marray

April 19, 2010

`boxplot` *Boxplots for cDNA microarray spot statistics*

Description

The function `boxplot` produces boxplots of microarray spot statistics for the classes "`marrayRaw`", "`marrayNorm`". We encourage users to use `boxplot` rather than `maBoxplot`. The name of the arguments have changed slightly.

Usage

```
## S4 method for signature 'marrayRaw':
boxplot(x, xvar="maPrintTip", yvar="maM", ...)
## S4 method for signature 'marrayNorm':
boxplot(x, xvar="maPrintTip", yvar="maM", ...)
```

Arguments

`x`	Microarray object of class "`marrayRaw`", "`marrayNorm`"
`xvar`	Name of accessor method for the spot statistic used to stratify the data, typically a slot name for the microarray layout object (see "`marrayLayout`") such as `maPlate` or a method such as `maPrintTip`. If `x` is NULL, the data are not stratified.
`yvar`	Name of accessor method for the spot statistic of interest, typically a slot name for the microarray object `m`, such as `maM`.
`...`	Optional graphical parameters, see `par`.

Details

If there are more than one array in the batch, the function produces a boxplot for each array in the batch. Such plots are useful when assessing the need for between array normalization, for example, to deal with scale differences among different arrays. Default graphical parameters are chosen for convenience using the function `maDefaultPar` (e.g. color palette, axis labels, plot title) but the user has the option to overwrite these parameters at any point.

Author(s)

Jean Yang and Sandrine Dudoit

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

See Also

maBoxplot, maDefaultPar.

Examples

```
# To see the demo type demo(marrayPlots)

# Examples use swirl dataset, for description type ? swirl
data(swirl)

# Boxplots of pre-normalization log-ratios M for each of the 16
# print-tip-groups for the Swirl 93 array.
# - Default arguments
boxplot(swirl[,3])

# All spots
boxplot(swirl[,3], xvar=NULL, col="green")

# Boxplots of pre-normalization red foreground intensities for each grid row
# for the Swirl 81 array.
boxplot(swirl[,1], xvar="maGridRow", yvar = "maRf", main = "Swirl array 81: pre-normaliza

# Boxplots of pre-normalization log-ratios for each array in swirl
boxplot(swirl, main="Swirl arrays: pre-normalization log-ratios")
```

[-methods *Subsetting methods for microarray objects*

Description

Subsetting methods were defined for the microarray classes, marrayInfo, marrayLayout,marrayRaw and marrayNorm. These methods create instances of the given class, for a subset of spots and/or arrays in a batch.

Methods

x = ANY generic method.

x = marrayInfo `x[i, j]` extract object of class "marrayInfo" for spots or arrays with indices i and labels with indices j.

x = marrayLayout `x[i]` extract object of class "marrayLayout" for spots with indices i.

x = marrayRaw `x[i, j]` extract object of class "marrayRaw" for spots with indices i and arrays with indices j.

x = marrayNorm `x[i, j]` extract object of class "marrayNorm" for spots with indices i and arrays with indices j.

cbind — *Combine marrayRaw, marrayNorm or marrayInfo Objects*

Description

Combine a series of `marrayRaw`, `marrayNorm` and `marrayInfo` objects.

Usage

```
## S3 method for class 'marrayRaw':
cbind(..., deparse.level=1)
## S3 method for class 'marrayNorm':
cbind(..., deparse.level=1)
## S3 method for class 'marrayInfo':
rbind(..., deparse.level=1)
```

Arguments

`...`	`marrayRaw` objects or `marrayNorm` objects
`deparse.level`	not currently used, see `cbind` in the base package

Details

`cbind` combines data objects assuming the same gene lists but different arrays. `rbind` combines data objects assuming equivalent arrays, i.e., the same RNA targets, but different genes.

For `cbind`, the matrices o f expression data from the individual objects are cbinded. The data.frames of target information, if they exist, are rbinded. The combined data object will preserve any additional components or attributes found in the first object to be combined. For `rbind`, the matrices of expression data are rbinded while the target information, in any, is unchanged.

Author(s)

Jean Yang

See Also

`cbind` in the base package.

checkTargetInfo — *Verifying the order between intensities matrix and target file information*

Description

Check that the foreground and backgruond intensities are stored in the same order as provided in the first column of target file.

Usage

```
checkTargetInfo(mraw)
```

Arguments

mraw	Object of class marrayRaw or marryNorm.

Value

A logical value. This function returns "TRUE" if the first column from the Target information is the same order as the foreground and backgruond intensities.

Author(s)

Yee Hwa (Jean) Yang

Examples

```
datadir <- system.file("swirldata", package="marray")
swirl.targets <- read.marrayInfo(file.path(datadir, "SwirlSample.txt"))
data(swirl)
swirl@maTargets <- swirl.targets

checkTargetInfo(swirl)

checkTargetInfo(swirl[, 2:4])

## reorder
swirl@maTargets <- swirl.targets[c(2:4, 1),]
checkTargetInfo(swirl)
```

---

coerce-methods *Coerce an object to belong to a given microarray class*

Description

Coercing methods were defined to convert microarray objects of one class into objects of another class, e.g., instances of the "marrayRaw" class into instances of the "marrayNorm" class.

Methods

from = marrayRaw, to = marrayNorm convert an object of class "marrayRaw" into an object of class "marrayNorm".

Note

Use Package convert to convert object to other data types such as ExpressionSet and MAList.

dim — *Retrieve the Dimensions of an marrayRaw, marrayNorm or marrayInfo Object*

Description

Retrieve the number of rows (genes) and columns (arrays) for an marrayRaw, marrayNorm or marrayInfo object.

Usage

```
## S3 method for class 'marrayRaw':
dim(x)
```

Arguments

x	an object of class marrayRaw, marrayNorm or marrayInfo

Details

Microarray data objects share many analogies with ordinary matrices in which the rows correspond to spots or genes and the columns to arrays. These methods allow one to extract the size of microarray data objects in the same way that one would do for ordinary matrices.

A consequence is that row and column commands `nrow(x)`, `ncol(x)` and so on also work.

Value

Numeric vector of length 2. The first element is the number of rows (genes) and the second is the number of columns (arrays).

Author(s)

modified from Gordon Smyth's function

See Also

dim in the base package.

Examples

```
M <- A <- matrix(11:14,4,2)
rownames(M) <- rownames(A) <- c("a","b","c","d")
colnames(M) <- colnames(A) <- c("A1","A2")
MA <- new("marrayNorm", maM=M,maA=A)
dim(MA)
dim(M)
```

findID *Find ID when given an accession number*

Description

Search gene ID with a vector of accession number from gene names or ID values.

Usage

```
findID(text, Gnames = gnames, ID = "Name")
```

Arguments

text	A character strings of gene names or id names.
Gnames	An objects of marrayRaw, marrayNorm, ExpressionSet or data.frame of gene names information.
ID	The column of ID corresponding to 'text'.

Value

A numeric vector the gene ID.

Author(s)

Yee Hwa (Jean) Yang

See Also

grep

Examples

```
data(swirl)
findID("fb24a09", swirl, ID="ID")
findID("geno1", swirl)
```

htmlPage *Display gene list as a HTML page*

Description

Given a set of index to a data.frame containing gene names information. We create a web page with one element per genes that contains URLs links to various external database links. E.g Operon oligodatabase , Riken, GenBank and PubMed web sites.

Usage

```
htmlPage(genelist, filename = "GeneList.html", geneNames =
                Gnames, mapURL = SFGL, othernames, title, table.head,
                table.center = TRUE, disp = c("browser", "file")[1])

table2html(restable, filename = "GeneList.html", mapURL = SFGL,
                title, table.head, table.center = TRUE, disp =
                c("browser", "file")[1])
```

Arguments

restable	A data.frame that contains only the information you wish to display in the html file. The rows corresponds to a different DNA spots.
genelist	A numeric vector of index to a data.frame
filename	The name of the file to store the HTML in.
geneNames	A data.frame containing the information related the each DNA spots.
mapURL	A matrix of characters containing the URL for various external database. E.g SFGL.
othernames	A data.frame containing other information.
title	Title of the HTML page
table.head	A character vector of column labels for the table
table.center	A logical indicating whether the table should be centered
disp	Either "File" or "Browser" (default is Browser). File will save the information in html file, while Browser will create an html files and display information in the user's browser.

Details

This function is an extension to ll.htmlpage

Value

No value is return, the function produce a html file "filename" and output the results in a browser.

Author(s)

Yee Hwa Yang

See Also

ll.htmlpage, URLstring, widget.mapGeneInfo

Examples

```
##library(annotate)
data(swirl)
Gnames <- maGeneTable(swirl)
swirlmap <- mapGeneInfo(Name = "none", ID="genbank")
## htmlPage(100:110, geneNames = Gnames, mapURL = swirlmap, title="Swirl")

moreinfo <- round(maM(swirl), 2)
```

```
swirlmap <- mapGeneInfo(Name = "pubmed", ID="genbank")
##htmlPage(100:110, geneNames = Gnames, mapURL = swirlmap, othernames=moreinfo, title="Sw
```

---

image *Color image for cDNA microarray spot statistics*

Description

We encourage users calling "image" rather than "maImage". The name of the arguments are change slightly. The function `image` creates spatial images of shades of gray or colors that correspond to the values of a statistic for each spot on the array. The statistic can be the intensity log-ratio M, a spot quality measure (e.g. spot size or shape), or a test statistic. This function can be used to explore whether there are any spatial effects in the data, for example, print-tip or cover-slip effects.

Usage

```
## S4 method for signature 'marrayRaw':
image(x, xvar = "maM", subset = TRUE, col, contours=FALSE,  bar = TRUE, overlay=
## S4 method for signature 'marrayNorm':
image(x, xvar = "maM", subset = TRUE, col, contours=FALSE,  bar = TRUE, overlay=
```

Arguments

`x`	Microarray object of class "marrayRaw", "marrayNorm"
`xvar`	Name of accessor function for the spot statistic of interest, typically a slot name for the microarray object `x`, such as `maM`.
`subset`	A "logical" or "numeric" vector indicating the subset of spots to display on the image.
`col`	List of colors such as that generated by rainbow, heat.colors, topo.colors, terrain.colors, or similar functions. In addition to these color palette functions, a new function `maPalette` was defined to generate color palettes from user supplied low, middle, and high color values.
`contours`	If `contours=TRUE`, contours are plotted, otherwise they are not shown.
`bar`	If `bar=TRUE`, a calibration color bar is shown to the right of the image.
`overlay`	A logical vector of spots to be highlighted on the image plots.
`ol.col`	Color of the overlay spots.
`colorinfo`	A logical value indicating whether the function should return the color scale information.
`...`	Optional graphical parameters, see `par`.

Details

This function calls the general function `maImage.func`, which is not specific to microarray data. If there are more than one array in the batch, the plot is done for the first array, by default. Default color palettes were set for different types of spot statistics using the `maPalette` function. When `x=c("maM", "maMloc", "maMscale")`, a green-to-red color palette is used. When `x=c("maGb", "maGf", "maLG")`, a white-to-green color palette is used. When `x=c("maRb", "maRf", "maLR")`, a white-to-red color palette is used. The user has the option to overwrite these parameters at any point.

Value

If `colorinfo` is set to TRUE, the following list with elements will be returned.

`x.col`	vector of colors to be used for calibration color bar.
`x.bar`	vector of values to be used for calibration color bar.
`summary`	six number summary of the spot statistics, from the function `summary`.

Author(s)

Jean Yang and Sandrine Dudoit

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

See Also

`maImage`, `maImage.func`, `maColorBar`, `maPalette`

Examples

```
# Examples use swirl dataset, for description type ? swirl
data(swirl)

# Microarray color palettes
Gcol <- maPalette(low = "white", high = "green", k = 50)
Rcol <- maPalette(low = "white", high = "red", k = 50)
BYcol <- maPalette(low = "blue", mid="gray", high = "yellow", k = 50)

# Color images of green and red background and foreground intensities
##image(swirl[, 2], xvar ="maGb")
##image(swirl[, 2], xvar ="maGf", subset = TRUE, col = Gcol, contours = FALSE, bar = TRUE
##image(swirl[, 1], xvar ="maRb", contour=TRUE)
##image(swirl[, 4], xvar ="maRf", bar=FALSE)

# Color images of pre-normalization intensity log-ratios
##image(swirl[, 1])

# Color images with overlay spots
##image(swirl[, 3], xvar = "maA", overlay = maTop(maA(swirl[, 3]), h = 0.1, l = 0.1), bar

# Color image of print-tip-group
##image(swirl[, 1],xvar = "maPrintTip")
```

ma2D *Stratified bivariate robust local regression*

Description

This function performs robust local regression of a variable z on predictor variables x and y, separately within values of a fourth variable g. It is used by maNorm2D for 2D spatial location normalization.

Usage

```
ma2D(x, y, z, g, w=NULL, subset=TRUE, span=0.4, ...)
```

Arguments

x	A numeric vector of predictor variables.
y	A numeric vector of predictor variables.
z	A numeric vector of responses.
g	Variables used to stratify the data.
w	An optional numeric vector of weights.
subset	A "logical" or "numeric" vector indicating the subset of points used to compute the fits.
span	The argument span which controls the degree of smoothing in the loess function.
...	Misc arguments

Details

z is regressed on x and y, separately within values of g using the loess function.

