Package 'metabolomics'

February 20, 2015

Type Package

Title Analysis of Metabolomics Data

Version 0.1.4

Date 2014-12-27

Author Alysha M De Livera and Jairus B Bowne

Maintainer Alysha M De Livera <alyshad@unimelb.edu.au>

Description A collection of functions to aid in the statistical analysis of metabolomic data

License GPL-2 | GPL-3

LazyLoad yes

Depends limma, crmn, gplots, R (>= 3.1.0)

NeedsCompilation no

Repository CRAN

Date/Publication 2014-12-29 19:25:29

R topics documented:

`metabolomics-package` *metabolomics package*

Description

The metabolomics package is a collection of functions for the analysis of metabolomics data.

Unless otherwise stated, the standard data input format is a data frame with sample names in the first column to be read as row names, group names in the second column, and the variables in the remaining columns (see below). These variables can be metabolites, masses, retention times, bins, areas or any other metabolomics variables of interest.

Sample	Group	met1	met2	met3	...	metN
S1	A	0.6358	0.0851	0.3665	...	1.0024
S2	A	0.5871	0.0935	0.3421	...	1.0329
S3	B	0.6650	1.0705	0.6710	...	0.7319
S4	B	0.6907	1.0341	0.6858	...	0.7376

Details

Package:	metabolomics
Type:	Package
Version:	0.1.3
Date:	2013-09-11
License:	GPL-3
LazyLoad:	yes

Author(s)

Alysha M De Livera, Jairus B Bowne

Maintainer: Alysha M De Livera <alyshad@unimelb.edu.au>

ColList *A list of colours for plots*

Description

Generates a list of colours for plots. This includes up to 15 colourblind-safe colours

Usage

```
ColList(n)
```

Arguments

n	The number of colours required.

Value

A vector of characters with the names of the colours.

Author(s)

Alysha M De Livera, Jairus B Bowne

See Also

RColorBrewer.

Examples

```
# Colour palette with colourblind-safe colours
cols <- ColList(15)
plot(1:15, 1:15, pch = 16, col = cols, cex = 2)
```

ContrastMatrix *Contrast matrix*

Description

Generates a contrast matrix with specified contrasts.

Usage

```
ContrastMatrix(contrasts, levels)
```

Arguments

contrasts	A character vector with specified contrasts.
levels	A character vector or a factor with levels in the design matrix.

Value

A contrast matrix.

Author(s)

Alysha M De Livera, Jairus B Bowne

See Also

makeContrasts.

Examples

```
ContrastMatrix(contrasts = c("A-B", "B-C"), levels = c("A", "B", "C", "D"))
```

Dendrogram *Hierarchical Cluster Analysis Dendrograms*

Description

Produces a dendrogram.

Usage

```
Dendrogram(inputdata, distmethod = "manhattan", aggmethod = "ward",
    main = "Dendrogram", cex = 1, ...)
```

Arguments

inputdata	A data frame in the input data format. This should have sample names in the first column to be read as row names, group names in the second column, and the variables in the remaining columns. These variables can be metabolites, masses, retention times, bins, areas or any other metabolomics variables of interest.
distmethod	The distance measure to be used. This must be one of "euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowski".
aggmethod	The agglomeration method to be used. This should be one of "ward", "single", "complete", "average", "mcquitty", "median" or "centroid".
main	Plot title.
cex	A numerical value giving the amount by which plotting text and symbols should be magnified relative to the default.
...	Arguments to be passed on to other methods.

Value

A dendrogram plot.

Author(s)

Alysha M De Livera, Jairus B Bowne

See Also

dist, hclust.

Examples

```
data(treated)
treated.log <- LogTransform(treated)$output
Dendrogram(treated.log)
```

`editcolnames` *Edit column names of a metabolomic data matrix*

Description

Edits column names of a metabolomic data matrix to remove the letter 'X' appearing at the beginning of metabolite names when they begin with a number.

Usage

```
editcolnames(y)
```

Arguments

y	A data matrix with metabolite names which need to be corrected.

Value

A data matrix with corrected metabolite names.

Author(s)

Alysha M De Livera, Jairus B Bowne

FoldChange *Fold change*

Description

Computes fold change for each metabolite.

