Package ‘NormalizeMets’

March 23, 2018

Type Package

Title Analysis of Metabolomics Data

Version 0.25

Date 2018-03-21

Author Alysha M De Livera, Gavriel Olshansky

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

Description Metabolomics data are inevitably subject to a component of unwanted variation, due to factors such as batch effects, matrix effects, and confounding biological variation. This package is a collection of functions designed to implement, assess, and choose a suitable normalization method for a given metabolomics study (De Livera et al (2015) <doi:10.1021/ac502439y>).

License GPL (>= 2)

LazyLoad yes

LazyData true

VignetteBuilder knitr

Depends R (>= 3.4.0)

Imports impute, crmn, limma, plotly, e1071, AUC, htmlwidgets, ggplot2, GGally, grid, rmarkdown, knitr

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-03-23 08:58:27 UTC

R topics documented:

`NormalizeMets-package` *NormalizeMets package*

Description

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

Metabolomics data are inevitably subject to a component of unwanted variation, due to factors such as batch effects, matrix effects, and confounding biological variation. This package is a collection of functions designed to implement, assess, and choose a suitable normalization method for a given metabolomics study.

Details

Run `browseVignettes("NormalizeMets")` or `vignette("NormalizeMets_vignette", package = "NormalizeMets")` for more details.

Author(s)

Alysha M De Livera, Gavriel Olshansky

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

`alldata-class` *Metabolomic alldata- class*

Description

The container for output from most of the functions in the `NormalizeMets` package.

Details

This object class is a list containing the output matrices that are produced by the function. See NormalizeMets Vignette for details.

Author(s)

Alysha M De Livera, Gavriel Olshansky

Examples

```
data(alldata_eg)
dataview(alldata_eg$featuredata)
dataview(alldata_eg$sampledata)
dataview(alldata_eg$metabolitedata)
```

`alldataC` *A designed metabolomics dataset*

Description

alldataC is a subsample of the data that consists of featuredata, sampledata and metabolitedata as described by redestig et al 2009

Author(s)

Alysha M De Livera, Gavriel Olshansky

References

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.

Examples

```
data(alldataC)
dataview(alldataC$featuredata)
dataview(alldataC$sampledata)
```

`alldatacheck` *Check all data*

Description

Some preliminary checks of the data formats for the NormalizeMets package

Usage

```
alldatacheck(featuredata, sampledata = NULL, metabolitedata = NULL)
```

Arguments

`featuredata` featuredata A data frame in the featuredata format. This is a dataframe with metabolites in columns and samples in rows. Unique sample names should be provided as row names. See NormalizeMets Vignette for details.

`sampledata` A dataframe that contains sample specific information. Unique sample names should be provided as row names.

`metabolitedata` A dataframe that contains metabolite specific information.

Author(s)

Alysha M De Livera, Gavriel Olshansky

alldata_eg *LC-MS metabolomics dataset*

Description

alldata_eg is a subset of a cohort study dataset described by De Livera et al. (2015), and consists of featuredata, sampledata and metabolitedata

Author(s)

Alysha M De Livera, Gavriel Olshansky

References

De Livera, Alysha M, M. Aho-Sysi, Laurent Jacob, J. Gagnon-Bartch, Sandra Castillo, J.A. Simpson, and Terence P. Speed. 2015. Statistical methods for handling unwanted variation in metabolomics data. *Analytical Chemistry* 87 (7). American Chemical Society: 3606-15.

Examples

```
data(alldata_eg)
dataview(alldata_eg$featuredata)
dataview(alldata_eg$metabolitedata)
dataview(alldata_eg$sampledata)
```

ComparePcaPlots *Compare PCA Plots*

Description

Produces a comparison of principal component multiplots

Usage

```
ComparePcaPlots(lfeaturedata, lgroupdata, saveplot = FALSE, plotname = "",
  savetype = c("png", "bmp", "jpeg", "tiff", "pdf"), y.axis = 1,
  x.axis = 2, center = TRUE, scale = TRUE, lmain = NULL, n = 3,
  showlegend = TRUE, usercols = NULL, cex_val = 0.7, ...)
```

Arguments

`lfeaturedata` A list of data frames in the featuredata format. This is a dataframe with metabolites in columns and samples in rows. Unique sample names should be provided as row names. See NormalizeMets Vignette for details.

`lgroupdata` A list of data frames or a vectors with group names.

`saveplot` A logical indication whether to save the plot produced.

`plotname` Name of the output file if the file is to be saved. This is the general name for all the graphs and the specific type prefix will be added automatically.

`savetype` The required format for the plot to be saved in. Threre is a choice of `"png"`,`"bmp"`,`"jpeg"`,`"tiff"`,`"pdf"` type files.

`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.

`lmain` A list of plot titles

`n` The number of principal components to be plotted. The default value is set to 3.

`showlegend` A logical indication whether to print a legend for the plot.

`usercols` A character string with colours to be used or TRUE for ColList to be used (ColList is the defualt set of colours used in other plots in this package). A value other then `NULL` will automatically set `showlegend` to `FALSE`.

`cex_val` A numeric indicating the size of some text elements.

`...` Arguments to be passed on to other methods.

Author(s)

Alysha M De Livera, Gavriel Olshansky.

Examples

```
   data(mixdata)
#    ComparePcaPlots(list(mixdata$featuredata,mixdata$featuredata*1.2),
#            list(mixdata$sampledata[,3],mixdata$sampledata[,3]))
```

`ComparePvalHist` *p-value Histogram*

Description

Make a p-value Histogram of results

Usage

```
ComparePvalHist(lpvals = NULL, normmeth = NULL, saveplot = FALSE,
  savetype = c("png", "bmp", "jpeg", "tiff", "pdf"), xlab = "P-Values",
  ylab = "Frequency", ylim = NULL, xlim = c(0, 1), col = "grey",
  plotname = "PvalHistComp", ...)
```

Arguments

`lpvals` A list with vectors of p-values

`normmeth` A vector with the normalization method used corresponding in order to the data supplied to be displayed on the plot.

`saveplot` A logical indication whether to save the plot produced.

`savetype` The required format for the plot to be saved in. Threre is a choice of `"png"`,`"bmp"`,`"jpeg"`,`"tiff"`,`"pdf` type files.

`xlab` x-axis label

`ylab` y-axis label

`ylim` y-axis limit

`xlim` x-axis limit

`col` a colour to be used to fill the bars. The default of NULL yields unfilled bars.

`plotname` Name of the output file if the file is to be saved. This is the general name for all the graphs and the specific type prefix will be added automatically.

`...` Other parameters for the `hist` function.

