Package 'iq'

November 6, 2026

Type Package

Title Protein Quantification in Mass Spectrometry-Based Proteomics

Version 2.2.0

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

Description An implementation of the MaxLFQ algorithm by
Cox et al. (2014) <doi:10.1074/mcp.M113.031591> using least-
squares analysis in a comprehensive
pipeline for processing proteomics data in data-independent acquisition mode
(Pham et al. 2020 <doi:10.1093/bioinformatics/btz961>, and a new algorithm using the conjugate gradient method
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),
a robust linear model, and fragment weighting. A new iq format enables the processing of large-scale datasets.
In general, the tool can be used to integrate
multiple proportional observations into a single quantitative value.

Depends R (>= 4.0.0)

License BSD_3_clause + file LICENSE

Encoding UTF-8

LazyData true

Suggests knitr, rmarkdown, arrow

VignetteBuilder knitr

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

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

NeedsCompilation yes

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

Repository CRAN

Date/Publication 2026-11-06 12:00:02 UTC

Contents

connected_component . . . 2
create_protein_list . . . 3
create_protein_table . . . 4
extract_annotation . . . 5
fast_MaxLFQ . . . 6
fast_preprocess . . . 7
fast_read . . . 9
fast_read_data_frame . . . 11
fast_read_parquet . . . 12
huber_weight . . . 14
iq_format_to_protein_list . . . 15
long_format_to_iq_format . . . 16
maxLFQ . . . 18
meanInt . . . 19
median_polish . . . 20
plot_protein . . . 21
preprocess . . . 22
process . . . 24
process_iq_format . . . 28
process_long_format . . . 29
process_matrix . . . 32
process_wide_format . . . 34
read_fasta_file . . . 35
rescale . . . 36
spikeins . . . 37
topN . . . 38

Index **39**

`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, or a data frame with numeric columns, 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.

For an empty protein_list, estimate is a matrix without rows and annotation is empty.

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, verbose = TRUE)
```

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`, and `protein_list` must contain indices from 1 to its length.
`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`, and `sample_list` must contain indices from 1 to its length.
`verbose`	A logical value. The default `TRUE` prints progress and summary messages; `FALSE` prints none of them. Warnings and errors are always shown.

Details

The MaxLFQ problem of each protein is solved with the `"maxlfq_bit"` method of `process_matrix`.

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,
                boxplot_max_points = 10000,
                verbose = 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 not above this value in log2 space are ignored; NULL for no cutoff. Plot a histogram of all intensities to set this parameter. Zero and negative intensities (not finite in log2 space) are always removed.
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.
boxplot_max_points	The maximum number of values per sample in the boxplots (evenly spaced values are used for a larger sample). The median normalization always uses all values. NULL for all values.
verbose	A logical value. The default TRUE prints progress and summary messages; FALSE prints none of them. Warnings and errors are always shown.

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",
          verbose = TRUE)
```

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	One or more columns. 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, or several columns, one per fragment ion (e.g. `paste0("Fr.", 0:11, ".Quantity")` in DIA-NN 2).
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. Rows with a value less than the given value, or a missing value (empty, NaN or not a number), are kept. Default PG.Qvalue < 0.01 and EG.Qvalue < 0.01.

`filter_double_greater`
: A named vector of strings. Rows with a value greater than the given value, or a missing value (empty, NaN or not a number), are kept.

`intensity_col_sep`
: A separator character when entries in a single intensity column contain multiple values. It cannot be used with several intensity columns.

`intensity_col_id`
: Identities of multiple quantitative values: a single column split by `intensity_col_sep`, or one column per intensity column.

`na_string`
: The value considered as NA, in addition to empty values and NaN. It is used with `intensity_col_sep` or several intensity columns.

`verbose`
: A logical value. The default `TRUE` prints progress and summary messages; `FALSE` prints none of them. Warnings and errors are always shown.

Details

When there are multiple quantitative values per row (a single intensity column split by `intensity_col_sep`, or several intensity columns), this function will replicate entries in other columns 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: the `secondary_id` columns and the suffix of the intensity value (`1`, `2`, ... or its `intensity_col_id`), joined with `_`.

`ion_prefix`
: For each ion, its `secondary_id` part, without the suffix of the intensity value (used for `peptide_extractor` in `process`).

`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).

