Method:PCA of current OCL samples expression")
print("Data for current Ordered Condition List")
oclTbl <- mae.readHSfile("R-inputCurOCL.txt")
####--- End of Prologue - Don't edit above this line
#    ... Enter your analysis R script here ...
####--- Start of Epilogue - Don't edit below this lin
####--- End of Epilogue ---####
```

Export/import data processing options

☑ Gene-filter, else no gene-filter

☐ Raw-data, else normalized-data

☐ use vertically stacking for multiple samples [Future]

☐ use (F1,F2) or (Cy3/Cy5) channels else mean or ratio

☐ add separate expr. bkgrd data channels data [Future]

☑ Post display plots generated by the R execution

☑ Use '--vanilla' option for R to start up R faster

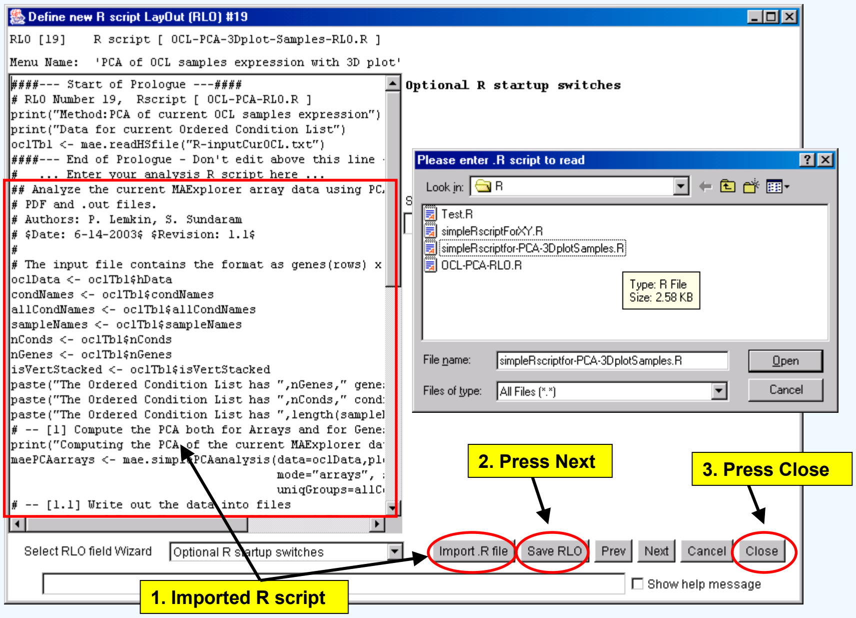
Select RLO field Wizard: Export/import data processing options

☐ Show help message

1. Set processing options

2. Press Next

Import R Code From file(s) to New RLO



Define new R script LayOut (RLO) #19

```
RLO [19]    R script [ OCL-PCA-3Dplot-Samples-RLO.R ]

Menu Name:  'PCA of OCL samples expression with 3D plot'
```

```
####--- Start of Prologue ---####
# RLO Number 19,  Rscript [ OCL-PCA-RLO.R ]
print("Method:PCA of current OCL samples expression")
print("Data for current Ordered Condition List")
oclTbl <- mae.readHSfile("R-inputCurOCL.txt")
