Package: MSstatsConvert (via r-universe)

July 6, 2024

Title Import Data from Various Mass Spectrometry Signal Processing Tools to MSstats Format

Version 1.15.1

Description MSstatsConvert provides tools for importing reports of Mass Spectrometry data processing tools into R format suitable for statistical analysis using the MSstats and MSstatsTMT packages.

License Artistic-2.0

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.0

biocViews MassSpectrometry, Proteomics, Software, DataImport, QualityControl

Depends R (>= 4.0)

Imports data.table, log4r, methods, checkmate, utils, stringi

Suggests tinytest, covr, knitr, rmarkdown

Collate 'clean_Metamorpheus.R' 'clean_DIANN.R' 'clean_Philosopher.R' 'clean_Spectronaut.R' 'clean_SpectroMine.R' 'clean_Skyline.R' 'clean_ProteomeDiscoverer.R' 'clean_Progenesis.R' 'clean_OpenSWATH.R' 'clean_OpenMS.R' 'clean_MaxQuant.R' 'clean_DIAUmpire.R' 'MSstatsConvert_core_functions.R' 'converters_DIANNtoMSstatsFormat.R' 'converters_DIAUmpiretoMSstatsFormat.R' 'converters_FragPipetoMSstatsFormat.R' 'converters_MaxQtoMSstatsFormat.R' 'converters_MetamorpheusToMSstatsFormat.R' 'converters_OpenMStoMSstatsFormat.R' 'converters_OpenSWATHtoMSstatsFormat.R' 'converters_PDtoMSstatsFormat.R' 'converters_ProgenesistoMSstatsFormat.R' 'converters_SkylinetoMSstatsFormat.R'

'converters_SpectronauttoMSstatsFormat.R'
'utils_MSstatsConvert.R' 'utils_annotation.R'
'utils_balanced_design.R' 'utils_checks.R' 'utils_classes.R'
'utils_clean_features.R' 'utils_documentation.R'
'utils_dt_operations.R' 'utils_filtering.R' 'utils_fractions.R'
'utils_logging.R' 'utils_shared_peptides.R'

VignetteBuilder knitr

Repository https://bioc.r-universe.dev

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

RemoteRef HEAD

RemoteSha b4db270f6923ce819b5a64d801a5626d9a8b4af8

Contents

`.cleanRawPD` *Clean raw Proteome Discoverer data*

Description

Clean raw Proteome Discoverer data

Usage

```
.cleanRawPD(
  msstats_object,
  quantification_column,
  protein_id_column,
  sequence_column,
  remove_shared,
  remove_protein_groups = TRUE,
  intensity_columns_regexp = "Abundance"
)
```

Arguments

`msstats_object` an object of class `MSstatsSpectroMineFiles`.

`quantification_column`
chr, name of a column used for quantification.

`protein_id_column`
chr, name of a column with protein IDs.

`sequence_column`
chr, name of a column with peptide sequences.

`remove_shared` lgl, if TRUE, shared peptides will be removed.

`remove_protein_groups`
if TRUE, proteins with numProteins > 1 will be removed.

`intensity_columns_regexp`
regular expressions that defines intensity columns. Defaults to "Abundance", which means that columns that contain the word "Abundance" will be treated as corresponding to intensities for different channels.

Value

data.table

.validatePDTMTInputColumns

Helper method to validate input has necessary columns

Description

Helper method to validate input has necessary columns

Usage

```
.validatePDTMTInputColumns(
  pd_input,
  protein_id_column,
  num_proteins_column,
  channels
)
```

Arguments

`pd_input`	data.frame input
`protein_id_column`	column name for protein passed from user
`num_proteins_column`	column name for number of protein groups passed from user
`channels`	list of column names for channels

as.data.frame.MSstatsValidated

Convert output of converters to data.frame

Description

Convert output of converters to data.frame

Usage

```
## S3 method for class 'MSstatsValidated'
as.data.frame(x, ...)
```

Arguments

`x`	object of class MSstatsValidated
`...`	Additional arguments to be passed to or from other methods.

Value

data.frame

as.data.table.MSstatsValidated *Convert output of converters to data.table*

Description

Convert output of converters to data.table

Usage

```
## S3 method for class 'MSstatsValidated'
as.data.table(x, ...)
```

Arguments

x	object of class MSstatsValidated
...	Additional arguments to be passed to or from other methods.

