Package ‘msmsEDA’

April 6, 2024

Type Package
Title Exploratory Data Analysis of LC-MS/MS data by spectral counts
Version 1.40.0
Date 2014-01-19
Author Josep Gregori, Alex Sanchez, and Josep Villanueva
Maintainer Josep Gregori <josep.gregori@gmail.com>
Depends R (>= 3.0.1), MSnbase
Imports MASS, gplots, RColorBrewer
Description Exploratory data analysis to assess the quality of a set of LC-MS/MS experiments, and visualize de influence of the involved factors.
License GPL-2
Encoding latin1
biocViews ImmunoOncology, Software, MassSpectrometry, Proteomics
git_url https://git.bioconductor.org/packages/msmsEDA
git_branch RELEASE_3_18
git_last_commit 4137b23
git_last_commit_date 2023-10-24
Repository Bioconductor 3.18
Date/Publication 2024-04-05

R topics documented:

msmsEDA-package ............................................. 2
batch.neutralize ............................................. 3
count.stats ..................................................... 4
counts.hc ...................................................... 5
counts.heatmap ............................................... 6
counts.pca ..................................................... 7
disp.estimates ............................................... 8
filter.flags .................................................. 9
Description

Exploratory data analysis to assess the quality of a set of label-free LC-MS/MS experiments, quantified by spectral counts, and visualize the influence of the involved factors. Visualization tools to assess quality and to discover outliers and eventual confounding.

Details

Package:    msmsEDA
Type:       Package
Version:    1.2.0
Date:       2014-01-18
License:    GPL-2

pp.msms.data     data preprocessing
gene.table      extract gene symbols from protein description
count.stats     summaries by sample
counts.pca      principal components analysis
counts.hc       hierarchical clustering of samples
norm.counts     normalization of spectral counts matrix
counts.heatmap  experiment heatmap
disp.estimates  dispersion analysis and plots
filter.flags    flag informative features
spc.barplots    sample sizes barplots
spc.boxplots     samples SpC boxplots
spc.densityplot  samples SpC density plots
spc.scatterplot scatterplot comparing two conditions
batch.neutralize batch effects correction
batch.neutralize

**Author(s)**

Josep Gregori, Alex Sanchez and Josep Villanueva
Maintainer: Josep Gregori <josep.gregori@gmail.com>

**References**


**Description**

Computes the SpC matrix where the fixed effects of a blocking factor are substracted.

**Usage**

```r
batch.neutralize(dat, fbatch, half=TRUE, sqrt.trans=TRUE)
```

**Arguments**

- `dat` A SpC matrix with proteins in the rows and samples in the columns.
- `fbatch` A blocking factor of length equal to the number of columns in the expression matrix.
- `half` When FALSE, the contrast coefficients are of the contr.treatment style. When TRUE, the contrast coefficients are of the contr.sum style, its aim is to distribute equally the effect to each batch level, instead of having untouched reference levels.
- `sqrt.trans` When TRUE the fit is done on the square root transformed SpC matrix.

**Details**

A model with intercept and the blocking factor is fitted. The batch effects corrected SpC matrix is computed by substracting the estimated effect of the given blocking factor. When there is no clear reference batch level, the default option `half=TRUE` should be preferred. The square root transformation is known to stabilize the variance of Poisson distributed counts (with variance equal to the mean). The linear model fitting gives more accurate errors and p-values on the square root transformed SpC matrix. Nevertheless with exploratory data analysis purposes, both the raw and square root transformed SpC matrix may give good results.

**Value**

The batch effects corrected SpC matrix.
count.stats

Summary of statistics of spectral counts by sample in the dataset

Description

Computes the number of proteins identified, the total spectral counts, and a summary of each sample

Usage

count.stats(msnset)

Arguments

msnset A MSnSet with spectral counts in the expression matrix.

Value

A data frame with one row by sample and with variables:

proteins Number of identified proteins in sample
counts Total spectral counts in sample
min Min spectral counts
counts.hc

Tukey’s lower hinge spectral counts

Median spectral counts

Tukey’s upper hinge spectral counts

Max spectral counts

Author(s)

Josep Gregori

See Also

MSnSet, fivenum

Examples

data(msms.dataset)
msnset <- pp.msms.data(msms.dataset)
res <- count.stats(msnset)
res

counts.hc Hierarchical clustering on an spectral counts matrix.

Description

Hierarchical clustering of samples in an spectral counts matrix, coloring tree branches according to factor levels.

