Package 'mspms'

April 8, 2026

Type Package

Title Tools for the analysis of MSP-MS data

Version 1.3.1

Description This package provides functions for the analysis of data generated by the multiplex substrate profiling by mass spectrometry for proteases (MSP-MS) method. Data exported from upstream proteomics software is accepted as input and subsequently processed for analysis.
Tools for statistical analysis, visualization, and interpretation of the data are provided.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.4.0)

biocViews Proteomics, MassSpectrometry, Preprocessing

LazyData true

Imports QFeatures, limma, SummarizedExperiment, magrittr, rlang, dplyr, purrr, stats, tidyr, stringr, ggplot2, ggseqlogo, heatmaply, readr, rstatix, tibble, ggpubr, imputeLCMD

Suggests knitr, testthat (>= 3.0.0), downloadthis, DT, rmarkdown, BiocStyle

Config/testthat/edition 3

URL https://github.com/baynec2/mspms

BugReports https://github.com/baynec2/mspms/issues

VignetteBuilder knitr

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

git_branch devel

git_last_commit 20f2c4b

git_last_commit_date 2025-11-12

Repository Bioconductor 3.23

Date/Publication 2026-04-07

Author Charlie Bayne [aut, cre] (ORCID:
<https://orcid.org/0000-0003-1870-5298>)

Maintainer Charlie Bayne <baynec2@gmail.com>

Contents

`mspms-package` *mspms: Tools for the analysis of MSP-MS data*

Description

This package provides functions for the analysis of data generated by the multiplex substrate profiling by mass spectrometry for proteases (MSP-MS) method. Data exported from upstream proteomics software is accepted as input and subsequently processed for analysis. Tools for statistical analysis, visualization, and interpretation of the data are provided.

Author(s)

Maintainer: Charlie Bayne <baynec2@gmail.com> (ORCID)

See Also

Useful links:

- https://github.com/baynec2/mspms
- Report bugs at https://github.com/baynec2/mspms/issues

`add_cleavages` *add_cleavages*

Description

Adds cleavage information to a tibble by wraping the n_term_cleavage and c_term_cleavage functions into a consolidated function.

Usage

```
add_cleavages(joined_with_library, n_residues = 4)
```

Arguments

`joined_with_library`	a tibble containing columns named "peptide", "library_match_sequence", and "library_real_sequence".
`n_residues`	the number of residues to the left and right of the cleavage site to include in the output.

Value

a tibble with cleavage information added.

`add_peptide_data` *add_peptide_data*

Description

adds peptide information for every peptide in the data.

Usage

```
add_peptide_data(tibble, qf)
```

Arguments

`tibble`	tibble you would like to add peptide info to. Must have column named peptide
`qf`	a QFeatures object with rowData for peptides. cleavage_seq, cleavage_pos, and cleavage_type.

Value

a tibble with column named peptide.

`all_possible_8mers_from_228_library`

all_possible_8mers_from_228_library All possible 8mers from the standard (as of 26April2024) 228 MSP-MS peptide library (This is equivalent to the result of mspms::calculate_all_cleavages(mspms::peptide_library$real_cleavage_seq,n=4)) vector of the 14 AA peptides used in the library.

Description

all_possible_8mers_from_228_library All possible 8mers from the standard (as of 26April2024)
228 MSP-MS peptide library (This is equivalent to the result of mspms::calculate_all_cleavages(mspms::peptide_library$real
vector of the 14 AA peptides used in the library.

Usage

```
all_possible_8mers_from_228_library
```

Format

‘all_possible_8mers_from_228_library‘ A vector with 2964 entries

Source

<standard peptide library used with MSP-MS method in the O’Donoghue lab as of 26April2024>

`calculate_all_cleavages`

calculate_all_cleavages calculate all possible cleavages for a defined peptide library containing peptides of the same length.

Description

calculate_all_cleavages calculate all possible cleavages for a defined peptide library containing peptides of the same length.

Usage

```
calculate_all_cleavages(peptide_library_seqs, n_AA_after_cleavage = 4)
```

Arguments

`peptide_library_seqs`
: The sequences of each peptide in the peptide library. They should all be the same length.

`n_AA_after_cleavage`
: The number of AA after (and before) the cleavage site to consider.

Value

a vector of all the possible cleavages for the peptide library sequences

Examples

```
calculate_all_cleavages(mspms::peptide_library$library_real_sequence,
  n_AA_after_cleavage = 4
)
```

`calc_AA_count_of_motif` *calc_AA_count_of_motif*

Description

Calculate the counts of amino acids at each position of a motif for all the sequences in a vector.

Usage

```
calc_AA_count_of_motif(cleavage_motif)
```

Arguments

`cleavage_motif` a vector of cleavage motifs

Value

a matrix of counts

calc_AA_fc *calc_AA_fc*

Description

Calculate the fold change of each amino acid by position.

Usage

```
calc_AA_fc(experimental_prop_matrix, background_prop_matrix, sig_zscores)
```

Arguments

experimental_prop_matrix	a matrix of the experimental proportions (from your vector of cleavage sequences) at each position.
background_prop_matrix	a matrix of the background proportions of AAs at each position
sig_zscores	a tibble of the significant zscores.

