Package: iq (via r-universe)

October 28, 2026

Type Package

Title Protein Quantification in Mass Spectrometry-Based Proteomics

Version 2.1.0

Maintainer Thang Pham <t.pham@amsterdamumc.nl>

Description An implementation of the MaxLFQ algorithm by Cox et al. (2014) <doi:10.1074/mcp.M113.031591> in a comprehensive pipeline for processing proteomics data in data-independent acquisition mode (Pham et al. 2020 <doi:10.1093/bioinformatics/btz961>; Pham et al. 2026 <doi:10.1021/acs.jproteome.5c01038>). It offers additional options for protein quantification using the N most intense fragment ions, using all fragment ions, the median polish algorithm by Tukey (1977, ISBN:0201076160), and a robust linear model. In general, the tool can be used to integrate multiple proportional observations into a single quantitative value.

Depends R (>= 2.10)

License BSD_3_clause + file LICENSE

LinkingTo Rcpp, RcppEigen

Encoding UTF-8

LazyData true

Suggests knitr, rmarkdown

VignetteBuilder knitr

URL https://github.com/tvpham/iq

BugReports https://github.com/tvpham/iq/issues

NeedsCompilation yes

Author Thang Pham [aut, cre, cph, ctb] (ORCID: <https://orcid.org/0000-0003-0333-2492>), Alex Henneman [ctb] (ORCID: <https://orcid.org/0000-0002-3746-4410>)

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

Date/Publication 2026-10-27 23:01:30 UTC

RemoteUrl https://github.com/cran/iq

RemoteRef HEAD

RemoteSha 22758813df05c2b62216b4127ca3accb03a283a2

Contents

connected_component . . . 2
create_protein_list . . . 3
create_protein_table . . . 4
extract_annotation . . . 5
fast_MaxLFQ . . . 6
fast_preprocess . . . 7
fast_read . . . 8
huber_weight . . . 10
long_format_to_iq_format . . . 11
maxLFQ . . . 12
meanInt . . . 13
median_polish . . . 14
plot_protein . . . 15
preprocess . . . 16
process_iq_format . . . 18
process_long_format . . . 19
process_matrix . . . 21
process_wide_format . . . 22
rescale . . . 23
spikeins . . . 24
topN . . . 25

Index **26**

`connected_component` *Group connected samples*

Description

Two samples are directly connected if they share at least one quantified fragment ion. Two samples in a connected component are either directly connected or indirectly via other samples.

Usage

```
connected_component(X)
```

Arguments

`X`	A matrix of intensities in the log2 space. Columns are samples and rows are fragment ions.

Value

A vector with length equal to the number of columns of the input containing membership of the connected components.

Author(s)

Thang V. Pham

References

Pham TV, Tran CTM, Henneman AA, Pham LHC, Le DG, Can AH, Bui PHL, Piersma SR, Jimenez CR. Boosting the speed and accuracy of protein quantification algorithms in mass spectrometry-based proteomics. *Journal of Proteome Research* 2026 Feb 6; 25(2):1198–1203.

`create_protein_list` *Creating a list of matrices of fragment ion intensities for all proteins*

Description

For each protein, a numerical matrix is formed where the columns are samples and rows are fragment ions.

Usage

```
create_protein_list(preprocessed_data)
```

Arguments

`preprocessed_data`
: A data frame of four components as output of the `preprocess` function.

Value

A list where each element contains the quantitative data of a protein. The column names are sample names and the row names fragment ions.

Author(s)

Thang V. Pham

References

Pham TV, Tran CTM, Henneman AA, Pham LHC, Le DG, Can AH, Bui PHL, Piersma SR, Jimenez CR. Boosting the speed and accuracy of protein quantification algorithms in mass spectrometry-based proteomics. *Journal of Proteome Research* 2026 Feb 6; 25(2):1198–1203.

Pham TV, Henneman AA, Jimenez CR. iq: an R package to estimate relative protein abundances from ion quantification in DIA-MS-based proteomics. *Bioinformatics* 2020 Apr 15;36(8):2611-2613.

See Also

preprocess

Examples

```
data("spikeins")
head(spikeins)
# This example set of spike-in proteins has been 'median-normalized'.
norm_data <- iq::preprocess(spikeins, median_normalization = FALSE, pdf_out = NULL)
protein_list <- iq::create_protein_list(norm_data)
```

---

create_protein_table *Protein quantification for a list of proteins*

Description

Travels through the input list and quantifies all proteins one by one.

Usage

```
create_protein_table(protein_list, method = "maxLFQ", ...)
```

Arguments

protein_list	The input protein list
method	Possible values are "maxLFQ", "median_polish", "topN", and "meanInt".
...	Additional parameters for individual quantitation methods.

