

# Package ‘MSstatsConvert’

November 17, 2025

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

**Version** 1.21.0

## **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** 7.3.2

**biocViews** MassSpectrometry, Proteomics, Software, DataImport,  
QualityControl

**Depends** R (>= 4.0)

**Imports** data.table, log4r, methods, checkmate, utils, stringi, Rcpp,  
parallel

**Suggests** tinytest, covr, knitr, arrow, rmarkdown

**LinkingTo** Rcpp

**Collate** 'clean\_ProteinProspector.R' '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'  
'RcppExports.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_ProgenesitoMSstatsFormat.R'
'converters_ProteinProspectortoMSstatsTMTFormat.R'
'converters_SkylinetoMSstatsFormat.R'
'converters_SpectronauttoMSstatsFormat.R'
'utils_MSstatsConvert.R' 'utils_annotation.R'
'utils_anomaly_score.R' 'utils_balanced_design.R'
'utils_checks.R' 'utils_classes.R' 'utils_clean_features.R'
'utils_data_health.R' 'utils_documentation.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** devel

**git\_last\_commit** c3ae371

**git\_last\_commit\_date** 2025-10-29

**Repository** Bioconductor 3.23

**Date/Publication** 2025-11-16

**Author** Mateusz Staniak [aut, cre],  
 Devon Kohler [aut],  
 Anthony Wu [aut],  
 Meena Choi [aut],  
 Ting Huang [aut],  
 Olga Vitek [aut]

**Maintainer** Mateusz Staniak <mtst@mstaniak.pl>

## Contents

|                                        |    |
|----------------------------------------|----|
| .addFractions . . . . .                | 4  |
| .adjustIntensities . . . . .           | 5  |
| .aggregatePSMstoPeptideIons . . . . .  | 5  |
| .checkAnnotation . . . . .             | 6  |
| .checkDDA . . . . .                    | 6  |
| .checkDuplicatedMeasurements . . . . . | 7  |
| .checkMSstatsParams . . . . .          | 7  |
| .checkMultiRun . . . . .               | 8  |
| .checkOverlappedFeatures . . . . .     | 8  |
| .cleanByFeature . . . . .              | 9  |
| .cleanRawDIANN . . . . .               | 9  |
| .cleanRawDIAUmpire . . . . .           | 10 |
| .cleanRawMaxQuant . . . . .            | 11 |
| .cleanRawMetamorpheus . . . . .        | 11 |
| .cleanRawOpenMS . . . . .              | 12 |
| .cleanRawOpenSWATH . . . . .           | 12 |
| .cleanRawPD . . . . .                  | 13 |
| .cleanRawPDMSstats . . . . .           | 14 |

|                                               |    |
|-----------------------------------------------|----|
| .cleanRawPDTMT . . . . .                      | 14 |
| .cleanRawPhilosopher . . . . .                | 15 |
| .cleanRawProgenesis . . . . .                 | 16 |
| .cleanRawSkyline . . . . .                    | 17 |
| .cleanRawSpectroMineTMT . . . . .             | 17 |
| .cleanRawSpectronaut . . . . .                | 18 |
| .countCommonFeatures . . . . .                | 18 |
| .fillValues . . . . .                         | 19 |
| .filterByPattern . . . . .                    | 19 |
| .filterByScore . . . . .                      | 20 |
| .filterExact . . . . .                        | 21 |
| .filterFewMeasurements . . . . .              | 21 |
| .filterManyColumns . . . . .                  | 22 |
| .filterOverlapped . . . . .                   | 23 |
| .findAvailable . . . . .                      | 23 |
| .fixBasicColumns . . . . .                    | 24 |
| .fixColumnTypes . . . . .                     | 24 |
| .fixMissingValues . . . . .                   | 25 |
| .getChannelColumns . . . . .                  | 25 |
| .getCorrectFraction . . . . .                 | 26 |
| .getDataTable . . . . .                       | 26 |
| .getFullDesign . . . . .                      | 27 |
| .getMissingRunsPerFeature . . . . .           | 27 |
| .getOverlappingFeatures . . . . .             | 28 |
| .handleFiltering . . . . .                    | 28 |
| .handleFractions . . . . .                    | 29 |
| .handleFractionsLF . . . . .                  | 29 |
| .handleFractionsTMT . . . . .                 | 30 |
| .handleIsotopicPeaks . . . . .                | 30 |
| .handleSharedPeptides . . . . .               | 31 |
| .handleSingleFeaturePerProtein . . . . .      | 31 |
| .logConverterOptions . . . . .                | 32 |
| .logSuccess . . . . .                         | 33 |
| .makeBalancedDesign . . . . .                 | 33 |
| .makeExactFilterMessage . . . . .             | 34 |
| .makeScoreFilterMessage . . . . .             | 34 |
| .mergeAnnotation . . . . .                    | 35 |
| .MSstatsFormat . . . . .                      | 36 |
| .nullAppender . . . . .                       | 36 |
| .onLoad . . . . .                             | 37 |
| .removeOverlappingFeatures . . . . .          | 37 |
| .removeSharedPeptides . . . . .               | 38 |
| .selectMSstatsColumns . . . . .               | 38 |
| .sharedParametersAmongConverters . . . . .    | 39 |
| .standardizeColnames . . . . .                | 40 |
| .summarizeMultipleMeasurements . . . . .      | 40 |
| .summarizeMultiplePSMs . . . . .              | 41 |
| .validateMSstatsConverterParameters . . . . . | 41 |

|                                               |    |
|-----------------------------------------------|----|
| as.data.frame.MSstatsValidated . . . . .      | 43 |
| as.data.table.MSstatsValidated . . . . .      | 43 |
| CheckDataHealth . . . . .                     | 44 |
| DIANNtoMSstatsFormat . . . . .                | 44 |
| DIAUmpiretoMSstatsFormat . . . . .            | 46 |
| FragPipetoMSstatsFormat . . . . .             | 48 |
| getDataTypes . . . . .                        | 50 |
| getInputFile . . . . .                        | 50 |
| MaxQtoMSstatsFormat . . . . .                 | 51 |
| MetamorpheusToMSstatsFormat . . . . .         | 53 |
| MSstatsAnomalyScores . . . . .                | 55 |
| MSstatsBalancedDesign . . . . .               | 56 |
| MSstatsClean . . . . .                        | 57 |
| MSstatsConvert . . . . .                      | 61 |
| MSstatsImport . . . . .                       | 61 |
| MSstatsInputFiles-class . . . . .             | 62 |
| MSstatsLogsSettings . . . . .                 | 63 |
| MSstatsMakeAnnotation . . . . .               | 64 |
| MSstatsPreprocess . . . . .                   | 65 |
| MSstatsSaveSessionInfo . . . . .              | 67 |
| OpenMStoMSstatsFormat . . . . .               | 68 |
| OpenSWATHtoMSstatsFormat . . . . .            | 69 |
| PDtoMSstatsFormat . . . . .                   | 71 |
| ProgenesistoMSstatsFormat . . . . .           | 73 |
| ProteinProspectortoMSstatsTMTFormat . . . . . | 75 |
| SkylinetoMSstatsFormat . . . . .              | 76 |
| SpectronauttoMSstatsFormat . . . . .          | 78 |

## **Index** **81**

---

|                      |                                                                 |
|----------------------|-----------------------------------------------------------------|
| <i>.addFractions</i> | <i>Add a Fraction column to the output of MSstatsPreprocess</i> |
|----------------------|-----------------------------------------------------------------|

---

### **Description**

Add a Fraction column to the output of MSstatsPreprocess

### **Usage**

```
.addFractions(input)
```

### **Arguments**

|       |                             |
|-------|-----------------------------|
| input | output of MSstatsPreprocess |
|-------|-----------------------------|

### **Value**

data.table

---

|                                 |                                                                      |
|---------------------------------|----------------------------------------------------------------------|
| <code>.adjustIntensities</code> | <i>Fix invalid intensities: infinite to NA, between 0 and 1 to 0</i> |
|---------------------------------|----------------------------------------------------------------------|

---

### **Description**

Fix invalid intensities: infinite to NA, between 0 and 1 to 0

### **Usage**

```
.adjustIntensities(input)
```

### **Arguments**

|                    |                         |
|--------------------|-------------------------|
| <code>input</code> | <code>data.table</code> |
|--------------------|-------------------------|

### **Value**

`data.table`

---

|                                          |                                                         |
|------------------------------------------|---------------------------------------------------------|
| <code>.aggregatePSMstoPeptideIons</code> | <i>Aggregate multiple PSMs to a single peptide ion.</i> |
|------------------------------------------|---------------------------------------------------------|

---

### **Description**

Aggregate multiple PSMs to a single peptide ion.

### **Usage**

```
.aggregatePSMstoPeptideIons(input, feature_columns, summary_function = sum)
```

### **Arguments**

|                               |                                                                |
|-------------------------------|----------------------------------------------------------------|
| <code>input</code>            | data.table preprocessed by one of the cleanRaw* functions.     |
| <code>feature_columns</code>  | chr, names of columns that define features.                    |
| <code>summary_function</code> | function that will be used to aggregate intensities if needed. |

### **Value**

`data.table`

---

|                               |                                         |
|-------------------------------|-----------------------------------------|
| <code>.checkAnnotation</code> | <i>Check if the annotation is valid</i> |
|-------------------------------|-----------------------------------------|

---

**Description**

Check if the annotation is valid

**Usage**

```
.checkAnnotation(input, annotation)
```

**Arguments**

|            |                                                          |
|------------|----------------------------------------------------------|
| input      | data processed by the MSstatsClean                       |
| annotation | annotation created by the MSstatsMakeAnnotation function |

**Value**

TRUE invisibly if the annotation is correct, throws an error otherwise

---

|                        |                                   |
|------------------------|-----------------------------------|
| <code>.checkDDA</code> | <i>Check validity of DDA data</i> |
|------------------------|-----------------------------------|

---

**Description**

Check validity of DDA data

**Usage**

```
.checkDDA(input)
```

**Arguments**

|       |                                                            |
|-------|------------------------------------------------------------|
| input | data.table preprocessed by one of the cleanRaw* functions. |
|-------|------------------------------------------------------------|

**Value**

logical  
logical, TRUE means that the input dataset comes from a DDA experiment

---

`.checkDuplicatedMeasurements`

*Check if there are duplicated measurements within run*

---

### **Description**

Check if there are duplicated measurements within run

### **Usage**

```
.checkDuplicatedMeasurements(input)
```

### **Arguments**

input                      output of MSstatsPreprocess

### **Value**

character vector of feature labels

---

`.checkMSstatsParams`

*Check validity of parameters to the MSstatsImport function.*

---

### **Description**

Check validity of parameters to the MSstatsImport function.