Value

a matrix

calc_AA_motif_zscore *calc_AA_motif_zscore*

Description

Calculate the Z score for the amino acids at each position

Usage

```
calc_AA_motif_zscore(
  background_count_matrix,
  background_prop_matrix,
  experimental_count_matrix,
  experimental_prop_matrix
)
```

Arguments

`background_count_matrix`
: the count matrix from the background sequences

`background_prop_matrix`
: the proportion matrix from the background sequences

`experimental_count_matrix`
: the count matrix from the experimental sequences

`experimental_prop_matrix`
: the proportion matrix from the experimental sequences

Value

a data frame of Zscores for each amino acid at each position.

`calc_AA_percent_difference` *calc_AA_percent_difference*

Description

Calculate the percent difference between a matrix of background proportions and a matrix of experimentally observed proportions.

Usage

```
calc_AA_percent_difference(background_prop_matrix, experimental_prop_matrix)
```

Arguments

`background_prop_matrix`
: a proportion matrix of amino acids per position from background cleavage sequences

`experimental_prop_matrix`
: a proportion matrix of amino acids per position from experimental cleavage sequences

Value

a data frame of percent differences

calc_AA_prop_of_motif *calc_AA_prop_of_motif*

Description

Calculate the proportion of amino acids at each position in a vector of motifs.

Usage

```
calc_AA_prop_of_motif(count_matrix)
```

Arguments

count_matrix	this is a matrix of the counts of cleavage motifs

Value

a matrix with proportions of counts.

calc_limma_contrasts *calc_limma_contrasts*

Description

Calculates limma contrasts given colData. The contrasts returned are pairwise relative to T0 for each timepoint assayed.

Usage

```
calc_limma_contrasts(colData, design_mat)
```

Arguments

colData	colData from mspms experiment
design_mat	design_mat as returned by calc_limma_design_matrix

Value

a contrast matrix

calc_limma_design_matrix

calc_limma_design_matrix

Description

Calculates a limma compatible design matrix for mspms data.

Usage

```
calc_limma_design_matrix(colData, norm_data)
```

Arguments

colData	colData with condition and time variables as factors
norm_data	normalized data from QFeatures object to use

Value

a model matrix

calc_per_samples_library_nd

calc_per_samples_library_nd Calculate the percentage of samples each library_id peptide was not detected in.

Description

calc_per_samples_library_nd Calculate the percentage of samples each library_id peptide was not detected in.

Usage

```
calc_per_samples_library_nd(
  processed_qf,
  peptide_library_ids = mspms::peptide_library$library_id
)
```

Arguments

processed_qf	a QFeatures object with a SummarizedExperiment named "peptides". Intended to be prepared by one of the pre-processing prepare_x_data functions of the mspms R package.
peptide_library_ids	a character vector containing the names of the library_ids

Value

a tibble containing percentage of samples each library id was detected in, both as full length, and as cleavage products.

`calc_sig_zscores` *calc_sig_zscores Determine which Zscores are significant at the given alpha for a matrix of scores*

Description

calc_sig_zscores Determine which Zscores are significant at the given alpha for a matrix of scores

Usage

```
calc_sig_zscores(zscores, pval = 0.05)
```

Arguments

`zscores`	= a data frame of zscores
`pval`	= p value threshold for significance. Default is 0.05

Value

a tibble of significant zscores

`check_file_is_valid` *check_file_is_valid*

Description

Validate that an input data frame contains all expected columns.

Usage

```
check_file_is_valid(data, expected_names, tool_name)
```

Arguments

`data`	A data frame read into R.
`expected_names`	A character vector of expected column names.
`tool_name`	A short string identifying the originating software (e.g., "PEAKS", "PD",etc).

Value

Raises an error (‘stop‘) with an informative message if required columns are missing; otherwise returns ‘invisible(NULL)‘.

check_file_is_valid_diann

check_file_is_valid_diann Check to make sure the input data looks like the expected DIA-NN output file.

Description

check_file_is_valid_diann Check to make sure the input data looks like the expected DIA-NN output file.

Usage

```
check_file_is_valid_diann(diann_data)
```

Arguments

diann_data	pg_matrix.tsv file generated by DIA-NN and read into R.

Value

a stop command with a informative message if file looks unexpected. otherwise, nothing.

check_file_is_valid_fragpipe

check_file_is_valid_fragpipe Check to make sure the input data looks like the expected FragPipe file.

Description

check_file_is_valid_fragpipe Check to make sure the input data looks like the expected FragPipe file.

Usage

```
check_file_is_valid_fragpipe(fragpipe_data)
```

Arguments

fragpipe_data	combined_peptide.tsv file generated by FragPipe read into R.

Value

a stop command with a informative message if file looks unexpected. otherwise, nothing.

check_file_is_valid_pd

check_file_is_valid_pd Check to make sure the input data looks like the expected ProteomeDiscoverer file.

Description

check_file_is_valid_pd Check to make sure the input data looks like the expected ProteomeDiscoverer file.

Usage

```
check_file_is_valid_pd(pd_data)
```

Arguments

pd_data	PeptideGroups.txt file generated by ProteomeDiscover and read into R.

Value

a stop command with a informative message if file looks unexpected. otherwise, nothing.

check_file_is_valid_peaks

check_file_is_valid_peaks Check to make sure the input data looks like the expected PEAKS file.

Description

check_file_is_valid_peaks Check to make sure the input data looks like the expected PEAKS file.

Usage

```
check_file_is_valid_peaks(peaks_data)
```

Arguments

peaks_data	protein-peptides-lfq.csv file generated by PEAKS read into R.

