Package 'MSstatsPTM'

November 28, 2021

Type Package

Title Statistical Characterization of Post-translational Modifications

Version 1.4.1

Date 2021-11-08

Description MSstatsPTM provides general statistical methods for quantitative characterization of post-translational modifications (PTMs). Supports DDA, DIA, and tandem mass tag (TMT) labeling. Typically, the analysis involves the quantification of PTM sites (i.e., modified residues) and their corresponding proteins, as well as the integration of the quantification results. MSstatsPTM provides functions for summarization, estimation of PTM site abundance, and detection of changes in PTMs across experimental conditions.

License Artistic-2.0

Depends R (>= 4.0)

Imports dplyr, gridExtra, stringr, stats, ggplot2, grDevices, MSstatsTMT, MSstatsConvert, MSstats, data.table, Rcpp, Biostrings, checkmate, ggrepel

Suggests BiocStyle, knitr, rmarkdown, tinytest, covr

LazyData true

LinkingTo Rcpp

VignetteBuilder knitr

biocViews ImmunoOncology, MassSpectrometry, Proteomics, Software, DifferentialExpression, OneChannel, TwoChannel, Normalization, QualityControl

BugReports https://github.com/Vitek-Lab/MSstatsPTM/issues

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

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

git_branch RELEASE_3_14

git_last_commit 7f957ad

git_last_commit_date 2021-11-08

Date/Publication 2021-11-28

Author Devon Kohler [aut, cre],
Tsung-Heng Tsai [aut],
Ting Huang [aut],
Mateusz Staniak [aut],
Meena Choi [aut],
Olga Vitek [aut]

Maintainer Devon Kohler <kohler.d@northeastern.edu>

R topics documented:

`annotSite` *Annotate modification site*

Description

`annotSite` annotates modified sites as their residues and locations.

Usage

```
annotSite(aaIndex, residue, lenIndex = NULL)
```

Arguments

aaIndex	An integer vector. Location of the sites.
residue	A string vector. Amino acid residue.
lenIndex	An integer. Default is NULL

Value

A string.

Examples

```
annotSite(10, "K")
annotSite(10, "K", 3L)
```

---

dataProcessPlotsPTM *Visualization for explanatory data analysis*

Description

To illustrate the quantitative data and quality control of MS runs, dataProcessPlotsPTM takes the quantitative data from dataSummarizationPTM or dataSummarizationPTM_TMT to plot the following : (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.

Usage

```
dataProcessPlotsPTM(
  data,
  type = "PROFILEPLOT",
  ylimUp = FALSE,
  ylimDown = FALSE,
  x.axis.size = 10,
  y.axis.size = 10,
  text.size = 4,
  text.angle = 90,
  legend.size = 7,
  dot.size.profile = 2,
  ncol.guide = 5,
  width = 10,
  height = 12,
  ptm.title = "All PTMs",
  protein.title = "All Proteins",
  which.PTM = "all",
  which.Protein = NULL,
```

```
  originalPlot = TRUE,
  summaryPlot = TRUE,
  address = ""
)
```

Arguments

data	name of the list with PTM and (optionally) Protein data, which can be the output of the MSstatsPTM dataSummarizationPTM or dataSummarizationPTM_TMT functions.
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.
ylimUp	upper limit for y-axis in the log scale. FALSE(Default) for Profile Plot and QC Plot uses the upper limit as rounded off maximum of log2(intensities) after normalization + 3..
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.
ptm.title	title of overall PTM QC plot
protein.title	title of overall Protein QC plot
which.PTM	PTM list to draw plots. List can be names of PTMs or order numbers of PTMs. Default is "all", which generates all plots for each protein. For QC plot, "allonly" will generate one QC plot with all proteins.
which.Protein	List of proteins to plot. Will plot all PTMs associated with listed Proteins. Default is NULL which will default to which.PTM.
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.

Value

plot or pdf

Examples

```
# QCPlot
dataProcessPlotsPTM(summary.data,
                    type = 'QCPLOT',
                    which.PTM = "allonly",
                    address = FALSE)

#ProfilePlot
dataProcessPlotsPTM(summary.data,
                    type = 'PROFILEPLOT',
                    which.PTM = "Q9UQ80_K376",
                    address = FALSE)
```

---

dataSummarizationPTM *Process MS PTM and global protein data*

Description

Utilizes functionality from MSstats to clean, summarize, and normalize PTM and protein level data. Imputes missing values, protein and PTM level summarization from peptide level quantification. Applies global median normalization on peptide level data and normalizes between runs.