Value

data.tables

DIANNtoMSstatsFormat *Import Diann files*

Description

Import Diann files

Usage

```
DIANNtoMSstatsFormat(
  input,
  annotation = NULL,
  global_qvalue_cutoff = 0.01,
  qvalue_cutoff = 0.01,
  pg_qvalue_cutoff = 0.01,
  useUniquePeptide = TRUE,
  removeFewMeasurements = TRUE,
  removeOxidationMpeptides = TRUE,
  removeProtein_with1Feature = TRUE,
  use_log_file = TRUE,
  append = FALSE,
  verbose = TRUE,
  log_file_path = NULL,
  MBR = TRUE,
  quantificationColumn = "FragmentQuantCorrected",
  ...
)
```

Arguments

`input` name of MSstats input report from Diann, which includes feature-level data.

`annotation` name of 'annotation.txt' data which includes Condition, BioReplicate, Run.

`global_qvalue_cutoff` The global qvalue cutoff

`qvalue_cutoff` local qvalue cutoff for library

`pg_qvalue_cutoff` local qvalue cutoff for protein groups Run should be the same as filename.

`useUniquePeptide` should unique pepties 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

`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.

`MBR` True if analysis was done with match between runs

`quantificationColumn` Use 'FragmentQuantCorrected'(default) column for quantified intensities. 'FragmentQuantRaw' can be used instead.

`...` additional parameters to `data.table::fread`.

Value

data.frame in the MSstats required format.

Author(s)

Elijah Willie

Examples

```
input_file_path = system.file("tinytest/raw_data/DIANN/diann_input.tsv",
                              package="MSstatsConvert")
annotation_file_path = system.file("tinytest/raw_data/DIANN/annotation.csv",
                              package = "MSstatsConvert")
input = data.table::fread(input_file_path)
annot = data.table::fread(annotation_file_path)
output = DIANNtoMSstatsFormat(input, annotation = annot, MBR = FALSE,
                              use_log_file = FALSE)
head(output)
```

DIAUmpiretoMSstatsFormat

Import DIA-Umpire files

Description

Import DIA-Umpire files

Usage

```
DIAUmpiretoMSstatsFormat(
  raw.frag,
  raw.pep,
  raw.pro,
  annotation,
  useSelectedFrag = TRUE,
  useSelectedPep = TRUE,
  removeFewMeasurements = TRUE,
  removeProtein_with1Feature = FALSE,
  summaryforMultipleRows = max,
  use_log_file = TRUE,
  append = FALSE,
  verbose = TRUE,
  log_file_path = NULL,
  ...
)
```

Arguments

raw.frag	name of FragSummary_date.xls data, which includes feature-level data.
raw.pep	name of PeptideSummary_date.xls data, which includes selected fragments information.
raw.pro	name of ProteinSummary_date.xls data, which includes selected peptides information.
annotation	name of annotation data which includes Condition, BioReplicate, Run information.

`useSelectedFrag`
TRUE will use the selected fragment for each peptide. 'Selected_fragments' column is required.

`useSelectedPep` TRUE will use the selected peptide for each protein. 'Selected_peptides' column is required.

`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.

`...` additional parameters to `data.table::fread`.

Value

data.frame in the MSstats required format.

Author(s)

Meena Choi, Olga Vitek

Examples

```
diau_frag = system.file("tinytest/raw_data/DIAUmpire/dia_frag.csv",
                              package = "MSstatsConvert")
diau_pept = system.file("tinytest/raw_data/DIAUmpire/dia_pept.csv",
                              package = "MSstatsConvert")
diau_prot = system.file("tinytest/raw_data/DIAUmpire/dia_prot.csv",
                              package = "MSstatsConvert")
annot = system.file("tinytest/raw_data/DIAUmpire/annot_diau.csv",
                     package = "MSstatsConvert")
diau_frag = data.table::fread(diau_frag)
diau_pept = data.table::fread(diau_pept)
diau_prot = data.table::fread(diau_prot)
annot = data.table::fread(annot)
diau_frag = diau_frag[, lapply(.SD, function(x) if (is.integer(x)) as.numeric(x) else x)]
# In case numeric columns are not interpreted correctly
```

```
diau_imported = DIAUmpiretoMSstatsFormat(diau_frag, diau_pept, diau_prot,
                                         annot, use_log_file = FALSE)
head(diau_imported)
```

