Package: MSstatsPTM (via r-universe)

November 18, 2026

Type Package

Title Statistical Characterization of Post-translational Modifications

Version 2.14.0

Date 2024-11-25

Description MSstatsPTM provides general statistical methods for quantitative characterization of post-translational modifications (PTMs). Supports DDA, DIA, SRM, 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 (>= 4.3)

Imports dplyr, gridExtra, stringr, stats, ggplot2, stringi, grDevices, MSstatsTMT, MSstatsConvert (>= 1.19.1), MSstats, data.table, Rcpp, Biostrings, checkmate, ggrepel, plotly, htmltools, rlang

Suggests knitr, rmarkdown, tinytest, covr, mockery, arrow, testthat (>= 3.0.0)

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

URL https://vitek-lab.github.io/MSstatsPTM/

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/testthat/edition 3

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libssl-dev zlib1g-dev

Repository Bioconductor 3.23

Date/Publication 2026-04-28 12:53:49 UTC

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

RemoteRef RELEASE_3_23

RemoteSha 0ac46adccfe3179bce815dd10777abf7a2603d30

Contents

.fixTerminus . . . 3
annotSite . . . 3
dataProcessPlotsPTM . . . 4
dataProcessPTM . . . 6
dataSummarizationPTM . . . 7
dataSummarizationPTM_TMT . . . 10
designSampleSizePTM . . . 12
DIANNtoMSstatsPTMFormat . . . 13
fragpipe_annotation . . . 16
fragpipe_annotation_protein . . . 17
fragpipe_input . . . 18
fragpipe_input_protein . . . 18
FragPipetoMSstatsPTMFormat . . . 19
groupComparisonPlotsPTM . . . 22
groupComparisonPTM . . . 24
locateMod . . . 26
locatePTM . . . 27
maxq_lf_annotation . . . 27
maxq_lf_evidence . . . 28
maxq_tmt_annotation . . . 29
maxq_tmt_evidence . . . 30
MaxQtoMSstatsPTMFormat . . . 30
MetamorpheusToMSstatsPTMFormat . . . 34
MSstatsPTMSiteLocator . . . 36
pd_annotation . . . 38
pd_psm_input . . . 38
pd_testing_output . . . 39
PDtoMSstatsPTMFormat . . . 40
ProgenesistoMSstatsPTMFormat . . . 43
ProteinProspectortoMSstatsPTMFormat . . . 44
PStoMSstatsPTMFormat . . . 47
raw.input . . . 48
raw.input.tmt . . . 49
SkylinetoMSstatsPTMFormat . . . 50

spectronaut_annotation 52
spectronaut_input 53
SpectronauttoMSstatsPTMFormat 53
summary.data 56
summary.data.tmt 57
tidyFasta 58

Index **59**

`.fixTerminus` *Fix terminus location adjustments*

Description

Fix terminus location adjustments

Usage

```
.fixTerminus(data, terminus_id, unmod_pep_col)
```

Arguments

`data`	data.table containing peptide data
`terminus_id`	character string identifying the terminus (e.g. N-terminus)
`unmod_pep_col`	character string specifying the column name containing unmodified peptide sequences

Value

data.table with corrected Start positions

`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 = "",
  isPlotly = FALSE
)
```

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 the lower limit as floor of minimum log2(intensities) after normalization - 3, or 0 if that value is negative.
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 summarization.
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.
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 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)
```

---

dataProcessPTM *Data processing and summarization of peptide-level quantification to PTM and protein level quantification*

Description

Function to perform data processing and summarization on an experiment targeting post-translational modifications. Performs normalization, missing value imputation, feature selection, and summarization. Can optionally take an additional global protein quantification experiment for protein-level correction of PTM changes. Can take either label free or tandem mass tag (TMT) labeled data.

Usage

```
dataProcessPTM(
  data,
  ptm_label_type,
  protein_label_type,
  MBimpute_ptm = FALSE,
  MBimpute_protein = TRUE,
  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, however the PROTEIN dataset is optional.

`ptm_label_type` Indicator of labeling type for PTM dataset. Must be one of `LF` or `TMT`

`protein_label_type`
Indicator of labeling type for PROTEIN dataset. Must be one of `LF` or `TMT`

`MBimpute_ptm` Missing value imputation on the feature-level for the PTM dataset. TRUE (default) imputes missing values by Accelated failure model. FALSE uses minimum value to impute the missing value for each peptide precursor ion.

`MBimpute_protein`
Missing value imputation on the feature-level for the PROTEIN dataset. TRUE (default) imputes missing values by Accelated failure model. FALSE uses minimum value to impute the missing value for each peptide precursor ion.

`...` Additional parameters passed to either the dataProcess function from `MSstats` or the proteinSummarization function from `MSstatsTMT`.

`dataSummarizationPTM` *Data summarization function for label-free MS experiments targeting PTMs.*

Description

Utilizes functionality from MSstats to clean, summarize, and normalize PTM and protein level data. Imputes missing values, performs normalization, and summarizes data. PTM data is summarized up to the modification and protein data is summarized up to the protein level. Takes as input the output of the included converters (see included `raw.input` data object for required input format).

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) unmodified 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.
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

Data summarization function for TMT labelled MS experiments targeting PTMs.

Description

Utilizes functionality from MSstatsTMT to clean, summarize, and normalize PTM and protein level data. Imputes missing values, performs normalization, and summarizes data. PTM data is summarized up to the modification and protein data is summarized up to the protein level. Takes as input the output of the included converters (see included `raw.input.tmt` data object for required input format).

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)
```

`designSampleSizePTM` *Planning future experimental designs of PTM experiments in sample size calculation*

Description

Calculate sample size for future experiments of a PTM experiment based on intensity-based linear model. Calculation is only available for group comparison experimental designs (not including time series). Two options of the calculation: (1) number of biological replicates per condition, (2) power.

Usage

```
designSampleSizePTM(
  data,
  desiredFC,
  FDR = 0.05,
  numSample = TRUE,
  power = 0.8,
  use_log_file = TRUE,
  append = FALSE,
  verbose = TRUE,
  log_file_path = NULL,
  base = "MSstatsPTM_log_"
)
```

Arguments

`data`	output of the groupComparisonPTM function.
`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 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.

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

```
model.lf.msstatsptm = groupComparisonPTM(summary.data,
                                    ptm_label_type="LF",
                                    protein_label_type="LF",
                                    verbose = FALSE)

#(1) Minimal number of biological replicates per condition
designSampleSizePTM(data=model.lf.msstatsptm, numSample=TRUE,
                 desiredFC=c(2.0,2.75), FDR=0.05, power=0.8)
#(2) Power calculation
designSampleSizePTM(data=model.lf.msstatsptm, numSample=5,
                 desiredFC=c(2.0,2.75), FDR=0.05, power=TRUE)
```

DIANNtoMSstatsPTMFormat

Convert the output of DIA-NN PSM file into MSstatsPTM format

Description

Takes as input the report.tsv file from DIA-NN and converts it into MSstatsPTM format. Requires PSM and an annotation file. Optionally an additional report.tsv file for a corresponding global profiling run can be included.