If the file cannot be read (e.g. a missing file or column) or no values are left after filtering, `fast_read` stops with an error.

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.

`fast_read_data_frame` *Reading data from a data frame*

Description

Reads a long-format data frame for iq processing, with the same filtering rules as `fast_read` and `fast_read_parquet`.

Usage

```
fast_read_data_frame(input_data,
                     sample_id = "Run",
                     primary_id = "Protein.Group",
                     secondary_id = "Precursor.Id",
                     intensity_col = "Precursor.Normalised",
                     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 = NULL,
                     intensity_col_id = NULL,
                     na_string = "0",
                     verbose = TRUE)
```

Arguments

`input_data`	A long-format data frame.
`sample_id`	Unique values in this column form the list of samples.
`primary_id`	Unique values in this column form the list of proteins to be quantified.
`secondary_id`	One or more columns. A concatenation of these columns determines the fragment ions used for quantification.
`intensity_col`	The column for intensities, or several columns, one per fragment ion (e.g. `paste0("Fr.", 0:11, ".Quantity")` in DIA-NN 2).
`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 numbers. Rows with a value less than the given value, or a missing value, are kept. Default Q.Value < 0.01 and PG.Q.Value < 0.01.

`filter_double_greater`
: A named vector of numbers. Rows with a value greater than the given value, or a missing value, are kept.

`intensity_col_sep`
: A separator character when a single text intensity column contains multiple values. It is not used for numeric intensity columns.

`intensity_col_id`
: Identities of multiple quantitative values: one column per intensity column, or a single text column split by `intensity_col_sep`.

`na_string`
: The value considered as NA, in addition to NA and NaN. For numeric intensity columns, a number (default 0).

`verbose`
: A logical value. The default `TRUE` prints progress and summary messages; `FALSE` prints none of them. Warnings and errors are always shown.

Details

A filter value is missing if it is NA, NaN, empty, or text that is not a number; such rows are kept.

Columns can be character, factor, integer, double or logical. Numbers used as identifiers (e.g. a charge column in `secondary_id`) are converted to text.

When there are multiple quantitative values per row, the `secondary_id` is appended with the corresponding entries of `intensity_col_id` when it is provided. Otherwise, integer values 1, 2, 3, etc... are used.

Value

A list as returned by `fast_read`.

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.

`fast_read_parquet` *Reading data from a parquet file*

Description

Reads a long-format parquet file, for example a DIA-NN `report.parquet`, one row group at a time for iq processing. Only the needed columns are read, and only one row group is held in memory at a time. Requires the **arrow** package.

Usage

```
fast_read_parquet(filename,
                  sample_id = "Run",
                  primary_id = "Protein.Group",
                  secondary_id = "Precursor.Id",
                  intensity_col = "Precursor.Normalised",
                  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 = NULL,
                  intensity_col_id = NULL,
                  na_string = "0",
                  verbose = TRUE)
```

Arguments

filename	A long-format parquet file.
sample_id	Unique values in this column form the list of samples.
primary_id	Unique values in this column form the list of proteins to be quantified.
secondary_id	One or more columns. A concatenation of these columns determines the fragment ions used for quantification.
intensity_col	The column for intensities, or several columns, one per fragment ion (e.g. `paste0("Fr.", 0:11, ".Quantity")` in DIA-NN 2).
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 numbers. Rows with a value less than the given value, or a missing value, are kept. Default Q.Value < 0.01 and PG.Q.Value < 0.01.
filter_double_greater	A named vector of numbers. Rows with a value greater than the given value, or a missing value, are kept.
intensity_col_sep	A separator character when a single text intensity column contains multiple values.
intensity_col_id	Identities of multiple quantitative values: one column per intensity column, or a single text column split by `intensity_col_sep`.
na_string	The value considered as NA, in addition to NA and NaN. For numeric intensity columns, a number (default 0).
verbose	A logical value. The default `TRUE` prints progress and summary messages; `FALSE` prints none of them. Warnings and errors are always shown.

Details

Filtering and missing values follow `fast_read`, and the result is the same as `fast_read` on the same data written as a tab-separated text file. A filter value is missing if it is NA, NaN, empty, or text that is not a number; such rows are kept.

When there are multiple quantitative values per row, the `secondary_id` is appended with the corresponding entries of `intensity_col_id` when it is provided. Otherwise, integer values 1, 2, 3, etc... are used.