Usage

counts.hc(msnset, do.plot=TRUE, facs=NULL, wait=TRUE)

Arguments

msnset A MSnSet with spectral counts in the expression matrix.
do.plot A logical indicating whether to plot the dendrograms.
facs NULL, or a data frame with factors. See details below.
wait This function may draw different plots, one by given factor in facs. When in interactive mode the default is to wait for confirmation before proceeding to the next plot. When wait is FALSE and R in interactive mode, instructs not to wait for confirmation.

Details

The hierarchical clustering is done by means of hclust with default parameters. If do.plot is TRUE, a dendrogram is plotted for each factor, with branches colored as per factor level. If facs is NULL then the factors are taken from pData(msnset).
Value
Invisibly returns the value obtained from hclust.

Author(s)
Josep Gregori

See Also
MSnSet, hclust

Examples
```r
data(msms.dataset)
msnset <- pp.msms.data(msms.dataset)
hc <- counts.hc(msnset)
str(hc)
```

counts.heatmap

Heatmap of an spectral counts matrix.

Description
Heatmap showing the clustering of proteins and samples in a matrix of spectral counts

Usage
```
counts.heatmap(msnset, etit=NULL, fac=NULL, to.pdf=FALSE)
```

Arguments
- **msnset**: A MSnSet with spectral counts in the expression matrix.
- **etit**: The root name of the pdf file names where the heatmaps are sent.
- **fac**: A factor which is used for the column color bar.
- **to.pdf**: A logical indicating whether the heatmaps are sent to a pdf file.

Details
A heatmap of the msnset expression matrix is plot. If to.pdf is TRUE two heatmaps are plot, the first is fitted on an A4 page, the second is plotted with 3mm by row, allocating enough height to make the rownames readable. If fac is not NULL then a column color bar will show the levels of the factor. If to.pdf is TRUE the heatmaps are sent to pdf files whose names are the concatenation of etit and "-HeatMap.pdf" and "-FullHeatMap.pdf", otherwise etit has no effect.

Value
No value is returned
counts.pca

Author(s)
Josep Gregori

See Also
MSnSet, heatmap and heatmap.2

Examples

```r
data(msms.dataset)
msnset <- pp.msms.data(msms.dataset)
counts.heatmap(msnset, fac = pData(msnset)$treat)
```

counts.pca  Principal components analysis of an spectral counts matrix.

Description
A summary and different plots are given as a result of principal components analysis of an spectral counts matrix.

Usage
```r
counts.pca(msnset, facs = NULL, do.plot = TRUE, snms = NULL, wait = TRUE)
```

Arguments
- `msnset`: A MSnSet with spectral counts in the expression matrix.
- `do.plot`: A logical indicating whether to plot the PCA PC1/PC2 map.
- `facs`: NULL or a data frame with factors. See details below.
- `snms`: Character vector with sample short names to be plotted. If NULL then 'Xnn' is plotted where 'nn' is the column number in the dataset.
- `wait`: This function may draw different plots, one by given factor in facs. When in interactive mode the default is to wait for confirmation before proceeding to the next plot. When wait is FALSE and R in interactive mode, instructs not to wait for confirmation.

Details
The spectral counts matrix is decomposed by means of prcomp. If do.plot is TRUE, a plot is generated for each factor showing the PC1/PC2 samples map, with samples colored as per factor level. If facs is NULL then the factors are taken from pData(msnset).
disp.estimates

Value
Invisibly returns a list with values:

- pca: The return value obtained from prcomp.
- pc.vars: The percentage of variability corresponding to each principal component.

Author(s)
Josep Gregori

See Also
MSnSet, prcomp

Examples

data(msms.dataset)
msnset <- pp.msms.data(msms.dataset)
lst <- counts.pca(msnset)
str(lst)
print(lst$pc.vars[,1:4])

disp.estimates Residual dispersion estimates

Description
Estimates the residual dispersion of each row of a spectral counts matrix as the ratio residual variance to mean of mean values by level, for each factor in facs. Different plots are drawn to help in the interpretation of the results.

