Part IV. Object Oriented Programming

Biobase, affy, and marrayXXX packages

Sandrine Dudoit, Robert Gentleman, Rafael Irizarry, and Yee Hwa Yang

© Copyright 2002, all rights reserved
Biobase package

- The Biobase package provides class definitions and other infrastructure tools that will be used by other packages.
- The two important classes defined in Biobase are
  - `phenoData`: sample level covariate data.
  - `exprSet`: the sample level covariate data combined with the expression data and a few other quantities of interest.
exprSet class

Slots for the exprSet class

• **exprs**: a matrix of expression measures, genes are rows, samples are columns.
• **se.exprs**: standard errors for the expression measures, if available.
• **phenoData**: an object of class phenoData that describes the target samples.
• **annotation**: a character vector.
• **description**: an object of class MIAME notes: a character vector.
exprSet class

- **exprs**: Matrix of expression measures, genes x samples
- **se.exprs**: Matrix of SEs for expression measures
- **phenoData**: Sample level covariates, instance of class `phenoData`
- **annotation**: Name of annotation data
- **description**: MIAME information
- **notes**: Any notes
One of the most important tasks is to align the expression data and the phenotypic data (and to keep that alignment through the analysis).

To achieve this, the `exprSet` class combines these two data sources into one object, and provides subsetting and access methods that make it easy to manipulate the data while ensuring that they are correctly aligned.
exprSet class

- A design principle that was adopted for the exprSet and other classes was that they should be closed under the subset operation.
- So any subsetting, either of rows or columns, will return a valid exprSet object.
- This makes it easier to use exprSet in other software packages
exprSet class

Some methods for the exprSet class

• **show** controls the printing (you seldom want a few hundred thousand numbers rolling by).

• subset, [ and $, are both designed to keep correct subsets of the exprs, se.exprs, and phenoData objects.

• **split** splits the exprSet into two or more parts depending on the vector used for splitting.
exprSet class

- `geneNames`, retrieves the gene names (row names of `exprs`).

- `phenoData`, `pData`, and `sampleNames` provide access to `phenoData` slots.

- `write.exprs` writes the expression values to a file for processing or storage.
**phenoData class**

Slots for the `phenoData` class

- **pData**: a dataframe, where the samples are rows and the variables are columns (this is the standard format).

- **varLabels**: a vector containing the variable names (as they appear in `pData`) and a longer description of the variables.
**phenoData class**

- Methods for the `phenoData` class include
  - `[`, the subset operator, this method ensures that when a subset is taken, both the `pData` and the `varLabels` objects have the appropriate subsets taken.
  - `$`, extracts the appropriate column of the `pData` slot (as for a dataframe).
  - `show`, a method to control printing, we show only the `varLabels` (and the size).
Biobase package

- The data package golubEsets contains instances of the exprSet class for the ALL AML study of Golub et al. (1999).
- Try

  ```
  library(golubEsets)
data(golubTrain)
show(golubTrain)
golubTrain[1:100,1:4]
pData(golubTrain)
  ```
Pre-processing cDNA microarray data

- **marrayClasses**: class definitions for cDNA microarray data;
  - basic methods for manipulating microarray objects: printing, plotting, subsetting, class conversions, etc.

- **marrayInput**: reading in intensity data and textual data describing probes and targets;
  - automatic generation of microarray data objects;
  - widgets for point & click interface.

- **marrayPlots**: diagnostic plots.

- **marrayNorm**: robust adaptive location and scale normalization procedures.
marrayClasses package

• Based on *Minimum Information About a Microarray Experiment - MIAME* - document.

• Microarray classes should represent
  – gene expression measures
    • scanned images, i.e., raw data,
    • image quantitation data, i.e., output from image analysis,
    • normalized expression levels, i.e., log-ratios M;
  – reliability information for these measurements;
  – information on the probe sequences spotted on the arrays;
  – information on the target samples hybridized to the arrays.
marrayLayout class

Array layout parameters

- maNspots: Total number of spots
- maNgr: Dimensions of grid matrix
- maNgc: Dimensions of spot matrices
- maNsr: Current subset of spots
- maNsc: Plate IDs for each spot
- maSub: Control status labels for each spot
- maPlate: Any notes
marrayRaw class

