MmPalateMiRNA

March 24, 2012

`MADvsMedianPlot` *Spread vs location of probe intensities*

Description

Plots of the spread (median absolute deviation) versus the location (median) of probe intensity levels.

Usage

```
MADvsMedianPlot(x, ...)

## S4 method for signature 'list'
MADvsMedianPlot(
     x,
     channel=c("G", "R"),
     group=NULL,
     subset=NULL,
     ...)
```

Arguments

`x`	A list containing `MAList` and/or `NChannelSet` objects
`channel`	The channel to use for calculating distances, one of either "G" (green or control channel) or "R" (red or experimental channel)
`group`	An optional character string specifying the name of a factor to create separate panel displays, which must be in `x$genes` (for `RGList` objects)
`subset`	An optional character vector specifying the which levels of `group` to use in creating separate panel displays
`...`	arguments to pass to `densityplot`

Methods

`signature(x = "list")` The method for `list` objects is intended to work with lists of normalized data sets, as either `MAList` or `NChannelSet` objects. This method will produce separate panel displays for each normalized data set, additionally color-coded by the `group` argument if supplied.

References

D. Sarkar, R. Parkin, S. Wyman, A. Bendoraite, C. Sather, J. Delrow, A. K. Godwin, C. Drescher, W. Huber, R. Gentleman, and M. Tewari. Quality assessment and data analysis for microRNA expression arrays. Nucleic Acids Res, 37(2):e17, 2009.

See Also

`levelplot` for pairwise distance plots between arrays, `densityplot` for density plots of log2 intensity values, and `MAplot` for MA plots.

Examples

```
data(PalateData)
reducedSet <- filterArray(PalateData, keep=c("MIR", "LET", "POSCON", "CALIB"),
                          frac=1.1, number=3, reps=4)
ndata.none <- normalizeWithinArrays(reducedSet, method="none")
ndata.median <- normalizeWithinArrays(reducedSet, method="median")
ndata.loess <- normalizeWithinArrays(reducedSet, method="loess")
ndata.quantile <- normalizeBetweenArrays(reducedSet, method="quantile")
ndata.all <- list(ndata.none, ndata.median, ndata.loess,
                  ndata.quantile)
res <- MADvsMedianPlot(ndata.all, channel="R", group="probe.type",
                 subset=c("MMU miRNAs", "Other miRNAs", "Control"))
print(res)
```

`MAplot` *MA plot*

Description

Plots of the log2 expression ratios (M values) versus the mean log2 expression values (A values) for each probe for each array.

Usage

```
MAplot(x, ...)

## S4 method for signature 'MAList'
MAplot(
    x,
    ...)

## S4 method for signature 'NChannelSet'
MAplot(
    x,
    ...)
```

Arguments

`x`	Either an `MAList` object or an `NChannelSet` object
`...`	arguments to pass to `xyplot`

Details

The so-called "MA" plot can be used to evaluate whether their is a bias associated with overall intenstity level for each array. Loess smoothed regression lines are superimposed on each plot to demonstrate the trend.

Methods

`signature(x = "MAList")` `M` and `A` values are stored as matrices in `x`

`signature(x = "NChannelSet")` `M` and `A` values are calculated from the `R` and `G` matrices returned by `assayData(x)`

See Also

`densityplot` for density plots of log2 intensity values, `levelplot` for pairwise distance plots between arrays, and `MADvsMedianPlot` for median absolute deviation versus median plots.

Examples

```
data(PalateData)
reducedSet <- filterArray(PalateData, keep=c("MIR", "LET", "POSCON", "CALIB"),
                          frac=1.1, number=3, reps=4)
ndata.quantile <- normalizeBetweenArrays(reducedSet, method="quantile")
res <- MAplot(ndata.quantile)
print(res)
```

`MmPalateMiRNA-package`

R package compendium for the analysis of murine palate two-color miRNA expression data

---

Description

R package compendium for the analysis of two-color miRNA expression data, during the period of murine embryonic palate development (gestational days (GD) 12, 13, and 14). Samples were hybridized to Miltenyi Biotech miRXplore Microarrays. The compendium covers a wide range of steps which occur in a typical miRNA microarray data analysis, including pre-processing, normalization, differential expression analysis, clustering, target identification, and gene-set enrichment analysis.