Usage

```
FoldChange(inputdata, paired = FALSE, plot.hist = TRUE,
    saveoutput = FALSE, outputname = "fc.results")
```

Arguments

inputdata	A log transformed data frame in the input data format. See metabolomics for details.
paired	A logical indicating whether the data is paired.
plot.hist	A logical indicating whether a histogram of the fold changes should be plotted.
saveoutput	A logical indicating whether the output should be saved as a .csv file.
outputname	The name of the output file if the output has to be saved.

Value

The output is a matrix with fold changes for each metabolite, and a histogram of the log2 fold changes.

Author(s)

Alysha M De Livera, Jairus B Bowne

GroupSummary *Summary of the groups*

Description

Computes the mean, standard deviation, and coefficient of variation for each group in a data matrix.

Usage

```
GroupSummary(inputdata)
```

Arguments

inputdata	A data frame in the input data format. See metabolomics for details.

Value

A list containing:

means	A matrix of means for each group.
std	A matrix of standard deviation for each group.
cv	A matrix of coefficient of variation for each group.

Author(s)

Alysha M De Livera, Jairus B Bowne

Examples

```
data(treated)
treated.log <- LogTransform(treated)$output

#Means of each metabolite in each group
GroupSummary(treated.log)$means

#Standard deviation of each metabolite in each group
GroupSummary(treated.log)$std

#Coefficient of variation of each metabolite in each group
GroupSummary(treated.log)$cv
```

HeatMap *Heat map*

Description

Produces a heat map of a metabolomics data matrix optionally clustered according to specified methods.

Usage

```
HeatMap(inputdata, colramp = redgreen(75),
    scale = c("row", "column", "none"),
    dendrogram = c("column", "row", "both", "none"),
    distmethod = "euclidean", aggmethod = "complete",
    margins = c(5, 5), key = TRUE, keysize = 1.5,
    cexRow = 0.5, ColSideColors = NULL, ...)
```

Arguments

inputdata	A data frame in the input data format. See metabolomics for details.
colramp	Colours for the image.
scale	A character indicating if the values should be scaled metabolite-wise ("row") or group-wise ("column").
dendrogram	A character indicating whether to draw "none", "row", "column" or "both" dendrograms.
distmethod	The distance measure to be used. This must be one of "euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowski".
aggmethod	The agglomeration method to be used. This should be one of "ward", "single", "complete", "average", "mcquitty", "median" or "centroid".
margins	A numeric vector of length 2 containing the margins for group and metabolite names, respectively.
key	A logical indicating whether a colour key must be drawn.
keysize	A numeric indicating the size of the key.
cexRow	A numeric indicating the size of the metabolite names.
ColSideColors	A character vector indicating the colours different groups.
...	Arguments to be passed on to other methods.

Author(s)

Alysha M De Livera, Jairus B Bowne

See Also

par, heatmap.2.

LinearModelFit *Linear models*

Description

Fit a linear model to each metabolite in a metabolomics data matrix, optionally fitting ruv2 method to remove unwanted variation, and compute t-statistics, F-statistic, and corresponding p-values. Either ordinary statistics or empirical Bayes statistics can be obtained.

Usage

```
LinearModelFit(datamat,
    factormat = NULL, covariatemat = NULL, contrastmat = NULL,
    ruv2 = TRUE, k = NULL, nc = NULL,
    moderated = FALSE, padjmethod = "BH",
    saveoutput = FALSE, outputname = "results", ...)
```

Arguments

datamat	A numerical data matrix with samples in rows and metabolites in columns.
factormat	A matrix consisting of biological factors of interest.
covariatemat	A matrix consisting of optional covariates (or an intercept) to be included in the model.
contrastmat	An optional contrast matrix indicating which contrasts need to be tested to answer the biological question of interest.
ruv2	A logical indicating whether to use the ruv2 method for removing unwanted variation.
k	If ruv2=TRUE, the number of unwanted variation factors to be included in the model.
nc	If ruv2=TRUE, a vector indicating which metabolites should be used as the non-changing metabolites in the model.
moderated	A logical indicating whether moderated statistics should be computed.
padjmethod	A character string specifying p value adjustment method for multiple comparisons. Must be one of "bonferroni", "holm" (Holm 1979), "hochberg" (Hochberg 1988), "hommel" (Hommel 1988), "BH" (Benjamini and Hochberg 1995), "BY" (Benjamini and Yekutieli 2001), or "none". The default method is set to "BH".
saveoutput	A logical indicating whether the normalised data matrix should be saved as a csv file.
outputname	The name of the output file if the output has to be saved.
...	further arguments to be passed to or from methods.

