Package 'MSstats'

April 16, 2019

Title Protein Significance Analysis in DDA, SRM and DIA for Label-free or Label-based Proteomics Experiments

Version 3.14.1

Date 2018-07-30

Description A set of tools for statistical relative protein significance analysis in DDA, SRM and DIA experiments.

Maintainer Meena Choi <mnchoi67@gmail.com>

License Artistic-2.0

Depends R (>= 3.4)

Imports lme4, marray, limma, gplots, ggplot2, methods, grid, ggrepel, preprocessCore, reshape2, survival, minpack.lm, utils, grDevices, graphics, stats, doSNOW, snow, foreach, data.table, MASS, dplyr, tidyr, stringr, randomForest

Suggests BiocStyle, knitr, rmarkdown, MSstatsBioData

VignetteBuilder knitr

biocViews ImmunoOncology, MassSpectrometry, Proteomics, Software, Normalization, QualityControl, TimeCourse

LazyData true

URL http://msstats.org

BugReports https://groups.google.com/forum/#!forum/msstats

RoxygenNote 6.0.1

NeedsCompilation no

Author Meena Choi [aut, cre],
Cyril Galitzine [aut],
Ting Huang [aut],
Tsung-Heng Tsai [aut],
Olga Vitek [aut]

git_url https://git.bioconductor.org/packages/MSstats

git_branch RELEASE_3_8

git_last_commit 3dfe3d7

git_last_commit_date 2019-01-04

Date/Publication 2019-04-15

R topics documented:

`MSstats-package` *Tools for protein significance analysis in DDA,SRM and DIA proteomic experiments for label-free workflows or workflows with stable isotope labeled reference*

Description

A set of tools for protein significance analysis in SRM, DDA and DIA experiments.

Details

Package:	MSstats
License:	Artistic-2.0
LazyLoad:	yes

The package includes four main sections: I. explanatory data analysis (data pre-processing and quality control of MS runs), II. model-based analysis (finding differentially abundant proteins), III.

statistical design of future experiments (sample size calculations), and IV. protein quantification (estimation of protein abundance). Section I contains functions for (1) data pre-processing and quality control of MS runs (see dataProcess) and (2) visualizing for explanatory data analysis (see dataProcessPlots). Section II contains functions for (1) finding differentially abundant proteins (see groupComparison) and (2) visualizing for the testing results (see groupComparisonPlots) and for checking normality assumption (see modelBasedQCPlots). Section III contains functions for (1) calculating sample size (see designSampleSize) and (2) visualizing for the sample size calculations (see designSampleSizePlots). Section IV contains functions for (1) per-protein group quantification and patient quantification (see quantification)

Examples of data in MSstats are (1) example of required input data format from label-based SRM experiment SRMRawData; (2) example of required input data format from DDA experiment DDARawData; (3) example of required input data format from label-free SWATH experiment DIARawData.

The functions for converting the output from spectral processing tools, (1) Skyline, SkylinetoMSstatsFormat, (2) MaxQuant, MaxQtoMSstatsFormat, (3) Progenesis, ProgenesistoMSstatsFormat, (4) Spectronaut, SpectronauttoMSstatsFormat, (5) Proteome discovere, PDtoMSstatsFormat, (6) OpenMS, OpenMStoMSstatsFormat, (7) OpenSWATH, OpenSWATHtoMSstatsFormat, and (8) DIAUmpire, DIAUmpiretoMSstatsFormatare available.

Author(s)

Meena Choi, Cyril Galitzine, Tsung-Heng Tsai, Olga Vitek.

Maintainer: Meena Choi (<mnchoi67@gmail.com>)

References

Meena Choi, Ching-Yun Chang, Timothy Clough, Daniel Broudy, Trevor Killeen, Brendan MacLean and Olga Vitek. "MSstats: an R package for statistical analysis of quantitative mass spectrometry-based proteomic experiments" *Bioinformatics*, 30(17):2524-2526, 2014.

Ching-Yun Chang, Paola Picotti, Ruth Huttenhain, Viola Heinzelmann-Schwarz, Marko Jovanovic, Ruedi Aebersold, Olga Vitek. "Protein significance analysis in selected reaction monitoring (SRM) measurements." *Molecular & Cellular Proteomics*, 11:M111.014662, 2012.

Timothy Clough, Safia Thaminy, Susanne Ragg, Ruedi Aebersold, Olga Vitek. "Statistical protein quantification and significance analysis in label-free LC-M experiments with complex designs" *BMC Bioinformatics*, 13:S16, 2012.

`dataProcess` *Data pre-processing and quality control of MS runs of raw data*

Description

Data pre-processing and quality control of MS runs of the original raw data into quantitative data for model fitting and group comparison. Log transformation is automatically applied and additional variables are created in columns for model fitting and group comparison process. Three options of data pre-processing and quality control of MS runs in dataProcess are (1) Transformation: logarithm transformation with base 2 or 10; (2) Normalization: to remove systematic bias between MS runs.

Usage

```
dataProcess(raw,
            logTrans=2,
            normalization="equalizeMedians",
            nameStandards=NULL,
            address="",
            fillIncompleteRows=TRUE,
            featureSubset="all",
            remove_noninformative_feature_outlier=FALSE,
            n_top_feature=3,
            summaryMethod="TMP",
            equalFeatureVar=TRUE,
            censoredInt="NA",
            cutoffCensored="minFeature",
            MBimpute=TRUE,
            remove50missing=FALSE,
            maxQuantileforCensored=0.999,
            clusters=NULL)
```

Arguments

raw	name of the raw (input) data set.
logTrans	logarithm transformation with base 2(default) or 10.
normalization	normalization to remove systematic bias between MS runs. There are three different normalizations supported. 'equalizeMedians'(default) represents constant normalization (equalizing the medians) based on reference signals is performed. 'quantile' represents quantile normalization based on reference signals is performed. 'globalStandards' represents normalization with global standards proteins. FALSE represents no normalization is performed.
nameStandards	vector of global standard peptide names. only for normalization with global standard peptides.
fillIncompleteRows	If the input dataset has incomplete rows, TRUE(default) adds the rows with intensity value=NA for missing peaks. FALSE reports error message with list of features which have incomplete rows.
featureSubset	"all"(default) uses all features that the data set has. "top3" uses top 3 features which have highest average of log2(intensity) across runs. "topN" uses top N features which has highest average of log2(intensity) across runs. It needs the input for n_top_feature option. "highQuality" is under development. Currently it will use top 3 features.
remove_noninformative_feature_outlier	It only works with featureSubset="highQuality". TRUE allows to remove 1) the features with column : feature_quality="Noninformative" which are featurew with bad quality, 2) outliers that are flagged in the column, is_outlier=TRUE. FALSE (default) does not allow to remove the proteins, in which all features are interfered. In this case, the proteins, which will completely loss all features by the algorithm, will keep the most abundant peptide.
n_top_feature	The number of top features for featureSubset='topN'. Default is 3, which means to use top 3 features.

`summaryMethod` "TMP"(default) means Tukey's median polish, which is robust estimation method. "linear" uses linear mixed model.

`equalFeatureVar`
only for summaryMethod="linear". default is TRUE. Logical variable for whether the model should account for heterogeneous variation among intensities from different features. Default is TRUE, which assume equal variance among intensities from features. FALSE means that we cannot assume equal variance among intensities from features, then we will account for heterogeneous variation from different features.

`censoredInt` Missing values are censored or at random. 'NA' (default) assumes that all 'NA's in 'Intensity' column are censored. '0' uses zero intensities as censored intensity. In this case, NA intensities are missing at random. The output from Skyline should use '0'. Null assumes that all NA intensites are randomly missing.

`cutoffCensored` Cutoff value for censoring. only with censoredInt='NA' or '0'. Default is 'minFeature', which uses minimum value for each feature.'minFeatureNRun' uses the smallest between minimum value of corresponding feature and minimum value of corresponding run. 'minRun' uses minumum value for each run.

`MBimpute` only for summaryMethod="TMP" and censoredInt='NA' or '0'. TRUE (default) imputes 'NA' or '0' (depending on censoredInt option) by Accelated failure model. FALSE uses the values assigned by cutoffCensored.

`remove50missing`
only for summaryMethod="TMP". TRUE removes the runs which have more than 50% missing values. FALSE is default.

`address` the name of folder that will store the results. Default folder is the current working directory. The other assigned folder has to be existed under the current working directory. An output csv file is automatically created with the default name of "BetweenRunInterferenceFile.csv". The command address can help to specify where to store the file as well as how to modify the beginning of the file name.

`maxQuantileforCensored`
Maximum quantile for deciding censored missing values. default is 0.999

`clusters` a user specified number of clusters. default is NULL, which does not use cluster.

Details

- raw : See `SRMRawData` for the required data structure of raw (input) data.
- logTrans : if logTrans=2, the measurement of Variable ABUNDANCE is log-transformed with base 2. Same apply to logTrans=10.
- normalization : if normalization=TRUE and logTrans=2, the measurement of Variable ABUNDANCE is log-transformed with base 2 and normalized. Same as for logTrans=10.
- featureSubset : After the data was normalized, we deeply looked at each single feature (which is a precursor in DDA, a fragment in DIA, and a transition in SRM) and quantify its unexplainable variation. Ultimately, we remove the features with interference.
- equalFeatureVar : If the unequal variation of error for different peptide features is detected, then a possible solution is to account for the unequal error variation by means of a procedure called iteratively re-weighted least squares. equalFeatureVar=FALSE performs an iterative fitting procedure, in which features are weighted inversely proportionaly to the variation in their intensities, so that feature with large variation are given less importance in the estimation of parameters in the model.

Value

A list of data.frame *ProcessedData* is the data.frame of reformatted input of dataProcess including extra columns, such as log2-transformed and normalized intensities (abundance column); *RunlevelData* is the the data.frame for run-level summarized data.

Warning

When a transition is missing completely in a condition or a MS run, a warning message is sent to the console notifying the user of the missing transitions.

The types of experiment that MSstats can analyze are LC-MS, SRM, DIA(SWATH) with label-free or labeled synthetic peptides. MSstats does not support for metabolic labeling or iTRAQ experiments.

Author(s)

Meena Choi, Olga Vitek.

Maintainer: Meena Choi (<mnchoi67@gmail.com>)

References

Meena Choi, Ching-Yun Chang, Timothy Clough, Daniel Broudy, Trevor Killeen, Brendan MacLean and Olga Vitek. "MSstats: an R package for statistical analysis of quantitative mass spectrometry-based proteomic experiments" *Bioinformatics*, 30(17):2524-2526, 2014.

Ching-Yun Chang, Paola Picotti, Ruth Huttenhain, Viola Heinzelmann-Schwarz, Marko Jovanovic, Ruedi Aebersold, Olga Vitek. "Protein significance analysis in selected reaction monitoring (SRM) measurements" *Molecular & Cellular Proteomics*, 11:M111.014662, 2012.