Value

A list as returned by `fast_read`.

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.

`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, or a data frame with numeric columns, 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.

---

iq_format_to_protein_list

Protein data from the iq format

Description

Reads the data of one or more proteins from the iq format into a list of matrices, as returned by `create_protein_list`, for example to plot a protein with `plot_protein`.

Usage

```
iq_format_to_protein_list(iqformat_folder, proteins, normalization = TRUE)
```

Arguments

`iqformat_folder`	A folder with data in the iq format, as written by `long_format_to_iq_format`.
`proteins`	A vector of protein identifiers, as in the first column of `_protein.tsv`.
`normalization`	A logical value. The default `TRUE` applies the normalization of each sample stored in `_sample.tsv`, as `process_iq_format` does; `FALSE` gives the intensities as read.

Details

Only the files of the requested proteins are read. The intensities of an ion in a sample are summed, as in `process_iq_format`, so that `process_matrix` on a matrix of the list gives the same result as the row of that protein in the output of `process_iq_format` with the same method. A protein without values (it has no protein file, e.g. all its values are below the intensity cut-off) gets a matrix without rows, with a warning. An identifier that is not in `_protein.tsv` stops with an error.

Value

A list with one element for each protein in `proteins`, named by its identifier: a matrix of log2 intensities with the ions in rows (named as in `_ion.tsv`) and all samples in columns (named as in `_sample.tsv`); `NA` for a missing value.

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

long_format_to_iq_format, process_iq_format, plot_protein, process_matrix

---

long_format_to_iq_format

Long format input data to iq format

Description

This function reads long-format input data (a text file or a parquet file), filters it, and writes the results to the iq format. The input is read and written incrementally, so it is never held in memory as a whole.

Usage

```
long_format_to_iq_format(input_filename,
                         output_iqformat_folder,
                         sample_id = "Run",
                         primary_id = "Protein.Group",
                         secondary_id = c("Precursor.Id"),
                         intensity_col = "Precursor.Normalised",
                         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 = NULL,
                         intensity_col_id = NULL,
                         na_string = "0",
                         normalization = "median",
                         log2_intensity_cutoff = 0,
                         pdf_out = "qc-plots-iq-format.pdf",
                         pdf_width = 12,
                         pdf_height = 8,
                         show_boxplot = TRUE,
                         peptide_extractor = NULL,
                         rfasta = NULL,
                         buffer_mb = 1024,
                         verbose = TRUE)
```

Arguments

input_filename	The name of a parquet file (ending in .parquet; requires the **arrow** package) or of a tab-separated text file. The filtering is the same as in fast_read_parquet and fast_read. For a data frame, use process.
output_iqformat_folder	A folder for the iq format, new or empty.
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. Several columns, one per fragment ion, can be given (e.g. paste0("Fr.", 0:11, ".Quantity") for a DIA-NN 2 parquet file, with intensity_col_sep = NULL).
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 na_string 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	See peptide_extractor in process: if not NULL, the fragment statistics of each protein are computed and stored in _protein.tsv.
rfasta	See rfasta in process: the sequence coverage, with peptide_extractor.
buffer_mb	Memory in megabytes for buffering data. It changes only the memory use and speed, not the result.
verbose	A logical value. The default TRUE prints progress and summary messages; FALSE prints none of them. Warnings and errors are always shown.

Details

The input is read in chunks: blocks of lines of a text file, or row groups of a parquet file (converted to R in slices of rows). The values of each chunk are buffered (`buffer_mb`) and appended to the files. Only the lists of proteins, samples and ions are kept in memory; for a parquet file, also one row group at a time, as decoded by **arrow**.

The defaults are those of `process` (DIA-NN precursor quantities `Precursor.Normalised` per `Run`, filtered on `Q.Value` and `PG.Q.Value` below 0.01, log2 intensity cut-off 0), so that the same arguments give the same proteins as `process`.

The preprocessing is that of `fast_preprocess`: values that are zero or negative are left out, as are values not above `log2_intensity_cutoff` (by default 0). The protein files contain the intensities as read; the normalization, a shift of each sample in log2 space, is stored in the `normalization` column of `_sample.tsv` and applied by `process_iq_format`. If `peptide_extractor` is given, the fragment statistics of `process` (`n_fragments`, `n_peptides`, `peptides`, `XIC` and, with `rfasta`, `coverage`) are computed while the input is read, from all values read as in `process` (also those left out by the preprocessing), and stored as further columns of `_protein.tsv`; `process_iq_format` writes them with the protein annotation. Results are the same as with `process` on the same input.