### **Usage**

```
.checkMSstatsParams(  
  input,  
  annotation,  
  feature_columns,  
  remove_shared_peptides,  
  remove_single_feature_proteins,  
  feature_cleaning  
)
```

### **Value**

none, throws an error if any of the assertions fail

---

|                             |                                      |
|-----------------------------|--------------------------------------|
| <code>.checkMultiRun</code> | <i>Check if fractionation exists</i> |
|-----------------------------|--------------------------------------|

---

**Description**

Check if fractionation exists

**Usage**

```
.checkMultiRun(input)
```

**Arguments**

|       |                             |
|-------|-----------------------------|
| input | output of MSstatsPreprocess |
|-------|-----------------------------|

**Value**

list of two elements: `has_fractions` (logical) indicates if fractions was detected in the input dataset, `is_risky` (logical) indicates if there was a problem with detecting fractionation.

---

|                                       |                                                                 |
|---------------------------------------|-----------------------------------------------------------------|
| <code>.checkOverlappedFeatures</code> | <i>Check if any features are measured in multiple fractions</i> |
|---------------------------------------|-----------------------------------------------------------------|

---

**Description**

Check if any features are measured in multiple fractions

**Usage**

```
.checkOverlappedFeatures(input)
```

**Arguments**

|       |                             |
|-------|-----------------------------|
| input | output of MSstatsPreprocess |
|-------|-----------------------------|

**Value**

data.table

---

|                              |                                       |
|------------------------------|---------------------------------------|
| <code>.cleanByFeature</code> | <i>Perform by-feature operations.</i> |
|------------------------------|---------------------------------------|

---

**Description**

Perform by-feature operations.

**Usage**

```
.cleanByFeature(  
  input,  
  feature_columns,  
  cleaning_control,  
  anomaly_metrics = c()  
)
```

**Arguments**

|                               |                                                                                                        |
|-------------------------------|--------------------------------------------------------------------------------------------------------|
| <code>input</code>            | data.table preprocessed by one of the <code>cleanRaw*</code> functions.                                |
| <code>feature_columns</code>  | character vector of names of columns that define features.                                             |
| <code>cleaning_control</code> | named list of two or three elements. See the documentation for <code>MSstatsImport</code> for details. |
| <code>anomaly_metrics</code>  | character vector of quality metric column names to be used as features in an anomaly detection model.  |

**Value**

data.table

---

|                             |                              |
|-----------------------------|------------------------------|
| <code>.cleanRawDIANN</code> | <i>Clean raw Diann files</i> |
|-----------------------------|------------------------------|

---

**Description**

Clean raw Diann files

**Usage**

```
.cleanRawDIANN(  
  msstats_object,  
  MBR = TRUE,  
  quantificationColumn = "FragmentQuantCorrected"  
)
```

**Arguments**

- `msstats_object` an object of class `MSstatsDIANNFiles`.
- `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`.

**Value**

`data.table`

---

|                                 |                                  |
|---------------------------------|----------------------------------|
| <code>.cleanRawDIAUmpire</code> | <i>Clean raw DIAUmpire files</i> |
|---------------------------------|----------------------------------|

---

**Description**

Clean raw DIAUmpire files

**Usage**

`.cleanRawDIAUmpire(msstats_object, use_frag, use_pept)`

**Arguments**

- `msstats_object` Object that inherits from `MSstatsInputFiles` class.
- `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.

**Value**

`data.table`

---

|                                |                                       |
|--------------------------------|---------------------------------------|
| <code>.cleanRawMaxQuant</code> | <i>Clean raw output from MaxQuant</i> |
|--------------------------------|---------------------------------------|

---

**Description**

Clean raw output from MaxQuant

**Usage**

```
.cleanRawMaxQuant(  
  msstats_object,  
  protein_id_col,  
  remove_by_site = FALSE,  
  channel_columns = "Reporterintensitycorrected"  
)
```

**Arguments**

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

**Value**

`data.table`

---

|                                    |                                     |
|------------------------------------|-------------------------------------|
| <code>.cleanRawMetamorpheus</code> | <i>Clean raw Metamorpheus files</i> |
|------------------------------------|-------------------------------------|

---

**Description**

Clean raw Metamorpheus files

**Usage**

```
.cleanRawMetamorpheus(msstats_object, MBR = TRUE, qvalue_cutoff = 0.05)
```

**Arguments**

`msstats_object` an object of class `MSstatsMetamorpheusFiles`.  
`MBR` If TRUE, the function will include peaks detected by MBR  
`qvalue_cutoff` The q-value cutoff for filtering peaks detected by MBR

**Value**

data.table

---

|                              |                                     |
|------------------------------|-------------------------------------|
| <code>.cleanRawOpenMS</code> | <i>Clean raw output from OpenMS</i> |
|------------------------------|-------------------------------------|

---

**Description**

Clean raw output from OpenMS

**Usage**

`.cleanRawOpenMS(msstats_object)`

**Arguments**

`msstats_object` an object of class `MSstatsSpectroMineFiles`.

**Value**

data.table

---

|                                 |                                  |
|---------------------------------|----------------------------------|
| <code>.cleanRawOpenSWATH</code> | <i>Clean raw OpenSWATH files</i> |
|---------------------------------|----------------------------------|

---

**Description**

Clean raw OpenSWATH files

**Usage**

`.cleanRawOpenSWATH(msstats_object)`

**Arguments**

`msstats_object` an object of class `MSstatsSpectroMineFiles`.

**Value**

data.table

---

|                    |                                           |
|--------------------|-------------------------------------------|
| <i>.cleanRawPD</i> | <i>Clean raw Proteome Discoverer data</i> |
|--------------------|-------------------------------------------|

---

**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

---

|                                 |                            |
|---------------------------------|----------------------------|
| <code>.cleanRawPDMSstats</code> | <i>Clean raw PD output</i> |
|---------------------------------|----------------------------|

---

**Description**

Clean raw PD output

**Usage**

```
.cleanRawPDMSstats(  
  msstats_object,  
  quantification_column,  
  protein_id_column,  
  sequence_column,  
  remove_shared,  
  run_column = "SpectrumFile"  
)
```

**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.

**Value**

data.table

---

|                             |                                                    |
|-----------------------------|----------------------------------------------------|
| <code>.cleanRawPDTMT</code> | <i>Clean raw TMT data from Proteome Discoverer</i> |
|-----------------------------|----------------------------------------------------|

---

**Description**

Clean raw TMT data from Proteome Discoverer

**Usage**

```
.cleanRawPDTMT(  
  msstats_object,  
  remove_shared = TRUE,  
  remove_protein_groups = TRUE,  
  protein_id_column = "ProteinAccessions",  
  intensity_columns_regexp = "Abundance",  
  run_column = "SpectrumFile"  
)
```

**Arguments**

**msstats\_object** an object of class `MSstatsSpectroMineFiles`.

**remove\_shared** `lgl`, if `TRUE`, shared peptides will be removed.

**remove\_protein\_groups**  
if `TRUE`, proteins with `numProteins > 1` will be removed.

**protein\_id\_column**  
chr, name of a column with protein IDs.

**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`

---

*.cleanRawPhilosopher*    *Clean raw Philosopher files*

---

**Description**

Clean raw Philosopher files

**Usage**

```
.cleanRawPhilosopher(  
  msstats_object,  
  protein_id_col,  
  peptide_id_col,  
  channels,  
  remove_shared_peptides  
)
```

**Arguments**

`msstats_object` object of class `MSstatsPhilosopherFiles`  
`protein_id_col` character name of a column that identifies proteins  
`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

**Value**

data.table

---

`.cleanRawProgenesis` *Clean raw Progenesis output*

---

**Description**

Clean raw Progenesis output

**Usage**

```
.cleanRawProgenesis(msstats_object, runs, fix_colnames = TRUE)
```

**Arguments**

`msstats_object` an object of class `MSstatsSpectroMineFiles`.  
`runs` chr, vector of Run labels.  
`fix_colnames` lgl, if TRUE, one of the rows will be used as colnames.

**Value**

data.table

---

|                               |                                    |
|-------------------------------|------------------------------------|
| <code>.cleanRawSkyline</code> | <i>Clean raw data from Skyline</i> |
|-------------------------------|------------------------------------|

---

### **Description**

Clean raw data from Skyline

### **Usage**

```
.cleanRawSkyline(msstats_object)
```

### **Arguments**

`msstats_object` an object of class `MSstatsSpectroMineFiles`.

### **Value**

`data.table`

---

|                                      |                                       |
|--------------------------------------|---------------------------------------|
| <code>.cleanRawSpectroMineTMT</code> | <i>Clean raw SpectroMine TMT data</i> |
|--------------------------------------|---------------------------------------|

---

### **Description**

Clean raw SpectroMine TMT data

### **Usage**

```
.cleanRawSpectroMineTMT(msstats_object)
```

### **Arguments**

`msstats_object` an object of class `MSstatsSpectroMineFiles`.

### **Value**

`data.table`

---

`.cleanRawSpectronaut`    *Clean raw Spectronaut output.*

---

### Description

Clean raw Spectronaut output.

### Usage

```
.cleanRawSpectronaut(
  msstats_object,
  intensity,
  calculateAnomalyScores,
  anomalyModelFeatures
)
```

### Arguments

`msstats_object`    an object of class `MSstatsSpectronautFiles`.  
`intensity`            chr, specifies which column will be used for Intensity.  
`calculateAnomalyScores`  
                       logical, whether to calculate anomaly scores  
`anomalyModelFeatures`  
                       character vector, specifies which columns will be used for anomaly detection  
                       model. Can be NULL if `calculateAnomalyScores=FALSE`.

### Value

`data.table`

---

`.countCommonFeatures`    *Get common values from two vectors of features*

---

### Description

Get common values from two vectors of features

### Usage

```
.countCommonFeatures(features_1, features_2)
```

### Arguments

`features_1`            vector of feature names  
`features_2`            vector of feature\_names

**Value**

character vector of common values of features\_1 and features\_2

---

|                          |                                     |
|--------------------------|-------------------------------------|
| <code>.fillValues</code> | <i>Set column to a single value</i> |
|--------------------------|-------------------------------------|

---

**Description**

Set column to a single value

**Usage**

```
.fillValues(input, fill_list)
```

**Arguments**

|                        |                                                                                                    |
|------------------------|----------------------------------------------------------------------------------------------------|
| <code>input</code>     | data.table preprocessed by one of the <code>cleanRaw*</code> functions.                            |
| <code>fill_list</code> | named list, names correspond to column names, elements to values that will be used in the columns. |

**Value**

data.table

---

|                               |                                    |
|-------------------------------|------------------------------------|
| <code>.filterByPattern</code> | <i>Handle filtering by pattern</i> |
|-------------------------------|------------------------------------|

---

**Description**

Handle filtering by pattern

**Usage**

```
.filterByPattern(input, col_name, patterns, filter, drop)
```

**Arguments**

|                       |                                                                            |
|-----------------------|----------------------------------------------------------------------------|
| <code>input</code>    | data.table preprocessed by one of the <code>.cleanRaw*</code> functions.   |
| <code>col_name</code> | chr, name of the column with peptide sequences.                            |
| <code>patterns</code> | chr, regular expression - matching peptides will be removed from the data. |
| <code>filter</code>   | lgl, if TRUE, peptides will be actually filtered.                          |
| <code>drop</code>     | lgl, if TRUE, the column will be dropped.                                  |

**Value**

data.table

---

|                             |                                                        |
|-----------------------------|--------------------------------------------------------|
| <code>.filterByScore</code> | <i>Filter PSMs / proteins by a given score column.</i> |
|-----------------------------|--------------------------------------------------------|

---

**Description**

Filter PSMs / proteins by a given score column.

