Package ‘iq’

June 27, 2020

Type Package

Title Protein Quantification in Mass Spectrometry-Based Proteomics

Version 1.9

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


Depends R (>= 2.10)

License BSD_3_clause + file LICENSE

Encoding UTF-8

LazyData true

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation yes

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

Repository CRAN

Date/Publication 2020-06-27 12:40:03 UTC

R topics documented:

create_protein_list .................................................. 2
create_protein_table .................................................. 3
extract_annotation ................................................... 4
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

See Also
preprocess
create_protein_table

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.
annotation     A vector of annotations, one for each protein.

Author(s)

Thang V. Pham

References


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, 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
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 sample 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.

annotation  
A vector of strings indicating membership in case of multiple connected components for each row of estimate.

Author(s)

Thang V. Pham

References

**See Also**

`fast_read`, `fast_preprocess`

---

**fast_preprocess**  
*Data filtering and normalization*

---

**Description**

Filters out low intensities and performs median normalization.

**Usage**

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

**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.

**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**

fast_read

See Also

fast_read

---

**Description**

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

**Usage**

```r
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_double_less = c("PG.Qvalue" = "0.01", "EG.Qvalue" = "0.01"))
```

**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 filter are kept.
- `filter_double_less`: A named vector of strings. Only rows satisfying the filter are kept. Default PG.Qvalue < 0.01 and EG.Qvalue < 0.01.

**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

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.
anotation
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
**meanInt**  

*The meanInt algorithm for protein quantification*

---

**Description**

Estimates protein abundances by averaging all associated ion intensities

**Usage**

```r
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.
- **annotation**
  - Reserved, currently an empty string.

**Author(s)**

Thang V. Pham

**References**

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.

annotation  
Reserved, currently an empty string

Author(s)

Thang V. Pham

References


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


Examples

```r
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)
```

Description

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

Usage

```r
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",
    ...)
preprocess

```r
median_normalization = TRUE,
log2_intensity_cutoff = 0,
pdf_out = "qc-plots.pdf",
pdf_width = 12,
pdf_height = 8)
```

**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.

**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**

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)

---

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)
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.
annotation Reserved, currently an empty string.

Author(s)

Thang V. Pham

References

Index

*Topic datasets
  spikeins, 13

create_protein_list, 2, 3
create_protein_table, 3
extract_annotation, 4
fast_MaxLFQ, 5
fast_preprocess, 6, 6
fast_read, 6, 7, 7
maxLFQ, 3, 8
meanInt, 3, 9
median_polish, 3, 10
plot_protein, 10
preprocess, 2, 4, 11
spikeins, 13
topN, 3, 14