Usage

```
dataSummarizationPTM(
  data,
  logTrans = 2,
  normalization = "equalizeMedians",
  normalization.PTM = "equalizeMedians",
  nameStandards = NULL,
  nameStandards.PTM = NULL,
  featureSubset = "all",
  featureSubset.PTM = "all",
  remove_uninformative_feature_outlier = FALSE,
  remove_uninformative_feature_outlier.PTM = FALSE,
  min_feature_count = 2,
  min_feature_count.PTM = 1,
  n_top_feature = 3,
```

```
  n_top_feature.PTM = 3,
  summaryMethod = "TMP",
  equalFeatureVar = TRUE,
  censoredInt = "NA",
  MBimpute = TRUE,
  MBimpute.PTM = TRUE,
  remove50missing = FALSE,
  fix_missing = NULL,
  maxQuantileforCensored = 0.999,
  use_log_file = TRUE,
  append = TRUE,
  verbose = TRUE,
  log_file_path = NULL,
  base = "MSstatsPTM_log_"
)
```

Arguments

data	name of the list with PTM and (optionally) Protein data.tables, which can be the output of the MSstatsPTM converter functions
logTrans	logarithm transformation with base 2(default) or 10
normalization	normalization for the protein level dataset, 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
normalization.PTM	normalization for PTM level dataset. Default is "equalizeMedians" Can be adjusted to any of the options described above.
nameStandards	vector of global standard peptide names for protein dataset. only for normalization with global standard peptides.
nameStandards.PTM	Same as above for PTM dataset. "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" flags uninformative feature and outliers
featureSubset	"all" (default) uses all features that the data set has. "top3" uses top 3 features which have highest average of log-intensity across runs. "topN" uses top N features which has highest average of log-intensity across runs. It needs the input for n_top_feature option. "highQuality" flags uninformative feature and outliers.
featureSubset.PTM	For PTM dataset only. Options same as above.

`remove_uninformative_feature_outlier`
: For protein dataset only. It only works after users used featureSubset="highQuality" in dataProcess. TRUE allows to remove 1) the features are flagged in the column, feature_quality="Uninformative" which are features with bad quality, 2) outliers that are flagged in the column, is_outlier=TRUE, for run-level summarization. FALSE (default) uses all features and intensities for run-level summarization.

`remove_uninformative_feature_outlier.PTM`
: For PTM dataset only. Options same as above.

`min_feature_count`
: optional. Only required if featureSubset = "highQuality". Defines a minimum number of informative features a protein needs to be considered in the feature selection algorithm.

`min_feature_count.PTM`
: For PTM dataset only. Options the same as above. Default is 1 due to low average feature count for PTMs.

`n_top_feature`
: For protein dataset only. The number of top features for featureSubset='topN'. Default is 3, which means to use top 3 features.

`n_top_feature.PTM`
: For PTM dataset only. Options same as above.

`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.

`MBimpute`
: For protein dataset only. 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.

`MBimpute.PTM`
: For PTM dataset only. Options same as above.

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

`fix_missing`
: Default is Null. Optional, same as the 'fix_missing' parameter in MSstatsConvert::MSstatsBalancedDesign function

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

`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 will 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.
base	start of the file name.

Value

list of summarized PTM and Protein results. These results contain the reformatted input to the summarization function, as well as run-level summarization results.

Examples

```
head(raw.input$PTM)
head(raw.input$PROTEIN)

quant.lf.msstatsptm <- dataSummarizationPTM(raw.input, verbose = FALSE)
head(quant.lf.msstatsptm$PTM$ProteinLevelData)
```

---

dataSummarizationPTM_TMT

Process MS PTM and global protein data produced via tandem mass tag labeling

---

Description

Utilizes functionality from MSstatsTMT to clean, summarize, and normalize PTM and protein level data. Imputes missing values, protein and PTM level summarization from peptide level quantification. Applies global median normalization on peptide level data and normalizes between runs.