Timothy Clough, Safia Thaminy, Susanne Ragg, Ruedi Aebersold, Olga Vitek. "Statistical protein quantification and significance analysis in label-free LC-M experiments with complex designs" *BMC Bioinformatics*, 13:S16, 2012.

Examples

```
# Consider a raw data (i.e. SRMRawData) for a label-based SRM experiment from a yeast study
# with ten time points (T1-T10) of interests and three biological replicates.
# It is a time course experiment. The goal is to detect protein abundance changes
# across time points.

head(SRMRawData)

# Log2 transformation and normalization are applied (default)
QuantData<-dataProcess(SRMRawData)
head(QuantData$ProcessedData)

# Log10 transformation and normalization are applied
QuantData1<-dataProcess(SRMRawData, logTrans=10)
head(QuantData1$ProcessedData)

# Log2 transformation and no normalization are applied
QuantData2<-dataProcess(SRMRawData,normalization=FALSE)
head(QuantData2$ProcessedData)
```

dataProcessPlots *Visualization for explanatory data analysis*

Description

To illustrate the quantitative data after data-preprocessing and quality control of MS runs, dataProcessPlots takes the quantitative data from function (dataProcess) as input and automatically generate three types of figures in pdf files as output : (1) profile plot (specify "ProfilePlot" in option type), to identify the potential sources of variation for each protein; (2) quality control plot (specify "QCPlot" in option type), to evaluate the systematic bias between MS runs; (3) mean plot for conditions (specify "ConditionPlot" in option type), to illustrate mean and variability of each condition per protein.

Usage

```
dataProcessPlots(data=data,
type=type,
featureName="Transition",
ylimUp=FALSE,
ylimDown=FALSE,
scale=FALSE,
interval="CI",
x.axis.size=10,
y.axis.size=10,
text.size=4,
text.angle=0,
legend.size=7,
dot.size.profile=2,
dot.size.condition=3,
width=10,
height=10,
which.Protein="all",
originalPlot=TRUE,
summaryPlot=TRUE,
save_condition_plot_result=FALSE,
address="")
```

Arguments

data	name of the (output of dataProcess function) data set.
type	choice of visualization. "ProfilePlot" represents profile plot of log intensities across MS runs. "QCPlot" represents quality control plot of log intensities across MS runs. "ConditionPlot" represents mean plot of log ratios (Light/Heavy) across conditions.
featureName	for "ProfilePlot" only, "Transition" (default) means printing feature legend in transition-level; "Peptide" means printing feature legend in peptide-level; "NA" means no feature legend printing.
ylimUp	upper limit for y-axis in the log scale. FALSE(Default) for Profile Plot and QC Plot use the upper limit as rounded off maximum of log2(intensities) after normalization + 3. FALSE(Default) for Condition Plot is maximum of log ratio + SD or CI.

`ylimDown` lower limit for y-axis in the log scale. FALSE(Default) for Profile Plot and QC Plot is 0. FALSE(Default) for Condition Plot is minumum of log ratio - SD or CI.

`scale` for "ConditionPlot" only, FALSE(default) means each conditional level is not scaled at x-axis according to its actual value (equal space at x-axis). TRUE means each conditional level is scaled at x-axis according to its actual value (unequal space at x-axis).

`interval` for "ConditionPlot" only, "CI"(default) uses confidence interval with 0.95 significant level for the width of error bar. "SD" uses standard deviation for the width of error bar.

`x.axis.size` size of x-axis labeling for "Run" in Profile Plot and QC Plot, and "Condition" in Condition Plot. Default is 10.

`y.axis.size` size of y-axis labels. Default is 10.

`text.size` size of labels represented each condition at the top of graph in Profile Plot and QC plot. Default is 4.

`text.angle` angle of labels represented each condition at the top of graph in Profile Plot and QC plot or x-axis labeling in Condition plot. Default is 0.

`legend.size` size of feature legend (transition-level or peptide-level) above graph in Profile Plot. Default is 7.

`dot.size.profile`
size of dots in profile plot. Default is 2.

`dot.size.condition`
size of dots in condition plot. Default is 3.

`width` width of the saved file. Default is 10.

`height` height of the saved file. Default is 10.

`which.Protein` Protein list to draw plots. List can be names of Proteins or order numbers of Proteins from levels(data$ProcessedData$PROTEIN). Default is "all", which generates all plots for each protein. For QC plot, "allonly" will generate one QC plot with all proteins.

`originalPlot` TRUE(default) draws original profile plots.

`summaryPlot` TRUE(default) draws profile plots with summarization for run levels.

`save_condition_plot_result`
TRUE saves the table with values using condition plots. Default is FALSE.

`address` the name of folder that will store the results. Default folder is the current working directory. The other assigned folder has to be existed under the current working directory. An output pdf file is automatically created with the default name of "ProfilePlot.pdf" or "QCplot.pdf" or "ConditionPlot.pdf" or "ConditionPlot_value.csv". The command address can help to specify where to store the file as well as how to modify the beginning of the file name. If address=FALSE, plot will be not saved as pdf file but showed in window.

Details

- Profile Plot : identify the potential sources of variation of each protein. QuantData$ProcessedData is used for plots. X-axis is run. Y-axis is log-intensities of transitions. Reference/endogenous signals are in the left/right panel. Line colors indicate peptides and line types indicate transitions. In summarization plots, gray dots and lines are the same as original profile plots with QuantData$ProcessedData. Dark dots and lines are for summarized intensities from QuantData$RunlevelData.

- QC Plot : illustrate the systematic bias between MS runs. After normalization, the reference signals for all proteins should be stable across MS runs. QuantData$ProcessedData is used for plots. X-axis is run. Y-axis is log-intensities of transition. Reference/endogenous signals are in the left/right panel. The pdf file contains (1) QC plot for all proteins and (2) QC plots for each protein separately.
- Condition Plot : illustrate the systematic difference between conditions. Summarized intensnties from QuantData$RunlevelData are used for plots. X-axis is condition. Y-axis is summarized log transformed intensity. If scale is TRUE, the levels of conditions is scaled according to its actual values at x-axis. Red points indicate the mean for each condition. If interval is "CI", blue error bars indicate the confidence interval with 0.95 significant level for each condition. If interval is "SD", blue error bars indicate the standard deviation for each condition.The interval is not related with model-based analysis.

The input of this function is the quantitative data from function (dataProcess).

Value

pdf will be generated under the working directory.

Author(s)

Meena Choi, Olga Vitek.

Maintainer: Meena Choi (<mnchoi67@gmail.com>)

References

Meena Choi, Ching-Yun Chang, Timothy Clough, Daniel Broudy, Trevor Killeen, Brendan MacLean and Olga Vitek. "MSstats: an R package for statistical analysis of quantitative mass spectrometry-based proteomic experiments" *Bioinformatics*, 30(17):2524-2526, 2014.

Ching-Yun Chang, Paola Picotti, Ruth Huttenhain, Viola Heinzelmann-Schwarz, Marko Jovanovic, Ruedi Aebersold, Olga Vitek. "Protein significance analysis in selected reaction monitoring (SRM) measurements." *Molecular & Cellular Proteomics*, 11:M111.014662, 2012.

Timothy Clough, Safia Thaminy, Susanne Ragg, Ruedi Aebersold, Olga Vitek. "Statistical protein quantification and significance analysis in label-free LC-M experiments with complex designs" *BMC Bioinformatics*, 13:S16, 2012.

Examples

```
# Consider quantitative data (i.e. QuantData) from a yeast study with ten time points of interests,
# three biological replicates, and no technical replicates which is a time-course experiment.
# The goal is to provide pre-analysis visualization by automatically generate two types of figures
# in two separate pdf files.
# Protein IDHC (gene name IDP2) is differentially expressed in time point 1 and time point 7,
# whereas, Protein PMG2 (gene name GPM2) is not.

QuantData<-dataProcess(SRMRawData)
head(QuantData$ProcessedData)

# Profile plot
dataProcessPlots(data=QuantData,type="ProfilePlot")

# Quality control plot
dataProcessPlots(data=QuantData,type="QCPlot")
```

```
# Quantification plot for conditions
dataProcessPlots(data=QuantData,type="ConditionPlot")
```

DDARawData *Example dataset from a label-free DDA, a controlled spike-in experiment.*

Description

This is a data set obtained from a published study (Mueller, et. al, 2007). A controlled spike-in experiment, where 6 proteins, (horse myoglobin, bovine carbonic anhydrase, horse Cytochrome C, chicken lysozyme, yeast alcohol dehydrogenase, rabbit aldolase A) were spiked into a complex background in known concentrations in a latin square design. The experiment contained 6 mixtures, and each mixture was analyzed in label-free LC-MS mode with 3 technical replicates (resulting in the total of 18 runs). Each protein was represented by 7-21 peptides, and each peptide was represented by 1-5 transition.

Usage

```
DDARawData
```

Format

data.frame

Details

The raw data (input data for MSstats) is required to contain variable of ProteinName, PeptideSequence, PrecursorCharge, FragmentIon, ProductCharge, IsotopeLabelType, Condition, BioReplicate, Run, Intensity. The variable names should be fixed.

If the information of one or more columns is not available for the original raw data, please retain the column variables and type in fixed value. For example, the original raw data does not contain the information of PrecursorCharge and ProductCharge, we retain the column PrecursorCharge and ProductCharge and then type in NA for all transitions in RawData.

Variable Intensity is required to be original signal without any log transformation and can be specified as the peak of height or the peak of area under curve.

Value

data.frame with the required format of MSstats.

Author(s)

Meena Choi, Olga Vitek.

Maintainer: Meena Choi (<mnchoi67@gmail.com>)

References

Meena Choi, Ching-Yun Chang, Timothy Clough, Daniel Broudy, Trevor Killeen, Brendan MacLean and Olga Vitek. "MSstats: an R package for statistical analysis of quantitative mass spectrometry-based proteomic experiments" *Bioinformatics*, 30(17):1514-1526, 2014.

Timothy Clough, Safia Thaminy, Susanne Ragg, Ruedi Aebersold, Olga Vitek. "Statistical protein quantification and significance analysis in label-free LC-M experiments with complex designs" *BMC Bioinformatics*, 13:S16, 2012.

Mueller, L. N., Rinner, O., Schmidt, A., Letarte, S., Bodenmiller, B., Brusniak, M., Vitek, O., Aebersold, R., and Muller, M. (2007). SuperHirn - a novel tool for high resolution LC-MS based peptide/protein profiling. Proteomics, 7, 3470-3480. 3, 34

Examples

```
head(DDARawData)
```

`DDARawData.Skyline` *Example dataset from a label-free DDA, a controlled spike-in experiment, processed by Skyline.*

Description

This is a data set obtained from a published study (Mueller, et. al, 2007). A controlled spike-in experiment, where 6 proteins, (horse myoglobin, bovine carbonic anhydrase, horse Cytochrome C, chicken lysozyme, yeast alcohol dehydrogenase, rabbit aldolase A) were spiked into a complex background in known concentrations in a latin square design. The experiment contained 6 mixtures, and each mixture was analyzed in label-free LC-MS mode with 3 technical replicates (resulting in the total of 18 runs). Each protein was represented by 7-21 peptides, and each peptide was represented by 1-5 transition. Skyline is used for processing.