Pre-normalization intensity data for a batch of arrays

- **maRf**
  - Matrix of red and green foreground intensities

- **maGf**

- **maRb**
  - Matrix of red and green background intensities

- **maGb**

- **maW**
  - Matrix of spot quality weights

- **maLayout**
  - Array layout parameters - marrayLayout

- **maGnames**
  - Description of spotted probe sequences - marrayInfo

- **maTargets**
  - Description of target samples - marrayInfo

- **maNotes**
  - Any notes
marrayNorm class

Post-normalization intensity data for a batch of arrays

- maA: Matrix of average log-intensities, A
- maM: Matrix of normalized intensity log-ratios, M
- maMloc: Matrix of location and scale normalization values
- maMscale: Matrix of spot quality weights
- maW: Matrix of average log-intensities, A
- maLayout: Array layout parameters - marrayLayout
- maGnames: Description of spotted probe sequences - marrayInfo
- maTargets: Description of target samples - marrayInfo
- maNormCall: Function call
- maNotes: Any notes
marrayInput package

- **marrayInput** provides functions for reading microarray data into R and creating microarray objects of class `marrayLayout`, `marrayInfo`, and `marrayRaw`.

- **Input**
  - Image quantitation data, i.e., output files from image analysis software. E.g. `.gpr` for GenePix, `.spot` for Spot.
  - Textual description of probe sequences and target samples. E.g. gal files, god lists.
marrayInput package

- Widgets for graphical user interface
  - widget.marrayLayout
  - widget.marrayInfo
  - widget.marrayRaw
marrayPlots package

- **Diagnostic plots** of spot statistics.
  E.g. red and green log-intensities, intensity log-ratios M, average log-intensities A, spot area.
  - *maImage*: 2D spatial images.
  - *maBoxplot*: boxplots.
  - *maPlot*: scatter-plots with fitted curves and text highlighted.

- **Stratify** plots according to layout parameters such as print-tip-group, plate.
  E.g. MA-plots with loess fits by print-tip-group.

- See `demo(marrayPlots)`. 
2D spatial images

Cy3 background intensity

Cy5 background intensity
Boxplots by print-tip-group

Swirl 93 array: pre-normalization log-ratio $M$

Intensity
log-ratio, $M$
MA-plot by print-tip-group

\begin{align*}
M &= \log_2 R - \log_2 G, \\
A &= \frac{\log_2 R + \log_2 G}{2}
\end{align*}

Swirl 93 array: pre-normalization log-ratio $M$

Intensity log-ratio, $M$

Average log-intensity, $A$
marrayNorm package

- **maNormMain**: main normalization function, allows robust adaptive location and scale normalization for a batch of arrays
  - intensity or A-dependent location normalization (**maNormLoess**);
  - 2D spatial location normalization (**maNorm2D**);
  - median location normalization (**maNormMed**);
  - scale normalization using MAD (**maNormMAD**);
  - composite normalization;
  - your own normalization function.

- **maNorm**: simple wrapper function.
  - **maNormScale**: simple wrapper function for scale normalization.
Boxplots of normalized M

Global median normalization

Global loess normalization

Within-print-tip-group loess normalization

2D spatial normalization
MA-plots of normalized M

Global median normalization

Within-print-tip-group loess normalization

Global loess normalization

2D spatial normalization
Pre-processing Affymetrix data

- Bioconductor R package **affy**.
- Background estimation.
- Probe-level normalization: quantile, curve-fitting.
- Expression measures: AvDiff, Signal, Li & Wong (2001), RMA.
- Two main functions: **ReadAffy**, **express**
affy package

- Pre-processing for Affymetrix chip data.
- Class definitions for probe-level data: Cdf, Cel, PPSet, Plob.
- Basic methods for manipulating microarray objects: printing, plotting, subsetting.
- Functions and widgets for data input from CDF and CEL files, and automatic generation of microarray data objects.
• Diagnostic plots: 2D spatial images, boxplots, MA-plots, etc.
• Background estimation.
• Probe-level normalization: quantile and curve-fitting normalization.
• Expression measures: MAS 4.0 AvDiff, MAS 5.0 Signal, MBEI (Li & Wong, 2001), RMA (Irizarry et al., 2002).
• Two main functions: `ReadAffy`, `express`.

affy package
Reading in data: `ReadAffy()`
Reading in data: ReadAffy()
Methods for Quality Control:

image, hist, boxplot, mva.pairs
hist

hg_u95a.cdf – C2.5

log intensity
boxplot

hg_u95a.cdf : MM

hg_u95a.cdf : PM
mva.pairs

MVA plot

A2.5

M

0.277

B2.5

V

0.204

B2.5

A

0.378

0.231

C2.5

A

0.189

0.231

C2.5
Computing Expression express

- Works on probe level objects
- Takes as arguments
  - Normalize (TRUE or FALSE)
  - Method (method of normalization)
  - bg (arbitrary function that performs an action on pm, mm matrix)
  - Summary (arbitrary function to summarize columns, e.g. medpolish)