Author(s)

Alysha M De Livera, Gavriel Olshansky

Examples

```
data("alldata_eg")
featuredata_eg<-alldata_eg$featuredata
dataview(featuredata_eg)
sampledata_eg<-alldata_eg$sampledata
dataview(sampledata_eg)
metabolitedata_eg<-alldata_eg$metabolitedata
dataview(metabolitedata_eg)
```

```
logdata <- LogTransform(featuredata_eg)
dataview(logdata$featuredata)
imp <-  MissingValues(logdata$featuredata,sampledata_eg,metabolitedata_eg,
                     feature.cutof=0.8, sample.cutoff=0.8, method="knn")
dataview(imp$featuredata)

#Linear model fit using unadjusted data
factormat<-model.matrix(~gender +Age +bmi, sampledata_eg)
unadjustedFit<-LinearModelFit(featuredata=imp$featuredata,
                             factormat=factormat,
                             ruv2=FALSE)
unadjustedFit

#Linear model fit using `is' normalized data
Norm_is <-NormQcmets(imp$featuredata, method = "is",
                    isvec = imp$featuredata[,which(metabolitedata_eg$IS ==1)[1]])
isFit<-LinearModelFit(featuredata=Norm_is$featuredata,
                     factormat=factormat,
                     ruv2=FALSE)
isFit

#Linear model fit with ruv-2 normalization
ruv2Fit<-LinearModelFit(featuredata=imp$featuredata,
                       factormat=factormat,
                       ruv2=TRUE,k=2,
                       qcmets = which(metabolitedata_eg$IS ==1))
ruv2Fit

#Exploring metabolites associated with age
lpvals_age<-list(unadjusted=unadjustedFit$p.value[,"Age"],
                is=isFit$p.value[,"Age"],
                ruv2=ruv2Fit$p.value[,"Age"])

ComparePvalHist(lpvals = lpvals_age,ylim=c(0,40),
     normmeth = c("unadjusted","is","ruv2"))
```

`CompareRlaPlots` *CompareRlaPlots*

Description

Produces within group and across group relative log abundance plots to visually compare between different normalization methods

Usage

```
CompareRlaPlots(lfeaturedata, groupdata, normmeth = NULL, type = c("ag",
  "wg"), yrange = NULL, plottitle = "RLA plots Comparison",
  saveinteractiveplot = FALSE, savenoninteractive = FALSE,
  interactivesavename = "RlaPlotsComp", ...)
```

Arguments

`lfeaturedata` A list containing data frames in the featuredata format.

`groupdata` A vector containing group information.

`normmeth` A vector with the normalization method used corresponding in order to the data supplied to be displayed on the plot.

`type` A character string indicating whether within group ("wg") or across group ("ag") RLA plots need to be plotted.

`yrange` A vector with the first entry corresponding to the minimum y-axis value and the second to the maximum y-axis value to show as default on all the plots. This can be zoomed out as the plot is interactive.

`plottitle` The title to be displayed on the plot.

`saveinteractiveplot`
A boolean indicating whether the interactive plto should be save as a `.html` file.

`savenoninteractive`
A boolean indicating whether a `.png` version of the plto should be save as a.

`interactivesavename`
A character string to be used as the filename for saving the interactive plot.

`...` Other arguments to `RlaPlots` function.

Examples

```
data(UVdata)
# Not RUN due to user input; we set k=1 each and saved normalized data as uv_ruvrandclust
# uv_ruvrand_norm<-NormQcmets(featuredata=UVdata$featuredata,
#                            method=”ruvrandclust”,
#                            qcmets=which(UVdata$metabolitedata$neg_control==1),
#                            k=1)
data(”uv_ruvrandclust”)
lfeaturedata<-list(unadj=UVdata$featuredata,ruv=uv_ruvrandclust$featuredata,
   ruvuv=uv_ruvrandclust$uvdata)
#CompareRlaPlots(lfeaturedata,
#             groupdata=interaction(UVdata$sampledata$temperature,UVdata$sampledata$instrument),
#                normmeth=c(”Unadjusted:”, ”RUVrandclust normalized:”,
#                           ”RUVrandclust: removed uv:”),
#               yrange=c(-3,3))
```

`CompareVolcanoPlots` *Compare Volcano plots*

Description

Produces a volcano plot that can be sued to compare between different normalisation methods.

Usage

```
CompareVolcanoPlots(lcoef, lpvals, normmeth = NULL, plimit = 0.05,
  coeflimit = 1, yrange = NULL, negcontrol = NULL, poscontrol = NULL,
  xlab = "Coefficients", ylab = "-log(p-value)", labelunderlim = FALSE,
  labelsig = FALSE, saveinteractiveplot = FALSE,
  interactiveplotname = "interactiveVolcanPlot", ...)
```

Arguments

`lcoef`	A list of vectors of coefficients with metabolite names, each vector corresponding to a different noramlization method.
`lpvals`	A list of vector of corresponding p-values.
`normmeth`	A vector with the normalization method used corresponding in order to the data supplied to be displayed on the plot.
`plimit`	A numeric indicating the p value cutoff. The default is set to 0.05.
`coeflimit`	A numeric indicating the lower fold cutoff. The default is set to 2.
`yrange`	A numeric for the maximum y value (scale of y-axis is -log(p-value)), can only be set to a value as big as the maximum y-value in the plots.
`negcontrol`	A vector with the names of the metabolites used as negative controls, to be coloured differently.
`poscontrol`	A vector with the names of the metabolites used as positive controls, to be coloured differently.
`xlab`	*x*-axis label.
`ylab`	*y*-axis label.
`labelunderlim`	A logical indicating whether to label points that are not significant.
`labelsig`	A logical indicating whether all significant points should be labeled.
`saveinteractiveplot`	A logical indication whether the interactive plot produced should be saved as a `.html` file.
`interactiveplotname`	A character string indicating the name to be used for saving the interactive plot.
`...`	Arguments to VolcanoPlot function

Author(s)

Alysha M De Livera, Gavriel Olshansky

See Also

`VolcanoPlot`

Examples

```
data("alldata_eg")
featuredata_eg<-alldata_eg$featuredata
dataview(featuredata_eg)
sampledata_eg<-alldata_eg$sampledata
dataview(sampledata_eg)
metabolitedata_eg<-alldata_eg$metabolitedata
dataview(metabolitedata_eg)

logdata <- LogTransform(featuredata_eg)
dataview(logdata$featuredata)
imp <-  MissingValues(logdata$featuredata,sampledata_eg,metabolitedata_eg,
                     feature.cutof=0.8, sample.cutoff=0.8, method="knn")
dataview(imp$featuredata)

#Linear model fit using unadjusted data
factormat<-model.matrix(~gender +Age +bmi, sampledata_eg)
unadjustedFit<-LinearModelFit(featuredata=imp$featuredata,
                             factormat=factormat,
                             ruv2=FALSE)
unadjustedFit

#Linear model fit using `is' normalized data
Norm_is <-NormQcmets(imp$featuredata, method = "is",
                    isvec = imp$featuredata[,which(metabolitedata_eg$IS ==1)[1]])
isFit<-LinearModelFit(featuredata=Norm_is$featuredata,
                     factormat=factormat,
                     ruv2=FALSE)
isFit

#Linear model fit with ruv-2 normalization
ruv2Fit<-LinearModelFit(featuredata=imp$featuredata,
                       factormat=factormat,
                       ruv2=TRUE,k=2,
                       qcmets = which(metabolitedata_eg$IS ==1))
ruv2Fit

#Exploring metabolites associated with age
lcoef_age<-list(unadjusted=unadjustedFit$coefficients[,"Age"],
               is_age=isFit$coefficients[,"Age"],
               ruv2_age=ruv2Fit$coefficients[,"Age"])
lpvals_age<-list(unadjusted=unadjustedFit$p.value[,"Age"],
                is=isFit$p.value[,"Age"],
                ruv2=ruv2Fit$p.value[,"Age"])

negcontrols<-metabolitedata_eg$names[which(metabolitedata_eg$IS==1)]

CompareVolcanoPlots(lcoef=lcoef_age,
                   lpvals_age,
                   xlab="Coef",
                   negcontrol=negcontrols)
```

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

See Also

makeContrasts.

Examples

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

Corr *Computes correlation matrix for a metabolomics dataset or compares the correlation between two metabolomics datasets*

Description

Computes correlation matrix for a metabolomics dataset or compares the correlation between two metabolomics datasets

Usage

```
Corr(featuredata1 = NULL, featuredata2 = NULL, method = c("pearson",
  "kendall", "spearman"), padjmethod = c("BH", "holm", "hochberg", "hommel",
  "bonferroni", "BY", "fdr", "none"), saveoutput = FALSE, outputname = NULL)
```

Arguments

`featuredata1` A data frame in the featuredata format. This is a dataframe with metabolites in columns and samples in rows. Unique sample names should be provided as row names.