Usage

```
DDARawData.Skyline
```

Format

data.frame

Details

The raw data (input data for MSstats) is required to contain variable of ProteinName, PeptideSequence, PrecursorCharge, FragmentIon, ProductCharge, IsotopeLabelType, Condition, BioReplicate, Run, Intensity. The variable names should be fixed.

This is 'MSstats input' format from Skyline used by 'MSstats_report.skyr'. The column names, 'FileName' and 'Area', should be changed to 'Run' and 'Intensity'. There are two extra columns called 'StandardType' and 'Truncated'.'StandardType' column can be used for normalization='globalStandard' in `dataProcess`. 'Truncated' columns can be used to remove the truncated peaks with skylineReport=TRUE in `dataProcess`.

If the information of one or more columns is not available for the original raw data, please retain the column variables and type in fixed value. For example, the original raw data does not contain

the information of PrecursorCharge and ProductCharge, we retain the column PrecursorCharge and ProductCharge and then type in NA for all transitions in RawData.

Variable Intensity is required to be original signal without any log transformation and can be specified as the peak of height or the peak of area under curve.

Value

data.frame with the required format of MSstats.

Author(s)

Meena Choi, Olga Vitek.

Maintainer: Meena Choi (<mnchoi67@gmail.com>)

References

Meena Choi, Ching-Yun Chang, Timothy Clough, Daniel Broudy, Trevor Killeen, Brendan MacLean and Olga Vitek. "MSstats: an R package for statistical analysis of quantitative mass spectrometry-based proteomic experiments" *Bioinformatics*, 30(17):1514-1526, 2014.

Timothy Clough, Safia Thaminy, Susanne Ragg, Ruedi Aebersold, Olga Vitek. "Statistical protein quantification and significance analysis in label-free LC-M experiments with complex designs" *BMC Bioinformatics*, 13:S16, 2012.

Examples

```
head(DDARawData.Skyline)
```

---

designSampleSize *Planning future experimental designs of Selected Reaction Monitoring (SRM), Data-Dependent Acquisition (DDA or shotgun), and Data-Independent Acquisition (DIA or SWATH-MS) experiments in sample size calculation*

Description

Calculate sample size for future experiments of a Selected Reaction Monitoring (SRM), Data-Dependent Acquisition (DDA or shotgun), and Data-Independent Acquisition (DIA or SWATH-MS) experiment based on intensity-based linear model. Two options of the calculation: (1) number of biological replicates per condition, (2) power.

Usage

```
designSampleSize(data=data,desiredFC=desiredFC,FDR=0.05,numSample=TRUE,power=0.9)
```

Arguments

data	'fittedmodel' in testing output from function groupComparison.
desiredFC	the range of a desired fold change which includes the lower and upper values of the desired fold change.
FDR	a pre-specified false discovery ratio (FDR) to control the overall false positive. Default is 0.05

numSample	minimal number of biological replicates per condition. TRUE represents you require to calculate the sample size for this category, else you should input the exact number of biological replicates.
power	a pre-specified statistical power which defined as the probability of detecting a true fold change. TRUE represent you require to calculate the power for this category, else you should input the average of power you expect. Default is 0.9

Details

The function fits the model and uses variance components to calculate sample size. The underlying model fitting with intensity-based linear model with technical MS run replication. Estimated sample size is rounded to 0 decimal.

Value

A list of the sample size calculation results including Variable desiredFC, numSample, numPep, numTran, FDR, and power.

Warning

It can only obtain either one of the categories of the sample size calculation (numSample, numPep, numTran, power) at the same time.

Author(s)

Meena Choi, Ching-Yun Chang, Olga Vitek.

Maintainer: Meena Choi (<mnchoi67@gmail.com>)

References

Meena Choi, Ching-Yun Chang, Timothy Clough, Daniel Broudy, Trevor Killeen, Brendan MacLean and Olga Vitek. "MSstats: an R package for statistical analysis of quantitative mass spectrometry-based proteomic experiments" *Bioinformatics*, 30(17):2524-2526, 2014.

Ching-Yun Chang, Paola Picotti, Ruth Huttenhain, Viola Heinzelmann-Schwarz, Marko Jovanovic, Ruedi Aebersold, Olga Vitek. "Protein significance analysis in selected reaction monitoring (SRM) measurements." *Molecular & Cellular Proteomics*, 11:M111.014662, 2012.

Timothy Clough, Safia Thaminy, Susanne Ragg, Ruedi Aebersold, Olga Vitek. "Statistical protein quantification and significance analysis in label-free LC-M experiments with complex designs" *BMC Bioinformatics*, 13:S16, 2012.

Examples

```
# Consider quantitative data (i.e. QuantData) from yeast study.
# A time course study with ten time points of interests and three biological replicates.

QuantData<-dataProcess(SRMRawData)
head(QuantData$ProcessedData)

## based on multiple comparisons  (T1 vs T3; T1 vs T7; T1 vs T9)
comparison1<-matrix(c(-1,0,1,0,0,0,0,0,0,0),nrow=1)
comparison2<-matrix(c(-1,0,0,0,0,0,1,0,0,0),nrow=1)
comparison3<-matrix(c(-1,0,0,0,0,0,0,0,1,0),nrow=1)
comparison<-rbind(comparison1,comparison2, comparison3)
```

```
row.names(comparison)<-c("T3-T1","T7-T1","T9-T1")

testResultMultiComparisons<-groupComparison(contrast.matrix=comparison,data=QuantData)


## Calculate sample size for future experiments:

#(1) Minimal number of biological replicates per condition

designSampleSize(data=testResultMultiComparisons$fittedmodel, numSample=TRUE,
desiredFC=c(1.25,1.75), FDR=0.05, power=0.8)


#(2) Power calculation

designSampleSize(data=testResultMultiComparisons$fittedmodel, numSample=2,
desiredFC=c(1.25,1.75), FDR=0.05, power=TRUE)
```

`designSampleSizeClassification` *Estimate the optimal size of training data for classification problem*

Description

For classification problem (such as disgnosys of disease), calculate the mean predictive accuray under different size of training data for future experiments of a Selected Reaction Monitoring (SRM), Data-Dependent Acquisition (DDA or shotgun), and Data-Independent Acquisition (DIA or SWATH-MS) experiment based on simulation.

Usage

```
designSampleSizeClassification(data, n_sample = 5, sample_incr = 20,
  protein_desc = 0.2, iter = 10)
```

Arguments

`data`	output from function `dataProcess`
`n_sample`	number of different sample size to simulate. Default is 5
`sample_incr`	number of samples per condition to increase at each step. Default is 20
`protein_desc`	the fraction of proteins to reduce at each step. Proteins are ranked based on their mean abundance across all the samples. Default is 0.2. If protein_desc = 0.0, protein number will not be changed.
`iter`	number of times to repeat simulation experiments. Default is 10

Details

The function fits intensity-based linear model on the input prelimiary data *data* and uses variance components and mean abundance to simulate new training data with different sample size and protein number. Random forest model is fitted on simulated train data and used to predict the input preliminary data *data*. The above procedure is repeated *iter* times. Mean predictive accuracy and variance under different size of training data are reported.

Value

meanPA is the mean predictive accuracy matrix under different size of training data.

varPA is variance of predictive accuracy under different size of training data.

Author(s)

Ting Huang, Meena Choi, Olga Vitek.

Maintainer: Meena Choi (<mnchoi67@gmail.com>)

References

T. Huang et al. TBD 2018

Examples

```
# Consider the training set from a colorectal cancer study
# Subjects are from control group or colorectal cancer group
# 72 proteins were targeted with SRM
require(MSstatsBioData)
set.seed(1235)
data(SRM_crc_training)
QuantCRCSRM <- dataProcess(SRM_crc_training, normalization = FALSE)
# estimate the mean predictive accuray under different sizes of training data
# n_sample is the number of different sample size to simulate
# Datasets with 10 different sample size and 3 different protein numbers are simulated
result.crc.srm <- designSampleSizeClassification(data=QuantCRCSRM,
n_sample = 10,
sample_incr = 10,
protein_desc = 0.33,
iter = 50)
result.crc.srm$meanPA # mean predictive accuracy
```

---

designSampleSizeClassificationPlots

Visualization for sample size calculation in classification problem

---

Description

To illustrate the mean classification accuracy under different protein number and sample size. The input is the result from function designSampleSizeClassification.

Usage

```
designSampleSizeClassificationPlots(data)
```

Arguments

data	output from function designSampleSizeClassification

Details

Data in the example is based on the results of sample size calculation in classification problem from function designSampleSizeClassification

Value

Plot for sample size estimation. x-axis : sample size, y-axis: mean predictive accuracy. Color: different protein number.

Author(s)

Ting Huang, Meena Choi, Olga Vitek.

Maintainer: Meena Choi (<mnchoi67@gmail.com>)

References

T. Huang et al. TBD 2018

Examples

```
# Consider the training set from a colorectal cancer study
# Subjects are from control group or colorectal cancer group
# 72 proteins were targeted with SRM
require(MSstatsBioData)
set.seed(1235)
data(SRM_crc_training)
QuantCRCSRM <- dataProcess(SRM_crc_training, normalization = FALSE)
# estimate the mean predictive accuray under different sizes of training data
# n_sample is the number of different sample size to simulate
# Datasets with 10 different sample size and 3 different protein numbers are simulated
result.crc.srm <- designSampleSizeClassification(data=QuantCRCSRM,
n_sample = 10,
sample_incr = 10,
protein_desc = 0.33,
iter = 50)
designSampleSizeClassificationPlots(data=result.crc.srm)
```

`designSampleSizePlots` *Visualization for sample size calculation*

Description

To illustrate the relationship of desired fold change and the calculated minimal number sample size which are (1) number of biological replicates per condition, (2) number of peptides per protein, (3) number of transitions per peptide, and (4) power. The input is the result from function (`designSampleSize`.

Usage

```
designSampleSizePlots(data=data)
```

Arguments

`data`	output from function designSampleSize.

Details

Data in the example is based on the results of sample size calculation from function `designSampleSize`.

Value

Plot for estimated sample size with assigned variable.

Author(s)

Meena Choi, Ching-Yun Chang, Olga Vitek.

Maintainer: Meena Choi (<mnchoi67@gmail.com>)

References

Meena Choi, Ching-Yun Chang, Timothy Clough, Daniel Broudy, Trevor Killeen, Brendan MacLean and Olga Vitek. "MSstats: an R package for statistical analysis of quantitative mass spectrometry-based proteomic experiments" *Bioinformatics*, 30(17):2524-2526, 2014.

Ching-Yun Chang, Paola Picotti, Ruth Huttenhain, Viola Heinzelmann-Schwarz, Marko Jovanovic, Ruedi Aebersold, Olga Vitek. "Protein significance analysis in selected reaction monitoring (SRM) measurements." *Molecular & Cellular Proteomics*, 11:M111.014662, 2012.