Details

Package:	MmPalateMiRNA
Type:	Package
Version:	1.0
Date:	2011-09-14
License:	LGPL
LazyLoad:	yes

The package contains several functions which are helpful during the pre-processing steps of array data, which are specific to `RGList` objects and, in the case of the `fixOutliers` and `fixMVs` functions, depend on the replicated structure of Miltenyi Biotech miRXplore Microarrays. Additionally, methods are available to produce diagnostic plots for `RGList` objects and lists of normalized data sets (`MAList` and / or `NChannelSet` objects), which build on the generic functions in `lattice`. Lastly, the main focus of the package is the package vignette "MmPalateMiRNA", which contains an extended example covering the typical steps in an miRNA microarray data analysis.

Author(s)

Guy Brock, Partha Mukhopadhyay, Vasyl Pihur, Bob Green, M. Michele Pisano Maintainer: Guy Brock <guy.brock@louisville.edu>

References

P. Mukhopadhyay, G. Brock, V. Pihur, C. Webb, M.M. Pisano, and R.M. Greene. Developmental microRNA expression profiling of murine embryonic orofacial tissue. Birth Defects Res A Clin Mol Teratol, 88(7):511-34, 2010.

R. Gentleman. Reproducible research: a bioinformatics case study. Stat Appl Genet Mol Biol, 4:Article2, 2005.

D. Sarkar, R. Parkin, S. Wyman, A. Bendoraite, C. Sather, J. Delrow, A. K. Godwin, C. Drescher, W. Huber, R. Gentleman, and M. Tewari. Quality assessment and data analysis for microRNA expression arrays. Nucleic Acids Res, 37(2):e17, 2009.

`PalateData` *Murine Secondary Palate Development miRNA Expression Data*

Description

This data set contains two-color miRNA microarray expression data obtained from mouse embryonic tissue during gestational days (GD) 12, 13, and 14, which represents the critical period of palate development in the mouse.

Usage

```
data(PalateData)
```

Format

The data are in the format of an "`RGList`", which in this case is a list with the following 9 elements:

`R` matrix of dimension 6336 x 9 which contains the red channel foreground measurements

`G` matrix of dimension 6336 x 9 which contains the green channel foreground measurements

`Rb` matrix of dimension 6336 x 9 which contains the red channel background measurements

`Gb` matrix of dimension 6336 x 9 which contains the green channel background measurements

`source` source of the images, here "imagene"

`Field.Dimensions` numeric vector giving the field dimensions of the array (Metarows, Metacols, Rows and Cols) `weights` matrix of dimension 6336 x 9 which contains the quality weights associated with each spot on the arrays

`printer` list containing information on the process used to print the spots on the arrays (number of grid rows / columns and number of spot rows / columns per grid - coincides with `Field.Dimensions`)

`genes` A `data.frame` containing information on each probe. Has the following columns:

- `Field` field position for the probe
- `Meta Row` meta row position for the probe
- `Meta Column` meta column position for the probe
- `Row` row position for the probe
- `Column` column position for the probe
- `Gene ID` unique gene identifier provided by Miltenyi Biotec
- `ID` unique probe identifier constructed by concatentating the "Gene ID" with "Meta Row", "Meta Column", "Row", and "Column" information
- `Name` name of the microRNA
- `Name.stem` base name of the microRNA
- `probe.type` type of probe, "MMU miRNAs", "Other miRNAs", "Control", "Empty", and "Other"

Details

RNA samples were isolated from mouse embryonic orofacial tissues (GD-12 - GD-14) and fluorescently labeled with Hy5 (red). Control samples (miRXplore Universal Reference) were labeled with Hy3 (green). The two sets of samples were hybridized to miRXplore Microarrays (Miltenyi Biotec) sing the a-Hyb Hybridization Station (Miltenyi Biotec). Probes for a total of 1336 mature miRNAs (from human, mouse, rat and virus), including positive control and calibration probes, were spotted in quadruplicate on each microarray. Each array included probes for 588 murine miRNAs.

Source

P. Mukhopadhyay, G. Brock, V. Pihur, C. Webb, M.M. Pisano, and R.M. Greene. Developmental microRNA expression profiling of murine embryonic orofacial tissue. Birth Defects Res A Clin Mol Teratol, 88(7):511-34, 2010.