Value

a stop command with a informative message if file looks unexpected. otherwise, nothing.

`check_file_is_valid_sage` *check_file_is_valid_sage Check to make sure the input data looks like the expected PEAKS file.*

Description

check_file_is_valid_sage Check to make sure the input data looks like the expected PEAKS file.

Usage

```
check_file_is_valid_sage(sage_data)
```

Arguments

`sage_data`	read in lfq.tsv file output prodcued by Sage into R.

Value

a stop command with a informative message if file looks unexpected. otherwise, nothing.

`check_peptide_library` *check_peptide_library*

Description

check_peptide_library

Usage

```
check_peptide_library(peptide_library)
```

Arguments

`peptide_library`	

Value

an informative error if the column names of the peptide library are unexpected. Otherwise nothing.

cleavage *Generalized cleavage function with dynamic column names*

Description

Finds cleavage sequences for either N-terminal or C-terminal cleavages relative to a peptide library sequence.

Usage

```
cleavage(
  peptide_sequence,
  library_match_sequence,
  library_real_sequence,
  n_residues = 4,
  terminus = c("nterm", "cterm")
)
```

Arguments

peptide_sequence	Peptide sequence, single-letter code. "_" denotes cleavage site.
library_match_sequence	Sequence matched by proteomics software (may differ from real sequence).
library_real_sequence	True peptide sequence.
n_residues	Number of residues to include on each side of the cleavage site.
terminus	"nterm" or "cterm", specifying which terminus to analyze.

Value

tibble with peptide, cleavage sequence, and cleavage position. Column names are dynamically named based on the terminus.

colData *colData A tibble containing the colData associated with an experiment to proc*

Description

colData A tibble containing the colData associated with an experiment to proc

Usage

```
colData
```

Format

'colData' A tibble: 42 × 4

Source

colData corresponding to cathepsin A-D MSP-MS experiment

---

consolidate_cleavages *consolidate_cleavages*

Description

Consolidate the n term and c term cleavage data. The nterm and cterm cleavage information are consolidated into a single column and rows

Usage

```
consolidate_cleavages(cleavage_added_data)
```

Arguments

cleavage_added_data	a tibble where cleavage information has been added by add_cleavages()

Value

a tibble with the cleavage information combined into a single column and rows with no cleavage information or double information removed.

---

count_cleavages_per_pos *count_cleavages_per_pos*

Description

Count the number of cleavages per position

Usage

```
count_cleavages_per_pos(data, peptide_library = mspms::peptide_library)
```

Arguments

data	a tibble containing columns named peptide, cleavage_pos, condition, and time. Other column names can be included.
peptide_library	a peptide library tibble.

Value

a tibble with all positions filled.

`cterm_cleavage` *C-terminal cleavage sequence extraction*

Description

Wrapper for ‘cleavage()‘ that extracts the C-terminal cleavage sequence and cleavage position from a peptide relative to its library sequence.

Usage

```
cterm_cleavage(...)
```

Value

A tibble with columns: - ‘peptide‘: the input peptide sequence - ‘cterm‘: the C-terminal cleavage sequence (n residues on each side) - ‘cterm_cleavage_pos‘: the position of the C-terminal cleavage in the library sequence

---

`generate_report` *generate_report*

Description

wrapper function to generate an automatic .html report of a basic mspms analysis.

Usage

```
generate_report(
  prepared_data,
  peptide_library = mspms::peptide_library,
  n_residues = 4,
  outdir = getwd(),
  output_file = paste0(Sys.Date(), "_mspms_report.html")
)
```

Arguments

`prepared_data`	a QFeatures object containing a SummarizedExperiment named "peptides".
`peptide_library`	peptide library used with experiment. Contains columns "library_id", "library_match_sequence", and "library_real_sequence".
`n_residues`	the number of amino acid residues before and after the cleavage site to generate a cleavage seq for.
`outdir`	the output directory you would like to render the report to.
`output_file`	the file name to export.

Value

a knited .html report of the mspms analysis.

Examples

```
generate_report(mspms::peaks_prepared_data)
```

`icelogo_col_scheme` *icelogo_col_scheme Defining a color scheme for our iceLogos*

Description

icelogo_col_scheme Defining a color scheme for our iceLogos

Usage

```
icelogo_col_scheme()
```

Value

a ggseqlogo color scheme function

`limma_stats` *limma_stats*

Description

Calculates statistics for each condition relative to time 0 using limma for differential analysis. Results are then formatted to be consistent with results produced by other statistic approaches used in the mspms package (log2fc_t_test).

Usage

```
limma_stats(processed_qf)
```

Arguments

`processed_qf`	mspms data in a QFeatures object.

Value

a tibble containing statistics

Examples

```
mspms_limma_results <- limma_stats(mspms::processed_qf)
```

load_colData *load_colData*

Description

load a .csv file containing sample colData. Check for errors

Usage

```
load_colData(colData_filepath)
```

Arguments

colData_filepath	filepath to .csv file containing colData.

Value

a tibble

log2fc_t_test *log2fc_t_test*

Description

Calculates the log2 fold change and t-test statistics given a user specified reference variable and value.