FragPipetoMSstatsFormat

Import FragPipe files

Description

Import FragPipe files

Usage

```
FragPipetoMSstatsFormat(
  input,
  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 FragPipe msstats.csv export. ProteinName, PeptideSequence, PrecursorCharge, FragmentIon, ProductCharge, IsotopeLabelType, Condition, BioReplicate, Run, Intensity are required.

`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.

`...` additional parameters to `data.table::fread`.

Value

data.frame in the MSstats required format.

Author(s)

Devon Kohler

Examples

```
fragpipe_raw = system.file("tinytest/raw_data/FragPipe/fragpipe_input.csv",
                           package = "MSstatsConvert")
fragpipe_raw = data.table::fread(fragpipe_raw)
fragpipe_imported = FragPipetoMSstatsFormat(fragpipe_raw, use_log_file = FALSE)
head(fragpipe_imported)
```

---

getInputFile *Get one of files contained in an instance of* `MSstatsInputFiles` *class.*

Description

Get one of files contained in an instance of `MSstatsInputFiles` class.

Usage

```
getInputFile(msstats_object, file_type)

## S4 method for signature 'MSstatsInputFiles'
getInputFile(msstats_object, file_type = "input")

## S4 method for signature 'MSstatsPhilosopherFiles'
getInputFile(msstats_object, file_type = "input")
```

Arguments

`msstats_object` object that inherits from `MSstatsPhilosopherFiles` class.

`file_type` character name of a type file. Usually equal to "input".

Value

data.table

data.table

data.table

Examples

```
evidence_path = system.file("tinytest/raw_data/MaxQuant/mq_ev.csv",
                            package = "MSstatsConvert")
pg_path = system.file("tinytest/raw_data/MaxQuant/mq_pg.csv",
                      package = "MSstatsConvert")
evidence = read.csv(evidence_path)
pg = read.csv(pg_path)
imported = MSstatsImport(list(evidence = evidence, protein_groups = pg),
                         "MSstats", "MaxQuant")
class(imported)
head(getInputFile(imported, "evidence"))
```

MaxQtoMSstatsFormat *Import MaxQuant files*

Description

Import MaxQuant files

Usage

```
MaxQtoMSstatsFormat(
  evidence,
  annotation,
  proteinGroups,
  proteinID = "Proteins",
  useUniquePeptide = TRUE,
  summaryforMultipleRows = max,
  removeFewMeasurements = 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.

`annotation` name of ’annotation.txt’ data which includes Raw.file, Condition, BioReplicate, Run, IsotopeLabelType information.

`proteinGroups` name of ’proteinGroups.txt’ data. It needs to matching protein group ID. If proteinGroups=NULL, use ’Proteins’ column in ’evidence.txt’.

`proteinID` ’Proteins’(default) or ’Leading.razor.protein’ for Protein ID.

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

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

`removeFewMeasurements`
TRUE (default) will remove the features that have 1 or 2 measurements across runs.

`removeMpeptides`
TRUE will remove the peptides including ’M’ sequence. FALSE is default.

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

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

`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.

`...` additional parameters to `data.table::fread`.

Value

data.frame in the MSstats required format.

Note

Warning: MSstats does not support for metabolic labeling or iTRAQ experiments.

Author(s)

Meena Choi, Olga Vitek.

Examples

```
mq_ev = data.table::fread(system.file("tinytest/raw_data/MaxQuant/mq_ev.csv",
                                      package = "MSstatsConvert"))
mq_pg = data.table::fread(system.file("tinytest/raw_data/MaxQuant/mq_pg.csv",
                                      package = "MSstatsConvert"))
annot = data.table::fread(system.file("tinytest/raw_data/MaxQuant/annotation.csv",
                                      package = "MSstatsConvert"))
maxq_imported = MaxQtoMSstatsFormat(mq_ev, annot, mq_pg, use_log_file = FALSE)
head(maxq_imported)
```

MetamorpheusToMSstatsFormat
Import Metamorpheus files

Description

Import Metamorpheus files

Usage

```
MetamorpheusToMSstatsFormat(
  input,
  annotation = NULL,
  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.

`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.

...
: additional parameters to `data.table::fread`.

Value

data.frame in the MSstats required format.

Author(s)

Anthony Wu

Examples

```
input = system.file("tinytest/raw_data/Metamorpheus/AllQuantifiedPeaks.tsv",
                                package = "MSstatsConvert")
input = data.table::fread(input)
annot = system.file("tinytest/raw_data/Metamorpheus/Annotation.tsv",
                                package = "MSstatsConvert")
annot = data.table::fread(annot)
metamorpheus_imported = MSstatsConvert:::MetamorpheusToMSstatsFormat(input, annotation = annot)
head(metamorpheus_imported)
```

MSstatsBalancedDesign *Creates balanced design by removing overlapping fractions and filling incomplete rows*

Description

Creates balanced design by removing overlapping fractions and filling incomplete rows

Usage