Timothy Clough, Safia Thaminy, Susanne Ragg, Ruedi Aebersold, Olga Vitek. "Statistical protein quantification and significance analysis in label-free LC-M experiments with complex designs" *BMC Bioinformatics*, 13:S16, 2012.

Examples

```
# Based on the results of sample size calculation from function designSampleSize,
# we generate a series of sample size plots for number of biological replicates, or peptides,
# or transitions or power plot.

QuantData<-dataProcess(SRMRawData)
head(QuantData$ProcessedData)

## based on multiple comparisons  (T1 vs T3; T1 vs T7; T1 vs T9)
comparison1<-matrix(c(-1,0,1,0,0,0,0,0,0,0),nrow=1)
comparison2<-matrix(c(-1,0,0,0,0,0,1,0,0,0),nrow=1)
comparison3<-matrix(c(-1,0,0,0,0,0,0,0,1,0),nrow=1)
comparison<-rbind(comparison1,comparison2, comparison3)
row.names(comparison)<-c("T3-T1","T7-T1","T9-T1")

testResultMultiComparisons<-groupComparison(contrast.matrix=comparison,data=QuantData)


# plot the calculated sample sizes for future experiments:

# (1) Minimal number of biological replicates per condition

result.sample<-designSampleSize(data=testResultMultiComparisons$fittedmodel, numSample=TRUE,
desiredFC=c(1.25,1.75), FDR=0.05, power=0.8)
designSampleSizePlots(data=result.sample)

# (2) Power

result.power<-designSampleSize(data=testResultMultiComparisons$fittedmodel, numSample=2,
desiredFC=c(1.25,1.75), FDR=0.05, power=TRUE)
designSampleSizePlots(data=result.power)
```

DIARawData *Example dataset from a label-free DIA, a group comparison study of S.Pyogenes.*

Description

This example dataset was obtained from a group comparison study of S. Pyogenes. Two conditions, S. Pyogenes with 0% and 10% of human plasma added (denoted Strep 0% and Strep 10%), were profiled in two replicates, in the label-free mode, with a SWATH-MS-enabled AB SCIEX TripleTOF 5600 System. The identification and quantification of spectral peaks was assisted by a spectral library, and was performed using OpenSWATH software (http: //proteomics.ethz.ch/openswath.html). For reasons of space, the example dataset only contains two proteins from this study. Protein FabG shows strong evidence of differential abundance, while protein Probable RNA helicase exp9 only shows moderate evidence of dif- ferential abundance between conditions.

Usage

```
DIARawData
```

Format

data.frame

Details

The raw data (input data for MSstats) is required to contain variable of ProteinName, PeptideSequence, PrecursorCharge, FragmentIon, ProductCharge, IsotopeLabelType, Condition, BioReplicate, Run, Intensity. The variable names should be fixed.

If the information of one or more columns is not available for the original raw data, please retain the column variables and type in fixed value. For example, the original raw data does not contain the information of PrecursorCharge and ProductCharge, we retain the column PrecursorCharge and ProductCharge and then type in NA for all transitions in RawData.

Variable Intensity is required to be original signal without any log transformation and can be specified as the peak of height or the peak of area under curve.

Value

data.frame with the required format of MSstats.

Author(s)

Meena Choi, Olga Vitek.

Maintainer: Meena Choi (<mnchoi67@gmail.com>)

Examples

```
head(DIARawData)
```

`DIAUmpiretoMSstatsFormat` *Generate MSstats required input format for DIA-Umpire output*

Description

Convert DIA-Umpire output into the required input format for MSstats.

Usage

```
DIAUmpiretoMSstatsFormat(raw.frag, raw.pep, raw.pro,
    annotation,
    useSelectedFrag = TRUE,
    useSelectedPep = TRUE,
    fewMeasurements="remove",
    removeProtein_with1Feature = FALSE,
    summaryforMultipleRows=max)
```

Arguments

`raw.frag`	name of FragSummary_date.xls data, which includes feature-level data.
`raw.pep`	name of PeptideSummary_date.xls data, which includes selected fragments information.
`raw.pro`	name of ProteinSummary_date.xls data, which includes selected peptides information.
`annotation`	name of annotation data which includes Condition, BioReplicate, Run information.
`useSelectedFrag`	TRUE will use the selected fragment for each peptide. 'Selected_fragments' column is required.
`useSelectedPep`	TRUE will use the selected peptide for each protein. 'Selected_peptides' column is required.
`fewMeasurements`	'remove'(default) will remove the features that have 1 or 2 measurements across runs.
`removeProtein_with1Feature`	TRUE will remove the proteins which have only 1 feature, which is the combination of peptide, precursor charge, fragment and charge. FALSE is default.
`summaryforMultipleRows`	max(default) or sum - when there are multiple measurements for certain feature and certain run, use highest or sum of multiple intensities.

Value

data.frame with the required format of MSstats.

Author(s)

Meena Choi, Olga Vitek.

Maintainer: Meena Choi (<mnchoi67@gmail.com>)

Examples

```
# Manual will be updated.
# Output of DIAUmpiretoMSstatsFormat function
# should have the same 10 columns as an example dataset.

head(DDARawData)
```

`groupComparison` *Finding differentially abundant proteins across conditions in targeted Selected Reaction Monitoring (SRM), Data-Dependent Acquisition (DDA or shotgun), and Data-Independent Acquisition (DIA or SWATH-MS) experiment*

Description

Tests for significant changes in protein abundance across conditions based on a family of linear mixed-effects models in targeted Selected Reaction Monitoring (SRM), Data-Dependent Acquisition (DDA or shotgun), and Data-Independent Acquisition (DIA or SWATH-MS) experiment. It is applicable to multiple types of sample preparation, including label-free workflows, workflows that use stable isotope labeled reference proteins and peptides, and workflows that use fractionation. Experimental design of case-control study (patients are not repeatedly measured) or time course study (patients are repeatedly measured) is automatically determined based on proper statistical model.

Usage

```
groupComparison(contrast.matrix=contrast.matrix,
                data=data)
```

Arguments

`contrast.matrix`	comparison between conditions of interests.
`data`	name of the (output of dataProcess function) data set.

Details

- contrast.matrix : comparison of interest. Based on the levels of conditions, specify 1 or -1 to the conditions of interests and 0 otherwise. The levels of conditions are sorted alphabetically. Command levels(QuantData$ProcessedData$GROUP_ORIGINAL) can illustrate the actual order of the levels of conditions.

The underlying model fitting functions are `lm` and `lmer` for the fixed effects model and mixed effects model, respectively.

The input of this function is the quantitative data from function (`dataProcess`).

Value

A list of data.frame *ComparisonResult* is the data.frame for the result of significance analysis ; *fittedModel* is the the data.frame for run-level summarized data.

Warning

When a feature is missing completely in a condition or a MS run, a warning message is sent to the console notifying the user of the missing feature. Additional filtering or imputing process is required before model fitting.

Author(s)

Meena Choi, Ching-Yun Chang, Olga Vitek.

Maintainer: Meena Choi (<mnchoi67@gmail.com>)

References

Meena Choi, Ching-Yun Chang, Timothy Clough, Daniel Broudy, Trevor Killeen, Brendan MacLean and Olga Vitek. "MSstats: an R package for statistical analysis of quantitative mass spectrometry-based proteomic experiments" *Bioinformatics*, 30(17):2524-2526, 2014.

Ching-Yun Chang, Paola Picotti, Ruth Huttenhain, Viola Heinzelmann-Schwarz, Marko Jovanovic, Ruedi Aebersold, Olga Vitek. "Protein significance analysis in selected reaction monitoring (SRM) measurements." *Molecular & Cellular Proteomics*, 11:M111.014662, 2012.

Timothy Clough, Safia Thaminy, Susanne Ragg, Ruedi Aebersold, Olga Vitek. "Statistical protein quantification and significance analysis in label-free LC-M experiments with complex designs" *BMC Bioinformatics*, 13:S16, 2012.

Examples

```
# Consider quantitative data (i.e. QuantData) from yeast study with ten time points of interests,
# three biological replicates, and no technical replicates.
# It is a time-course experiment and we attempt to compare differential abundance
# between time 1 and 7 in a set of targeted proteins.
# In this label-based SRM experiment, MSstats uses the fitted model with expanded scope of
# Biological replication.

QuantData <- dataProcess(SRMRawData)
head(QuantData$ProcessedData)

levels(QuantData$ProcessedData$GROUP_ORIGINAL)
comparison <- matrix(c(-1,0,0,0,0,0,1,0,0,0),nrow=1)
row.names(comparison) <- "T7-T1"

# Tests for differentially abundant proteins with models:
# label-based SRM experiment with expanded scope of biological replication.

testResultOneComparison <- groupComparison(contrast.matrix=comparison, data=QuantData)

# table for result
testResultOneComparison$ComparisonResult
```

groupComparisonPlots *Visualization for model-based analysis and summarizing differentially abundant proteins*

Description

To summarize the results of log-fold changes and adjusted p-values for differentially abundant proteins, groupComparisonPlots takes testing results from function (groupComparison) as input and automatically generate three types of figures in pdf files as output : (1) volcano plot (specify "VolcanoPlot" in option type) for each comparison separately; (2) heatmap (specify "Heatmap" in option type) for multiple comparisons ; (3) comparison plot (specify "ComparisonPlot" in option type) for multiple comparisons per protein.

Usage

```
groupComparisonPlots(data=data,
type=type,
sig=0.05,
FCcutoff=FALSE,
logBase.pvalue=10,
ylimUp=FALSE,
ylimDown=FALSE,
xlimUp=FALSE,
x.axis.size=10,
y.axis.size=10,
dot.size=3,
text.size=4,
legend.size=13,
ProteinName=TRUE,
colorkey=TRUE,
numProtein=100,
clustering="both",
width=10,
height=10,
which.Comparison="all",
which.Protein="all",
address="")
```

Arguments

data	'ComparisonResult' in testing output from function groupComparison.
type	choice of visualization. "VolcanoPlot" represents volcano plot of log fold changes and adjusted p-values for each comparison separately. "Heatmap" represents heatmap of adjusted p-values for multiple comparisons. "ComparisonPlot" represents comparison plot of log fold changes for multiple comparisons per protein.
sig	FDR cutoff for the adjusted p-values in heatmap and volcano plot. level of significance for comparison plot. 100(1-sig)% confidence interval will be drawn. sig=0.05 is default.