Usage

```
DIANNtoMSstatsPTMFormat(
  input,
  annotation,
  input_protein = NULL,
  annotation_protein = NULL,
  fasta_path = NULL,
  use_unmod_peptides = FALSE,
  protein_id_col = "Protein.Group",
  fasta_protein_name = "uniprot_ac",
  global_qvalue_cutoff = 0.01,
  qvalue_cutoff = 0.01,
  pg_qvalue_cutoff = 0.01,
  useUniquePeptide = TRUE,
  removeFewMeasurements = TRUE,
  removeOxidationMpeptides = TRUE,
  removeProtein_with1Feature = FALSE,
  MBR = TRUE,
  quantificationColumn = "FragmentQuantCorrected",
  use_log_file = TRUE,
  append = FALSE,
  verbose = TRUE,
  log_file_path = NULL
)
```

Arguments

input	name of MSstats input report from Diann, which includes fragment-level data. Output fragment data with –export-quant flag in DIA-NN 2.0
annotation	name of 'annotation.txt' data which includes Condition, BioReplicate, Run.
input_protein	same as input for global profiling run. Default is NULL.
annotation_protein	same as annotation for global profiling run. Default is NULL.
fasta_path	A string of path to a FASTA file, used to match PTM peptides.
use_unmod_peptides	Boolean if the unmodified peptides in the input file should be used to construct the unmodified protein output. Only used if input_protein is not provided. Default is FALSE.
protein_id_col	Use 'Protein.Groups'(default) column for protein name.
fasta_protein_name	Name of column that matches with the protein names in protein_id_col. The protein names in these two columns must match in order to join the FASTA file with the DIA-NN output. Default is "uniprot_ac" for uniprot ID. For uniprot mnemonic ID, use "entry_name"
global_qvalue_cutoff	The qvalue cutoff for the Q.Value column, i.e. the run-specific precursor q-value. Default is 0.01.

qvalue_cutoff If MBR is false, the qvalue cutoff for the Global.Q.Value column, i.e. global precursor q-value. If MBR is true, the qvalue cutoff for the Lib.Q.Value column, i.e. the q-value for the library created after the first MBR pass. Default is 0.01.

pg_qvalue_cutoff
If MBR is false, the qvalue cutoff for the Global.PG.Q.Value column, i.e. the global q-value for the protein group. If MBR is true, the qvalue cutoff for the Lib.PG.Q.Value column, i.e. the protein group q-value for the library created after the first MBR pass. Default is 0.01.

useUniquePeptide
should unique peptides be removed

removeFewMeasurements
should proteins with few measurements be removed

removeOxidationMpeptides
should peptides with oxidation be removed

removeProtein_with1Feature
should proteins with a single feature be removed

MBR True if analysis was done with match between runs

quantificationColumn
Use 'FragmentQuantCorrected'(default) column for quantified intensities for DIANN 1.8.x. Use 'FragmentQuantRaw' for quantified intensities for DIANN 1.9.x. Use 'auto' for quantified intensities for DIANN 2.x where each fragment intensity is a separate column, e.g. Fr0Quantity.

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 one or two data.frame of class MSstatsTMT, named PTM and PROTEIN

Examples

```
# Example from PRIDE ID PXD053502
input = system.file("tinytest/raw_data/DIANN/report.tsv",
                                        package = "MSstatsPTM")
input = data.table::fread(input)
annot = system.file("tinytest/raw_data/DIANN/annot.csv",
                                        package = "MSstatsPTM")
annot = data.table::fread(annot)
fasta_path = system.file("extdata", "diann.fasta",
                         package="MSstatsPTM")
```

```
msstatsptm_format = DIANNtoMSstatsPTMFormat(
    input,
    annot,
    protein_id_col = "Protein.Names",
    fasta_path = fasta_path,
    fasta_protein_name = "entry_name",
    use_log_file = FALSE
)

head(msstatsptm_format$PTM)

# Example DIANN 2.0
input = system.file("tinytest/raw_data/DIANN/diann_2_ptm.parquet",
                                        package = "MSstatsPTM")
input = arrow::read_parquet(input)
annot = system.file("tinytest/raw_data/DIANN/annotation_diann_2.0_ptm.csv",
                                        package = "MSstatsPTM")
annot = data.table::fread(annot)
fasta_path = system.file("extdata", "diann.fasta",
                         package="MSstatsPTM")

msstatsptm_format = DIANNtoMSstatsPTMFormat(
    input,
    annot,
    protein_id_col = "Protein.Names",
    fasta_path = fasta_path,
    fasta_protein_name = "entry_name",
    use_log_file = FALSE,
    quantificationColumn = "auto"
)

head(msstatsptm_format$PTM)
```

`fragpipe_annotation` *Example annotation file for a TMT FragPipe experiment.*

Description

Automatically created by FragPipe, manually checked by the user and input into the FragPipetoMSstatsPTMFormat converter. Requires the correct columns and maps the experimental desing into the MSstats format. Specify unique bioreplicates for group comparison designs, and the same bioreplicate for repeated measure designs. The columns and descriptions are below.

Usage

```
fragpipe_annotation
```

Format

A data.table with 7 columns.

Details

- Run : Run name that matches exactly with FragPipe run. Used to join evidence and metadata in annotation file.
- Fraction : If multiple fractions were used (i.e. the same mixture split into multiple fractions) enter that here. TechRepMixture : Multiple runs using the same bioreplicate
- Channel : Mixture channel used
- Condition : Name of condition that was used for each run.
- Mixture : The unique mixture (plex) name
- BioReplicate : Name of biological replicate. Repeating the same name here will tell MSstatsPTM that the experiment is a repeated measure design.

Examples

```
head(fragpipe_annotation)
```

---

fragpipe_annotation_protein

Example annotation file for a global profiling run TMT FragPipe experiment.

Description

Automatically created by FragPipe, manually checked by the user and input into the FragPipetoMSstatsPTMFormat converter. Requires the correct columns and maps the experimental desing into the MSstats format. Specify unique bioreplicates for group comparison designs, and the same bioreplicate for repeated measure designs. The columns and descriptions are below.

Usage

```
fragpipe_annotation_protein
```

Format

A data.table with 7 columns.

Details

- Run : Run name that matches exactly with FragPipe run. Used to join evidence and metadata in annotation file.
- Fraction : If multiple fractions were used (i.e. the same mixture split into multiple fractions) enter that here. TechRepMixture : Multiple runs using the same bioreplicate
- Channel : Mixture channel used
- Condition : Name of condition that was used for each run.
- Mixture : The unique mixture (plex) name
- BioReplicate : Name of biological replicate. Repeating the same name here will tell MSstatsPTM that the experiment is a repeated measure design.

Examples

```
head(fragpipe_annotation_protein)
```

`fragpipe_input` *Output of FragPipe TMT PTM experiment*

Description

This dataset was provided by the FragPipe team at the Nesvilab. It was processed using Philosopher and targeted Phosphorylation.

Usage

```
fragpipe_input
```

Format

A data.table with 29 columns and 246 rows.

Examples

```
head(fragpipe_input)
```

`fragpipe_input_protein`
Output of FragPipe TMT global profiling experiment

Description

This dataset was provided by the FragPipe team at the Nesvilab. It was processed using Philosopher and targeted Phosphorylation.