Value

A numeric vector of fitted values.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

See Also

maNormMain, maNorm2D, loess.

Examples

```
# See examples for maNormMain.
```

maBoxplot *Boxplots for cDNA microarray spot statistics*

Description

The function maBoxplot produces boxplots of microarray spot statistics for the classes marrayRaw and marrayNorm.We encourage users to use "boxplot" rather than "maBoxplot". The name of the arguments have changed.

Usage

```
maBoxplot(m, x="maPrintTip", y="maM", ...)
```

Arguments

m	Microarray object of class "marrayRaw" and "marrayNorm"
x	Name of accessor method for the spot statistic used to stratify the data, typically a slot name for the microarray layout object (see "marrayLayout") such as maPlate or a method such as maPrintTip. If x is NULL, the data are not stratified.
y	Name of accessor method for the spot statistic of interest, typically a slot name for the microarray object m, such as maM.
...	Optional graphical parameters, see par.

Details

If there are more than one array in the batch, the function produces a boxplot for each array in the batch. Such plots are useful when assessing the need for between array normalization, for example, to deal with scale differences among different arrays. Default graphical parameters are chosen for convenience using the function maDefaultPar (e.g. color palette, axis labels, plot title) but the user has the option to overwrite these parameters at any point.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

See Also

boxplot, maDefaultPar.

Examples

```
## see example in boxplot
```

maColorBar *Calibration bar for color images*

Description

This function produces a color image (color bar) which can be used for the legend to another color image obtained from the functions image, maImage, or maImage.func.

Usage

```
maColorBar(x, horizontal=TRUE, col=heat.colors(50), scale=1:length(x), k=10, ...
```

Arguments

x	If "numeric", a vector containing the "z" values in the color image, i.e., the values which are represented in the color image. Otherwise, a "character" vector representing colors.
horizontal	If TRUE, the values of x are represented as vertical color strips in the image, else, the values are represented as horizontal color strips.
col	Vector of colors such as that generated by rainbow, heat.colors, topo.colors, terrain.colors, or similar functions. In addition to these color palette functions, a new function maPalette was defined to generate color palettes from user supplied low, middle, and high color values.
scale	A "numeric" vector specifying the "z" values in the color image. This is used when the argument x is a "character" vector representing color information.
k	Object of class "numeric", for the number of labels displayed on the bar.
...	Optional graphical parameters, see par.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine, Yee Hwa (Jean) Yang.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

See Also

image, maImage, maImage.func, maPalette.

Examples

```
par(mfrow=c(3,1))
Rcol <- maPalette(low="white", high="red", k=10)
Gcol <- maPalette(low="white", high="green", k=50)
RGcol <- maPalette(low="green", high="red", k=100)
maColorBar(Rcol)
maColorBar(Gcol, scale=c(-5,5))
maColorBar(1:50, col=RGcol)
```

```
par(mfrow=c(1,3))
x<-seq(-1, 1, by=0.01)
maColorBar(x, col=Gcol, horizontal=FALSE, k=11)
maColorBar(x, col=Gcol, horizontal=FALSE, k=21)
maColorBar(x, col=Gcol, horizontal=FALSE, k=51)
```

---

maCompCoord *Generate grid and spot matrix coordinates*

Description

This function generates grid and spot matrix coordinates from ranges of rows and columns for the grid and spot matrices. Spots on the array are numbered consecutively starting from the top left grid and the top left spot within each grid.

Usage

```
maCompCoord(grows, gcols, srows, scols)
```

Arguments

grows	numeric vector of grid rows.
gcols	numeric vector of grid columns.
srows	numeric vector of spot rows.
scols	numeric vector of spot columns.

Value

a matrix of spot four-coordinates, with rows corresponding to spots and columns to grid row, grid column, spot row, and spot column coordinates.

Author(s)

Yee Hwa (Jean) Yang, Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

See Also

marrayLayout, maCoord2Ind, maInd2Coord, maCompInd.

Examples

```
maCompCoord(1:2,1,1:4,1:3)
```

maCompInd *Generate spot indices*

Description

This function generates spot indices from ranges of rows and columns for the grid and spot matrices. Spots on the array are numbered consecutively starting from the top left grid and the top left spot within each grid.

Usage

```
maCompInd(grows, gcols, srows, scols, L)
```

Arguments

grows	numeric vector of grid rows.
gcols	numeric vector of grid columns.
srows	numeric vector of spot rows.
scols	numeric vector of spot columns.
L	object of class "marrayLayout".

Value

a numeric vector of spot indices.

Author(s)

Yee Hwa (Jean) Yang, Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

See Also

marrayLayout, maCoord2Ind, maInd2Coord, maCompCoord.

Examples

```
L <- new("marrayLayout", maNgr=4, maNgc=4, maNsr=22, maNsc=24)
maCompInd(1:2,1,1:4,1:3,L)
```

maCompLayout *Generate a marrayLayout object*

Description

Take a matrix of cooordiates and generate a marrayLayout object.

Usage

```
maCompLayout(mat, ncolumns = 4)
```

Arguments

mat	a matrix of coordinates, this can either be n by 3 matrix with columns (Block, Row, Column) or n by 4 matrix with columns (Grid.R, Grid.C, Spot.R, Spot.C)
ncolumns	For n by 3 matrix, the number of meta-grid columns. By default, it is set to 4.

Value

An object of class "marrayLayout".

Author(s)

Jean Yang

Examples

```
X <- cbind(Block = c(1,1,2,2,3,3,4,4), Rows=c(1,2,1,2,1,2,1,2), Columns=rep(1,8))
maCompLayout(X, ncolumns=2)
```

maCompNormA *Weights for composite normalization*

Description

This function is used for composite normalization with intensity dependent weights. The function should be used as an argument to the main normalization function maNormMain. It only applies when two normalization procedures are combined.

Usage

```
maCompNormA()
maCompNormEq()
```

Value

A function which takes as arguments `x` and `n`, the spot average log-intensities A and the number of normalization procedures. This latter function returns a matrix of weights for combining two normalization procedures, rows correspond to spots and columns to normalization procedures. The weights for the first procedure are given by the empirical cumulative distribution function of the spot average log-intensities A. Note that when performing composite normalization as described in Yang et al. (2002), the first normalization procedure is the global fit and the second procedure is the within-print-tip-group fit.

For `maCompEq`, equal weights are given for each procedure.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine, Yee Hwa (Jean) Yang.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. P. Speed (2002). Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research*, Vol. 30, No. 4.

See Also

`maNormMain`, `maNormLoess`, `ecdf`.

Examples

```
# See examples for maNormMain
```

---

maCompPlate *Generate plate IDs*

Description

This function generates plate IDs from the dimensions of the grid and spot matrices. Note that this function only applies to arrays with a regular plate layout, where the number of spots is a multiple of the number of wells on a plate (usually 96 or 384) and each well contributes exactly one spot. It should thus be used with caution.

Usage

```
maCompPlate(x, n=384)
```

Arguments

`x`	object of class "`marrayLayout`", "`marrayRaw`" and "`marrayNorm`"
`n`	object of class "numeric", number of wells in each plate, usually 384 or 96.

Details

Having plate IDs may be useful for the purpose of normalization. Normalization by plate can be done using the function maNormMain.

Value

a vector of plate IDs (factor).

Author(s)

Yee Hwa (Jean) Yang, Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

See Also

marrayLayout, marrayRaw, marrayNorm

Examples

```
L<-new("marrayLayout", maNgr=4, maNgc=4, maNsr=22, maNsc=24)
plate<-maCompPlate(L,384)
table(plate)
maPlate(L)<-plate
```

---

maCoord2Ind *Convert grid and spot matrix coordinates to spot indices*

Description

This functions converts grid and spot matrix coordinates (four coordinates) to spot indices, where spots on the array are numbered consecutively starting from the top left grid and the top left spot within each grid.

Usage

```
maCoord2Ind(x, L)
```

Arguments

x	a matrix of spot four-coordinates, with rows corresponding to spots and columns to grid row, grid column, spot row, and spot column coordinates.
L	an object of class "marrayLayout".

Value

a numeric vector of spot indices.

Author(s)

Yee Hwa (Jean) Yang, Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

See Also

marrayLayout, maInd2Coord, maCompCoord, maCompInd.

Examples

```
L <- new("marrayLayout", maNgr=4, maNgc=4, maNsr=22, maNsc=24)
coord<-cbind(rep(2,4),rep(1,4),rep(1,4),1:4)
maCoord2Ind(coord, L)
```

---

maDefaultPar *Default graphical parameters for microarray objects*

Description

This function returns default graphical parameters for microarray objects. The parameters may be passed as arguments to the functions maBoxplot and maPlot.

Usage

```
maDefaultPar(m, x, y, z)
```

Arguments

m	Microarray object of class "marrayRaw" and "marrayNorm".
x	Name of accessor method for the abscissa spot statistic, typically a slot name for the microarray object m, such as maA.
y	Name of accessor method for the ordinate spot statistic, typically a slot name for the microarray object m, such as maM.
z	Name of accessor method for the spot statistic used to stratify the data, typically a slot name for the microarray layout object (see "marrayLayout") such as maPlate or a method such as maPrintTip.

Value

A list with elements

def.box	default graphical parameters for maBoxplot.
def.plot	default graphical parameters for maPlot.
def.lines	default graphical parameters for functions such as maLoessLines used in maPlot.
def.legend	default graphical parameters for functions such as maLegendLines used in maPlot.
def.text	default graphical parameters for functions such as maText used in maPlot.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

See Also

maBoxplot, maPlot, maLegendLines, maLoessLines, maText, maDotsDefaults.

Examples

```
# See examples for maPlot.
```

---

maDotsDefaults *Replace graphical default parameters by user supplied parameters*

Description

This function may be used to compare default graphical parameters for microarray diagnostic plots to user supplied parameters given in `...` User supplied parameters overwrite the defaults. It is used in maBoxplot, maPlot, and maImage.

Usage

```
maDotsDefaults(dots, defaults)
```

Arguments

dots	List of user supplied parameters, e.g. from `list(...)`.
defaults	List of default parameters, e.g. from the function maDefaultPar.

Value

args	List of graphical parameters.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

See Also

maDefaultPar, maBoxplot, maPlot, maImage.

Examples

```
dots<-list(xlab="X1", ylab="Y1")
defaults<-list(xlab="X1", ylab="Y2", col=2)
pars<-maDotsDefaults(dots, defaults)

do.call("plot",c(list(x=1:10), pars))
```

`maDotsMatch` *Replace default arguments of a function by user supplied values*

Description

This function may be used to replace default arguements for any functions to user supplied parameters.

Usage

```
maDotsMatch(dots, defaults)
```

Arguments

`dots`	List of user supplied argements, e.g. from `list(...)`.
`defaults`	List of formal arguments of a function, e.g. from the function `formals`.

Value

`args`	List of argument of a function.

Author(s)

Jean Yee Hwa Yang

See Also

`maDefaultPar`, `maDotsDefaults`

Examples

```
dots<-list(x=1:10, y=11:20)
argsfun <- maDotsMatch(dots, formals(args(plot)))
do.call("plot", argsfun)
```

maGenControls *Generating a vector recording the control status of the spotted probe sequences.*

Description

ControlCode is a matrix representing certain regular expression pattern and the control status of the spotted probe sequences. This function uses 'grep' searches for matches to 'pattern' (its first argument) within the character vector 'x' (second argument).

Usage

```
maGenControls(Gnames, controlcode, id = "ID")
```

Arguments

Gnames	An object of class matrix, data.frame or marrayInfo which contains description of spotted probe sequences.
controlcode	A character matrix of n by 2 columns. The first column contains a few regular expression of spotted probe sequences and the second column contains the corresponding control status.
id	the column number of column name in Gnames that contains description of each spot on the array.

Value

A vector of characters recording the control status of the spotted probe sequences.

Author(s)

Jean Yee Hwa Yang

See Also

grep

Examples

```
data(swirl)
maControls(swirl) <- maGenControls(maGnames(swirl), id="Name")
table(maControls(swirl))
```

maGeneTable *Table of spot coordinates and gene names*

Description

This function produces a table of spot coordinates and gene names for objects of class "marrayRaw" and "marrayNorm".

Usage

```
maGeneTable(object)
```

Arguments

object	microarray object of class "marrayRaw" and "marrayNorm".

Value

an object of class data.frame, with rows corresponding to spotted probe sequences. The first four columns are the grid matrix and spot matrix coordinates, and the remaining columns are the spot descriptions stored in the maGnames slot of the microarray object.

Author(s)

Yee Hwa (Jean) Yang

See Also

marrayInfo, marrayLayout, marrayRaw, marrayNorm, maCompCoord.

Examples

```
# Example uses swirl dataset, for description type ? swirl

data(swirl)

tab<-maGeneTable(swirl)
tab[1:10,]
```

maImage.func *Color image for cDNA microarray spot statistics*

Description

This function creates spatial images of shades of gray or colors that correspond to the values of a statistic for each spot on the array. The statistic can be the intensity log-ratio M, a spot quality measure (e.g. spot size or shape), or a test statistic. This function can be used to explore whether there are any spatial effects in the data, for example, print-tip or cover-slip effects. This function is called by maImage.

