

Package ‘DAPAR’

April 22, 2016

Type Package

Title Tools for the Differential Analysis of Proteins Abundance with R

Version 1.0.0

Date 2015-10-01

Author Samuel Wiczorek [cre,aut],
Florence Combes [aut],
Thomas Burger [aut],
Cosmin Lazar [ctb],
Alexia Dorffer [ctb]

Maintainer Samuel Wiczorek <samuel.wiczorek@cea.fr>

Description This package contains a collection of functions for the
visualisation and the statistical analysis of proteomic data.

License Artistic-2.0

VignetteBuilder knitr

Depends R (>= 3.2), MSnbase

Suggests BiocGenerics, testthat, BiocStyle, Prostar

Imports RColorBrewer,stats,preprocessCore,Cairo,png,
lattice,reshape2,gplots,pcaMethods,ggplot2,
limma,knitr,tmvtnorm,norm,impute, imputeLCMD, XLConnect

biocViews Proteomics, Normalization, Preprocessing, MassSpectrometry

NeedsCompilation no

R topics documented:

boxPlotD	2
corrMatrixD	3
createMSnset	4
densityPlotD	5
diffAna	5
diffAnaComputeFDR	6
diffAnaGetSignificant	7

diffAnaLimma	8
diffAnaSave	8
diffAnaVolcanoplot	9
diffAnaWelch	10
getIndicesConditions	11
getNumberOfEmptyLines	12
getPaletteForLabels	12
getPourcentageOfMV	13
heatmapD	14
mvFilter	14
mvHisto	15
mvImage	16
mvImputation	17
mvPerLinesHisto	17
mvPieChart	18
mvTypePlot	19
normalizedD	19
test	20
testWithoutNA	20
UPSprotx2	21
varianceDistD	21
writeMSnsetToExcel	22

Index 23

boxPlotD	<i>Builds a boxplot from an object of class MSnSet</i>
----------	--

Description

Boxplot for quantitative proteomics data

Usage

```
boxPlotD(obj, Xaxis = "Label")
```

Arguments

obj	An object of class MSnSet .
Xaxis	A vector containing the indices of columns in <code>pData()</code> to use as X-axis (Default is "Label").

Value

A boxplot

Author(s)

Florence Combes, Samuel Wieczorek

See Also[densityPlotD](#)**Examples**

```
data(UPSprotx2)
boxPlotD(UPSprotx2)
```

corrMatrixD	<i>Displays a correlation matrix of the quantitative data of the exprs() table.</i>
-------------	---

Description

Correlation matrix based on a [MSnSet](#) object

Usage

```
corrMatrixD(obj, indLegend = NULL)
```

Arguments

obj	An object of class MSnSet .
indLegend	A vector of indices in the columns in the <code>pData()</code> table chosen for the labels in the axes.

Value

A colored correlation matrix

Author(s)

Florence Combes, Samuel Wiczorek

Examples

```
data(UPSprotx2)
corrMatrixD(UPSprotx2)
```

createMSnset *Creates an object of class [MSnSet](#) from text file*

Description

Builds an object of class [MSnSet](#) from a single tabulated-like file for quantitative and meta-data and a dataframe for the samples description. It differs from the original [MSnSet](#) builder which requires three separated files tabulated-like quantitative proteomic data into a [MSnSet](#) object, including meta-data.

Usage

```
createMSnset(file, metadata = NULL, indExpData, indFData, indiceID,  
             logData = FALSE, replaceZeros = FALSE, pep_prot_data = NULL)
```

Arguments

file	The name of a tab-separated file that contains the data.
metadata	A dataframe describing the samples (in lines).
indExpData	A vector of string where each element is the name of a column in designTable that have to be integrated in the fData() table of the MSnSet object.
indFData	The name of column in file that will be the name of rows for the exprs() and fData() tables
indiceID	The indice of the column containing the ID of entities (peptides or proteins)
logData	A boolean value to indicate if the data have to be log-transformed (Default is FALSE)
replaceZeros	A boolean value to indicate if the 0 and NaN values of intensity have to be replaced by NA (Default is FALSE)
pep_prot_data	A string that indicates whether the dataset is about peptides or proteins.