Value

A list of two components is returned

estimate	A table of protein abundances for all samples in log2 space.
annotation	A vector of annotations, one for each protein.

Author(s)

Thang V. Pham

References

Pham TV, Tran CTM, Henneman AA, Pham LHC, Le DG, Can AH, Bui PHL, Piersma SR, Jimenez CR. Boosting the speed and accuracy of protein quantification algorithms in mass spectrometry-based proteomics. *Journal of Proteome Research* 2026 Feb 6; 25(2):1198–1203.

Pham TV, Henneman AA, Jimenez CR. iq: an R package to estimate relative protein abundances from ion quantification in DIA-MS-based proteomics. *Bioinformatics* 2020 Apr 15;36(8):2611-2613.

See Also

create_protein_list, maxLFQ, median_polish, topN, meanInt

Examples

```
data("spikeins")
# This example set of spike-in proteins has been 'median-normalized'.
norm_data <- iq::preprocess(spikeins, median_normalization = FALSE, pdf_out = NULL)
protein_list <- iq::create_protein_list(norm_data)
result <- iq::create_protein_table(protein_list)
head(result)
```

---

extract_annotation *Protein annotation extraction*

Description

Extracts annotation columns from a long-format input

Usage

```
extract_annotation(protein_ids, quant_table, primary_id = "PG.ProteinGroups",
                   annotation_columns = NULL)
```

Arguments

protein_ids	A vector of protein ids.
quant_table	A long-format input table. The input is typically the same as input to the preprocess function.
primary_id	The column containing protein ids.
annotation_columns	A vector of columns for annotation.

Value

A table of proteins and associated annotation extracted from the input.

Author(s)

Thang V. Pham

References

Pham TV, Tran CTM, Henneman AA, Pham LHC, Le DG, Can AH, Bui PHL, Piersma SR, Jimenez CR. Boosting the speed and accuracy of protein quantification algorithms in mass spectrometry-based proteomics. *Journal of Proteome Research* 2026 Feb 6; 25(2):1198–1203.

Pham TV, Henneman AA, Jimenez CR. iq: an R package to estimate relative protein abundances from ion quantification in DIA-MS-based proteomics. *Bioinformatics* 2020 Apr 15;36(8):2611-2613.

See Also

preprocess

Examples

```
data("spikeins")
extra_names <- iq::extract_annotation(levels(spikeins$PG.ProteinGroups),
                                      spikeins,
                                    annotation_columns = c("PG.Genes", "PG.ProteinNames"))
```

`fast_MaxLFQ` *The MaxLFQ algorithm*

Description

A fast implementation of the MaxLFQ algorithm.

Usage

```
fast_MaxLFQ(norm_data, row_names = NULL, col_names = NULL)
```

Arguments

`norm_data`	A list of four vectors with equal length `protein_list`, `sample_list`, `id` and `quant` as prepared by the `fast_preprocess` function or the `quant_table` component returned by the `fast_read` function. Note that `quant` should contain log2 intensities.
`row_names`	A vector of character strings for row names. If `NULL`, unique values in the `protein_list` component of `norm_data` will be used. Otherwise, it should be the first column of the `protein` component returned by the `fast_read`.
`col_names`	A vector of character strings for column names. If `NULL`, unique values in the `sample_list` component of `norm_data` will be used. Otherwise, it should be the `sample` component returned by the `fast_read`.

Value

A list is returned with two components

`estimate`	A quantification result table in log2 space.
`annotation`	A vector of strings indicating membership in case of multiple connected components for each row of `estimate`.

Author(s)

Thang V. Pham

References

Pham TV, Tran CTM, Henneman AA, Pham LHC, Le DG, Can AH, Bui PHL, Piersma SR, Jimenez CR. Boosting the speed and accuracy of protein quantification algorithms in mass spectrometry-based proteomics. *Journal of Proteome Research* 2026 Feb 6; 25(2):1198–1203.

Pham TV, Henneman AA, Jimenez CR. iq: an R package to estimate relative protein abundances from ion quantification in DIA-MS-based proteomics. *Bioinformatics* 2020 Apr 15;36(8):2611-2613.

See Also

`fast_read`, `fast_preprocess`

`fast_preprocess` *Data filtering and normalization*

Description

Filters out low intensities and performs median normalization.

Usage

```
fast_preprocess(quant_table,
                median_normalization = TRUE,
                log2_intensity_cutoff = 0,
                pdf_out = "qc-plots-fast.pdf",
                pdf_width = 12,
                pdf_height = 8,
                show_boxplot = TRUE)
```

Arguments

`quant_table`	The `quant_table` component as returned by `fast_read`.
`median_normalization`	A logical value. The default `TRUE` value is to perform median normalization.
`log2_intensity_cutoff`	Entries lower than this value in log2 space are ignored. Plot a histogram of all intensities to set this parameter.
`pdf_out`	A character string specifying the name of the PDF output. A `NULL` value will suppress the PDF output.
`pdf_width`	Width of the pdf output in inches.
`pdf_height`	Height of the pdf output in inches.
`show_boxplot`	A logical value. The default `TRUE` value is to create boxplots of fragment intensities for each sample.