Usage

```
maImage.func(x, L, subset=TRUE, col=heat.colors(12), contours=FALSE, overlay=NUL
```

Arguments

x	A "numeric" vector of spot statistics.
L	An object of class "marrayLayout", if L is missing we will assume the dimension of x.
subset	A "logical" or "numeric" vector indicating the subset of spots to display on the image.
col	A list of colors such as that generated by rainbow, heat.colors, topo.colors, terrain.colors, or similar functions. In addition to these color palette functions, a new function maPalette was defined to generate color palettes from user supplied low, middle, and high color values.
contours	If contours=TRUE, contours are plotted, otherwise they are not shown.
overlay	A logical vector of spots to be highlighted on the image plots.
ol.col	Color of the overlay spots.
...	Optional graphical parameters, see par.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

See Also

image, maImage, maColorBar, maPalette.

Examples

```
# See examples for image.
```

---

maImage *Color image for cDNA microarray spot statistics*

Description

We encourage users calling "image" rather than "maImage". The name of the arguments are change slightly.

The function maImage creates spatial images of shades of gray or colors that correspond to the values of a statistic for each spot on the array. The statistic can be the intensity log-ratio M, a spot quality measure (e.g. spot size or shape), or a test statistic. This function can be used to explore whether there are any spatial effects in the data, for example, print-tip or cover-slip effects.

Usage

```
maImage(m, x="maM", subset=TRUE, col, contours=FALSE, bar=TRUE,
overlay=NULL, ol.col=1, colorinfo=FALSE, ...)
```

Arguments

m	Microarray object of class "marrayRaw" and "marrayNorm".
x	Name of accessor function for the spot statistic of interest, typically a slot name for the microarray object m, such as maM.
subset	A "logical" or "numeric" vector indicating the subset of spots to display on the image.
col	List of colors such as that generated by rainbow, heat.colors, topo.colors, terrain.colors, or similar functions. In addition to these color palette functions, a new function maPalette was defined to generate color palettes from user supplied low, middle, and high color values.
contours	If contours=TRUE, contours are plotted, otherwise they are not shown.
bar	If bar=TRUE, a calibration color bar is shown to the right of the image.
overlay	A logical vector of spots to be highlighted on the image plots.
ol.col	Color of the overlay spots.
colorinfo	A logical value indicating whether the function should return the color scale information.
...	Optional graphical parameters, see par.

Details

This function calls the general function maImage.func, which is not specific to microarray data. If there are more than one array in the batch, the plot is done for the first array, by default. Default color palettes were set for different types of spot statistics using the maPalette function. When x=c("maM", "maMloc", "maMscale"), a green-to-red color palette is used. When x=c("maGb", "maGf", "maLG"), a white-to-green color palette is used. When x=c("maRb", "maRf", "maLR"), a white-to-red color palette is used. The user has the option to overwrite these parameters at any point.

Value

If colorinfo is set to TRUE, the following list with elements will be returned.

x.col	vector of colors to be used for calibration color bar.
x.bar	vector of values to be used for calibration color bar.
summary	six number summary of the spot statistics, from the function summary.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

See Also

image, maImage.func, maColorBar, maPalette, summary.

Examples

```
# To see the demo type demo(marrayPlots)

# Examples use swirl dataset, for description type ? swirl
data(swirl)

# Microarray color palettes
Gcol <- maPalette(low = "white", high = "green", k = 50)
Rcol <- maPalette(low = "white", high = "red", k = 50)
RGcol <- maPalette(low = "green", high = "red", k = 50)

# Color images of green and red background and foreground intensities
maImage(swirl[, 3], x="maGb")
maImage(swirl[, 3], x = "maGf", subset = TRUE, col = Gcol, contours = FALSE, bar = TRUE,
maImage(swirl[, 3], x = "maRb", contour=TRUE)
maImage(swirl[, 3], x = "maRf", bar=FALSE)

# Color images of pre-normalization intensity log-ratios
maImage(swirl[, 1])
maImage(swirl[, 3], x = "maM", subset = maTop(maM(swirl[, 3]), h = 0.1, l = 0.1), col = R

# Color image of print-tip-group
maImage(swirl[, 1],x="maPrintTip")
```

---

maInd2Coord *Convert spot indices to grid and spot matrix coordinates*

Description

This functions converts spot indices to grid and spot matrix coordinates (four coordinates), where spots on the array are numbered consecutively starting from the top left grid and the top left spot within each grid.

Usage

```
maInd2Coord(x, L)
```

Arguments

x	a numeric vector of spot indices.
L	an object of class "marrayLayout".

Value

a matrix of spot four-coordinates, with rows corresponding to spots and columns to grid row, grid column, spot row, and spot column coordinates.

Author(s)

Yee Hwa (Jean) Yang, Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

See Also

marrayLayout, maCoord2Ind, maCompCoord, maCompInd.

Examples

```
L <- new("marrayLayout", maNgr=4, maNgc=4, maNsr=22, maNsc=24)
maInd2Coord(c(1:10,529:538), L)
```

---

maLegendLines *Add a legend to a plot*

Description

This function may be used to add a legend for lines in plots such as those produced by plot, maPlot, or maPlot.func.

Usage

```
maLegendLines(legend="", col=2, lty=1, lwd=2.5, ncol=1, ...)
```

Arguments

legend	A vector of "character" strings to appear in the legend.
col	Line colors for the legend.
lty	Line types for the legend.
lwd	Line widths for the legend.
ncol	The number of columns in which to set the legend items (default is 1, a vertical legend).
...	Optional graphical parameters, see par.

Value

A function with bindings for legend, col, lty, lwd, ncol, and .... This latter function takes as arguments x and y, the coordinates for the location of the legend on the plot, and it adds the legend to the current plot.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

See Also

legend, maPlot, maPlot.func.

Examples

```
# See examples for maPlot.
```

---

maLoessLines *Add smoothed fits to a plot*

Description

This function may be used to compute and plot loess or lowess fits for an existing plot. The plot can be produced by plot, maPlot, or maPlot.func.

Usage

```
maLoessLines(subset=TRUE, weights=NULL, loess.args=list(span = 0.4,
degree=1, family="symmetric", control=loess.control(trace.hat =
"approximate", iterations=5, surface="direct")), col=2, lty=1, lwd=2.5, ...)

maLowessLines(subset = TRUE, f = 0.3, col = 2, lty = 1, lwd = 2.5, ...)
```

Arguments

subset	A "logical" or "numeric" vector indicating the subset of points used to compute the fits.
weights	Optional "numeric" vector of weights – for maLoessLines only.
loess.args	List of optional arguments for the loess functions – for maLoessLines only.
f	The smoother span for the lowess function – for maLowessLines only.
col	The fitted line colors.
lty	The fitted line types.
lwd	The fitted line widths.
...	Optional graphical parameters, see par.

Value

A function with bindings for subset, weights, loess.args, col, lty, lwd, and .... This latter function takes as arguments x and y, the abscissa and ordinates of points on the plot, and z a vector of discrete values used to stratify the points. Loess (or lowess) fits are performed separately within values of z.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

See Also

loess, lowess, maPlot, maPlot.func.

Examples

```
# See examples for maPlot.
```

---

maLoess *Stratified univariate robust local regression*

Description

This function performs robust local regression of a variable y on predictor variable x, separately within values of a third variable z. It is used by maNormLoess for intensity dependent location normalization.

Usage

```
maLoess(x, y, z, w=NULL, subset=TRUE, span=0.4, ...)
```

Arguments

x	A numeric vector of predictor variables.
y	A numeric vector of responses.
z	Variables used to stratify the data.
w	An optional numeric vector of weights.
subset	A "logical" or "numeric" vector indicating the subset of points used to compute the fits.
span	The argument span which controls the degree of smoothing in the loess function.
...	Misc arguments.

Details

y is regressed on x, separately within values of z using the loess function.

Value

A numeric vector of fitted values.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

Y. H. Yang, S. Dudoit, P. Luu, and T. P. Speed (2001). Normalization for cDNA microarray data. In M. L. Bittner, Y. Chen, A. N. Dorsel, and E. R. Dougherty (eds), *Microarrays: Optical Technologies and Informatics*, Vol. 4266 of *Proceedings of SPIE*.

Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. P. Speed (2002). Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research*, Vol. 30, No. 4.

See Also

`maNormMain`, `maNormLoess`, `loess`.

Examples

```
# See examples for maNormMain.
```

`maMAD` *Stratified MAD calculation*

Description

This function computes the median absolute deviation (MAD) of values in `y` separately within values of `x`. It is used by `maNormMAD` for MAD scale normalization.

Usage

```
maMAD(x, y, geo=TRUE, subset=TRUE)
```

Arguments

`x`	Variables used to stratify the data.
`y`	A numeric vector.
`geo`	If `TRUE`, the MAD of each group is divided by the geometric mean of the MADs across groups (cf. Yang et al. (2002)). This allows observations to retain their original units.
`subset`	A "logical" or "numeric" vector indicating the subset of points used to compute the MAD.

Value

A numeric vector of MAD values.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

Y. H. Yang, S. Dudoit, P. Luu, and T. P. Speed (2001). Normalization for cDNA microarray data. In M. L. Bittner, Y. Chen, A. N. Dorsel, and E. R. Dougherty (eds), *Microarrays: Optical Technologies and Informatics*, Vol. 4266 of *Proceedings of SPIE*.

Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. P. Speed (2002). Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research*, Vol. 30, No. 4.

See Also

maNormMain, maNormMAD, mad.

Examples

```
# See examples for maNormMain.
```

---

maMed *Stratified median calculation*

Description

This function computes the median of values in y separately within values of x. It is used by maNormMed for median location normalization.

Usage

```
maMed(x, y, subset=TRUE)
```

Arguments

x	Variables used to stratify the data.
y	A numeric vector.
subset	A "logical" or "numeric" vector indicating the subset of points used to compute the median.

Value

A numeric vector of median values.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

Y. H. Yang, S. Dudoit, P. Luu, and T. P. Speed (2001). Normalization for cDNA microarray data. In M. L. Bittner, Y. Chen, A. N. Dorsel, and E. R. Dougherty (eds), *Microarrays: Optical Technologies and Informatics*, Vol. 4266 of *Proceedings of SPIE*.

Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. P. Speed (2002). Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research*, Vol. 30, No. 4.

See Also

maNormMain, maNormMed, median.

Examples

```
# See examples for maNormMain.
```

---

na *Basic Statistical Functions for Handling Missing Values*

Description

Basic statistical functions for handling missing values or NA.
In log.na, sum.na, mean.na and var.na, quantile.na, length.na, missing values are omitted from the calculation.
The function cor.na calls cor with the argument use="pairwise.complete.obs".
The function order.na only handles vector arguments and not lists. However, it gives the option of omitting the NAs (na.last=NA), of placing the NAs at the start of the ordered vector (na.last=F) or at the end (na.last=T).
The function scale.na is a modified version of scale which allows NAs in the variance calculation. If scale = T, the function f in scale.na uses var.na to perform the variance calculation. The function prod.na is similar to the prod function with na.rm=TRUE. This function returns the product of all the values present in its arguments, omitting any missing values.

Author(s)

Yee Hwa Yang, <jean@biostat.berkeley.edu>

See Also

log, sum, mean, var, cor, order, scale, prod.

---

maNorm2D *2D spatial location normalization function*

Description

This function is used for 2D spatial location normalization, using the robust local regression function loess. It should be used as an argument to the main normalization function maNormMain.

Usage

```
maNorm2D(x="maSpotRow", y="maSpotCol", z="maM", g="maPrintTip", w=NULL,
subset=TRUE, span=0.4, ...)
```

Arguments

x	Name of accessor method for spot row coordinates, usually maSpotRow.
y	Name of accessor method for spot column coordinates, usually maSpotCol.
z	Name of accessor method for spot statistics, usually the log-ratio maM.
g	Name of accessor method for print-tip-group indices, usually maPrintTip.
w	An optional numeric vector of weights.
subset	A "logical" or "numeric" vector indicating the subset of points used to compute the fits.
span	The argument span which controls the degree of smoothing in the loess function.
...	Misc arguments

Details

The spot statistic named in z is regressed on spot row and column coordinates, separately within print-tip-group, using the loess function.

Value

A function with bindings for the above arguments. This latter function takes as argument an object of class "marrayRaw" (or possibly "marrayNorm"), and returns a vector of fitted values to be substracted from the raw log-ratios. It calls the function ma2D, which is not specific to microarray objects.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

See Also

maNormMain, ma2D, loess.

Examples

```
# See examples for maNormMain.
```

---

maNormLoess *Intensity dependent location normalization function*

Description

This function is used for intensity dependent location normalization, using the robust local regression function loess. It should be used as an argument to the main normalization function maNormMain.

Usage

```
maNormLoess(x="maA", y="maM", z="maPrintTip", w=NULL, subset=TRUE,
span=0.4, ...)
```

Arguments

x	Name of accessor method for spot statistics, usually maA.
y	Name of accessor method for spot statistics, usually maM.
z	Name of accessor method for spot statistic used to stratify the data, usually a layout parameter, e.g. maPrintTip or maPlate. If z is not a character, e.g. NULL, the data are not stratified.
w	An optional numeric vector of weights.
subset	A "logical" or "numeric" vector indicating the subset of points used to compute the fits.
span	The argument span which controls the degree of smoothing in the loess function.
...	Misc arguments

Value

A function with bindings for the above arguments. This latter function takes as argument an object of class "marrayRaw" (or possibly "marrayNorm"), and returns a vector of fitted values to be substracted from the raw log-ratios. It calls the function maLoess, which is not specific to microarray objects.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

Y. H. Yang, S. Dudoit, P. Luu, and T. P. Speed (2001). Normalization for cDNA microarray data. In M. L. Bittner, Y. Chen, A. N. Dorsel, and E. R. Dougherty (eds), *Microarrays: Optical Technologies and Informatics*, Vol. 4266 of *Proceedings of SPIE*.

Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. P. Speed (2002). Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research*, Vol. 30, No. 4.

See Also

`maNormMain`, `maLoess`, `loess`.

Examples

```
# See examples for maNormMain.
```

`maNormMAD` *MAD scale normalization function*

Description

This function is used for scale normalization using the median absolute deviation (MAD) of intensity log-ratios for a group of spots. It can be used for within or between array normalization. The function should be used as an argument to the main normalization function `maNormMain`.

Usage

```
maNormMAD(x=NULL, y="maM", geo=TRUE, subset=TRUE)
```

Arguments

`x`	Name of accessor function for spot statistic used to stratify the data, usually a layout parameter, e.g. `maPrintTip` or `maPlate`. If `x` is not a character, e.g. NULL, the data are not stratified.
`y`	Name of accessor function for spot statistics, usually `maM`.
`geo`	If `TRUE`, the MAD of each group is divided by the geometric mean of the MADs across groups (cf. Yang et al. (2002)). This allows observations to retain their original units.
`subset`	A "logical" or "numeric" vector indicating the subset of points used to compute the scale normalization values.

Value

A function with bindings for the above arguments. This latter function takes as argument an object of class "marrayRaw" (or possibly "marrayNorm"), and returns a vector of values used to scale the location normalized log-ratios. It calls the function maMAD, which is not specific to microarray objects.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

Y. H. Yang, S. Dudoit, P. Luu, and T. P. Speed (2001). Normalization for cDNA microarray data. In M. L. Bittner, Y. Chen, A. N. Dorsel, and E. R. Dougherty (eds), *Microarrays: Optical Technologies and Informatics*, Vol. 4266 of *Proceedings of SPIE*.

Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. P. Speed (2002). Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research*, Vol. 30, No. 4.

See Also

maNormMain, maMAD, mad.

Examples

```
# See examples for maNormMain.
```

---

maNormMain — *Main function for location and scale normalization of cDNA microarray data*

Description

This is the main function for location and scale normalization of cDNA microarray data. Normalization is performed for a batch of arrays using location and scale normalization procedures specified by the lists of functions f.loc and f.scale. Typically, only one function is given in each list, otherwise composite normalization is performed using the weights computed by the functions a.loc and a.scale. The function operates on an object of class "marrayRaw" (or possibly "marrayNorm", if normalization is performed in several steps) and returns an object of class "marrayNorm". Simple wrapper functions are provided by maNorm and maNormScale.

Usage

```
maNormMain(mbatch, f.loc=list(maNormLoess()), f.scale=NULL,
a.loc=maCompNormEq(), a.scale=maCompNormEq(), Mloc=TRUE, Mscale=TRUE, echo=FALSE
```

Arguments

mbatch	An object of class "marrayRaw", containing intensity data for the batch of arrays to be normalized. An object of class "marrayNorm" may also be passed if normalization is performed in several steps.
f.loc	A list of location normalization functions, e.g., maNormLoess, maNormMed, or maNorm2D.
f.scale	A list of scale normalization functions, .e.g, maNormMAD.
a.loc	For composite normalization, a function for computing the weights used in combining several location normalization functions, e.g., maCompNormA.
a.scale	For composite normalization, a function for computing the weights used in combining several scale normalization functions.
Mloc	If TRUE, the location normalization values are stored in the slot maMloc of the object of class "marrayNorm" returned by the function, if FALSE, these values are not retained.
Mscale	If TRUE, the scale normalization values are stored in the slot maMscale of the object of class "marrayNorm" returned by the function, if FALSE, these values are not retained.
echo	If TRUE, the index of the array currently being normalized is printed.

Details

When both location and scale normalization functions (f.loc and f.scale) are passed, location normalization is performed before scale normalization. That is, scale values are computed for the location normalized log-rations. The same results could be obtained by two applications of the function maNormMain, first with only the location normalization function and f.scale=NULL, and second with only the scale normalization function and f.loc=NULL.

Value

mnorm	An object of class "marrayNorm", containing the normalized intensity data.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

Y. H. Yang, S. Dudoit, P. Luu, and T. P. Speed (2001). Normalization for cDNA microarray data. In M. L. Bittner, Y. Chen, A. N. Dorsel, and E. R. Dougherty (eds), *Microarrays: Optical Technologies and Informatics*, Vol. 4266 of *Proceedings of SPIE*.

Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. P. Speed (2002). Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research*, Vol. 30, No. 4.

See Also

maNorm, maNormScale, maNormLoess, maLoess, maNormMAD, maMAD, maNormMed, maMed, maNorm2D, ma2D, maCompNormA, maCompNormEq.

Examples

```
# Examples use swirl dataset, for description type ? swirl
data(swirl)

# Within-print-tip-group loess location normalization of batch swirl
# - Default normalization
swirl.norm<-maNormMain(swirl)

boxplot(swirl.norm)
boxplot(swirl.norm[,3])
plot(swirl.norm[,3])

# Global median normalization for arrays 81 and 82
swirl.norm <- maNormMain(swirl[,1:2], f.loc = list(maNormMed(x=NULL,y="maM")))

# Global loess normalization for array 81
swirl.norm <- maNormMain(swirl[,1], f.loc = list(maNormLoess(x="maA",y="maM",z=NULL)))

# Composite normalization as in Yang et al. (2002)
# No MSP controls are available here, so all spots are used for illustration
# purposes
swirl.norm <- maNormMain(swirl[,1], f.loc = list(maNormLoess(x="maA",y="maM",z=NULL),maNo
```

---

maNormMed *Median location normalization function*

Description

This function is used for location normalization using the median of intensity log-ratios for a group of spots. The function should be used as an argument to the main normalization function maNormMain.

Usage

```
maNormMed(x=NULL, y="maM", subset=TRUE)
```

Arguments

x	Name of accessor method for spot statistic used to stratify the data, usually a layout parameter, e.g. maPrintTip or maPlate. If x is not a character, e.g. NULL, the data are not stratified.
y	Name of accessor method for spot statistics, usually maM.
subset	A "logical" or "numeric" vector indicating the subset of points used to compute the location normalization values.

Value

A function with bindings for the above arguments. This latter function takes as argument an object of class "marrayRaw" (or possibly "marrayNorm"), and returns a vector of fitted values to be subtracted from the raw log-ratios. It calls the function maMed, which is not specific to microarray objects.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

Y. H. Yang, S. Dudoit, P. Luu, and T. P. Speed (2001). Normalization for cDNA microarray data. In M. L. Bittner, Y. Chen, A. N. Dorsel, and E. R. Dougherty (eds), *Microarrays: Optical Technologies and Informatics*, Vol. 4266 of *Proceedings of SPIE*.

Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. P. Speed (2002). Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research*, Vol. 30, No. 4.

See Also

maNormMain, maMed, median.

Examples

```
# See examples for maNormMain.
```

maNorm *Simple location and scale normalization function*

Description

This function is a simple wrapper function around the main normalization function maNormMain. It allows the user to choose from a set of six basic location and scale normalization procedures. The function operates on an object of class "marrayRaw" (or possibly "marrayNorm", if normalization is performed in several steps) and returns an object of class "marrayNorm".

Usage

```
maNorm(mbatch, norm=c("printTipLoess", "none", "median", "loess",
"twoD", "scalePrintTipMAD"), subset=TRUE, span=0.4, Mloc=TRUE,
Mscale=TRUE, echo=FALSE, ...)
```

Arguments

mbatch	Object of class marrayRaw, containing intensity data for the batch of arrays to be normalized. An object of class "marrayNorm" may also be passed if normalization is performed in several steps.
norm	Character string specifying the normalization procedures: **none** no normalization **median** for global median location normalization **loess** for global intensity or A-dependent location normalization using the loess function **twoD** for 2D spatial location normalization using the loess function **printTipLoess** for within-print-tip-group intensity dependent location normalization using the loess function **scalePrintTipMAD** for within-print-tip-group intensity dependent location normalization followed by within-print-tip-group scale normalization using the median absolute deviation (MAD). This argument can be specified using the first letter of each method.
subset	A "logical" or "numeric" vector indicating the subset of points used to compute the normalization values.
span	The argument span which controls the degree of smoothing in the loess function.
Mloc	If TRUE, the location normalization values are stored in the slot maMloc of the object of class "marrayNorm" returned by the function, if FALSE, these values are not retained.
Mscale	If TRUE, the scale normalization values are stored in the slot maMscale of the object of class "marrayNorm" returned by the function, if FALSE, these values are not retained.
echo	If TRUE, the index of the array currently being normalized is printed.
...	Misc arguments

Details

See maNormMain for details and also more general procedures.

Value

mnorm An object of class "marrayNorm", containing the normalized intensity data.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

Y. H. Yang, S. Dudoit, P. Luu, and T. P. Speed (2001). Normalization for cDNA microarray data. In M. L. Bittner, Y. Chen, A. N. Dorsel, and E. R. Dougherty (eds), *Microarrays: Optical Technologies and Informatics*, Vol. 4266 of *Proceedings of SPIE*.

Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. P. Speed (2002). Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research*, Vol. 30, No. 4.

See Also

maNormMain, maNormScale.

Examples

```
# Examples use swirl dataset, for description type ? swirl
data(swirl)

# Global median normalization for swirl arrays 2 and 3
mnorm<-maNorm(swirl[,2:3], norm="median", echo=TRUE)

# Within-print-tip-group loess location normalization for swirl array 1
mnorm<-maNorm(swirl[,1], norm="p", span=0.45)
```

---

maNormScale *Simple scale normalization function*

Description

This function is a simple wrapper function around the main normalization function maNormMain. It allows the user to choose from a set of two basic scale normalization procedures. The function operates on an object of class "marrayRaw" (or possibly "marrayNorm", if normalization is performed in several steps) and returns an object of class "marrayNorm". This function can be used to conormalize a batch of arrays (norm="globalMAD" option).

Usage

```
maNormScale(mbatch, norm=c("globalMAD", "printTipMAD"), subset=TRUE, geo=TRUE,
```

Arguments

mbatch	An object of class "marrayRaw", containing intensity data for the batch of arrays to be normalized. An object of class marrayNorm may also be passed if normalization is performed in several steps.
norm	A character string specifying the normalization procedures: **globalMAD** for global scale normalization using the median absolute deviation (MAD), this allows between slide scale normalization **printTipMAD** for within-print-tip-group scale normalization using the median absolute deviation (MAD). This argument can be specified using the first letter of each method.
subset	A "logical" or "numeric" vector indicating the subset of points used to compute the normalization values.
geo	If TRUE, the MAD of each group is divided by the geometric mean of the MADs across groups (cf. Yang et al. (2002)). This allows observations to retain their original units.
Mscale	If TRUE, the scale normalization values are stored in the slot maMscale of the object of class "marrayNorm" returned by the function, if FALSE, these values are not retained.
echo	If TRUE, the index of the array currently being normalized is printed.

Details

See maNormMain for details and more general procedures.

Value

mnorm	An object of class "marrayNorm", containing the normalized intensity data.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

Y. H. Yang, S. Dudoit, P. Luu, and T. P. Speed (2001). Normalization for cDNA microarray data. In M. L. Bittner, Y. Chen, A. N. Dorsel, and E. R. Dougherty (eds), *Microarrays: Optical Technologies and Informatics*, Vol. 4266 of *Proceedings of SPIE*.

Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. P. Speed (2002). Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research*, Vol. 30, No. 4.

See Also

maNormMain, maNorm.

Examples

```
# Examples use swirl dataset, for description type ? swirl
data(swirl)

# Global median normalization followed by global MAD normalization for
# only arrays 2 and 3 in the batch swirl

mnorm1<-maNorm(swirl[,2:3], norm="m")
mnorm2<-maNormScale(mnorm1, norm="g")
```

---

maNum2Logic *Convert a numeric vector of indices to a logical vector*

Description

This function converts a numeric vector of indices to a logical vector. It is used for subsetting purposes.

Usage

```
maNum2Logic(n=length(subset), subset=TRUE)
```

Arguments

n	the length of the logical vector to be produced.
subset	a numeric vector of indices. A logical vector may also be supplied, in which case it is also the value of the function.

Value

a logical vector.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

Examples

```
maNum2Logic(10, 1:3)
```

maPalette *Microarray color palette*

Description

This function returns a vector of color names corresponding to a range of colors specified in the arguments.

Usage

```
maPalette(low = "white", high = c("green", "red"), mid=NULL, k =50)
```

Arguments

low	Color for the lower end of the color palette, specified using any of the three kinds of R colors, i.e., either a color name (an element of colors), a hexadecimal string of the form "#rrggbb", or an integer i meaning palette()[i].
high	Color for the upper end of the color palette, specified using any of the three kinds of R colors, i.e., either a color name (an element of colors), a hexadecimal string of the form "#rrggbb", or an integer i meaning palette()[i].
mid	Color for the middle portion of the color palette, specified using any of the three kinds of R colors, i.e., either a color name (an element of colors), a hexadecimal string of the form "#rrggbb", or an integer i meaning palette()[i].
k	Number of colors in the palette.

