Package 'MSstatsConvert'

October 14, 2021

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

Version 1.2.2

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 7.1.1

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

VignetteBuilder knitr

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

git_branch RELEASE_3_13

git_last_commit bcc4339

git_last_commit_date 2021-06-15

Date/Publication 2021-10-14

Author Mateusz Staniak [aut, cre],
Meena Choi [aut],
Ting Huang [aut],
Olga Vitek [aut]

Maintainer Mateusz Staniak <mtst@mstaniak.pl>

R topics documented:

`.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

`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")
```

Arguments

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

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

Value

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"))
```

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
)
```

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.

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

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)
```

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.

Value

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, MSstatsImport 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_")
```

Index