Value

A list is returned with the same components as input data in which low intensities are filtered out and median normalization is performed if requested.

Author(s)

Thang V. Pham

References

Pham TV, Tran CTM, Henneman AA, Pham LHC, Le DG, Can AH, Bui PHL, Piersma SR, Jimenez CR. Boosting the speed and accuracy of protein quantification algorithms in mass spectrometry-based proteomics. *Journal of Proteome Research* 2026 Feb 6; 25(2):1198–1203.

Pham TV, Henneman AA, Jimenez CR. iq: an R package to estimate relative protein abundances from ion quantification in DIA-MS-based proteomics. *Bioinformatics* 2020 Apr 15;36(8):2611-2613.

See Also

fast_read

`fast_read` *Reading data from an input file*

Description

A highly efficient reading of a tab-separated text file for iq processing.

Usage

```
fast_read(filename,
          sample_id = "R.Condition",
          primary_id = "PG.ProteinGroups",
       secondary_id = c("EG.ModifiedSequence", "FG.Charge", "F.FrgIon", "F.Charge"),
          intensity_col = "F.PeakArea",
          annotation_col = c("PG.Genes", "PG.ProteinNames"),
          filter_string_equal = c("F.ExcludedFromQuantification" = "False"),
          filter_string_not_equal = NULL,
          filter_double_less = c("PG.Qvalue" = "0.01", "EG.Qvalue" = "0.01"),
          filter_double_greater = NULL,
          intensity_col_sep = NULL,
          intensity_col_id = NULL,
          na_string = "0")
```

Arguments

filename	A long-format tab-separated text file with a primary column of protein identification, secondary columns of fragment ions, a column of sample names, a column for quantitative intensities, and extra columns for annotation.
primary_id	Unique values in this column form the list of proteins to be quantified.
secondary_id	A concatenation of these columns determines the fragment ions used for quantification.
sample_id	Unique values in this column form the list of samples.
intensity_col	The column for intensities.
annotation_col	Annotation columns
filter_string_equal	A named vector of strings. Only rows satisfying the condition are kept.
filter_string_not_equal	A named vector of strings. Only rows satisfying the condition are kept.
filter_double_less	A named vector of strings. Only rows satisfying the condition are kept. Default PG.Qvalue < 0.01 and EG.Qvalue < 0.01.
filter_double_greater	A named vector of strings. Only rows satisfying the condition are kept.
intensity_col_sep	A separator character when entries in the intensity column contain multiple values.
intensity_col_id	The column for identities of multiple quantitative values.
na_string	The value considered as NA.

Details

When entries in the intensity column contain multiple values, this function will replicate entries in other column and the `secondary_id` will be appended with corresponding entries in `intensity_col_id` when it is provided. Otherwise, integer values 1, 2, 3, etc... will be used.

Value

A list is returned with following components

protein	A table of proteins in the first column followed by annotation columns.
sample	A vector of samples.
ion	A vector of fragment ions to be used for quantification.
quant_table	A list of four components: protein_list (index pointing to `protein`)), sample_list (index pointing to `sample`), id (index pointing to `ion`), and quant (intensities).

Author(s)

Thang V. Pham

References

Pham TV, Tran CTM, Henneman AA, Pham LHC, Le DG, Can AH, Bui PHL, Piersma SR, Jimenez CR. Boosting the speed and accuracy of protein quantification algorithms in mass spectrometry-based proteomics. *Journal of Proteome Research* 2026 Feb 6; 25(2):1198–1203.

Pham TV, Henneman AA, Jimenez CR. iq: an R package to estimate relative protein abundances from ion quantification in DIA-MS-based proteomics. *Bioinformatics* 2020 Apr 15;36(8):2611-2613.

`huber_weight` *Calculate Huber weights for an individual matrix*

Description

Calculate Huber weights for an individual matrix

Usage

```
huber_weight(X, x, c = 1.4826, k = 1.345)
```

Arguments

`X`	A matrix of intensities in the log2 space. Columns are samples and rows are fragment ions.
`x`	An initial estimate
`c`	Weighting function parameter.
`k`	Weighting function parameter.

Value

A numeric weight vector of size equal to the number of rows of X.