Examples

```
data(PalateData)
```

`checkMVs` *Check an 'RGList' object for missing values*

Description

Checks each of the red and green foreground and background channels in an `RGList` object for missing values.

Usage

```
checkMVs(obj)
## S4 method for signature 'RGList'
checkMVs(obj)
```

Arguments

`obj` An `RGList` object

Value

Returns a list with the following components

`R.na` index of missing values in the red channel (`obj$R`)

`Rb.na` index of missing values in the red background channel (`obj$Rb`)

`G.na` index of missing values in the green channel (`obj$G`)

`Gb.na` index of missing values in the green background channel (`obj$Gb`)

Methods

`signature(obj = "RGList")`

See Also

`fixMVs`, `checkOutliers`, `fixOutliers`

Examples

```
data(PalateData)
mvs <- checkMVs(PalateData)
```

`checkOutliers` *Check 'RGList' object for outlying values*

Description

Checks each of the red and green foreground and background channels in an `RGList` for outlying values.

Usage

```
checkOutliers(obj)
## S4 method for signature 'RGList'
checkOutliers(obj)
```

Arguments

`obj` An `RGList` object

Details

Detects outliers outside range of mean +/- 2.665 standard deviations. Returns the indexes of outlying observations in each channel (R,Rb and G,Gb).

Value

Returns a list with the following components

`Rout`	index of outliers in the red channel (`obj$R`)
`Rbout`	index of outliers in the red background channel (`obj$Rb`)
`Gout`	index of outliers in the green channel (`obj$G`)
`Gbout`	index of outliers in the green background channel (`obj$Gb`)

Methods

`signature(obj = "RGList")`

See Also

`fixOutliers`, `checkMVs`, `fixMVs`

Examples

```
data(PalateData)
outliers <- checkOutliers(PalateData)
```

`clustPlot` *Plot expression profiles*

Description

Produces plots of clustered expression profiles, with seperate plots for each cluster. The average expression profile for each cluster is superimposed as well.

Usage

```
clustPlot(cl, mat, nrow, ncol)
```

Arguments

`cl`	integer vector giving the cluster membership for each item
`mat`	matrix of values to be plotted
`nrow`	number of rows to use for plotting
`ncol`	number of columns to use for plotting

Details

The figure region will be subdivided into `nrow` by `ncol` separate plots, using `mfrow`. The average expression profile and the number of genes belonging to each cluster is superimposed on each of the plots.

References

G.N. Brock, V. Pihur, S. Datta, and S. Datta. clValid, an R package for cluster validation. Journal of Statistical Software, 25, 2008.

See Also

See the package vignette for illustration on usage

Examples

```
## generate some fake data and cluster
set.seed(101)
mat <- matrix(rnorm(500), nrow=100, ncol=5)
clusts <- hclust(dist(mat))
cl <- cutree(clusts, 6)
clustPlot(cl, mat, 3, 2)
```

densityplot *Density plots of log2 intensity values*

Description

Plots the estimated density of log2 intensity values for two-color microarrays

Usage

```
## S4 method for signature 'RGList,missing'
densityplot(
    x,
    channel=c("G", "R"),
    group=NULL,
    subset=NULL,
    ...)

## S4 method for signature 'list,missing'
densityplot(
    x,
    channel=c("G", "R"),
    group=NULL,
    subset=NULL,
    ...)
```

Arguments

`x`	Either an `RGList` object, or a list containing `MAList` and/or `NChannelSet` objects
`channel`	The channel to use for calculating distances, one of either "G" (green or control channel) or "R" (red or experimental channel)
`group`	An optional character string specifying the name of a factor to create separate panel displays, which must be in `x$genes` (for `RGList` objects)
`subset`	An optional character vector specifying the which levels of `group` to use in creating separate panel displays
`...`	arguments to pass to `densityplot`

Methods

`signature(x = "RGList", data = "missing")` For `RGList` objects, separate panel displays can be produced for different types of probes, as determined by the `group` argument.

`signature(x = "list", data = "missing")` The method for `list` objects is intended to work with lists of normalized data sets, as either `MAList` or `NChannelSet` objects. This method will produce separate panel displays for each normalized data set, additionally subsetted by the `group` argument if supplied. The `useOuterStrips` function in the **latticeExtra** package can be used for 'outer' strip labels in the latter case.