Value

An instance of class [MSnSet](#).

Author(s)

Florence Combes, Samuel Wieczorek

Examples

```
exprsFile <- system.file("extdata", "prot10.txt", package="DAPAR")  
metadataFile <- system.file("extdata", "samples_prot10.txt", package="DAPAR")  
metadata = read.table(metadataFile, header=TRUE, sep="\t", as.is=TRUE)  
indExpData <- c(2:7)  
indFData <- c(8:13)  
indiceID <- 1  
createMSnset(exprsFile, metadata, indExpData, indFData, indiceID)
```

densityPlotD	<i>Builds a densityplot from an object of class MSnSet</i>
--------------	--

Description

Densityplot of quantitative proteomics data over samples.

Usage

```
densityPlotD(obj, highlightLabel = NULL, lab2Show = NULL)
```

Arguments

obj	An object of class MSnSet .
highlightLabel	The name of the Label to highlight in the density plot.
lab2Show	A vector of labels to show in densityplot.

Value

A density plot

Author(s)

Florence Combes, Samuel Wieczorek

See Also

[boxPlotD](#), [varianceDistD](#)

Examples

```
data(UPSprotx2)
densityPlotD(UPSprotx2)
```

diffAna	<i>This function performs a differential analysis on an MSnSet object (adapted from limma)</i>
---------	--

Description

Performs a differential analysis on an [MSnSet](#) object, based on [limma](#) functions.

Usage

```
diffAna(obj, design)
```

Arguments

obj An object of class `MsnSet`.
 design The design matrix as described in the limma package documentation

Value

A dataframe with the p-value and log(Fold Change) associated to each element (peptide/protein)

Author(s)

Florence Combes, Samuel Wieczorek

Examples

```
data(UPSprotx2)
design <- cbind(cond1=1, cond2 = rep(0,nrow(pData(UPSprotx2))))
rownames(design) <- rownames(pData(UPSprotx2))
indices <- getIndicesConditions(UPSprotx2, "10fmol", "5fmol")
design[indices$iCond2,2] <- 1
diffAna(UPSprotx2, design)
```

diffAnaComputeFDR	<i>Computes the FDR corresponding to the p-values of the differential analysis.</i>
-------------------	---

Description

This function returns the FDR corresponding to the p-values of the differential analysis. The FDR is computed with the function `p.adjust{stats}`, with the BH correction method (Benjamini & Hochberg (1995)).

Usage

```
diffAnaComputeFDR(data, threshold_PVal = 0, threshold_LogFC = 0)
```

Arguments

data The result of the differential analysis processed by `diffAna`
 threshold_PVal The threshold on p-value to distinguish between differential and non-differential data
 threshold_LogFC The threshold on log(Fold Change) to distinguish between differential and non-differential data

Value

The computed FDR value (floating number)

Author(s)

Alexia Dorffer

Examples

```
data(UPSprotx2)
obj <- mvImputation(UPSprotx2, "QRILC")
condition1 <- '10fmol'
condition2 <- '5fmol'
limma <- diffAnaLimma(obj, condition1, condition2)
diffAnaComputeFDR(limma)
```

diffAnaGetSignificant *Returns a MSnSet object with only proteins significant after differential analysis.*

Description

Returns a MSnSet object with only proteins significant after differential analysis.

Usage

```
diffAnaGetSignificant(obj)
```

Arguments

obj An object of class [MSnSet](#).

Value

A MSnSet

Author(s)

Alexia Dorffer

Examples

```
data(UPSprotx2)
condition1 <- "10fmol"
condition2 <- "5fmol"
resLimma <- diffAnaLimma(UPSprotx2, condition1, condition2)
obj <- diffAnaSave(UPSprotx2, resLimma, "limma", condition1, condition2)
signif <- diffAnaGetSignificant(obj)
```

diffAnaLimma *Performs differential analysis on an MSnSet object, calling the limma package functions*

Description

Method to perform differential analysis on an MSnSet object (calls the limma package function).