FCcutoff	for volcano plot or heatmap, whether involve fold change cutoff or not. FALSE (default) means no fold change cutoff is applied for significance analysis. FC-cutoff = specific value means specific fold change cutoff is applied.
logBase.pvalue	for volcano plot or heatmap, (-) logarithm transformation of adjusted p-value with base 2 or 10(default).
ylimUp	for all three plots, upper limit for y-axis. FALSE (default) for volcano plot/heatmap use maximum of -log2 (adjusted p-value) or -log10 (adjusted p-value). FALSE (default) for comparison plot uses maximum of log-fold change + CI.
ylimDown	for all three plots, lower limit for y-axis. FALSE (default) for volcano plot/heatmap use minimum of -log2 (adjusted p-value) or -log10 (adjusted p-value). FALSE (default) for comparison plot uses minimum of log-fold change - CI.
xlimUp	for Volcano plot, the limit for x-axis. FALSE (default) for use maximum for absolute value of log-fold change or 3 as default if maximum for absolute value of log-fold change is less than 3.
x.axis.size	size of axes labels, e.g. name of the comparisons in heatmap, and in comparison plot. Default is 10.
y.axis.size	size of axes labels, e.g. name of targeted proteins in heatmap. Default is 10.
dot.size	size of dots in volcano plot and comparison plot. Default is 3.
text.size	size of ProteinName label in the graph for Volcano Plot. Default is 4.
legend.size	size of legend for color at the bottom of volcano plot. Default is 7.
ProteinName	for volcano plot only, whether display protein names or not. TRUE (default) means protein names, which are significant, are displayed next to the points. FALSE means no protein names are displayed.
colorkey	TRUE(default) shows colorkey.
numProtein	The number of proteins which will be presented in each heatmap. Default is 100. Maximum possible number of protein for one heatmap is 180.
clustering	Determines how to order proteins and comparisons. Hierarchical cluster analysis with Ward method(minimum variance) is performed. 'protein' means that protein dendrogram is computed and reordered based on protein means (the order of row is changed). 'comparison' means comparison dendrogram is computed and reordered based on comparison means (the order of comparison is changed). 'both' means to reorder both protein and comparison. Default is 'protein'.
width	width of the saved file. Default is 10.
height	height of the saved file. Default is 10.
which.Comparison	list of comparisons to draw plots. List can be labels of comparisons or order numbers of comparisons from levels(data$Label), such as levels(testResultMultiComparisons$Compa Default is "all", which generates all plots for each protein.
which.Protein	Protein list to draw comparison plots. List can be names of Proteins or order numbers of Proteins from levels(testResultMultiComparisons$ComparisonResult$Protein). Default is "all", which generates all comparison plots for each protein.
address	the name of folder that will store the results. Default folder is the current working directory. The other assigned folder has to be existed under the current working directory. An output pdf file is automatically created with the default name of "VolcanoPlot.pdf" or "Heatmap.pdf" or "ComparisonPlot.pdf". The command address can help to specify where to store the file as well as how to modify the beginning of the file name. If address=FALSE, plot will be not saved as pdf file but showed in window.

Details

- Volcano plot : illustrate actual log-fold changes and adjusted p-values for each comparison separately with all proteins. The x-axis is the log fold change. The base of logarithm transformation is the same as specified in "logTrans" from dataProcess. The y-axis is the negative log2 or log10 adjusted p-values. The horizontal dashed line represents the FDR cutoff. The points below the FDR cutoff line are non-significantly abundant proteins (colored in black). The points above the FDR cutoff line are significantly abundant proteins (colored in red/blue for up-/down-regulated). If fold change cutoff is specified (FCcutoff = specific value), the points above the FDR cutoff line but within the FC cutoff line are non-significantly abundant proteins (colored in black)/
- Heatmap : illustrate up-/down-regulated proteins for multiple comparisons with all proteins. Each column represents each comparison of interest. Each row represents each protein. Color red/blue represents proteins in that specific comparison are significantly up-regulated/down-regulated proteins with FDR cutoff and/or FC cutoff. The color scheme shows the evidences of significance. The darker color it is, the stronger evidence of significance it has. Color gold represents proteins are not significantly different in abundance.
- Comparison plot : illustrate log-fold change and its variation of multiple comparisons for single protein. X-axis is comparison of interest. Y-axis is the log fold change. The red points are the estimated log fold change from the model. The blue error bars are the confidence interval with 0.95 significant level for log fold change. This interval is only based on the standard error, which is estimated from the model.

The input of this function is "ComparisonResult" in the testing results from function (groupComparison).

Value

pdf file

Author(s)

Meena Choi, Olga Vitek.

Maintainer: Meena Choi (<mnchoi67@gmail.com>)

References

Meena Choi, Ching-Yun Chang, Timothy Clough, Daniel Broudy, Trevor Killeen, Brendan MacLean and Olga Vitek. "MSstats: an R package for statistical analysis of quantitative mass spectrometry-based proteomic experiments" *Bioinformatics*, 30(17):2524-2526, 2014.

Ching-Yun Chang, Paola Picotti, Ruth Huttenhain, Viola Heinzelmann-Schwarz, Marko Jovanovic, Ruedi Aebersold, Olga Vitek. "Protein significance analysis in selected reaction monitoring (SRM) measurements." *Molecular & Cellular Proteomics*, 11:M111.014662, 2012.

Timothy Clough, Safia Thaminy, Susanne Ragg, Ruedi Aebersold, Olga Vitek. "Statistical protein quantification and significance analysis in label-free LC-M experiments with complex designs" *BMC Bioinformatics*, 13:S16, 2012.

Examples

```
QuantData<-dataProcess(SRMRawData)
head(QuantData$ProcessedData)

## based on multiple comparisons  (T1 vs T3; T1 vs T7; T1 vs T9)
```

```
comparison1<-matrix(c(-1,0,1,0,0,0,0,0,0,0),nrow=1)
comparison2<-matrix(c(-1,0,0,0,0,0,1,0,0,0),nrow=1)
comparison3<-matrix(c(-1,0,0,0,0,0,0,0,1,0),nrow=1)
comparison<-rbind(comparison1,comparison2, comparison3)
row.names(comparison)<-c("T3-T1","T7-T1","T9-T1")

testResultMultiComparisons<-groupComparison(contrast.matrix=comparison,data=QuantData)

testResultMultiComparisons$ComparisonResult

# Volcano plot with FDR cutoff = 0.05 and no FC cutoff
groupComparisonPlots(data=testResultMultiComparisons$ComparisonResult, type="VolcanoPlot",
logBase.pvalue=2, address="Ex1_")

# Volcano plot with FDR cutoff = 0.05, FC cutoff = 70, upper y-axis limit = 100,
# and no protein name displayed
# FCcutoff=70 is for demonstration purpose
groupComparisonPlots(data=testResultMultiComparisons$ComparisonResult, type="VolcanoPlot",
FCcutoff=70, logBase.pvalue=2, ylimUp=100, ProteinName=FALSE,address="Ex2_")

# Heatmap with FDR cutoff = 0.05
groupComparisonPlots(data=testResultMultiComparisons$ComparisonResult, type="Heatmap",
logBase.pvalue=2, address="Ex1_")

# Heatmap with FDR cutoff = 0.05 and FC cutoff = 70
# FCcutoff=70 is for demonstration purpose
groupComparisonPlots(data=testResultMultiComparisons$ComparisonResult, type="Heatmap",
FCcutoff=70, logBase.pvalue=2, address="Ex2_")

# Comparison Plot
groupComparisonPlots(data=testResultMultiComparisons$ComparisonResult, type="ComparisonPlot",
address="Ex1_")

# Comparison Plot
groupComparisonPlots(data=testResultMultiComparisons$ComparisonResult, type="ComparisonPlot",
ylimUp=8, ylimDown=-1, address="Ex2_")
```

`linear_quantlim` *Calculation of the LOB and LOD with a linear fit*

Description

This function calculates the value of the LOB (limit of blank) and LOD (limit of detection) from the (Concentration, Intensity) spiked in data. The function also returns the values of the linear curve fit that allows it to be plotted. At least 2 blank samples (characterized by Intensity = 0) are required by this function which are used to calculate the background noise. The LOB is defined as the concentration at which the value of the linear fit is equal to the 95% upper bound of the noise. The LOD is the concentration at which the latter is equal to the 90% lower bound of the prediction interval (5% quantile) of the linear fit. A weighted linear fit is used with weights for every unique concentration proportional to the inverse of variance between replicates.

Usage

```
linear_quantlim(datain, alpha = 0.05, Npoints = 100, Nbootstrap = 500)
```

Arguments

datain	Data frame that contains the input data. The input data frame has to contain the following columns : CONCENTRATION, INTENSITY (both of which are measurements from the spiked in experiment) and NAME which designates the name of the assay (e.g. the name of the peptide or protein)
alpha	Probability level to estimate the LOB/LOD
Npoints	Number of points to use to discretize the concentration line between 0 and the maximum spiked concentration
Nbootstrap	Number of bootstrap samples to use to calculate the prediction interval of the fit. This number has to be increased for very low alpha values or whenever very accurate assay characterization is required.

Details

- datain : Each line of the data frame contains one measurement from the spiked-in experiment. Multiple different INTENSITY values for the same CONCENTRATION are assumed to correspond to different replicates. Blank Samples are characterized by CONCENTRATION = 0.

Value

- Data frame that contains the output of the function. It contains the following columns: i) CONCENTRATION: Concentration values at which the value of the fit is calculated ii) MEAN: The value of the curve fit iii) LOW: The value of the lower bound of the 95% prediction interval iv) UP: The value of the upper bound of the 95% prediction interval v) LOB: The value of the LOB (one column with identical values) vi) LOD: The value of the LOD (one column with identical values) vii) SLOPE: Value of the slope of the linear curve fit where only the spikes above LOD are considered viii) INTERCEPT: Value of the intercept of the linear curve fit where only the spikes above LOD are considered ix) NAME: The name of the assay (identical to that provided in the input) x) METHOD which is always set to LINEAR when this function is used. Each line of the data frame corresponds to a unique concentration value at which the value of the fit and prediction interval are evaluated. More unique concentrations values than in the input data frame are used to increase the accuracy of the LOB/D calculations.

Warning

The LOB and LOD can only be calculated when more than 2 blank samples are included. The data should ideally be plotted using the companion function plot_quantlim to ensure that a linear fit is suited to the data.

Author(s)

Cyril Galitzine, Olga Vitek.

Maintainer: Cyril Galitzine (<cyrildgg@gmail.com>), Meena Choi (<mnchoi67@gmail.com>)

References

C. Galitzine et al. "Nonlinear regression improves accuracy of characterization of multiplexed mass spectrometric assays" *Mol Cell Proteomics*, doi:10.1074/mcp.RA117.000322, 2018.

Examples

```
# Consider data from a spiked-in contained in an example dataset
head(SpikeInDataLinear)

## Not run:
# Call function
linear_quantlim_out <- linear_quantlim(SpikeInDataLinear)

## End(Not run)
```

`MaxQtoMSstatsFormat` *Generate MSstats required input format for MaxQuant output*

Description

Convert MaxQuant output into the required input format for MSstats.