**Usage**

```
.filterByScore(
  input,
  score_column,
  score_threshold,
  direction,
  behavior,
  handle_na = "keep",
  fill_value = NA,
  filter = TRUE,
  drop = TRUE
)
```

**Arguments**

|                              |                                                                                                                                         |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| <code>input</code>           | data.table preprocessed by one of the <code>.cleanRaw*</code> functions.                                                                |
| <code>score_column</code>    | chr, name of the column that contains scores.                                                                                           |
| <code>score_threshold</code> | num, values below or above this threshold will be removed from the data.                                                                |
| <code>direction</code>       | chr, if "greater" only values above the threshold will be retained, if "smaller" - below the threshold.                                 |
| <code>behavior</code>        | chr, if "remove", values below/above the threshold will be removed, if "replace", they will be set to <code>fill_value</code> .         |
| <code>fill_value</code>      | if <code>behavior = "replace"</code> , values below/above the threshold will be replaced with <code>fill_value</code> . Defaults to NA. |
| <code>filter</code>          | If TRUE, filtering will be performed.                                                                                                   |
| <code>drop</code>            | if TRUE, <code>score_column</code> will be removed.                                                                                     |

**Value**

data.table

---

|                     |                                      |
|---------------------|--------------------------------------|
| <i>.filterExact</i> | <i>Filter out specified symbols.</i> |
|---------------------|--------------------------------------|

---

**Description**

Filter out specified symbols.

**Usage**

```
.filterExact(
  input,
  col_name,
  filter_symbols,
  behavior,
  fill_value,
  filter,
  drop
)
```

**Arguments**

|                             |                                                                                                                                        |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| <code>input</code>          | data.table preprocessed by one of the <code>.cleanRaw*</code> functions.                                                               |
| <code>col_name</code>       | chr, name of the column that will be the base for filtering                                                                            |
| <code>filter_symbols</code> | character vector of symbols that will be removed                                                                                       |
| <code>behavior</code>       | chr, if "remove", values below/above the threshold will be removed, if "replace", they will be set to <code>fill_value</code> .        |
| <code>fill_value</code>     | if <code>behavior</code> = "replace", values below/above the threshold will be replaced with <code>fill_value</code> . Defaults to NA. |
| <code>filter</code>         | lgl, if TRUE, decoy proteins will be removed from the data.                                                                            |
| <code>drop</code>           | lgl, if TRUE, column that contains decoy proteins will be dropped.                                                                     |

**Value**

data.table

---

|                               |                                                                                      |
|-------------------------------|--------------------------------------------------------------------------------------|
| <i>.filterFewMeasurements</i> | <i>Remove features with a small number of (non-missing) measurements across runs</i> |
|-------------------------------|--------------------------------------------------------------------------------------|

---

**Description**

Remove features with a small number of (non-missing) measurements across runs

**Usage**

```
.filterFewMeasurements(
  input,
  min_intensity,
  remove_few,
  feature_columns = NULL
)
```

**Arguments**

|                              |                                                                                                                                                   |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>input</code>           | <code>data.table</code> pre-processed by one of the <code>.cleanRaw*</code> functions.                                                            |
| <code>min_intensity</code>   | minimum intensity that will be considered non-missing.                                                                                            |
| <code>remove_few</code>      | logical, if TRUE, features that have less than three measurements will be removed. If FALSE, only features with all missing runs will be removed. |
| <code>feature_columns</code> | chr, vector of names of columns that define features.                                                                                             |

**Value**

`data.table`

---

|                                 |                                                                        |
|---------------------------------|------------------------------------------------------------------------|
| <code>.filterManyColumns</code> | <i>Filter rows that contain specified symbols in multiple columns.</i> |
|---------------------------------|------------------------------------------------------------------------|

---

**Description**

Filter rows that contain specified symbols in multiple columns.

**Usage**

```
.filterManyColumns(input, filter_columns, filter_symbols)
```

**Arguments**

|                             |                                                                                                          |
|-----------------------------|----------------------------------------------------------------------------------------------------------|
| <code>input</code>          | <code>data.table</code> preprocessed by one of the <code>cleanRaw*</code> functions.                     |
| <code>filter_columns</code> | chr, names of columns in which elements will be matched and removed.                                     |
| <code>filter_symbols</code> | chr, vector of strings. Rows with corresponding elements in <code>filter_columns</code> will be removed. |

**Value**

`data.table`

---

|                                |                                   |
|--------------------------------|-----------------------------------|
| <code>.filterOverlapped</code> | <i>Remove overlapped features</i> |
|--------------------------------|-----------------------------------|

---

**Description**

Remove overlapped features

**Usage**

```
.filterOverlapped(input, summary_function, overlapped_features)
```

**Arguments**

|                     |                                                                                                            |
|---------------------|------------------------------------------------------------------------------------------------------------|
| input               | data.table preprocessed by one of the <code>.cleanRaw*</code> functions and merged with annotation.        |
| summary_function    | summary function (mean, sum, max) that will be used to pick one feature from multiple overlapping features |
| overlapped_features | features that overlap.                                                                                     |

**Value**

data.table

---

|                             |                                                                |
|-----------------------------|----------------------------------------------------------------|
| <code>.findAvailable</code> | <i>Select an available options from a set of possibilities</i> |
|-----------------------------|----------------------------------------------------------------|

---

**Description**

Select an available options from a set of possibilities

**Usage**

```
.findAvailable(possibilities, option_set, fall_back = NULL)
```

**Arguments**

|               |                                                                                                       |
|---------------|-------------------------------------------------------------------------------------------------------|
| possibilities | possible legal values of a variable                                                                   |
| option_set    | set of values that includes one of the possibilities                                                  |
| fall_back     | if there is none of the possibilities in option_set, or there are multiple hits, default to fall_back |

**Value**

same as option\_set, usually character

---

|                               |                                                                               |
|-------------------------------|-------------------------------------------------------------------------------|
| <code>.fixBasicColumns</code> | <i>Remove underscores from sequences and change intensity type to numeric</i> |
|-------------------------------|-------------------------------------------------------------------------------|

---

**Description**

Remove underscores from sequences and change intensity type to numeric

**Usage**

```
.fixBasicColumns(input)
```

**Arguments**

|       |            |
|-------|------------|
| input | data.table |
|-------|------------|

**Value**

data.table

---

|                              |                                           |
|------------------------------|-------------------------------------------|
| <code>.fixColumnTypes</code> | <i>Change classes of multiple columns</i> |
|------------------------------|-------------------------------------------|

---

**Description**

Change classes of multiple columns

**Usage**

```
.fixColumnTypes(
  input,
  numeric_columns = NULL,
  character_columns = NULL,
  factor_columns = NULL
)
```

**Arguments**

|                   |                                                                      |
|-------------------|----------------------------------------------------------------------|
| input             | data.table preprocessed by one of the cleanRaw* functions.           |
| numeric_columns   | chr, vector of names of columns that will be converted to numeric.   |
| character_columns | chr, vector of names of columns that will be converted to character. |
| factor_columns    | chr, vector of names of columns that will be converted to factor.    |

**Value**

data.table

---

|                                |                                         |
|--------------------------------|-----------------------------------------|
| <code>.fixMissingValues</code> | <i>Change labels for missing values</i> |
|--------------------------------|-----------------------------------------|

---

**Description**

Change labels for missing values

**Usage**

```
.fixMissingValues(input, fix_missing = NULL)
```

**Arguments**

|                          |                                                                                                                                                                                                                                                 |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>input</code>       | output of <code>MSstatsPreprocess</code>                                                                                                                                                                                                        |
| <code>fix_missing</code> | missing values can be labeled by NA, 0 or both. If NULL, data were processed by Skyline, so missing values will be denoted by both NA and 0. If "na_to_zero", NA values will be replaced by 0. If "zero_to_na", 0 values will be replaced by NA |

**Value**

data.table

---

|                                 |                                                    |
|---------------------------------|----------------------------------------------------|
| <code>.getChannelColumns</code> | <i>Get intensity columns from wide-format data</i> |
|---------------------------------|----------------------------------------------------|

---

**Description**

Get intensity columns from wide-format data

**Usage**

```
.getChannelColumns(col_names, ...)
```

**Arguments**

|                        |                                                                                          |
|------------------------|------------------------------------------------------------------------------------------|
| <code>col_names</code> | names of columns, where some of the columns store intensity value for different channels |
| <code>...</code>       | varying number of strings that define channel columns.                                   |

**Value**

character vector of column names that correspond to channel intensities

---

|                                  |                                                                                                        |
|----------------------------------|--------------------------------------------------------------------------------------------------------|
| <code>.getCorrectFraction</code> | <i>Get a name of fraction with the largest number of measurements or the largest average intensity</i> |
|----------------------------------|--------------------------------------------------------------------------------------------------------|

---

**Description**

Get a name of fraction with the largest number of measurements or the largest average intensity

**Usage**

```
.getCorrectFraction(input)
```

**Arguments**

|       |                             |
|-------|-----------------------------|
| input | output of MSstatsPreprocess |
|-------|-----------------------------|

**Value**

character - label of the fraction that has most measurements or highest mean intensity for a given feature

---

|                            |                                                                                 |
|----------------------------|---------------------------------------------------------------------------------|
| <code>.getDataTable</code> | <i>Read file from a provided path or convert given data.frame to data.table</i> |
|----------------------------|---------------------------------------------------------------------------------|

---

**Description**

Read file from a provided path or convert given data.frame to data.table

**Usage**

```
.getDataTable(input, ...)
```

**Arguments**

|       |                                                      |
|-------|------------------------------------------------------|
| input | report from a signal processing tool or a path to it |
| ...   | additional parameters for data.table::fread          |

**Value**

data.table

---

|                             |                                                                              |
|-----------------------------|------------------------------------------------------------------------------|
| <code>.getFullDesign</code> | <i>Create a data.frame of each combination of values for given variables</i> |
|-----------------------------|------------------------------------------------------------------------------|