####--- End of Prologue - Don't edit above this line
#    ... Enter your analysis R script here ...
## Analyze the current MAExplorer array data using PC
# PDF and .out files.
# Authors: P. Lemkin, S. Sundaram
# $Date: 6-14-2003$ $Revision: 1.1$
#
# The input file contains the format as genes(rows) x
oclData <- oclTbl$hData
condNames <- oclTbl$condNames
allCondNames <- oclTbl$allCondNames
sampleNames <- oclTbl$sampleNames
nConds <- oclTbl$nConds
nGenes <- oclTbl$nGenes
isVertStacked <- oclTbl$isVertStacked
paste("The Ordered Condition List has ",nGenes," gene
paste("The Ordered Condition List has ",nConds," cond
paste("The Ordered Condition List has ",length(sample
# -- [1] Compute the PCA both for Arrays and for Gene
print("Computing the PCA of the current MAExplorer da
maePCAarrays <- mae.simplePCAanalysis(data=oclData,pl
                                      mode="arrays",
                                      uniqGroups=allC
# -- [1.1] Write out the data into files
```

Optional R startup switches

Please enter .R script to read

Look in: R

File name: simpleRscriptfor-PCA-3DplotSamples.R

Files of type: All Files (*.*)

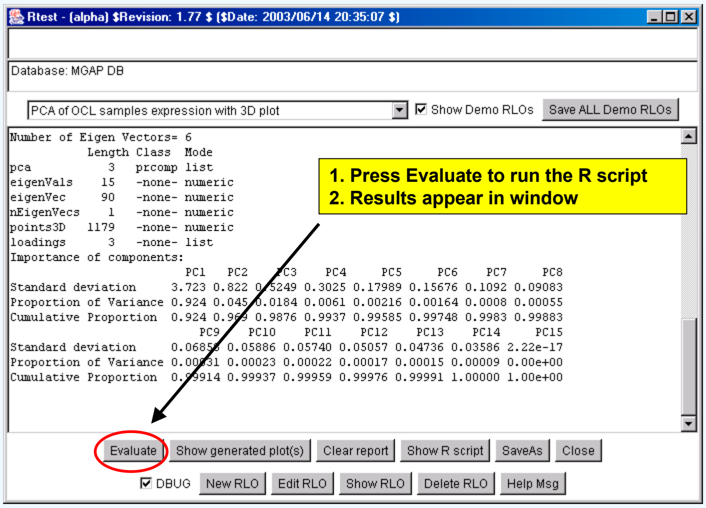
Select RLO field Wizard: Optional R startup switches

1. Imported R script

2. Press Next

3. Press Close

Executing the New RLO from RtestPlugin



Rtest - (alpha) $Revision: 1.77 $ ($Date: 2003/06/14 20:35:07 $)

Database: MGAP DB

PCA of OCL samples expression with 3D plot

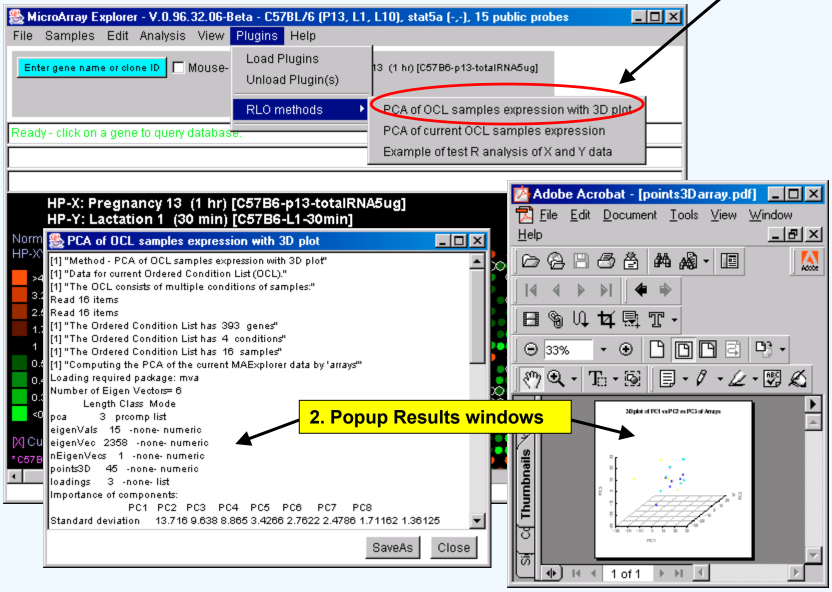
☑ Show Demo RLOs

```
Number of Eigen Vectors= 6
           Length Class  Mode
pca          3    prcomp list
eigenVals   15    -none- numeric
eigenVec    90    -none- numeric
nEigenVecs   1    -none- numeric
points3D  1179    -none- numeric
loadings     3    -none- list
Importance of components:
                         PC1   PC2    PC3    PC4     PC5     PC6    PC7     PC8
Standard deviation     3.723 0.822 0.5249 0.3025 0.17989 0.15676 0.1092 0.09083
Proportion of Variance 0.924 0.045 0.0184 0.0061 0.00216 0.00164 0.0008 0.00055
Cumulative Proportion  0.924 0.969 0.9876 0.9937 0.99585 0.99748 0.9983 0.99883
                          PC9    PC10    PC11    PC12    PC13    PC14     PC15
Standard deviation     0.06858 0.05886 0.05740 0.05057 0.04736 0.03586 2.22e-17
Proportion of Variance 0.00031 0.00023 0.00022 0.00017 0.00015 0.00009 0.00e+00
Cumulative Proportion  0.99914 0.99937 0.99959 0.99976 0.99991 1.00000 1.00e+00
```

☑ DBUG

1. Press Evaluate to run the R script
2. Results appear in window

Executing the New RLO From MAExplorer RLO Methods menu

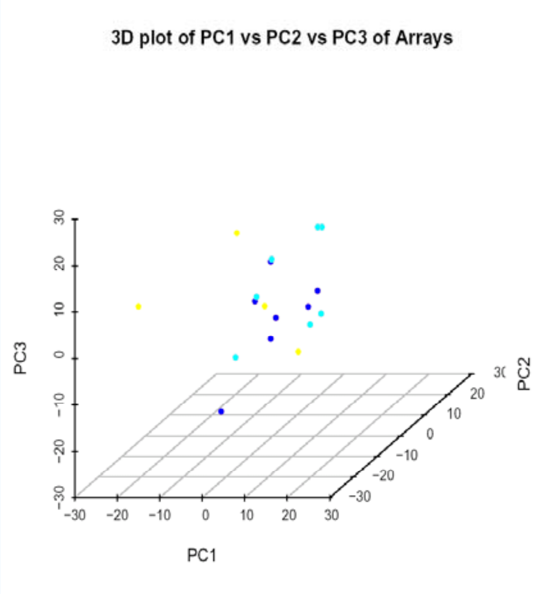


```
[1] "Method - PCA of OCL samples expression with 3D plot"
[1] "Data for current Ordered Condition List (OCL)."