Value

`NULL`, invisibly. The output is written to the folder `output_iqformat_folder`: `_protein.tsv` (proteins and annotation, and the fragment statistics if `peptide_extractor` is given), `_sample.tsv` (samples and their normalization), `_ion.tsv` (ions), and one file `<i>.tsv` for each protein with values (`i` its row in `_protein.tsv`), with the sample index, ion index and intensity of each value.

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_read_parquet`, `fast_preprocess`, `process_iq_format`, `process`

`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, 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,
           verbose = 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 not above this value in log2 space are ignored; NULL for no cutoff. Plot a histogram of all intensities to set this parameter. Zero and negative intensities (not finite in log2 space) are always removed.
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. For a numeric intensity column, a number (default 0).
show_boxplot	A logical value. The default TRUE value is to create boxplots of fragment intensities for each sample.
verbose	A logical value. The default TRUE prints progress messages; FALSE prints none of them. Warnings and errors are always shown.

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` *Protein quantification from a long-format table*

Description

Reads a long-format table of ion intensities (a tab-separated text file, a parquet file or a data frame), filters and normalizes the data, quantifies all proteins with one of the methods of `process_matrix`, and writes a table with one row per protein and one column per sample.

Usage

```
process(input_data,
        output_filename,
        sample_id = "Run",
        primary_id = "Protein.Group",
        secondary_id = "Precursor.Id",
        intensity_col = "Precursor.Normalised",
        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 = NULL,
        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, boxplot_max_points = 10000,
peptide_extractor = NULL,
rfasta = NULL,
method = "maxlfq_bit",
p1 = NULL, p2 = NULL, k = 1.345, min_M = 15,
rescale_method = "to_mean",
n_threads = -1,
verbose = TRUE)
```

Arguments

input_data	A data frame, the name of a parquet file (ending in .parquet), or the name of a tab-separated text file. See fast_read_data_frame, fast_read_parquet and fast_read.
output_filename	Output filename.
sample_id	The column of sample names.
primary_id	The column of protein identifiers. Unique values form the list of proteins.
secondary_id	The column(s) identifying an ion within a protein, concatenated with "_".
intensity_col	The column of intensities. Several columns can be given, one per fragment ion, e.g. paste0("Fr.", 0:11, ".Quantity") in DIA-NN 2.
annotation_col	Annotation columns, written after the protein identifier.
filter_string_equal	A named vector of strings. Only rows where the column equals the value are kept.
filter_string_not_equal	A named vector of strings. Only rows where the column differs from the value are kept.
filter_double_less	A named vector of numbers. Rows with a value less than the given value, or a missing value, are kept.
filter_double_greater	A named vector of numbers. Rows with a value greater than the given value, or a missing value, are kept.
intensity_col_sep	A separator character when a single text intensity column contains multiple values (e.g. ";" for the fragment intensities in Fragment.Quant.Corrected of DIA-NN 1.x), or NULL (the default) for one value per column.
intensity_col_id	Identities of multiple intensity values: one column per intensity column, or a single text column split by intensity_col_sep. If NULL, the values are numbered 1, 2, 3, ...
na_string	An intensity value considered missing, in addition to NA, NaN and empty values. For a text file, it is used with intensity_col_sep or several intensity columns.

`normalization` "median" for median normalization (see `fast_preprocess`) or "none".

`log2_intensity_cutoff`
: Values with a log2 intensity not above this cutoff are removed; the default 0 removes intensities of 1 or less. NULL for no cutoff. See `fast_preprocess`.

`pdf_out` A PDF file for quality control plots, or NULL for none. See `fast_preprocess`.

`pdf_width` Width of the PDF pages.

`pdf_height` Height of the PDF pages.

`show_boxplot` Whether to draw boxplots of the intensities per sample.

`boxplot_max_points`
: The maximum number of values per sample in the boxplots (evenly spaced values are used for a larger sample), which keeps the plots fast and small for large data; the median normalization always uses all values. NULL for all values.