Usage

```
log2fc_t_test(processed_qf, reference_variable = "time", reference_value = 0)
```

Arguments

processed_qf	mspms data in a QFeatures object.
reference_variable	the colData variable to use as reference
reference_value	the value of the colData variable to use as reference

Value

a tibble containing log2fc and t test statistics

Examples

```
log2fc_and_t_test <- log2fc_t_test(mspms::processed_qf)
```

`log2fc_t_test_data` *log2fc_t_test_data A tibble containing the results of t-tests and log2fc compared to time 0 14,497 × 19*

Description

log2fc_t_test_data A tibble containing the results of t-tests and log2fc compared to time 0 14,497 × 19

Usage

```
log2fc_t_test_data
```

Format

'peaks_prepared_data' A tibble: 14,497 × 19

Source

<mspms processed data originally from PEAKS files found in "tests/testdata/protein-peptides-id.csv" and "tests/testdata/protein-peptides-lfq.csv">

`mspms_log2fc` *mspms_log2fc*

Description

calculates the log2fc for each time point within each condition relative to a specified value for a specified reference variable.

Usage

```
mspms_log2fc(processed_qf, reference_variable = "time", reference_value = 0)
```

Arguments

`processed_qf`	a QFeatures object with a SummarizedExperiment named "peptides_norm".
`reference_variable`	the variable to used as a reference (denominator of log2 fold change).
`reference_value`	the value of the reference variable to use as the reference

Value

a tibble with the t test statistics for each peptide within each group with the supplied value at the supplied variable as reference.

`mspms_tidy` *mspms_tidy Convert a SummarizedExperiment object within a QFeatures object into a tidy tibble.*

Description

mspms_tidy Convert a SummarizedExperiment object within a QFeatures object into a tidy tibble.

Usage

```
mspms_tidy(processed_qf, se_name = "peptides_norm")
```

Arguments

`processed_qf`	a QFeature object containing rowData and colData.
`se_name`	the name of the SummarizedExperiment you would like to extract

Value

a tibble containing all the rowData, colData, and assay data for the specified SummarizedExperiment.

Examples

```
mspms_data <- mspms_tidy(mspms::processed_qf)
```

`mspms_tidy_data` *mspms_tidy_data A tibble containing tidy data derived from QFeatures object*

Description

mspms_tidy_data A tibble containing tidy data derived from QFeatures object

Usage

```
mspms_tidy_data
```

Format

‘mspms_tidy_data‘ A tibble:

Source

processed_qf

mspms_t_tests *mspms_t_tests*

Description

performs t-tests for each peptide within each group for the user specified. FDR adjustment is performed.

Usage

```
mspms_t_tests(processed_qf, reference_variable = "time", reference_value = "0")
```

Arguments

processed_qf	a QFeatures object with a SummarizedExperiment named "peptides_norm".
reference_variable	the variable to used as a reference.
reference_value	the value of the reference variable to use as the reference

Value

a tibble with the t test statistics for each peptide within each group with the supplied value at the supplied variable as reference.

nterm_cleavage *N-terminal cleavage sequence extraction*

Description

Wrapper for ‘cleavage()‘ that extracts the N-terminal cleavage sequence and cleavage position from a peptide relative to its library sequence.

Usage

```
nterm_cleavage(...)
```

Value

A tibble with columns: - ‘peptide‘: the input peptide sequence - ‘nterm‘: the N-terminal cleavage sequence (n residues on each side) - ‘nterm_cleavage_pos‘: the position of the N-terminal cleavage in the library sequence

peaks_prepared_data *peaks_prepared_data A QFeatures object prepared from PEAKS data of cathepsin data/.*

Description

peaks_prepared_data A QFeatures object prepared from PEAKS data of cathepsin data/.

Usage

```
peaks_prepared_data
```

Format

‘peaks_prepared_data‘ An instance of class QFeatures containing 1 assays: [1] peptides: SummarizedExperiment with 2071 rows and 42 columns

peptides Peptide Sequence Detected ...

Source

<mspms processed data originally from PEAKS files found in "tests/testdata/protein-peptides-id.csv" and "tests/testdata/protein-peptides-lfq.csv">

peptide_library *peptide_library*

Description

This is the 228 peptide library used by the O'Donoghue lab as of 26April2024.

Usage

```
peptide_library
```

Format

‘peptide_library‘ A data frame with 228 rows and 3 columns:

library_reference_id reference id of the detected peptide as put in upstream software

library_match_sequence the sequence match to the peptide library, methionine is replaced with norleucine,which should function the same as methionine for proteases but has the same mass as L

library_real_sequence Ls corresponding to norleucine are replaced back with n (for norleucine) ...

Source

<O'Donoghue lab as of 26April2024 >

`plot_all_icelogos` *plot_all_icelogos*

Description

Easily plot a iceLogo corresponding to peptides of interest across each condition of an experiment.

Usage

```
plot_all_icelogos(
  sig_cleavage_data,
  type = "percent_difference",
  pval = 0.05,
  background_universe = mspms::all_possible_8mers_from_228_library
)
```

Arguments

`sig_cleavage_data`	a tibble of data of interest containing a column labeled peptide, cleavage_seq, and condition
`type`	this is the type of iceLogo you would like to generate, can be either "percent_difference" or "fold_change".
`pval`	this is the pvalue threshold (<=) to consider significant when determining the significance of the sig_cleavages relative to the background at each position of the iceLogo.
`background_universe`	this is a list cleavages you would like to compare to as background of the iceLogo

Value

a ggplot object that shows the motif of the cleavage sequences

Examples

```
# Determining cleavages of interest
sig_cleavage_data <- mspms::log2fc_t_test_data %>%
  dplyr::filter(p.adj <= 0.05, log2fc > 3)
# Plotting a iceLogo for each condition.
plot_all_icelogos(sig_cleavage_data)
```

plot_cleavages_per_pos *plot_cleavages_per_pos*

Description

plot the number of cleavages at each

Usage

```
plot_cleavages_per_pos(sig_cleavage_data, ncol = NULL)
```

Arguments

sig_cleavage_data	a tibble of data of interest containing a column labeled peptide, cleavage_seq, condition, and cleavage_pos.
ncol	the number of columns to plot.