Usage

```
fragpipe_input_protein
```

Format

A data.table with 27 columns and 47 rows.

Examples

```
head(fragpipe_input_protein)
```

FragPipetoMSstatsPTMFormat *Convert output of TMT labeled Fragpipe data into MSstatsPTM format.*

Description

Takes as input TMT experiments which are the output of Fragpipe and converts into MSstatsPTM format. Requires `msstats.csv` file and an annotation file. Optionally an additional `msstats.csv` file can be uploaded if a corresponding global profiling run was performed. Site localization is performed and only high probability localizations are kept.

Usage

```
FragPipetoMSstatsPTMFormat(
  input,
  annotation = NULL,
  input_protein = NULL,
  annotation_protein = NULL,
  use_unmod_peptides = FALSE,
  label_type = "TMT",
  protein_id_col = "Protein",
  peptide_id_col = "Peptide.Sequence",
  mod_id_col = "STY",
  localization_cutoff = 0.75,
  remove_unlocalized_peptides = TRUE,
  Purity_cutoff = 0.6,
  PeptideProphet_prob_cutoff = 0.7,
  useUniquePeptide = TRUE,
  rmPSM_withfewMea_withinRun = FALSE,
  rmPeptide_OxidationM = TRUE,
  rmProtein_with1Feature = FALSE,
  summaryforMultipleRows = sum,
  use_log_file = TRUE,
  append = FALSE,
  verbose = TRUE,
  log_file_path = NULL
)
```

Arguments

`input`	data.frame of `msstats.csv` file produced by Philosopher
`annotation`	annotation with Run, Fraction, TechRepMixture, Mixture, Channel, BioReplicate, Condition columns or a path to file. Refer to the example 'annotation' for the meaning of each column. Channel column should be consistent with the channel columns (Ignore the prefix "Channel ") in msstats.csv file. Run column should be consistent with the Spectrum.File columns in msstats.csv file.

`input_protein` same as `input` for global profiling run. Default is NULL.

`annotation_protein`
: same as `annotation` for global profiling run. Default is NULL.

`use_unmod_peptides`
: Boolean if the unmodified peptides in the input file should be used to construct the unmodified protein output. Only used if `input_protein` is not provided. Default is `FALSE`.

`label_type` Type of labeling used for experiment. Must be one of "LF" or "TMT". Default is "TMT".

`protein_id_col` Use 'Protein'(default) column for TMT. This needs to be changed to "ProteinName" for label free. For TMT, 'Master.Protein.Accessions' can be used instead to get the protein ID with single protein.

`peptide_id_col` Use 'Peptide.Sequence'(default) column for TMT. Must be changed to "PeptideSequence" for label free. "Modified.Peptide.Sequence" can be used instead to get the modified peptide sequence.

`mod_id_col` Column containing the modified Amino Acids. For example, a Phosphorylation experiment may pass `STY`. The corresponding column with `STY` combined with the mass (e.x. `STY.79.9663`) will be selected. Default is `STY`.

`localization_cutoff`
: Minimum localization score required to keep modification. Default is .75.

`remove_unlocalized_peptides`
: Boolean indicating if peptides without all sites localized should be kept. Default is TRUE (non-localized sites will be removed).

`Purity_cutoff` Cutoff for purity. Default is 0.6. Purity refers to how much of the detected ion signal within a specific inclusion window belongs to the target molecule or its closely related forms, compared to any other unwanted signals or noise. Higher values indicate greater purity.

`PeptideProphet_prob_cutoff`
: Cutoff for the peptide identification probability. Default is 0.7. The probability is confidence score determined by PeptideProphet and higher values indicate greater confidence.

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

`rmPSM_withfewMea_withinRun`
: TRUE will remove the features that have 1 or 2 measurements within each Run. Default is FALSE.

`rmPeptide_OxidationM`
: TRUE (default) will remove the peptides including oxidation (M) sequence.

`rmProtein_with1Feature`
: TRUE will remove the proteins which have only 1 peptide and charge. Default is FALSE.

`summaryforMultipleRows`
: sum (default) or max - when there are multiple measurements for certain feature in certain run, select the feature with the largest summation or maximal value.

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

`list` of one or two `data.frame` of class `MSstatsTMT`, named `PTM` and `PROTEIN`

Examples

```
# TMT Example (with global profiling run)
head(fragpipe_input)
head(fragpipe_annotation)
head(fragpipe_input_protein)
head(fragpipe_annotation_protein)

msstats_data = FragPipetoMSstatsPTMFormat(fragpipe_input,
                                          fragpipe_annotation,
                                          fragpipe_input_protein,
                                          fragpipe_annotation_protein,
                                          label_type="TMT",
                                          mod_id_col = "STY",
                                          localization_cutoff=.75,
                                          remove_unlocalized_peptides=TRUE)
head(msstats_data$PTM)
head(msstats_data$PROTEIN)

# LFQ Example
input = system.file("tinytest/raw_data/Fragpipe/MSstats.csv",
                                        package = "MSstatsPTM")
input = data.table::fread(input)
annot = system.file("tinytest/raw_data/Fragpipe/experiment_annotation.tsv",
                                        package = "MSstatsPTM")
annot = data.table::fread(annot)
input_protein = system.file("tinytest/raw_data/Fragpipe/msstats_proteome_lf.csv",
                                        package = "MSstatsPTM")
input_protein = data.table::fread(input_protein)

msstats_data = FragPipetoMSstatsPTMFormat(input,
                                          annot,
                                          input_protein = input_protein,
                                          label_type="LF",
                                          mod_id_col = "STY",
                                          localization_cutoff=.75,
                                          protein_id_col = "ProteinName",
                                          peptide_id_col = "PeptideSequence")
```

```
# If no global profiling run is available, omit input_protein and set:
# msstats_data = FragPipetoMSstatsPTMFormat(input, annot,
#   label_type = "LF", mod_id_col = "STY",
#   localization_cutoff = .75, protein_id_col = "ProteinName",
#   peptide_id_col = "PeptideSequence", use_unmod_peptides = FALSE)

head(msstats_data$PTM)
head(msstats_data$PROTEIN)
```

`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 = NULL,
  width = 10,
  height = 10,
  which.Comparison = "all",
```

```
  which.PTM = "all",
  address = "",
  isPlotly = FALSE
)
```

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

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

```
model.lf.msstatsptm = groupComparisonPTM(summary.data,
                                    ptm_label_type="LF",
                                    protein_label_type="LF")
groupComparisonPlotsPTM(data = model.lf.msstatsptm,
                        type = "VolcanoPlot",
                        FCcutoff= 2,
                        logBase.pvalue = 2,
                        address=FALSE)
```

---

groupComparisonPTM *Perform differential analysis on MS-based proteomics experiments targeting PTMs*

Description

Takes summarized PTM and protein data from dataSummarizationPTM or dataSummarizationPTM_TMT functions and performs differential analysis. Leverages unmodified protein data to perform adjustment and deconvolute the effect of the PTM and unmodified protein. If protein data is unavailable, PTM data can still be passed into the function, however adjustment can not be performed. All model results are returned for completeness.

Usage

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

Arguments

data	list of summarized datasets. Output of MSstatsPTM summarization function dataSummarizationPTM or dataSummarizationPTM_TMT depending on acquisition type.
data.type	Type of data. Must be one of LF or TMT. Will be deprecated in favor of ptm_label_type and protein_label_type.
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 experiments 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.
save_fitted_models	logical, if TRUE, fitted models will be added to the output.
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.

`ptm_label_type` Indicator of labeling type for PTM dataset. Must be one of `LF` or `TMT`

`protein_label_type`
Indicator of labeling type for PROTEIN dataset. Must be one of `LF` or `TMT`

Value

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

Examples