Usage

```
MaxQtoMSstatsFormat(evidence,
      annotation,
      proteinGroups,
      proteinID="Proteins",
      useUniquePeptide=TRUE,
      summaryforMultipleRows=max,
      fewMeasurements="remove",
      removeMpeptides=FALSE,
      removeOxidationMpeptides=FALSE,
      removeProtein_with1Peptide=FALSE)
```

Arguments

`evidence`	name of 'evidence.txt' data, which includes feature-level data.
`annotation`	name of 'annotation.txt' data which includes Raw.file, Condition, BioReplicate, Run, IsotopeLabelType information.
`proteinGroups`	name of 'proteinGroups.txt' data. It needs to matching protein group ID. If proteinGroups=NULL, use 'Proteins' column in 'evidence.txt'.
`proteinID`	'Proteins'(default) or 'Leading.razor.protein' for Protein ID.
`useUniquePeptide`	TRUE(default) removes peptides that are assigned for more than one proteins. We assume to use unique peptide for each protein.
`summaryforMultipleRows`	max(default) or sum - when there are multiple measurements for certain feature and certain run, use highest or sum of all.
`fewMeasurements`	'remove'(default) will remove the features that have 1 or 2 measurements across runs. 'keep' will keep all features.

removeMpeptides
: TRUE will remove the peptides including 'M' sequence. FALSE is default.

removeOxidationMpeptides
: TRUE will remove the peptides including 'oxidation (M)' in modification. FALSE is default.

removeProtein_with1Peptide
: TRUE will remove the proteins which have only 1 peptide and charge. FALSE is default.

Value

data.frame with the required format of MSstats.

Warning

MSstats does not support for metabolic labeling or iTRAQ experiments.

Author(s)

Meena Choi, Olga Vitek.

Maintainer: Meena Choi (<mnchoi67@gmail.com>)

Examples

```
# Please check section 4.3. Suggested workflow with MaxQuant output for DDA in MSstats user manual.
# Output of MaxQtoMSstatsFormat function should have the same 10 columns as an example dataset.

head(DDARawData)
```

---

modelBasedQCPlots *Visualization for model-based quality control in fitting model*

Description

To check the assumption of linear model for whole plot inference, modelBasedQCPlots takes the results after fitting models from function (groupComparison) as input and automatically generate two types of figures in pdf files as output : (1) normal quantile-quantile plot (specify "QQPlot" in option type) for checking normally distributed errors.; (2) residual plot (specify "ResidualPlot" in option type).

Usage

```
modelBasedQCPlots(data,
                  type,
                  axis.size=10,
                  dot.size=3,
                  text.size=7,
                  legend.size=7,
                  width=10,
                  height=10,
                  which.Protein='all',
                  address="")
```

Arguments

data	output from function groupComparison.
type	choice of visualization. "QQPlots" represents normal quantile-quantile plot for each protein after fitting models. "ResidualPlots" represents a plot of residuals versus fitted values for each protein in the dataset.
axis.size	size of axes labels. Default is 10.
dot.size	size of points in the graph for residual plots and QQ plots. Default is 3.
text.size	size of labeling for feature names only in normal quantile-quantile plots separately for each feature. Default is 7.
legend.size	size of legend for feature names only in residual plots. Default is 7.
width	width of the saved file. Default is 10.
height	height of the saved file. Default is 10.
which.Protein	Protein list to draw plots. List can be names of Proteins or order numbers of Proteins from levels(testResultOneComparison$ComparisonResult$Protein). Default is "all", which generates all plots for each protein.
address	the name of folder that will store the results. Default folder is the current working directory. The other assigned folder has to be existed under the current working directory. If type="residualPlots" or "QQPlots", "ResidualPlots.pdf" or "QQPlots.plf" will be generated. The command address can help to specify where to store the file as well as how to modify the beginning of the file name. If address=FALSE, plot will be not saved as pdf file but showed in window.

Details

Results based on statistical models for whole plot level inference are accurate as long as the assumptions of the model are met. The model assumes that the measurement errors are normally distributed with mean 0 and constant variance. The assumption of a constant variance can be checked by examining the residuals from the model.

- QQPlots : a normal quantile-quantile plot for each protein is generated in order to check whether the errors are well approximated by a normal distribution. If points fall approximately along a straight line, then the assumption is appropriate for that protein. Only large deviations from the line are problematic.
- ResidualPlots : The plots of residuals against predicted(fitted) values. If it shows a random scatter, then the assumption is appropriate.

The input of this function is the result from function (groupComparison).

Value

pdf file

Author(s)

Meena Choi, Olga Vitek.

Maintainer: Meena Choi (<mnchoi67@gmail.com>)

References

Meena Choi, Ching-Yun Chang, Timothy Clough, Daniel Broudy, Trevor Killeen, Brendan MacLean and Olga Vitek. "MSstats: an R package for statistical analysis of quantitative mass spectrometry-based proteomic experiments" *Bioinformatics*, 30(17):2524-2526, 2014.

Ching-Yun Chang, Paola Picotti, Ruth Huttenhain, Viola Heinzelmann-Schwarz, Marko Jovanovic, Ruedi Aebersold, Olga Vitek. "Protein significance analysis in selected reaction monitoring (SRM) measurements." *Molecular & Cellular Proteomics*, 11:M111.014662, 2012.

Timothy Clough, Safia Thaminy, Susanne Ragg, Ruedi Aebersold, Olga Vitek. "Statistical protein quantification and significance analysis in label-free LC-M experiments with complex designs" *BMC Bioinformatics*, 13:S16, 2012.

Examples

```
QuantData <- dataProcess(SRMRawData)
head(QuantData$ProcessedData)

levels(QuantData$ProcessedData$GROUP_ORIGINAL)
comparison <- matrix(c(-1,0,0,0,0,0,1,0,0,0),nrow=1)
row.names(comparison) <- "T7-T1"

# Tests for differentially abundant proteins with models:
# label-based SRM experiment with expanded scope of biological replication.

testResultOneComparison <- groupComparison(contrast.matrix=comparison, data=QuantData)

# normal quantile-quantile plots
modelBasedQCPlots(data=testResultOneComparison, type="QQPlots", address="")

# residual plots
modelBasedQCPlots(data=testResultOneComparison, type="ResidualPlots", address="")
```

---

nonlinear_quantlim *Calculation of the LOB and LOD with a nonlinear fit*

Description

This function calculates the value of the LOB (limit of blank) and LOD (limit of detection) from the (Concentration, Intensity) spiked in data. This function should be used instead of the linear function whenever a significant threshold is present at low concentrations. Such threshold is characterized by a signal that is dominated by noise where the mean intensity is constant and independent of concentration. The function also returns the values of the nonlinear curve fit that allows it to be plotted. At least 2 blank samples (characterized by Intensity = 0) are required by this function which are used to calculate the background noise. The LOB is defined as the concentration at which the value of the nonlinear fit is equal to the 95% upper bound of the noise. The LOD is the concentration at which the latter is equal to the 90% lower bound (5% quantile) of the prediction interval of the nonlinear fit. A weighted nonlinear fit is used with weights for every unique concentration proportional to the inverse of variance between replicates. The details behind the calculation of the nonlinear fit can be found in the Reference.

Usage

```
nonlinear_quantlim(datain, alpha = 0.05, Npoints = 100, Nbootstrap = 2000)
```

Arguments

datain	Data frame that contains the input data. The input data frame has to contain the following columns : CONCENTRATION, INTENSITY (both of which are measurements from the spiked in experiment) and NAME which designates the name of the assay (e.g. the name of the peptide or protein)
alpha	Probability level to estimate the LOB/LOD
Npoints	Number of points to use to discretize the concentration line between 0 and the maximum spiked concentration
Nbootstrap	Number of bootstrap samples to use to calculate the prediction interval of the fit. This number has to be increased for very low alpha values or whenever very accurate assay characterization is required.

Details

- datain : Each line of the data frame contains one measurement from the spiked-in experiment. Multiple different INTENSITY values for the same CONCENTRATION are assumed to correspond to different replicates. Blank Samples are characterized by CONCENTRATION = 0.

Value

- Data frame that contains the output of the function. It contains the following columns: i) CONCENTRATION: Concentration values at which the value of the fit is calculated ii) MEAN: The value of the curve fit iii) LOW: The value of the lower bound of the 95% prediction interval iv) UP: The value of the upper bound of the 95% prediction interval v) LOB: The value of the LOB (one column with identical values) vi) LOD: The value of the LOD (one column with identical values) vii) SLOPE: Value of the slope of the linear curve fit where only the spikes above LOD are considered viii) INTERCEPT: Value of the intercept of the linear curve fit where only the spikes above LOD are considered ix) NAME: The name of the assay (identical to that provided in the input) x) METHOD which is always set to NONLINEAR when this function is used. Each line of the data frame corresponds to a unique concentration value at which the value of the fit and prediction interval are evaluated. More unique concentrations values than in the input data frame are used to increase the accuracy of the LOB/D calculations.

Warning

The LOB and LOD can only be calculated when more than 2 blank samples are included. The data should ideally be plotted using the companion function plot_quantlim to ensure that the fit is suited to the data.

Author(s)

Cyril Galitzine, Olga Vitek.

Maintainer: Cyril Galitzine (<cyrildgg@gmail.com>), Meena Choi (<mnchoi67@gmail.com>)

References

C. Galitzine et al. "Nonlinear regression improves accuracy of characterization of multiplexed mass spectrometric assays" *Mol Cell Proteomics*, doi:10.1074/mcp.RA117.000322, 2018.

Examples

```
# Consider data from a spiked-in contained in an example dataset. This dataset contains
# a significant threshold at low concentrations that is not well captured by a linear fit

head(SpikeInDataNonLinear)

## Not run:
# Call function
nonlinear_quantlim_out <- nonlinear_quantlim(SpikeInDataNonLinear)

## End(Not run)
```

`OpenMStoMSstatsFormat` *Generate MSstats required input format for OpenMS output*

Description

Preprocess MSstats input report from OpenSWATH and convert into the required input format for MSstats.

Usage

```
OpenMStoMSstatsFormat(input,
    annotation=NULL,
    useUniquePeptide=TRUE,
    fewMeasurements="remove",
    removeProtein_with1Feature=FALSE,
    summaryforMultipleRows=max)
```

Arguments

`input`	name of MSstats input report from OpenMS, which includes feature(peptide ion)-level data.
`annotation`	name of 'annotation.txt' data which includes Condition, BioReplicate, Run. Run should be the same as filename.
`useUniquePeptide`	TRUE(default) removes peptides that are assigned for more than one proteins. We assume to use unique peptide for each protein.
`fewMeasurements`	'remove'(default) will remove the features that have 1 or 2 measurements across runs.
`removeProtein_with1Feature`	TRUE will remove the proteins which have only 1 peptide and charge. FALSE is default.

summaryforMultipleRows
: max(default) or sum - when there are multiple measurements for certain feature and certain run, use highest or sum of multiple intensities.

Value

data.frame with the required format of MSstats.

Author(s)

Meena Choi, Olga Vitek.

Maintainer: Meena Choi (<mnchoi67@gmail.com>)

Examples

```
# Example will be ready in next version.
```

OpenSWATHtoMSstatsFormat

Generate MSstats required input format for OpenSWATH output

Description

Preprocess MSstats input report from OpenSWATH and convert into the required input format for MSstats.