References

D. Sarkar, R. Parkin, S. Wyman, A. Bendoraite, C. Sather, J. Delrow, A. K. Godwin, C. Drescher, W. Huber, R. Gentleman, and M. Tewari. Quality assessment and data analysis for microRNA expression arrays. Nucleic Acids Res, 37(2):e17, 2009.

See Also

`levelplot` for pairwise distance plots between arrays, `MADvsMedianPlot` for median absolute deviation versus median plots, and `MAplot` for MA plots

Examples

```
data(PalateData)
res <- densityplot(PalateData, channel="G", group="probe.type",
                   subset = c("Other miRNAs",   "MMU miRNAs", "Control"),
                   col=rep(1:3, each=3), lty=rep(1:3, 3),
                   key = list(lines=list(col=rep(1:3, each=3), lty=rep(1:3, 3)),
                     columns=3))
print(res)
```

`filterArray` *Filter an 'RGList' object to remove probes*

Description

Filters an `RGList` object to remove probes with foreground intensities not sufficiently above the background intensity. Additionally can filter probes based on character strings, to remove e.g. control probes.

Usage

```
filterArray (obj, ...)
## S4 method for signature 'RGList'
filterArray(obj, keep, frac, number, reps)
```

Arguments

`obj`	An `RGList` object
`keep`	Character vector to be used as a text filter. Only gene names (as contained in `obj$genes$Name`) which contain these text strings will be retained.

`frac`	Fraction to use as a background filter. Only those probes with foreground values (both red and green) greater than 'frac' times the background values pass the filter.
`number`	The number of samples required to pass the background filter for each probe.
`reps`	The number of replicates for each probe required to pass the filtering step.
`...`	allows additional arguments to be passed to specific methods

Value

Returns an `RGList` object identical in structure to the input object, but with reduced dimension according to the filtering steps.

Methods

```
signature(obj = "RGList")
```

See Also

`checkMVs`, `fixMVs`, `checkOutliers`, `fixOutliers`

Examples

```
data(PalateData)
reducedSet <- filterArray(PalateData, keep=c("MIR", "LET", "POSCON", "CALIB"),
                          frac=1.1, number=3, reps=4)
```

fixMVs *'Fix' an 'RGList' object with missing values.*

Description

Imputes missing values in one of the red foreground, red background, green foreground, or green background matrices of an `RGList` object created from Miltenyi Biotech miRXplore Microarrays. Uses the replicate structure of the array to impute the missing values. Implicit assumption is that only one of the four replicated values for a probe is an missing value.

Usage

```
fixMVs(mat, idx, gene.ids)
```

Arguments

`mat`	One of the red foreground (`R`), red background (`Rb`), green foreground (`G`), or green background (`Gb`) matrices in an `RGList` object, which contains missing values.
`idx`	Index of missing values, as returned by the `checkMVs` function. See examples for usage.
`gene.ids`	Vector of gene IDs for each probe. See examples for usage.

Details

The function is specific to `RGList` objects which were created from Miltenyi Biotech miRXplore Microarrays, since it depends on the replicated structure of that array (probes spotted in quadruplicate) to impute the missing probe values.

Value

Returns a matrix with the missing probe values imputed.

See Also

`checkMVs`, `checkOutliers`, `fixOutliers`

Examples

```
data(PalateData)
mvs <- checkMVs(PalateData)
PalateData$Rb <- fixMVs(PalateData$Rb, mvs$Rb.na, PalateData$genes$Gene)
```

`fixOutliers` *'Fix' an 'RGList' object with outlying values.*

Description

Imputes outlying values in one of the red foreground, red background, green foreground, or green background matrices of an `RGList` object created from Miltenyi Biotech miRXplore Microarrays. Uses the replicate structure of the array to impute the outlying values. Implicit assumption is that only one of the four replicated values for a probe is an outlying value.

Usage

```
fixOutliers(mat, idx, gene.ids)
```

Arguments

`mat`	One of the red foreground (`R`), red background (`Rb`), green foreground (`G`), or green background (`Gb`) matrices in an `RGList` object, which contains outlying values.
`idx`	Index of outlying values, as returned by the `checkOutliers` function. See examples for usage.
`gene.ids`	Vector of gene IDs for each probe. See examples for usage.