```
model.lf.msstatsptm = groupComparisonPTM(summary.data,
                                         ptm_label_type="LF",
                                         protein_label_type="LF",
                                         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", "\\*")
```

maxq_lf_annotation *Example annotation file for a label-free MaxQuant experiment.*

Description

Must be manually created by the user and input into the MaxQtoMSstatsPTMFormat converter. Requires the correct columns and maps the experimental desing into the MSstats format. Specify unique bioreplicates for group comparison designs, and the same bioreplicate for repeated measure designs. The columns and descriptions are below.

Usage

```
maxq_lf_annotation
```

Format

A data.table with 5 columns.

Details

- Run : Run name that matches exactly with MaxQuant run. Used to join evidence and metadata in annotation file.
- Condition : Name of condition that was used for each run.
- BioReplicate : Name of biological replicate. Repeating the same name here will tell MSstatsPTM that the experiment is a repeated measure design.
- Raw.file : Run name that matches exactly with MaxQuant run. Used to join evidence and metadata in annotation file.
- IsotopeLabelType: Name of isotope label. May be all `L` or unique depending on experimental design.

Examples

```
head(maxq_lf_annotation)
```

---

maxq_lf_evidence *Example MaxQuant evidence file from the output of a label free experiment*

Description

Experiment was performed by the Olsen lab and published on Nat. Commun. (citation below).

Usage

```
maxq_lf_evidence
```

Format

a data.table with 63 columns and 511 rows, the output of MaxQuant

Details

Bekker-Jensen, D.B., Bernhardt, O.M., Hogrebe, A. et al. Rapid and site-specific deep phosphoproteome profiling by data-independent acquisition without the need for spectral libraries. Nat Commun 11, 787 (2020). https://doi.org/10.1038/s41467-020-14609-1

The experiment was processed using MaxQuant by the computational proteomics team at Pfizer (Liang Xue and Pierre Jean).

The experiment did not contain a global profiling run, but we show an example of extracting the unmodified peptides and using them in place of the profiling run.

Examples

```
head(maxq_lf_evidence)
```

`maxq_tmt_annotation` *Example annotation file for a TMT MaxQuant experiment.*

Description

Must be manually created by the user and input into the MaxQtoMSstatsPTMFormat converter. Requires the correct columns and maps the experimental desing into the MSstats format. Specify unique bioreplicates for group comparison designs, and the same bioreplicate for repeated measure designs. The columns and descriptions are below.

Usage

```
maxq_tmt_annotation
```

Format

A data.table with 7 columns.

Details

- Run : Run name that matches exactly with MaxQuant run. Used to join evidence and metadata in annotation file.
- Fraction : If multiple fractions were used (i.e. the same mixture split into multiple fractions) enter that here. TechRepMixture : Multiple runs using the same bioreplicate
- Channel : Mixture channel used
- Condition : Name of condition that was used for each run.
- Mixture : The unique mixture (plex) name
- BioReplicate : Name of biological replicate. Repeating the same name here will tell MSstatsPTM that the experiment is a repeated measure design.

Examples

```
head(maxq_tmt_annotation)
```

`maxq_tmt_evidence` *Example MaxQuant evidence file from the output of a TMT experiment*

Description

Experiment was performed by the Olsen lab and published on Nat. Commun. (citation below).

Usage

```
maxq_tmt_evidence
```

Format

a data.table with 96 columns and 199 rows, the output of MaxQuant

Details

Hogrebe, A., von Stechow, L., Bekker-Jensen, D.B. et al. Benchmarking common quantification strategies for large-scale phosphoproteomics. Nat Commun 9, 1045 (2018). https://doi.org/10.1038/s41467-018-03309-6

The experiment was processed using MaxQuant by the computational proteomics team at Pfizer (Liang Xue and Pierre Jean).

The experiment did not contain a global profiling run, but we show an example of extracting the unmodified peptides and using them in place of the profiling run.

Examples

```
head(maxq_tmt_evidence)
```

`MaxQtoMSstatsPTMFormat` *Convert output of label-free or TMT MaxQuant experiments into MSstatsPTM format*

Description

Takes as input LF/TMT experiments from MaxQ and converts the data into the format needed for MSstatsPTM. Requires modified evidence.txt file from MaxQ 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. Optionally can use `Phospho(STY)Sites.txt` (or other PTM specific files) from MaxQuant, but this is not recommended. If PTM specific file provided, the raw intensities must be provided, not a ratio.

Usage

```
MaxQtoMSstatsPTMFormat(
  evidence = NULL,
  annotation = NULL,
  fasta_path,
  fasta_protein_name = "uniprot_ac",
  mod_id = "\\(Phospho \\(STY\\)\\)",
  sites_data = NULL,
  evidence_prot = NULL,
  proteinGroups = NULL,
  annotation_protein = NULL,
  use_unmod_peptides = FALSE,
  labeling_type = "LF",
  mod_num = "Single",
  TMT_keyword = "TMT",
  ptm_keyword = "phos",
  which_proteinid_ptm = "Proteins",
  which_proteinid_protein = "Proteins",
  remove_other_mods = TRUE,
  removeMpeptides = FALSE,
  removeOxidationMpeptides = FALSE,
  removeProtein_with1Peptide = FALSE,
  use_log_file = TRUE,
  append = FALSE,
  verbose = TRUE,
  log_file_path = NULL
)
```

Arguments

evidence	name of 'evidence.txt' data, which includes feature-level data for enriched (PTM) data.
annotation	data frame annotation file for the ptm level data. Contains column Run, Fraction, TechRepMixture, Mixture, Channel, BioReplicate, Condition.
fasta_path	A string of path to a FASTA file, used to match PTM peptides.
fasta_protein_name	Name of fasta column that matches with protein name in evidence file. Default is uniprot_ac.
mod_id	Character that indicates the modification of interest. Default is \\(Phospho\\). Note \\ must be included before special characters.
sites_data	(Not recommended. Only used if evidence file not provided. Only works for TMT labeled data) Modified peptide output from MaxQuant. For example, a phosphorylation experiment would require the Phospho(STY)Sites.txt file
evidence_prot	name of 'evidence.txt' data, which includes feature-level data for global profiling (unmodified) data.
proteinGroups	name of 'proteinGroups.txt' data. It needs to matching protein group ID in evidence_prot.

annotation_protein
: data frame annotation file for the protein level data. Contains column Run, Fraction, TechRepMixture, Mixture, Channel, BioReplicate, Condition.

use_unmod_peptides
: Boolean if the unmodified peptides in the input file should be used to construct the unmodified protein output. Only used if input_protein is not provided. Default is FALSE.

labeling_type
: Either TMT or LF (Label-Free) depending on experimental design. Default is LF.

mod_num
: (Only if sites.data is used) 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.

TMT_keyword
: (Only if sites.data is used) the sub-name of columns in sites.data file. Default is TMT. This corresponds to the columns in the format Reporter.intensity.corrected.1.TMT1phos___ Specifically, this parameter indicates the first section of the string TMT1phos (Before the mixture number). If TMT is present in the string, set this value to TMT. Else if TMT is not there (ie string is in the format 1phos) leave this parameter as an empty string (").

ptm_keyword
: (Only if sites.data is used) the sub-name of columns in the sites.data file. Default is phos. This corresponds to the columns in the format Reporter.intensity.corrected.1.TMT1ph Specifically, this parameter indicates the second section of the string TMT1phos (After the mixture number). If the string is present, set this parameter. Else if this part of the string is empty (ie string is in the format TMT1) leave this parameter as an empty string (").

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.

remove_other_mods
: Remove peptides which include modfications other than the one listed in mod_id. Default is TRUE. For example, in an experiment targeting Phosphorylation, setting this parameter to TRUE would remove peptides like (Acetyl (Protein N-term))AAAAPDSRVS(Phospho (STY))EEENLK. Set this parameter to FALSE to keep peptides with extraneous modifications.

removeMpeptides
: If Oxidation (M) modifications should be removed. Default is TRUE.

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.

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

Examples