Author(s)

Thang V. Pham

References

Pham TV, Tran CTM, Henneman AA, Pham LHC, Le DG, Can AH, Bui PHL, Piersma SR, Jimenez CR. Boosting the speed and accuracy of protein quantification algorithms in mass spectrometry-based proteomics. *Journal of Proteome Research* 2026 Feb 6; 25(2):1198–1203.

long_format_to_iq_format *Long format input data to iq format*

Description

This function reads long-format input data, filters it, and writes the results to the iq format.

Usage

```
long_format_to_iq_format(input_data,
                         output_data,
                         sample_id = "Run",
                         primary_id = "Protein.Group",
                         secondary_id = c("Precursor.Id"),
                         intensity_col = "Intensities",
                         annotation_col = NULL,
                         filter_string_equal = NULL,
                         filter_string_not_equal = NULL,
                         filter_double_less = NULL,
                         filter_double_greater = NULL,
                         intensity_col_sep = ";",
                         intensity_col_id = NULL,
                         na_string = "0",
                         normalization = "median",
                         log2_intensity_cutoff = 0,
                         pdf_out = "qc-plots-iq.pdf",
                         pdf_width = 12,
                         pdf_height = 8,
                         show_boxplot = FALSE)
```

Arguments

input_data	See filename in fast_read.
output_data	Output data.
sample_id	See sample_id in fast_read.
primary_id	See primary_id in fast_read.
secondary_id	See secondary_id in fast_read.
intensity_col	See intensity_col in fast_read.
annotation_col	See annotation_col in fast_read.
filter_string_equal	See filter_string_equal in fast_read.
filter_string_not_equal	See filter_string_not_equal in fast_read.

filter_double_less
: See filter_double_less in fast_read.

filter_double_greater
: See filter_double_greater in fast_read.

intensity_col_sep
: See intensity_col_sep in fast_read.

intensity_col_id
: See intensity_col_id in fast_read.

na_string
: See intensity_col_id in fast_read.

normalization
: Normalization type. Possible values are median and none. The default value median is for median normalization in fast_preprocess.

log2_intensity_cutoff
: See log2_intensity_cutoff in fast_preprocess.

pdf_out
: See pdf_out in fast_preprocess.

pdf_width
: See pdf_width in fast_preprocess.

pdf_height
: See pdf_height in fast_preprocess.

show_boxplot
: See show_boxplot in fast_preprocess.

Value

The output is written to a new directory named output_data.

Author(s)

Thang V. Pham

References

Pham TV, Tran CTM, Henneman AA, Pham LHC, Le DG, Can AH, Bui PHL, Piersma SR, Jimenez CR. Boosting the speed and accuracy of protein quantification algorithms in mass spectrometry-based proteomics. *Journal of Proteome Research* 2026 Feb 6; 25(2):1198–1203.

See Also

fast_read, fast_preprocess

---

maxLFQ *The MaxLFQ algorithm for protein quantification*

Description

Estimates protein abundances by aiming to maintain the fragment intensity ratios between samples.

Usage

```
maxLFQ(X)
```

Arguments

X	A matrix of ion intensities in log2 space. Columns are samples and rows are fragment ions.

Value

A list of two components is returned

estimate	A vector with length equal to the number of columns of the input containing the protein abundances in log2 space.
annotation	An empty string if all quantified samples are connected. Otherwise, a string of membership of the connected components is returned.

Author(s)

Thang V. Pham

References

Pham TV, Tran CTM, Henneman AA, Pham LHC, Le DG, Can AH, Bui PHL, Piersma SR, Jimenez CR. Boosting the speed and accuracy of protein quantification algorithms in mass spectrometry-based proteomics. *Journal of Proteome Research* 2026 Feb 6; 25(2):1198–1203.

Cox J, Hein MY, Luber CA, et al. Accurate proteome-wide label-free quantification by delayed normalization and maximal peptide ratio extraction, termed MaxLFQ. *Mol Cell Proteomics*. 2014;13(9):2513–2526.

Pham TV, Henneman AA, Jimenez CR. iq: an R package to estimate relative protein abundances from ion quantification in DIA-MS-based proteomics. *Bioinformatics* 2020 Apr 15;36(8):2611-2613.

---

meanInt *The meanInt algorithm for protein quantification*

Description

Estimates protein abundances by averaging all associated ion intensities

Usage

```
meanInt(X, aggregation_in_log_space = TRUE)
```

Arguments

X	A matrix of ion intensities in log2 space. Columns are samples and rows are fragment ions.
aggregation_in_log_space	A logical value. If FALSE, the data aggregation is performed in the original intensity space.

Value

A list of two components is returned

`estimate`	A vector with length equal to the number of columns of the input containing the protein abundances in log2 space.
`annotation`	Reserved, currently an empty string.

Author(s)

Thang V. Pham

References

Pham TV, Tran CTM, Henneman AA, Pham LHC, Le DG, Can AH, Bui PHL, Piersma SR, Jimenez CR. Boosting the speed and accuracy of protein quantification algorithms in mass spectrometry-based proteomics. *Journal of Proteome Research* 2026 Feb 6; 25(2):1198–1203.