Usage

```
dataSummarizationPTM_TMT(
  data,
  method = "msstats",
  global_norm = TRUE,
  global_norm.PTM = TRUE,
  reference_norm = TRUE,
  reference_norm.PTM = TRUE,
  remove_norm_channel = TRUE,
  remove_empty_channel = TRUE,
  MBimpute = TRUE,
  MBimpute.PTM = TRUE,
  maxQuantileforCensored = NULL,
```

```
  use_log_file = TRUE,
  append = FALSE,
  verbose = TRUE,
  log_file_path = NULL
)
```

Arguments

data	Name of the output of MSstatsPTM converter function or peptide-level quantified data from other tools. It should be a list containing one or two data tables, named PTM and PROTEIN for modified and unmodified datasets. The list must at least contain the PTM dataset. The data 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 for unmodified peptide level data (equalizing the medians across all the channels and MS runs). Default is TRUE. It will be performed before protein-level summarization.
global_norm.PTM	Same as above for modified peptide level data. Default is TRUE.
reference_norm	Reference channel based normalization between MS runs on unmodified 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.
reference_norm.PTM	Same as above for modified peptide level data. Default is TRUE.
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.
MBimpute.PTM	Same as above for modified peptide level data. Default is TRUE
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 will 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

list of two data.tables

Examples

```
head(raw.input.tmt$PTM)
head(raw.input.tmt$PROTEIN)

quant.tmt.msstatsptm <- dataSummarizationPTM_TMT(raw.input.tmt,
                                                 method = "msstats",
                                                 verbose = FALSE)
head(quant.tmt.msstatsptm$PTM$ProteinLevelData)
```

---

groupComparisonPlotsPTM

Visualization for model-based analysis and summarization

Description

To analyze the results of modeling changes in abundance of modified peptides and overall protein, groupComparisonPlotsPTM takes as input the results of the groupComparisonPTM function. It asses the results of three models: unadjusted PTM, adjusted PTM, and overall protein. To asses the results of the model, the following visualizations can be created: (1) VolcanoPlot (specify "VolcanoPlot" in option type), to plot peptides or proteins and their significance for each model. (2) Heatmap (specify "Heatmap" in option type), to evaluate the fold change between conditions and peptides/proteins

Usage

```
groupComparisonPlotsPTM(
  data = data,
  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,
  text.angle = 0,
```

```
  legend.size = 13,
  ProteinName = TRUE,
  colorkey = TRUE,
  numProtein = 50,
  width = 10,
  height = 10,
  which.Comparison = "all",
  which.PTM = "all",
  address = ""
)
```

Arguments

data	name of the list with models, which can be the output of the MSstatsPTM groupComparisonPTM function
type	choice of visualization, one of VolcanoPlot or Heatmap
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. FCcutoff = 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.
text.angle	angle of x-axis labels represented each comparison at the bottom of graph in comparison plot. Default is 0.
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 50.
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$Label) , such as levels(testResultMultiComparisons$Comparisc Default is "all", which generates all plots for each protein.
which.PTM	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". 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

Value

plot or pdf

Examples

```
model.lf.msstatsptm <- groupComparisonPTM(summary.data,
                                    data.type = "LabelFree")
groupComparisonPlotsPTM(data = model.lf.msstatsptm,
                        type = "VolcanoPlot",
                        FCcutoff= 2,
                        logBase.pvalue = 2,
                        address=FALSE)
```

---

groupComparisonPTM *Model PTM and/or protein data and make adjustments if needed*

Description

Takes summarized PTM and protein data from proteinSummarization. If protein data is unavailable, PTM data only can be passed into the function. Including protein data allows for adjusting PTM Fold Change by the change in protein abundance without modification. MSstatsContrastMatrix

Usage

```
groupComparisonPTM(
  data,
  data.type,
  contrast.matrix = "pairwise",
  moderated = FALSE,
  adj.method = "BH",
  log_base = 2,
  use_log_file = TRUE,
  append = FALSE,
  verbose = TRUE,
  log_file_path = NULL,
  base = "MSstatsPTM_log_"
)
```

Arguments

data	list of summarized datasets. Output of MSstatsPTM summarization function dataSummarizationPTM or dataSummarizationPTM_TMT depending on acquisition type.
data.type	string indicating experimental acquisition type. "TMT" is used for TMT labeled experiments. For all other experiments (Label Free/ DDA/ DIA) use "Label-Free".
contrast.matrix	comparison between conditions of interests. Default models full pairwise comparison between all conditions
moderated	For TMT experiments only. TRUE will moderate t statistic; FALSE (default) uses ordinary t statistic. Default is FALSE.
adj.method	For TMT experiemnts only. Adjusted method for multiple comparison. "BH" is default. "BH" is used for all other experiment types
log_base	For non-TMT experiments only. The base of the logarithm used in summarization.
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 will 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.
base	start of the file name.

