Package: MSstatsBig (via r-universe)

November 20, 2026

Type Package

Title MSstats Preprocessing for Larger than Memory Data

Version 1.10.0

Description MSstats package provide tools for preprocessing, summarization and differential analysis of mass spectrometry (MS) proteomics data. Recently, some MS protocols enable acquisition of data sets that result in larger than memory quantitative data. MSstats functions are not able to process such data. MSstatsBig package provides additional converter functions that enable processing larger than memory data sets.

License Artistic-2.0

Encoding UTF-8

RoxygenNote 7.3.3

Imports arrow, DBI, dplyr, MSstats, MSstatsConvert, readr, sparklyr, utils

Suggests testthat, mockery, knitr, rmarkdown

VignetteBuilder knitr

biocViews MassSpectrometry, Proteomics, Software

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libxml2-dev libssl-dev libx11-dev

Repository Bioconductor 3.23

Date/Publication 2026-04-28 13:01:48 UTC

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

RemoteRef RELEASE_3_23

RemoteSha 69ab29d6f7b5db0879d854b4fef400d8131e7e20

Contents

bigDIANNtoMSstatsFormat 2
bigFragPipetoMSstatsFormat 3
bigSpectronauttoMSstatsFormat 5
MSstatsAddAnnotationBig 6
MSstatsPreprocessBig 7

Index **10**

`bigDIANNtoMSstatsFormat` *Convert out-of-memory DIANN files to MSstats format.*

Description

Convert out-of-memory DIANN files to MSstats format.

Usage

```
bigDIANNtoMSstatsFormat(
  input_file,
  annotation = NULL,
  output_file_name,
  backend,
  MBR = TRUE,
  quantificationColumn = "FragmentQuantCorrected",
  global_qvalue_cutoff = 0.01,
  qvalue_cutoff = 0.01,
  pg_qvalue_cutoff = 0.01,
  max_feature_count = 100,
  filter_unique_peptides = FALSE,
  aggregate_psms = FALSE,
  filter_few_obs = FALSE,
  remove_annotation = FALSE,
  calculateAnomalyScores = FALSE,
  anomalyModelFeatures = c(),
  connection = NULL
)
```

Arguments

`input_file`	name of the input text file in 10-column MSstats format.
`annotation`	name of 'annotation.txt' data which includes Condition, BioReplicate, Run.
`output_file_name`	name of an output file which will be saved after pre-processing
`backend`	"arrow" or "sparklyr". Option "sparklyr" requires a spark installation and connection to spark instance provided in the 'connection' parameter.
`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.

`global_qvalue_cutoff`
: The qvalue cutoff for the Q.Value column, i.e. the run-specific precursor q-value. Default is 0.01.

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

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

`max_feature_count`
: maximum number of features per protein. Features will be selected based on highest average intensity.

`filter_unique_peptides`
: If TRUE, shared peptides will be removed. Please refer to the 'Details' section for additional information.

`aggregate_psms`
: If TRUE, multiple measurements per PSM in a Run will be aggregated (by taking maximum value). Please refer to the 'Details' section for additional information.

`filter_few_obs`
: If TRUE, feature with less than 3 observations across runs will be removed. Please refer to the 'Details' section for additional information.

`remove_annotation`
: If TRUE, columns BioReplicate and Condition will be removed to reduce output file size. These will need to be added manually later before using dataProcess function. Only applicable to sparklyr backend.

`calculateAnomalyScores`
: If TRUE, will carry anomaly model features through pipeline

`anomalyModelFeatures`
: Character vector of column names to be carried through the pipeline

`connection`
: Connection to a spark instance created with the 'spark_connect' function from 'sparklyr' package.

Value

either arrow object or sparklyr table that can be optionally collected into memory by using dplyr::collect function.

`bigFragPipetoMSstatsFormat` *Convert out-of-memory FragPipe files to MSstats format.*

Description

Convert out-of-memory FragPipe files to MSstats format.