Value

The result is an object of class MArrayLM, containing t statistics, F statistics, corresponding adjusted and unadjusted p-values (De Livera *et al*., 2012a, 2012b).

If moderated=TRUE, moderated statistics will be computed by empirical Bayes shrinkage of the standard errors towards a common value (Loennstedt *et al*., 2002; Smyth 2004).

Author(s)

Alysha M De Livera, Jairus B Bowne

References

Benjamini, Y., Hochberg, Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)* 57(1): 289-300.

Benjamini, Y., Yekutieli, D. (2001) The Control of the False Discovery Rate in Multiple Testing under Dependency. *The Annals of Statistics* 29(4): 1165-1188.

De Livera, A. M., Dias, D. A., De Souza, D., Rupasinghe, T., Pyke, J., Tull, D., Roessner, U., McConville, M., Speed, T. P. (2012a) Normalising and integrating metabolomics data. *Analytical Chemistry* 84(24): 10768-10776.

De Livera, A.M., Olshansky, M., Speed, T. P. (2013) Statistical analysis of metabolomics data. *Methods in Molecular Biology* In press.

Gagnon-Bartsch, Johann A., Speed, T. P. (2012) Using control genes to correct for unwanted variation in microarray data. *Biostatistics* 13(3): 539-552.

Hochberg, Y. (1988) A sharper Bonferroni procedure for multiple tests of significance. *Biometrika* 75(4): 800-802.

Holm, S. (1979) A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics* 6(2): 65-70.

Hommel, G. (1988) A stagewise rejective multiple test procedure based on a modified Bonferroni test. *Biometrika* 75(2): 383-386.

Loennstedt, I., Speed, T. P. (2002) Replicated microarray data. *Statistica Sinica* 12: 31-46.

Smyth, G. K. (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology* 3(1): 3.

See Also

eBayes, ContrastMatrix

Examples

```
##A paired study

#Log transformed data
data(treated)
treated.log <- LogTransform(treated)$output
#Separating by treatment group
treated.group<-factor(treated.log[,1],levels=unique(treated.log[,1]))
premat<-treated.log[which(treated.log[,1]=="pre"),-1]
postmat<-treated.log[which(treated.log[,1]=="post"),-1]

#Linear model fit with ordinary statistics
ordFit<-LinearModelFit(datamat=data.matrix(postmat-premat),
                  ruv2=FALSE,
                  factormat=matrix(1,nrow=nrow(postmat)))
TwoGroupPlots(treated.log[,-1],
              tstats = ordFit$t[,1],
              foldchanges =  ordFit$coef[,1],
              pvalues =  ordFit$p.val[,1],
              padjmethod = "BH",
              fcutoff = log(2),
              pcutoff = 0.05)


#Compare with the TwoGroup function
TwoGrpComp<-TwoGroup(treated.log, paired = TRUE)
TwoGroupPlots(datamat=treated.log[,-1],
              tstats = TwoGrpComp$output[, 1],
              foldchanges = TwoGrpComp$output[, 4],
```

```
              pvalues = TwoGrpComp$output[, 2],
              padjmethod = "BH",
              fcutoff = log(2),
              pcutoff = 0.05)

#Linear model fit with moderated statistics
modFit<-LinearModelFit(datamat=data.matrix(postmat-premat),
                       ruv2=FALSE,
                       moderated=TRUE,
                       factormat=matrix(1,nrow=nrow(postmat)))
TwoGroupPlots(treated.log[,-1],
              tstats = modFit$t[,1],
              foldchanges =  modFit$coef[,1],
              pvalues =  modFit$p.val[,1],
              padjmethod = "BH",
              fcutoff = log(2),
              pcutoff = 0.05)
```

`LogTransform` *Log transformation*

Description

Log transform a metabolomics data matrix.