---

**Description**

Create a data.frame of each combination of values for given variables

**Usage**

```
.getFullDesign(input, group_col, feature_col, measurement_col, is_tmt)
```

**Arguments**

|                                |                                                                                                                                                                     |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>input</code>             | output of MSstatsPreprocess                                                                                                                                         |
| <code>group_col</code>         | name of column in input. Combination of values of <code>feature_col</code> and <code>measurement_col</code> will be created within each unique value of this column |
| <code>is_tmt</code>            | if TRUE, data will be treated as coming from TMT experiment.                                                                                                        |
| <code>'feature_col'</code>     | name of the column that labels features                                                                                                                             |
| <code>'measurement_col'</code> | name of a column with measurement labels - Runs in label-free case, Channels in TMT case.                                                                           |

**Value**

data.table

---

|                                        |                                  |
|----------------------------------------|----------------------------------|
| <code>.getMissingRunsPerFeature</code> | <i>Get names of missing runs</i> |
|----------------------------------------|----------------------------------|

---

**Description**

Get names of missing runs

**Usage**

```
.getMissingRunsPerFeature(input)
```

**Arguments**

|                    |                             |
|--------------------|-----------------------------|
| <code>input</code> | output of MSstatsPreprocess |
|--------------------|-----------------------------|

**Value**

data.table

---

`.getOverlappingFeatures`*Get features that are overlapped among multiple runs*

---

**Description**

Get features that are overlapped among multiple runs

**Usage**

```
.getOverlappingFeatures(input)
```

**Arguments**

|       |                                                                                                     |
|-------|-----------------------------------------------------------------------------------------------------|
| input | data.table preprocessed by one of the <code>.cleanRaw*</code> functions and merged with annotation. |
|-------|-----------------------------------------------------------------------------------------------------|

**Value**

data.table

---

`.handleFiltering`*Handle PSM/proteins scores*

---

**Description**

Handle PSM/proteins scores

**Usage**

```
.handleFiltering(input, score_filtering, exact_filtering, pattern_filtering)
```

**Arguments**

|                   |                                                                          |
|-------------------|--------------------------------------------------------------------------|
| input             | data.table preprocessed by one of the <code>.cleanRaw*</code> functions. |
| score_filtering   | list of by-score filtering controls.                                     |
| exact_filtering   | list of exact filtering controls.                                        |
| pattern_filtering | list of by-pattern filtering controls.                                   |

**Value**

data.table

---

|                               |                                                                     |
|-------------------------------|---------------------------------------------------------------------|
| <code>.handleFractions</code> | <i>Check if there are overlapping features and remove if needed</i> |
|-------------------------------|---------------------------------------------------------------------|

---

### **Description**

Check if there are overlapping features and remove if needed

### **Usage**

```
.handleFractions(input)
```

### **Arguments**

|                    |                                                                                                     |
|--------------------|-----------------------------------------------------------------------------------------------------|
| <code>input</code> | data.table preprocessed by one of the <code>.cleanRaw*</code> functions and merged with annotation. |
|--------------------|-----------------------------------------------------------------------------------------------------|

### **Value**

data.table

---

|                                 |                                    |
|---------------------------------|------------------------------------|
| <code>.handleFractionsLF</code> | <i>Handle overlapping features</i> |
|---------------------------------|------------------------------------|

---

### **Description**

Handle overlapping features

### **Usage**

```
.handleFractionsLF(input)
```

### **Arguments**

|                    |                                          |
|--------------------|------------------------------------------|
| <code>input</code> | output of <code>MSstatsPreprocess</code> |
|--------------------|------------------------------------------|

### **Value**

data.table

---

|                                  |                                                                                               |
|----------------------------------|-----------------------------------------------------------------------------------------------|
| <code>.handleFractionsTMT</code> | <i>Remove peptide ions overlapped among multiple fractions of the same biological mixture</i> |
|----------------------------------|-----------------------------------------------------------------------------------------------|

---

**Description**

Remove peptide ions overlapped among multiple fractions of the same biological mixture

**Usage**

```
.handleFractionsTMT(input)
```

**Arguments**

|       |                                                                                                     |
|-------|-----------------------------------------------------------------------------------------------------|
| input | data.table preprocessed by one of the <code>.cleanRaw*</code> functions and merged with annotation. |
|-------|-----------------------------------------------------------------------------------------------------|

**Value**

data.table

---

|                                   |                              |
|-----------------------------------|------------------------------|
| <code>.handleIsotopicPeaks</code> | <i>Handle isotopic peaks</i> |
|-----------------------------------|------------------------------|

---

**Description**

Handle isotopic peaks

**Usage**

```
.handleIsotopicPeaks(input, aggregate = FALSE)
```

**Arguments**

|           |                                                                         |
|-----------|-------------------------------------------------------------------------|
| input     | data.table preprocessed by one of the <code>cleanRaw*</code> functions. |
| aggregate | if TRUE, isotopic peaks will be summed.                                 |

**Value**

data.table

---

`.handleSharedPeptides` *Handle shared peptides.*

---

**Description**

Handle shared peptides.

**Usage**

```
.handleSharedPeptides(  
  input,  
  remove_shared = TRUE,  
  protein_column = "ProteinName",  
  peptide_column = "PeptideSequence"  
)
```

**Arguments**

`input` data.table pre-processed by one of the `.cleanRaw*` functions.  
`remove_shared` `lgl`, if `TRUE`, shared peptides will be removed  
`protein_column` `chr`, name of the column with names of proteins.  
`peptide_column` `chr`, name of the column with peptide sequences.

**Value**

data.table

---

`.handleSingleFeaturePerProtein`  
*Remove proteins only identified by a single feature*

---

**Description**

Remove proteins only identified by a single feature

**Usage**

```
.handleSingleFeaturePerProtein(input, remove_single_feature)
```

**Arguments**

`input` data.table pre-processed by one of the `.cleanRaw*` functions.  
`remove_single_feature`  
`lgl`, if `TRUE`, proteins with a single feature will be removed.

**Value**

data.table

---

*.logConverterOptions*    *Log information about converter options*

---

**Description**

Log information about converter options

**Usage**

```
.logConverterOptions(
  feature_columns,
  remove_shared_peptides,
  remove_single_feature_proteins,
  feature_cleaning,
  is_tmt = FALSE
)
```

**Arguments**

**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.

**is\_tmt**  
If TRUE, the dataset comes from a TMT experiment

**Value**

TRUE invisibly if message was logged

---

|             |                                                                |
|-------------|----------------------------------------------------------------|
| .logSuccess | <i>Make a message about successful data cleaning/importing</i> |
|-------------|----------------------------------------------------------------|

---

### Description

Make a message about successful data cleaning/importing

### Usage

```
.logSuccess(tool, event)
```

### Arguments

|      |                                  |
|------|----------------------------------|
| tool | name of a signal processing tool |
|------|----------------------------------|

### Value

TRUE invisibly if logging was successful

---

|                     |                                                    |
|---------------------|----------------------------------------------------|
| .makeBalancedDesign | <i>Fill missing rows to create balanced design</i> |
|---------------------|----------------------------------------------------|

---

### Description

Fill missing rows to create balanced design

### Usage

```
.makeBalancedDesign(input, fill_missing, anomaly_metrics = c())
```

### Arguments

|                 |                                                                                                                        |
|-----------------|------------------------------------------------------------------------------------------------------------------------|
| input           | output of MSstatsPreprocess                                                                                            |
| fill_missing    | if TRUE, missing Intensities values will be added to data                                                              |
| anomaly_metrics | character vector of quality metric column names to be used as features in an anomaly detection model. and marked as NA |

### Value

data.table

---

```
.makeExactFilterMessage
```

*Make a message about filtering based on fixed values*

---

### Description

Make a message about filtering based on fixed values

### Usage

```
.makeExactFilterMessage(col_name, filter_symbols, behavior, fill_value)
```

### Arguments

|                             |                                                                                                                                         |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| <code>col_name</code>       | chr, name of the column that will be the base for filtering                                                                             |
| <code>filter_symbols</code> | character vector of symbols that will be removed                                                                                        |
| <code>behavior</code>       | chr, if "remove", values below/above the threshold will be removed, if "replace", they will be set to <code>fill_value</code> .         |
| <code>fill_value</code>     | if <code>behavior = "replace"</code> , values below/above the threshold will be replaced with <code>fill_value</code> . Defaults to NA. |

### Value

character - message

---

```
.makeScoreFilterMessage
```

*Make a message about filtering based on a score*

---

### Description

Make a message about filtering based on a score

### Usage

```
.makeScoreFilterMessage(
  score_column,
  score_threshold,
  direction,
  behavior,
  fill_value
)
```

**Arguments**

|                              |                                                                                                                                         |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| <code>score_column</code>    | chr, name of the column that contains scores.                                                                                           |
| <code>score_threshold</code> | num, values below or above this threshold will be removed from the data.                                                                |
| <code>direction</code>       | chr, if "greater" only values above the threshold will be retained, if "smaller" - below the threshold.                                 |
| <code>behavior</code>        | chr, if "remove", values below/above the threshold will be removed, if "replace", they will be set to <code>fill_value</code> .         |
| <code>fill_value</code>      | if <code>behavior = "replace"</code> , values below/above the threshold will be replaced with <code>fill_value</code> . Defaults to NA. |

**Value**

character - message

---

|                               |                                           |
|-------------------------------|-------------------------------------------|
| <code>.mergeAnnotation</code> | <i>Merge annotation with feature data</i> |
|-------------------------------|-------------------------------------------|

---

**Description**

Merge annotation with feature data

**Usage**

```
.mergeAnnotation(input, annotation)
```

**Arguments**

|                         |                                                              |
|-------------------------|--------------------------------------------------------------|
| <code>annotation</code> | data.table with annotation                                   |
| <code>data.table</code> | preprocessed by one of the <code>.cleanRaw</code> functions. |

**Value**

data.table

---

|                             |                                                      |
|-----------------------------|------------------------------------------------------|
| <code>.MSstatsFormat</code> | <i>Output format for further analysis by MSstats</i> |
|-----------------------------|------------------------------------------------------|

---

**Description**

Output format for further analysis by MSstats

**Usage**

```
.MSstatsFormat(input, anomaly_metrics = c())
```

**Arguments**

|                              |                                                                                                      |
|------------------------------|------------------------------------------------------------------------------------------------------|
| <code>input</code>           | <code>data.table</code>                                                                              |
| <code>anomaly_metrics</code> | character vector of quality metric column names to be used as features in an anomaly detection model |

**Value**

object of class `MSstatsValidated` that inherits from `data.frame`

---

|                            |                                                  |
|----------------------------|--------------------------------------------------|
| <code>.nullAppender</code> | <i>log4r appender used not to write messages</i> |
|----------------------------|--------------------------------------------------|

---

**Description**

A convenience function written to save time on checking if messages should be printed or logs should be written to a file.