Usage

```
bigFragPipetoMSstatsFormat(
  input_file,
  output_file_name,
  backend,
  max_feature_count = 100,
  filter_unique_peptides = FALSE,
  aggregate_psms = FALSE,
  filter_few_obs = FALSE,
  remove_annotation = FALSE,
  connection = NULL
)
```

Arguments

input_file	name of the input text file in 10-column MSstats format.
output_file_name	name of an output file which will be saved after pre-processing
backend	"arrow" or "sparklyr". Option "sparklyr" requires a spark installation and connection to spark instance provided in the ‘connection‘ parameter.
max_feature_count	maximum number of features per protein. Features will be selected based on highest average intensity.
filter_unique_peptides	If TRUE, shared peptides will be removed. Please refer to the ‘Details‘ section for additional information.
aggregate_psms	If TRUE, multiple measurements per PSM in a Run will be aggregated (by taking maximum value). Please refer to the ‘Details‘ section for additional information.
filter_few_obs	If TRUE, feature with less than 3 observations across runs will be removed. Please refer to the ‘Details‘ section for additional information.
remove_annotation	If TRUE, columns BioReplicate and Condition will be removed to reduce output file size. These will need to be added manually later before using dataProcess function. Only applicable to sparklyr backend.
connection	Connection to a spark instance created with the ‘spark_connect‘ function from ‘sparklyr‘ package.

Value

either arrow object or sparklyr table that can be optionally collected into memory by using dplyr::collect function.

Examples

```
converted_data <- bigFragPipetoMSstatsFormat(
  system.file("extdata", "fgexample.csv", package = "MSstatsBig"),
```

```
  "output_file.csv",
  backend = "arrow")
converted_data <- dplyr::collect(converted_data)
head(converted_data)
```

`bigSpectronauttoMSstatsFormat` *Convert out-of-memory Spectronaut files to MSstats format.*

Description

Convert out-of-memory Spectronaut files to MSstats format.

Usage

```
bigSpectronauttoMSstatsFormat(
  input_file,
  output_file_name,
  backend,
  intensity = "F.NormalizedPeakArea",
  filter_by_excluded = FALSE,
  filter_by_identified = FALSE,
  filter_by_qvalue = FALSE,
  qvalue_cutoff = 0.01,
  max_feature_count = 100,
  filter_unique_peptides = FALSE,
  aggregate_psms = FALSE,
  filter_few_obs = FALSE,
  remove_annotation = FALSE,
  calculateAnomalyScores = FALSE,
  anomalyModelFeatures = c(),
  connection = NULL
)
```

Arguments

`input_file`	name of the input text file in 10-column MSstats format.
`output_file_name`	name of an output file which will be saved after pre-processing
`backend`	"arrow" or "sparklyr". Option "sparklyr" requires a spark installation and connection to spark instance provided in the ‘connection‘ parameter.
`intensity`	Name of the intensity column to be used in Spectronaut
`filter_by_excluded`	if TRUE, will filter by the ‘F.ExcludedFromQuantification‘ column.
`filter_by_identified`	if TRUE, will filter by the ‘EG.Identified‘ column.

filter_by_qvalue
: if TRUE, will filter by EG.Qvalue and PG.Qvalue columns.

qvalue_cutoff
: cutoff which will be used for q-value filtering.

max_feature_count
: maximum number of features per protein. Features will be selected based on highest average intensity.

filter_unique_peptides
: If TRUE, shared peptides will be removed. Please refer to the 'Details' section for additional information.

aggregate_psms
: If TRUE, multiple measurements per PSM in a Run will be aggregated (by taking maximum value). Please refer to the 'Details' section for additional information.

filter_few_obs
: If TRUE, feature with less than 3 observations across runs will be removed. Please refer to the 'Details' section for additional information.

remove_annotation
: If TRUE, columns BioReplicate and Condition will be removed to reduce output file size. These will need to be added manually later before using dataProcess function. Only applicable to sparklyr backend.

calculateAnomalyScores
: If TRUE, will carry anomaly model features through pipeline

anomalyModelFeatures
: Character vector of column names to be carried through the pipeline

connection
: Connection to a spark instance created with the 'spark_connect' function from 'sparklyr' package.

Value

either arrow object or sparklyr table that can be optionally collected into memory by using dplyr::collect function.