Value

a ggplot2 object

Examples

```
# Defining the significant peptides
sig_cleavage_data <- log2fc_t_test_data %>%
  dplyr::filter(p.adj <= 0.05, log2fc > 3)
# Plotting
p1 <- mspms::plot_cleavages_per_pos(sig_cleavage_data)
p1
```

plot_heatmap *plot_heatmap*

Description

This produces a heatmaply interactive heatmap of the QFeatures object with color bars representing the condition and time for each sample in each row.

Usage

```
plot_heatmap(
  mspms_tidy_data,
  value_colname = "peptides_norm",
  scale = "column",
  plot_method = "plotly",
  show_dendrogram = c(TRUE, TRUE)
)
```

Arguments

mspms_tidy_data	tidy mspms data (prepared from QFeatures object by mspms_tidy())
value_colname	the name of the column containing values.
scale	how would you like the data scaled? default is none, but can also be "row", "column", or "none"
plot_method	what plot method would you like to use, can use plotly or ggplot2.
show_dendrogram	Logical vector of length two, controlling whether the row and/or column dendrograms are displayed. If a logical scalar is provided, it is repeated to become a logical vector of length two.

Details

Each column has a colored bar representing whether the peptide is a cleavage product or a full length member of the peptide library.

Value

a heatmaply interactive heatmap

Examples

```
plot_heatmap(mspms::mspms_tidy_data)
```

---

plot_icelogo *plot_icelogo*

Description

This function plots the cleavage motifs that were enriched relative to background as implemented in the iceLogo method. https://iomics.ugent.be/icelogoserver/resources/manual.pdf

Usage

```
plot_icelogo(
  cleavage_seqs,
  background_universe = mspms::all_possible_8mers_from_228_library,
  pval = 0.05,
  type = "percent_difference"
)
```

Arguments

cleavage_seqs	these are the cleavage sequences of interest
background_universe	this is a list of cleavage sequences to use as the background in building the iceLogo.
pval	this is the pvalue threshold (<=) to consider significant when determining the significance of the sig_cleavages relative to the background at each position of the iceLogo.
type	this is the type of visualization you would like to perform, accepted values are either "percent_difference" or "fold_change".

Value

a ggplot2 object

Examples

```
# Determining significant cleavages for catA
catA_sig_cleavages <- mspms::log2fc_t_test_data %>%
  dplyr::filter(p.adj <= 0.05, log2fc > 3) %>%
  dplyr::filter(condition == "CatA") %>%
  dplyr::pull(cleavage_seq) %>%
  unique()

# Plotting icelogo
plot_icelogo(catA_sig_cleavages,
  background_universe = all_possible_8mers_from_228_library
)
```

---

plot_nd_peptides *plot_nd_peptides*

Description

plot the percentage of samples each peptide from library was undetected in (if the percentage is > 0).

Usage

```
plot_nd_peptides(
  processed_qf,
  peptide_library_ids = mspms::peptide_library$library_id
)
```

Arguments

processed_qf	a QFeatures object containing a SummarizedExperiment named "peptides"
peptide_library_ids	a vector of all peptide library ids in the experiment.

Value

a ggplot2 object

Examples

```
plot_nd_peptides(mspms::processed_qf)
```

---

plot_pca *plot_pca*

Description

Easily create a PCA plot from a QFeatures object containing mspms data. Ellipses are drawn around the points at a 95 Shape and colors are user specified.

Usage

```
plot_pca(
  mspms_tidy_data,
  value_colname = "peptides_norm",
  color = "time",
  shape = "condition"
)
```

Arguments

mspms_tidy_data	tidy mspms data (prepared from QFeatures object by mspms_tidy)
value_colname	the name of the column containing values.
color	the name of the variable you would like to color by.
shape	the name of the variable that you would like to determine shape by.

Value

a ggplot2 object

Examples

```
plot_pca(mspms::mspms_tidy_data)
```

`plot_qc_check` *plot_qc_check plot the the percentage of the peptide library undetected in each sample per each sample group.*

Description

plot_qc_check plot the the percentage of the peptide library undetected in each sample per each sample group.

Usage

```
plot_qc_check(
  processed_qf,
  peptide_library = mspms::peptide_library$library_id,
  full_length_threshold = NULL,
  cleavage_product_threshold = NULL,
  ncol = 2
)
```

Arguments

`processed_qf`	QFeatures object containing a SummarizedExperiment named "peptides"
`peptide_library`	a vector of all peptide library ids in the experiment.
`full_length_threshold`	percent to use as threshold visualized as a vertical blue dashed line
`cleavage_product_threshold`	percent to use as a threshold visualized as a red dashed line
`ncol`	n columns.

Value

a ggplot2 object.