Usage
disp.estimates(msnset, facs=NULL, do.plot=TRUE, etit=NULL, to.pdf=FALSE, wait=TRUE)

Arguments
- msnset: A MSnSet with spectral counts in the expression matrix.
- facs: A factor or a data frame with factors.
- do.plot: A logical indicating whether to produce dispersion distribution plots.
- etit: Root name of the pdf file where to send the plots.
- to.pdf: A logical indicating whether a pdf file should be produced.
- wait: This function draws different plots, two by given factor in facs. When in interactive mode and to.pdf FALSE, the default is to wait for confirmation before proceeding to the next plot. When wait is FALSE and R in interactive mode and to.pdf FALSE, instructs not to wait for confirmation.
Details

Estimates the residual dispersion of each protein in the spectral counts matrix, for each factor in facs, and returns the quantiles at \( c(0.25, 0.5, 0.75, 0.9, 0.95, 0.99, 1) \) of the distribution of dispersion values for each factor. If facs is NULL the factors are taken from pData(msnset). If do.plot is TRUE this function produces a density plot of dispersion values, and the scatterplot of residual variance vs mean values, in log10 scale. If do.pdf is TRUE etit provides the root name for the pdf file name, ending with "-DispPlots.pdf". If etit is NULL a default value of "MSMS" is provided. A different set of plots is produced for each factor in facs.

Value

Invisibly returns a matrix with the quantiles at \( c(0.25, 0.5, 0.75, 0.9, 0.95, 0.99, 1) \) of the residual dispersion estimates. Each row has the residual dispersion values attributable to each factor in facs.

Author(s)

Josep Gregori

Examples

data(msms.dataset)
msnset <- pp.msms.data(msms.dataset)
disp.q <- disp.estimates(msnset)
disp.q

filter.flags 

Flag proteins with a minimum signal and/or sufficient dispersion.

Description

In general the spectral counts (SpC) matrix of a LC-MS/MS experiment is a sparse matrix, where most of the features have very low signal. Besides, the features with low variance to mean ratio (dispersion) will be scarcely informative in a biomarker discovery experiment. Given a minimum number of spectral counts and/or a fraction of the features to be excluded by low dispersion, this function returns a vector of logicals flagging all features with values above the given thresholds.

Usage

filter.flags(data, minSpC=2, frac.out=0.4)

Arguments

data A SpC matrix with proteins in the rows and samples in the columns.
minSpC All features with SpC below this threshold will be flagged as FALSE.
frac.out The fraction of features to be excluded, with the lowest observed dispersion. These will be flagged as FALSE.
Details
The less informative features in a SpC matrix are flagged as FALSE. Those with high enough signal and dispersion are flagged as TRUE. This vector of logicals may be used to filter the SpC matrix which is used in plots where only the relevant information matters, and where the high number of 0 may distort the plot and difficult its interpretation.

Value
A vector of logical values.

Author(s)
Josep Gregori

Examples
```r
data(msms.dataset)
fraction <- 0.3
msnset <- pp.msms.data(msms.dataset)
flags <- filter.flags(exprs(msnset), minSpC=2, frac.out=fraction)
cat("\nNumber of informative features:", sum(flags), "\n")
```

---

**gene.table**

*Gene symbols associated to protein accessions*

Description
Given a character vector with protein accessions, and a character vector with protein descriptions including gene symbols, returns a character vector with gene symbols whose names are the protein accessions. A character pattern should also be given to match the gene symbols.

Usage
```r
gene.table(Accession, Protein, patt = "GN=[A-Z0-9_]\", off = 3)
```

Arguments
- **Accession**: A character vector with protein accessions
- **Protein**: A character vector of protein descriptions including gene name symbols.
- **patt**: A character pattern to match the gene symbol within the protein description.
- **off**: Offset from the first character in the pattern corresponding to the gene symbol.

Details
NA is inserted where no match is found
**Value**

A character vector with gene symbols, whose names are the corresponding protein accessions.

**Author(s)**

Josep Gregori

**Examples**

```r
data(pnms)
head(pnms)
gene.smb <- gene.table(pnms$Accession,pnms$Proteins)
head(gene.smb)
```

---

**msms.dataset**  
**LC-MS/MS dataset**

**Description**

A MSnSet with a spectral counts matrix as expression and two factors in the phenoData.  
The spectral counts matrix has samples in the columns, and proteins in the rows.  
The factors give the treatment and batch conditions of each sample in the dataset.

**Usage**

```r
data(msms.dataset)
```

**Format**

A MSnSet

**References**

Josep Gregori, Laura Villarreal, Olga Mendez, Alex Sanchez, Jose Baselga, Josep Villanueva,  
"Batch effects correction improves the sensitivity of significance tests in spectral counting-based  
Epub 2012 May 12.