[1] "The OCL consists of multiple conditions of samples:"
Read 16 items
Read 16 items
[1] "The Ordered Condition List has  393  genes"
[1] "The Ordered Condition List has  4  conditions"
[1] "The Ordered Condition List has  16  samples"
[1] "Computing the PCA of the current MAExplorer data by 'arrays'"
Loading required package: mva
Number of Eigen Vectors= 6
          Length Class  Mode
pca          3   prcomp list
eigenVals   15   -none- numeric
eigenVec  2358   -none- numeric
nEigenVecs   1   -none- numeric
points3D    45   -none- numeric
loadings     3   -none- list
Importance of components:
                    PC1   PC2   PC3    PC4    PC5    PC6     PC7     PC8
Standard deviation  13.716 9.638 8.865 3.4266 2.7622 2.4786 1.71162 1.36125
```

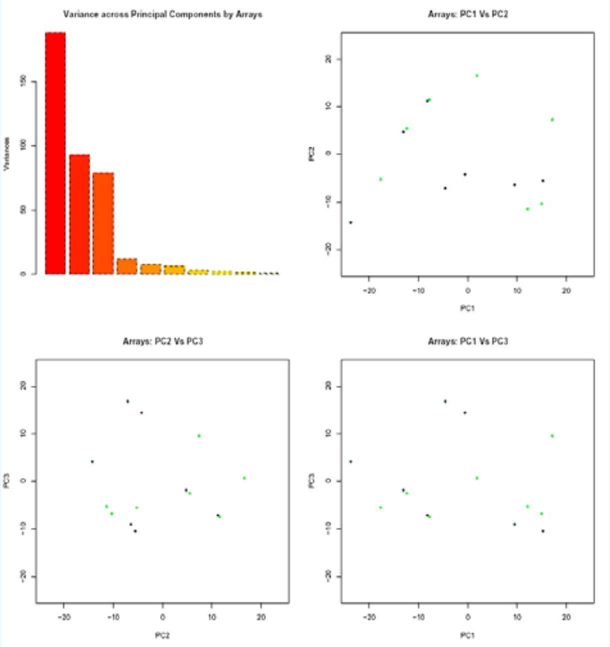
1. Select command

2. Popup Results windows

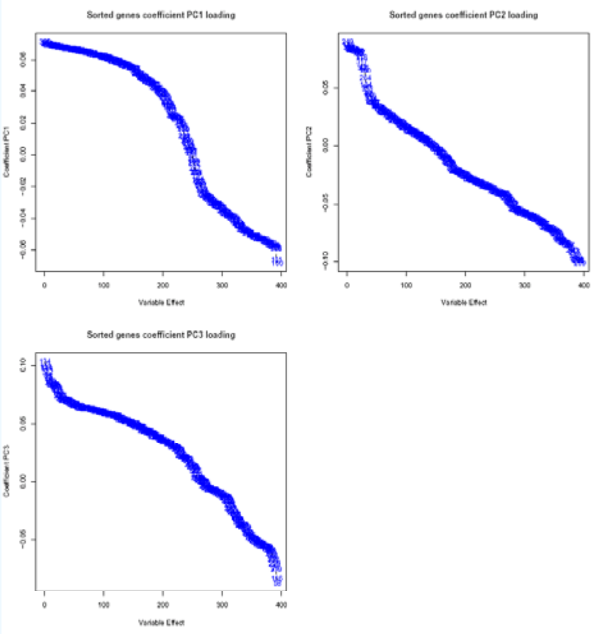
Plot (1 of 5) Generated for New RLO: PCA of Multiple Conditions w/Replicate Samples



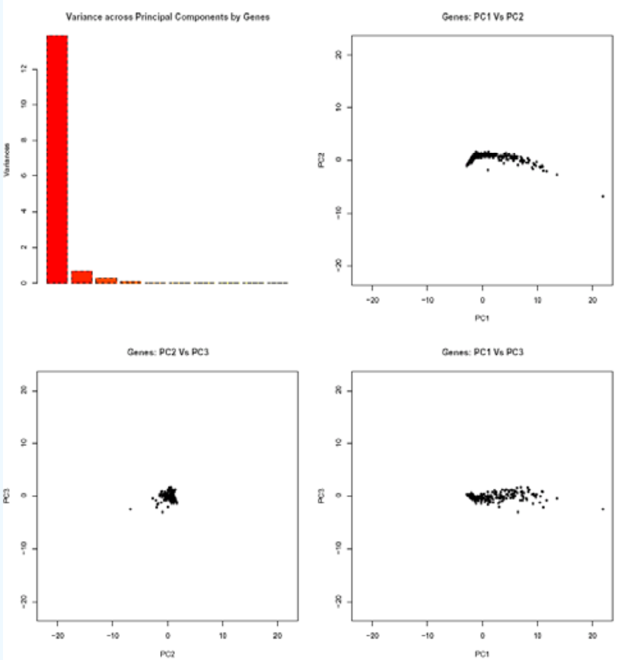
Plot (2 of 5) Variance across PCA by Arrays



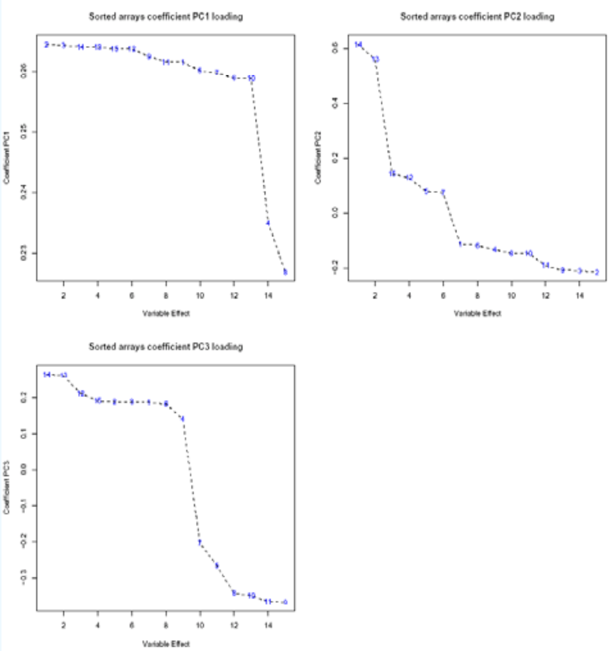
Plot (3 of 5) PCA loadings, by Arrays



Plot (4 of 5) Variance across PCA by Genes



Plot (5 of 5) PCA loadings, by Genes



MAExplorer Java API - MJAReval Class

- MJAReval Application Programming Interface (API) class supports the R evaluation and RLO database functionality
- Define and maintain a permanent R LayOut (RLO) database and evaluate RLOs directly from MAExplorer “RLO methods” menu
- Create new R scripts using RLO wizard templates & imported R code
- Evaluate R scripts using MAExplorer data in the context of their RLO
- Post-process R output to
  a) show R text output (stdout)
  b) display PDF, JPEG, SVG, etc. files
  c) apply actions to import R output back into MAExplorer database

Summary

- MAExplorer is a flexible fully Open Source Java-based microarray data-mining tool available at **http://maexplorer.sourceforge.net/**
- RtestPlugin is a plugin for MAExplorer to create R programs to analyze MAExplorer data
- RtestPlugin lets user’s edit and create R scripts and associated R LayOut (RLO) methods using RLO templates and imported R code
- RtestPlugin has demonstration RLO examples
- RLO methods may be added to the permanent MAExplorer menus
- RLO methods may then be used to analyze any MAExplorer data
- Results from an R evaluation may be imported back into the MAExplorer database