Value

list of modeling results. Includes PTM, PROTEIN, and ADJUSTED data.tables with their corresponding model results.

Examples

```
model.lf.msstatsptm <- groupComparisonPTM(summary.data,
                                          data.type = "LabelFree",
                                          verbose = FALSE)
```

`locateMod` *Locate modified sites with a peptide*

Description

`locateMod` locates modified sites with a peptide.

Usage

```
locateMod(peptide, aaStart, residueSymbol)
```

Arguments

`peptide`	A string. Peptide sequence.
`aaStart`	An integer. Starting index of the peptide.
`residueSymbol`	A string. Modification residue and denoted symbol.

Value

A string.

Examples

```
locateMod("P*EP*TIDE", 3, "\\*")
```

`locatePTM` *Annotate modified sites with associated peptides*

Description

`PTMlocate` annotates modified sites with associated peptides.

Usage

```
locatePTM(peptide, uniprot, fasta, modResidue, modSymbol, rmConfound = FALSE)
```

Arguments

peptide	A string vector of peptide sequences. The peptide sequence does not include its preceding and following AAs.
uniprot	A string vector of Uniprot identifiers of the peptides' originating proteins. UniProtKB entry isoform sequence is used.
fasta	A data.table with FASTA information. Output of tidyFasta.
modResidue	A string. Modifiable amino acid residues.
modSymbol	A string. Symbol of a modified site.
rmConfound	A logical. TRUE removes confounded unmodified sites, FALSE otherwise. Default is FALSE.

Value

A data frame with three columns: uniprot_iso, peptide, site.

Examples

```
fasta <- tidyFasta(system.file("extdata", "O13297.fasta", package="MSstatsPTM"))
locatePTM("DRVSYIHNDSC*TR", "O13297", fasta, "C", "\\*")
```

MaxQtoMSstatsPTMFormat

Convert output of TMT labeled MaxQuant experiment into MSstatsPTM format

---

Description

Takes as input TMT experiments from MaxQ and converts the data into the format needed for MSstatsPTM. Requires only the modified file from MaxQ (for example Phospho(STY)Sites) and an annotation file for PTM data. To adjust modified peptides for changes in global protein level, unmodified TMT experimental data must also be returned.

Usage

```
MaxQtoMSstatsPTMFormat(
  sites.data,
  annotation,
  evidence = NULL,
  proteinGroups = NULL,
  mod.num = "Single",
  keyword = "phos",
  which.proteinid.ptm = "Protein",
  which.proteinid.protein = "Leading.razor.protein",
  removeMpeptides = FALSE
)
```

Arguments

sites.data	modified peptide output from MaxQuant. For example, a phosphorylation experiment would require the Phospho(STY)Sites.txt file
annotation	data frame which contains column Run, Fraction, TechRepMixture, Mixture, Channel, BioReplicate, Condition.
evidence	for global protein dataset. name of 'evidence.txt' data, which includes feature-level data.
proteinGroups	for global protein dataset, name of 'proteinGroups.txt' data.
mod.num	For modified peptide dataset. The number modifications per peptide to be used. If "Single", only peptides with one modification will be used. Otherwise "Total" can be selected which does not cap the number of modifications per peptide. "Single" is the default. Selecting "Total" may confound the effect of different modifications.
keyword	the sub-name of columns in the sites.data file. For phosphorylation data, this value should be "phos". The default is "phos".
which.proteinid.ptm	For PTM dataset, which column to use for protein name. Use 'Proteins'(default) column for protein name. 'Leading.proteins' or 'Leading.razor.protein' or 'Gene.names' can be used instead to get the protein ID with single protein. However, those can potentially have the shared peptides.
which.proteinid.protein	For Protein dataset, which column to use for protein name. Same options as above.
removeMpeptides	If Oxidation (M) modifications should be removed. Default is TRUE.

