Package ‘PRONE’

November 13, 2024

Type Package

Title The PROteomics Normalization Evaluator

Version 1.1.0

Description High-throughput omics data are often affected by systematic biases introduced throughout all the steps of a clinical study, from sample collection to quantification.
Normalization methods aim to adjust for these biases to make the actual biological signal more prominent.
However, selecting an appropriate normalization method is challenging due to the wide range of available approaches.
Therefore, a comparative evaluation of unnormalized and normalized data is essential in identifying an appropriate normalization strategy for a specific data set.
This R package provides different functions for preprocessing, normalizing, and evaluating different normalization approaches.
Furthermore, normalization methods can be evaluated on downstream steps, such as differential expression analysis and statistical enrichment analysis.
Spike-in data sets with known ground truth and real-world data sets of biological experiments acquired by either tandem mass tag (TMT) or label-free quantification (LFQ) can be analyzed.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.4.0), SummarizedExperiment

Imports dplyr, magrittr, data.table, RColorBrewer, ggplot2, S4Vectors, ComplexHeatmap, stringr, NormalyzerDE, tibble, limma, MASS, edgeR, matrixStats, preprocessCore, stats, gtools, methods, ROTS, ComplexUpset, tidyr, purrr, circlize, gprofiler2, plotROC, MSnbase, UpSetR, dendsort, vsn, Biobase, reshape2, POMA, ggtext, scales, DEqMS

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

BugReports `https://github.com/lisiarend/PRONE/issues`

URL `https://github.com/lisiarend/PRONE`

VignetteBuilder knitr

Config/testthat/edition 3

biocViews Proteomics, Preprocessing, Normalization, DifferentialExpression, Visualization

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

git_branch devel

git_last_commit fd54f01

git_last_commit_date 2024-10-29

Repository Bioconductor 3.21

Date/Publication 2024-11-12

Author Lis Arend [aut, cre] (ORCID: <`https://orcid.org/0000-0001-7990-8385`>)

Maintainer Lis Arend <`lis.arend@tum.de`>

Contents

`PRONE-package` *PRONE: The PROteomics Normalization Evaluator*

Description

High-throughput omics data are often affected by systematic biases introduced throughout all the steps of a clinical study, from sample collection to quantification. Normalization methods aim to adjust for these biases to make the actual biological signal more prominent. However, selecting an appropriate normalization method is challenging due to the wide range of available approaches. Therefore, a comparative evaluation of unnormalized and normalized data is essential in identifying an appropriate normalization strategy for a specific data set. This R package provides different functions for preprocessing, normalizing, and evaluating different normalization approaches. Furthermore, normalization methods can be evaluated on downstream steps, such as differential expression analysis and statistical enrichment analysis. Spike-in data sets with known ground truth and real-world data sets of biological experiments acquired by either tandem mass tag (TMT) or label-free quantification (LFQ) can be analyzed.

Author(s)

Maintainer: Lis Arend `<lis.arend@tum.de>` (ORCID)

See Also

Useful links:

- `https://github.com/lisiarend/PRONE`
- Report bugs at `https://github.com/lisiarend/PRONE/issues`

`apply_thresholds` *Apply other thresholds to DE results*

Description

Apply other thresholds to DE results

Usage

```
apply_thresholds(
  de_res,
  logFC = TRUE,
  logFC_up = 1,
  logFC_down = -1,
  p_adj = TRUE,
  alpha = 0.05
)
```

Arguments

`de_res`	data table resulting of run_DE
`logFC`	Boolean specifying whether to apply a logFC threshold (TRUE) or not (FALSE)
`logFC_up`	Upper log2 fold change threshold (dividing into up regulated)
`logFC_down`	Lower log2 fold change threshold (dividing into down regulated)
`p_adj`	Boolean specifying whether to apply a threshold on adjusted p-values (TRUE) or on raw p-values (FALSE)
`alpha`	Threshold for adjusted p-values or p-values

Value

data table updating the Change column with the newly applied thresholds

Examples

```
data(tuberculosis_TMT_de_res)
de_res <- apply_thresholds(tuberculosis_TMT_de_res, logFC = FALSE,
                           p_adj = TRUE, alpha = 0.01)
```

`check_DEqMS_parameter` *Helper function to check whether the DEqMS_PSMs_column is in the SummarizedExperiment object*

Description

Helper function to check whether the DEqMS_PSMs_column is in the SummarizedExperiment object

Usage

```
check_DEqMS_parameter(se, DEqMS_PSMs_column)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`DEqMS_PSMs_column`	String specifying which column name to use for DEqMS (default NULL)

Value

None

`check_DE_parameters` *Check parameters for DE analysis*

Description

Check parameters for DE analysis

Usage

```
check_DE_parameters(
  se,
  ain = NULL,
  condition = NULL,
  comparisons = NULL,
  DE_method = "limma",
  covariate = NULL,
  logFC = TRUE,
  logFC_up = 1,
  logFC_down = -1,
  p_adj = TRUE,
  p_adj_method = "BH",
```