`peptide_extractor`
: A function that maps ion identifiers to peptides, to compute fragment statistics per protein. It is called with the ions of one protein and must return one character value per ion. See Details.

`rfasta` A named list or a named character vector of protein sequences, named by protein accession as in `primary_id` (a protein group is split at ;), to compute the sequence coverage. Used with `peptide_extractor`. See `read_fasta_file`.

`method` The quantification method: one of the methods of `process_matrix` ("maxlfq", "maxlfq_bit", "weighted_maxlfq", "median_polish", "weighted_median_polish", "rlm", "weighted_rlm", "sum", "mean" or "topN"), or a function. See Details.

`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`.

`rescale_method` See `method` in `rescale`. The default "to_mean" shifts each connected component of samples so that the mean of its estimates equals the mean of its data (in log2 space). NULL for no rescaling.

`n_threads` The number of threads, used in parallel over proteins. See `n_threads` in `process_matrix`.

`verbose` A logical value. The default TRUE prints progress and summary messages; FALSE prints none of them. Warnings and errors are always shown.

Details

Input. The defaults are the same for all inputs: the precursor quantities `Precursor.Normalised` per Run of a DIA-NN report. A data frame is read in place, without copying (`fast_read_data_frame`). A parquet file is read one row group at a time, only the needed columns (`fast_read_parquet`; requires the **arrow** package). Any other input is read as a tab-separated text file (`fast_read`). Filtering is the same for all inputs; a missing filter value (NA, NaN, empty, or text that is not a number) keeps the row.

Preprocessing. Intensities are log2-transformed; zero and negative intensities are removed, as are values not above `log2_intensity_cutoff` (by default 0, i.e. intensities of 1 or less); and samples are median normalized (`fast_preprocess`).

Quantification. For each protein, a matrix of log2 intensities is formed with ions as rows (in input order) and all samples as columns; an ion with more than one value in a sample gets the log2 of the summed intensities. The matrix is quantified with `method` and the result rescaled with `rescale_method`, as in `process_iq_format`, without writing iq-format files. For the built-in methods, all proteins are processed in C++ in parallel. A function given as `method` is called in R for each protein with this matrix, with the ion identifiers as row names and the sample names as column names, and must return a numeric vector with one value per sample. Values that are not finite (`-Inf`, `Inf`, `NaN`) become `NA`. For example, `function(X) log2(colSums(2^X, na.rm = TRUE))` with `rescale_method = NULL` gives the log2 summed intensities per sample, with `NA` for a sample without data, as the method `"sum"`.

Output. A tab-separated file with one row for each protein that has data after preprocessing. The columns are `primary_id`, the `annotation_col` columns, the fragment statistics (if `peptide_extractor` is given), and one column per sample with the log2 estimates; a sample without data for a protein is `NA`.

Fragment statistics. If `peptide_extractor` is not `NULL`, the columns `n_fragments` (the number of MS2 fragment ions or precursors depending on `intensity_col`), `n_peptides` (the number of unique values of `peptide_extractor` on the ions), `peptides` and `XIC` (the summed intensities, before preprocessing) are added, and `coverage` (sequence coverage in percent for each protein in the group) if `rfasta` is given. `peptide_extractor` is called with the `secondary_id` part of the ion identifiers, without the suffix of the intensity values (`1`, `2`, ... or the `intensity_col_id` values), e.g. a precursor id such as `(UniMod:1)AAAAAAGAGPEMVR2`, also when `intensity_col` has one column per fragment ion; the `ion_prefix` component returned by `fast_read`, `fast_read_parquet` or `fast_read_data_frame` shows these values. An example extractor for DIA-NN precursor ids is `function(x) gsub("\\(UniMod:\\d+\\)|\\d", "", x)`, which removes the modifications and the charge state: `(UniMod:1)AAAAAAGAGPEMVR2` becomes `AAAAAAGAGPEMVR`.

Value

`NULL`, invisibly. The result is written to `output_filename`.