Usage

```
OpenSWATHtoMSstatsFormat(input,
    annotation = NULL,
    filter_with_mscore = TRUE,
    mscore_cutoff = 0.01,
    useUniquePeptide = TRUE,
    fewMeasurements="remove",
    removeProtein_with1Feature = FALSE,
    summaryforMultipleRows=max)
```

Arguments

input
: name of MSstats input report from OpenSWATH, which includes feature-level data.

annotation
: name of 'annotation.txt' data which includes Condition, BioReplicate, Run. Run should be the same as filename.

filter_with_mscore
: TRUE(default) will filter out the features that have greater than mscore_cutoff in m_score column. Those features will be removed.

mscore_cutoff
: Cutoff for m_score. default is 0.01.

useUniquePeptide
: TRUE(default) removes peptides that are assigned for more than one proteins. We assume to use unique peptide for each protein.

`fewMeasurements`
: 'remove' (default) will remove the features that have 1 or 2 measurements across runs.

`removeProtein_with1Feature`
: TRUE will remove the proteins which have only 1 peptide and charge. FALSE is default.

`summaryforMultipleRows`
: max(default) or sum - when there are multiple measurements for certain feature and certain run, use highest or sum of multiple intensities.

Value

data.frame with the required format of MSstats.

Author(s)

Meena Choi, Olga Vitek.

Maintainer: Meena Choi (<mnchoi67@gmail.com>)

Examples

```
# Example will be ready in next version.
```

`PDtoMSstatsFormat` *Generate MSstats required input format for Proteome discoverer output*

Description

Convert Proteome discoverer output into the required input format for MSstats.

Usage

```
PDtoMSstatsFormat(input,
      annotation,
      useNumProteinsColumn=FALSE,
      useUniquePeptide=TRUE,
      summaryforMultipleRows=max,
      fewMeasurements="remove",
      removeOxidationMpeptides=FALSE,
      removeProtein_with1Peptide=FALSE,
      which.quantification = 'Precursor.Area',
      which.proteinid = 'Protein.Group.Accessions',
      which.sequence = 'Sequence' )
```

Arguments

`input`
: name of Proteome discover PSM output, which is long-format. "Protein.Group.Accessions", "#Proteins", "Sequence", "Modifications", "Charge", "Intensity", "Spectrum.File" are required.

`annotation`
: name of 'annotation.txt' or 'annotation.csv' data which includes Condition, BioReplicate, Run information. 'Run' will be matched with 'Spectrum.File'.

`useNumProteinsColumn`
: TRUE removes peptides which have more than 1 in # Proteins column of PD output.

`useUniquePeptide`
: TRUE(default) removes peptides that are assigned for more than one proteins. We assume to use unique peptide for each protein.

`summaryforMultipleRows`
: max(default) or sum - when there are multiple measurements for certain feature and certain run, use highest or sum of multiple intensities.

`fewMeasurements`
: 'remove'(default) will remove the features that have 1 or 2 measurements across runs.

`removeOxidationMpeptides`
: TRUE will remove the modified peptides including 'Oxidation (M)' in 'Modifications' column. FALSE is default.

`removeProtein_with1Peptide`
: TRUE will remove the proteins which have only 1 peptide and charge. FALSE is default.

`which.quantification`
: Use 'Precursor.Area'(default) column for quantified intensities. 'Intensity' or 'Area' can be used instead.

`which.proteinid`
: Use 'Protein.Accessions'(default) column for protein name. 'Master.Protein.Accessions' can be used instead.

`which.sequence`
: Use 'Sequence'(default) column for peptide sequence. 'Annotated.Sequence' can be used instead.

Value

data.frame with the required format of MSstats.

Author(s)

Meena Choi, Olga Vitek.

Maintainer: Meena Choi (<mnchoi67@gmail.com>)

Examples

```
# Please check section 4.5.
## Suggested workflow with Proteome Discoverer output for DDA in MSstats user manual.
# Output of PDtoMSstatsFormat function should have the same 10 columns as an example dataset.

head(DDARawData)
```

`plot_quantlim` *Plot of the curve used to calculate LOB and LOD*

Description

This function allows to plot the curve fit that is used to calculate the LOB and LOD with functions nonlinear_quantlim() and linear_quantlim(). The function outputs for each calibration curve, two pdf files each containg one plot. On the first, designated by *_overall.pdf, the entire concentration range is plotted. On the second plot, designated by *_zoom.pdf,, the concentration range between 0 and xlim_plot (if specified in the argument of the function) is plotted. When no xlim_plot value is specified, the region close to LOB and LOD is automatically plotted.

Usage

```
plot_quantlim(spikeindata, quantlim_out, alpha, dir_output, xlim_plot)
```

Arguments

`spikeindata`	Data frame that contains the experimental spiked in data. This data frame should be identical to that used as input by function functions nonlinear_quantlim() or linear_quantlim(). The data frame has to contain the following columns : CONCENTRATION, INTENSITY (both of which are measurements from the spiked in experiment) and NAME which designates the name of the assay (e.g. the name of the peptide or protein)
`quantlim_out`	Data frame that was output by functions nonlinear_quantlim() or linear_quantlim(). It has to contain at least the following columns: i) CONCENTRATION: Concentration values at which the value of the fit is calculated ii) MEAN: The value of the curve fit iii) LOW: The value of the lower bound of the 95% prediction interval iv) UP: The value of the upper bound of the 95% prediction interval v) LOB: The value of the LOB (one column with identical values) vi) LOD: The value of the LOD (one column with identical values) vii) NAME: The name of the assay (identical to that provided in the input) viii) METHOD which is LINEAR or NONLINEAR
`alpha`	Probability level to estimate the LOB/LOD
`dir_output`	String containg the path of the directly where the pdf files of the plots should be output.
`xlim_plot`	Optional argument containing the maximum xaxis value of the zoom plot. When no value is specified, a suitable value close to LOD is automatically chosen.

Details

- spikeindata : Each line of the data frame contains one measurement from the spiked-in experiment. Multiple different INTENSITY values for the same CONCENTRATION are assumed to correspond to different replicates. Blank Samples are characterized by CONCENTRATION = 0.

Value

- Data frame where each line corresponds to a unique concentration value at which the value of the fit and prediction interval are evaluated. More unique concentrations values than in the input data frame are used to increase the accuracy of the LOB/D calculations.

Warning

This plotting function should ideally be used every time nonlinear_quantlim() or linear_quantlim() are called to visually ensure that the fits and data are accurate.

Author(s)

Cyril Galitzine, Olga Vitek.

Maintainer: Cyril Galitzine (<cyrildgg@gmail.com>), Meena Choi (<mnchoi67@gmail.com>)

References

C. Galitzine et al. "Nonlinear regression improves accuracy of characterization of multiplexed mass spectrometric assays" *Mol Cell Proteomics*, doi:10.1074/mcp.RA117.000322, 2018.

Examples

```
# Consider data from a spiked-in contained in an example dataset. This dataset contains
# a significant threshold at low concentrations that is not well captured by a linear fit.

head(SpikeInDataNonLinear)

## Not run:
#Call function
nonlinear_quantlim_out <- nonlinear_quantlim(SpikeInDataNonLinear, alpha = 0.05)

plot_quantlim(spikeindata = SpikeInDataLinear, quantlim_out  = nonlinear_quantlim_out,
dir_output =  getwd(), alpha = 0.05)

## End(Not run)
```

ProgenesistoMSstatsFormat

Generate MSstats required input format for Progenesis output

Description

Convert Progenesis output into the required input format for MSstats.

Usage

```
ProgenesistoMSstatsFormat(input,
      annotation,
      useUniquePeptide=TRUE,
      summaryforMultipleRows=max,
      fewMeasurements="remove",
      removeOxidationMpeptides=FALSE,
      removeProtein_with1Peptide=FALSE)
```

Arguments

input	name of Progenesis output, which is wide-format. 'Accession', 'Sequence', 'Modification', 'Charge' and one column for each run are required.
annotation	name of 'annotation.txt' or 'annotation.csv' data which includes Condition, BioReplicate, Run information. It will be matched with the column name of input for MS runs.
useUniquePeptide	TRUE(default) removes peptides that are assigned for more than one proteins. We assume to use unique peptide for each protein.
summaryforMultipleRows	max(default) or sum - when there are multiple measurements for certain feature and certain run, use highest or sum of multiple intensities.
fewMeasurements	'remove'(default) will remove the features that have 1 or 2 measurements across runs.
removeOxidationMpeptides	TRUE will remove the modified peptides including 'Oxidation (M)' sequence. FALSE is default.
removeProtein_with1Peptide	TRUE will remove the proteins which have only 1 peptide and charge. FALSE is default.

Value

data.frame with the required format of MSstats.

Author(s)

Meena Choi, Olga Vitek.

Maintainer: Meena Choi (<mnchoi67@gmail.com>)

Examples

```
# Please check section 4.4.
# Suggested workflow with Progenesis output for DDA in MSstats user manual.
# Output of ProgenesistoMSstatsFormat function
# should have the same 10 columns as an example dataset.

head(DDARawData)
```

---

quantification *Protein sample quantification or group quantification*

Description

Model-based quantification for each condition or for each biological samples per protein in a targeted Selected Reaction Monitoring (SRM), Data-Dependent Acquisition (DDA or shotgun), and Data-Independent Acquisition (DIA or SWATH-MS) experiment. Quantification takes the processed data set by dataProcess as input and automatically generate the quantification results (data.frame) with long or matrix format.

Usage

```
quantification(data, type="Sample", format="matrix")
```

Arguments

data	name of the (processed) data set.
type	choice of quantification. "Sample" or "Group" for protein sample quantification or group quantification.
format	choice of returned format. "long" for long format which has the columns named Protein, Condition, LonIntensities (and BioReplicate if it is subject quantification), NumFeature for number of transitions for a protein, and NumPeaks for number of observed peak intensities for a protein. "matrix" for data matrix format which has the rows for Protein and the columns, which are Groups(or Conditions) for group quantification or the combinations of BioReplicate and Condition (labeled by "BioReplicate"_"Condition") for sample quantification. Default is "matrix"

Details

- Sample quantification : individual biological sample quantification for each protein. The label of each biological sample is a combination of the corresponding group and the sample ID. If there are no technical replicates or experimental replicates per sample, sample quantification is the same as run summarization from dataProcess. If there are technical replicates or experimental replicates, sample quantification is median among run quantification corresponding MS runs.
- Group quantification : quantification for individual group or individual condition per protein. It is median among sample quantification.
- The quantification for endogenous samples is based on run summarization from subplot model, with TMP robust estimation.

  The input of this function is the quantitative data from function (dataProcess).

Value

data.frame as described in details.

Author(s)

Meena Choi, Olga Vitek.

Maintainer: Meena Choi (<mnchoi67@gmail.com>)

References

Meena Choi, Ching-Yun Chang, Timothy Clough, Daniel Broudy, Trevor Killeen, Brendan MacLean and Olga Vitek. "MSstats: an R package for statistical analysis of quantitative mass spectrometry-based proteomic experiments" *Bioinformatics*, 30(17):2524-2526, 2014.