**Usage**

```
.nullAppender(level, ...)
```

**Arguments**

|                    |                    |
|--------------------|--------------------|
| <code>level</code> | log level          |
| <code>...</code>   | messages - ignored |

**Value**

NULL invisibly

---

|         |                                                          |
|---------|----------------------------------------------------------|
| .onLoad | <i>Set default logging object when package is loaded</i> |
|---------|----------------------------------------------------------|

---

### Description

Set default logging object when package is loaded

### Usage

```
.onLoad(...)
```

### Arguments

... ignored

### Value

none, sets options called MSstatsLog and MSstatsMsg

---

|                            |                                                                                       |
|----------------------------|---------------------------------------------------------------------------------------|
| .removeOverlappingFeatures | <i>Replace intensities of overlapped fractions with NA, keeping only one fraction</i> |
|----------------------------|---------------------------------------------------------------------------------------|

---

### Description

Replace intensities of overlapped fractions with NA, keeping only one fraction

### Usage

```
.removeOverlappingFeatures(input)
```

### Arguments

input output of MSstatsPreprocess

### Value

data.table

---

```
.removeSharedPeptides
```

*Remove peptides assigned to more than one protein.*

---

**Description**

Remove peptides assigned to more than one protein.

**Usage**

```
.removeSharedPeptides(input, protein_column, peptide_column)
```

**Arguments**

`input` data.table pre-processed by one of the `.cleanRaw*` functions.  
`protein_column` chr, name of the column with names of proteins.  
`peptide_column` chr, name of the column with peptide sequences.

**Value**

data.table

---

```
.selectMSstatsColumns
```

*Select columns for MSstats format*

---

**Description**

Select columns for MSstats format

**Usage**

```
.selectMSstatsColumns(input, anomaly_metrics)
```

**Arguments**

`input` data.table

**Value**

data.table

---

`.sharedParametersAmongConverters`
*A dummy function to store shared documentation items for converters.*


---

## Description

A dummy function to store shared documentation items for converters.

## Usage

```
.sharedParametersAmongConverters()
```

## Arguments

|                                         |                                                                                                                                                                                                                   |
|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>removeFewMeasurements</code>      | TRUE (default) will remove the features that have 1 or 2 measurements across runs.                                                                                                                                |
| <code>useUniquePeptide</code>           | TRUE (default) removes peptides that are assigned for more than one proteins. We assume to use unique peptide for each protein.                                                                                   |
| <code>summaryforMultipleRows</code>     | max(default) or sum - when there are multiple measurements for certain feature and certain run, use highest or sum of multiple intensities.                                                                       |
| <code>removeProtein_with1Feature</code> | TRUE will remove the proteins which have only 1 feature, which is the combination of peptide, precursor charge, fragment and charge. FALSE is default.                                                            |
| <code>removeProtein_with1Peptide</code> | TRUE will remove the proteins which have only 1 peptide and charge. FALSE is default.                                                                                                                             |
| <code>removeOxidationMpeptides</code>   | TRUE will remove the peptides including 'oxidation (M)' in modification. FALSE is default.                                                                                                                        |
| <code>removeMpeptides</code>            | TRUE will remove the peptides including 'M' sequence. FALSE is default.                                                                                                                                           |
| <code>use_log_file</code>               | logical. If TRUE, information about data processing will be saved to a file.                                                                                                                                      |
| <code>append</code>                     | logical. If TRUE, information about data processing will be added to an existing log file.                                                                                                                        |
| <code>verbose</code>                    | logical. If TRUE, information about data processing will be printed to the console.                                                                                                                               |
| <code>log_file_path</code>              | 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 <code>append = TRUE</code> , has to be a valid path to a file. |
| <code>...</code>                        | additional parameters to <code>data.table::fread</code> .                                                                                                                                                         |

---

|                                   |                                                                                |
|-----------------------------------|--------------------------------------------------------------------------------|
| <code>.standardizeColnames</code> | <i>Change column names to match read.table/read.csv/read.delim conventions</i> |
|-----------------------------------|--------------------------------------------------------------------------------|

---

**Description**

Change column names to match read.table/read.csv/read.delim conventions

**Usage**

```
.standardizeColnames(col_names)
```

**Arguments**

|           |                             |
|-----------|-----------------------------|
| col_names | chr, vector of column names |
|-----------|-----------------------------|

**Value**

character vector

---

|                                             |                                                                    |
|---------------------------------------------|--------------------------------------------------------------------|
| <code>.summarizeMultipleMeasurements</code> | <i>Summarize multiple measurements per feature in a single run</i> |
|---------------------------------------------|--------------------------------------------------------------------|

---

**Description**

Summarize multiple measurements per feature in a single run

**Usage**

```
.summarizeMultipleMeasurements(
  input,
  aggregator,
  feature_columns,
  anomaly_metrics = c()
)
```

**Arguments**

|                 |                                                                                                       |
|-----------------|-------------------------------------------------------------------------------------------------------|
| input           | data.table pre-processed by one of the .cleanRaw* functions.                                          |
| aggregator      | function that will be used to aggregate duplicated values.                                            |
| feature_columns | chr, vector of names of columns that define features.                                                 |
| anomaly_metrics | character vector of quality metric column names to be used as features in an anomaly detection model. |

**Value**

`data.table`

---

`.summarizeMultiplePSMs`

*Pick one PSM from a data.table of several PSMs.*

---

**Description**

Pick one PSM from a `data.table` of several PSMs.

**Usage**

`.summarizeMultiplePSMs(input, summary_function)`

**Arguments**

`input`                      `data.table` preprocessed by one of the `.cleanRaw*` functions.  
`summary_function`                      function that will be used to aggregate intensities if needed.

**Value**

character - label of a chosen PSM

---

`.validateMSstatsConverterParameters`

*Generic parameter validation for all MSstats converters using configuration object*

---

**Description**

Generic parameter validation for all MSstats converters using configuration object

**Usage**

`.validateMSstatsConverterParameters(config)`

**Arguments**

`config`                      A list containing all converter parameters. See details for required structure.

**Details**

The config list should contain the input and optionally other parameters:

- input: input data (required)
- annotation: annotation data (optional)
- intensity: intensity type (optional)
- filter\_with\_Qvalue: Q-value filter setting (default: FALSE)
- qvalue\_cutoff: Q-value cutoff (default: 0.01)
- useUniquePeptide: unique peptide setting (default: TRUE)
- removeFewMeasurements: remove few measurements setting (default: TRUE)
- removeProtein\_with1Feature: remove single feature proteins setting (default: FALSE)
- summaryforMultipleRows: aggregation function (default: max)
- calculateAnomalyScores: anomaly detection setting (default: FALSE)
- anomalyModelFeatures: anomaly model features (default: c())
- anomalyModelFeatureTemporal: temporal features (default: c())
- removeMissingFeatures: missing feature threshold (default: 0.5)
- anomalyModelFeatureCount: feature count for anomaly model (default: 100)
- runOrder: run order data (default: NULL)
- n\_trees: number of trees (default: 100)
- max\_depth: max tree depth (default: "auto")
- numberOfCores: number of cores (default: 1)
- use\_log\_file: logging setting (default: TRUE)
- append: append setting (default: FALSE)
- verbose: verbose setting (default: TRUE)
- log\_file\_path: log file path (default: NULL)
- excludedFromQuantificationFilter: filter setting (default: NULL)

**Value**

NULL (throws error if validation fails)

---

```
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

---

|                 |                                                                                                                                                                                                  |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CheckDataHealth | <i>Takes as input the output of the SpectronauttoMSstatsFormat function and calculates various quality metrics to assess the health of the data. Requires Anomaly Detection model to be fit.</i> |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

### Description

Takes as input the output of the SpectronauttoMSstatsFormat function and calculates various quality metrics to assess the health of the data. Requires Anomaly Detection model to be fit.

### Usage

```
CheckDataHealth(input)
```

### Arguments

|       |                                                            |
|-------|------------------------------------------------------------|
| input | MSstats input which is the output of Spectronaut converter |
|-------|------------------------------------------------------------|

### Value

list of two data.tables

---

|                      |                           |
|----------------------|---------------------------|
| DIANNtoMSstatsFormat | <i>Import Diann files</i> |
|----------------------|---------------------------|

---

### 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 fragment-level data. Output fragment data with <code>–export-quant</code> flag in DIA-NN 2.0                                                                                                                                                                       |
| annotation                 | name of 'annotation.txt' data which includes Condition, BioReplicate, Run.                                                                                                                                                                                                                                                 |
| 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. Run should be the same as filename. 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                                                                                                                                                                                                                                                                           |
| 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 <code>append = TRUE</code> , 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 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. <code>Fr0Quantity</code> .                        |
| ...                        | additional parameters to <code>data.table::fread</code> .                                                                                                                                                                                                                                                                  |

**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)

# For DIANN 2.0, set quantificationColumn = 'auto'
input_file_path_2_0 = system.file("tinytest/raw_data/DIANN/diann_2.0.parquet",
                                   package="MSstatsConvert")
annotation_file_path_2_0 = system.file("tinytest/raw_data/DIANN/annotation_diann_2.0.csv",
                                       package = "MSstatsConvert")
input_2_0 = arrow::read_parquet(input_file_path_2_0)
annot_2_0 = data.table::fread(annotation_file_path_2_0)
output_2_0 = DIANNtoMSstatsFormat(input_2_0, annotation = annot_2_0, MBR = FALSE,
                                  use_log_file = FALSE, quantificationColumn = 'auto')

head(output_2_0)
```

---

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

|                                         |                                                                                                                                                                                                                   |
|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>raw.frag</code>                   | name of FragSummary_date.xls data, which includes feature-level data.                                                                                                                                             |
| <code>raw.pep</code>                    | name of PeptideSummary_date.xls data, which includes selected fragments information.                                                                                                                              |
| <code>raw.pro</code>                    | name of ProteinSummary_date.xls data, which includes selected peptides information.                                                                                                                               |
| <code>annotation</code>                 | name of annotation data which includes Condition, BioReplicate, Run information.                                                                                                                                  |
| <code>useSelectedFrag</code>            | TRUE will use the selected fragment for each peptide. 'Selected_fragments' column is required.                                                                                                                    |
| <code>useSelectedPep</code>             | TRUE will use the selected peptide for each protein. 'Selected_peptides' column is required.                                                                                                                      |
| <code>removeFewMeasurements</code>      | TRUE (default) will remove the features that have 1 or 2 measurements across runs.                                                                                                                                |
| <code>removeProtein_with1Feature</code> | TRUE will remove the proteins which have only 1 feature, which is the combination of peptide, precursor charge, fragment and charge. FALSE is default.                                                            |
| <code>summaryforMultipleRows</code>     | max(default) or sum - when there are multiple measurements for certain feature and certain run, use highest or sum of multiple intensities.                                                                       |
| <code>use_log_file</code>               | logical. If TRUE, information about data processing will be saved to a file.                                                                                                                                      |
| <code>append</code>                     | logical. If TRUE, information about data processing will be added to an existing log file.                                                                                                                        |
| <code>verbose</code>                    | logical. If TRUE, information about data processing will be printed to the console.                                                                                                                               |
| <code>log_file_path</code>              | 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 <code>append = TRUE</code> , has to be a valid path to a file. |
| <code>...</code>                        | additional parameters to <code>data.table::fread</code> .                                                                                                                                                         |