Usage

```
LogTransform(inputdata, base = exp(1),
    saveoutput = FALSE, outputname = "log.results")
```

Arguments

`inputdata`	A data frame in the input data format. See `metabolomics` for details.
`base`	The base with respect to which logarithms are computed. The default computes the natural logarithm.
`saveoutput`	A logical indicating whether the output should be saved as a .csv file.
`outputname`	The name of the output file if the output has to be saved.

Value

The result is an object of class `metabdata`. This is a list containing the following elements:

`output`	A normalised data matrix in the input data format.
`samples`	A character string containing the names of the samples.
`groups`	A character string containing the names of the groups.

Author(s)

Alysha M De Livera, Jairus B Bowne

Examples

```
data(treated)
log.data <- LogTransform(treated)$output
log.data
```

`metabdata-class` *Metabolomic data - class*

Description

Container for output produced by the `LogTransform`, `MissingValues` and `Normalise` functions in the `metabolomics` package.

Slots/Components

This object class is a list structure that has:

`output` The data matrix that is produced by the function.

`samples` The names of the samples (the row names of the input data).

`groups` The groups in the input data.

The input data format is specified in `metabolomics`.

Author(s)

Alysha M De Livera, Jairus B Bowne

Examples

```
data(treated)
log.treated <- LogTransform(treated)
names(log.treated)
```

metabolomicsChanges *Changes made to the metabolomics package*

Description

Write first n lines of the recent changes made to the metabolomics package

Usage

```
metabolomicsChanges(n = 5)
```

Arguments

n	The number of lines to write of changelog.

Value

Text with first n lines of the recent changes made to the metabolomics package.

Author(s)

Alysha M De Livera, Jairus B Bowne

Examples

```
metabolomicsChanges(8)
```

MetBoxPlots *Box plots of a specified metabolite across groups*

Description

Produces box plots of a specified metabolite across groups.

Usage

```
MetBoxPlots(inputdata, metname, cols = NULL,
    main = NULL, cex.main = NULL, ...)
```

Arguments

inputdata	A data frame in the input data format. See metabolomics for details.
metname	A character string with the name of the metabolite.
cols	A character string indicating colours for the groups.
main	A title for the plot.
cex.main	Magnification of title relative to par()$cex.
...	Other graphical parameters. See par.

Value

Returns box plots of a specified metabolite across groups.

Author(s)

Alysha M De Livera, Jairus B Bowne

Examples

```
data(treated)
treated.log<-LogTransform(treated)$output
dev.new()
MetBoxPlots(treated.log, "Suberate", col = c("blue", "red"),
    main = "Suberate")
```

MissingValues *Missing value replacement*

Description

Replaces missing values for large metabolomics data matrices.

Usage

```
MissingValues(inputdata, column.cutoff=NULL, group.cutoff = NULL,
    complete.matrix = FALSE, seed = 100,
    saveoutput = FALSE, outputname = "missing.values.rep")
```

Arguments

inputdata	A data frame in the input data format. See metabolomics for details.
column.cutoff	A value between zero and one. If the proportion of missing values is equal to or more than the column.cutoff in all groups, that whole column will be deleted.
group.cutoff	A value between zero and one. If the proportion of missing values in a group is equal to or more than the group.cutoff, those missing values will be replaced by a random number between zero and the minimum of the entire matrix.
complete.matrix	A logical indicating whether a complete matrix is required. If TRUE, the remaining missing values (preferably only a very few) will be replaced by the average of the abundances in the rest of the group.
seed	An integer, denoting state for random number generation in R.
saveoutput	A logical indicating whether the output should be saved. If TRUE, the results will be saved as a csv file.
outputname	The name of the output file if the output has to be saved.

Value

The output is an object of class metabdata.

Author(s)

Alysha M De Livera, Jairus B Bowne

Examples

```
mat <- matrix(rnorm(50),
    nr = 5,
    dimnames = list(paste("s", 1:5, sep = ""), paste("m", 1:10, sep = "")))
mat[, 5] <- NA
mat[5, 7] <- NA
inputdata <- data.frame(Group = rep("A", 5), mat)
MissingValues(inputdata, group.cutoff = 0.7, column.cutoff = 0.8)$output
```

MmsPlot *Mean, Median, Standard deviation plot*

Description

Produces a plot showing the mean, median and standard deviation of samples or metabolites.