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_read_parquet`, `fast_read_data_frame`, `fast_preprocess`, `process_matrix`, `rescale`, `process_iq_format`, `read_fasta_file`

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(iqformat_folder,
                  output_filename = NULL,
                  method = "maxlfq_bit",
                  p1 = NULL, p2 = NULL, k = 1.345, min_M = 15,
                  rescale_method = "to_mean",
                  buffer_mb = 1024,
                  n_threads = -1,
                  verbose = TRUE)
```

Arguments

iqformat_folder	A folder with data in the iq format, as written by long_format_to_iq_format.
output_filename	Output file name. If NULL, a file name derived from iqformat_folder will be used.
method	See method in process_matrix, or a function. A function is called for each protein with the matrix of log2 intensities (ions in rows, with the ion indices of the iq format as row names; samples in columns, with the sample names as column names; NA for missing values) and must return a numeric vector with one value per sample. Values that are not finite (-Inf, Inf, NaN) become NA.
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.
rescale_method	Rescale the output intensities. See method in rescale. The default "to_mean" shifts each connected component of samples so that the mean of its estimates equals the mean of its data (in log2 space); NULL for no rescaling.
buffer_mb	Memory in megabytes for a batch of proteins quantified together (see Details), or NULL to quantify the proteins one by one in R.
n_threads	See n_threads in process_matrix.
verbose	A logical value. The default TRUE prints progress and summary messages; FALSE prints none of them. Warnings and errors are always shown.

Details

The protein files are read in batches of about buffer_mb megabytes of values, and the proteins of a batch are quantified in parallel (n_threads), as in process. With buffer_mb = NULL, or a function as method, each protein is read with read.delim and quantified with process_matrix in R. The log2 values are shifted by the normalization of each sample (the normalization column of _sample.tsv, if present. All columns of _protein.tsv are written before the samples: the protein annotation, and the fragment statistics if long_format_to_iq_format computed them.

Value

NULL, invisibly. A tab-separated table is written to output_filename (by default iqformat_folder followed by -quant.tsv), with one row for each protein with values, columns for the protein annotation and statistics, and the 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.

See Also

long_format_to_iq_format, process, process_matrix, rescale

---

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,
                     verbose = TRUE)
```

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 na_string 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 that maps ion identifiers to peptides, to compute fragment statistics per protein, or NULL (the default) for none. It is called with the ions of one protein and must return one character value per ion. See Value.

rfasta — A named list of protein sequences, named by protein accession as in primary_id (a protein group is split at ;), e.g. from read_fasta_file. With peptide_extractor and an input file, the sequence coverage of each protein is added.

verbose — A logical value. The default TRUE prints progress and summary messages; FALSE prints none of them. Warnings and errors are always shown.

Value

An input data frame is filtered and processed with preprocess and fast_MaxLFQ; 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: the columns n_fragments and n_peptides, and for an input file also peptides, XIC (the summed intensities, before preprocessing) and, with rfasta, coverage. 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("\\(UniMod:\\d+\\)|\\d|_", "", x), which removes modifications, digit numbers, and underscores in an ion descriptor to obtain 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.

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 = "maxlfq_bit",
               p1 = NULL, p2 = NULL, k = 1.345, min_M = 15,
               rescale_method = "to_mean",
               n_threads = -1)
```

Arguments

X	A matrix, or a data frame with numeric columns, 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", "weighted_rlm", "sum", "mean" or "topN"; or a function. The default is "maxlfq_bit", as in process and process_iq_format. See Details for "sum", "mean" and "topN". A function is called with X and must return a numeric vector with one value per sample (column of X). Values that are not finite (-Inf, Inf, NaN) become NA. For example, function(X) log2(colSums(2^X, na.rm = TRUE)) with rescale_method = NULL gives the log2 summed intensities per sample, with NA for a sample without data, as the method "sum".
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", "maxlfq" and "weighted_maxlfq", it specifies the memory level of the solver (default 0). For "median_polish", "weighted_median_polish", "rlm" and "weighted_rlm", it is the maximum number of optimization rounds. For "topN", it is the number of ions (default 3).
k	Weighting function parameter.
min_M	Minimum number of rows for the weighted methods.
rescale_method	The result is rescaled with rescale using this method. The default "to_mean" shifts each connected component of samples so that the mean of its estimates equals the mean of its data (in log2 space), as in process and process_iq_format. NULL for no rescaling.
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.

Details

The methods "sum", "mean" and "topN" summarize the ions of each sample directly: the log2 of the summed intensities, the mean of the log2 intensities, and the mean of the log2 intensities of the p2 ions with the highest mean log2 intensity over the samples (as topN). A sample without data gets NA. To keep these values as they are, use rescale_method = NULL; the default rescale_method shifts them like the estimates of the other methods.

The methods "maxlfq" and "weighted_maxlfq" solve the MaxLFQ least-squares problem with the iterative solver of "maxlfq_bit". A package built with the original MaxLFQ implementation (using the Eigen library, see the file src/Makevars of the source package) solves it exactly instead.

Value

A numeric row vector, one value per sample (column of X). For the built-in methods, a matrix with a single row gives that row (before rescaling; the "to_mean" and "to_sum" rescaling leave it unchanged).

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.

Examples