Value

A "character" vector of color names. This can be used to create a user-defined color palette for subsequent graphics by palette, in a col= specification in graphics functions, or in par.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine, Yee Hwa (Jean) Yang.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

See Also

image, maColorBar, maImage, maImage.func.

Examples

```
par(mfrow=c(1,4))
pal <- maPalette(low="red", high="green")
maColorBar(seq(-2,2, 0.2), col=pal, horizontal=FALSE, k=21)
pal <- maPalette(low="red", high="green", mid="yellow")
maColorBar(seq(-2,2, 0.2), col=pal, horizontal=FALSE, k=21)
pal <- maPalette()
```

```
maColorBar(seq(-2,2, 0.2), col=pal, horizontal=FALSE, k=21)
pal <- maPalette(low="purple", high="purple",mid="white")
maColorBar(seq(-2,2, 0.2), col=pal, horizontal=FALSE, k=21)
```

`mapGeneInfo` *Creating URL strings for external database links*

Description

These functions are used with `htmlPage`. The function `mapGeneInfo`, takes all the arguments and generate a character matrix of two columns. The first columns representing the name of the argument and the second columns represents the value of an argument. The function `widget.mapGeneInfo` allows the user to enter this information interactively.

Usage

```
mapGeneInfo(widget = FALSE, Gnames, Name = "pubmed", ID =
                  "genbank", ACC = "SMDacc", ...)
widget.mapGeneInfo(Gnames)
```

Arguments

`widget`	A logical value specifying if widgets should be used.
`Name`	The external database for spot description, E.g. "pubmed".
`ID`	The external database for spot ID, E.g. "operon", "Riken", "locuslink".
`ACC`	The external database for gene accession number, E.g. "genebank".
`Gnames`	An object of class `matrix`, `data.frame` or `marrayInfo` which contains description of spotted probe sequences.
`...`	Other column names

Details

The function `mapGeneInfo` generates a character matrix with the first column representing the column headings of "Gnames" and the second column representing the corresponding names in the list `URLstring`. For example, if a particular column in "Gnames" with column names "ID" contains genebank accession number, then the function `mapGeneInfo` generates a row containing "ID" in the first column and "genbank" in the second. Examples are `SFGL` and `UCBFGL`.

`URLstring` is a list contains the URL to various external database, E.g. operon, Riken, genbank. The current choices are: "pubmed", "locuslink", "riken", "SMDclid", "SMDacc", "operonh2", "operonh1" , "operonm2", "operonm1" and "genbank" . "SMDclid" and "SMDacc" are links to Stanford Microarray Databases.

Author(s)

Jean Yee Hwa Yang

Examples

```
mapGeneInfo(ID="genebank", ll="locuslink")
mapGeneInfo(ID="locuslink", Sample.ID="riken")
```

maPlot.func *Scatter-plots with fitted curves and text*

Description

This function produces scatter-plots of x vs. y. It also allows the user to highlight and annotate subsets of points on the plot, and display fitted curves from robust local regression or other smoothing procedures.

Usage

```
maPlot.func(x, y, z,
lines.func = maLowessLines(subset = TRUE, f = 0.3, col = 1:length(unique(z)), lt
     text.func = maText(),
legend.func = maLegendLines(legend = as.character(unique(z)), col = 1:length(uni
 ...)
```

Arguments

x	A "numeric" vector for the abscissa.
y	A "numeric" vector for the ordinates.
z	A vector of statistic used to stratify the data, smoothed curves are fitted separately within values of z
lines.func	A function for computing and plotting smoothed fits of y as a function of x, separately within values of z, e.g. maLoessLines.
text.func	A function for highlighting a subset of points, e.g., maText.
legend.func	A function for adding a legend to the plot, e.g. maLegendLines.
...	Optional graphical parameters, see par.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

See Also

maPlot, maLoessLines, maLegendLines, maText, plot, lowess, loess, legend.

Examples

```
# See examples for maPlot.
```

maPlot *Scatter-plots for cDNA microarray spot statistics*

Description

The function maPlot produces scatter-plots of microarray spot statistics for the classes "marrayRaw" and "marrayNorm". It also allows the user to highlight and annotate subsets of points on the plot, and display fitted curves from robust local regression or other smoothing procedures.

Usage

```
maPlot(m, x="maA", y="maM", z="maPrintTip", lines.func, text.func, legend.func,
```

Arguments

m	Microarray object of class "marrayRaw" and "marrayNorm".
x	Name of accessor function for the abscissa spot statistic, typically a slot name for the microarray object m, such as maA.
y	Name of accessor function for the ordinate spot statistic, typically a slot name for the microarray object m, such as maM.
z	Name of accessor method for the spot statistic used to stratify the data, typically a slot name for the microarray layout object (see "marrayLayout") such as maPlate or a method such as maPrintTip. If z is NULL, the data are not stratified.
lines.func	Function for computing and plotting smoothed fits of y as a function of x, separately within values of z, e.g. maLoessLines. If lines.func is NULL, no fitting is performed.
text.func	Function for highlighting a subset of points, e.g., maText. If text.func is NULL, no points are highlighted.
legend.func	Function for adding a legend to the plot, e.g. maLegendLines. If legend.func is NULL, there is no legend.
...	Optional graphical parameters, see par.

Details

This function calls the general function maPlot.func, which is not specific to microarray data. If there are more than one array in the batch, the plot is done for the first array, by default. Default graphical parameters are chosen for convenience using the function maDefaultPar (e.g. color palette, axis labels, plot title) but the user has the option to overwrite these parameters at any point.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

See Also

maPlot.func, maDefaultPar, maLoessLines, maLegendLines, maText, plot, lowess, loess, legend.

Examples

```
# To see the demo type demo(marrayPlots)

# Examples use swirl dataset, for description type ? swirl
data(swirl)

# - Default arguments
maPlot(swirl)

# Lowess fit using all spots
maPlot(swirl, z=NULL, legend.func=NULL)

# Loess fit using all spots
maPlot(swirl, z=NULL, legend.func=maLegendLines(legend="All spots",col="green"), lines.fu

# Pre-normalization MA-plot for the Swirl 81 array, with the lowess fits for
# individual grid columns and 1% tails of M highlighted
defs <- maDefaultPar(swirl[, 1], x = "maA", y = "maM", z = "maGridCol")
legend.func <- do.call("maLegendLines", defs$def.legend)
lines.func <- do.call("maLowessLines", c(list(TRUE, f = 0.3), defs$def.lines))
text.func<-maText(subset=maTop(maM(swirl)[,1],h=0.01,l=0.01), labels="o", col="violet")
maPlot(swirl[, 1], x = "maA", y = "maM", z = "maGridCol", lines.func=lines.func, text.fun
```

---

marrayInfo-class *Class "marrayInfo", description of target samples or spotted probe sequences*

Description

This class is used to store information on target samples hybridized to a batch of arrays or probe sequences spotted onto these arrays. It is not specific to the microarray context.

Objects from the Class

Objects can be created by calls of the form `new('marrayInfo', maLabels = ...., # Object of class character maInfo = ...., # Object of class data.frame maNotes = ...., # Object of class character )`

Slots

`maLabels`: Object of class `"character"`, vector of spot or array labels.

`maInfo`: Object of class `"data.frame"`. If the object of class `"marrayInfo"` is used to describe probe sequences, rows of `maInfo` correspond to spots and columns to various gene identifiers and annotations. If the object of class `"marrayInfo"` is used to describe target

samples hybridized to the arrays, rows of `maInfo` correspond to arrays and columns to various descriptions of the hybridizations, e.g., names of Cy3 and Cy5 samples, labels for the arrays etc.

`maNotes`: Object of class `"character"`, any notes on the target samples or spotted probe sequences.

Methods

[`signature(x = "marrayInfo")`: subsetting operator for spots on the array or arrays in the batch, ensures that all slots are subset properly.

maGnames<- `signature(object = "marrayRaw", value = "marrayInfo")`: slot assignment method.

maGnames<- `signature(object = "marrayNorm", value = "marrayInfo")`: slot assignment method.

maGnames<- `signature(object = "marraySpots", value = "marrayInfo")`: slot assignment method.

maInfo `signature(object = "marrayInfo")`: slot accessor method.

maInfo<- `signature(object = "marrayInfo", value = "data.frame")`: slot assignment method.

maLabels `signature(object = "marrayInfo")`: slot accessor method.

maLabels<- `signature(object = "marrayInfo", value = "character")`: slot assignment method.

maLabels<- `signature(object = "marrayInfo", value = "numeric")`: slot assignment method.

maNotes `signature(object = "marrayInfo")`: slot accessor method.

maNotes<- `signature(object = "marrayInfo", value = "character")`: slot assignment method.

maTargets<- `signature(object = "marrayRaw", value = "marrayInfo")`: slot assignment method.

maTargets<- `signature(object = "marrayNorm", value = "marrayInfo")`: slot assignment method.

print `signature(x = "marrayInfo")`: print method for `"marrayInfo"` class.

Author(s)

Jean Yang and Sandrine Dudoit

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

See Also

`marrayLayout`, `marrayRaw`, `marrayNorm`.

Examples

```
## See marrayRaw
```

Internal functions *Internal marray functions*

Description

Internal marray functions

Details

These are not to be called by the user.

marrayLayout-class *Class "marrayLayout", classes and methods for layout parameters of cDNA microarrays*

Description

This class is used to keep track of important layout parameters for two-color cDNA microarrays. It contains slots for: the total number of spotted probe sequences on the array, the dimensions of the spot and grid matrices, the plate origin of the probes, information on spotted control sequences (e.g. probe sequences which should have equal abundance in the two target samples, such as housekeeping genes). The terms *print-tip-group*, *grid*, *spot matrix*, and *sector* are used interchangeably and refer to a set of spots printed using the same print-tip.

Objects from the Class

Objects can be created by calls of the form `new('marrayLayout',`

```
maNgr = ...., # Object of class numeric
maNgc = ...., # Object of class numeric
maNsr = ...., # Object of class numeric
maNsc = ...., # Object of class numeric
maNspots = ...., # Object of class numeric
maSub = ...., # Object of class logical
maPlate = ...., # Object of class factor
maControls = ...., # Object of class factor
maNotes = ...., # Object of class character
)
```

Slots

`maNgr`**:** Object of class `"numeric"`, number of rows for the grid matrix.

`maNgc`**:** Object of class `"numeric"`, number of columns for the grid matrix.

`maNsr`**:** Object of class `"numeric"`, number of rows for the spot matrices.

`maNsc`**:** Object of class `"numeric"`, number of columns for the spot matrices.

`maNspots`**:** Object of class `"numeric"`, total number of spots on the array, equal to $maNgrxmaNgcxmaNsrxmaN$

`maSub`**:** Object of class `"logical"`, indicating which spots are currently being considered.

`maPlate`**:** Object of class `"factor"`, recording the plate origin of the spotted probe sequences.

`maControls`: Object of class `"factor"`, recording the control status of the spotted probe sequences.

`maNotes`: Object of class `"character"`, any notes concerning the microarray layout, e.g., printing conditions.

Methods

[`signature(x = "marrayLayout")`: subsetting operator for spots on the array, ensures that all slots are subset properly.

maControls<- `signature(object = "marrayLayout")`: slot assignment method.

maControls `signature(object = "marrayLayout")`: slot accessor method.

maGridCol `signature(object = "marrayLayout")`: method which computes a vector of grid column coordinates for each spot.

maGridRow `signature(object = "marrayLayout")`: method which computes a vector of grid row coordinates for each spot.

maLayout<- `signature(object = "marrayRaw", value = "marrayLayout")`: slot assignment method.

maLayout<- `signature(object = "marrayNorm", value = "marrayLayout")`: slot assignment method.

maNgc `signature(object = "marrayLayout")`: slot accessor method.

maNgc<- `signature(object = "marrayLayout", value = "numeric")`: slot assignment method.

maNgr `signature(object = "marrayLayout")`: slot accessor method.

maNgr<- `signature(object = "marrayLayout", value = "numeric")`: slot assignment method.

maNotes `signature(object = "marrayLayout")`: slot accessor method.

maNotes<- `signature(object = "marrayLayout", value = "character")`: slot assignment method.

maNsc `signature(object = "marrayLayout")`: slot accessor method.

maNsc<- `signature(object = "marrayLayout", value = "numeric")`: slot assignment method.

maNspots `signature(object = "marrayLayout")`: slot accessor method.

maNspots<- `signature(object = "marrayLayout", value = "numeric")`: slot assignment method.

maNsr `signature(object = "marrayLayout")`: slot accessor method.

maNsr<- `signature(object = "marrayLayout", value = "numeric")`: slot assignment method.

maPlate `signature(object = "marrayLayout")`: slot accessor method.

maPlate<- `signature(object = "marrayLayout")`: slot assignment method.

maPrintTip `signature(object = "marrayLayout")`: method which computes a vector of print-tip-group indices for each spot.

maSpotCol `signature(object = "marrayLayout")`: method which computes a vector of spot column coordinates for each spot.

maSpotRow `signature(object = "marrayLayout")`: method which computes a vector of spot row coordinates for each spot.

maSub `signature(object = "marrayLayout")`: slot accessor method.

maSub<- `signature(object = "marrayLayout", value = "logical")`: slot assignment method.

maSub<- `signature(object = "marrayLayout", value = "numeric")`: slot assignment method.

print `signature(x = "marrayLayout")`: print method for "marrayLayout" class.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

See Also

marrayRaw, marrayNorm, marrayInfo and [-methods.