Usage

```
MmsPlot(inputdata, variables = c("samples", "metabolites"), refvec = NULL,
    main = "Mean, Median and Standard Deviation", ...)
```

Arguments

inputdata	A data frame in the input data format. See metabolomics for details.
variables	A character string indicating whether the samples or the metabolites should be plotted.
refvec	A vector of reference values to be plotted, such as an internal standard or sample weights.
main	A title for the plot.
...	Other graphical parameters. See par.

Value

A plot showing the mean, median and standard deviation of samples or metabolites and an object of class results.

Author(s)

Alysha M De Livera, Jairus B Bowne

Examples

```
data(treated)
treated.log<-LogTransform(treated)$output
MmsPlot(treated.log)
MmsPlot(treated.log, refvec = treated.log[, 2])
```

`Normalise` *Normalisation*

Description

Normalise a metabolomic data matrix according to a specified method.

Usage

```
Normalise(inputdata,
    method = c("median","mean","sum","ref", "is", "nomis","ccmn", "ruv2"),
    refvec = NULL, ncomp = NULL, k = NULL, nc = NULL,
    saveoutput = FALSE, outputname = NULL)
```

Arguments

`inputdata`	A log transformed data frame in the input data format. See `metabolomics` for details.
`method`	A character string indicating the required normalization method. Must be one of "`median`", "`mean`", "`sum`", "`ref`", "`is`", "`nomis`", "`ccmn`" or "`ruv2`". See Details for information.
`refvec`	A vector of internal standards to be used with the method "`is`", or a reference vector to be used with the method "`ref`".
`ncomp`	Number of PCA components to be used for the "`ccmn`" method. If `NULL`, this will be determined by cross validation as described by Redestig (2012).
`k`	Number of factors of unwanted variation to be included in the "`ruv2`" model.
`nc`	A vector indicating which metabolites should be used as the non-changing metabolites in the "`ruv2`" model, or as multiple internal standards in the "`ccmn`", "`nomis`" and "`ruv2`" methods.
`saveoutput`	A logical indicating whether the normalised data matrix should be saved as a .csv file.
`outputname`	The name of the output file if the output has to be saved.

Details

The normalisation methods based on scaling include normalisation to a total sum, or by the median or mean of each sample, and are denoted by "sum", "median", and "mean" respectively. The method "ref" normalises the metabolite abundances to a specified reference vector.

The normalisation methods based on internal or external standards include "is" which uses a single standard, Cross-contribution Compensating Multiple internal standard Normalisation, "ccmn" (Redestig *et al*., 2009); normalization using optimal selection of multiple internal standards, "nomis" (Sysi-aho *et al*. 2007); and "ruv2" (De Livera *et al*. 2012a).

The Remove Unwanted Varitation "ruv2" method generates a matrix of unwanted variation using non-changing metabolites including any internal or external standards. This matrix of unwanted variation can then be used for identifying differentiallly abundant metabolites in the LinearModelFit function. The RUV2 method attempts to capture both observed and unobserved technical and biological variation (De Livera *et al*. 2012a, Gagnon-Bartsch *et al*. 2012).

An overview of these normalisation methods are given by De Livera *et al*. (2012a, 2012b). Both the "ruv2" and "ccmn" methods use the factors of interest (groups), and therefore should not be used for those unsupervised methods where the groups must be treated as unknown.

Value

The result is an object of class metabdata.

Author(s)

Alysha M De Livera, Jairus B Bowne

References

De Livera, A. M., Dias, D. A., De Souza, D., Rupasinghe, T., Pyke, J., Tull, D., Roessner, U., McConville, M., Speed, T. P. (2012a) Normalising and integrating metabolomics data. *Analytical Chemistry* 84(24): 1076-10776.

De Livera, A.M., Olshansky, M., Speed, T. P. (2013) Statistical analysis of metabolomics data. *Methods in Molecular Biology* In press.

Gagnon-Bartsch, Johann A., Speed, T. P. (2012) Using control genes to correct for unwanted variation in microarray data. *Biostatistics* 13(3): 539-552.