Pham TV, Henneman AA, Jimenez CR. iq: an R package to estimate relative protein abundances from ion quantification in DIA-MS-based proteomics. *Bioinformatics* 2020 Apr 15;36(8):2611-2613.

---

median_polish *A wrapper for the R implementation of the median polish algorithm*

Description

Estimates protein abundances using the Tukey median polish algorithm.

Usage

```
median_polish(X)
```

Arguments

`X`	A matrix of ion intensities in log2 space. Columns are samples and rows are fragment ions.

Value

A list of two components is returned

`estimate`	A vector with length equal to the number of columns of the input containing the protein abundances in log2 space.
`annotation`	Reserved, currently an empty string

Author(s)

Thang V. Pham

References

Pham TV, Tran CTM, Henneman AA, Pham LHC, Le DG, Can AH, Bui PHL, Piersma SR, Jimenez CR. Boosting the speed and accuracy of protein quantification algorithms in mass spectrometry-based proteomics. *Journal of Proteome Research* 2026 Feb 6; 25(2):1198–1203.

Pham TV, Henneman AA, Jimenez CR. iq: an R package to estimate relative protein abundances from ion quantification in DIA-MS-based proteomics. *Bioinformatics* 2020 Apr 15;36(8):2611-2613.

Tukey JW. *Exploratory Data Analysis*, Reading Massachusetts: Addison-Wesley, 1977.

`plot_protein` *Plotting the underlying quantitative data for a protein*

Description

Displays the underlying data for a protein.

Usage

```
plot_protein(X, main = "", col = NULL, split = 0.6, ...)
```

Arguments

`X`	Protein data matrix.
`main`	Title of the plot.
`col`	Colors of the rows of the data matrix.
`split`	Fraction of the plotting area for the main figure. The remaining one is for legend. Set this parameter to `NULL` to ignore the legend area.
`...`	Additional parameters for plotting.

Value

A `NULL` value is returned.

Author(s)

Thang V. Pham

References

Pham TV, Tran CTM, Henneman AA, Pham LHC, Le DG, Can AH, Bui PHL, Piersma SR, Jimenez CR. Boosting the speed and accuracy of protein quantification algorithms in mass spectrometry-based proteomics. *Journal of Proteome Research* 2026 Feb 6; 25(2):1198–1203.

Pham TV, Henneman AA, Jimenez CR. iq: an R package to estimate relative protein abundances from ion quantification in DIA-MS-based proteomics. *Bioinformatics* 2020 Apr 15;36(8):2611-2613.

Examples

```
data("spikeins")
head(spikeins)
# This example set of spike-in proteins has been 'median-normalized'.
norm_data <- iq::preprocess(spikeins, median_normalization = FALSE, pdf_out = NULL)
protein_list <- iq::create_protein_list(norm_data)
iq::plot_protein(protein_list$P00366, main = "Protein P00366", split = NULL)
```

---

preprocess *Data preprocessing for protein quantification*

Description

Prepares a long-format input including removing low-intensity ions and performing median normalization.

Usage

```
preprocess(quant_table,
           primary_id = "PG.ProteinGroups",
       secondary_id = c("EG.ModifiedSequence", "FG.Charge", "F.FrgIon", "F.Charge"),
           sample_id = "R.Condition",
           intensity_col = "F.PeakArea",
           median_normalization = TRUE,
           log2_intensity_cutoff = 0,
           pdf_out = "qc-plots.pdf",
           pdf_width = 12,
           pdf_height = 8,
           intensity_col_sep = NULL,
           intensity_col_id = NULL,
           na_string = "0",
           show_boxplot = TRUE)
```

Arguments

quant_table	A long-format table with a primary column of protein identification, secondary columns of fragment ions, a column of sample names, and a column for quantitative intensities.
primary_id	Unique values in this column form the list of proteins to be quantified.
secondary_id	A concatenation of these columns determines the fragment ions used for quantification.
sample_id	Unique values in this column form the list of samples.
intensity_col	The column for intensities.
median_normalization	A logical value. The default TRUE value is to perform median normalization.

`log2_intensity_cutoff`
: Entries lower than this value in log2 space are ignored. Plot a histogram of all intensities to set this parameter.

`pdf_out`
: A character string specifying the name of the PDF output. A `NULL` value will suppress the PDF output.

`pdf_width`
: Width of the pdf output in inches.

`pdf_height`
: Height of the pdf output in inches.

`intensity_col_sep`
: A separator character when entries in the intensity column contain multiple values.

`intensity_col_id`
: The column for identities of multiple quantitative values.