```
# TMT experiment
head(maxq_tmt_evidence)
head(maxq_tmt_annotation)

msstats_format_tmt = MaxQtoMSstatsPTMFormat(evidence=maxq_tmt_evidence,
                        annotation=maxq_tmt_annotation,
                fasta=system.file("extdata", "maxq_tmt_fasta.fasta", package="MSstatsPTM"),
                        fasta_protein_name="uniprot_ac",
                        mod_id="\\(Phospho \\(STY\\)\\)",
                        use_unmod_peptides=TRUE,
                        labeling_type = "TMT",
                        which_proteinid_ptm = "Proteins")

head(msstats_format_tmt$PTM)
head(msstats_format_tmt$PROTEIN)

# LF experiment
head(maxq_lf_evidence)
head(maxq_lf_annotation)

msstats_format_lf = MaxQtoMSstatsPTMFormat(evidence=maxq_lf_evidence,
                        annotation=maxq_lf_annotation,
                fasta=system.file("extdata", "maxq_lf_fasta.fasta", package="MSstatsPTM"),
                        fasta_protein_name="uniprot_ac",
                        mod_id="\\(Phospho \\(STY\\)\\)",
                        use_unmod_peptides=TRUE,
                        labeling_type = "LF",
                        which_proteinid_ptm = "Proteins")
head(msstats_format_lf$PTM)
head(msstats_format_lf$PROTEIN)
```

`MetamorpheusToMSstatsPTMFormat` *Import Metamorpheus files into PTM format*

Description

Import Metamorpheus files into PTM format

Usage

```
MetamorpheusToMSstatsPTMFormat(
  input,
  annotation,
  fasta_path,
  input_protein = NULL,
  annotation_protein = NULL,
  use_unmod_peptides = FALSE,
  mod_ids = c("\\[Common Biological:Phosphorylation on S\\]"),
  useUniquePeptide = TRUE,
  removeFewMeasurements = TRUE,
  removeProtein_with1Feature = FALSE,
  summaryforMultipleRows = max,
  use_log_file = TRUE,
  append = FALSE,
  verbose = TRUE,
  log_file_path = NULL
)
```

Arguments

`input`	name of Metamorpheus output file, which is tabular format. Use the AllQuantifiedPeaks.tsv file from the Metamorpheus output.
`annotation`	name of 'annotation.txt' data which includes Condition, BioReplicate.
`fasta_path`	string containing path to the corresponding fasta file for the modified peptide dataset.
`input_protein`	same as `input` for global profiling run. Default is NULL.
`annotation_protein`	same as `annotation` for global profiling run. Default is NULL.
`use_unmod_peptides`	If `protein_input` is not provided, unmodified peptides can be extracted from `input` to be used in place of a global profiling run. Default is `FALSE`.
`mod_ids`	List of modifications of interest. Default is a list with only `Common Biological:Phosphorylation on S`. Please note that the 'mod_ids' parameter currently supports lists of size 1 only. Future updates aim to extend its functionality to accommodate lists of greater sizes. Note \\ must be included before special characters.

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.

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

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

Author(s)

Anthony Wu

Examples

```
input = system.file("tinytest/raw_data/Metamorpheus/AllQuantifiedPeaks.tsv",
                                package = "MSstatsPTM")
input = data.table::fread(input)
annot = system.file("tinytest/raw_data/Metamorpheus/ExperimentalDesign.tsv",
                                package = "MSstatsPTM")
annot = data.table::fread(annot)
input_protein = system.file("tinytest/raw_data/Metamorpheus/AllQuantifiedPeaksGlobalProteome.tsv",
                                package = "MSstatsPTM")
input_protein = data.table::fread(input_protein)
annot_protein = system.file("tinytest/raw_data/Metamorpheus/ExperimentalDesignGlobalProteome.tsv",
                                package = "MSstatsPTM")
annot_protein = data.table::fread(annot_protein)
fasta_path=system.file("extdata", "metamorpheus_fasta.fasta",
                                package="MSstatsPTM")
metamorpheus_imported = MetamorpheusToMSstatsPTMFormat(
    input,
    annot,
    fasta_path=fasta_path,
```

```
    input_protein=input_protein,
    annotation_protein=annot_protein,
    use_unmod_peptides=FALSE,
    mod_ids = c("\\[Common Fixed:Carbamidomethyl on C\\]")
)
head(metamorpheus_imported$PTM)
head(metamorpheus_imported$PROTEIN)
```

`MSstatsPTMSiteLocator` *Locate modification site number and amino acid*

Description

Locate modification site number and amino acid

Usage

