Package 'mimager'

May 2, 2024

Version 1.29.0

Type Package

Title mimager: The Microarray Imager

Description Easily visualize and inspect microarrays for spatial artifacts.

License MIT + file LICENSE

LazyData TRUE

Depends Biobase

Imports BiocGenerics, S4Vectors, preprocessCore, grDevices, methods, grid, gtable, scales, DBI, affy, affyPLM, oligo, oligoClasses

Collate 'mimager-package.r' 'all-generics.r' 'mimage.r' 'marray.r' 'mindex.r' 'build-legend.r' 'checks.r' 'transformations.r' 'utils-affy.r' 'utils-arrays.r' 'utils-plot.r'

Suggests knitr, rmarkdown, BiocStyle, testthat, lintr, Matrix, abind, affydata, hgu95av2cdf, oligoData, pd.hugene.1.0.st.v1

VignetteBuilder knitr

URL https://github.com/aaronwolen/mimager

BugReports https://github.com/aaronwolen/mimager/issues

RoxygenNote 5.0.1

biocViews Infrastructure, Visualization, Microarray

git_url https://git.bioconductor.org/packages/mimager

git_branch devel

git_last_commit 087a076

git_last_commit_date 2024-04-30

Repository Bioconductor 3.20

Date/Publication 2024-05-01

Author Aaron Wolen [aut, cre, cph]

Maintainer Aaron Wolen <aaron@wolen.com>

Contents

---

arank *Array rank*

Description

Determines the rank of values within each matrix of a three-dimensional array.

Usage

```
arank(x, na.last = TRUE, ties.method = "first")
```

Arguments

x	a three-dimensional array of matrices
na.last	for controlling the treatment of NAs. If TRUE, missing values in the data are put last; if FALSE, they are put first; if NA, they are removed; if "keep" they are kept with rank NA.
ties.method	a character string specifying how ties are treated, see 'Details'; can be abbreviated.

Value

an array with the same dimensions as x

See Also

rank

Other array.transformations: arle

Examples

```
# microarray visualization
if (require(affydata, quietly = TRUE)) {
  data("Dilution", package = "affydata")
  x <- arank(marray(Dilution, transpose = TRUE))
}
```

`arle` *Array relative log expression*

Description

The relative log expression (RLE) quantifies the extent to which each sample in a dataset differs from a "reference" sample, which represents each probe's median value across all samples.

Usage

```
arle(x, log2 = TRUE, normalize = TRUE)
```

Arguments

`x`	a three-dimensional `array` of matrices
`log2`	should values be log_2 transformed
`normalize`	should values be quantile normalized

Value

an `array` with the same dimensions as `x`

See Also

`RLE`

Other array.transformations: `arank`

Examples

```
# microarray visualization
if (require(affydata, quietly = TRUE)) {
  data("Dilution", package = "affydata")
  x <- arle(marray(Dilution, transpose = TRUE))
}
```

marray *Microarray array*

Description

Convert S4 microarray data structures into a three-dimensional array of matrices, where each matrix corresponds to an individual sample's microarray with values arranged to reflect the physical position of the corresponding feature (i.e., probe) on the microarray surface.

Usage

```
marray(object, type = NULL, select = NULL, transpose = NULL)

## S4 method for signature 'AffyBatch'
marray(object, type = "pm", select = NULL,
  transpose = FALSE)

## S4 method for signature 'PLMset'
marray(object, type = "residuals", select = NULL,
  transpose = FALSE)

## S4 method for signature 'FeatureSet'
marray(object, type = "pm", select = NULL,
  transpose = FALSE)

## S4 method for signature 'oligoPLM'
marray(object, type = "residuals", select = NULL,
  transpose = FALSE)
```

Arguments

object	a valid Bioconductor microarray data structure
type	for microarray objects type refers to *probe type*; for objects containing probe-level models (e.g., PLMsets) type refers to the *value type* (i.e, "residuals" or "weights"). See probe type section for more information.
select	a numeric, character or logical vector indicating samples to include
transpose	TRUE (the default), ensures the reconstructed microarrays are vertically oriented, as is typically expected. Set to FALSE to return an array in the orientation strictly specified by the platform coordinates

Value

three-dimensional array

Probe types

For microarray data structures the type argument determines the *type* of probe that should be included. The following table provides a list of valid values for each supported microarray class:

AffyBatch	"all"	"pm"	"mm"	-
ExpressionFeatureSet	"all"	"pm"	"mm"	-
GeneFeatureSet	"all"	"pm"	-	"bg"
ExonFeatureSet	"all"	"pm"	"mm"	"bg"
SnpFeatureSet	"all"	"pm"	"mm"	-

Examples

```
if (require(affydata, quietly = TRUE)) {
  data("Dilution", package = "affydata")
  dilution.array <- marray(Dilution, select = c("20A", "10A"))
}
```

---

mimage *Microarray image*

Description

Visualize microarray probe intensities arranged by their physical location on the array. A false color image is produced for each sample in the microarray object and arranged in a grid.