## 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**

|                                         |                                                                                                                                                                                                                   |
|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>input</code>                      | name of FragPipe msstats.csv export. ProteinName, PeptideSequence, PrecursorCharge, FragmentIon, ProductCharge, IsotopeLabelType, Condition, BioReplicate, Run, Intensity are required.                           |
| <code>useUniquePeptide</code>           | TRUE (default) removes peptides that are assigned for more than one proteins. We assume to use unique peptide for each protein.                                                                                   |
| <code>removeFewMeasurements</code>      | TRUE (default) will remove the features that have 1 or 2 measurements across runs.                                                                                                                                |
| <code>removeProtein_with1Feature</code> | TRUE will remove the proteins which have only 1 feature, which is the combination of peptide, precursor charge, fragment and charge. FALSE is default.                                                            |
| <code>summaryforMultipleRows</code>     | max(default) or sum - when there are multiple measurements for certain feature and certain run, use highest or sum of multiple intensities.                                                                       |
| <code>use_log_file</code>               | logical. If TRUE, information about data processing will be saved to a file.                                                                                                                                      |
| <code>append</code>                     | logical. If TRUE, information about data processing will be added to an existing log file.                                                                                                                        |
| <code>verbose</code>                    | logical. If TRUE, information about data processing will be printed to the console.                                                                                                                               |
| <code>log_file_path</code>              | 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 <code>append = TRUE</code> , has to be a valid path to a file. |
| <code>...</code>                        | additional parameters to <code>data.table::fread</code> .                                                                                                                                                         |

**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)
```

---

|             |                                                              |
|-------------|--------------------------------------------------------------|
| getDataType | <i>Get type of dataset from an MSstatsInputFiles object.</i> |
|-------------|--------------------------------------------------------------|

---

### Description

Get type of dataset from an MSstatsInputFiles object.

### Usage

```
getDataType(msstats_object)

## S4 method for signature 'MSstatsInputFiles'
getDataType(msstats_object)
```

### Arguments

msstats\_object object that inherits from MSstatsInputFiles class.

### Value

character - label of a data type. Currently, "MSstats" or "MSstatsTMT"  
character "MSstats" or "MSstatsTMT".

### 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)
getDataType(imported) # "MSstats"
```

---

|              |                                                                              |
|--------------|------------------------------------------------------------------------------|
| getInputFile | <i>Get one of files contained in an instance of MSstatsInputFiles class.</i> |
|--------------|------------------------------------------------------------------------------|

---

### 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 | <i>Import MaxQuant files</i> |
|---------------------|------------------------------|

---

**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 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. |
| ...                        | 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,
  MBR = TRUE,
  qvalue_cutoff = 0.05,
  useUniquePeptide = TRUE,
  removeFewMeasurements = TRUE,
  removeProtein_with1Feature = FALSE,
  summaryforMultipleRows = max,
  use_log_file = TRUE,
  append = FALSE,
  verbose = TRUE,
  log_file_path = NULL,
  ...
)
```

**Arguments**

|                                         |                                                                                                                                                                                                     |
|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>input</code>                      | name of Metamorpheus output file, which is tabular format. Use the AllQuantifiedPeaks.tsv file from the Metamorpheus output.                                                                        |
| <code>annotation</code>                 | name of 'annotation.txt' data which includes Condition, BioReplicate.                                                                                                                               |
| <code>MBR</code>                        | If TRUE, the function will include peaks detected by MBR                                                                                                                                            |
| <code>qvalue_cutoff</code>              | The q-value cutoff for filtering peaks detected by MBR                                                                                                                                              |
| <code>useUniquePeptide</code>           | TRUE (default) removes peptides that are assigned for more than one proteins. We assume to use unique peptide for each protein.                                                                     |
| <code>removeFewMeasurements</code>      | TRUE (default) will remove the features that have 1 or 2 measurements across runs.                                                                                                                  |
| <code>removeProtein_with1Feature</code> | TRUE will remove the proteins which have only 1 feature, which is the combination of peptide, precursor charge, fragment and charge. FALSE is default.                                              |
| <code>summaryforMultipleRows</code>     | max(default) or sum - when there are multiple measurements for certain feature and certain run, use highest or sum of multiple intensities.                                                         |
| <code>use_log_file</code>               | logical. If TRUE, information about data processing will be saved to a file.                                                                                                                        |
| <code>append</code>                     | logical. If TRUE, information about data processing will be added to an existing log file.                                                                                                          |
| <code>verbose</code>                    | logical. If TRUE, information about data processing will be printed to the console.                                                                                                                 |
| <code>log_file_path</code>              | 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. |
| <code>...</code>                        | additional parameters to <code>data.table::fread</code> .                                                                                                                                           |

**Value**

data.frame in the MSstats required format.

**Author(s)**

Anthony Wu

**Examples**

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

---

**MSstatsAnomalyScores**    *Run Anomaly Model*

---

**Description**

Run Anomaly Model

**Usage**

```
MSstatsAnomalyScores(  
  input,  
  quality_metrics,  
  temporal_direction,  
  missing_run_count,  
  n_feat,  
  run_order,  
  n_trees,  
  max_depth,  
  cores  
)
```

**Arguments**

|                                 |                                                                                                                                 |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| <code>input</code>              | data.table preprocessed by the MSstatsBalancedDesign function                                                                   |
| <code>quality_metrics</code>    | character vector of quality metrics to use in the model                                                                         |
| <code>temporal_direction</code> | character vector of same length as <code>quality_metrics</code> indicating temporal feature to create.                          |
| <code>missing_run_count</code>  | numeric, maximum allowed fraction of missing runs per feature.                                                                  |
| <code>n_feat</code>             | numeric, maximum number of features per protein to use in the model.                                                            |
| <code>run_order</code>          | data.frame with two columns: Run and Order. Order should be numeric and indicate the order of runs.                             |
| <code>n_trees</code>            | numeric, number of trees to use in the isolation forest model. Default is 100.                                                  |
| <code>max_depth</code>          | numeric or "auto", maximum depth of each tree. Default is "auto" which sets depth to $\log_2(N)$ where N is the number of runs. |
| <code>cores</code>              | numeric, number of cores to use for parallel processing. Default is 1.                                                          |

**Value**

data.table

---

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,
  anomaly_metrics = c()
)
```

## Arguments

|                  |                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| input            | data.table processed by the MSstatsPreprocess function                                                                                                                                                                                                                                                                                                                                                                |
| feature_columns  | str, names of columns that define spectral features                                                                                                                                                                                                                                                                                                                                                                   |
| fill_incomplete  | if TRUE (default), ensures that rows with missing data for specific features are added as NA. For example, if the y10 ion of peptideA is measured in the "disease" samples but entirely missing for the "healthy" samples, rows with NA values will be created for the y10 ion of peptideA in the "healthy" group. This process increases the number of rows to account for all possible feature-sample combinations. |
| 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.                                                                                                                                                                                                                                                                                                                                      |
| anomaly_metrics  | character vector of names of columns with quality metrics                                                                                                                                                                                                                                                                                                                                                             |

## 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

Clean Protein Prospector 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,
  calculateAnomalyScores,
  anomalyModelFeatures
)

## 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, MBR = TRUE, qvalue_cutoff = 0.05)

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

```

## Arguments

|                                      |                                                                                                                                                                                                                 |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>msstats_object</code>          | object that inherits from <code>MSstatsInputFiles</code> class.                                                                                                                                                 |
| <code>...</code>                     | additional parameter to specific cleaning functions.                                                                                                                                                            |
| <code>use_frag</code>                | TRUE will use the selected fragment for each peptide. 'Selected_fragments' column is required.                                                                                                                  |
| <code>use_pept</code>                | TRUE will use the selected fragment for each protein 'Selected_peptides' column is required.                                                                                                                    |
| <code>protein_id_col</code>          | character, name of a column with names of proteins.                                                                                                                                                             |
| <code>remove_by_site</code>          | logical, if TRUE, proteins only identified by site will be removed.                                                                                                                                             |
| <code>channel_columns</code>         | character, regular expression that identifies channel columns in TMT data.                                                                                                                                      |
| <code>runs</code>                    | chr, vector of Run labels.                                                                                                                                                                                      |
| <code>fix_colnames</code>            | lgl, if TRUE, one of the rows will be used as colnames.                                                                                                                                                         |
| <code>quantification_column</code>   | chr, name of a column used for quantification.                                                                                                                                                                  |
| <code>protein_id_column</code>       | chr, name of a column with protein IDs.                                                                                                                                                                         |
| <code>sequence_column</code>         | chr, name of a column with peptide sequences.                                                                                                                                                                   |
| <code>remove_shared</code>           | lgl, if TRUE, shared peptides will be removed.                                                                                                                                                                  |
| <code>remove_protein_groups</code>   | if TRUE, proteins with <code>numProteins &gt; 1</code> will be removed.                                                                                                                                         |
| <code>intensity_columns_regex</code> | 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. |
| <code>intensity</code>               | chr, specifies which column will be used for Intensity.                                                                                                                                                         |
| <code>calculateAnomalyScores</code>  | logical, whether to calculate anomaly scores                                                                                                                                                                    |
| <code>anomalyModelFeatures</code>    | character vector, specifies which columns will be used for anomaly detection model. Can be NULL if <code>calculateAnomalyScores=FALSE</code> .                                                                  |

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 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.  
 qvalue\_cutoff The q-value cutoff for filtering peaks detected by MBR

### 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  
 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 | <i>MSstatsConvert: An R Package to Convert Data from Mass Spectrometry Signal Processing Tools to MSstats Format</i> |
|----------------|----------------------------------------------------------------------------------------------------------------------|

---

## 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.

## Author(s)

**Maintainer:** Mateusz Staniak <mtst@mstaniak.pl>

Authors:

- Devon Kohler <kohler.d@northeastern.edu>
- Anthony Wu <wu.anthon@northeastern.edu>
- Meena Choi <mnchoi67@gmail.com>
- Ting Huang <thuang0703@gmail.com>
- Olga Vitek <o.vitek@northeastern.edu>

---

|               |                                                   |
|---------------|---------------------------------------------------|
| MSstatsImport | <i>Import files from signal processing tools.</i> |
|---------------|---------------------------------------------------|

---

## Description

Import files from signal processing tools.

## Usage

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

**Arguments**

|                           |                                                                                                                                                                             |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>input_files</code>  | list of paths to input files or <code>data.frame</code> objects. Interpretation of this parameter depends on values of parameters <code>type</code> and <code>tool</code> . |
| <code>type</code>         | chr, "MSstats" or "MSstatsTMT".                                                                                                                                             |
| <code>tool</code>         | chr, name of a signal processing tool that generated input files.                                                                                                           |
| <code>tool_version</code> | not implemented yet. In the future, this parameter will allow handling different versions of each signal processing tools.                                                  |
| <code>...</code>          | optional additional parameters to <code>data.table::fread</code> .                                                                                                          |

**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"))
```

---

`MSstatsInputFiles-class`

*Class to model files that describe a single MS dataset.*

---

**Description**

Class to model files that describe a single MS dataset.