```
MSstatsPTMSiteLocator(
  data,
  protein_name_col = "ProteinName",
  unmod_pep_col = "PeptideSequence",
  mod_pep_col = "PeptideModifiedSequence",
  clean_mod = FALSE,
  fasta_file = NULL,
  fasta_protein_name = "header",
  mod_id = "\\*",
  localization_scores = FALSE,
  localization_cutoff = 0.75,
  remove_unlocalized_peptides = TRUE,
  terminus_included = FALSE,
  terminus_id = "\\.",
  mod_id_is_numeric = FALSE,
  remove_underscores = FALSE,
  remove_other_mods = FALSE,
  bracket = FALSE,
  replace_text = FALSE
)
```

Arguments

`data`	`data.table` of enriched experimental run. Must include `ProteinName`, `PeptideSequence`, `PeptideModifiedSequence`, and (optionally) `Start` columns.
`protein_name_col`	Name of column indicating protein. Default is `ProteinName`.
`unmod_pep_col`	Name of column indicating unmodified peptide sequence. Default is `PeptideSequence`.
`mod_pep_col`	Name of column indicating modified peptide sequence. Default is `PeptideModifiedSequence`.

clean_mod Remove special characters and numbers around modification name. Default is FALSE

fasta_file File path to FASTA file that matches with proteins in data. Can be either string or data.table processed with tidyFasta() function. Default to NULL if peptide number included in data.

fasta_protein_name
Name of fasta file column that matches with protein_name_col. Default is header.

mod_id String that indicates what amino acid was modified in PeptideSequence.

localization_scores
Boolean indicating if mod id is a localization score. If TRUE, mod_id will be ignored and localization cutoff will be used to determine sites. Default is FALSE.

localization_cutoff
Default is .75. Localization probabilities below cutoffs will be removed. localization_scores must be TRUE.

remove_unlocalized_peptides
Default is TRUE. If localization_scores is TRUE and probabilities are below localization_cutoff, the modification site will not be able to be determined. These unlocalized peptides can be kept or removed. If FALSE the unlocalized peptides will still be used in modeling the sites that could be localized.

terminus_included
Boolean indicating if the PeptideSequence includes the terminus amino acid.

terminus_id String that indicates what the terminus amino acid is. Default is '.'.

mod_id_is_numeric
Boolean indicating if modification identifier is a number instead of a character (i.e. +80 vs *).

remove_underscores
Boolean indicating if underscores around peptide exist. These should be removed to properly count where in sequence the modification occurred.

remove_other_mods
keeping mods that are not of interest can mess up the amino acid count. Remove them if they are causing issues.

bracket bracket type that encompasses PTM (usually [or (). Always pass opening bracket (there is a function to grab the close bracket). Default is FALSE (i.e. no bracket).

replace_text If PTM is noted by text (i.e. Phospho) and needs to be replaced by an indicator (*)

Value

data.table with site location added into Protein column.

Examples

```
##TODO
```

pd_annotation — *Example annotation file for a label-free Proteome Discoverer experiment.*

Description

Must be manually created by the user and input into the PDtoMSstatsPTMFormat converter. Requires the correct columns and maps the experimental desing into the MSstats format. Specify unique bioreplicates for group comparison designs, and the same bioreplicate for repeated measure designs. The columns and descriptions are below.

Usage

```
pd_annotation
```

Format

A data.table with 3 columns.

Details

- Run : Run name that matches exactly with PD run. Used to join evidence and metadata in annotation file.
- Condition : Name of condition that was used for each run.
- BioReplicate : Name of biological replicate. Repeating the same name here will tell MSstatsPTM that the experiment is a repeated measure design.

Examples

```
head(pd_annotation)
```

pd_psm_input — *Example Proteome Discoverer evidence file from the output of a label free experiment*

Description

Experiment was performed by the Olsen lab and published on Nat. Commun. (citation below).

Usage

```
pd_psm_input
```

Format

a data.table with 60 columns and 1657 rows, the output of PD

Details

Bekker-Jensen, D.B., Bernhardt, O.M., Hogrebe, A. et al. Rapid and site-specific deep phosphoproteome profiling by data-independent acquisition without the need for spectral libraries. Nat Commun 11, 787 (2020). https://doi.org/10.1038/s41467-020-14609-1

The experiment was processed using Proteome Discoverer by the computational proteomics team at Pfizer (Liang Xue and Pierre Jean).

The experiment did not contain a global profiling run, but we show an example of extracting the unmodified peptides and using them in place of the profiling run.

Examples

```
head(pd_psm_input)
```

`pd_testing_output` *Example output of Proteome Discoverer converter*

Description

output using example data provided in package

Usage

```
pd_testing_output
```

Format

a list with 2 data.frames

Details

The experiment did not contain a global profiling run, but we show an example of extracting the unmodified peptides and using them in place of the profiling run.

Examples

```
head(pd_testing_output)
```

PDtoMSstatsPTMFormat *Convert Proteome Discoverer output into MSstatsPTM format*

Description

Import Proteome Discoverer files, identify modification site location.

Usage

```
PDtoMSstatsPTMFormat(
  input,
  annotation,
  fasta_path,
  protein_input = NULL,
  annotation_protein = NULL,
  labeling_type = "LF",
  mod_id = "\\(Phospho\\)",
  use_localization_cutoff = FALSE,
  keep_all_mods = FALSE,
  use_unmod_peptides = FALSE,
  fasta_protein_name = "uniprot_iso",
  localization_cutoff = 75,
  remove_unlocalized_peptides = TRUE,
  useNumProteinsColumn = FALSE,
  useUniquePeptide = TRUE,
  summaryforMultipleRows = max,
  removeFewMeasurements = TRUE,
  removeOxidationMpeptides = FALSE,
  removeProtein_with1Peptide = FALSE,
  which_quantification = "Precursor.Area",
  which_proteinid = "Protein.Group.Accessions",
  use_log_file = TRUE,
  append = FALSE,
  verbose = TRUE,
  log_file_path = NULL
)
```

Arguments

input	PD report corresponding with enriched experimental data.
annotation	name of 'annotation.txt' or 'annotation.csv' data which includes Condition, BioReplicate, Run information. 'Run' will be matched with 'Spectrum.File'
fasta_path	string containing path to the corresponding fasta file for the modified peptide dataset.
protein_input	PD report corresponding with unmodified experimental data.

`annotation_protein`
: Same format as `annotation` corresponding to unmodified data.

`labeling_type`
: type of experimental design, must be one of `LF` for label free or `TMT` for tandem mass tag.

`mod_id`
: Character that indicates the modification of interest. Default is `\\(Phospho\\)`. Note `\\` must be included before special characters.

`use_localization_cutoff`
: Boolean indicating whether to use a custom localization cutoff or rely on PD's modifications column. `TRUE` is default and apply custom cutoff `localization_cutoff`.

`keep_all_mods`
: Boolean indicating whether to keep or remove peptides not in `mod_id`. Default is FALSE.

`use_unmod_peptides`
: If `protein_input` is not provided, unmodified peptides can be extracted from `input` to be used in place of a global profiling run. Default is `FALSE`.

`fasta_protein_name`
: Name of fasta column that matches with protein name in evidence file. Default is `uniprot_iso`.

`localization_cutoff`
: Minimum localization score required to keep modification. Default is .75.

`remove_unlocalized_peptides`
: Boolean indicating if peptides without all sites localized should be kept. Default is TRUE (non-localized sites will be removed).

`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.

`removeFewMeasurements`
: TRUE (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_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.

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 data.table

Examples

```
# Global profiling example
input = system.file("tinytest/raw_data/PD/pd-ptm-input.csv",
package = "MSstatsPTM")
input = data.table::fread(input)
annot = system.file("tinytest/raw_data/PD/pd-ptm-annot.csv",
                    package = "MSstatsPTM")
annot = data.table::fread(annot)
input_protein = system.file("tinytest/raw_data/PD/protein-input.csv",
                            package = "MSstatsPTM")
input_protein = data.table::fread(input_protein)
annot_protein = system.file("tinytest/raw_data/PD/protein-annot.csv",
                            package = "MSstatsPTM")
annot_protein = data.table::fread(annot_protein)
fasta_path=system.file("extdata", "pd_with_proteome.fasta",
                       package="MSstatsPTM")
pd_imported = PDtoMSstatsPTMFormat(
    input,
    annotation = annot,
    protein_input = input_protein,
    annotation_protein = annot_protein,
    fasta_path = fasta_path,
    mod_id = "\\(GG\\)",
    labeling_type = "TMT",
    use_localization_cutoff = FALSE,
    which_proteinid = "Master.Protein.Accessions")

head(pd_imported$PTM)
head(pd_imported$PROTEIN)