Usage

```
mimage(object, type = NULL, select = NULL, colors = NULL,
  legend.label = NULL, nrow = NULL, ncol = NULL, fixed = FALSE,
  empty.rows = "fill", empty.thresh = 0.6, transform = NULL,
  trim = 0.01, fontsize = 12)

## S4 method for signature 'AffyBatch'
mimage(object, type = "pm", select = NULL,
  colors = NULL, legend.label = "Intensity", nrow = NULL, ncol = NULL,
  fixed = FALSE, empty.rows = "fill", empty.thresh = 0.6,
  transform = log2, trim = 0.01, fontsize = 12)

## S4 method for signature 'PLMset'
mimage(object, type = "residuals", select = NULL,
  colors = NULL, legend.label = type, nrow = NULL, ncol = NULL,
  fixed = FALSE, empty.rows = "fill", empty.thresh = 0.6,
  transform = identity, trim = 0.01, fontsize = 12)

## S4 method for signature 'FeatureSet'
```

```
mimage(object, type = "pm", select = NULL,
  colors = NULL, legend.label = "Intensity", nrow = NULL, ncol = NULL,
  fixed = FALSE, empty.rows = "fill", empty.thresh = 0.6,
  transform = log2, trim = 0.01, fontsize = 12)

## S4 method for signature 'oligoPLM'
mimage(object, type = "residuals", select = NULL,
  colors = NULL, legend.label = type, nrow = NULL, ncol = NULL,
  fixed = FALSE, empty.rows = "fill", empty.thresh = 0.6,
  transform = identity, trim = 0.01, fontsize = 12)

## S4 method for signature 'array'
mimage(object, type = NULL, select = NULL,
  colors = NULL, legend.label = "Values", nrow = NULL, ncol = NULL,
  fixed = FALSE, empty.rows = "ignore", empty.thresh = 1,
  transform = identity, trim = 0, fontsize = 12)
```

Arguments

object	a valid Bioconductor microarray data structure
type	for microarray objects type refers to *probe type*; for objects containing probe-level models (e.g., PLMsets) type refers to the *value type* (i.e, "residuals" or "weights"). See probe type section for more information.
select	a numeric, character or logical vector indicating samples to include
colors	a vector of colors used to represent probe values
legend.label	Legend label
nrow	optional, number of rows in grid layout
ncol	optional, number of columns in grid layout
fixed	Force images to assume a fixed aspect ratio corresponding to their physical dimensions
empty.rows	Should empty rows be filled with values from neighboring rows (the default, "fill"), should they be dropped ("drop") entirely, or should they be left alone ("ignore")
empty.thresh	what proportion of features must be missing from a row to consider that row empty
transform	a function to be applied to the values prior to visualizatio
trim	a percentile (default = 0.02) or range or 2 values see **trimming** section for details
fontsize	font size for labels and legend

Value

invisibly a gtable matrix of grobs

Trimming

By default, a 98% winsorization is performed prior to visualization, pulling in values outside of the 1st and 99th percentiles to their respective endpoints. This can be modified using the `trim` argument to provide either a new percentile, or a range of 2 values defining the min/max of the trimmed endpoints. Set `trim = 0` to avoid trimming entirely.

Empty Rows

As a result of platform design, the presence unindexed probes or probe selection by the user (e.g., including only `"pm"` probes), the `matrix`-representation of a microarray can contain numerous rows comprised entirely (or mostly) of missing values, which may produce undesirable rasterization artifacts in the microarray image. To avoid this, empty rows are filled with values from a neighboring row. The threshold for what constitutes an empty row can be tweaked with the `empty.thresh` argument.

Probe types

For microarray data structures the `type` argument determines the *type* of probe that should be included. The following table provides a list of valid values for each supported microarray class:

AffyBatch	"all"	"pm"	"mm"	-
ExpressionFeatureSet	"all"	"pm"	"mm"	-
GeneFeatureSet	"all"	"pm"	-	"bg"
ExonFeatureSet	"all"	"pm"	"mm"	"bg"
SnpFeatureSet	"all"	"pm"	"mm"	-

Examples

```
# standard array visualization
mimage(iris3)

# microarray visualization
if (require(affydata, quietly = TRUE)) {
  data("Dilution", package = "affydata")
  mimage(Dilution, select = c("20A", "10A"))
}
```

---

mimager *mimager: The Microarray Imager*

Description

mimager simplifies the creation of microarray images (sometimes called "pseudo-images") for the purpose of identifying problematic regional aberrations. Notable features include support for many of Bioconductor's core microarray data structures, providng compatibility with a variety of common microarray platforms, and the ability to visualize multiple microarrays simultaneously.

Details

The following Bioconductor microarray data structures are currently supported:

- `AffyBatch-class` for Affymetrix GeneChip probe level data
- `PLMset-class` for probe-level linear models fitted to Affymetrix GeneChip probe level data
- `FeatureSet-class` for storing Expression/Exon/SNP data from a variety of oligonucleotide platforms
- `oligoPLM-class` for probe-level linear models fitted to any of the `FeatureSet`-like classes

Index