`featuredata2` A data frame in the featuredata format. This is a dataframe with metabolites in columns and samples in rows. Unique sample names should be provided as row names.

`method` Must be one of "pearson", "spearman" or "kendall"

`padjmethod` p-value adjustment method. Must be one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr" or "none"

`saveoutput` A logical indicating whether the results 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 'results'. @seealso `comp.2.cc.fdr`.

Author(s)

Alysha M De Livera, Gavriel Olshansky

References

Fukushima, A. Gene (2013) 518, 209-214

Examples

```
data("featuredata_roots")
featuredata_roots[featuredata_roots==0]<-NA
imp_data<-MissingValues(LogTransform(featuredata_roots)$featuredata)$featuredata
Corr( imp_data[c(1:17),], imp_data[c(18:37),])
Corr( imp_data[c(1:17),])
```

`dataview` *Data Viewing*

Description

Prints out a trimmed version of the data matrix

Usage

```
dataview(data)
```

Arguments

`data` a data frame or a data table

Value

The trimmed data

Author(s)

Gavriel Olshanksy and Alysha M De Livera

Examples

```
data("alldata_eg")
featuredata_eg<-alldata_eg$featuredata
dataview(featuredata_eg)
```

`Dendrogram` *Dendrogram*

Description

Performs hierarchical cluster analysis given a distance measure and an agglomeration method, and produces a dendrogram.

Usage

```
Dendrogram(featuredata, groupdata, saveplot = FALSE,
  plotname = "dendrogram", savetype = c("png", "bmp", "jpeg", "tiff",
  "pdf"), distmethod = "manhattan", aggmethod = "ward.D",
  main = "Dendrogram", cex = 0.8, clust = FALSE, rect = FALSE,
  nclust = NULL, height = NULL, bordercol = 2, ...)
```

Arguments

`featuredata` A data frame in the featuredata format. This should have sample names in the first column to be read as row names and the metabolomics variables in the remaining columns.

`groupdata` A data frame or a table with optional sample names in the first column to be read as row names and group names in the following column.

`saveplot` A logical indication whether to save the dendrogram produced.

`plotname` Name of the output file if the file is to be saved.

`savetype` The required format for the plot to be saved in. Threre is a choice of `"png"`,`"bmp"`,`"jpeg"`,`"tiff"`,`"pdf"` type files.

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", #ward -> ward.D "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.
clust	A logical indicating whether the results from heirarchical clustering should be grouped
rect	A logical indicatng whether rectanges should be drawn highlighting the groups from clust above
nclust	The desired number of clusters for clust or rect.
height	The desifed height to obtain clusters for clust or rect. Either nclust or height must be supplied for clust or rect.
bordercol	If rect=TRUE, a vector with border colors for the rectangles.
...	Arguments to be passed on to other methods.

Value

A dendrogram plot and a list containing an object of class ‘hclust’ and a vector with cluster membership if clust or rect is set to TRUE.

Author(s)

Alysha M De Livera, Gavriel Olshansky

See Also

dist, hclust.

Examples

```
data(mixdata) #unadjusted data
Dendrogram(mixdata$featuredata,mixdata$sampledata[,1])
```

Didata *Direct infusion mass spectrometry metabolomics dataset*

Description

Didata (MetaboLights: MTBLS79) consists of featuredata and sampledata as described by Kirwan et al 2014

Author(s)

Alysha M De Livera, Gavriel Olshansky

References

Kirwan,J.A., Weber,R.J., Broadhurst,D.I. and Viant,M.R. (2014) Direct infusion mass spectrometry metabolomics dataset: a benchmark for data processing and quality control. Sci. Data, 1, 140012.

Haug,K., Salek,R.M., Conesa,P., Hastings,J., de Matos,P., Rijnbeek,M., Mahendraker,T., Williams,M., Neumann,S., Rocca-Serra,P. et al. (2013) MetaboLights–an open-access general-purpose repository for metabolomics studies and associated meta-data. Nucleic Acids Res., 41, D781-786.

Examples

```
data(Didata)
dataview(Didata$featuredata)
dataview(Didata$sampledata)
```

---

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.

featuredata_roots *A metabolomics dataset analyzed by GC-TOF/MS.*

Description

featuredata_roots consists of featuredata that has been described by Fukushima et al 2013.

Author(s)

Alysha M De Livera, Gavriel Olshansky

References

Fukushima, A. Gene (2013) 518, 209-214

Examples

```
data(featuredata_roots)
dataview(featuredata_roots)
```

GenerateReport *Generate Report*

Description

Generates an interactive report based on basic user input. The user can choose up to 3 normalisation methods that will be compared to the unadjusted data using various diagnostics to assess the normalisation. Guidance on choosing criteria is also provided.

Usage

```
GenerateReport(featuredata = NULL, sampledata = NULL,
  metabolitedata = NULL, normmeth = list(method1 = c("nomis"), method2 =
  c("ccmn"), method3 = c("ruv2")), factorOI = NULL, covars = NULL,
  gfactor = NULL, missingvals = c("knn", "replace", "none"),
  logTrans = TRUE, k = NULL, fitintercept = TRUE, isvec = NULL,
  qcmets = NULL, rlsc.sampledata = NULL, ccmn.factor = NULL,
  volcano.yrange = NULL, scaling.refvec = NULL,
  reportName = "General_Report", ...)
```

Arguments

`featuredata` featuredata A data frame in the featuredata format. This is a dataframe with metabolites in columns and samples in rows. Unique sample names should be provided as row names.

`sampledata` A dataframe with sample information matching featuredata.

`metabolitedata` A dataframe with metabolite information matching featuredata.

`normmeth` A list of up to 3 normalisation methods. Must be one of "is", "ccmn", "nomis", "ruv2", "ruvrand", "rlsc", "median", "mean", "sum". For combined methods, the list should consist of vector with entries corresponding in order to the 2 methods to beused jointly.

`factorOI` factor of interest to be used, should correpond to column number or column name in sampledata corresponding to the factor of interest for the analysis.

`covars` names of the other covariates to be included when fitting the linear model for biomarker identification. Should correspond to column name in sampledata

`gfactor` A vector indicating the groups that need to be explored in the plots

`missingvals` The method to be used for removing missing values. Should be either "knn" or "replace".

`logTrans` A logical indication whether the data is to be log transformed.

`k` k Number of factors of unwanted variation to be included in the "`ruv`" models.

`fitintercept` A logical indication whether an intercept component should be fitted in the linear model.

`isvec` A vector of internal standards to be used with the method "`is`".

`qcmets` A vector indicating which metabolites should be used as the internal, external standards or other quality control metabolites in the "`ruv`" models, or as multiple internal standards in the "`ccmn`" and "`nomis`" methods.

`rlsc.sampledata`
For the "`rlsc`" method, a dataframe that contains sample specific information. Unique sample names should be provided as row names. For this function, this should have, the batch number, the class and the run order, with column names 'batch', 'class' and 'order' respectively. For the QCs samples, 'class' should be allocated as 0.

`ccmn.factor` For the ccmn method. A vector describing biological factors.

`volcano.yrange` In the volcano plot, a numeric for the maximum y value (scale of y-axis is -log(p-value)), can only be set to a value as big as the maximum y-value in the plots.