Examples

```
## See marrayRaw
```

---

marrayNorm-class *Class "marrayNorm", classes and methods for post-normalization cDNA microarray intensity data*

Description

This class represents post-normalization intensity data for a batch of cDNA microarrays. A *batch of arrays* consists of a collection of arrays with the same layout ("marrayLayout"). The class contains slots for the average log-intensities A, the normalized log-ratios M, the location and scale normalization values, the layout of the arrays, and descriptions of the target samples hybridized to the arrays and probe sequences spotted onto the arrays.

Objects from the Class

Objects can be created by calls of the form `new('marrayNorm',`

```
maA = ...., # Object of class matrix
maM = ...., # Object of class matrix
maMloc = ...., # Object of class matrix
maMscale = ...., # Object of class matrix
maW = ...., # Object of class matrix
maLayout = ...., # Object of class marrayLayout
maGnames = ...., # Object of class marrayInfo
maTargets = ...., # Object of class marrayInfo
maNotes = ...., # Object of class character
maNormCall = ...., # Object of class call
)
```

Slots

`maA`: Object of class `"matrix"`, average log-intensities (base 2) A, rows correspond to spotted probe sequences, columns to arrays in the batch.

`maM`: Object of class `"matrix"`, intensity log-ratios (base 2) M, rows correspond to spotted probe sequences, columns to arrays in the batch.

`maMloc`: Object of class `"matrix"`, location normalization values, rows correspond to spotted probe sequences, columns to arrays in the batch.

`maMscale`: Object of class `"matrix"`, scale normalization values, rows correspond to spotted probe sequences, columns to arrays in the batch.

`maW`: Object of class `"matrix"`, spot quality weights, rows correspond to spotted probe sequences, columns to arrays in the batch.

`maLayout`: Object of class `"marrayLayout"`, layout parameters for cDNA microarrays.

`maGnames`: Object of class `"marrayInfo"`, description of spotted probe sequences.

`maTargets`: Object of class `"marrayInfo"`, description of target samples hybridized to the arrays.

`maNotes`: Object of class `"character"`, any notes concerning the microarray experiments, e.g. hybridization or scanning conditions.

`maNormCall`: Object of class `"call"`, function call for normalizing the batch of arrays.

Methods

[`signature(x = "marrayNorm")`: subsetting operator for spots on the array and arrays in the batch, ensures that all slots are subset properly.

coerce `signature(from = "marrayRaw", to = "marrayNorm")`: coerce an object of class `"marrayRaw"` into an object of class `marrayNorm`.

maA `signature(object = "marrayNorm")`: slot accessor method.

maA<- `signature(object = "marrayNorm", value = "matrix")`: slot assignment method.

maControls<- `signature(object = "marrayNorm")`: slot assignment method.

maControls `signature(object = "marrayNorm")`: slot accessor method.

maGnames `signature(object = "marrayNorm")`: slot accessor method.

maGnames<- `signature(object = "marrayNorm", value = "marrayInfo")`: slot assignment method.

maGridCol `signature(object = "marrayNorm")`: method which computes a vector of grid column coordinates for each spot.

maGridRow `signature(object = "marrayNorm")`: method which computes a vector of grid row coordinates for each spot.

maLayout `signature(object = "marrayNorm")`: slot accessor method.

maLayout<- `signature(object = "marrayNorm", value = "marrayLayout")`: slot assignment method.

maM `signature(object = "marrayNorm")`: slot accessor method.

maM<- `signature(object = "marrayNorm", value = "matrix")`: slot assignment method.

maMloc `signature(object = "marrayNorm")`: slot accessor method.

maMloc<- `signature(object = "marrayNorm", value = "matrix")`: slot assignment method.

maMscale `signature(object = "marrayNorm")`: slot accessor method.

maMscale<- `signature(object = "marrayNorm", value = "matrix")`: slot assignment method.

maNgc `signature(object = "marrayNorm")`: slot accessor method.

maNgc<- `signature(object = "marrayNorm", value = "numeric")`: slot assignment method.

maNgr `signature(object = "marrayNorm")`: slot accessor method.

maNgr<- `signature(object = "marrayNorm", value = "numeric")`: slot assignment method.

maNormCall `signature(object = "marrayNorm")`: slot accessor method.

maNotes `signature(object = "marrayNorm")`: slot accessor method.

maNotes<- `signature(object = "marrayNorm", value = "character")`: slot assignment method.

maNsamples `signature(object = "marrayNorm")`: slot accessor method.

maNsc `signature(object = "marrayNorm")`: slot accessor method.

maNsc<- `signature(object = "marrayNorm", value = "numeric")`: slot assignment method.

maNspots `signature(object = "marrayNorm")`: slot accessor method.

maNspots<- `signature(object = "marrayNorm", value = "numeric")`: slot assignment method.

maNsr `signature(object = "marrayNorm")`: slot accessor method.

maNsr<- `signature(object = "marrayNorm", value = "numeric")`: slot assignment method.

maPlate `signature(object = "marrayNorm")`: slot accessor method.

maPlate<- `signature(object = "marrayNorm")`: slot assignment method.

maPrintTip `signature(object = "marrayNorm")`: method which computes a vector of print-tip-group indices for each spot.

maSpotCol `signature(object = "marrayNorm")`: method which computes a vector of spot column coordinates for each spot.

maSpotRow `signature(object = "marrayNorm")`: method which computes a vector of spot row coordinates for each spot.

maSub `signature(object = "marrayNorm")`: slot accessor method.

maSub<- `signature(object = "marrayNorm")`: slot assignment method.

maTargets `signature(object = "marrayNorm")`: slot accessor method.

maTargets<- `signature(object = "marrayNorm", value = "marrayInfo")`: slot assignment method.

maW `signature(object = "marrayNorm")`: slot accessor method.

maW<- `signature(object = "marrayNorm", value = "matrix")`: slot assignment method.

print `signature(x = "marrayNorm")`: print method for "marrayNorm" class.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

See Also

marrayLayout, marrayRaw, marrayInfo

Examples

```
# Examples use swirl dataset, for description type ? swirl

data(swirl)

# Median normalization
mnorm<-maNorm(swirl[,2:3],norm="m")

# Object of class marrayNorm for the second and third swirl arrays
mnorm

# Function call
maNormCall(mnorm)

#  Object of class marrayInfo -- Probe sequences
maGnames(mnorm)

#  Object of class marrayInfo -- Target samples
 maTargets(mnorm)

# Density plot of log-ratios M for third array
plot(density(maM(mnorm[,2])), lwd=2, col=2, main="Density plots of log-ratios M")
lines(density(maM(swirl[,3])), lwd=2)
abline(v=0)
legend(2,1,c("Pre-normalization","Post-normalization"))
```

---

marrayRaw-class *Class "marrayRaw", classes and methods for pre-normalization cDNA microarray intensity data*

Description

This class represents pre-normalization intensity data for a batch of cDNA microarrays. A *batch of arrays* consists of a collection of arrays with the same layout ("marrayLayout"). The class contains slots for the green (Cy3) and red (Cy5) foreground and background intensities, the layout of the arrays, and descriptions of the target samples hybridized to the arrays and probe sequences spotted onto the arrays.

Objects from the Class

Objects can be created by calls of the form `new('marrayRaw',`

```
maRf = ...., # Object of class matrix
maGf = ...., # Object of class matrix
maRb = ...., # Object of class matrix
maGb = ...., # Object of class matrix
maW = ...., # Object of class matrix
maLayout = ...., # Object of class marrayLayout
maGnames = ...., # Object of class marrayInfo
maTargets = ...., # Object of class marrayInfo
maNotes = ...., # Object of class character
)
```

Slots

`maRf`: Object of class "`matrix`", red foreground intensities, rows correspond to spotted probe sequences, columns to arrays in the batch.

`maGf`: Object of class "`matrix`", green foreground intensities, rows correspond to spotted probe sequences, columns to arrays in the batch.

`maRb`: Object of class "`matrix`", red background intensities, rows correspond to spotted probe sequences, columns to arrays in the batch.

`maGb`: Object of class "`matrix`", green background intensities, rows correspond to spotted probe sequences, columns to arrays in the batch.

`maW`: Object of class "`matrix`", spot quality weights, rows correspond to spotted probe sequences, columns to arrays in the batch.

`maLayout`: Object of class "`marrayLayout`", layout parameters for the cDNA microarrays.

`maGnames`: Object of class "`marrayInfo`", description of spotted probe sequences.

`maTargets`: Object of class "`marrayInfo`", description of target samples hybridized to the arrays.

`maNotes`: Object of class "`character`", any notes concerning the microarray experiments, e.g. hybridization or scanning conditions.

Methods

[`signature(x = "marrayRaw")`: subsetting operator for spots on the array and arrays in the batch, ensures that all slots are subset properly.

coerce `signature(from = "marrayRaw", to = "marrayNorm")`: coerce an object of class "`marrayRaw`" into an object of class "`marrayNorm`".

maA `signature(object = "marrayRaw")`: function which computes average log-intensities (base 2) A for an object of class "`marrayRaw`".

maControls<- `signature(object = "marrayRaw")`: slot assignment method.

maControls `signature(object = "marrayRaw")`: slot accessor method.

maGb `signature(object = "marrayRaw")`: slot accessor method.

maGb<- `signature(object = "marrayRaw", value = "matrix")`: slot assignment method.

maGb<- `signature(object = "marrayRaw", value = "NULL")`: slot assignment method.

maGf `signature(object = "marrayRaw")`: slot accessor method.

maGf<- `signature(object = "marrayRaw", value = "matrix")`: slot assignment method.

maGnames `signature(object = "marrayRaw")`: slot accessor method.

maGnames<- `signature(object = "marrayRaw", value = "marrayInfo")`: slot assignment method.

maGridCol `signature(object = "marrayRaw")`: method which computes a vector of grid column coordinates for each spot.

maGridRow `signature(object = "marrayRaw")`: method which computes a vector of grid row coordinates for each spot.

maLayout `signature(object = "marrayRaw")`: slot accessor method.

maLayout<- `signature(object = "marrayRaw", value = "marrayLayout")`: slot assignment method.

maLG `signature(object = "marrayRaw")`: method which computes green log-intensities (base 2) for an object of class "marrayRaw".

maLR `signature(object = "marrayRaw")`: method which computes red log-intensities (base 2) for an object of class "marrayRaw".

maM `signature(object = "marrayRaw")`: method which computes intensity log-ratios (base 2) M for an object of class "marrayRaw".

maNgc `signature(object = "marrayRaw")`: slot accessor method.

maNgc<- `signature(object = "marrayRaw", value = "numeric")`: slot assignment method.

maNgr `signature(object = "marrayRaw")`: slot accessor method.

maNgr<- `signature(object = "marrayRaw", value = "numeric")`: slot assignment method.

maNotes `signature(object = "marrayRaw")`: slot accessor method.

maNotes<- `signature(object = "marrayRaw", value = "character")`: slot assignment method.

maNsamples `signature(object = "marrayRaw")`: slot accessor method.

maNsc `signature(object = "marrayRaw")`: slot accessor method.

maNsc<- `signature(object = "marrayRaw", value = "numeric")`: slot assignment method.

maNspots `signature(object = "marrayRaw")`: slot accessor method.

maNspots<- `signature(object = "marrayRaw", value = "numeric")`: slot assignment method.

maNsr `signature(object = "marrayRaw")`: slot accessor method.

maNsr<- `signature(object = "marrayRaw", value = "numeric")`: slot assignment method.

maPlate `signature(object = "marrayRaw")`: slot accessor method.

maPlate<- `signature(object = "marrayRaw")`: slot assignment method.

maPrintTip `signature(object = "marrayRaw")`: method which computes a vector of print-tip-group indices for each spot.

maRb `signature(object = "marrayRaw")`: slot accessor method.

maRb<- `signature(object = "marrayRaw", value = "matrix")`: slot assignment method.

maRb<- `signature(object = "marrayRaw", value = "NULL")`: slot assignment method.

maRf `signature(object = "marrayRaw")`: slot accessor method.

maRf<- `signature(object = "marrayRaw", value = "matrix")`: slot assignment method.

maSpotCol `signature(object = "marrayRaw")`: method which computes a vector of spot column coordinates for each spot.

maSpotRow `signature(object = "marrayRaw")`: method which computes a vector of spot row coordinates for each spot.

maSub `signature(object = "marrayRaw")`: slot accessor method.

maSub<- `signature(object = "marrayRaw")`: slot assignment method.

maTargets `signature(object = "marrayRaw")`: slot accessor method.

maTargets<- `signature(object = "marrayRaw", value = "marrayInfo")`: slot assignment method.

maW `signature(object = "marrayRaw")`: slot accessor method.

maW<- `signature(object = "marrayRaw", value = "matrix")`: slot assignment method.

print `signature(x = "marrayRaw")`: print method for "marrayRaw" class.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

See Also

marrayLayout, marrayNorm, marrayInfo.