`MSstatsDIAUmpireFiles`: class for DIAUmpire files.

`MSstatsMaxQuantFiles`: class for MaxQuant files.

`MSstatsOpenMSFiles`: class for OpenMS files.

`MSstatsOpenSWATHFiles`: class for OpenSWATH files.

`MSstatsProgenesisFiles`: class for Progenesis files.

`MSstatsProteomeDiscovererFiles`: class for ProteomeDiscoverer files.

`MSstatsSkylineFiles`: class for Skyline files.

`MSstatsSkylineFiles`: class for SpectroMine files.

`MSstatsSpectronautFiles`: class for Spectronaut files.

MSstatsPhilosopherFiles: class for Philosopher files.

MSstatsDIANNFiles: class for DIA-NN files.

MSstatsFragPipeFiles: class for FragPipe files.

MSstatsMetamorpheusFiles: class for Metamorpheus files.

MSstatsProteinProspectorFiles: class for ProteinProspector files.

## Slots

**files** named list of files generated by a signal processing tools. In most cases, this will be a single file named input. In some cases, multiple files are used, for example MaxQuant outputs evidence and proteinGroups files.

**type** character: "MSstats" or "MSstatsTMT".

**tool** character: name of a signal processing tools that generated the output. Possible values are: DIAUmpire, MaxQuant, OpenMS, OpenSWATH, Progenesis, ProteomeDiscoverer, Skyline, SpectroMine, Spectronaut.

**version** description of a software version of the signal processing tool. Not implemented yet.

---

|                     |                                                                  |
|---------------------|------------------------------------------------------------------|
| MSstatsLogsSettings | <i>Set how MSstats will log information from data processing</i> |
|---------------------|------------------------------------------------------------------|

---

## 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

|                      |                                                                                                                                                                                                     |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>use_log_file</b>  | logical. If TRUE, information about data processing will be saved to a file.                                                                                                                        |
| <b>append</b>        | logical. If TRUE, information about data processing will be added to an existing log file.                                                                                                          |
| <b>verbose</b>       | logical. If TRUE, information about data processing will be printed to the console.                                                                                                                 |
| <b>log_file_path</b> | 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 | <i>Preprocess outputs from MS signal processing tools for analysis with MSstats</i> |
|-------------------|-------------------------------------------------------------------------------------|

---

**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,
  anomaly_metrics = c(),
  ...
)
```

**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 be summed.

`anomaly_metrics`  
character vector of names of columns with quality metrics. Default is missing and is not required if anomaly model not run.

... 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.            |

|                            |                                                                                                                                                                                                                   |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>use_log_file</code>  | logical. If TRUE, information about data processing will be saved to a file.                                                                                                                                      |
| <code>append</code>        | logical. If TRUE, information about data processing will be added to an existing log file.                                                                                                                        |
| <code>verbose</code>       | logical. If TRUE, information about data processing will be printed to the console.                                                                                                                               |
| <code>log_file_path</code> | 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 <code>append = TRUE</code> , has to be a valid path to a file. |
| <code>...</code>           | additional parameters to <code>data.table::fread</code> .                                                                                                                                                         |

**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 = OpenMSstoMSstatsFormat(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 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. |
| ...                        | 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 | <i>Import Proteome Discoverer files</i> |
|-------------------|-----------------------------------------|

---

**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, BioRepl-icate, Run information. 'Run' will be matched with 'Spectrum.File'. |
| useNumProteinsColumn | TRUE removes peptides which have more than 1 in # Proteins column of PD output.                                                                         |

|                                         |                                                                                                                                                                                                                   |
|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>useUniquePeptide</code>           | TRUE (default) removes peptides that are assigned for more than one proteins. We assume to use unique peptide for each protein.                                                                                   |
| <code>summaryforMultipleRows</code>     | max(default) or sum - when there are multiple measurements for certain feature and certain run, use highest or sum of multiple intensities.                                                                       |
| <code>removeFewMeasurements</code>      | TRUE (default) will remove the features that have 1 or 2 measurements across runs.                                                                                                                                |
| <code>removeOxidationMpeptides</code>   | TRUE will remove the peptides including 'oxidation (M)' in modification. FALSE is default.                                                                                                                        |
| <code>removeProtein_with1Peptide</code> | TRUE will remove the proteins which have only 1 peptide and charge. FALSE is default.                                                                                                                             |
| <code>which.quantification</code>       | Use 'Precursor.Area'(default) column for quantified intensities. 'Intensity' or 'Area' can be used instead.                                                                                                       |
| <code>which.proteinid</code>            | Use 'Protein.Accessions'(default) column for protein name. 'Master.Protein.Accessions' can be used instead.                                                                                                       |
| <code>which.sequence</code>             | Use 'Sequence'(default) column for peptide sequence. 'Annotated.Sequence' can be used instead.                                                                                                                    |
| <code>use_log_file</code>               | logical. If TRUE, information about data processing will be saved to a file.                                                                                                                                      |
| <code>append</code>                     | logical. If TRUE, information about data processing will be added to an existing log file.                                                                                                                        |
| <code>verbose</code>                    | logical. If TRUE, information about data processing will be printed to the console.                                                                                                                               |
| <code>log_file_path</code>              | 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 <code>append = TRUE</code> , has to be a valid path to a file. |
| <code>...</code>                        | additional parameters to <code>data.table::fread</code> .                                                                                                                                                         |

**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 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.

... 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)
```