Value

a list of two data.tables named 'PTM' and 'PROTEIN' in the format required by MSstatsPTM.

Examples

```
head(raw.input.tmt$PTM)
head(raw.input.tmt$PROTEIN)
```

MSstatsPTM *MSstatsPTM: A package for detecting differencially abundant post translational modifications (PTM) in shotgun mass spectrometry-bsed proteomic experiments.*

Description

A set of tools for detecting differentially abundant PTMs and proteins in shotgun mass spectrometry-based proteomic experiments. The package can handle a variety of acquisition types, including label free, DDA, DIA, and TMT. The package includes tools to convert raw data from different spectral processing tools, summarize feature intensities, and fit a linear mixed effects model. Additionally the package includes functionality to plot a variety of data visualizations.

functions

- `MaxQtoMSstatsPTMFormat` : Generates MSstatsPTM required input format for TMT MaxQuant outputs.
- `ProgenesistoMSstatsPTMFormat` : Generates MSstatsPTM required input format for non-TMT Proteoviz outputs.
- `SpectronauttoMSstatsPTMFormat` : Generates MSstatsPTM required input format for non-TMT Spectronaut outputs.
- `dataSummarizationPTM` : Summarizes PSM level quantification to peptide (modification) and protein level quantification. For use in non-TMT analysis
- `dataSummarizationPTM_TMT` : Summarizes PSM level quantification to peptide (modification) and protein level quantification. For use in TMT analysis.
- `dataProcessPlotsPTM` : Visualization for explanatory data analysis. Specifically gives ability to plot Profile and Quality Control plots.
- `groupComparisonPTM` : Tests for significant changes in PTM and protein abundance across conditions. Adjusts PTM fold change for changes in protein abundance.
- `groupComparisonPlotsPTM` : Visualization for model-based analysis and summarization

---

`ProgenesistoMSstatsPTMFormat`

Converts non-TMT Progenesis output into the format needed for MSstatsPTM

Description

Converts non-TMT Progenesis output into the format needed for MSstatsPTM

Usage

```
ProgenesistoMSstatsPTMFormat(
  ptm_input,
  annotation,
  global_protein_input = FALSE,
  fasta_path = FALSE,
  useUniquePeptide = TRUE,
  summaryforMultipleRows = max,
  removeFewMeasurements = TRUE,
```

```
  removeOxidationMpeptides = FALSE,
  removeProtein_with1Peptide = FALSE,
  mod.num = "Single"
)
```

Arguments

ptm_input	name of Progenesis output with modified peptides, 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, and Type (PTM or Protein) information. It will be matched with the column name of input for MS runs. Please note PTM and global Protein run names are often different, which is why an additional Type column indicating Protein or PTM is required.
global_protein_input	name of Progenesis output with unmodified peptides, which is wide-format. 'Accession', Sequence', 'Modification', 'Charge' and one column for each run are required
fasta_path	string containing path to the corresponding fasta file for the modified peptide dataset.
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.
removeFewMeasurements	TRUE (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.
mod.num	For modified peptide dataset, must be one of `Single` or `Total`. The default is `Single`. The number modifications per peptide to be used. If "Single", only peptides with one modification will be used. Otherwise "Total" includes peptides with more than one modification. Selecting "Total" may confound the effect of different modifications.

Value

a list of two data.tables named 'PTM' and 'PROTEIN' in the format required by MSstatsPTM.

Examples

```
# Example annotation file
annotation <- data.frame('Condition' = c('Control', 'Control', 'Control',
                         'Treatment', 'Treatment', 'Treatment'),
                         'BioReplicate' = c(1,2,3,4,5,6),
                         'Run' = c('prot_run_1', 'prot_run_2', 'prot_run_3',
                                   'phos_run_1', 'phos_run_2', 'phos_run_3'),
                         'Type' = c("Protein", "Protein", "Protein", "PTM",
                                    "PTM", "PTM"))

# The output should be in the following format.
head(raw.input$PTM)
head(raw.input$PROTEIN)
```

`raw.input` *Example of input PTM dataset for LabelFree/DDA/DIA experiments.*

Description

It can be the output of MSstatsPTM converter ProgenesistoMSstatsPTMFormat or other MSstats converter functions (Please see MSstatsPTM_LabelFree_Workflow vignette). The dataset is formatted as a list with two data.tables named PTM and PROTEIN. In each data.table the variables are as follows:

Usage

```
raw.input
```

Format

A list of two data.tables named PTM and PROTEIN with 1745 and 478 rows respectively.