```
MSstatsBalancedDesign(
  input,
  feature_columns,
  fill_incomplete = TRUE,
  handle_fractions = TRUE,
  fix_missing = NULL,
  remove_few = TRUE
)
```

Arguments

`input` `data.table` processed by the `MSstatsPreprocess` function

`feature_columns`
str, names of columns that define spectral features

`fill_incomplete`
if TRUE (default), Intensity values for missing runs will be added as NA

`handle_fractions`
if TRUE (default), overlapping fractions will be resolved

`fix_missing` str, optional. Defaults to NULL, which means no action. If not NULL, must be one of the options: "zero_to_na" or "na_to_zero". If "zero_to_na", Intensity values equal exactly to 0 will be converted to NA. If "na_to_zero", missing values will be replaced by zeros.

`remove_few` lgl, if TRUE, features with one or two measurements across runs will be removed.

Value

data.frame of class `MSstatsValidated`

Examples

```
unbalanced_data = system.file("tinytest/raw_data/unbalanced_data.csv",
                              package = "MSstatsConvert")
unbalanced_data = data.table::as.data.table(read.csv(unbalanced_data))
balanced = MSstatsBalancedDesign(unbalanced_data,
                                 c("PeptideSequence", "PrecursorCharge",
                                   "FragmentIon", "ProductCharge"))
dim(balanced) # Now balanced has additional rows (with Intensity = NA)
# for runs that were not included in the unbalanced_data table
```

MSstatsClean *Clean files generated by a signal processing tools.*

Description

Clean files generated by a signal processing tools.

Clean DIAUmpire files

Clean MaxQuant files

Clean OpenMS files

Clean OpenSWATH files

Clean Progenesis files

Clean ProteomeDiscoverer files

Clean Skyline files

Clean SpectroMine files

Clean Spectronaut files

Clean Philosopher files

Clean DIA-NN files

Clean Metamorpheus files

Usage

```
MSstatsClean(msstats_object, ...)

## S4 method for signature 'MSstatsDIAUmpireFiles'
MSstatsClean(msstats_object, use_frag, use_pept)

## S4 method for signature 'MSstatsMaxQuantFiles'
MSstatsClean(
  msstats_object,
  protein_id_col,
  remove_by_site = FALSE,
  channel_columns = "Reporterintensitycorrected"
)

## S4 method for signature 'MSstatsOpenMSFiles'
MSstatsClean(msstats_object)

## S4 method for signature 'MSstatsOpenSWATHFiles'
MSstatsClean(msstats_object)

## S4 method for signature 'MSstatsProgenesisFiles'
MSstatsClean(msstats_object, runs, fix_colnames = TRUE)
```

```
## S4 method for signature 'MSstatsProteomeDiscovererFiles'
MSstatsClean(
  msstats_object,
  quantification_column,
  protein_id_column,
  sequence_column,
  remove_shared,
  remove_protein_groups = TRUE,
  intensity_columns_regexp = "Abundance"
)

## S4 method for signature 'MSstatsSkylineFiles'
MSstatsClean(msstats_object)

## S4 method for signature 'MSstatsSpectroMineFiles'
MSstatsClean(msstats_object)

## S4 method for signature 'MSstatsSpectronautFiles'
MSstatsClean(msstats_object, intensity)

## S4 method for signature 'MSstatsPhilosopherFiles'
MSstatsClean(
  msstats_object,
  protein_id_col,
  peptide_id_col,
  channels,
  remove_shared_peptides
)

## S4 method for signature 'MSstatsDIANNFiles'
MSstatsClean(
  msstats_object,
  MBR = TRUE,
  quantificationColumn = "FragmentQuantCorrected"
)