`scaling.refvec` A reference vector for the scaling method

`reportName` The name that should be used to save the report.

`...` Arguments to be passed onto other methods.

Author(s)

Alysha M De Livera, Gavriel Olshansky

`HeatMap` *Heat map*

Description

Produces an interactive or a non-interactive heat map of a metabolomics data matrix optionally clustered according to specified methods

Usage

```
HeatMap(featuredata, groupdata, saveplot = FALSE, plotname = "heatmap",
  savetype = c("png", "bmp", "jpeg", "tiff", "pdf"), interactiveplot = TRUE,
  saveinteractiveplot = FALSE, return.interactive = FALSE,
  numeric.mets = FALSE, colramp = c(75, "magenta", "green"),
  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

`featuredata` A data frame in the met data format. This should have sample names in the first column to be read as row names and the metabolomics variables in the remaining columns.

`groupdata` A data frame or a table with optional sample names in the first column to be read as row names and group names in the following column.

`saveplot` A logical indication whether to save the plot produced.

`plotname` Name of the output file if the file is to be saved.

`savetype` The required format for the plot to be saved in. Threre is a choice of `"png"`,`"bmp"`,`"jpeg"`,`"tiff"`,`"pdf"` type files.

`interactiveplot` A logical indication whether an interactive plot is to be shown.

`saveinteractiveplot` A logical indication whether to save the interactive plot as an `"html"` file.

`return.interactive` A logical indication whether an interactive plot should be returned as a variable by the function.

`numeric.mets` A logical indication whether metabolite names are numeric. If `TRUE`, "m" is added before each metabolite name displyed on the graph.

`colramp` A vector containing (in order), the desired number of color elements in the panel, color to use for the lowest, color to use for the highest for the non-interactive plot.

`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 for non-interactive plots.

`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, Gavriel Olshansky

See Also

par, heatmap.2.

Examples

```
    data(mixdata)  #unadjusted data
    HeatMap(mixdata$featuredata,mixdata$sampledata[,1],
             saveplot = FALSE,
            interactiveplot = TRUE, scale = "row",
            dendrogram = "none", colramp=c(75,"magenta","green"))
```

LinearModelFit *Linear models*

Description

Fit a linear model to each metabolite in a metabolomics data matrix, and obtain the coefficients, 95 The featuredata must be log transformed, but does not have to be normalised a priori as the LinearModelFit function can be used to fit the ruv2 method to accommodate the unwanted variation in the model. Either ordinary statistics or empirical Bayes statistics can be obtained.

Usage

```
LinearModelFit(featuredata, factormat = NULL, contrastmat = NULL,
  ruv2 = TRUE, k = NULL, qcmets = NULL, moderated = FALSE,
  padjmethod = "BH", ci_alpha = 0.05, saveoutput = FALSE,
  outputname = "results", ...)
```

Arguments

`featuredata` featuredata A data frame in the featuredata format. This is a dataframe with metabolites in columns and samples in rows. Unique sample names should be provided as row names.

`factormat` A design matrix for the linear model, consisting of biological factors of interest.

`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.

`qcmets` If `ruv2=TRUE`, a vector indicating which metabolites should be used as the internal, external standards or other quality control metabolites.

`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"`.

`ci_alpha` Significance level for the confidence intervals.

`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 F statistics, t statistics, corresponding confidence intervals, and adjusted and unadjusted p-values (see 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, Gavriel Olshansky

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, Alysha M De, M. Aho-Sysi, Laurent Jacob, J. Gagnon-Bartch, Sandra Castillo, J.A. Simpson, and Terence P. Speed. 2015. Statistical Methods for Handling Unwanted Variation in Metabolomics Data. *Analytical Chemistry* 87 (7). American Chemical Society: 3606-3615.

De Livera, A.M., Olshansky, M., Speed, T. P. (2013) Statistical analysis of metabolomics data. *Methods in Molecular Biology* (Clifton, N.J.) 1055: 291-307.

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

```
data("alldata_eg")
featuredata_eg<-alldata_eg$featuredata
dataview(featuredata_eg)
sampledata_eg<-alldata_eg$sampledata
dataview(sampledata_eg)
metabolitedata_eg<-alldata_eg$metabolitedata
dataview(metabolitedata_eg)

logdata <- LogTransform(featuredata_eg)
dataview(logdata$featuredata)
imp <-  MissingValues(logdata$featuredata,sampledata_eg,metabolitedata_eg,
                     feature.cutof=0.8, sample.cutoff=0.8, method="knn")
dataview(imp$featuredata)

#Linear model fit using unadjusted data
factormat<-model.matrix(~gender +Age +bmi, sampledata_eg)
unadjustedFit<-LinearModelFit(featuredata=imp$featuredata,
                             factormat=factormat,
                             ruv2=FALSE)
unadjustedFit
```

```
#Linear model fit using `is' normalized data
Norm_is <-NormQcmets(imp$featuredata, method = "is",
                    isvec = imp$featuredata[,which(metabolitedata_eg$IS ==1)[1]])
isFit<-LinearModelFit(featuredata=Norm_is$featuredata,
                     factormat=factormat,
                     ruv2=FALSE)
isFit

#Linear model fit with ruv-2 normalization
ruv2Fit<-LinearModelFit(featuredata=imp$featuredata,
                       factormat=factormat,
                       ruv2=TRUE,k=2,
                       qcmets = which(metabolitedata_eg$IS ==1))
ruv2Fit

#Linear model fit with ruv-2 normalization, obtaining moderated statistics
ruv2FitMod<-LinearModelFit(featuredata=imp$featuredata,
                          factormat=factormat,
                          ruv2=TRUE,k=2,moderated = TRUE,
                          qcmets = which(metabolitedata_eg$IS ==1))
ruv2FitMod
```

LogTransform *Log transformation*

Description

Log transform a metabolomics feature data matrix.

Usage

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

Arguments

featuredata	featuredata A data frame in the featuredata format. This is a dataframe with metabolites in columns and samples in rows. Unique sample names should be provided as row names. See NormalizeMets Vignette 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.
zerotona	A logical indicating whether any zeros should be converted to missing prior to log transforming. By default, this is set to FALSE.

Value

The result is an object of class alldata.

Author(s)

Alysha M De Livera

Examples

```
data(alldata_eg)
lg <- LogTransform(alldata_eg$featuredata)
dataview(lg$featuredata)
```

metgroupCheck *Check data compatibility*

Description

Check if the group data matches the metabolite data in size, returns error message if not compatible

Usage

```
metgroupCheck(metdata, groupdata)
```

Arguments

metdata	A data frame in the met data format. This should have sample names in the first column to be read as row names and the variables in the remaining columns. These variables can be metabolites, masses, retention times, bins, areas or any other metabolomics variables of interest.
groupdata	A data frame or a table with optional sample names in the first column to be read as row names and group names in the following column.

Author(s)

Alysha M De Livera, Gavriel Olshansky

MissingValues *Missing value replacement*

Description

Missing value imputation for metabolomics data matrices

Usage