Usage

```
diffAnaLimma(obj, condition1, condition2)
```

Arguments

obj An object of class [MSnSet](#).
condition1 A vector that contains the names of the conditions considered as condition 1
condition2 A vector that contains the names of the conditions considered as condition 2

Value

A dataframe as returned by the limma package

Author(s)

Florence Combes, Samuel Wieczorek

Examples

```
data(UPSprotx2)  
condition1 <- '10fmol'  
condition2 <- '5fmol'  
diffAnaLimma(UPSprotx2, condition1, condition2)
```

diffAnaSave *Returns a [MSnSet](#) object with the results of the differential analysis performed with [limma](#) package.*

Description

This method returns a [MSnSet](#) object with the results of differential analysis.

Usage

```
diffAnaSave(obj, data, method = "limma", condition1, condition2,  
          threshold_pVal = 0, threshold_logFC = 0, fdr = 0)
```


Arguments

obj	An object of class <code>MSnSet</code> .
data	The result of the differential analysis processed by <code>diffAna</code>
method	The method used for differential analysis. Available choices are : "limma", "Welch"
condition1	A vector containing the names (some values of the slot "Label" of <code>pData()</code> of the first condition.
condition2	A vector containing the names (some values of the slot "Label" of <code>pData()</code> of the second condition.
threshold_pVal	A float that indicates the threshold on p-value choosen to discriminate differential proteins.
threshold_logFC	A float that indicates the threshold on log(Fold Change) to discriminatedifferential proteins.
fdr	The FDR based on the values of <code>threshold_pVal</code> and <code>threshold_logFC</code>

Value

A `MSnSet`

Author(s)

Alexia Dorffer

Examples

```
data(UPSprotx2)
condition1 <- '10fmol'
condition2 <- '5fmol'
limma <- diffAnaLimma(UPSprotx2, condition1, condition2)
obj <- diffAnaSave(UPSprotx2, limma, "limma", condition1, condition2)
```

diffAnaVolcanoplot *Volcanoplot of the differential analysis*

Description

Plots a volcano plot after the differential analysis. Typically, the log of Fold Change is represented on the X-axis and the log₁₀ of the p-value is drawn on the Y-axis. When the `threshold_pVal` and the `threshold_logFC` are set, two lines are drawn respectively on the y-axis and the X-axis to visually distinguish between differential and non differential data.

Usage

```
diffAnaVolcanoplot(logFC = NULL, pVal = NULL, threshold_pVal = 1e-60,
  threshold_logFC = 0, conditions = NULL)
```

Arguments

logFC	A vector of the log(fold change) values of the differential analysis.
pVal	A vector of the p-value values returned by the differential analysis.
threshold_pVal	A floating number which represents the p-value that separates differential and non-differential data.
threshold_logFC	A floating number which represents the log of the Fold Change that separates differential and non-differential data.
conditions	A list of the names of condition 1 and 2 used for the differential analysis.

Value

A volcanoplot

Author(s)

Florence Combes, Samuel Wieczorek

Examples

```
data(UPSprotx2)
condition1 <- "Cut3"
condition2 <- "WT"
data <- diffAnaLimma(UPSprotx2, condition1, condition2)
diffAnaVolcanoplot(data$logFC, data$P.Value)
```

diffAnaWelch	<i>Performs a differential analysis on a MSnSet object using the Welch t-test</i>
--------------	---

Description

Computes differential analysis on an [MSnSet](#) object, using the Welch t-test (`t.test{stats}`).

Usage

```
diffAnaWelch(obj, condition1, condition2)
```

Arguments

obj	An object of class MSnSet .
condition1	A vector containing the names of the conditions considered as condition 1
condition2	A vector containing the names of the conditions considered as condition 2

Value

A dataframe with two slots : P.Value (for the p-value) and logFC (the log of the Fold Change).

Author(s)

Florence Combes, Samuel Wieczorek

Examples

```
data(UPSprotx2)
condition1 <- '10fmol'
condition2 <- '5fmol'
diffAnaWelch(UPSprotx2, condition1, condition2)
```

getIndicesConditions *Gets the conditions indices*

Description

Returns a list for the two conditions where each slot is a vector of indices for the samples in the `pData()` table

Usage

```
getIndicesConditions(obj, cond1, cond2)
```

Arguments

<code>obj</code>	An object of class <code>MsnSet</code> .
<code>cond1</code>	A vector of Labels (a slot in the <code>pData()</code> table) for the condition 1.
<code>cond2</code>	A vector of Labels (a slot in the <code>pData()</code> table) for the condition 2.