Redestig, H., Fukushima, A., Stenlund, H., Moritz, T., Arita, M., Saito, K., Kusano, M. (2009) Compensation for systematic cross-contribution improves normalization of mass spectrometry based metabolomics data. *Analytical Chemistry* 81(19): 7974-7980.

Sysi-Aho, M., Katajamaa, M., Yetukuri, L., Oresic, M. (2007) Normalization method for metabolomics data using optimal selection of multiple internal standards. *BMC Bioinformatics* 8(1): 93.

See Also

normFit.

Examples

```
## Reading the data
data(mix)
Y <- log(exprs(mix))
inputdata <- data.frame(pData(mix)$type, t(Y))
batch <- pData(mix)$runorder
nc <- which(with(fData(mix), tag == "IS")==TRUE)

## Normalise by the median
norm_med <- Normalise(inputdata, method = "median")

## Normalise by an internal standard
norm_is <- Normalise(inputdata, method = "is",
    refvec=inputdata[, nc[1]])

## Normalise by a reference vector, in this case an internal standard
norm_ref <- Normalise(inputdata, method = "ref",
    refvec = inputdata[, nc[1]])

## Normalise by the sum
norm_sum <- Normalise(inputdata, method = "sum")

## Normalise by the NOMIS method
norm_nomis <- Normalise(inputdata, method = "nomis", nc = nc)

## Normalise by the CCMN method
norm_ccmn <- Normalise(inputdata, method = "ccmn", nc = nc, ncomp = 2)

## Normalise using RUV2 method
norm_ruv2 <- Normalise(inputdata, method = "ruv2", nc = nc, k = 9)

## Pca Plots of unwanted variation
PcaPlots(data.frame(batch, norm_ruv2$output[, -1]),
    main = "Unwanted batch variation")
```

`PcaPlots` *PCA plots*

Description

Produces PCA plots.

Usage

```
PcaPlots(inputdata, y.axis = 1, x.axis = 2, center=TRUE, scale = TRUE, main = NULL,
    varplot = FALSE, multiplot = FALSE, n = 5, cols = NULL, ...)
```

Arguments

`inputdata`	A log transformed data frame in the input data format. See `metabolomics` for details.
`y.axis`	The principal component to be plotted on the *y*-axis.
`x.axis`	The principal component to be plotted on the *x*-axis.
`center`	A logical indicating whether the variables should be scaled to have zero mean.
`scale`	A logical indicating whether the variables should be scaled to have unit variance before the analysis takes place.
`main`	Plot title.
`varplot`	A logical indicating whether explained variance should be plotted.
`multiplot`	If `TRUE`, pairs plots of the first *n* principal components will be plotted.
`n`	The number of principal components to be plotted if `multiplot=TRUE`. The default value is set to 5.
`cols`	A character string with colours to be used.
`...`	Arguments to be passed on to other methods.

Author(s)

Alysha M De Livera, Jairus B Bowne

See Also

`prcomp`.

Examples

```
data(treated)
treated.log <- LogTransform(treated)$output
PcaPlots(treated.log, scale=FALSE, center=TRUE, multiplot = TRUE, varplot = TRUE)
```

`Random seed` *Random seed*

Description

See `.Random.seed`

results-class *Metabolomic data results - class*

Description

The container for output from the MmsPlot and TwoGroup functions in the metabolomics package.

Details

This object class is a list structure containing the output that is produced by the function.

Author(s)

Alysha M De Livera, Jairus B Bowne

Examples

```
data(treated)
treated.twogroup <- TwoGroup(treated)
names(treated.twogroup)
```

RlaPlots *RLA plots*

Description

Produces within group and across group relative log abundance plots to visualise a metabolomics data matrix.

Usage

```
RlaPlots(inputdata, type = c("ag", "wg"), cols = NULL,
    cex.axis = 0.8, las = 2, ylim = c(-2, 2), oma = c(7, 4, 4, 2) + 0.1, ...)
```

Arguments

inputdata	A log transformed data frame in the input data format. See metabolomics for details.
type	A character string indicating whether within group ("wg") or across group ("ag") RLA plots need to be plotted.
cols	A character string with colours to be used for the box plots.
cex.axis	The magnification to be used for *x*- and *y*-labels relative to the current setting of cex.
las	A numeric in 0, 1, 2, 3 denoting the style of axis labels. See par.
ylim	A vector containing *y*-axis limits.
oma	A vector giving the size of the outer margins.
...	Other graphical parameters. See par.