```
MissingValues(featuredata, sampledata = NULL, metabolitedata = NULL,
  feature.cutoff = 0.8, sample.cutoff = 0.8, method = c("knn", "replace",
  "none"), k = 10, featuremax.knn = 0.8, samplemax.knn = 0.8,
  seed = 100, saveoutput = FALSE, outputname = "nomissing")
```

Arguments

`featuredata` A data frame in the featuredata format. This is a dataframe with metabolites in columns and samples in rows. Unique sample names should be provided as row names. See NormalizeMets Vignette for details.

`sampledata` A dataframe with sample information matching featuredata.

`metabolitedata` A dataframe with metabolite information matching featuredata.

`feature.cutoff` A value between zero and one. Used to exclude features that have a large proportion of missing values. If the proportion of missing values is equal to or more than the feature.cutoff, that feature will be deleted.

`sample.cutoff` A value between zero and one. Used to exclude samples that have a large proportion of missing values. If the proportion of missing values is equal to or more than the sample.cutoff in any row, that whole sample will be deleted.

`method` Missing value replacement method. Should be either "knn" (the kth nearest neighbour algorithm), "replace" (replacing by half the minimum detectable signal), or "none".

`k` The number of nearest neighbours to be used in the knn algorithm

`featuremax.knn` For the knn algorithm. The maximum proportion of missing data allowed in any feature. For any features with more than featuremax.knn proportion missing, missing values are imputed using the overall mean per sample.

`samplemax.knn` For the knn algorithm. The maximum proportion of missing data allowed in any sample. If any sample has more than samplemax.knn missing data, the program halts and reports an error.

`seed` For the knn algorithm for very large matrices. 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 alldata.

Author(s)

Alysha M De Livera, Gavriel Olshansky

Examples

```
data(alldata_eg)
featuredata_eg<-alldata_eg$featuredata
sampledata_eg<-alldata_eg$sampledata
metabolitedata_eg<-alldata_eg$metabolitedata
logdata <- LogTransform(featuredata_eg)

imp <-  MissingValues(logdata$featuredata,sampledata_eg,metabolitedata_eg,
                  feature.cutof=0.8, sample.cutoff=0.8, method="knn")
imp
dataview(imp$featuredata)
```

mixdata   *A designed metabolomics dataset*

Description

mixdata consists of featuredata, sampledata and metabolitedata as described by redestig et al 2009. A total of 42 samples with dilution mixtures with 44 components, set at three different alternating concentrations, have been run using GC-TOF MS.

Author(s)

Alysha M De Livera, Gavriel Olshansky

References

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.

Examples

```
data(mixdata)
dataview(mixdata$featuredata)
dataview(mixdata$metabolitedata)
```

multiplot *Multiple plot function ggplot objects can be passed in ..., or to plotlist (as a list of ggplot objects)*

Description

Multiple plot function ggplot objects can be passed in ..., or to plotlist (as a list of ggplot objects)

Usage

```
multiplot(..., plotlist = NULL, file, cols = 1, layout = NULL)
```

Arguments

`...`	polts seperated by ',' If the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE), then plot 1 will go in the upper left, 2 will go in the upper right, and 3 will go all the way across the bottom.
`plotlist`	A list of plots.
`file`	A file to save to??..
`cols`	Number of columns in layout
`layout`	A matrix specifying the layout. If present, 'cols' is ignored.

Details

A function adapted from Winston Chang, 'R Graphics Cookbook'.

NormCombined *Normalisation methods based on a combination of methods*

Description

Normalise a metabolomic data matrix using a combination of methods

Usage

```
NormCombined(featuredata, methods = c("rlsc", "median"),
  savefinaloutput = FALSE, finaloutputname = NULL, ...)
```

Arguments

`featuredata` featuredata A data frame in the featuredata format. This is a dataframe with metabolites in columns and samples in rows. Unique sample names should be provided as row names.

`methods` A character vector indicating which two methods should be used in order.

`savefinaloutput`
A logical indicating whether the final normalised data matrix should be saved as a .csv file.

`finaloutputname`
The name of the final output file if the output has to be saved.

`...` Inputs for the `NormScaling`, `NormQcmets`, and `NormQcsamples` as appropriate.

Value

The result is an object of class `alldata`.

Author(s)

Alysha M De Livera, Gavriel Olshansky

References

De Livera, Alysha M De, M. Aho-Sysi, Laurent Jacob, J. Gagnon-Bartch, Sandra Castillo, J.A. Simpson, and Terence P. Speed. 2015. Statistical Methods for Handling Unwanted Variation in Metabolomics Data. *Analytical Chemistry* 87 (7). American Chemical Society: 3606-3615.

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.

Dunn,W.B., Broadhurst,D., Begley,P., Zelena,E., Francis-McIntyre,S., Anderson,N., Brown,M., Knowles,J.D., Halsall,A., Haselden,J.N. et al. (2011) Procedures for large-scale metabolic profiling of serum and plasma using gas chromatography and liquid chromatography coupled to mass spectrometry. Nat. Protoc., 6, 1060-1083

Examples

```
##Reading the data
data(Didata)
NormCombined(featuredata=Didata$featuredata[order(Didata$sampledata$order),],
          sampledata=Didata$sampledata[order(Didata$sampledata$order),],
          methods=c("rlsc", "median"),
          savefinaloutput=FALSE,
          finaloutputname=NULL)
```

`NormQcmets` *Normalisation methods based on quality control metabolites*

Description

Normalise a metabolomic data matrix using internal, external standards and other quality control metabolites

Usage

```
NormQcmets(featuredata, factors = NULL, factormat = NULL, method = c("is",
  "nomis", "ccmn", "ruv2", "ruvrand", "ruvrandclust"), isvec = NULL,
  ncomp = NULL, k = NULL, plotk = TRUE, lambda = NULL, qcmets = NULL,
  maxIter = 200, nUpdate = 100, lambdaUpdate = TRUE, p = 2,
  saveoutput = FALSE, outputname = NULL, ...)
```

Arguments

`featuredata` featuredata A data frame in the featuredata format. This is a dataframe with metabolites in columns and samples in rows. Unique sample names should be provided as row names.

`factors` For the ccmn method. A vector or a dataframe containing biological factors.

`factormat` For the ruv2 method. A design matrix for the linear model, consisting of biological factors of interest.

`method` A character string indicating the required normalization method. Must be one of "`is`", "`nomis`", "`ccmn`", "`ruv2`", "`ruvrand` " or "`ruvrandclust`". See Details for information.

`isvec` A vector of internal standards to be used with the method "`is`".

`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 "`ruv`" models.

`plotk` For the "`ruvrand`" method. A logical indicating whether a bargraph for the proportion of variance explained by the factors of unwanted variation needs to be plotted `lambda` The regularization parameter for the "`ruvrand`" method which depends on k. If not entered, it will be estimated. See DeLivera et al, 2015 for details.

`qcmets` A vector indicating which metabolites should be used as the internal, external standards or other quality control metabolites in the "`ruv`" models, or as multiple internal standards in the "`ccmn`" and "`nomis`" methods.

`maxIter` For the "`ruvrandclust`" method. Maximum number of iterations for "`ruvrandclust`" method.

`nUpdate` For the "`ruvrandclust`" method. Update the unwanted variation component every nUpdate iterations.

`lambdaUpdate` For the "`ruvrandclust`" method. A logical indicating whether the regularization parameter needs to be updated

`p` For the "`ruvrandclust`" method. The number of clusters to be used in the k-means clustering.

`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.

`...` Other arguments to be passed onto `LinearModelFit`.

Details

These normalisation methods 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), "`ruv2`" (De Livera *et al*. 2012a), and "`ruvrand`", "`ruvrandclust`" (De Livera *et al*. 2015).

An overview of these normalisation methods are given by De Livera *et al*. (2015).

Value

If the method is 'ruv2', the function will return an object of class `MArrayLM`, containing F statistics, t statistics, corresponding confidence intervals, and adjusted and unadjusted p-values. See `LinearModelFit`. For all other methods, the result is an object of class `alldata`. Additionally, the list also contains the removed unwanted variation component (UVcomp),and the results from the optimization algorithm (opt) for the "`ruvrandclust`" method @seealso `normFit`.

Author(s)

Alysha M De Livera, Gavriel Olshansky

References

De Livera, Alysha M De, M. Aho-Sysi, Laurent Jacob, J. Gagnon-Bartch, Sandra Castillo, J.A. Simpson, and Terence P. Speed. 2015. Statistical Methods for Handling Unwanted Variation in Metabolomics Data. *Analytical Chemistry* 87 (7). American Chemical Society: 3606-3615.

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.

Examples