Examples

```
plot_qc_check(mspms::processed_qf)
```

plot_time_course *plot_time_course*

Description

Easily plot a time course of all peptides in a QFeatures object by peptide.

Usage

```
plot_time_course(
  mspms_tidy_data,
  value_colname = "peptides_norm",
  summarize_by_mean = FALSE
)
```

Arguments

mspms_tidy_data	tidy mspms data (prepared from QFeatures object by mspms_tidy())
value_colname	the name of the column containing values.
summarize_by_mean	whether to summarise by mean (TRUE- show error bars +- 1 standard deviation) or not (FALSE)

Value

a ggplot2 object

Examples

```
# Determining peptide of interest
max_log2fc_pep <- mspms::log2fc_t_test_data %>%
  dplyr::filter(p.adj <= 0.05, log2fc > 3) %>%
  dplyr::filter(log2fc == max(log2fc)) %>%
  dplyr::pull(peptide)

# Defining QFeatures filter
filtered <- mspms::mspms_tidy_data %>%
  dplyr::filter(peptide == max_log2fc_pep) %>%
  plot_time_course()
```

plot_volcano *plot_volcano*

Description

create a volcano plot to generate log2fc and adjusted p values for experimental conditions

Usage

```
plot_volcano(
  log2fc_t_test_data,
  log2fc_threshold = 3,
  padj_threshold = 0.05,
  facets = "grid",
  ncol = 1
)
```

Arguments

log2fc_t_test_data	a tibble containing the log2fc and adjusted p values
log2fc_threshold	the log2fc threshold that you want displayed on plot
padj_threshold	the padj threshold that you want displayed on plot
facets	how facets should be displayed. Accepted values are grid and wrap
ncol	ncol to include if facets = "wrap"

Value

a ggplot2 object

Examples

```
p1 <- mspms::plot_volcano(mspms::log2fc_t_test_data, log2fc_threshold = 3)
p1
```

prepared_to_qf *convert prepared data to a QFeatures object*

Description

convert prepared data to a QFeatures object

Usage

```
prepared_to_qf(
  prepared_data,
  colData,
  peptide_library = mspms::peptide_library,
  n_residues = 4
)
```

Arguments

prepared_data	data prepared within one of the prepare functions
colData	sample metadata
peptide_library	the peptide library used.
n_residues	the number of residues reported in the cleavage site

Value

a QFeatures object

`prepare_diann` *prepare_diann*

Description

prepare data from the pr_matrix.tsv diann output. This can be either from DIA-NN or from Fragpipe (as it uses DIA-NN for quantification internally for MSFragger-DIA workflows)

Usage

```
prepare_diann(
  precursor_filepath,
  colData_filepath,
  peptide_library = mspms::peptide_library,
  n_residues = 4
)
```

Arguments

precursor_filepath	filepath to report.pr_matrix.tsv file exported from DIA-NN.
colData_filepath	file path to .csv file containing colData. Must have columns named "quantCols","group","condition",and "time".

peptide_library
: peptide library used with experiment. Contains columns "library_id", "library_match_sequence", and "library_real_sequence".

n_residues
: the number of amino acid residues before and after the cleavage site to generate a cleavage seq for.

Value

a QFeatures object.

Examples

```
precursor_filepath <- system.file(
  "extdata/diann_report.pr_matrix.tsv",
  package = "mspms"
)
colData_filepath <- system.file("extdata/diann_colData.csv", package = "mspms")
prepare_diann(precursor_filepath, colData_filepath)
```

---

prepare_fc *prepare_fc*

Description

Prepare fold changes of amino acids by position for Icelogo visualization.

Usage

```
prepare_fc(fold_change, sig_zscores)
```

Arguments

fold_change
: a matrix of the fold changes of the AA by position.

sig_zscores
: a tibble of the significant zscores.

Value

a matrix of the fold changes of the significant AAs at each position.

prepare_file *Generic preparation function for MSP-MS input files*

Description

Generic preparation function for MSP-MS input files

Usage

```
prepare_file(
  filepath,
  colData_filepath,
  peptide_library = mspms::peptide_library,
  n_residues = 4,
  read_fun,
  validate_fun,
  transform_fun
)
```

Arguments

filepath	path to the input file
colData_filepath	path to colData CSV
peptide_library	peptide library used in experiment
n_residues	number of residues to include around cleavage site
read_fun	function to read the file (e.g., read_tsv, read_csv, read_delim)
validate_fun	function to validate the file (check_file_is_valid_*)
transform_fun	function to transform file into standard peptide format

Value

a QFeatures object

prepare_for_PCA *prepare_for_PCA()*

Description

prepare QFeatures object for PCA analysis

Usage

```
prepare_for_PCA(mspms_tidy_data, value_colname = "peptides_norm")
```

Arguments

mspms_tidy_data	tidy mspms data (prepared from QFeatures object by mspms_tidy())
value_colname	the name of the column containing values.

Value

a tibble

`prepare_fragpipe` *prepare_fragpipe*

Description

Prepare a label free quantification file exported from Fragpipe for subsequent mspms analysis.

