Package: MSstatsTMT (via r-universe)

November 24, 2026

Title Protein Significance Analysis in shotgun mass spectrometry-based proteomic experiments with tandem mass tag (TMT) labeling

Version 2.22.0

Date 2026-08-31

Description The package provides statistical tools for detecting differentially abundant proteins in shotgun mass spectrometry-based proteomic experiments with tandem mass tag (TMT) labeling. It provides multiple functionalities, including aata visualization, protein quantification and normalization, and statistical modeling and inference. Furthermore, it is inter-operable with other data processing tools, such as Proteome Discoverer, MaxQuant, OpenMS and SpectroMine.

License Artistic-2.0

Depends R (>= 4.2)

Imports limma, lme4, lmerTest, methods, data.table, stats, utils, ggplot2, grDevices, graphics, MSstats, MSstatsConvert, checkmate, plotly, htmltools

Suggests BiocStyle, knitr, rmarkdown, testthat

VignetteBuilder knitr

biocViews ImmunoOncology, MassSpectrometry, Proteomics, Software

Encoding UTF-8

LazyData true

URL http://msstats.org/msstatstmt/

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

Config/roxygen2/version 8.1.0

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libssl-dev

Repository Bioconductor 3.24

Date/Publication 2026-11-23 22:41:38 UTC

RemoteUrl https://github.com/bioc/MSstatsTMT

RemoteRef RELEASE_3_24

RemoteSha f19cc088e979b3b8d4b657de60778f9266055466

Contents

dataProcessPlotsTMT 2
designSampleSizeTMT 4
groupComparisonTMT 6
input.pd 8
proteinSummarization 9

Index **11**

`dataProcessPlotsTMT` *Visualization for explanatory data analysis - TMT experiment*

Description

To illustrate the quantitative data and quality control of MS runs, dataProcessPlotsTMT takes the quantitative data and summarized data from function ‘proteinSummarization‘ as input and generate two 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 and channels.

Usage

```
dataProcessPlotsTMT(
  data,
  type,
  featureName = "Transition",
  ylimUp = FALSE,
  ylimDown = FALSE,
  x.axis.size = 10,
  y.axis.size = 10,
  text.size = 2,
  text.angle = 90,
  legend.size = 7,
  dot.size.profile = 2,
  ncol.guide = 5,
  width = 10,
  height = 10,
  which.Protein = "all",
  originalPlot = TRUE,
  summaryPlot = TRUE,
  address = "",
  isPlotly = FALSE
)
```

Arguments

data	the output of proteinSummarization function. It is a list with data frames 'FeatureLevelData' and 'ProteinLevelData'
type	choice of visualization. "ProfilePlot" represents profile plot of log intensities across MS runs. "QCPlot" represents box plots of log intensities across channels and MS runs.
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. FALSE(Default) for Profile Plot and QC Plot uses the upper limit as rounded off maximum of log2(intensities) after normalization + 3..
ylimUp	upper limit for y-axis in the log scale.
ylimDown	lower limit for y-axis in the log scale. FALSE(Default) for Profile Plot and QC Plot uses 0..
x.axis.size	size of x-axis labeling for "Run" and "channel in Profile Plot and QC Plot.
y.axis.size	size of y-axis labels. Default is 10.
text.size	size of labels represented each condition at the top of Profile plot and QC plot. Default is 4.
text.angle	angle of labels represented each condition at the top of Profile plot and QC plot. Default is 0.
legend.size	size of legend above Profile plot. Default is 7.
dot.size.profile	size of dots in Profile plot. Default is 2.
ncol.guide	number of columns for legends at the top of plot. Default is 5.
width	width of the saved pdf file. Default is 10.
height	height of the saved pdf file. Default is 10.
which.Protein	Protein list to draw plots. List can be names of Proteins or order numbers of Proteins. 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, without normalization.
summaryPlot	TRUE(default) draws profile plots with protein summarization for each channel and MS run.
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". 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.
isPlotly	Parameter to use Plotly or ggplot2. If set to TRUE, MSstats will save Plotly plots as HTML files. If set to FALSE MSstats will save ggplot2 plots as PDF files

Value

plot or pdf

Examples

```
data(input.pd)
quant.msstats = proteinSummarization(input.pd,
                                     method="msstats",
                                     global_norm=TRUE,
                                     reference_norm=TRUE)

## Profile plot
dataProcessPlotsTMT(data=quant.msstats,
                    type='ProfilePlot',
                    width = 21,
                    height = 7)

## NottoRun: QC plot
# dataProcessPlotsTMT(data=quant.msstats,
                     # type='QCPlot',
                     # width = 21,
                     # height = 7)
```

---

designSampleSizeTMT *Planning future experimental designs of Tandem Mass Tag (TMT) experiments acquired with Data-Dependent Acquisition (DDA or shotgun)*

Description

Calculate sample size for future experiments of a TMT experiment based on intensity-based linear model. Two options of the calculation: (1) number of biological replicates per condition, (2) power.