Examples

```
# Examples use swirl dataset, for description type ? swirl
require(limma)
data(swirl)

# Object of class marrayRaw for the 4 swirl arrays
swirl

# Object of class marrayLayout
maLayout(swirl)

# Access only the first 100 spots of the third array
swirl[1:100,3]

# Accessor methods -- How many spots on the array
maNspots(swirl)
```

```
# Density plot of log-ratios M for third array
plot(density(maM(swirl[,3])))

# Assignment methods -- Replace maNotes slot
maNotes(swirl)
maNotes(swirl)<-"This is a zebrafish microarray"
maNotes(swirl)
```

`maSelectGnames` *Select genes according to the values of a few different statistics*

Description

Select genes by considering the `union` or `intersect` of multiple statistics.

Usage

```
maSelectGnames(statdata, crit1 = 50, crit2 = crit1, sub = TRUE, selectstat, oper
```

Arguments

`statdata`	A numerical matrix where the rows corresponds to genes and the columns corresponds to various statistics corresponding to a particular gene.
`crit1`	The number of points to be selected. If crit1 < 1, the crit1*100% spots with the smallest M values will be selected. If crit1 >= 1, the crit spots with the smallest M values are selected.
`crit2`	Similar to "crit1". If crit2 < 1, the crit2*100% spots with the largest M values will be selected. If crit2 >= 1, the crit2 spots with the largest M values are selected.
`sub`	A "logical" or "numeric" vector indicating the subset of genes to be consider.
`selectstat`	A integer value indicating the statistics where the final ranking is based on.
`operate`	The operation used to combined different rankings

Details

This functions calls `stat.gnames` to select say the 100 most extreme genes from various statistics and combined the different gene lists by either union or intersection.

Value

A vector of numeric values.

Author(s)

Jean Yee Hwa Yang

See Also

`stat.gnames`, `order`

Examples

```
X <- matrix(rnorm(1000), 100,10)
Xstat <- cbind(mean=apply(X, 1, mean, na.rm=TRUE),
               var=apply(X, 1, var, na.rm=TRUE))
maSelectGnames(Xstat, crit1=50)
```

---

maText *Highlight points on a plot*

Description

This function may be used to highlight a subset of points on an existing plot, such as a plot produced by `plot`, `maPlot`, or `maPlot.func`.

Usage

```
maText(subset=NULL, labels=as.character(1:length(subset)), ...)
```

Arguments

`subset`	A "logical" or "numeric" vector indicating the subset of points to highlight.
`labels`	One or more character strings or expressions specifying the text to be written.
`...`	Optional graphical parameters, see `par`.

Value

A function with bindings for `subset`, `labels`, and `....` This latter function takes as arguments `x` and `y`, the absissa and ordinates of points on the plot.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

See Also

`text`, `maPlot`, `maPlot.func`.

Examples

```
# See examples for maPlot.
```

maTop *Identify extreme values*

Description

This function determines which values in a numeric vector are above or below user supplied cut-offs.

Usage

```
maTop(x, h=1, l=1)
```

Arguments

x	A "numeric" vector.
h	A "numeric", upper cut-off.
l	A "numeric", lower cut-off.

Value

A "logical" vector indicating which entries are above or below the cut-offs.

Author(s)

Sandrine Dudoit, http://www.stat.berkeley.edu/~sandrine.

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

See Also

maPlot, maImage, quantile.

Examples

```
# See examples for maPlot.
```

maTwoSamples *Changing signs for two sample analysis*

Description

Taking target file information and flip the dye swaps experiments.

Usage

```
maTwoSamples(targetfile, normdata, Trt, Ctl, targetID = "TargetName", slidesID =
```

Arguments

targetfile	A data.frame containing target samples information.
normdata	A R object of class 'marrayNorm'
Trt	A character string representing "treatment" sample.
Ctl	A character string representing "controls" sample.
targetID	A character string representing the column name in 'targetfile' containing target samples information.
slidesID	A character string representing the column name in 'targetfile' containing the slide label.
dyesID	A character string representing the column name in 'targetfile' containing dye labeled information.
RedID	The character use to represent the Cy5 dye.
path	A character string representing the data directory. By default this is set to the current working directory (".") .
output	Save and tab delimited file

Value

An objects of 'marrayNorm' with the dye assignment adjusted.

Author(s)

Yee Hwa (Jean) Yang

opVersionID *Determine the operon oligo set ID*

Description

This functions looks the operon ID and determine whether it belongs to "Human Genome Oligo Set V1", "Human Genome Oligo Set V2", "Mouse Genome Oligo Set V1" or "Mouse Genome Oligo Set V2".

Usage

```
opVersionID(opID)
```

Arguments

opID	A character strings representing operon ID

Value

A value "operonh1", "operonh2", "operonm1" or "operonm2" to represents "Human Genome Oligo Set V1", "Human Genome Oligo Set V2", "Mouse Genome Oligo Set V1" or "Mouse Genome Oligo Set V2".

Author(s)

Jean Yee Hwa Yang

References

http://oparray.operon.com/

See Also

URLstring, htmlPage

Examples

```
opVersionID("M000205_01")
URLstring[opVersionID("M000205_01")]
```

---

plot *Scatter-plots for cDNA microarray spot statistics*

Description

The function maPlot or plot produces scatter-plots of microarray spot statistics for the classes "marrayRaw", "marrayNorm". It also allows the user to highlight and annotate subsets of points on the plot, and display fitted curves from robust local regression or other smoothing procedures.

Usage

```
## S3 method for class 'marrayRaw':
plot(x, xvar = "maA", yvar = "maM", zvar="maPrintTip", lines.func,text.func,lege
## S3 method for class 'marrayNorm':
plot(x, xvar = "maA", yvar = "maM", zvar="maPrintTip", lines.func,text.func,lege
addText(object, xvar="maA", yvar="maM", subset=NULL, labels=as.character(1:lengt
addPoints(object, xvar="maA", yvar="maM", subset=TRUE, ...)
addLines(object, xvar="maA", yvar="maM", zvar="maPrintTip", subset=TRUE, ...)
## S4 method for signature 'marrayRaw':
text(x, xvar = "maA", yvar = "maM", ...)
## S4 method for signature 'marrayNorm':
text(x, xvar = "maA", yvar = "maM", ...)
## S4 method for signature 'marrayRaw':
lines(x, xvar = "maA", yvar = "maM", zvar = "maPrintTip", ...)
## S4 method for signature 'marrayNorm':
```

```
lines(x, xvar = "maA", yvar = "maM", zvar = "maPrintTip",...)
## S4 method for signature 'marrayRaw':
points(x, xvar = "maA", yvar = "maM", ...)
## S4 method for signature 'marrayNorm':
points(x, xvar = "maA", yvar = "maM", ...)
```

Arguments

x	Microarray object of class "marrayRaw", "marrayNorm".
object	Microarray object of class "marrayRaw", "marrayNorm".
xvar	Name of accessor function for the abscissa spot statistic, typically a slot name for the microarray object x, such as maA.
yvar	Name of accessor function for the ordinate spot statistic, typically a slot name for the microarray object x, such as maM.
zvar	Name of accessor method for the spot statistic used to stratify the data, typically a slot name for the microarray layout object (see "marrayLayout") such as maPlate or a method such as maPrintTip. If zvar is NULL, the data are not stratified.
lines.func	Function for computing and plotting smoothed fits of y as a function of x, separately within values of zvar, e.g. maLoessLines. If lines.func is NULL, no fitting is performed.
text.func	Function for highlighting a subset of points, e.g., maText. If text.func is NULL, no points are highlighted.
legend.func	Function for adding a legend to the plot, e.g. maLegendLines. If legend.func is NULL, there is no legend.
subset	logical vector or numeric values indicating the subset of points to be plotted.
labels	One or more character strings or expressions specifying the text to be written.
...	Optional graphical parameters, see par.

Details

This function calls the general function maPlot.func, which is not specific to microarray data. If there are more than one array in the batch, the plot is done for the first array, by default. Default graphical parameters are chosen for convenience using the function maDefaultPar (e.g. color palette, axis labels, plot title) but the user has the option to overwrite these parameters at any point.

Author(s)

Jean Yee Hwa Yang

References

S. Dudoit and Y. H. Yang. (2002). Bioconductor R packages for exploratory analysis and normalization of cDNA microarray data. In G. Parmigiani, E. S. Garrett, R. A. Irizarry and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.

See Also

maPlot.func, maDefaultPar, maLoessLines, maLegendLines, maText, plot, lowess, loess, legend.

Examples

```
# To see the demo type demo(marrayPlots)

# Examples use swirl dataset, for description type ? swirl
data(swirl)

# Pre-normalization MA-plot for the Swirl 93 array, with the lowess fits for
# individual print-tip-groups.
# - Default arguments
plot(swirl[,3])

# Lowess fit using all spots
plot(swirl[,3], zvar=NULL, legend.func=NULL)

# Loess fit using all spots
plot(swirl[,3], zvar=NULL, legend.func=maLegendLines(legend="All spots",col="green"), lin
```

---

summary-methods *Printing summary methods for microarray objects*

Description

Print methods were defined for the microarray classes, "marrayInfo", "marrayLayout", "marrayRaw", "marrayNorm". These methods produce summaries of the intensity and textual data stored in different classes of microarray objects.

Methods

x = ANY generic print method

x = marrayLayout for an object of class "marrayLayout", the method prints main layout parameters such as the number of spots and the dimensions of the spot and grid matrices.

x = marrayInfo for an object of class "marrayInfo", the method prints the first 10 rows of the "maInfo" and "maLabels" slots.

x = marrayRaw for an object of class "marrayRaw", the method prints a short description of the microarray layout "maLayout" and the target samples hybridized to the arrays "maTargets", and a summary of the distribution of the log-ratio statistics "maM".

x = marrayNorm for an object of class "marrayNorm", the method prints a short description of the microarray layout "maLayout" and the target samples hybridized to the arrays "maTargets", and a summary of the distribution of the log-ratio statistics "maM".

---

read.Galfile *Reading GenePix Gal file*

Description

Reading a standard Gal file containing gene information.

Usage

```
read.Galfile(galfile, path = ".", info.id = c("ID", "Name"),
layout.id =c(Block="Block", Row="Row", Column="Column"),
labels = "ID", notes = "", sep = "\t", skip = NULL,   ncolumns=4, ...)
```

Arguments

galfile	a character string representing the Gal file.
path	a character string representing the data directory. By default this is set to the current working directory (".")
info.id	the column numbers or names in 'fname' that contain the required information.
layout.id	the column names in 'fname' that specified the printer layout information.
labels	the column number in fname which contains the names that the user would like to use to label spots or arrays (e.g. for default titles in maPlot.
notes	object of class character, vector of explanatory text
sep	the field separator character. Values on each line of the file are separated by this character. The default is to read a tab delimited file.
skip	the number of lines of the data file to skip before beginning to read data.
ncolumns	an integer representing the number of columns of sub-array (print-tips) on a slides.
...	further arguments to scan.

Value

gnames	An object of class marrayInfo.
layout	An object of class marrayLayout.

Author(s)

Yee Hwa (Jean) Yang

See Also

read.marrayInfo, read.marrayLayout

Examples

```
library(marray)
datadir <- system.file("swirldata", package="marray")
try <- read.Galfile(galfile="fish.gal", path=datadir)
names(try)
try$layout
try$gnames
```

`read.marrayInfo` *Create objects of class marrayInfo*

Description

This function creates objects of class `marrayInfo`. The `marrayInfo` class is used to store information regarding the target mRNA samples co-hybridized on the arrays or the spotted probe sequences (e.g. data frame of gene names, annotations, and other identifiers).

Usage

```
read.marrayInfo(fname, info.id=NULL, labels=NULL, notes=fname, sep="\t",skip=0,
```

Arguments

`fname`	the name of the file that stores information on target samples or probe sequences. This is usually a file obtained from a database.
`info.id`	the column numbers in `fname` that contain the required information.
`labels`	the column number in `fname` which contains the names that the user would like to use to label spots or arrays (e.g. for default titles in `maPlot`.
`notes`	object of class character, vector of explanatory text
`sep`	the field separator character. Values on each line of the file are separated by this character. The default is to read a tab delimited file.
`skip`	the number of lines of the data file to skip before beginning to read data.
`quote`	the set of quoting characters. By default, this is disable by setting 'quote="\"'".
`...`	further arguments to `scan`.

Value

An object of class `marrayInfo`.

Author(s)

Jean Yang, <yeehwa@stat.berkeley.edu>

References

http://www.bioconductor.org/

Examples

```
datadir <- system.file("swirldata", package="marray")

## Reading target sample information
swirl.samples <- read.marrayInfo(file.path(datadir, "SwirlSample.txt"))

## Reading probe sequence information
swirl.gnames <- read.marrayInfo(file.path(datadir, "fish.gal"),
                                info.id=4:5, labels=5, skip=21)
```

`read.marrayLayout` *Create objects of class marrayLayout*

Description

This function creates objects of class `marrayLayout` to store layout parameters for two-color cDNA microarrays.