Usage

```
prepare_fragpipe(
  combined_peptide_filepath,
  colData_filepath,
  peptide_library = mspms::peptide_library,
  n_residues = 4
)
```

Arguments

combined_peptide_filepath	file path the combined_peptide.tsv file generated by FragPipe.
colData_filepath	file path to .csv file containing colData. Must have columns named "quantCols","group","condition",and "time".
peptide_library	peptide library used with experiment. Contains columns "library_id", "library_match_sequence", and "library_real_sequence".
n_residues	the number of amino acid residues before and after the cleavage site to generate a cleavage seq for.

Value

a QFeatures object containing a summarizedExperiment named "peptides"

Examples

```
fragpipe_combined_peptide <- system.file("extdata/fragpipe_combined_peptide.tsv", package = "mspms")
colData_filepath <- system.file("extdata/colData.csv", package = "mspms")
# Prepare the data
fragpipe_prepared_data <- mspms::prepare_fragpipe(fragpipe_combined_peptide, colData_filepath)
```

prepare_icelogo_data *prepare_icelogo_data*

Description

Prepare the final matrix containing iceLogo data for plotting.

Usage

```
prepare_icelogo_data(
  cleavage_seqs,
  background_universe = mspms::all_possible_8mers_from_228_library,
  pval = 0.05,
  type = "percent_difference"
)
```

Arguments

cleavage_seqs	the cleavage sequences that are observed in the experiment
background_universe	a vector of the cleavage sequences to use as the background.
pval	the p-value threshold to consdier
type	the type of iceLogo calculation to perform. Accepted values are "percent_difference" or "fold_change".

Value

a matrix of enriched amino acids per position

---

prepare_pd *prepare_pd Prepare a label free quantification file exported from Proteome Discoverer for subsequent mspms analysis.*

Description

prepare_pd Prepare a label free quantification file exported from Proteome Discoverer for subsequent mspms analysis.

Usage

```
prepare_pd(
  peptide_groups_filepath,
  colData_filepath,
  peptide_library = mspms::peptide_library,
  n_residues = 4
)
```

Arguments

peptide_groups_filepath	filepath to PeptideGroups.txt file exported from proteome discoverer.
colData_filepath	file path to .csv file containing colData. Must have columns named "quantCols","group","condition",and "time".
peptide_library	peptide library used with experiment. Contains columns "library_id", "library_match_sequence", and "library_real_sequence".
n_residues	the number of amino acid residues before and after the cleavage site to generate a cleavage seq for.

Value

a QFeatures object containing a summarizedExperiment named "peptides"

Examples

```
peptide_groups_filepath <- system.file(
  "extdata/proteome_discoverer_PeptideGroups.txt",
  package = "mspms"
)
colData_filepath <- system.file("extdata/colData.csv", package = "mspms")
```

---

prepare_peaks *Prepare PEAKS label-free quantification data for MSP-MS analysis*

Description

This function reads, validates, transforms, and converts a PEAKS LFQ file into a ‘QFeatures‘ object compatible with the ‘mspms‘ workflow.

Usage

```
prepare_peaks(
  lfq_filepath,
  colData_filepath,
  quality_threshold = 0.3,
  peptide_library = mspms::peptide_library,
  n_residues = 4
)
```

Arguments

lfq_filepath	Path to the PEAKS '.csv' file containing peptide-level LFQ data.
colData_filepath	Path to a '.csv' file containing sample metadata ('colData'). Must include the columns '"quantCols"', '"group"', '"condition"', and '"time"'.
quality_threshold	Minimum quality score required for a peptide to be retained. Peptides below this threshold are filtered out (default '0.3').
peptide_library	A peptide library used in the experiment, typically 'mspms::peptide_library'. Must include '"library_id"', '"library_match_sequence"', and '"library_real_sequence"'.
n_residues	Number of amino acid residues to include on each side of the cleavage site when generating cleavage sequences (default '4').

Value

A 'QFeatures' object containing a 'SummarizedExperiment' named '"peptides"'.

Examples

```
lfq_filepath <- system.file(
  "extdata/peaks_protein-peptides-lfq.csv",
  package = "mspms"
)
colData_filepath <- system.file(
  "extdata/colData.csv",
  package = "mspms"
)
peaks_qf <- mspms::prepare_peaks(lfq_filepath, colData_filepath)
```

---

prepare_qc_check_data *prepare_qc_check Run simple quality control checks on the data. This checks to see how many peptides belonging to the library were identified in the data in each sample. Computes full length, and cleavage products independantly.*

Description

prepare_qc_check Run simple quality control checks on the data. This checks to see how many peptides belonging to the library were identified in the data in each sample. Computes full length, and cleavage products independantly.

Usage

```
prepare_qc_check_data(
  processed_qf,
  peptide_library_ids = mspms::peptide_library$library_id
)
```

Arguments

processed_qf	a QFeatures object with a SummarizedExperiment named "peptides". Intended to be prepared by one of the pre-processing prepare_x_data functions of the mspms R package.
peptide_library_ids	a character vector containing the names of the library_ids

Value

a tibble containing percentage of library_ids detected per sample, both as full length, and as cleavage products.

---

prepare_sage *prepare_sage Prepare a label free quantification file exported from Sage for subsequent mspms analysis.*

Description

prepare_sage Prepare a label free quantification file exported from Sage for subsequent mspms analysis.

Usage

```
prepare_sage(
  sage_lfq_filepath,
  colData_filepath,
  peptide_library = mspms::peptide_library,
  n_residues = 4
)
```

Arguments

sage_lfq_filepath	filepath to lfq.tsv file output from
colData_filepath	file path to .csv file containing colData. Must have columns named "quantCols","group","condition",and "time".
peptide_library	peptide library used with experiment. Contains columns "library_id", "library_match_sequence", and "library_real_sequence".
n_residues	the number of amino acid residues before and after the cleavage site to generate a cleavage seq for.