Examples

```
converted_data <- bigSpectronauttoMSstatsFormat(
  system.file("extdata", "spectronaut_input.csv", package = "MSstatsBig"),
  "output_file.csv",
  backend="arrow")
converted_data <- dplyr::collect(converted_data)
head(converted_data)
```

MSstatsAddAnnotationBig

Merge annotation to output of MSstatsPreprocessBig

Description

Merge annotation to output of MSstatsPreprocessBig

Usage

```
MSstatsAddAnnotationBig(input, annotation)
```

Arguments

input	output of MSstatsPreprocessBig
annotation	run annotation

Value

table of ‘input‘ and ‘annotation‘ merged by Run column.

Examples

```
converted_data <- bigFragPipetoMSstatsFormat(
  system.file("extdata", "fgexample.csv", package = "MSstatsBig"),
  "output_file.csv",
  backend = "arrow")
converted_data <- dplyr::collect(converted_data)
head(converted_data)
# Change annotation as an example:
converted_data$Condition <- NULL
converted_data$BioReplicate <- NULL
annot <- data.frame(Run = unique(converted_data[["Run"]]))
annot$BioReplicate <- rep(1:53, times = 2)
annot$Condition <- rep(1:2, each = 53)
head(MSstatsAddAnnotationBig(converted_data, annot))
```

MSstatsPreprocessBig *General converter for larger-than-memory csv files in MSstats format 10-column format*

Description

General converter for larger-than-memory csv files in MSstats format 10-column format

Usage

```
MSstatsPreprocessBig(
  input_file,
  output_file_name,
  backend,
  max_feature_count = 100,
  filter_unique_peptides = FALSE,
  aggregate_psms = FALSE,
  filter_few_obs = FALSE,
  remove_annotation = FALSE,
```

```
  calculateAnomalyScores = FALSE,
  anomalyModelFeatures = c(),
  connection = NULL
)
```

Arguments

input_file	name of the input text file in 10-column MSstats format.
output_file_name	name of an output file which will be saved after pre-processing
backend	"arrow" or "sparklyr". Option "sparklyr" requires a spark installation and connection to spark instance provided in the ‘connection‘ parameter.
max_feature_count	maximum number of features per protein. Features will be selected based on highest average intensity.
filter_unique_peptides	If TRUE, shared peptides will be removed. Please refer to the ‘Details‘ section for additional information.
aggregate_psms	If TRUE, multiple measurements per PSM in a Run will be aggregated (by taking maximum value). Please refer to the ‘Details‘ section for additional information.
filter_few_obs	If TRUE, feature with less than 3 observations across runs will be removed. Please refer to the ‘Details‘ section for additional information.
remove_annotation	If TRUE, columns BioReplicate and Condition will be removed to reduce output file size. These will need to be added manually later before using dataProcess function. Only applicable to sparklyr backend.
calculateAnomalyScores	If TRUE, will carry anomaly model features through pipeline
anomalyModelFeatures	Character vector of column names to be carried through the pipeline
connection	Connection to a spark instance created with the ‘spark_connect‘ function from ‘sparklyr‘ package.

Details

Filtering and aggregation may be very time consuming and the ability to perform them in a given R session depends on available memory, settings of external packages, etc. Hence, all value of related parameters (‘filter_unique_peptides‘, ‘aggregate_psms‘, ‘filter_few_obs‘) are set to FALSE by default and only feature selection is performed, which saves both computation time and memory. Appropriately configured spark backend provides the most consistent way to perform these operations.

Value

either arrow object or sparklyr table that can be optionally collected into memory by using dplyr::collect function.

Examples

```
converted_data <- bigFragPipetoMSstatsFormat(
  system.file("extdata", "fgexample.csv", package = "MSstatsBig"),
  "tencol_format.csv",
  backend="arrow")
procd <- MSstatsPreprocessBig("tencol_format.csv", "proc_out.csv", backend = "arrow")
head(dplyr::collect(procd))
```

Index

bigDIANNtoMSstatsFormat, 2
bigFragPipetoMSstatsFormat, 3
bigSpectronauttoMSstatsFormat, 5

MSstatsAddAnnotationBig, 6
MSstatsPreprocessBig, 7