```
## Reading the data
  data(mixdata)
  featuredata <- mixdata$featuredata
  sampledata<-mixdata$sampledata
  metabolitedata<-mixdata$metabolitedata
  isvec<-featuredata[,which(metabolitedata$type =="IS")[1]]
  factors<-sampledata$type
  qcmets<-which(metabolitedata$type =="IS")

 ## Normalise by an internal or an internal standard
 norm_is <- NormQcmets(featuredata, method = "is", isvec=isvec)
 PcaPlots(norm_is$featuredata, factors)

 ## Normalise by the NOMIS method
 norm_nomis <- NormQcmets(featuredata, method = "nomis", qcmets = qcmets)
 PcaPlots(norm_nomis$featuredata, factors)

 ## Normalise by the CCMN method
norm_ccmn <- NormQcmets(featuredata, factors, method = "ccmn", qcmets = qcmets, ncomp = 2)
 PcaPlots(norm_ccmn$featuredata, factors)

 ## Normalise using RUV-random method
 norm_ruvrand <- NormQcmets(featuredata, method = "ruvrand", qcmets = qcmets, k = 2)
 PcaPlots(norm_ruvrand$featuredata, factors)
 PcaPlots(norm_ruvrand$uvdata, sampledata$batch, main = "Unwanted batch variation")

 ## Normalise using RUV-random clustering method
 ##Not run
#norm_ruvrandclust <- NormQcmets(featuredata, method = "ruvrandclust", qcmets = qcmets, k = 2)
 #PcaPlots(norm_ruvrandclust$featuredata, factors)
#PcaPlots(norm_ruvrandclust$uvdata, sampledata$batch, main = "Unwanted batch variation")
```

NormQcsamples *Normalisation methods based on quality control samples*

Description

This function is based on the quality control sample based robust LOESS (locally estimated scatterplot smoothing) signal correction (QC-RLSC) method as described by Dunn *et al*. (2011) and impletemented in statTarget: Luan H (2017).

Usage

```
NormQcsamples(featuredata, sampledata, method = c("rlsc"), span = 0,
  deg = 2, lg = TRUE, saveoutput = FALSE,
  outputname = "qcsample_results", ...)
```

Arguments

`featuredata` featuredata A data frame in the featuredata format. This is a dataframe with metabolites in columns and samples in rows. Unique sample names should be provided as row names.

`sampledata` A dataframe that contains sample specific information. Unique sample names should be provided as row names. For this function, this should have, the batch number, the class and the run order, with column names 'batch', 'class' and 'order' respectively. For the QCs samples, 'class' should be allocated as 0.

`method` A character string indicating the required normalization method. In this case, only `"rlsc` method as described by Dunn *et al*. (2011).

`span` The smooting parameter. The default is 0.75.

`deg` The degree for the polynomial fit. Must be either 0, 1, or 2. Defaults to degree 2 for local polynomial fits.

`lg` A logical indicating whether the final normalized data matrix needs to be log transformed after rlsc method. Defaults to TRUE.

`saveoutput` A logical indicating whether the normalised data matrix and related plots should be saved.

`outputname` The name of the output file if the output has to be saved.

`...` Extra input for the `statTarget::shiftCor` function.

Value

The result is an object of class `alldata`.

Author(s)

Alysha M De Livera, Gavriel Olshansky

References

Luan H (2017). statTarget: Statistical Analysis of Metabolite Profile. R package version 1.6.0, https://github.com/13479776/statTarget.

Dunn,W.B., Broadhurst,D., Begley,P., Zelena,E., Francis-McIntyre,S., Anderson,N., Brown,M., Knowles,J.D., Halsall,A., Haselden,J.N. et al. (2011) Procedures for large-scale metabolic profiling of serum and plasma using gas chromatography and liquid chromatography coupled to mass spectrometry. Nat. Protoc., 6, 1060-1083

See Also

`statTarget::shiftCor`.

Examples

```
##Reading the data
data(Didata)
NormQcsamples(sampledata=Didata$sampledata[order(Didata$sampledata$order),],
              featuredata=Didata$featuredata[order(Didata$sampledata$order),],
              saveoutput=FALSE)
```

NormScaling *Normalisation methods based on scaling*

Description

Normalise a metabolomic data matrix according to a specified scaling method.

Usage

```
NormScaling(featuredata, method = c("median", "mean", "sum", "ref"),
  refvec = NULL, saveoutput = FALSE, outputname = NULL, ...)
```

Arguments

`featuredata`	featuredata A data frame in the featuredata format. This is a dataframe with metabolites in columns and samples in rows. Unique sample names should be provided as row names.
`method`	A character string indicating the required scaling-based normalization method. Must be one of "`median`", "`mean`", "`sum`", or "`ref`". See NormalizeMets Vignette for details.
`refvec`	A reference vector to be used with the method "`ref`".
`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.
`...`	Arguments to other functions

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.

Value

The result is an object of class alldata.

Author(s)

Alysha M De Livera, Gavriel Olshansky

References

De Livera, Alysha M De, M. Aho-Sysi, Laurent Jacob, J. Gagnon-Bartch, Sandra Castillo, J.A. Simpson, and Terence P. Speed. 2015. Statistical Methods for Handling Unwanted Variation in Metabolomics Data. Analytical Chemistry 87 (7). American Chemical Society: 3606-3615.

See Also

normFit.

Examples

```
## Reading the data
  data(mixdata)
  featuredata <- mixdata$featuredata
  sampledata<-mixdata$sampledata
  metabolitedata<-mixdata$metabolitedata
  refvec<-featuredata[,which(metabolitedata$type =="IS")[1]]

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

 ## Normalise by a reference vector, in this case an internal standard
 norm_is <- NormScaling(featuredata, method = "ref",
     refvec=refvec)

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

## Rla Plots after normalising by the median
 RlaPlots(norm_med$featuredata, group= sampledata$batch)
```

PcaPlots *PCA plots*

Description

Produces PCA plots of the metabolomics data.

Usage

```
PcaPlots(featuredata, groupdata, saveplot = FALSE,
  saveinteractiveplot = FALSE, plotname = "", savetype = c("png", "bmp",
  "jpeg", "tiff", "pdf"), interactiveonly = FALSE, interactiveplots = TRUE,
  y.axis = 1, x.axis = 2, center = TRUE, scale = TRUE,
  userinput = TRUE, returninteractive = FALSE, main = NULL,
  varplot = FALSE, multiplot = FALSE, n = 3, cols = NULL,
  cex_val = 0.7, ...)
```

Arguments

`featuredata` featuredata A data frame in the featuredata format. This is a dataframe with metabolites in columns and samples in rows. Unique sample names should be provided as row names. See NormalizeMets Vignette for details.

`groupdata` A data frame or a vector with group names.

`saveplot` A logical indication whether to save the plot produced.

`saveinteractiveplot`
A logical indication whether to save the interactive plots produced.

`plotname` Name of the output file if the file is to be saved. This is the general name for all the graphs and the specific type prefix will be added automatically.

`savetype` The required format for the plot to be saved in. Threre is a choice of `"png"`,`"bmp"`,`"jpeg"`,`"tiff"`,`"pdf"` type files.

`interactiveonly`
Alogical indicating whether to show interactive plots only.

`interactiveplots`
A logical indication whether an interactive plot is to be shown.

`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.

`userinput` A logical indication whether user input should be required to show plots progressively. Should be `FALSE` when the plots are to be passed on to other functions or saved as variables for later use.

`returninteractive`
A logical indication whether a list of the interactive plots should be returned by the function

`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.

`cex_val` A numeric indicating the size of some text elements.

`...` Arguments to be passed on to other methods.

Author(s)

Alysha M De Livera, Gavriel Olshansky

See Also

`prcomp`.

Examples