Usage

```
designSampleSizeTMT(
  data,
  desiredFC,
  FDR = 0.05,
  numSample = TRUE,
  power = 0.9,
  use_log_file = TRUE,
  append = FALSE,
  verbose = TRUE,
  log_file_path = NULL
)
```

Arguments

data	'FittedModel' in testing output from function groupComparisonTMT.
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 rate. 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
use_log_file	logical. If TRUE, information about data processing will be saved to a file.
append	logical. If TRUE, information about data processing will be added to an existing log file.
verbose	logical. If TRUE, information about data processing wil be printed to the console.
log_file_path	character. Path to a file to which information about data processing will be saved. If not provided, such a file will be created automatically. If 'append = TRUE', has to be a valid path to a file.

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

Value

data.frame - sample size calculation results including varibles: desiredFC, numSample, FDR, and power.

Examples

```
data(input.pd)
# use protein.summarization() to get protein abundance data
quant.pd.msstats = proteinSummarization(input.pd,
                                        method="msstats",
                                        global_norm=TRUE,
                                        reference_norm=TRUE)

test.pairwise = groupComparisonTMT(quant.pd.msstats, save_fitted_models = TRUE)
head(test.pairwise$ComparisonResult)

## Calculate sample size for future experiments:
#(1) Minimal number of biological replicates per condition
```

```
designSampleSizeTMT(data=test.pairwise$FittedModel, numSample=TRUE,
                desiredFC=c(1.25,1.75), FDR=0.05, power=0.8)
#(2) Power calculation
designSampleSizeTMT(data=test.pairwise$FittedModel, numSample=2,
                desiredFC=c(1.25,1.75), FDR=0.05, power=TRUE)
```

`groupComparisonTMT` *Finding differentially abundant proteins across conditions in TMT experiment*

Description

Tests for significant changes in protein abundance across conditions based on a family of linear mixed-effects models in TMT experiment. Experimental design of case-control study (patients are not repeatedly measured) is automatically determined based on proper statistical model.

Usage

```
groupComparisonTMT(
  data,
  contrast.matrix = "pairwise",
  moderated = FALSE,
  adj.method = "BH",
  remove_norm_channel = TRUE,
  remove_empty_channel = TRUE,
  save_fitted_models = FALSE,
  use_log_file = TRUE,
  append = FALSE,
  verbose = TRUE,
  log_file_path = NULL
)
```

Arguments

`data`	the output of `proteinSummarization` function. It is a list with data frames 'FeatureLevelData' and 'ProteinLevelData'
`contrast.matrix`	Comparison between conditions of interests. 1) default is "pairwise", which compare all possible pairs between two conditions. 2) Otherwise, users can specify the comparisons 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.
`moderated`	TRUE will moderate t statistic; FALSE (default) uses ordinary t statistic.
`adj.method`	adjusted method for multiple comparison. "BH" is default.
`remove_norm_channel`	TRUE(default) removes "Norm" channels from protein level data.

remove_empty_channel
: TRUE(default) removes "Empty" channels from protein level data.

save_fitted_models
: logical, if TRUE, fitted models will be added to

use_log_file
: logical. If TRUE, information about data processing will be saved to a file.

append
: logical. If TRUE, information about data processing will be added to an existing log file.

verbose
: logical. If TRUE, information about data processing wil be printed to the console.

log_file_path
: character. Path to a file to which information about data processing will be saved. If not provided, such a file will be created automatically. If 'append = TRUE', has to be a valid path to a file.