Ching-Yun Chang, Paola Picotti, Ruth Huttenhain, Viola Heinzelmann-Schwarz, Marko Jovanovic, Ruedi Aebersold, Olga Vitek. "Protein significance analysis in selected reaction monitoring (SRM) measurements." *Molecular & Cellular Proteomics*, 11:M111.014662, 2012.

Timothy Clough, Safia Thaminy, Susanne Ragg, Ruedi Aebersold, Olga Vitek. "Statistical protein quantification and significance analysis in label-free LC-M experiments with complex designs" *BMC Bioinformatics*, 13:S16, 2012.

Examples

```
# Consider quantitative data (i.e. QuantData) from a yeast study with ten time points of
# interests, three biological replicates, and no technical replicates which is
# a time-course experiment.
# Sample quantification shows model-based estimation of protein abundance in each biological
# replicate within each time point.
# Group quantification shows model-based estimation of protein abundance in each time point.

QuantData<-dataProcess(SRMRawData)
head(QuantData$ProcessedData)

# Sample quantification

sampleQuant<-quantification(QuantData)
head(sampleQuant)

# Group quantification

groupQuant<-quantification(QuantData, type="Group")
head(groupQuant)
```

SkylinetoMSstatsFormat

Generate MSstats required input format for Skyline output

Description

Preprocess MSstats input report from Skyline and convert into the required input format for MSstats.

Usage

```
SkylinetoMSstatsFormat(input,
      annotation = NULL,
      removeiRT = TRUE,
      filter_with_Qvalue = TRUE,
      qvalue_cutoff = 0.01,
      useUniquePeptide = TRUE,
      fewMeasurements="remove",
      removeOxidationMpeptides = FALSE,
      removeProtein_with1Feature = FALSE)
```

Arguments

input	name of MSstats input report from Skyline, which includes feature-level data.
annotation	name of 'annotation.txt' data which includes Condition, BioReplicate, Run. If annotation is already complete in Skyline, use annotation=NULL (default). It will use the annotation information from input.
removeiRT	TRUE(default) will remove the proteins or peptides which are labeld 'iRT' in 'StandardType' column. FALSE will keep them.

`filter_with_Qvalue`
: TRUE(default) will filter out the intensities that have greater than qvalue_cutoff in DetectionQValue column. Those intensities will be replaced with zero and will be considered as censored missing values for imputation purpose.

`qvalue_cutoff`
: Cutoff for DetectionQValue. default is 0.01.

`useUniquePeptide`
: TRUE(default) removes peptides that are assigned for more than one proteins. We assume to use unique peptide for each protein.

`fewMeasurements`
: 'remove' (default) will remove the features that have 1 or 2 measurements across runs.

`removeOxidationMpeptides`
: TRUE will remove the peptides including 'oxidation (M)' in modification. FALSE is default.

`removeProtein_with1Feature`
: TRUE will remove the proteins which have only 1 peptide and charge. FALSE is default.

Value

data.frame with the required format of MSstats.

Author(s)

Meena Choi, Olga Vitek.

Maintainer: Meena Choi (<mnchoi67@gmail.com>)

Examples

```
# Please check section 4.2. Suggested workflow with Skyline output for DDA in MSstats user manual.
# Output of SkylinetoMSstatsFormat function should have the same 10 columns as an example dataset.

head(DDARawData)
```

SpectronauttoMSstatsFormat

Generate MSstats required input format for Spectronaut output

---

Description

Convert Spectronaut output into the required input format for MSstats.

Usage

```
SpectronauttoMSstatsFormat(input,
      annotation = NULL,
      intensity = 'PeakArea',
      filter_with_Qvalue = TRUE,
      qvalue_cutoff = 0.01,
      useUniquePeptide = TRUE,
      fewMeasurements="remove",
      removeProtein_with1Feature = FALSE,
      summaryforMultipleRows=max)
```

Arguments

input	name of Spectronaut output, which is long-format. ProteinName, PeptideSequence, PrecursorCharge, FragmentIon, ProductCharge, IsotopeLabelType, Condition, BioReplicate, Run, Intensity, F.ExcludedFromQuantification are required. Rows with F.ExcludedFromQuantification=True will be removed.
annotation	name of 'annotation.txt' data which includes Condition, BioReplicate, Run. If annotation is already complete in Spectronaut, use annotation=NULL (default). It will use the annotation information from input.
intensity	'PeakArea'(default) uses not normalized peak area. 'NormalizedPeakArea' uses peak area normalized by Spectronaut.
filter_with_Qvalue	TRUE(default) will filter out the intensities that have greater than qvalue_cutoff in EG.Qvalue column. Those intensities will be replaced with zero and will be considered as censored missing values for imputation purpose.
qvalue_cutoff	Cutoff for EG.Qvalue. default is 0.01.
useUniquePeptide	TRUE(default) removes peptides that are assigned for more than one proteins. We assume to use unique peptide for each protein.
fewMeasurements	'remove' (default) will remove the features that have 1 or 2 measurements across runs.
removeProtein_with1Feature	TRUE will remove the proteins which have only 1 feature, which is the combination of peptide, precursor charge, fragment and charge. FALSE is default.
summaryforMultipleRows	max(default) or sum - when there are multiple measurements for certain feature and certain run, use highest or sum of multiple intensities.

Value

data.frame with the required format of MSstats.

Author(s)

Meena Choi, Olga Vitek.

Maintainer: Meena Choi (<mnchoi67@gmail.com>)

Examples

```
# Please check section 5.2.
# Suggested workflow with Spectronaut output for DIA in MSstats user manual.
# Output of SpectronauttoMSstatsFormat function
# should have the same 10 columns as an example dataset.

head(DDARawData)
```

SpikeInDataLinear *Example dataset from an MRM spike-in experiment with a linear behavior*

Description

This dataset is part of the CPTAC 7, study 3 (Addona et al., 2009). It corresponds to the spike-in data for peptide AGLCQTFVYGGCR at site 86. This particular data was chosen because it illustrates well a linear response for a spiked in experiment. The data is composed of 4 replicates at 10 different concentrations (including a blank sample with concentration 0).

Usage

```
SpikeInDataLinear
```

Format

data.frame

Details

The intensity reported is the sum of the intensity of all the different fragments of the peptide. Only the peptide being spiked (light peptide) is contained in the example data set. The intensity was normalized using the corresponding heavy peptide in log space such that intensity of the heavy remains constant for all concentrations and all replicates. The intensity was rescaled following the method described in Addona et al., 2009. The concentration and Intensity are both in units of fmol/uL.

Value

data.frame as described in details.

Author(s)

Cyril Galitzine, Olga Vitek.

Maintainer: Cyril Galitzine (<cyrildgg@gmail.com>), Meena Choi (<mnchoi67@gmail.com>)

References

T.A. Addonna et al. "Multi-site assessment of the precision and reproducibility of multiple reaction monitoring-based measurements of proteins in plasma." *Nat Biotechnol.* 2009 Jul;27(7):633-41

Examples

```
head(SpikeInDataLinear)
```

SpikeInDataNonLinear *Example dataset from an MRM spike-in experiment with a nonlinear behavior*

Description

This dataset is part of the CPTAC 7, study 3 (Addona et al., 2009). It corresponds to the spike-in data for peptide ESDTSYVSLK at site 19. This particular data was chosen because of the concentration threshold that is present at low concentrations that warrant the use of a nonlinear method. The data is composed of 4 replicates at 10 different concentrations (including a blank sample with concentration 0).

Usage

```
SpikeInDataNonLinear
```

Format

data.frame

Details

The intensity reported is the sum of the intensity of all the different fragments of the peptide. Only the peptide being spiked (light peptide) is contained in the example data set. The intensity was normalized using the corresponding heavy peptide in log space such that intensity of the heavy remains constant for all concentrations and all replicates. The intensity was rescaled following the method described in Addona et al., 2009. The concentration and Intensity are both in units of fmol/uL.

Value

data.frame as described in details.

Author(s)

Cyril Galitzine, Olga Vitek.

Maintainer: Cyril Galitzine (<cyrildgg@gmail.com>), Meena Choi (<mnchoi67@gmail.com>)

References

T.A. Addonna et al. "Multi-site assessment of the precision and reproducibility of multiple reaction monitoring-based measurements of proteins in plasma." *Nat Biotechnol.* 2009 Jul;27(7):633-41

Examples

```
head(SpikeInDataNonLinear)
```

SRMRawData *Example dataset from a SRM experiment with stable isotope labeled reference of a time course yeast study*

Description

This is a partial data set obtained from a published study (Picotti, et. al, 2009). The experiment targeted 45 proteins in the glycolysis/gluconeogenesis/TCA cycle/glyoxylate cycle network, which spans the range of protein abundance from less than 128 to 10E6 copies per cell. Three biological replicates were analyzed at ten time points (T1-T10), while yeasts transited through exponential growth in a glucose-rich medium (T1-T4), diauxic shift (T5-T6), post-diauxic phase (T7-T9), and stationary phase (T10). Prior to trypsinization, the samples were mixed with an equal amount of proteins from the same N15-labeled yeast sample, which was used as a reference. Each sample was profiled in a single mass spectrometry run, where each protein was represented by up to two peptides and each peptide by up to three transitions. The goal of this study is to detect significantly change in protein abundance across time points. Transcriptional activity under the same experimental conditions has been previously investigated by (DeRisi et. al., 1997). Genes coding for 29 of the proteins are differentially expressed between conditions similar to those represented by T7 and T1 and could be treated as external sources to validate the proteomics analysis. In this exampled data set, two of the targeted proteins are selected and validated with gene expression study: Protein IDHC (gene name IDP2) is differentially expressed in time point 1 and time point 7, whereas, Protein PMG2 (gene name GPM2) is not. The protein names are based on Swiss Prot Name.

Usage

```
SRMRawData
```

Format

data.frame

Details

The raw data (input data for MSstats) is required to contain variable of ProteinName, PeptideSequence, PrecursorCharge, FragmentIon, ProductCharge, IsotopeLabelType, Condition, BioReplicate, Run, Intensity. The variable names should be fixed.

If the information of one or more columns is not available for the original raw data, please retain the column variables and type in fixed value. For example, the original raw data does not contain the information of ProductCharge, we retain the column ProductCharge and type in NA for all transitions in RawData.

The column BioReplicate should label with unique patient ID (i.e., same patients should label with the same ID).

Variable Intensity is required to be original signal without any log transformation and can be specified as the peak of height or the peak of area under curve.

Value

data.frame with the required format of MSstats.

Author(s)

Meena Choi, Olga Vitek.

Maintainer: Meena Choi (<mnchoi67@gmail.com>)

References

Ching-Yun Chang, Paola Picotti, Ruth Huttenhain, Viola Heinzelmann-Schwarz, Marko Jovanovic, Ruedi Aebersold, Olga Vitek. Protein significance analysis in selected reaction monitoring (SRM) measurements. *Molecular & Cellular Proteomics*, 11:M111.014662, 2012.

Examples

```
head(SRMRawData)
```

Index