Value

A list with two slots `iCond1` and `iCond2` containing respectively the indices of samples in the `pData()` table of the dataset.

Author(s)

Florence Combes, Samuel Wieczorek

Examples

```
data(UPSprotx2)
getIndicesConditions(UPSprotx2, "Cut3", "WT")
```

`getNumberOfEmptyLines` *Returns the number of empty lines in the data*

Description

Returns the number of empty lines in the quantitative data (i.e. `exprs()` table).

Usage

```
getNumberOfEmptyLines(obj)
```

Arguments

`obj` An object of class `MSnSet`.

Value

An integer

Author(s)

Samuel Wiczorek

Examples

```
data(UPSprotx2)
getNumberOfEmptyLines(UPSprotx2)
```

`getPaletteForLabels` *Palette for plots in DAPAR*

Description

Selects colors for the plots in DAPAR based on the different conditions in the dataset. The palette is derived from the brewer palette "Dark2" (see [RColorBrewer](#)).

Usage

```
getPaletteForLabels(obj)
```

Arguments

`obj` An object of class `MSnSet`.

Value

A palette designed for the data manipulated in DAPAR

Author(s)

Florence Combes, Samuel Wieczorek

Examples

```
data(UPSprotx2)
getPaletteForLabels(UPSprotx2)
```

`getPourcentageOfMV` *Percentage of missing values*

Description

Returns the percentage of missing values in the quantitative data (`exprs()` table of the dataset).

Usage

```
getPourcentageOfMV(obj)
```

Arguments

`obj` An object of class `MsnSet`.

Value

A floating number

Author(s)

Florence Combes, Samuel Wieczorek

Examples

```
data(UPSprotx2)
getPourcentageOfMV(UPSprotx2)
```

heatmapD	<i>This function is a wrapper to heatmap.2 that displays quantitative data in the <code>exprs()</code> table of an object of class <code>MSnSet</code></i>
----------	--

Description

Heatmap of the quantitative proteomic data of a `MSnSet` object

Usage

```
heatmapD(obj, distance = "euclidean", cluster = "average")
```

Arguments

<code>obj</code>	An object of class <code>MSnSet</code> .
<code>distance</code>	The distance used by the clustering algorithm to compute the dendrogram. See <code>help(heatmap.2)</code>
<code>cluster</code>	the clustering algorithm used to build the dendrogram. See <code>help(heatmap.2)</code>

Value

A heatmap

Author(s)

Florence Combes, Samuel Wiczorek

Examples

```
data(testWithoutNA)
heatmapD(testWithoutNA)
```

mvFilter	<i>Filter lines in the matrix of intensities w.r.t. some criteria</i>
----------	---

Description

Filters the lines of `exprs()` table with conditions on the number of missing values. The user chooses the minimum amount of intensities that is acceptable and the filter delete lines that do not respect this condition. The condition may be on the whole line or condition by condition.

Usage

```
mvFilter(obj, type, th, processText = NULL)
```

Arguments

obj	An object of class <code>MSnSet</code> containing quantitative data.
type	Method used to choose the lines to delete. Values are : "none", "wholeMatrix", "allCond", "atLeastOneCond"
th	An integer value of the threshold
processText	A string to be included in the <code>MSnSet</code> object for log.

Details

The different methods are : "wholeMatrix": given a threshold `th`, only the lines that contain at least `th` values are kept. "allCond": given a threshold `th`, only the lines which contain at least `th` values for each of the conditions are kept. "atLeastOneCond": given a threshold `th`, only the lines that contain at least `th` values, and for at least one condition, are kept.

Value

An instance of class `MSnSet` that have been filtered.

Author(s)

Florence Combes, Samuel Wiczorek

Examples

```
data(UPSprotx2)
mvFilter(UPSprotx2, "wholeMatrix", 2)
```

mvHisto	<i>Histogram of missing values</i>
---------	------------------------------------

Description

This method plots a histogram of missing values.