```
data(mixdata)

# produce all results
PcaPlots(mixdata$featuredata,mixdata$sampledata[,3],multiplot = TRUE,
varplot = TRUE, interactiveplots = TRUE)

# return a list of the ineractive plots only
interactive.pca <- PcaPlots(mixdata$featuredata,mixdata$sampledata[,3],
interativeonly = TRUE, interactiveplots = TRUE,
userinput = FALSE, returninteractive = TRUE)
```

`RlaPlots` *RLA plots*

Description

Produces within group and across group relative log abundance plots to visualise a metabolomics data matrix. See De Livera *et al.* 2012, 2013, and 2015 for details.

Usage

```
RlaPlots(featuredata, groupdata, minoutlier = 0.5, type = c("ag", "wg"),
  saveplot = FALSE, plotname = "RLAPlot", savetype = c("png", "bmp",
  "jpeg", "tiff", "pdf"), interactiveplot = TRUE, interactiveonly = TRUE,
  saveinteractiveplot = FALSE, interactivesavename = "interactiveRlaPlot",
  cols = NULL, cex.axis = 0.7, las = 2, keeporder = FALSE,
  ylim = NULL, oma = c(3, 3, 3, 5) + 0.1, xlabel = "Samples",
  showlegend = TRUE, ...)
```

Arguments

`featuredata` featuredata A data frame in the featuredata format. This is a dataframe with metabolites in columns and samples in rows. Unique sample names should be provided as row names. See NormalizeMets Vignette for details.

`groupdata` A data frame or a vector with group names.

`minoutlier` A number indicating which samples names to show, samples names will only be shown for samples with a median reading greater than the number entered.

`type` A character string indicating whether within group ("wg") or across group ("ag") RLA plots need to be plotted.

`saveplot` A logical indication whether to save the plot produced.

`plotname` Name of the output file if the file is to be saved.

`savetype` The required format for the plot to be saved in. Threre is a choice of "png","bmp","jpeg","tiff","pdf type files.

`interactiveplot` A boolean indicator whether to make an interactive plot.

`interactiveonly` A boolean indicating whether only an interactive plot should be returned.

`saveinteractiveplot` A boolean indicating whether to save the interactive plot as an html file.

`interactivesavename` A character string to be used as the filename for saving the interactive plot.

`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`.

`keeporder` A logical indicator whether to keep the original sample order or group them by groupdata.

`ylim` A vector containing *y*-axis limits.

`oma` A vector giving the size of the outer margins.

`xlabel` Label for the x-axis

`showlegend` A logical indicator whether to display a legend for the plot.

`...` Other graphical parameters. See `par`.

Author(s)

Alysha M De Livera, Gavriel Olshansky

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., and Speed, T.P. 2013. Statistical Analysis of Metabolomics Data. Methods in Molecular Biology (Clifton, N.J.) 1055 (Jan): 291-307.

De Livera, A. M,. Aho-Sysi M., Laurent J., Gagnon-Bartch J., Sandra, C, Simpson, J.A., and Speed, T. P. 2015. Statistical Methods for Handling Unwanted Variation in Metabolomics Data. Analytical Chemistry 87 (7). American Chemical Society: 3606-3615.

Examples

```
data(mixdata)

RlaPlots(mixdata$featuredata, mixdata$sampledata[,1],
ylim = c(-2, 2), cols = c("green","purple"),cex.axis = 0.8)
```

SvmFit *support vector machine*

Description

Classification using support vector machine (svm) algorithm

Usage

```
SvmFit(featuredata, groupdata, kernel = "radial", cost = 1, gamma = NULL,
  crossvalid = TRUE, k = 5, tune = FALSE, pred = TRUE,
  pfeaturedata = featuredata, pgroupdata = groupdata, rocplot = TRUE,
  saveoutput = FALSE, outputname = "svm", main = NULL, ...)
```

Arguments

featuredata	featuredata A data frame in the featuredata format. This is a dataframe with metabolites in columns and samples in rows. Unique sample names should be provided as row names. See NormalizeMets Vignette for details.
groupdata	A vector with group names.
kernel	The kernel used. The default is the radial basis function with type C-classification.
cost	of constraint violation, defaults to 1.
gamma	parameter used for the kernel

`crossvalid` A logical indicating whether cross-validation needs to be conducted

`k` An integer specifying the k-fold cross-validation. Default is set to 5.

`tune` A logical with the default set to FALSE. If TRUE, a grid search will be conducted to tune the hyperparameters, over parameter ranges supplied by the user.

`pred` whether the predictions should be made

`pfeaturedata` The test dataset for the predictions. The default is featuredata

`pgroupdata` The test groupdata for the predictions. The default is groupdata

`rocplot` A logical indicating whether a receiver operating characteristic curve needs to be plotted, along with the area under the curve (AUC) printed.

`saveoutput` A logical indicating whether the outputs should be saved in the format `write.svm`

`outputname` The name of the output file if the output has to be saved.

`main` Plot title.

`...` Arguments to be passed on to other methods.

Value

If tune=FALSE, an object of class "svm" `svm` containing the fitted model or if tune=TRUE, an object of class `tune`

Author(s)

Alysha M De Livera, Gavriel Olshansky

See Also

`tune`

Examples

```
 data(alldataC)
SvmFit(featuredata=alldataC$featuredataC,
       groupdata=alldataC$groupdataC,
       crossvalid=TRUE,
       k=5,
       rocplot = TRUE)
```

`ToInputdata` *Convert to Inputdata*

Description

Returns an inputdata type data frame by combining group information from groupdata with metabolites information from featuredata

Usage

```
ToInputdata(featuredata, groupdata)
```

Arguments

`featuredata`	featuredata A data frame in the featuredata format. This is a dataframe with metabolites in columns and samples in rows. Unique sample names should be provided as row names. See NormalizeMets Vignette for details.
`groupdata`	A data frame or a vector with group names.

Value

`inputdata`	A data frame in the input data format. This has sample names in the first column, group names in the second column, and the metabolite variables in the remaining columns.

Author(s)

Alysha M De Livera, Gavriel Olshansky

`UVdata` *A designed metabolomics dataset*

Description

UVdata consists of featuredata, sampledata and metabolitedata as described by De Livera et al 2015

Author(s)

Alysha M De Livera, Gavriel Olshansky

References

De Livera, Alysha M De, M. Aho-Sysi, Laurent Jacob, J. Gagnon-Bartch, Sandra Castillo, J.A. Simpson, and Terence P. Speed. 2015. Statistical Methods for Handling Unwanted Variation in Metabolomics Data. *Analytical Chemistry* 87 (7). American Chemical Society: 3606-3615.

Examples