## S4 method for signature 'MSstatsMetamorpheusFiles'
MSstatsClean(msstats_object)
```

Arguments

`msstats_object` object that inherits from `MSstatsInputFiles` class.

`...` additional parameter to specific cleaning functions.

`use_frag` TRUE will use the selected fragment for each peptide. 'Selected_fragments' column is required.

`use_pept` TRUE will use the selected fragment for each protein 'Selected_peptides' column is required.

`protein_id_col` character, name of a column with names of proteins.

`remove_by_site` logical, if TRUE, proteins only identified by site will be removed.

`channel_columns`
character, regular expression that identifies channel columns in TMT data.

`runs` chr, vector of Run labels.

`fix_colnames` lgl, if TRUE, one of the rows will be used as colnames.

`quantification_column`
chr, name of a column used for quantification.

`protein_id_column`
chr, name of a column with protein IDs.

`sequence_column`
chr, name of a column with peptide sequences.

`remove_shared` lgl, if TRUE, shared peptides will be removed.

`remove_protein_groups`
if TRUE, proteins with numProteins > 1 will be removed.

`intensity_columns_regexp`
regular expressions that defines intensity columns. Defaults to "Abundance", which means that columns that contain the word "Abundance" will be treated as corresponding to intensities for different channels.

`intensity` chr, specifies which column will be used for Intensity.

`peptide_id_col` character name of a column that identifies peptides

`channels` character vector of channel labels

`remove_shared_peptides`
logical, if TRUE, shared peptides will be removed based on the IsUnique column from Philosopher output

`MBR` True if analysis was done with match between runs

`quantificationColumn`
Use 'FragmentQuantCorrected'(default) column for quantified intensities. 'FragmentQuantRaw' can be used instead.

Value

data.table

data.table

data.table

data.table

data.table

data.table

data.table

data.table

data.table

data.table

data.table

data.table

Examples

```
evidence_path = system.file("tinytest/raw_data/MaxQuant/mq_ev.csv",
                            package = "MSstatsConvert")
pg_path = system.file("tinytest/raw_data/MaxQuant/mq_pg.csv",
                      package = "MSstatsConvert")
evidence = read.csv(evidence_path)
pg = read.csv(pg_path)
imported = MSstatsImport(list(evidence = evidence, protein_groups = pg),
                         "MSstats", "MaxQuant")
cleaned_data = MSstatsClean(imported, protein_id_col = "Proteins")
head(cleaned_data)
```

MSstatsConvert — *MSstatsConvert: An R Package to Convert Data from Mass Spectrometry Signal Processing Tools to MSstats Format*

Description

MSstatsConvert helps convert data from different types of mass spectrometry experiments and signal processing tools to a format suitable for statistical analysis with the MSstats and MSstatsTMT packages.

Main functions

MSstatsLogsSettings for logs management, MSstatsImport for importing files created by signal processing tools, MSstatsClean for re-formatting imported files into a consistent format, MSstatsPreprocess for preprocessing cleaned files, MSstatsBalancedDesign for handling fractions and creating balanced data.

MSstatsImport — *Import files from signal processing tools.*

Description

Import files from signal processing tools.

Usage

```
MSstatsImport(input_files, type, tool, tool_version = NULL, ...)
```

Arguments

input_files — list of paths to input files or `data.frame` objects. Interpretation of this parameter depends on values of parameters `type` and `tool`.

type — chr, "MSstats" or "MSstatsTMT".

tool — chr, name of a signal processing tool that generated input files.

tool_version — not implemented yet. In the future, this parameter will allow handling different versions of each signal processing tools.

... — optional additional parameters to `data.table::fread`.

Value

an object of class `MSstatsInputFiles`.

Examples

```
evidence_path = system.file("tinytest/raw_data/MaxQuant/mq_ev.csv",
                            package = "MSstatsConvert")
pg_path = system.file("tinytest/raw_data/MaxQuant/mq_pg.csv",
                      package = "MSstatsConvert")
evidence = read.csv(evidence_path)
pg = read.csv(pg_path)
imported = MSstatsImport(list(evidence = evidence, protein_groups = pg),
                         "MSstats", "MaxQuant")
class(imported)
head(getInputFile(imported, "evidence"))
```

MSstatsLogsSettings *Set how MSstats will log information from data processing*

Description

Set how MSstats will log information from data processing

Usage

```
MSstatsLogsSettings(
  use_log_file = TRUE,
  append = FALSE,
  verbose = TRUE,
  log_file_path = NULL,
  base = "MSstats_log_",
  pkg_name = "MSstats"
)
```

Arguments

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.

base start of the file name.

pkg_name currently "MSstats", "MSstatsPTM" or "MSstatsTMT". Each package can use its own separate log settings.

Value

TRUE invisibly in case of successful logging setup.

Examples

```
# No logging and no messages
MSstatsLogsSettings(FALSE, FALSE, FALSE)
# Log, but do not display messages
MSstatsLogsSettings(TRUE, FALSE, FALSE)
# Log to an existing file
file.create("new_log.log")
MSstatsLogsSettings(TRUE, TRUE, log_file_path = "new_log.log")
# Do not log, but display messages
MSstatsLogsSettings(FALSE)
```

MSstatsMakeAnnotation *Create annotation*

Description

Create annotation

Usage