Usage

```
mvHisto(obj, indLegend = "auto", showValues = FALSE)
```

Arguments

obj	An object of class <code>MSnSet</code> .
indLegend	The indices of the column name's in <code>pData()</code> tab
showValues	A logical that indicates wether numeric values should be drawn above the bars.

Value

A histogram

Author(s)

Florence Combes, Samuel Wiczorek

Examples

```
data(UPSprotx2)
mvHisto(UPSprotx2, showValues=TRUE)
```

mvImage

Heatmap of missing values

Description

Plots a heatmap of the quantitative data. Each column represent one of the conditions in the object of class [MSnSet](#) and the color is proportional to the mean of intensity for each line of the dataset. The lines have been sorted in order to visualize easily the different number of missing values. A white square is plotted for missing values.

Usage

```
mvImage(obj)
```

Arguments

obj An object of class [MSnSet](#).

Value

A heatmap

Author(s)

Samuel Wiczorek, Thomas Burger

Examples

```
data(UPSprotx2)
mvImage(UPSprotx2)
```

mvImputation	<i>Missing values imputation</i>
--------------	----------------------------------

Description

This method is a wrapper to the `imputeLCMD` package adapted to objects of class `MsnSet`.

Usage

```
mvImputation(obj, method)
```

Arguments

obj	An object of class <code>MsnSet</code> .
method	The imputation method to be used. Choices are QRILC, KNN, BPCA and MLE.

Value

The object `obj` which has been imputed

Author(s)

Samuel Wieczorek

Examples

```
data(UPSprotx2)
mvImputation(UPSprotx2, "QRILC")
```

mvPerLinesHisto	<i>Histogram of missing values per lines</i>
-----------------	--

Description

This method plots a histogram which represents the distribution of the number of missing values (NA) per lines (ie proteins).

Usage

```
mvPerLinesHisto(obj, indLegend = "auto", showValues = TRUE)
```

Arguments

obj	An object of class <code>MsnSet</code> .
indLegend	The indice of the column name's in <code>pData()</code> tab
showValues	A logical that indicates wether numeric values should be drawn above the bars.

Value

A histogram

Author(s)

Florence Combes, Samuel Wieczorek

Examples

```
data(UPSprotx2)
mvPerLinesHisto(UPSprotx2)
```

mvPieChart

Pie chart of the missing values

Description

Draws a pie chart of the missing values in the quantitative data (i.e. `exprs()` table).

Usage

```
mvPieChart(obj)
```

Arguments

`obj` An object of class `MsnSet`.

Value

A pie chart

Author(s)

Samuel Wieczorek

Examples

```
data(UPSprotx2)
mvPieChart(UPSprotx2)
```

`mvTypePlot`*Distribution of missing values with respect to intensity values*

Description

This method plots a scatter plot which represents the distribution of missing values. The colors correspond to the different conditions (slot `Label` in in the dataset of class `MsnSet`). The x-axis represent the mean of intensity for one condition and one entity in the dataset (i. e. a protein) whereas the y-axis count the number of missing values for this entity and the considered condition. The data have been jittered for an easier vizualisation.

Usage

```
mvTypePlot(obj, threshold = 0)
```

Arguments

`obj` An object of class `MsnSet`.
`threshold` An integer for the intensity that delimits MNAR and MCAR missing values.

Value

A scatter plot

Author(s)

Florence Combes, Samuel Wieczorek

Examples

```
data(UPSprotx2)  
mvTypePlot(UPSprotx2)
```

`normalized`*Normalisation*

Description

Provides several methods to normalize quantitative data from a `MsnSet` object. They are organized in four main families : Strong Rescaling, Median Centering, Mean Centering, Mean CenteringScaling. For the first family, two sub-categories are available : the sum by columns and the quantiles method. For the three other families, two categories are available : "Overall" which means that the value for each protein (ie line in the expression data tab) is computed over all the samples ; "within conditions" which means that the value for each protein (ie line in the `exprs()` data tab) is computed condition by condition.