`na_string`
: The value considered as NA.

`show_boxplot`
: A logical value. The default `TRUE` value is to create boxplots of fragment intensities for each sample.

Details

When entries in the intensity column contain multiple values, this function will replicate entries in other column and the `secondary_id` will be appended with corresponding entries in `intensity_col_id` when it is provided. Otherwise, integer values 1, 2, 3, etc... will be used.

Value

A data frame is returned with following components

`protein_list`
: A vector of proteins.

`sample_list`
: A vector of samples.

`id`
: A vector of fragment ions to be used for quantification.

`quant`
: A vector of log2 intensities.

Author(s)

Thang V. Pham

References

Pham TV, Tran CTM, Henneman AA, Pham LHC, Le DG, Can AH, Bui PHL, Piersma SR, Jimenez CR. Boosting the speed and accuracy of protein quantification algorithms in mass spectrometry-based proteomics. *Journal of Proteome Research* 2026 Feb 6; 25(2):1198–1203.

Pham TV, Henneman AA, Jimenez CR. iq: an R package to estimate relative protein abundances from ion quantification in DIA-MS-based proteomics. *Bioinformatics* 2020 Apr 15;36(8):2611-2613.

Examples

```
data("spikeins")
head(spikeins)
# This example set of spike-in proteins has been 'median-normalized'.
norm_data <- iq::preprocess(spikeins, median_normalization = FALSE, pdf_out = NULL)
```

`process_iq_format` *Process data in the iq format*

Description

Estimates protein abundance of all proteins in the iq format data.

Usage

```
process_iq_format(input_data,
                  output_filename = NULL,
                  method = "maxlfq_bit",
                  p1 = NULL, p2 = NULL, k = 1.345, min_M = 15, n_threads = -1,
                  rescale_method = NULL)
```

Arguments

`input_data`	Input data in the iq format.
`output_filename`	Output file name. If `NULL`, a file name derived from `input_data` will be used.
`method`	See `method` in `process_matrix`.
`p1`	See `p1` in `process_matrix`.
`p2`	See `p2` in `process_matrix`.
`k`	See `k` in `process_matrix`.
`min_M`	See `min_M` in `process_matrix`.
`n_threads`	See `n_threads` in `process_matrix`.
`rescale_method`	Rescale the output intensities. See `method` in `rescale`.

Value

A tab-separated table with rows representing proteins and columns representing protein annotations and samples.

Author(s)

Thang V. Pham

References

Pham TV, Tran CTM, Henneman AA, Pham LHC, Le DG, Can AH, Bui PHL, Piersma SR, Jimenez CR. Boosting the speed and accuracy of protein quantification algorithms in mass spectrometry-based proteomics. *Journal of Proteome Research* 2026 Feb 6; 25(2):1198–1203.

`process_long_format` *Long format to a wide format table using the MaxLFQ algorithm*

Description

A convenient function combining multiple steps to process a long format table using the MaxLFQ algorithm.

Usage

```
process_long_format(input_data,
                    output_filename,
                    sample_id = "File.Name",
                    primary_id = "Protein.Group",
                    secondary_id = "Precursor.Id",
                    intensity_col = "Fragment.Quant.Corrected",
                    annotation_col = NULL,
                    filter_string_equal = NULL,
                    filter_string_not_equal = NULL,
                  filter_double_less = c("Q.Value" = "0.01", "PG.Q.Value" = "0.01"),
                    filter_double_greater = NULL,
                    intensity_col_sep = ";",
                    intensity_col_id = NULL,
                    na_string = "0",
                    normalization = "median",
                    log2_intensity_cutoff = 0,
                    pdf_out = "qc-plots.pdf",
                    pdf_width = 12,
                    pdf_height = 8,
                    show_boxplot = TRUE,
                    peptide_extractor = NULL,
                    rfasta = NULL)
```

Arguments

`input_data`	A data frame or a filename. See `filename` in `fast_read`.
`output_filename`	Output filename.
`sample_id`	See `sample_id` in `fast_read`.
`primary_id`	See `primary_id` in `fast_read`.
`secondary_id`	See `secondary_id` in `fast_read`.

intensity_col See intensity_col in fast_read.

annotation_col See annotation_col in fast_read.

filter_string_equal
: See filter_string_equal in fast_read.

filter_string_not_equal
: See filter_string_not_equal in fast_read.

filter_double_less
: See filter_double_less in fast_read.

filter_double_greater
: See filter_double_greater in fast_read.

intensity_col_sep
: See intensity_col_sep in fast_read.

intensity_col_id
: See intensity_col_id in fast_read.

na_string See intensity_col_id in fast_read.

normalization Normalization type. Possible values are median and none. The default value median is for median normalization in fast_preprocess.

log2_intensity_cutoff
: See log2_intensity_cutoff in fast_preprocess.

pdf_out See pdf_out in fast_preprocess.

pdf_width See pdf_width in fast_preprocess.

pdf_height See pdf_height in fast_preprocess.

show_boxplot See show_boxplot in fast_preprocess.

peptide_extractor
: A function to parse peptides

rfasta If available, calculates the sequence coverage.