```
MSstatsMakeAnnotation(input, annotation, ...)
```

Arguments

input data.table preprocessed by the MSstatsClean function

annotation data.table

... key-value pairs, where keys are names of columns of annotation

Value

data.table

Examples

```
evidence_path = system.file("tinytest/raw_data/MaxQuant/mq_ev.csv",
                            package = "MSstatsConvert")
pg_path = system.file("tinytest/raw_data/MaxQuant/mq_pg.csv",
                      package = "MSstatsConvert")
evidence = read.csv(evidence_path)
pg = read.csv(pg_path)
imported = MSstatsImport(list(evidence = evidence, protein_groups = pg),
                         "MSstats", "MaxQuant")
cleaned_data = MSstatsClean(imported, protein_id_col = "Proteins")
annot_path = system.file("tinytest/raw_data/MaxQuant/annotation.csv",
                         package = "MSstatsConvert")
mq_annot = MSstatsMakeAnnotation(cleaned_data, read.csv(annot_path),
                                 Run = "Rawfile")
head(mq_annot)
```

MSstatsPreprocess *Preprocess outputs from MS signal processing tools for analysis with MSstats*

Description

Preprocess outputs from MS signal processing tools for analysis with MSstats

Usage

```
MSstatsPreprocess(
  input,
  annotation,
  feature_columns,
  remove_shared_peptides = TRUE,
  remove_single_feature_proteins = TRUE,
  feature_cleaning = list(remove_features_with_few_measurements = TRUE,
    summarize_multiple_psms = max),
  score_filtering = list(),
  exact_filtering = list(),
  pattern_filtering = list(),
  columns_to_fill = list(),
  aggregate_isotopic = FALSE,
  ...
)
```

Arguments

`input` data.table processed by the MSstatsClean function.

`annotation` annotation file generated by a signal processing tool.

`feature_columns` character vector of names of columns that define spectral features.

`remove_shared_peptides` logical, if TRUE shared peptides will be removed.

`remove_single_feature_proteins` logical, if TRUE, proteins that only have one feature will be removed.

`feature_cleaning` named list with maximum two (for `MSstats` converters) or three (for `MSstatsTMT` converter) elements. If `handle_few_measurements` is set to "remove", feature with less than three measurements will be removed (otherwise it should be equal to "keep"). `summarize_multiple_psms` is a function that will be used to aggregate multiple feature measurements in a run. It should return a scalar and accept an `na.rm` parameter. For `MSstatsTMT` converters, setting `remove_psms_with_any_missing` will remove features which have missing values in a run from that run.

`score_filtering` a list of named lists that specify filtering options. Details are provided in the vignette.

`exact_filtering` a list of named lists that specify filtering options. Details are provided in the vignette.

`pattern_filtering` a list of named lists that specify filtering options. Details are provided in the vignette.

`columns_to_fill` a named list of scalars. If provided, columns with names defined by the names of this list and values corresponding to its elements will be added to the output `data.frame`.

`aggregate_isotopic` logical. If `TRUE`, isotopic peaks will by summed.

`...` additional parameters to `data.table::fread`.

Value

data.table

Examples

```
evidence_path = system.file("tinytest/raw_data/MaxQuant/mq_ev.csv",
                            package = "MSstatsConvert")
pg_path = system.file("tinytest/raw_data/MaxQuant/mq_pg.csv",
                      package = "MSstatsConvert")
evidence = read.csv(evidence_path)
pg = read.csv(pg_path)
```

```
imported = MSstatsImport(list(evidence = evidence, protein_groups = pg),
                         "MSstats", "MaxQuant")
cleaned_data = MSstatsClean(imported, protein_id_col = "Proteins")
annot_path = system.file("tinytest/raw_data/MaxQuant/annotation.csv",
                         package = "MSstatsConvert")
mq_annot = MSstatsMakeAnnotation(cleaned_data, read.csv(annot_path),
                                 Run = "Rawfile")