Details

The function is specific to `RGList` objects which were created from Miltenyi Biotech miRXplore Microarrays, since it depends on the replicated structure of that array (probes spotted in quadruplicate) to impute the outlying probe values.

Value

Returns a matrix with the outlying probe values imputed.

See Also

checkOutliers, checkMVs, fixMVs

Examples

```
data(PalateData)
outliers <- checkOutliers(PalateData)
PalateData$R <- fixOutliers(PalateData$R, outliers$Rout, PalateData$genes$Gene)
```

imputeKNN *Impute missing values*

Description

Imputes missing values in a data matrix using the K-nearest neighbor algorithm.

Usage

```
imputeKNN(data, k = 10, distance = "euclidean", rm.na = TRUE, rm.nan =
TRUE, rm.inf = TRUE )
```

Arguments

data	a data matrix
k	number of neighbors to use
distance	distance metric to use, one of "euclidean" or "correlation"
rm.na	should NA values be imputed?
rm.nan	should NaN values be imputed?
rm.inf	should Inf values be imputed?

Details

Uses the K-nearest neighbor algorithm, as described in Troyanskaya et al., 2001, to impute missing values in a data matrix. Elements are imputed row-wise, so that neighbors are selected based on the rows which are closest in distance to the row with missing values. There are two choices for a distance metric, either Euclidean (the default) or a correlation 'metric'. If the latter is selected, matrix values are first row-normalized to mean zero and standard deviation one to select neighbors. Values are 'un'-normalized by applying the inverse transformation prior to returning the imputed data matrix.

Value

A data matrix with missing values imputed.

Author(s)

Guy Brock

References

O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for dna microarrays. Bioinformatics, 17(6):520-5, 2001.

G.N. Brock, J.R. Shaffer, R.E. Blakesley, M.J. Lotz, and G.C. Tseng. Which missing value imputation method to use in expression profiles: a comparative study and two selection schemes. BMC Bioinformatics, 9:12, 2008.

See Also

See the package vignette for illustration on usage.

Examples

```
## generate some fake data and impute MVs
set.seed(101)
mat <- matrix(rnorm(500), nrow=100, ncol=5)
idx.mv <- sample(1:length(mat), 50, replace=FALSE)
mat[idx.mv] <- NA
imputed <- imputeKNN(mat)
```

`levelplot` *Pairwise distance between arrays*

Description

Calculates and plots the pairwise distance between arrays, as measured by the median of the absolute differences in log2 intensity values.

Usage

```
## S4 method for signature 'RGList,missing'
levelplot(
    x,
    channel=c("G", "R"),
    group=NULL,
    subset=NULL,
    ...)

## S4 method for signature 'list,missing'
levelplot(
    x,
    channel=c("G", "R"),
    order=NULL,
    ...)
```

Arguments

x	Either an RGList object, or a list containing MAList and/or NChannelSet objects
channel	The channel to use for calculating distances, one of either "G" (green or control channel) or "R" (red or experimental channel)
group	An optional character string specifying the name of a factor to create separate panel displays, which must be in x$genes (for RGList objects)
subset	An optional character vector specifying the which levels of group to use in creating separate panel displays
order	An optional numeric vector specifying the order of the arrays to use in producing the distance plots, i.e. for grouping certain arrays together
...	arguments to pass to levelplot

Methods

signature(x = "RGList", data = "missing") For RGList objects, separate panel displays can be produced for different types of probes, as determined by the group argument.

signature(x = "list", data = "missing") The method for list objects is intended to work with lists of normalized data sets, as either MAList or NChannelSet objects. This method will produce separate panel displays for each normalized data set.

References

D. Sarkar, R. Parkin, S. Wyman, A. Bendoraite, C. Sather, J. Delrow, A. K. Godwin, C. Drescher, W. Huber, R. Gentleman, and M. Tewari. Quality assessment and data analysis for microRNA expression arrays. Nucleic Acids Res, 37(2):e17, 2009.

See Also

densityplot for density plots of log2 intensity values, MADvsMedianPlot for median absolute deviation versus median plots, and MAplot for MA plots

Examples

```
data(PalateData)
res <- levelplot(PalateData[, c(1,5,9,2:4,6:8)],
                 channel="G", group="probe.type",
                 subset=c("MMU miRNAs", "Other miRNAs", "Control", "Empty"),
                 scales = list(rot=c(45, 45)))
print(res)
```

Index