Details

Across group RLA plots is obtained by standardising the metabolites by removing the median from each metabolite across all groups. The boxplots of these scaled metabolites can be used to compare the behaviour of metabolites in the samples between groups.This gives a visual inspection of variation in the data across groups.

For within group RLA plots, each metabolite is scaled by removing the median within each group. Boxplots of these can be used to visualise the tightness of the replicates within groups, and should have a median close to zero and low variation around the median.

See De Livera *et al*. 2012a and 2012b for further details.

Author(s)

Alysha M De Livera, Jairus B Bowne

References

De Livera, A. M., Dias, D. A., De Souza, D., Rupasinghe, T., Pyke, J., Tull, D., Roessner, U., McConville, M., Speed, T. P. (2012a) Normalising and integrating metabolomics data. *Analytical Chemistry* 84(24): 10768-10776.

De Livera, A.M., Olshansky, M., Speed, T. P. (2012b) Statistical analysis of metabolomics data. *Methods in Molecular Biology* In press.

Examples

```
data(treated)
treated.log <- LogTransform(treated)$output

#Across group RLA plot for comparing samples across groups
RlaPlots(treated.log, ylim = c(-5, 5))

#Within group RLA plot for comparing the replicates
RlaPlots(treated.log, "wg", ylim = c(-3, 3), cols = c("green","purple"))
```

ScatterPlot *Scatter plot*

Description

Produces a scatter plot.

Usage

```
ScatterPlot(inputdata, y.axis, x.axis, ylab = y.axis, xlab = x.axis, ...)
```

Arguments

inputdata	A log transformed data frame in the input data format. See metabolomics for details.
y.axis	A character with the name of the metabolite to be plotted on the *y*-axis.
x.axis	A character with the name of the metabolite to be plotted on the *x*-axis.
ylab	*y*-axis label.
xlab	*x*-axis label.
...	Other graphical parameters. See par.

Author(s)

Alysha M De Livera, Jairus B Bowne

Examples

```
data(treated)
ScatterPlot(treated, "Suberate", "Succinate")
```

---

treated *A metabolomics study with paired observations.*

Description

A data frame with data collected from several subjects before and after a specific treatment.

Usage

```
data(treated)
```

Author(s)

Alysha M De Livera, Jairus B Bowne

Examples

```
data(treated)
treated
```

TwoGroup *Comparing two biological conditions in a metabolomics data matrix*

Description

This function computes fold changes, t statistics, p-values, and adjusted p-values for each metabolite given a series of replicates and two biological conditions.

Usage

```
TwoGroup(inputdata, alternative = "two.sided", paired = FALSE,
    padjmethod = "BH", saveoutput = FALSE, outputname = "results", ...)
```

Arguments

inputdata	A log transformed data frame in the input data format. See metabolomics for details.
alternative	A character string specifying the alternative hypothesis for the t tests. This should be one of "two.sided", "greater" or "less". The default is set to "two.sided".
paired	A logical indicating whether the t-test should be paired or not. The default is set to "FALSE".
padjmethod	A character string specifying p-value adjustment method for multiple comparisons. Must be one of "bonferroni", "holm" (Holm 1979), "hochberg" (Hochberg 1988), "hommel" (Hommel 1988), "BH" (Benjamini and Hochberg 1995), "BY" (Benjamini and Yekutieli 2001), or "none". The default method is set to "BH".
saveoutput	A logical indicating whether the output should be saved. If TRUE the output will be saved as a .csv file.
outputname	The name of the output file if the output has to be saved.
...	Further arguments to be passed to or from methods.

Value

The result is an object of class "results".

output	A matrix with t statistics, p-values, adjusted p-values, fold changes, and standard errors.

Author(s)

Alysha M De Livera, Jairus B Bowne

References

Benjamini, Y., Hochberg, Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)* 57(1): 289-300.

Benjamini, Y., Yekutieli, D. (2001) The Control of the False Discovery Rate in Multiple Testing under Dependency. *The Annals of Statistics* 29(4): 1165-1188.