# No global profiling example
head(pd_psm_input)
head(pd_annotation)

msstats_format = PDtoMSstatsPTMFormat(pd_psm_input,
                                      pd_annotation,
                          system.file("extdata", "pd_fasta.fasta", package="MSstatsPTM"),
```

```
                                    use_unmod_peptides=TRUE,
                                    which_proteinid = "Master.Protein.Accessions")

head(msstats_format$PTM)
head(msstats_format$PROTEIN)
```

`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

```
input = system.file("tinytest/raw_data/Progenesis/progenesis_peptide.csv",
                                package = "MSstatsPTM")
input = data.table::fread(input)
colnames(input) = unlist(input[1,])
input = input[-1,]
annot = system.file("tinytest/raw_data/Progenesis/phospho_annotation.csv",
                                package = "MSstatsPTM")
annot = data.table::fread(annot)
prog_imported = ProgenesistoMSstatsPTMFormat(
    input,
    annot
)
head(prog_imported$PTM)
```

`ProteinProspectortoMSstatsPTMFormat`
: *Generate MSstatsPTM required input format from Protein Prospector output*

Description

Generate MSstatsPTM required input format from Protein Prospector output

Usage

```
ProteinProspectortoMSstatsPTMFormat(
  input,
  annotation,
  input_protein = NULL,
  annotation_protein = NULL,
  use_unmod_peptides = FALSE,
  mod_ids = c("Phospho"),
  useUniquePeptide = TRUE,
  removeFewMeasurements = TRUE,
  removeProtein_with1Feature = FALSE,
  summaryforMultipleRows = sum,
  use_log_file = TRUE,
  append = FALSE,
  verbose = TRUE,
  log_file_path = NULL
)
```

Arguments

`input`	Input txt peptide report file from Protein Prospector with "Keep Replicates", "Mods in Peptide", and "Protein Mods" options selected.
`annotation`	data frame which contains column Run, Fraction, TechRepMixture, Mixture, Channel, BioReplicate, Condition.
`input_protein`	same as `input` for global profiling run. Default is NULL.
`annotation_protein`	same as `annotation` for global profiling run. Default is NULL.
`use_unmod_peptides`	If `protein_input` is not provided, unmodified peptides can be extracted from `input` to be used in place of a global profiling run. Default is `FALSE`.
`mod_ids`	List of modifications of interest. Default is a list with only `Phospho`. Please note that the 'mod_ids' parameter currently supports lists of size 1 only. Future updates aim to extend its functionality to accommodate lists of greater sizes.
`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.

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

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

Author(s)

Anthony Wu

Examples

```
input = system.file("tinytest/raw_data/ProteinProspector/Prospector_PhosphoTMT.txt",
    package = "MSstatsPTM")
input = data.table::fread(input)
annot = system.file("tinytest/raw_data/ProteinProspector/Annotation.csv",
                                package = "MSstatsPTM")
annot = data.table::fread(annot)
input_protein = system.file("tinytest/raw_data/ProteinProspector/Prospector_TotalTMT.txt",
    package = "MSstatsConvert")
input_protein = data.table::fread(input_protein)
annot_protein = system.file("tinytest/raw_data/ProteinProspector/Annotation.csv",
                                package = "MSstatsConvert")
annot_protein = data.table::fread(annot_protein)
output <- ProteinProspectortoMSstatsPTMFormat(
    input,
    annot,
    input_protein,
    annot_protein
)
head(output)
```

PStoMSstatsPTMFormat *Convert Peaks Studio output into MSstatsPTM format*

Description

Currently only supports label-free quantification.

Usage

```
PStoMSstatsPTMFormat(
  input,
  annotation,
  input_protein = NULL,
  annotation_protein = NULL,
  use_unmod_peptides = FALSE,
  target_modification = NULL,
  remove_oxidation_peptides = FALSE,
  remove_multi_mod_types = FALSE,
  summaryforMultipleRows = max,
  use_log_file = TRUE,
  append = FALSE,
  verbose = TRUE,
  log_file_path = NULL
)
```

Arguments

input	name of Peaks Studio PTM output
annotation	name of annotation file which includes Raw.file, Condition, BioReplicate, Run. For example annotation see example below.
input_protein	name of Peaks Studio unmodified protein output (optional)
annotation_protein	name of annotation file which includes Raw.file, Condition, BioReplicate, Run for unmodified protein output.
use_unmod_peptides	Boolean if the unmodified peptides in the input file should be used to construct the unmodified protein output. Only used if input_protein is not provided. Default is FALSE
target_modification	Character name of modification of interest. To use all mod types, leave as NULL. Default is NULL. Note that if the name includes special characters, you must include "\" before the characters. Ex. "Phosphorylation \(STY\)"
remove_oxidation_peptides	Boolean if Oxidation (M) modifications should be removed. Default is FALSE

remove_multi_mod_types
: Used if target_modification is not NULL. TRUE will remove peptides with multiple types of modifications (ie acetylation and phosphorylation). FALSE will keep these peptides and summarize them seperately.

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

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

list of data.table

Examples

```
# 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 h` Mixture' = 1, 2 are the two technical replicates of one mixture, then they should match with same `Mixture' value. \item Run : MS run ID. \item Channel : Labeling information (126, ... 131). \ite` under BioReplicate.
- Intensity

Examples

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

`SkylinetoMSstatsPTMFormat` *Convert Skyline output into MSstatsPTM format*

Description

Currently only supports label-free quantification.

Usage

```
SkylinetoMSstatsPTMFormat(
  input,
  fasta_path,
  fasta_protein_name = "uniprot_iso",
  annotation = NULL,
  input_protein = NULL,
  annotation_protein = NULL,
  use_unmod_peptides = FALSE,
  removeiRT = TRUE,
  filter_with_Qvalue = TRUE,
  qvalue_cutoff = 0.01,
  use_unique_peptide = TRUE,
  remove_few_measurements = FALSE,
  remove_oxidation_peptides = FALSE,
  removeProtein_with1Feature = FALSE,
```

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

Arguments

input	name of Skyline PTM output
fasta_path	A string of path to a FASTA file, used to match PTM peptides.
fasta_protein_name	Name of fasta column that matches with protein name in evidence file. Default is uniprot_iso.
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.
input_protein	name of Skyline unmodified protein output (optional)
annotation_protein	name of 'annotation.txt' data which includes Condition, BioReplicate, Run for unmodified protein output. This can be the same as annotation.
use_unmod_peptides	Boolean if the unmodified peptides in the input file should be used to construct the unmodified protein output. Only used if input_protein is not provided. Default is FALSE.
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.
use_unique_peptide	TRUE (default) removes peptides that are assigned for more than one proteins. We assume to use unique peptide for each protein.
remove_few_measurements	TRUE will remove the features that have 1 or 2 measurements across runs. FALSE is default.
remove_oxidation_peptides	TRUE will remove the peptides including 'oxidation (M)' in modification. FALSE is default.
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.
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

list of data.table

Examples

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

---

spectronaut_annotation

Example annotation file for a label-free Spectronaut experiment.