Value

a QFeatures object containing a summarizedExperiment named "peptides"

Examples

```
sage_lfq_filepath <- system.file(
  "extdata/sage_lfq.tsv",
  package = "mspms"
)
colData_filepath <- system.file("extdata/colData.csv", package = "mspms")

prepare_sage(sage_lfq_filepath, colData_filepath)
```

`prepare_sig_p_dif` *prepare_sig_p_dif*

Description

Prepare significant percent difference data frame for iceLogo

Usage

```
prepare_sig_p_dif(percent_difference, sig_zscores)
```

Arguments

`percent_difference`	a data frame containing the percent differences
`sig_zscores`	a matrix of significant amino acids at each position based on z-scores

Value

a tibble

`processed_qf` *processed_qf A QFeatures object prepared from PEAKS data of Cathepsin data that has been processed (imputation/normalization)*

Description

processed_qf A QFeatures object prepared from PEAKS data of Cathepsin data that has been processed (imputation/normalization)

Usage

```
processed_qf
```

Format

'peaks_prepared_data' An instance of class QFeatures containing 5 assays: [1] peptides: SummarizedExperiment with 2071 rows and 42 columns [2] peptides_log: SummarizedExperiment with 2071 rows and 42 columns [3] peptides_log_norm: SummarizedExperiment with 2071 rows and 42 columns [4] peptides_log_impute_norm: SummarizedExperiment with 2071 rows and 42 columns [5] peptides_norm: SummarizedExperiment with 2071 rows and 42 columns

peptides Peptide Sequence Detected ...

Source

<mspms processed data originally from PEAKS files found in "tests/testdata/protein-peptides-id.csv" and "tests/testdata/protein-peptides-lfq.csv">

`process_qf` *process_qf*

Description

process_qf

Usage

```
process_qf(prepared_qf)
```

Arguments

`prepared_qf`	this is a QFeatures object containing a SummarizedExperiment named "peptides"

Value

a QFeatures object containing a SummarizedExperiments named "peptides","peptides_log","peptides_log_norm", "peptides_log_impute_norm",and "peptides_norm"

Examples

```
processed_qf <- process_qf(mspms::peaks_prepared_data)
```

remaining_cd_names *remaining_cd_names*

Description

determine what the remaining colData names are when removing the reference variable.

Usage

```
remaining_cd_names(processed_qf, reference_variable)
```

Arguments

processed_qf	a QFeatures object
reference_variable	name of reference variable

Value

a vector of the remaining names in the colData

rlog2 *rlog2 Reverse log2 transformation*

Description

rlog2 Reverse log2 transformation

Usage

```
rlog2(x)
```

Arguments

x	a numeric value

Value

a reverse log2 transformed value

transform_diann *Transform DIA-NN report (pr_matrix.tsv) into standard peptide format*

Description

Transform DIA-NN report (pr_matrix.tsv) into standard peptide format

Usage

```
transform_diann(df, peptide_library)
```

Arguments

df	DIA-NN pr_matrix.tsv read with read_tsv
peptide_library	peptide library used in the experiment

Value

a tibble with columns: peptide, library_id, sample intensities

transform_fragpipe *Transform FragPipe combined_peptide.tsv into standard peptide format*

Description

Transform FragPipe combined_peptide.tsv into standard peptide format

Usage

```
transform_fragpipe(df, peptide_library)
```

Arguments

df	FragPipe combined_peptide.tsv read with read_tsv
peptide_library	peptide library used in the experiment

Value

a tibble with columns: peptide, library_id, sample intensities

`transform_pd` *Transform Proteome Discoverer PeptideGroups.txt into standard peptide format*

Description

Transform Proteome Discoverer PeptideGroups.txt into standard peptide format

Usage

```
transform_pd(df, peptide_library)
```

Arguments

`df`	Proteome Discoverer PeptideGroups.txt read with read_delim
`peptide_library`	peptide library used in the experiment

Value

a tibble with columns: peptide, library_id, sample intensities

`transform_peaks` *Transform PEAKS LFQ file into standard peptide format*

Description

Transform PEAKS LFQ file into standard peptide format

Usage

```
transform_peaks(df, peptide_library, quality_threshold = 0.3)
```

Arguments

`df`	PEAKS LFQ data read in with read_csv
`peptide_library`	peptide library used in the experiment
`quality_threshold`	minimum peptide quality to keep (default 0.3)

Value

a tibble with columns: peptide, library_id, sample intensities

`transform_sage` *Transform Sage lfq.tsv into standard peptide format*

Description

Transform Sage lfq.tsv into standard peptide format

Usage

```
transform_sage(df, peptide_library)
```

Arguments

`df`	sage lfq.tsv read with read_tsv
`peptide_library`	peptide library used in the experiment

Value

a tibble with columns: peptide, library_id, sample intensities

---

`%>%` *Pipe operator*

Description

See `magrittr::%>%` for details.

Usage

```
lhs %>% rhs
```

Arguments

`lhs`	A value or the magrittr placeholder.
`rhs`	A function call using the magrittr semantics.

Value

The result of calling ‘rhs(lhs)‘.

Index