Details

#'

ProteinName : Name of protein with modification site mapped in with an underscore. ie "Protein_4_Y474"

- PeptideSequence
- Condition : Condition (ex. Healthy, Cancer, Time0)
- BioReplicate : Unique ID for biological subject.
- Run : MS run ID.
- Intensity
- PrecursorCharge
- FragmentIon
- ProductCharge
- IsotopeLabelType

Examples

```
head(raw.input$PTM)
head(raw.input$PROTEIN)
```

`raw.input.tmt` *Example of input PTM dataset for TMT experiments.*

Description

It can be the output of MSstatsPTM converter MaxQtoMSstatsPTMFormat or other MSstatsTMT converter functions (Please see MSstatsPTM_TMT_Workflow vignette). The dataset is formatted as a list with two data.tables named PTM and PROTEIN. In each data.table the variables are as follows:

Usage

```
raw.input.tmt
```

Format

A list of two data.tables named PTM and PROTEIN with 1716 and 29221 rows respectively.

Details

- ProteinName : Name of protein with modification site mapped in with an underscore. ie "Protein_4_Y474"
- PeptideSequence
- Charge
- PSM
- Mixture : Mixture of samples labeled with different TMT reagents, which can be analyzed in a single mass spectrometry experiment. If the channal doesn't have sample, please add Empty' under Condition. \item TechRepMixture : Technical replicate of one mixture. One mixture may have multiple technical replicates. For example, if TechRepMixture' = 1, 2 are the two technical replicates of one mixture, then they should match with same Mixture' value. \item Run : MS run ID. \item C nel : Labeling information (126, ... 131). \item Condition : Condition (ex. Healthy, Cancer, Time0) \item BioReplicate : Unique ID for biological subject. If the channal doesn't have sample, please add Empty' under BioReplicate.
- Intensity

Examples

```
head(raw.input.tmt$PTM)
head(raw.input.tmt$PROTEIN)
```

`SpectronauttoMSstatsPTMFormat` *Converts raw PTM MS data from Spectronautt into the format needed for MSstatsPTM.*

Description

Takes as as input both raw PTM and global protein outputs from Spectronaut.

Usage

```
SpectronauttoMSstatsPTMFormat(
  PTM.data,
  fasta,
  Protein.data = NULL,
  annotation = NULL,
  intensity = "PeakArea",
  filter_with_Qvalue = TRUE,
  qvalue_cutoff = 0.01,
  useUniquePeptide = TRUE,
  removeFewMeasurements = TRUE,
  removeProtein_with1Feature = FALSE,
  removeNonUniqueProteins = TRUE,
  modificationLabel = "Phospho",
  removeiRT = TRUE,
  summaryforMultipleRows = max,
  which.Conditions = "all"
)
```

Arguments

`PTM.data`	name of PTM Spectronaut output, which is long-format.
`fasta`	A string of path to a FASTA file, used to match PTM peptides.
`Protein.data`	name of Global Protein Spectronaut output, which is long-format.
`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.

`removeFewMeasurements`
: TRUE (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.

`removeNonUniqueProteins`
: TRUE will remove proteins that were not uniquely identified. IE if the protein column contains multiple proteins seperated by ";". TRUE is default

`modificationLabel`
: String of modification name. Default is 'Phospho'.

`removeiRT`
: TRUE will remove proteins that contain iRT. True 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.

`which.Conditions`
: list of conditions to format into MSstatsPTM format. If "all" all conditions will be used. Default is "all".

Value

a list of two data.tables named 'PTM' and 'PROTEIN' in the format required by MSstatsPTM.