Value

a list that consists of the following elements: (1) ComparisonResult: statistical testing results; (2) FittedModel: the fitted linear models

Examples

```
data(input.pd)
# use protein.summarization() to get protein abundance data
quant.pd.msstats = proteinSummarization(input.pd,
                                        method="msstats",
                                        global_norm=TRUE,
                                        reference_norm=TRUE)

test.pairwise = groupComparisonTMT(quant.pd.msstats, moderated = TRUE)
head(test.pairwise$ComparisonResult)

# Only compare condition 0.125 and 1
levels(quant.pd.msstats$ProteinLevelData$Condition)

# Compare condition 1 and 0.125
comparison=matrix(c(-1,0,0,1),nrow=1)

# Set the nafmes of each row
row.names(comparison)="1-0.125"

# Set the column names
colnames(comparison)= c("0.125", "0.5", "0.667", "1")
test.contrast = groupComparisonTMT(data = quant.pd.msstats,
contrast.matrix = comparison,
moderated = TRUE)
head(test.contrast$ComparisonResult)
```

`input.pd` *Example dataset in MSstatsTMT format.*

Description

It should include the required columns as below.

Usage

```
input.pd
```

Format

A data frame with 20110 rows and 11 variables.

Details

- ProteinName : Protein ID
- PeptideSequence : peptide sequence
- Charge : peptide charge
- PSM : peptide ion and spectra match
- Channel : Labeling information (126, ... 131)
- Condition : Condition (ex. Healthy, Cancer, Time0)
- BioReplicate : Unique ID for biological subject.
- Run : MS run ID
- Mixture : Unique ID for TMT mixture.
- TechRepMixture : Unique ID for technical replicate of one TMT mixture.
- Intensity: Protein Abundance

Examples

```
head(input.pd)
```

proteinSummarization *Summarizing peptide level quantification to protein level quantification*

Description

We assume missing values are censored and then impute the missing values. Protein-level summarization from peptide level quantification are performed. After all, global median normalization on peptide level data and normalization between MS runs using reference channels will be implemented.

Usage

```
proteinSummarization(
  data,
  method = "msstats",
  global_norm = TRUE,
  reference_norm = TRUE,
  remove_norm_channel = TRUE,
  remove_empty_channel = TRUE,
  MBimpute = TRUE,
  maxQuantileforCensored = NULL,
  use_log_file = TRUE,
  append = FALSE,
  verbose = TRUE,
  log_file_path = NULL,
  msstats_log_path = NULL
)
```

Arguments

data	Name of the output of PDtoMSstatsTMTFormat function or peptide-level quantified data from other tools. It should have columns ProteinName, PeptideSequence, Charge, PSM, Mixture, TechRepMixture, Run, Channel, Condition, BioReplicate, Intensity
method	Four different summarization methods to protein-level can be performed : "msstats"(default), "MedianPolish", "Median", "LogSum".
global_norm	Global median normalization on peptide level data (equalizing the medians across all the channels and MS runs). Default is TRUE. It will be performed before protein-level summarization.
reference_norm	Reference channel based normalization between MS runs on protein level data. TRUE(default) needs at least one reference channel in each MS run, annotated by 'Norm' in Condtion column. It will be performed after protein-level summarization. FALSE will not perform this normalization step. If data only has one run, then reference_norm=FALSE.

`remove_norm_channel`
: TRUE(default) removes 'Norm' channels from protein level data.

`remove_empty_channel`
: TRUE(default) removes 'Empty' channels from protein level data.

`MBimpute`
: only for method="msstats". TRUE (default) imputes missing values by Accelated failure model. FALSE uses minimum value to impute the missing value for each peptide precursor ion.

`maxQuantileforCensored`
: We assume missing values are censored. maxQuantileforCensored is Maximum quantile for deciding censored missing value, for instance, 0.999. Default is Null.

`use_log_file`
: logical. If TRUE, information about data processing will be saved to a file.

`append`
: logical. If TRUE, information about data processing will be added to an existing log file.

`verbose`
: logical. If TRUE, information about data processing wil be printed to the console.

`log_file_path`
: character. Path to a file to which information about data processing will be saved. If not provided, such a file will be created automatically. If 'append = TRUE', has to be a valid path to a file.

`msstats_log_path`
: path to a MSstats log file

Value

list that consists of two data.frames with feature-level (FeatureLevelData) and protein-level data (ProteinLevelData)

Examples

```
data(input.pd)
quant.pd.msstats <- proteinSummarization(input.pd,
                                         method = "msstats",
                                         global_norm = TRUE,
                                         reference_norm = TRUE)
head(quant.pd.msstats$ProteinLevelData)
```

Index

∗ **datasets**
input.pd, 8

dataProcessPlotsTMT, 2
designSampleSizeTMT, 4

groupComparisonTMT, 6

input.pd, 8

proteinSummarization, *3*, *6*, 9