# To filter M-peptides and oxidatin peptides
m_filter = list(col_name = "PeptideSequence", pattern = "M",
                filter = TRUE, drop_column = FALSE)
oxidation_filter = list(col_name = "Modifications", pattern = "Oxidation",
                        filter = TRUE, drop_column = TRUE)
msstats_format = MSstatsPreprocess(
cleaned_data, mq_annot,
feature_columns = c("PeptideSequence", "PrecursorCharge"),
columns_to_fill = list(FragmentIon = NA, ProductCharge = NA),
pattern_filtering = list(oxidation = oxidation_filter, m = m_filter)
)
# Output in the standard MSstats format
head(msstats_format)
```

MSstatsSaveSessionInfo

Save session information

Description

Save session information

Usage

```
MSstatsSaveSessionInfo(
  path = NULL,
  append = TRUE,
  base = "MSstats_session_info_"
)
```

Arguments

path	optional path to output file. If not provided, "MSstats_session_info" and current timestamp will be used as a file name
append	if TRUE and file given by the path parameter already exists, session info will be appended to the file
base	beginning of a file name

Value

TRUE invisibly after session info was saved

Examples

```
MSstatsSaveSessionInfo("session_info.txt")
MSstatsSaveSessionInfo("session_info.txt", base = "MSstatsTMT_session_info_")
```

OpenMStoMSstatsFormat *Import OpenMS files*

Description

Import OpenMS files

Usage

```
OpenMStoMSstatsFormat(
  input,
  annotation = NULL,
  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 MSstats input report from OpenMS, which includes feature(peptide ion)-level data.

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

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

`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.

`...`
: additional parameters to `data.table::fread`.

Value

data.frame in the MSstats required format.

Author(s)

Meena Choi, Olga Vitek.

Examples

```
openms_raw = data.table::fread(system.file("tinytest/raw_data/OpenMS/openms_input.csv",
                                           package = "MSstatsConvert"))
openms_imported = OpenMStoMSstatsFormat(openms_raw, use_log_file = FALSE)
head(openms_imported)
```

`OpenSWATHtoMSstatsFormat` *Import OpenSWATH files*

Description

Import OpenSWATH files

Usage

```
OpenSWATHtoMSstatsFormat(
  input,
  annotation,
  filter_with_mscore = TRUE,
  mscore_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 MSstats input report from OpenSWATH, which includes feature-level data.

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

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

`mscore_cutoff` Cutoff for m_score. Default is 0.01.

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

`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.

`...` additional parameters to `data.table::fread`.

Value

data.frame in the MSstats required format.

Author(s)

Meena Choi, Olga Vitek.

Examples

```
os_raw = system.file("tinytest/raw_data/OpenSWATH/openswath_input.csv",
                              package = "MSstatsConvert")
annot = system.file("tinytest/raw_data/OpenSWATH/annot_os.csv",
                    package = "MSstatsConvert")
os_raw = data.table::fread(os_raw)
annot = data.table::fread(annot)

os_imported = OpenSWATHtoMSstatsFormat(os_raw, annot, use_log_file = FALSE)
head(os_imported)
```

PDtoMSstatsFormat *Import Proteome Discoverer files*

Description

Import Proteome Discoverer files

Usage

```
PDtoMSstatsFormat(
  input,
  annotation,
  useNumProteinsColumn = FALSE,
  useUniquePeptide = TRUE,
  summaryforMultipleRows = max,
  removeFewMeasurements = TRUE,
  removeOxidationMpeptides = FALSE,
  removeProtein_with1Peptide = FALSE,
  which.quantification = "Precursor.Area",
  which.proteinid = "Protein.Group.Accessions",
  which.sequence = "Sequence",
  use_log_file = TRUE,
  append = FALSE,
  verbose = TRUE,
  log_file_path = NULL,
  ...
)
```

Arguments

`input` PD report or a path to it.

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

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

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

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

`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.

`which.sequence`
: Use 'Sequence'(default) column for peptide sequence. 'Annotated.Sequence' 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 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.

`...`
: additional parameters to `data.table::fread`.

Value

data.frame in the MSstats required format.

Author(s)

Meena Choi, Olga Vitek

Examples

```
pd_raw = system.file("tinytest/raw_data/PD/pd_input.csv",
                     package = "MSstatsConvert")
annot = system.file("tinytest/raw_data/PD/annot_pd.csv",
                    package = "MSstatsConvert")
```

```
pd_raw = data.table::fread(pd_raw)
annot = data.table::fread(annot)

pd_imported = PDtoMSstatsFormat(pd_raw, annot, use_log_file = FALSE)
head(pd_imported)
```

ProgenesistoMSstatsFormat

Import Progenesis files

Description

Import Progenesis files

Usage