```
# a data matrix for a single protein: log2 intensities, ions in rows, samples in columns
X <- matrix(c(10, 12, 12, NA, NA, 10), nrow = 2,
            dimnames = list(c("ion1", "ion2"),
                            c("sample1", "sample2", "sample3")))
X

# relative quantification: sample2 is 4-fold higher than sample1 (a difference of 2 in
# log2 space), the difference of ion1, the only ion they share; similarly, sample1 is
# 4-fold higher than sample3 (ion2)
process_matrix(X, method = "maxlfq_bit", rescale_method = NULL)  # 11 13 9

# rescaled to the total intensity: the same differences between samples, and the
# intensities of the estimates add up to those of the data
x <- process_matrix(X, method = "maxlfq_bit", rescale_method = "to_sum")
x                       # 10.93 12.93 8.93
sum(2^X, na.rm = TRUE)  # 10240
sum(2^x)                # 10240
```

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, as numbers or names (names as read.delim makes them, e.g. . for -).
data_in_log_space	A logical value. If FALSE, the numerical data will be log2-transformed; zero and negative values are then treated as missing.
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.

`read_fasta_file` *Reading a FASTA file*

Description

Reads protein sequences from a FASTA file into a named list, for example for the `rfasta` argument of `process` and `process_long_format`.

Usage

```
read_fasta_file(fasta_file_name, regex_id = "^>.*\\|(.*)\\|.*")
```

Arguments

`fasta_file_name`	A FASTA file.
`regex_id`	A regular expression applied to each header line (the line starting with `>`); the first group in parentheses is the name of the sequence. The default takes the accession between the two `|` of a UniProt header such as `>sp|P04406|G3P_HUMAN` ....

Details

A sequence is the concatenation of the lines below its header, with all white space removed. A header that does not match `regex_id` gives the whole header line, without `>`, as the name. Lines before the first header are ignored.

Value

A named list of sequences (character strings), in the order of the file.

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

process

---

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, or a data frame with numeric columns, of log2-transformed intensities. Columns represent samples, and rows represent fragment ions.
method	One of the methods for rescaling: "median_mean", "to_median", "to_mean" or "to_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 the "to_median" and "to_mean" methods, each component is rescaled to the median or mean of the input data. For the "to_sum" method, each component is shifted so that its values, in linear scale (2^x), add up to the summed intensities of its input data (2^X) in the samples of the component; a single-sample component gets the log2 of the summed intensities of that sample. For all methods, a sample without input data (an all-NA column of X) gets NA, and a component whose values in x are all NA is left unchanged.

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, 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, 37

connected_component, 2
create_protein_list, 3, *5*, *15*
create_protein_table, 4

extract_annotation, 5

fast_MaxLFQ, 6, *31*
fast_preprocess, *7*, 7, *17, 18*, *26, 27*, *30, 31*
fast_read, *7, 8*, 9, *11, 12*, *14*, *17, 18*, *25–27*, *30, 31*
fast_read_data_frame, 11, *25–27*
fast_read_parquet, *11*, 12, *17, 18*, *25–27*

huber_weight, 14

iq_format_to_protein_list, 15

long_format_to_iq_format, *15, 16*, 16, *28, 29*

maxLFQ, *5*, 18
meanInt, *5*, 19
median_polish, *5*, 20

plot_protein, *15, 16*, 21
preprocess, *4*, *6*, 22, *31*
process, *10*, *17, 18*, 24, *29*, *32*, *35, 36*
process_iq_format, *15, 16*, *18*, *27*, 28, *32*
process_long_format, 29, *35*
process_matrix, *6*, *15, 16*, *24*, *26–29*, 32
process_wide_format, 34

read_fasta_file, *26, 27*, *31*, 35
rescale, *26–29*, *32*, 36

spikeins, 37

topN, *5*, *33*, 38