Laurent Gatto and Kathryn S. Lilley, MSnbase - an R/Bioconductor package for isobaric tagged  
mass spectrometry data visualization, processing and quantitation, Bioinformatics 28(2), 288-289 (2012).

**See Also**

See **MSnSet** for detail on the class, and the **exprs** and **pData** accessors.
Examples

```r
data(msms.dataset)
msms.dataset
dim(msms.dataset)
head(exprs(msms.dataset))
head(pData(msms.dataset))
table(pData(msms.dataset)$treat)
table(pData(msms.dataset)$batch)
table(pData(msms.dataset)$treat, pData(msms.dataset)$batch)
```

---

### norm.counts

*Spectral counts matrix normalization*

#### Description

An spectral counts matrix is normalized by means of a set of samples divisors.

#### Usage

```r
norm.counts(msnset, div)
```

#### Arguments

- `msnset` A MSnSet with spectral counts in the expression matrix.
- `div` A vector of divisors by sample

#### Details

Each column in the data matrix is divided by the corresponding divisor to obtain the normalized matrix.

#### Value

A MSnSet object with the normalized spectral counts.

#### Author(s)

Josep Gregori

#### See Also

The MSnSet class documentation and normalize
Examples

```r
data(msms.dataset)
msnset <- pp.msms.data(msms.dataset)
(tspc <- apply(exprs(msnset),2,sum))
div <- tspc/median(tspc)
e.norm <- norm.counts(msnset, div)
apply(exprs(e.norm),2,sum)
e.norm
```

pnms

Accessions and gene symbols

Description

A data frame with accessions in one column, and protein description including gene symbols in the second column.

Usage

data(pnms)

Format

A data frame with 1160 observations on the following 2 variables.

Accession  a character vector with the protein accessions
Proteins  a character vector with a description of each protein, including the gene symbol

Examples

```r
data(pnms)
str(pnms)
head(pnms)
```

pp.msms.data

Spectral counts matrix pre-processing

Description

Given a MSnSet, possibly subsetted from a bigger dataset, removes the all zero rows, and those with row names (accessions) ending with `-R` in the corresponding expression matrix. NAs are replaced by zeroes, as usually a NA in a spectral counts matrix corresponds to a protein not identified in a sample.

Usage

```r
pp.msms.data(msnset)
```
spc.barplots

Arguments
msnset A MSnSet with spectral counts in the expression matrix.

Details
An `-R` protein corresponds to an artefactual identification.
Rows with all zeros are uninformative and may give rise to errors in the analysis.
A NA is understood as a unidentified protein in a sample.

Value
Returns an updated MSnSet object.
Its processingData slot shows that the object has been processed by pp.msms.data

Author(s)
Josep Gregori

See Also
MSnSet

Examples
data(msms.dataset)
dim(msms.dataset)
msnset <- pp.msms.data(msms.dataset)
dim(msnset)

spc.barplots Set of SpC barplots by sample

Description
Draws bars of height proportional to the sample size of each column in a SpC matrix. The sizes are scaled to the median of the total SpC by sample.

Usage
spc.barplots(msms.counts,fact=NULL,...)

Arguments
msms.counts A SpC matrix with proteins in the rows and samples in the columns.
fact NULL or a factor of length equal to the number of columns in the expression matrix. If provided the bars are colored by factor level.
... Extra parameters passed to the plot function.
Details

.

Author(s)

Josep Gregori

Examples

data(msms.dataset)
spc.boxplots(exprs(msms.dataset),fact=pData(msms.dataset)[,1],
           main="UPS1 200fm vs 600fm")

spc.boxplots
Set of SpC boxplots by sample

Description

Draws a boxplot for each column (sample) in a SpC matrix. The SpC are previously transformed by log2, with an offset of 0.1. If a factor is provided the boxplots are colored by factor level to better visualize the differences.

Usage

spc.boxplots(msms.counts,fact=NULL,minSpC=2,...)

Arguments

msms.counts A SpC matrix with proteins in the rows and samples in the columns.
minSpC All matrix cells with values below this threshold are excluded.
fact NULL or a factor of length equal to the number of columns in the expression matrix. If provided the boxplots are colored by factor level.
... Extra parameters passed to the plot function.

Details

More informative plots are obtained when excluding the cells with values below 2, the default for minSpC.