```
ProgenesistoMSstatsFormat(
  input,
  annotation,
  useUniquePeptide = TRUE,
  summaryforMultipleRows = max,
  removeFewMeasurements = TRUE,
  removeOxidationMpeptides = FALSE,
  removeProtein_with1Peptide = FALSE,
  use_log_file = TRUE,
  append = FALSE,
  verbose = TRUE,
  log_file_path = NULL,
  ...
)
```

Arguments

`input` name of Progenesis output, which is wide-format. 'Accession', 'Sequence', 'Modification', 'Charge' and one column for each run are required.

`annotation` name of 'annotation.txt' or 'annotation.csv' data which includes Condition, BioReplicate, Run information. It will be matched with the column name of input for MS runs.

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

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

`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.

`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.

`...` additional parameters to `data.table::fread`.

Value

data.frame in the MSstats required format.

Author(s)

Meena Choi, Olga Vitek, Ulrich Omasits

Examples

```
progenesis_raw = system.file("tinytest/raw_data/Progenesis/progenesis_input.csv",
                             package = "MSstatsConvert")
annot = system.file("tinytest/raw_data/Progenesis/progenesis_annot.csv",
                    package = "MSstatsConvert")
progenesis_raw = data.table::fread(progenesis_raw)
annot = data.table::fread(annot)

progenesis_imported = ProgenesistoMSstatsFormat(progenesis_raw, annot,
                                                use_log_file = FALSE)
head(progenesis_imported)
```

SkylinetoMSstatsFormat *Import Skyline files*

Description

Import Skyline files

Usage

```
SkylinetoMSstatsFormat(
  input,
  annotation = NULL,
  removeiRT = TRUE,
  filter_with_Qvalue = TRUE,
  qvalue_cutoff = 0.01,
  useUniquePeptide = TRUE,
  removeFewMeasurements = TRUE,
  removeOxidationMpeptides = FALSE,
  removeProtein_with1Feature = FALSE,
  use_log_file = TRUE,
  append = FALSE,
  verbose = TRUE,
  log_file_path = NULL,
  ...
)
```

Arguments

`input` name of MSstats input report from Skyline, which includes feature-level data.

`annotation` name of 'annotation.txt' data which includes Condition, BioReplicate, Run. If annotation is already complete in Skyline, use annotation=NULL (default). It will use the annotation information from input.

`removeiRT` TRUE (default) will remove the proteins or peptides which are labeled 'iRT' in 'StandardType' column. FALSE will keep them.

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

`qvalue_cutoff` Cutoff for DetectionQValue. default is 0.01.

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

`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_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.

...
: additional parameters to `data.table::fread`.

Value

data.frame in the MSstats required format.

Author(s)

Meena Choi, Olga Vitek

Examples

```
skyline_raw = system.file("tinytest/raw_data/Skyline/skyline_input.csv",
                          package = "MSstatsConvert")
skyline_raw = data.table::fread(skyline_raw)
skyline_imported = SkylinetoMSstatsFormat(skyline_raw)
head(skyline_imported)
```

`SpectronauttoMSstatsFormat`

Import Spectronaut files

Description

Import Spectronaut files

Usage

```
SpectronauttoMSstatsFormat(
  input,
  annotation = NULL,
  intensity = "PeakArea",
  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 output, which is long-format. ProteinName, PeptideSequence, PrecursorCharge, FragmentIon, ProductCharge, IsotopeLabelType, Condition, BioReplicate, Run, Intensity, F.ExcludedFromQuantification are required. Rows with F.ExcludedFromQuantification=True will be removed.

`annotation` name of 'annotation.txt' data which includes Condition, BioReplicate, Run. If annotation is already complete in Spectronaut, use annotation=NULL (default). It will use the annotation information from input.

`intensity` 'PeakArea'(default) uses not normalized peak area. 'NormalizedPeakArea' uses peak area normalized by Spectronaut.

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

`qvalue_cutoff` Cutoff for EG.Qvalue. default is 0.01.

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

`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.

`...` additional parameters to `data.table::fread`.

Value

data.frame in the MSstats required format.

Author(s)

Meena Choi, Olga Vitek

Examples

```
spectronaut_raw = system.file(”tinytest/raw_data/Spectronaut/spectronaut_input.csv”,
                              package = ”MSstatsConvert”)
spectronaut_raw = data.table::fread(spectronaut_raw)
spectronaut_imported = SpectronauttoMSstatsFormat(spectronaut_raw, use_log_file = FALSE)
head(spectronaut_imported)
```

Index