Hochberg, Y. (1988) A sharper Bonferroni procedure for multiple tests of significance. *Biometrika* 75(4): 800-802.

Holm, S. (1979) A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics* 6(2): 65-70.

Hommel, G. (1988) A stagewise rejective multiple test procedure based on a modified Bonferroni test. *Biometrika* 75(2): 383-386.

Examples

```
data(treated)
treated.log <- LogTransform(treated)$output
TwoGroup(treated.log, paired = TRUE,
    saveoutput = TRUE, outputname = "results")
```

`TwoGroupPlots` *Plots of differential metabolites*

Description

Produces plots for visualising differential metabolites.

Usage

```
TwoGroupPlots(datamat, tstats, foldchanges, pvalues,
padjmethod = "BH", fcutoff = log(2), pcutoff = 0.05, cexval = 0.7)
```

Arguments

`datamat`	A numerical data matrix with samples in rows and metabolites in columns
`tstats`	A vector of t statistics.
`foldchanges`	A vector of fold changes.
`pvalues`	A vector of corresponding p-values.
`padjmethod`	A character string specifying p-value adjustment method for multiple comparisons. Must be one of "`bonferroni`", "`holm`" (Holm 1979), "`hochberg`" (Hochberg 1988), "`hommel`" (Hommel 1988), "`BH`" (Benjamini and Hochberg 1995), "`BY`" (Benjamini and Yekutieli 2001), or "`none`". The default method is set to "`BH`".
`fcutoff`	A numeric indicating the fold change cut off. The default is set to 2.
`pcutoff`	A numeric indicating the adjusted p-value cut off. The default is set to 0.05.
`cexval`	The font size of the text labels.

Value

A list containing:

IncreasedMets	Names of increased metabolites.
DecreasedMets	Names of decreased metabolites.
DifferentialMets	Names of all differential metabolites.

Author(s)

Alysha M De Livera, Jairus B Bowne

References

Benjamini, Y., Hochberg, Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)* 57(1): 289-300.

Benjamini, Y., Yekutieli, D. (2001) The Control of the False Discovery Rate in Multiple Testing under Dependency. *The Annals of Statistics* 29(4): 1165-1188.

Hochberg, Y. (1988) A sharper Bonferroni procedure for multiple tests of significance. *Biometrika* 75(4): 800-802.

Holm, S. (1979) A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics* 6(2): 65-70.

Hommel, G. (1988) A stagewise rejective multiple test procedure based on a modified Bonferroni test. *Biometrika* 75(2): 383-386.

Examples

```
data(treated)
treated.log <- LogTransform(treated)$output
results <- TwoGroup(treated.log, paired = TRUE)$output
TwoGroupPlots(treated.log[,-1], tstats = results[, 1],
    foldchanges = results[, 4], pvalues = results[, 2], padjmethod = "BH",
    fcutoff = log(2), pcutoff = 0.05)
```

VolcanoPlot *Volcano plot*

Description

Produces a volcano plot given fold changes and p-values.

Usage

```
VolcanoPlot(folds, pvals, cexcutoff = 0.7, cexlab = 0.5, plimit = 0.05,
fclimit = 2, xlab = 'log2 Fold Change', ylab = '-log10 t-Test P-value',
main = "Volcano Plot", ...)
```

Arguments

folds	A vector of fold changes with metabolite names.
pvals	A vector of corresponding p-values with metabolite names.
cexcutoff	Font size of the cut-off labels.
cexlab	Font size of the variable labels.
plimit	A numeric indicating the p value cutoff. The default is set to 0.05.
fclimit	A numeric indicating the lower fold cutoff. The default is set to 2.
xlab	*x*-axis label.
ylab	*y*-axis label
main	Plot title.
...	Other graphical parameters. See par.

Author(s)

Alysha M De Livera, Jairus B Bowne

See Also

TwoGroup, TwoGroupPlots.

Examples

```
data(treated)
treated.log <- LogTransform(treated, base = 2)$output
results <- TwoGroup(treated.log)$output
pval <- results[, 2]
fc <- results[, 4]
VolcanoPlot(fc, pval, cexlab = 0.8)
```

Index