---

ProteinProspectortoMSstatsTMTFormat

*Generate MSstatsTMT required input format from Protein Prospector output*

---

## Description

Generate MSstatsTMT required input format from Protein Prospector output

## Usage

```
ProteinProspectortoMSstatsTMTFormat(
  input,
  annotation,
  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.                                             |
| 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 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**

data.frame of class "MSstatsTMT"

**Examples**

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

---

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**

|                                         |                                                                                                                                                                                                                                                                 |
|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>input</code>                      | name of MSstats input report from Skyline, which includes feature-level data.                                                                                                                                                                                   |
| <code>annotation</code>                 | name of 'annotation.txt' data which includes Condition, BioReplicate, Run. If annotation is already complete in Skyline, use <code>annotation=NULL</code> (default). It will use the annotation information from input.                                         |
| <code>removeiRT</code>                  | TRUE (default) will remove the proteins or peptides which are labeled 'iRT' in 'StandardType' column. FALSE will keep them.                                                                                                                                     |
| <code>filter_with_Qvalue</code>         | TRUE (default) will filter out the intensities that have greater than <code>qvalue_cutoff</code> in <code>DetectionQValue</code> column. Those intensities will be replaced with zero and will be considered as censored missing values for imputation purpose. |
| <code>qvalue_cutoff</code>              | Cutoff for <code>DetectionQValue</code> . default is 0.01.                                                                                                                                                                                                      |
| <code>useUniquePeptide</code>           | TRUE (default) removes peptides that are assigned for more than one proteins. We assume to use unique peptide for each protein.                                                                                                                                 |
| <code>removeFewMeasurements</code>      | TRUE (default) will remove the features that have 1 or 2 measurements across runs.                                                                                                                                                                              |
| <code>removeOxidationMpeptides</code>   | TRUE will remove the peptides including 'oxidation (M)' in modification. FALSE is default.                                                                                                                                                                      |
| <code>removeProtein_with1Feature</code> | TRUE will remove the proteins which have only 1 feature, which is the combination of peptide, precursor charge, fragment and charge. FALSE is default.                                                                                                          |
| <code>use_log_file</code>               | logical. If TRUE, information about data processing will be saved to a file.                                                                                                                                                                                    |
| <code>append</code>                     | logical. If TRUE, information about data processing will be added to an existing log file.                                                                                                                                                                      |
| <code>verbose</code>                    | logical. If TRUE, information about data processing will be printed to the console.                                                                                                                                                                             |
| <code>log_file_path</code>              | 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 <code>append = TRUE</code> , has to be a valid path to a file.                                               |
| <code>...</code>                        | additional parameters to <code>data.table::fread</code> .                                                                                                                                                                                                       |

**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",
  excludedFromQuantificationFilter = TRUE,
  filter_with_Qvalue = FALSE,
  qvalue_cutoff = 0.01,
  useUniquePeptide = TRUE,
  removeFewMeasurements = TRUE,
  removeProtein_with1Feature = FALSE,
  summaryforMultipleRows = max,
  calculateAnomalyScores = FALSE,
  anomalyModelFeatures = c(),
  anomalyModelFeatureTemporal = c(),
  removeMissingFeatures = 0.5,
  anomalyModelFeatureCount = 100,
  runOrder = NULL,
  n_trees = 100,
  max_depth = "auto",
  numberOfCores = 1,
  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.                                                                                                                                                                                                                                                    |
| excludedFromQuantificationFilter | Remove rows with F.ExcludedFromQuantification=TRUE Default is TRUE.                                                                                                                                                                                                                                                                                                  |
| filter_with_Qvalue               | FALSE(default) will not perform any filtering. TRUE 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.                                                                                                                                                                                                                          |
| calculateAnomalyScores           | Default is FALSE. If TRUE, will run anomaly detection model and calculate anomaly scores for each feature. Used downstream to weigh measurements in differential analysis.                                                                                                                                                                                           |
| anomalyModelFeatures             | character vector of quality metric column names to be used as features in the anomaly detection model. List must not be empty if calculateAnomalyScores=TRUE.                                                                                                                                                                                                        |
| anomalyModelFeatureTemporal      | character vector of temporal direction corresponding to columns passed to anomalyModelFeatures. Values must be one of: mean_decrease, mean_increase, dispersion_increase, or NULL (to perform no temporal feature engineering). Default is empty vector. If calculateAnomalyScores=TRUE, vector must have as many values as anomalyModelFeatures (even if all NULL). |
| removeMissingFeatures            | Remove features with missing values in more than this fraction of runs. Default is 0.5. Only used if calculateAnomalyScores=TRUE.                                                                                                                                                                                                                                    |

|                          |                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| anomalyModelFeatureCount | Feature selection for anomaly model. Anomaly detection works on the precursor-level and can be much slower if all features used. We will by default filter to the top-100 highest intensity features. This can be adjusted as necessary. To turn feature-selection off, set this value to a high number (e.g. 10000). Only used if calculateAnomalyScores=TRUE. |
| runOrder                 | Temporal order of MS runs. Should be a two column data.table with columns Run and Order, where Run matches the run name output by Spectronaut and Order is an integer. Used to engineer the temporal features defined in anomaly-ModelFeatureTemporal.                                                                                                          |
| n_trees                  | Number of trees to use in isolation forest when calculateAnomalyScores=TRUE. Default is 100.                                                                                                                                                                                                                                                                    |
| max_depth                | Max tree depth to use in isolation forest when calculateAnomalyScores=TRUE. Default is "auto" which calculates depth as $\log_2(N)$ where N is the number of runs. Otherwise must be an integer.                                                                                                                                                                |
| numberOfCores            | Number of cores for parallel processing anomaly detection model. When > 1, a logfile named 'MSstats_anomaly_model_progress.log' is created to track progress. Only works for Linux & Mac OS. Default is 1.                                                                                                                                                      |
| 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.                                                                                                                                                             |
| ...                      | 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

## \* internal

- .MSstatsFormat, 36
- .addFractions, 4
- .adjustIntensities, 5
- .aggregatePSMstoPeptideIons, 5
- .checkAnnotation, 6
- .checkDDA, 6
- .checkDuplicatedMeasurements, 7
- .checkMSstatsParams, 7
- .checkMultiRun, 8
- .checkOverlappedFeatures, 8
- .cleanByFeature, 9
- .cleanRawDIANN, 9
- .cleanRawDIAUmpire, 10
- .cleanRawMaxQuant, 11
- .cleanRawMetamorpheus, 11
- .cleanRawOpenMS, 12
- .cleanRawOpenSWATH, 12
- .cleanRawPDMSstats, 14
- .cleanRawPDTMT, 14
- .cleanRawPhilosopher, 15
- .cleanRawProgenesis, 16
- .cleanRawSkyline, 17
- .cleanRawSpectroMineTMT, 17
- .cleanRawSpectronaut, 18
- .countCommonFeatures, 18
- .fillValues, 19
- .filterByPattern, 19
- .filterByScore, 20
- .filterExact, 21
- .filterFewMeasurements, 21
- .filterManyColumns, 22
- .filterOverlapped, 23
- .findAvailable, 23
- .fixBasicColumns, 24
- .fixColumnTypes, 24
- .fixMissingValues, 25
- .getChannelColumns, 25
- .getCorrectFraction, 26
- .getDataTable, 26
- .getFullDesign, 27
- .getMissingRunsPerFeature, 27
- .getOverlappingFeatures, 28
- .handleFiltering, 28
- .handleFractions, 29
- .handleFractionsLF, 29
- .handleFractionsTMT, 30
- .handleIsotopicPeaks, 30
- .handleSharedPeptides, 31
- .handleSingleFeaturePerProtein, 31
- .logConverterOptions, 32
- .logSuccess, 33
- .makeBalancedDesign, 33
- .makeExactFilterMessage, 34
- .makeScoreFilterMessage, 34
- .mergeAnnotation, 35
- .nullAppender, 36
- .onLoad, 37
- .removeOverlappingFeatures, 37
- .removeSharedPeptides, 38
- .selectMSstatsColumns, 38
- .sharedParametersAmongConverters, 39
- .standardizeColnames, 40
- .summarizeMultipleMeasurements, 40
- .summarizeMultiplePSMs, 41
- .validateMSstatsConverterParameters, 41
- getDataType, 50
- MSstatsConvert, 61
- MSstatsInputFiles-class, 62
- .MSstatsFormat, 36
- .addFractions, 4
- .adjustIntensities, 5
- .aggregatePSMstoPeptideIons, 5
- .checkAnnotation, 6
- .checkDDA, 6
- .checkDuplicatedMeasurements, 7

- .checkMSstatsParams, 7
- .checkMultiRun, 8
- .checkOverlappedFeatures, 8
- .cleanByFeature, 9
- .cleanRawDIANN, 9
- .cleanRawDIAUmpire, 10
- .cleanRawMaxQuant, 11
- .cleanRawMetamorpheus, 11
- .cleanRawOpenMS, 12
- .cleanRawOpenSWATH, 12
- .cleanRawPD, 13
- .cleanRawPDMSstats, 14
- .cleanRawPDTMT, 14
- .cleanRawPhilosopher, 15
- .cleanRawProgenesis, 16
- .cleanRawSkyline, 17
- .cleanRawSpectroMineTMT, 17
- .cleanRawSpectronaut, 18
- .countCommonFeatures, 18
- .fillValues, 19
- .filterByPattern, 19
- .filterByScore, 20
- .filterExact, 21
- .filterFewMeasurements, 21
- .filterManyColumns, 22
- .filterOverlapped, 23
- .findAvailable, 23
- .fixBasicColumns, 24
- .fixColumnTypes, 24
- .fixMissingValues, 25
- .getChannelColumns, 25
- .getCorrectFraction, 26
- .getDataTable, 26
- .getFullDesign, 27
- .getMissingRunsPerFeature, 27
- .getOverlappingFeatures, 28
- .handleFiltering, 28
- .handleFractions, 29
- .handleFractionsLF, 29
- .handleFractionsTMT, 30
- .handleIsotopicPeaks, 30
- .handleSharedPeptides, 31
- .handleSingleFeaturePerProtein, 31
- .logConverterOptions, 32
- .logSuccess, 33
- .makeBalancedDesign, 33
- .makeExactFilterMessage, 34
- .makeScoreFilterMessage, 34
- .mergeAnnotation, 35
- .nullAppender, 36
- .onLoad, 37
- .removeOverlappingFeatures, 37
- .removeSharedPeptides, 38
- .selectMSstatsColumns, 38
- .sharedParametersAmongConverters, 39
- .standardizeColNames, 40
- .summarizeMultipleMeasurements, 40
- .summarizeMultiplePSMs, 41
- .validateMSstatsConverterParameters, 41
- as.data.frame.MSstatsValidated, 43
- as.data.table.MSstatsValidated, 43
- CheckDataHealth, 44
- DIANNtoMSstatsFormat, 44
- DIAUmpiretoMSstatsFormat, 46
- FragPipetoMSstatsFormat, 48
- getDataType, 50
- getDataType, MSstatsInputFiles-method (getDataType), 50
- getInputFile, 50
- getInputFile, MSstatsInputFiles-method (getInputFile), 50
- getInputFile, MSstatsPhilosopherFiles-method (getInputFile), 50
- MaxQtoMSstatsFormat, 51
- MetamorpheusToMSstatsFormat, 53
- MSstatsAnomalyScores, 55
- MSstatsBalancedDesign, 56, 61
- MSstatsClean, 57, 61
- MSstatsClean, MSstatsDIANNFiles-method (MSstatsClean), 57
- MSstatsClean, MSstatsDIAUmpireFiles-method (MSstatsClean), 57
- MSstatsClean, MSstatsMaxQuantFiles-method (MSstatsClean), 57
- MSstatsClean, MSstatsMetamorpheusFiles-method (MSstatsClean), 57
- MSstatsClean, MSstatsOpenMSFiles-method (MSstatsClean), 57
- MSstatsClean, MSstatsOpenSWATHFiles-method (MSstatsClean), 57

MSstatsClean, MSstatsPhilosopherFiles-method  
     (MSstatsClean), [57](#)  
 MSstatsClean, MSstatsProgenesisFiles-method  
     (MSstatsClean), [57](#)  
 MSstatsClean, MSstatsProteinProspectorFiles-method  
     (MSstatsClean), [57](#)  
 MSstatsClean, MSstatsProteomeDiscovererFiles-method  
     (MSstatsClean), [57](#)  
 MSstatsClean, MSstatsSkylineFiles-method  
     (MSstatsClean), [57](#)  
 MSstatsClean, MSstatsSpectroMineFiles-method  
     (MSstatsClean), [57](#)  
 MSstatsClean, MSstatsSpectronautFiles-method  
     (MSstatsClean), [57](#)  
 MSstatsConvert, [61](#)  
 MSstatsConvert-package  
     (MSstatsConvert), [61](#)  
 MSstatsDIANNFiles-class  
     (MSstatsInputFiles-class), [62](#)  
 MSstatsDIAUmpireFiles-class  
     (MSstatsInputFiles-class), [62](#)  
 MSstatsFragPipeFiles-class  
     (MSstatsInputFiles-class), [62](#)  
 MSstatsImport, [61](#), [61](#)  
 MSstatsInputFiles-class, [62](#)  
 MSstatsLogsSettings, [61](#), [63](#)  
 MSstatsMakeAnnotation, [64](#)  
 MSstatsMaxQuantFiles-class  
     (MSstatsInputFiles-class), [62](#)  
 MSstatsMetamorpheusFiles-class  
     (MSstatsInputFiles-class), [62](#)  
 MSstatsOpenMSFiles-class  
     (MSstatsInputFiles-class), [62](#)  
 MSstatsOpenSWATHFiles-class  
     (MSstatsInputFiles-class), [62](#)  
 MSstatsPhilosopherFiles-class  
     (MSstatsInputFiles-class), [62](#)  
 MSstatsPreprocess, [61](#), [65](#)  
 MSstatsProgenesisFiles-class  
     (MSstatsInputFiles-class), [62](#)  
 MSstatsProteinProspectorFiles-class  
     (MSstatsInputFiles-class), [62](#)  
 MSstatsProteomeDiscovererFiles-class  
     (MSstatsInputFiles-class), [62](#)  
 MSstatsSaveSessionInfo, [67](#)  
 MSstatsSkylineFiles-class  
     (MSstatsInputFiles-class), [62](#)  
 MSstatsSpectroMineFiles-class  
     (MSstatsInputFiles-class), [62](#)  
 MSstatsSpectronautFiles-class  
     (MSstatsInputFiles-class), [62](#)  
 MSstatsToMSstatsFormat, [68](#)  
 OpenMS  
     OpenSWATHtoMSstatsFormat, [69](#)  
     PDtoMSstatsFormat, [71](#)  
     ProgenesisistoMSstatsFormat, [73](#)  
     ProteinProspectortoMSstatsTMTFormat,  
         [75](#)  
     SkylinetoMSstatsFormat, [76](#)  
     SpectronauttoMSstatsFormat, [78](#)