Usage

```
read.marrayLayout(fname = NULL, ngr, ngc, nsr, nsc, pl.col = NULL, ctl.col = NUL
```

Arguments

`fname`	the name of the file that stores plate and control information. This is usually a file obtained from a database.
`ngr`	the number of rows of grids per image.
`ngc`	the number of columns of grids per image.
`nsr`	the number of rows of spots per grid.
`nsc`	the number of columns of spots per grid.
`pl.col`	the column number in `fname` that contains plate information.
`ctl.col`	the column number in `fname` that contains control information.
`sub.col`	the column number in `fname` that contains full ID information.
`notes`	object of class character, vector of explanatory text.
`skip`	the number of lines of the data file to skip before beginning to read data.
`sep`	the field separator character. Values on each line of the file are separated by this character. The default is to read a tab delimited file.
`quote`	the set of quoting characters. By default, this is disable by setting 'quote="\""'.
`...`	further arguments to `scan`.

Value

An object of class `marrayLayout`.

Author(s)

Jean Yang <yeehwa@stat.berkeley.edu>

References

http://www.bioconductor.org/

Examples

```
datadir <- system.file("swirldata", package="marray")

### Reading in control information from file
skip <-  grep("Row", readLines(file.path(datadir,"fish.gal"), n=100)) - 1
swirl.layout <- read.marrayLayout(fname=file.path(datadir,"fish.gal"), ngr=4, ngc=4,
nsr=22, nsc=24, ctl.col=4, skip=skip)

### Setting control information.
swirl.gnames <- read.marrayInfo(file.path(datadir,"fish.gal"), info.id=4:5, labels=5, ski
x <-  maInfo(swirl.gnames)[,1]
y <- rep(0, maNspots(swirl.layout))
y[x == "control"] <- 1
slot(swirl.layout, "maControls") <- as.factor(y)
```

---

read.marrayRaw *Create objects of class "marrayRaw"*

Description

This function reads in cDNA microarray data from a directory and creates objects of class "marrayRaw" from spot quantification data files obtained from image analysis software or databases.

Usage

```
read.marrayRaw(fnames, path=".", name.Gf=NULL, name.Gb=NULL, name.Rf=NULL,
name.Rb=NULL,name.W=NULL, layout=NULL, gnames=NULL, targets=NULL,
notes=NULL, skip=NULL, sep=" ", quote="\"", DEBUG=FALSE, ...)

read.GenePix(fnames = NULL, path = NULL, name.Gf = "F532 Median",
name.Gb ="B532 Median", name.Rf = "F635 Median", name.Rb = "B635 Median",
name.W ="Flags", layout = NULL, gnames = NULL, targets = NULL,
notes = NULL, skip=NULL, sep = " ", quote = "\"", DEBUG=FALSE, ...)

read.SMD(fnames = NULL, path = NULL, name.Gf = "Ch1 Intensity (Median)",
name.Gb = "Ch1 Background (Median)", name.Rf = "Ch2 Intensity (Median)",
name.Rb = "Ch2 Background (Median)", name.W = NULL, info.id = c("Name",
"Clone ID"), layout = NULL, gnames = NULL, targets = NULL, notes = NULL, skip =

read.Spot(fnames = NULL, path = ".", name.Gf = "Gmean", name.Gb =
"morphG", name.Rf = "Rmean", name.Rb = "morphR",name.W = NULL, layout =
NULL, gnames = NULL, targets = NULL, notes = NULL, skip = NULL, sep = "\t", quot

read.Agilent(fnames = NULL, path=NULL, name.Gf = "gMedianSignal", name.Gb = "gBG

widget.marrayRaw(ext = c("spot", "xls", "gpr"), skip = 0, sep = "\t",  quote = "
```

Arguments

fnames	a vector of character strings containing the file names of each spot quantification data file. These typically end in .spot for the software Spot or .gpr for the software GenePix.

path	a character string representing the data directory. By default this is set to the current working directory (".") . In the case where fnames contains the full path name, path should be set to NULL.
name.Gf	character string for the column header for green foreground intensities.
name.Gb	character string for the column header for green background intensities.
name.Rf	character string for the column header for red foreground intensities.
name.Rb	character string for the column header for red background intensities.
name.W	character string for the column header for spot quality weights.
layout	object of class "marrayLayout", containing microarray layout parameters.
gnames	object of class "marrayInfo" containing probe sequence information.
targets	object of class "marrayInfo" containing target sample information.
notes	object of class "character", vector of explanatory text.
info.id	object of class "character", vector containing the name of the colums of the SMD file containing oligo information you want to retrieve. By default, this is set to read Homo sapiens data. You may need to modify this argument if your are working on another genome.
skip	the number of lines of the data file to skip before beginning to read in data.
sep	the field separator character. Values on each line of the file are separated by this character. The default is to read a tab delimited file.
quote	the set of quoting characters. By default, this is disabled by setting quote="\"".
ext	a characters string representing suffix of different image analysis output files.
DEBUG	a logical value, if TRUE, a series of echo statements will be printed.
...	further arguments to scan.

Value

An object of class "marrayRaw".

Author(s)

Jean Yang, <yeehwa@stat.berkeley.edu>

References

http://www.bioconductor.org/.

See Also

scan, read.marrayLayout, read.marrayInfo

Examples

```
datadir <- system.file("swirldata", package="marray")

## Quick guide
swirl.targets <- read.marrayInfo(file.path(datadir, "SwirlSample.txt"))
data <- read.Spot(path=datadir, targets=swirl.targets)

## Alternate commands
```

```
skip <-  grep("Row", readLines(file.path(datadir,"fish.gal"), n=100)) - 1

swirl.layout <- read.marrayLayout(ngr=4, ngc=4, nsr=22, nsc=24)

swirl.targets <- read.marrayInfo(file.path(datadir, "SwirlSample.txt"))

swirl.gnames <- read.marrayInfo(file.path(datadir, "fish.gal"),
                                info.id=4:5, labels=5, skip=skip)

x <-  maInfo(swirl.gnames)[,1]
y <- rep(0, maNspots(swirl.layout))
y[x == "control"] <- 1
slot(swirl.layout, "maControls") <- as.factor(y)

fnames <- dir(path=datadir,pattern="spot")
swirl<- read.Spot(fnames, path=datadir,
                        layout = swirl.layout,
                        gnames = swirl.gnames,
                        targets = swirl.targets)
```

`rm.na` *Remove missing values*

Description

Remove NA's, NAN's and INF's from a vector.

Usage

```
rm.na(x)
```

Arguments

`x`	A numeric vector

Value

A vector with all NA's remove.

Author(s)

Jean Yang

Examples

```
x <- round(rnorm(10), 2)
x[c(2,4,5)] <- NA
x
rm.na(x)
```

ShowLargeObject-class

Show Large Data Object - class

Description

A virtual class including the data classes `marrayRaw`, `marrayNorm`, `marrayInfo`, `marrayLayout`, `PrinterInfo`, `RGData` and `MAData`, all of which typically contain large quantities of numerical data in vector, matrices and data.frames.

Methods

A `show` method is defined for objects of class `ShowLargeObject` which uses `printHead` to print only the leading elements or rows of components or slots which contain large quantities of data.

Author(s)

modifid from Gordon Smyth's function

stat.confband.text *Rank genes according to the value of a statistic.*

Description

Select values based on intensities binning.

Usage

```
stat.confband.text(M, A, crit1=0.025, crit2=crit1, nclass=5)
```

Arguments

`A`	a vector giving the x-coordinates of the points in the scatter plot. In the microarray context, this could be a vector of average log intensities. ie A
`M`	a vector giving the y-coordinates of the points in the scatter plot. In the microarray context, this could be a vector of log intensity ratios.
`crit1`	The number of points to be selected. If crit1 < 1, the crit1*100% spots with the smallest M values will be selected. If crit1 >= 1, the crit spots with the smallest M values are selected.
`crit2`	Similar to "crit1". If crit2 < 1, the crit2*100% spots with the largest M values will be selected. If crit2 >= 1, the crit2 spots with the largest M values are selected.
`nclass`	A single number giving the approximate number of intensity depedent groups to consider.

Value

A vector of selected spot index.

See Also

stat.gnames

Examples

```
library(marray)
data(swirl)
aveA <- apply(maA(swirl), 1, mean.na)
aveM <- apply(maM(swirl), 1, mean.na)
stat.confband.text(aveM, aveA, crit1=20, crit2=50, nclass=5)
```

---

stat.gnames *Sort Genes According to the Value of a Statistic*

Description

Lists genes and corresponding statistics in decreasing order of the statistics. This function applies to any type of statistic, including log ratios, one and two-sample t-statistics, and F-statistics. Missing values are ignored, as in sort(..., na.last=NA).

Usage

```
stat.gnames(x, gnames, crit= 50)
```

Arguments

x	a numeric vector containing the statistics for each gene. Missing values (NAs) are allowed.
gnames	a character vector containing the gene names.
crit	specifies the number of genes to be returned. If crit < 1, the crit*100% genes with the largest x values are listed. If crit >= 1, the crit genes with the largest x values are listed.

Value

List containing the following components

gnames	gene names sorted in decreasing order of the statistics in x.
t	statistics sorted in decreasing order.

Author(s)

Yee Hwa Yang, <yeehwa@stat.berkeley.edu>
Sandrine Dudoit, <sandrine@stat.berkeley.edu>

See Also

order, sort.

Examples

```
data(swirl)
aveM <- apply(maM(swirl), 1, mean.na)
Gnames <- maGeneTable(swirl)

stat.gnames(abs(aveM), Gnames, crit=10)
stat.gnames(aveM, Gnames, crit=0.01)
```

`swirl` *Gene expression data from Swirl zebrafish cDNA microarray experiment*

Description

The `swirlRaw` dataset consists of an object `swirl` of class `marrayRaw`, which represents pre-normalization intensity data for a batch of cDNA microarrays.

This experiment was carried out using zebrafish as a model organism to study early development in vertebrates. Swirl is a point mutant in the BMP2 gene that affects the dorsal/ventral body axis. Ventral fates such as blood are reduced, whereas dorsal structures such as somites and notochord are expanded. A goal of the Swirl experiment is to identify genes with altered expression in the swirl mutant compared to wild-type zebrafish. Two sets of dye-swap experiments were performed, for a total of four replicate hybridizations. For each of these hybridizations, target cDNA from the swirl mutant was labeled using one of the Cy3 or Cy5 dyes and the target cDNA wild-type mutant was labeled using the other dye. Target cDNA was hybridized to microarrays containing 8,448 cDNA probes, including 768 controls spots (e.g. negative, positive, and normalization controls spots). Microarrays were printed using 4×4 print-tips and are thus partitioned into a 4×4 grid matrix. Each grid consists of a 22×24 spot matrix that was printed with a single print-tip. Here, spot row and plate coordinates should coincide, as each row of spots corresponds to probe sequences from the same 384 well-plate.

Each of the four hybridizations produced a pair of 16-bit images, which were processed using the image analysis software package `Spot`. Raw images of the Cy3 and Cy5 fluorescence intensities for all fourhybridizations are available at `http://fgl.lsa.berkeley.edu/Swirl/index.html`.the dataset includes four output files `swirl.1.spot`, `swirl.2.spot`, `swirl.3.spot`, and `swirl.4.spot` from the `Spot` package. Each of these files contains 8,448 rows and 30 columns; rows correspond to spots and columns to different statistics from the `Spot` image analysis output. The file `fish.gal` is a gal file generated by the `GenePix` program; it contains information on individual probe sequences, such as gene names, spot ID, spot coordinates. Hybridization information for the mutant and wild-type target samples is stored in `SwirlSample.txt`.

Usage

```
data(swirl)
```

Source

These data were provided by Katrin Wuennenberg-Stapleton from the Ngai Lab at UC Berkeley. The swirl embryos for this experiment were provided by David Kimelman and David Raible at the University of Washington.

write.list *Data Output*

Description

Writes information from a list into a text file.

Usage

```
write.list(x, filename = "data", append = FALSE, closefile = TRUE, outfile)
```

Arguments

x	the list object to be written.
filename	a character string representing the file name.
append	logical; if true, the data x is appended to file filename.
closefile	logical indicating if the file connection should be closed.
outfile	file name or connections.

Details

This function may be called recursively if there exists list structure within a list.

Author(s)

Jean Yee Hwa Yang

See Also

write.table, write

Examples

```
data(swirl)
test <- list(A = 1:10, B= maM(swirl)[1:10,], C=list(x=1:10, y=1:4),
             D = summary(maA(swirl[,1])))
write.list(test, filename="test.txt")
```

write.marray *Data Output*

Description

Calls the function write.table with predefine argument. The entries in each line (row) are separated by tab.

Usage

```
write.marray(mraw, file="maRawResults.xls", val="maM", ...)
```

Arguments

mraw	the object to be written, either a marrayRaw or marrayNorm object.
file	a character string representing the file name.
val	a character string representing the slotNames to be written.
...	further arguments to write.table.

Details

see write.table

Author(s)

Jean Yee Hwa Yang

See Also

write.table, write.list

Examples

```
data(swirl)
write.marray(swirl[1:10,])
```

---

write.xls *Data Output*

Description

Calls the function write.table with predefine argument. The entries in each line (row) are separated by tab.

Usage

```
write.xls(res, file = "test.xls", ...)
```

Arguments

res	the object to be written, typically a data frame. If not, it is attempted to coerce x to a data frame.
file	a character string representing the file name.
...	further arguments to write.table.

Details

see write.table

Author(s)

Jean Yee Hwa Yang

See Also

write.table, write.list

Examples

```
data(swirl)
write.xls(maM(swirl)[1:10,], "normM.xls")
```

Index