Examples

```
# The output should be in the following format.
head(raw.input$PTM)
head(raw.input$PROTEIN)
```

---

summary.data *Example of output from dataSummarizationPTM function for non-TMT data*

Description

It is made from `raw.input`. It is the output of dataSummarizationPTM function from MSstatsPTM. It should include a list with two names `PTM` and `PROTEIN`. Each of these list values is also a list with two names `ProteinLevelData` and `FeatureLevelData`, which correspond to two data.tables.The columns in these two data.tables are listed below. The variables are as follows:

- FeatureLevelData :
  - PROTEIN : Protein ID with modification site mapped in. Ex. Protein_1002_S836

  - PEPTIDE : Full peptide with charge
  - TRANSITION: Charge
  - FEATURE : Combination of Protien, Peptide, and Transition Columns
  - LABEL :
  - GROUP : Condition (ex. Healthy, Cancer, Time0)
  - RUN : Unique ID for technical replicate of one TMT mixture.
  - SUBJECT : Unique ID for biological subject.
  - FRACTION : Unique Fraction ID
  - originalRUN : Run name
  - censored :
  - INTENSITY : Unique ID for TMT mixture.
  - ABUNDANCE : Unique ID for TMT mixture.
  - newABUNDANCE : Unique ID for TMT mixture.
  - predicted : Unique ID for TMT mixture.

- ProteinLevelData :
  - RUN : MS run ID
  - Protein : Protein ID with modification site mapped in. Ex. Protein_1002_S836
  - LogIntensities: Protein-level summarized abundance
  - originalRUN : Labeling information (126, ... 131)
  - GROUP : Condition (ex. Healthy, Cancer, Time0)
  - SUBJECT : Unique ID for biological subject.
  - TotalGroupMeasurements : Unique ID for technical replicate of one TMT mixture.
  - NumMeasuredFeature : Unique ID for TMT mixture.
  - MissingPercentage : Unique ID for TMT mixture.
  - more50missing : Unique ID for TMT mixture.
  - NumImputedFeature : Unique ID for TMT mixture.

Usage

```
summary.data
```

Format

A list of two lists with four data.tables.

Examples

```
head(summary.data)
```

summary.data.tmt *Example of output from dataSummarizationPTM_TMT function for TMT data*

Description

It is made from raw.input.tmt. It is the output of dataSummarizationPTM_TMT function from MSstatsPTM. It should include a list with two names PTM and PROTEIN. Each of these list values is also a list with two names ProteinLevelData and FeatureLevelData, which correspond to two data.tables.The columns in these two data.tables are listed below. The variables are as follows:

- FeatureLevelData :
  - ProteinName : MS run ID
  - PSM : Protein ID with modification site mapped in. Ex. Protein_1002_S836
  - censored: Protein-level summarized abundance
  - predicted : Labeling information (126, ... 131)
  - log2Intensity : Condition (ex. Healthy, Cancer, Time0)
  - Run : Unique ID for biological subject.
  - Channel : Unique ID for technical replicate of one TMT mixture.
  - BioReplicate : Unique ID for TMT mixture.
  - Condition : Unique ID for TMT mixture.
  - Mixture : Unique ID for TMT mixture.
  - TechRepMixture : Unique ID for TMT mixture.
  - PeptideSequence : Unique ID for TMT mixture.
  - Charge : Unique ID for TMT mixture.
- ProteinLevelData :
  - Mixture : MS run ID
  - TechRepMixture : Protein ID with modification site mapped in. Ex. Protein_1002_S836
  - Run: Protein-level summarized abundance
  - Channel : Labeling information (126, ... 131)
  - Protein : Condition (ex. Healthy, Cancer, Time0)
  - Abundance : Unique ID for biological subject.
  - BioReplicate : Unique ID for technical replicate of one TMT mixture.
  - Condition : Unique ID for TMT mixture.

Usage

```
summary.data.tmt
```

Format

A list of two lists with four data.tables.

Examples

```
head(summary.data.tmt)
```

`tidyFasta` *Read and tidy a FASTA file*

Description

`tidyFasta` reads and tidys FASTA file. Use this function as the first step in identifying modification sites.

Usage

```
tidyFasta(path)
```

Arguments

`path`	A string of path to a FASTA file.

Value

A data.table with columns named `header`, `sequence`, `uniprot_ac`, `uniprot_iso`, `entry_name`.

Examples

```
tidyFasta(system.file("extdata", "O13297.fasta", package="MSstatsPTM"))
```

Index