Usage

```
normalized(obj, family, method)
```

Arguments

obj An object of class [MSnSet](#)
family One of the following : Global Rescaling, Median Centering, Mean Centering, Mean Centering Scaling
method "Overall" or "within conditions"

Value

An instance of class [MSnSet](#) where the quantitative data in the `exprs()` tab has been normalized.

Author(s)

Florence Combes, Samuel Wiczorek

Examples

```
data(UPSprotx2)
normalized(UPSprotx2, "Median Centering", "within conditions")
```

test	<i>Test dataset</i>
------	---------------------

Description

Partial (small) dataset for unit tests containing missing values.

Format

An object of class [MSnSet](#)

testWithoutNA	<i>Test dataset</i>
---------------	---------------------

Description

Partial (small) dataset for unit tests without any missing values.

Format

An object of class [MSnSet](#)

UPSprotx2

UPSprotx2 dataset

Description

This dataset results from a controlled relative quantification proteomics experiment where the commercial Sigma mix UPS1 human proteins were spiked in a similar yeast lysate in 2 different concentrations (with a ratio of 2). As a consequence, it can be used to benchmark the quality of a statistical analysis: in the ideal case, after the differential analysis, only and all the human proteins should have thus been selected. The dataset is either available as a CSV file (see `inst/extdata/proteinGroups-UPSx2.txt`), or as a `MSnSet` structure (`data(UPSprotx2)`). In the latter case, the quantitative data are those of the raw intensities.

Format

An object of class `MSnSet` related to proteins quantification. It contains 6 samples divided into two conditions (5fmol and 10fmol) and 2394 proteins.

varianceDistD

Distribution of variance of proteins

Description

Builds a densityplot of the variance of entities in the `exprs()` table of a object. The variance is calculated for each condition (Label) present in the dataset (see the slot 'Label' in the `pData()` table)

Usage

```
varianceDistD(obj)
```

Arguments

`obj` An object of class `MSnSet`.

Value

A density plot

Author(s)

Florence Combes, Samuel Wieczorek

See Also

[densityPlotD](#).

Examples

```
data(UPSprotx2)
varianceDistD(UPSprotx2)
```

`writeMSnsetToExcel` *This function exports a `MSnSet` object to a Excel file.*

Description

This function exports a `MSnSet` data object to a Excel file. Each of the three data.frames in the `MSnSet` object (ie experimental data, phenoData and metaData are respectively integrated into separate sheets in the Excel file).

Usage

```
writeMSnsetToExcel(obj, filename, id)
```

Arguments

<code>obj</code>	An object of class <code>MSnSet</code> .
<code>filename</code>	A character string for the name of the Excel file.
<code>id</code>	An integer to select in the fdata frame which column has to be used as an index.

Value

A Excel file

Author(s)

Samuel Wieczorek

Examples

```
data(UPSprotx2)
writeMSnsetToExcel(UPSprotx2, "foo", 1)
```

Index

*Topic **data**

test, [20](#)
testWithoutNA, [20](#)
UPSProtx2, [21](#)

boxPlotD, [2, 5](#)

corrMatrixD, [3](#)
createMSnset, [4](#)

densityPlotD, [3, 5, 21](#)
diffAna, [5, 6, 9](#)
diffAnaComputeFDR, [6](#)
diffAnaGetSignificant, [7](#)
diffAnaLimma, [8](#)
diffAnaSave, [8](#)
diffAnaVolcanoplot, [9](#)
diffAnaWelch, [10](#)

getIndicesConditions, [11](#)
getNumberOfEmptyLines, [12](#)
getPaletteForLabels, [12](#)
getPourcentageOfMV, [13](#)

heatmap.2, [14](#)
heatmapD, [14](#)

limma, [5, 8](#)

MSnSet, [2–22](#)
mvFilter, [14](#)
mvHisto, [15](#)
mvImage, [16](#)
mvImputation, [17](#)
mvPerLinesHisto, [17](#)
mvPieChart, [18](#)
mvTypePlot, [19](#)

normalizeD, [19](#)

RColorBrewer, [12](#)

t.test, [10](#)
test, [20](#)
testWithoutNA, [20](#)

UPSProtx2, [21](#)

varianceDistD, [5, 21](#)

writeMSnsetToExcel, [22](#)