Author(s)

Josep Gregori

Examples

data(msms.dataset)
spc.boxplots(exprs(msms.dataset),fact=pData(msms.dataset)[,1],
           main="UPS1 200fm vs 600fm")
spc.densityplots  SpC density plots of a SpC matrix

Description

Draws superposed density plots, one for each column (sample) in a SpC matrix. The SpC are previously transformed by log2, with an offset of 0.1. If a factor is provided the density curves are colored by factor level to better visualize the differences.

Usage

spc.densityplots(msms.counts,fact=NULL,minSpC=2,...)

Arguments

- **msms.counts**: A SpC matrix with proteins in the rows and samples in the columns.
- **minSpC**: All matrix cells with values below this threshold are excluded.
- **fact**: NULL or a factor of length equal to the number of columns in the expression matrix. If provided the density curves are colored by factor level.
- **...**: Extra parameters passed to the plot function.

Details

More informative plots are obtained when excluding the cells with values below 2, the default for minSpC.

Author(s)

Josep Gregori

Examples

data(msms.dataset)
spc.densityplots(exprs(msms.dataset),fact=pData(msms.dataset)[,1],
    main="UPS1 200fm vs 600fm")
**spc.scatterplot**

**Scatterplot of SpC means comparing two conditions**

**Description**

Given a SpC matrix and a two levels factor, draws a scatterplot with SpC means of one condition in the x axis and SpC means of the second condition in the y axis.

**Usage**

```
spc.scatterplot(msms.counts, treat, trans="log2", minSpC=2, minLFC=1, ...)
```

**Arguments**

- `msms.counts`: A SpC matrix with proteins in the rows and samples in the columns.
- `treat`: A two level factor of length equal to the number of columns in the expression matrix. The two levels represent the conditions to be compared.
- `trans`: The transformation made on the means before plotting. One among "log2", "sqrt", or "none". The default is "log2".
- `minSpC`: Used as signal threshold.
- `minLFC`: Used as size effect threshold.
- `...`: Extra parameters passed to the plot function.

**Details**

The transformed means are plotted, one condition versus the other. The borders representing absolute log fold change 1 are drawn as dashed lines. All features with log fold change equal to or greater than \( \text{minLFC} \) and with mean SpC in the most abundant condition equal to or greater than \( \text{minSpC} \) are colored in red.

**Author(s)**

Josep Gregori

**Examples**

```r
data(msms.dataset)
spc.scatterplot(exprs(msms.dataset), treat=pData(msms.dataset)[,1], trans="log2", 
    minSpC=2, minLFC=1, main="UPS1 200fm vs 600fm")
```
## Index

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>array</td>
<td>pp.msms.data, 13</td>
</tr>
<tr>
<td>cluster</td>
<td>msmsEDA-package, 2</td>
</tr>
<tr>
<td>datasets</td>
<td>msms.dataset, 11, pnms, 13</td>
</tr>
<tr>
<td>distribution</td>
<td>disp.estimates, 8</td>
</tr>
<tr>
<td>hplot</td>
<td>counts.hc, 5, counts.heatmap, 6, counts.pca, 7, disp.estimates, 8, msmsEDA-package, 2</td>
</tr>
<tr>
<td>manip</td>
<td>batch.neutralize, 3, filter.flags, 9, gene.table, 10, norm.counts, 12, pp.msms.data, 13</td>
</tr>
<tr>
<td>multivariate</td>
<td>counts.hc, 5, counts.heatmap, 6, counts.pca, 7, msmsEDA-package, 2</td>
</tr>
<tr>
<td>package</td>
<td>msmsEDA-package, 2</td>
</tr>
<tr>
<td>plots</td>
<td>spc.barplots, 14, spc.boxplots, 15, spc.densityplots, 16, spc.scatterplot, 17</td>
</tr>
<tr>
<td>univar</td>
<td>count.stats, 4</td>
</tr>
</tbody>
</table>

batch.neutralize, 3

count.stats, 4, counts.hc, 5, counts.heatmap, 6, counts.pca, 7, disp.estimates, 8, filter.flags, 9, fivenum, 5, gene.table, 10, hclust, 6, heatmap, 7, heatmap.2, 7, msms.dataset, 11, msmsEDA (msmsEDA-package), 2, msmsEDA-package, 2, MSnSet, 4–8, 11, 12, 14, norm.counts, 12, normalize, 4, 12, pnms, 13, pp.msms.data, 13, prcomp, 8, spc.barplots, 14, spc.boxplots, 15, spc.densityplots, 16, spc.scatterplot, 17