Value

Either an input data frame is processed with fast_MaxLFQ or an input file is processed with fast_read, fast_preprocess, and fast_MaxLFQ. Subsequently, the result is written to output_filename. The quantification values are in log2 space. A NULL value is returned. If peptide_extractor is not NULL, fragment statistics for each protein will be calculated based on the result of the extractor function. Counting the number of peptides contributing to a protein is possible using an appropriate extractor function. An example value for peptide_extractor is function(x) gsub("[0-9].*$", "", x), which removes the charge state and fragment descriptors in an ion descriptor to obtain unique peptide sequences. One can examine the ion component returned by the fast_read function to derive a regular expression to be used in the gsub function above. Another example is function(x) gsub("\(UniMod:\d+\)|\d|_", "", x) which removes annotations for modifications on the peptides.

Author(s)

Thang V. Pham

References

Pham TV, Tran CTM, Henneman AA, Pham LHC, Le DG, Can AH, Bui PHL, Piersma SR, Jimenez CR. Boosting the speed and accuracy of protein quantification algorithms in mass spectrometry-based proteomics. *Journal of Proteome Research* 2026 Feb 6; 25(2):1198–1203.

Pham TV, Henneman AA, Jimenez CR. iq: an R package to estimate relative protein abundances from ion quantification in DIA-MS-based proteomics. *Bioinformatics* 2020 Apr 15;36(8):2611-2613.

See Also

`fast_read`, `fast_preprocess`, `fast_MaxLFQ`

`process_matrix` *Process individual matrix*

Description

Estimates a row vector best representing all rows in the input numeric matrix.

Usage

```
process_matrix(X,
               method,
               p1 = NULL, p2 = NULL, k = 1.345, min_M = 15, n_threads = -1)
```

Arguments

`X`	A matrix of intensities in the log2 space. Columns are samples and rows are fragment ions.
`method`	One of the quantification methods: `"maxlfq"`, `"maxlfq_bit"`, `"weighted_maxlfq"`, `"median_polish"`, `"weighted_median_polish"`, `"rlm"`, or `"weighted_rlm"`.
`p1`	Method parameter. For `"rlm"` and `"weighted_rlm"`, it is the robust parameter. For `"median_polish"` and `"weighted_median_polish"`, it is the convergence parameter. Unused in other cases.
`p2`	Method parameter. For `"maxlfq_bit"`, it specifies the memory level. For others, it is the maximum number of optimization rounds.
`k`	Weighting function parameter.
`min_M`	Minimum number of rows for the weighted methods.
`n_threads`	The number of threads to be used. When `n_threads` is 0, the maximal number of CPU cores is used. When `n_threads` is -1 (default), one CPU core less than the maximum is used, and so on.

Value

A numeric row vector.

Author(s)

Thang V. Pham

References

Pham TV, Tran CTM, Henneman AA, Pham LHC, Le DG, Can AH, Bui PHL, Piersma SR, Jimenez CR. Boosting the speed and accuracy of protein quantification algorithms in mass spectrometry-based proteomics. *Journal of Proteome Research* 2026 Feb 6; 25(2):1198–1203.

`process_wide_format` *Merging rows with identical values in a particular column in a table*

Description

Collapses rows with identical values in a particular column in a table. When the values in each row are proportional such as intensities of multiple fragments of a protein, the MaxLFQ algorithm is recommended.

Usage

```
process_wide_format(input_filename,
                    output_filename,
                    id_column,
                    quant_columns,
                    data_in_log_space = FALSE,
                    annotation_columns = NULL,
                    method = "maxLFQ")
```

Arguments

`input_filename`	Input filename of a tab-separated value text file.
`output_filename`	Output filename.
`id_column`	The column where unique values will be kept. Rows with identical values in this column are merged. Rows with empty values here are removed.
`quant_columns`	Columns containing numerical data to be merged.
`data_in_log_space`	A logical value. If `FALSE`, the numerical data will be log2-transformed.
`annotation_columns`	Columns in the input file apart from `id_column` and `quant_columns` that will be kept in the output.
`method`	Method for merging. Default value is `"maxLFQ"`. Possible values are `"maxLFQ"`, `"maxLFQ_R"`, `"median_polish"`, `"top3"`, `"top5"`, `"meanInt"`, `"maxInt"`, `"sum"`, `"least_na"` and any function for collapsing a numerical matrix to a row vector.