```
data(UVdata)
dataview(UVdata$featuredata)
dataview(UVdata$metabolitedata)
dataview(UVdata$sampledata)
```

uv_ruvrandclust *The UVdata normalized by the RUV random for clustering method*

Description

uv_ruvrandclust is a list containing featuredata (normalized data) and uvdata (the removed unwanted variation component). See the NormalizeMets Vignette and De Livera et al 2015.

Author(s)

Alysha M De Livera, Gavriel Olshansky

References

De Livera, Alysha M De, M. Aho-Sysi, Laurent Jacob, J. Gagnon-Bartch, Sandra Castillo, J.A. Simpson, and Terence P. Speed. 2015. Statistical Methods for Handling Unwanted Variation in Metabolomics Data. *Analytical Chemistry* 87 (7). American Chemical Society: 3606-3615.

Examples

```
data(uv_ruvrandclust)
dataview(uv_ruvrandclust$featuredata)
dataview(uv_ruvrandclust$uvdata)
```

VennPlot *Venn Diagram*

Description

Produces a Venn diagram showing the number of common metabolites.

Usage

```
VennPlot(lnames, group.labels = c("A", "B", "C"), saveplot = FALSE,
  savetype = c("png", "bmp", "jpeg", "tiff", "pdf"), plotname = "VennPlot",
  main = "Venn Diagram", cexval = 1, asp = 1, ...)
```

Arguments

`lnames` A list of up to three vectors, e.g. metabolite names.

`group.labels` A vector of reference values to be plotted, such as an internal standard or sample weights.

`saveplot` A logical indication whether to save the plot produced.

`savetype` The required format for the plot to be saved in. Threre is a choice of `"png"`,`"bmp"`,`"jpeg"`,`"tiff"`,`"pdf` type files.

`plotname` Name of the output file if the file is to be saved. This is the general name for all the graphs and the specific type prefix will be added automatically.

`main` A title for the plot.

`cexval` The font size of the text labels.

`asp` The aspect ratio of the plot. A value of 1 produces a square plot region.

`...` Other graphical parameters. See `par`.

Author(s)

Alysha M De Livera

Examples

```
data("alldata_eg")
featuredata_eg<-alldata_eg$featuredata
dataview(featuredata_eg)
sampledata_eg<-alldata_eg$sampledata
dataview(sampledata_eg)
metabolitedata_eg<-alldata_eg$metabolitedata
dataview(metabolitedata_eg)

logdata <- LogTransform(featuredata_eg)
dataview(logdata$featuredata)
imp <-  MissingValues(logdata$featuredata,sampledata_eg,metabolitedata_eg,
                     feature.cutof=0.8, sample.cutoff=0.8, method="knn")
dataview(imp$featuredata)

#Linear model fit using unadjusted data
factormat<-model.matrix(~gender +Age +bmi, sampledata_eg)
unadjustedFit<-LinearModelFit(featuredata=imp$featuredata,
                             factormat=factormat,
                             ruv2=FALSE)
unadjustedFit

#Linear model fit using `is' normalized data
Norm_is <-NormQcmets(imp$featuredata, method = "is",
                    isvec = imp$featuredata[,which(metabolitedata_eg$IS ==1)[1]])
isFit<-LinearModelFit(featuredata=Norm_is$featuredata,
                     factormat=factormat,
                     ruv2=FALSE)
```

```
isFit

#Linear model fit with ruv-2 normalization
ruv2Fit<-LinearModelFit(featuredata=imp$featuredata,
                       factormat=factormat,
                       ruv2=TRUE,k=2,
                       qcmets = which(metabolitedata_eg$IS ==1))
ruv2Fit

lnames<- list(names(ruv2Fit$coef[,"Age"])[which(ruv2Fit$p.value[,"Age"]<0.05)],
           names(unadjustedFit$coef[,"Age"])[which(unadjustedFit$p.value[,"Age"]<0.05)],
             names(isFit$coef[,"Age"])[which(isFit$p.value[,"Age"]<0.05)])

VennPlot(lnames, group.labels=c("ruv2","unadjusted","is"))
```

VolcanoPlot *Volcano plot*

Description

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

Usage

```
VolcanoPlot(coef, pvals, cexcutoff = 0.7, cexlab = 0.5, pointsize = 0.9,
  plimit = 0.05, coeflimit = 1, negcontrol = NULL, poscontrol = NULL,
  saveplot = FALSE, plotname = "VolcanoPlot", savetype = c("png", "bmp",
  "jpeg", "tiff", "pdf"), xlab = "Coefficients", ylab = "-log(p-value)",
  labelunderlim = FALSE, labelsig = FALSE, interactiveplot = TRUE,
  saveinteractiveplot = FALSE,
  interactiveplotname = "interactiveVolcanPlot", interactiveonly = FALSE,
  main = "Volcano Plot", fclabel = "", chooselegend = NULL,
  vlines = TRUE, tolabel = NULL, ...)
```

Arguments

`coef` A vector of coefficients with metabolite names.

`pvals` A vector of corresponding p-values.

`cexcutoff` Font size of the cut-off labels.

`cexlab` Font size of the variable labels.

`pointsize` A numeric indicating the size of the points on the graph.

`plimit` A numeric indicating the p value cutoff. The default is set to 0.05.

`coeflimit` A numeric indicating the lower fold cutoff. The default is set to 2.

`negcontrol` A vector with the names of the metabolites used as negative controls, to be coloured differently.

`poscontrol` A vector with the names of the metabolites used as positive controls, to be coloured differently.

`saveplot` A logical indicator whether to save the produced plot.

`plotname` A character string indicating the name to be used for saving the plot.

`savetype` The required format for the plot to be saved in. Threre is a choice of `"png"`,`"bmp"`,`"jpeg"`,`"tiff"`,`"pdf"` type files.

`xlab` *x*-axis label.

`ylab` *y*-axis label.

`labelunderlim` A logical indicating whether to label points that are not significant.

`labelsig` A logical indicating whether all significant points should be labeled.

`interactiveplot`
A logical indication whether an interactive plot should be shown.

`saveinteractiveplot`
A logical indication whether the interactive plot produced should be saved as a `.html` file.

`interactiveplotname`
A character string indicating the name to be used for saving the interactive plot.

`interactiveonly`
A boolean whether only an interactive version of the plot is required

`main` Plot title.

`fclabel` An optional character sting to label the vertical coefficient cutoff.

`chooselegend` Defualt to `NULL`. For internal use in other functions in the package.

`vlines` A logical indicating whether to show vertical coefficient cutoff lines.

`tolabel` A list of metabolite names on the graph to be labeled

`...` Other graphical parameters. See `par`.

Author(s)

Alysha M De Livera, Gavriel Olshansky

Examples

```
 data("alldata_eg")
logdata<-LogTransform(alldata_eg$featuredata)
sampledata<-alldata_eg$sampledata
metabolitedata<-alldata_eg$metabolitedata
imp <-  MissingValues(logdata$featuredata,sampledata,metabolitedata,
                     feature.cutof=0.8, sample.cutoff=0.8, method="knn")
featuredata<-imp$featuredata
qcmets<-which(metabolitedata[,1]=="IS")
factormat<-model.matrix(~gender +Age , sampledata)

#Linear model fit with ordinary statistics with ruv2
ordFit_ruv2<-LinearModelFit(featuredata=featuredata,
                           factormat=factormat,
                           ruv2=TRUE, qcmets=qcmets,
                           k=2)
#Volcano plot
VolcanoPlot(coef=ordFit_ruv2$coefficients[,3],
           pval=ordFit_ruv2$p.value[,3],
           cexlab = 0.8,
           interactiveplot = TRUE,
           coeflimit = 0.05,
           xlab="Coef",
           negcontrol= rownames(ordFit_ruv2$coefficients)
           [which(metabolitedata[,2]==1)],
           poscontrol= c("m74", "m161"),
           interactiveonly = TRUE)
```

Index