Description

Must be manually created by the user and input into the SpectronauttoMSstatsPTMFormat converter. Requires the correct columns and maps the experimental desing into the MSstats format. Specify unique bioreplicates for group comparison designs, and the same bioreplicate for repeated measure designs. The columns and descriptions are below.

Usage

```
spectronaut_annotation
```

Format

A data.table with 5 columns.

Details

- Run : Run name that matches exactly with Spectronaut run. Used to join evidence and metadata in annotation file.
- Condition : Name of condition that was used for each run.
- BioReplicate : Name of biological replicate. Repeating the same name here will tell MSstatsPTM that the experiment is a repeated measure design.
- Raw.file : Run name that matches exactly with Spectronaut run. Used to join evidence and metadata in annotation file.

Examples

```
head(spectronaut_annotation)
```

spectronaut_input *Example Spectronaut evidence file from the output of a label free experiment*

Description

Experiment was performed by the Olsen lab and published on Nat. Commun. (citation below).

Usage

```
spectronaut_input
```

Format

a data.table with 23 columns and 2683 rows, the output of Spectronaut

Details

Bekker-Jensen, D.B., Bernhardt, O.M., Hogrebe, A. et al. Rapid and site-specific deep phosphoproteome profiling by data-independent acquisition without the need for spectral libraries. Nat Commun 11, 787 (2020). https://doi.org/10.1038/s41467-020-14609-1

The experiment was processed using Spectronaut by the computational proteomics team at Pfizer (Liang Xue and Pierre Jean).

The experiment did not contain a global profiling run, but we show an example of extracting the unmodified peptides and using them in place of the profiling run.

Examples

```
head(spectronaut_input)
```

SpectronauttoMSstatsPTMFormat *Convert Spectronaut output into MSstatsPTM format*

Description

Converters label-free Spectronaut data into MSstatsPTM format. Requires PSM output from Spectronaut and a custom made annotation file, mapping the run name to the condition and bioreplicate. Can optionally take a seperate PSM file for a global profiling run. If no global profiling run provided, the function can extract the unmodified peptides from the PTM PSM file and use them as a global profiling run (not recommended).

Usage

```
SpectronauttoMSstatsPTMFormat(
  input,
  annotation = NULL,
  fasta_path = NULL,
  protein_input = NULL,
  annotation_protein = NULL,
  use_unmod_peptides = FALSE,
  intensity = "PeakArea",
  mod_id = "\\[Phospho \\(STY\\)\\]",
  fasta_protein_name = "uniprot_iso",
  remove_other_mods = TRUE,
  filter_with_Qvalue = TRUE,
  qvalue_cutoff = 0.01,
  useUniquePeptide = TRUE,
  removeFewMeasurements = TRUE,
  removeProtein_with1Feature = FALSE,
  summaryforMultipleRows = max,
  use_log_file = TRUE,
  append = FALSE,
  verbose = TRUE,
  log_file_path = NULL
)
```

Arguments

input	name of Spectronaut PTM 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.
fasta_path	string containing path to the corresponding fasta file for the modified peptide dataset.
protein_input	name of Spectronaut global protein output, which is as in the same format as `input` parameter.
annotation_protein	name of annotation file for global protein data, in the same format as above.
use_unmod_peptides	If `protein_input` is not provided, unmodified peptides can be extracted from `input` to be used in place of a global profiling run. Default is `FALSE`.
intensity	'PeakArea'(default) uses not normalized peak area. 'NormalizedPeakArea' uses peak area normalized by Spectronaut. Default is NULL
mod_id	Character that indicates the modification of interest. Default is `\\(Phospho\\)`. Note `\\` must be included before special characters.

`fasta_protein_name`
: Name of fasta column that matches with protein name in evidence file. Default is `uniprot_iso`.

`remove_other_mods`
: Remove peptides which include modfications other than the one listed in `mod_id`. Default is `TRUE`. For example, in an experiment targeting Phosphorylation, setting this parameter to `TRUE` would remove peptides like (Acetyl (Protein N-term))AAAAPDSRVS(Phospho (STY))EEENLK. Set this parameter to `FALSE` to keep peptides with extraneous modifications.

`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.

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

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

Examples

```
head(spectronaut_input)
head(spectronaut_annotation)

msstats_input = SpectronauttoMSstatsPTMFormat(spectronaut_input,
                  annotation=spectronaut_annotation,
            fasta_path=system.file("extdata", "spectronaut_fasta.fasta", package="MSstatsPTM"),
```

```
                    use_unmod_peptides=TRUE,
                    mod_id = "\\[Phospho \\(STY\\)\\]",
                    fasta_protein_name = "uniprot_iso"
                    )

head(msstats_input$PTM)
head(msstats_input$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

∗ **datasets**
    fragpipe_annotation, 16
    fragpipe_annotation_protein, 17
    fragpipe_input, 18
    fragpipe_input_protein, 18
    maxq_lf_annotation, 27
    maxq_lf_evidence, 28
    maxq_tmt_annotation, 29
    maxq_tmt_evidence, 30
    pd_annotation, 38
    pd_psm_input, 38
    pd_testing_output, 39
    raw.input, 48
    raw.input.tmt, 49
    spectronaut_annotation, 52
    spectronaut_input, 53
    summary.data, 56
    summary.data.tmt, 57
.fixTerminus, 3

annotSite, 3

dataProcess, *7*
dataProcessPlotsPTM, 4
dataProcessPTM, 6
dataSummarizationPTM, *5*, 7, *25*
dataSummarizationPTM_TMT, *5*, 10, *25*
designSampleSizePTM, 12
DIANNtoMSstatsPTMFormat, 13

fragpipe_annotation, 16
fragpipe_annotation_protein, 17
fragpipe_input, 18
fragpipe_input_protein, 18
FragPipetoMSstatsPTMFormat, 19

groupComparisonPlotsPTM, 22
groupComparisonPTM, *23*, 24

locateMod, 26
locatePTM, 27

maxq_lf_annotation, 27
maxq_lf_evidence, 28
maxq_tmt_annotation, 29
maxq_tmt_evidence, 30
MaxQtoMSstatsPTMFormat, 30
MetamorpheusToMSstatsPTMFormat, 34
MSstatsPTMSiteLocator, 36

pd_annotation, 38
pd_psm_input, 38
pd_testing_output, 39
PDtoMSstatsPTMFormat, 40
ProgenesistoMSstatsPTMFormat, 43
ProteinProspectortoMSstatsPTMFormat, 44
proteinSummarization, *7*
PStoMSstatsPTMFormat, 47

raw.input, 48, *56*
raw.input.tmt, 49, *57*

SkylinetoMSstatsPTMFormat, 50
spectronaut_annotation, 52
spectronaut_input, 53
SpectronauttoMSstatsPTMFormat, 53
summary.data, 56
summary.data.tmt, 57

tidyFasta, 58