Details

Method "maxLFQ_R" implements the MaxLFQ algorithm pure R. It is slower than "maxLFQ".

Method "maxInt" selects row with maximum intensity (top 1).

Method "sum" sum all intensities.

Method "least_na" selects row with the least number of missing values.

The value of method can be a function such as function(x) log2(colSums(2^x, na.rm = TRUE)) for summing all intensities in the original space.

Value

The result table is written to output_filename. A NULL value is returned.

Author(s)

Thang V. Pham

References

Pham TV, Tran CTM, Henneman AA, Pham LHC, Le DG, Can AH, Bui PHL, Piersma SR, Jimenez CR. Boosting the speed and accuracy of protein quantification algorithms in mass spectrometry-based proteomics. *Journal of Proteome Research* 2026 Feb 6; 25(2):1198–1203.

Pham TV, Henneman AA, Jimenez CR. iq: an R package to estimate relative protein abundances from ion quantification in DIA-MS-based proteomics. *Bioinformatics* 2020 Apr 15;36(8):2611-2613.

---

rescale *Rescale relative quantification*

Description

Each connected component of the input values can be rescaled to match the level of the overall input intensities.

Usage

```
rescale(x, X, method)
```

Arguments

x	A row vector containing the quantification result.
X	An input matrix of log2-transformed intensities. Columns represent samples, and rows represent fragment ions.
method	One of the methods for rescaling: "median-mean", "median-median", "mean-mean" or "sum".

Value

A row vector of rescaled values of the same size as `x`, with each connected component rescaled independently. For the `"median-mean"` method, single-sample components are rescaled to the median of input data, while all other components are rescaled to the mean. For other methods, each component is rescaled to the median, mean, or sum of the input data.

Author(s)

Thang V. Pham

References

Pham TV, Tran CTM, Henneman AA, Pham LHC, Le DG, Can AH, Bui PHL, Piersma SR, Jimenez CR. Boosting the speed and accuracy of protein quantification algorithms in mass spectrometry-based proteomics. *Journal of Proteome Research* 2026 Feb 6; 25(2):1198–1203.

---

spikeins *An example dataset of 12 spike-in proteins*

Description

A subset of the Bruderer 2015 dataset containing 12 spike-in proteins. The full dataset was exported from the Spectronaut software. The complete dataset has been median-normalized.

Usage

```
data("spikeins")
```

Format

A data frame with 18189 observations on the following 9 variables.

`R.Condition` Sample names.

`PG.ProteinGroups` Protein identifiers.

`EG.ModifiedSequence` Sequence of the fragment ions.

`FG.Charge` Fragment group charge.

`F.FrgIon` Fragment ions.

`F.Charge` Fragment charges.

`F.PeakArea` Quantitative values.

`PG.Genes` Gene names.

`PG.ProteinNames` Protein names.

Examples

```
data("spikeins")
head(spikeins)
```

topN *The topN algorithm for protein quantification*

Description

Estimates protein abundances using the N most intense ions.

Usage

```
topN(X, N = 3, aggregation_in_log_space = TRUE)
```

Arguments

X	A matrix of ion intensities in log2 space. Columns are samples and rows are fragment ions.
N	The number of top ions used for quantification.
aggregation_in_log_space	A logical value. If FALSE, data aggregation is performed in the original intensity space.

Value

A list of two components is returned

estimate	A vector with length equal to the number of columns of the input containing the protein abundances in log2 space.
annotation	Reserved, currently an empty string.

Author(s)

Thang V. Pham

References

Pham TV, Tran CTM, Henneman AA, Pham LHC, Le DG, Can AH, Bui PHL, Piersma SR, Jimenez CR. Boosting the speed and accuracy of protein quantification algorithms in mass spectrometry-based proteomics. *Journal of Proteome Research* 2026 Feb 6; 25(2):1198–1203.

Pham TV, Henneman AA, Jimenez CR. iq: an R package to estimate relative protein abundances from ion quantification in DIA-MS-based proteomics. *Bioinformatics* 2020 Apr 15;36(8):2611-2613.

Index

∗ **datasets**
spikeins, 24

connected_component, 2
create_protein_list, 3, *5*
create_protein_table, 4

extract_annotation, 5

fast_MaxLFQ, 6, *20, 21*
fast_preprocess, 7, *7*, *12*, *20, 21*
fast_read, *7*, 8, *8*, *11, 12*, *19–21*

huber_weight, 10

long_format_to_iq_format, 11

maxLFQ, *5*, 12
meanInt, *5*, 13
median_polish, *5*, 14

plot_protein, 15
preprocess, *4*, *6*, 16
process_iq_format, 18
process_long_format, 19
process_matrix, *18*, 21
process_wide_format, 22

rescale, *18*, 23

spikeins, 24

topN, *5*, 25