```
  alpha = 0.05,
  B = 100,
  K = 500,
  DEqMS_PSMs_column = NULL
)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`ain`	Vector of strings which assay should be used as input (default NULL). If NULL then all normalization of the se object are plotted next to each other.
`condition`	column name of condition (if NULL, condition saved in SummarizedExperiment will be taken)
`comparisons`	Vector of comparisons that are performed in the DE analysis (from specify_comparisons method)
`DE_method`	String specifying which DE method should be applied (limma, ROTS, DEqMS)
`covariate`	String specifying which column to include as covariate into limma
`logFC`	Boolean specifying whether to apply a logFC threshold (TRUE) or not (FALSE)
`logFC_up`	Upper log2 fold change threshold (dividing into up regulated)
`logFC_down`	Lower log2 fold change threshold (dividing into down regulated)
`p_adj`	Boolean specifying whether to apply a threshold on adjusted p-values (TRUE) or on raw p-values (FALSE)
`p_adj_method`	String specifying the method for adjusted p-values
`alpha`	Threshold for adjusted p-values or p-values
`B`	Number of bootstrapping for ROTS
`K`	Number of top-ranked features for reproducibility optimization
`DEqMS_PSMs_column`	String specifying which column name to use for DEqMS (default NULL)

Value

list of checked assays and condition column name

`check_input_assays` *Helper function to check whether all given assays are in SummarizedExperiment object*

Description

Helper function to check whether all given assays are in SummarizedExperiment object

Usage

```
check_input_assays(se, ain)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`ain`	Vector of strings which assay should be used as input (default NULL). If NULL then all normalization of the se object are plotted next to each other.

Value

NULL if no methods in SummarizedExperiment object, else all available methods ready for visualization

---

`check_plot_DE_parameters`

Helper function to check the parameters for plotting the DE results

Description

Helper function to check the parameters for plotting the DE results

Usage

```
check_plot_DE_parameters(de_res, ain, comparisons)
```

Arguments

`de_res`	data table resulting of run_DE
`ain`	String of normalization methods to visualize (must be valid normalization methods saved in de_res)
`comparisons`	Vector of comparisons (must be valid comparisons saved in de_res)

Value

list of valid inputs for plotting functions

`check_stats_spiked_DE_parameters` *Helper function to check the parameters for plotting the DE stats of spike-in data sets*

Description

Helper function to check the parameters for plotting the DE stats of spike-in data sets

Usage

```
check_stats_spiked_DE_parameters(stats, ain, comparisons)
```

Arguments

`stats`	data table resulting of get_spiked_stats_DE
`ain`	String of normalization methods to visualize (must be valid normalization methods saved in de_res)
`comparisons`	Vector of comparisons (must be valid comparisons saved in de_res)

Value

list of valid inputs for plotting functions

`detect_outliers_POMA` *Outlier detection via POMA R Package*

Description

Outlier detection via POMA R Package

Usage

```
detect_outliers_POMA(
  se,
  ain = "log2",
  condition = NULL,
  method = "euclidean",
  type = "median",
  group = TRUE,
  coeff = 1.5
)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`ain`	String which data type should be used (default raw)
`condition`	Column name of condition (if NULL, condition saved in SummarizedExperiment will be taken)
`method`	String specifying the method that should be used to calculate the distance matrix
`type`	String specifying the type of distance calculation to centroid or spatial median
`group`	String specifying if the outlier detection should be performed multi-variate (with conditions) or on the complete data set
`coeff`	This value corresponds to the classical 1.5 in Q3 + 1.5 * IQR formula to detect outliers. By changing this value, the permissiveness in outlier detection will change.

Value

list of two ggplot objects and a data.table with outlier samples

Examples

```
data(tuberculosis_TMT_se)
poma_res <- detect_outliers_POMA(tuberculosis_TMT_se, ain="raw",
                                 condition = NULL, method="euclidean",
                                 type="median", group=TRUE, coeff = 1.5)
```

`eigenMSNorm` *EigenMS Normalization*

Description

EigenMS fits an analysis of variance model to estimate the effects of the experimental factors on the data using the knowledge about the experimental design, and then applies singular value decomposition to identify systematic trends contributing to significant variation not explained by the experimental factors Log2-scaled data should be used as input (on_raw = FALSE).

Usage

```
eigenMSNorm(se, ain = "log2", aout = "EigenMS", on_raw = FALSE)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomic dataset
`ain`	String which assay should be used as input
`aout`	String which assay should be used to save normalized data
`on_raw`	Boolean specifying whether normalization should be performed on raw or log2-scaled data

Value

SummarizedExperiment containing the EigenMS normalized data as assay (on log2 scale)

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- eigenMSNorm(tuberculosis_TMT_se, ain = "log2",
                                    aout = "EigenMS", on_raw = FALSE)
```

`export_data` *Export the SummarizedExperiment object, the meta data, and the normalized data.*

Description

Export the SummarizedExperiment object, the meta data, and the normalized data.

Usage

```
export_data(se, out_dir, ain = NULL)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`out_dir`	Path of output directory
`ain`	Vector of strings which assay should be downloaded (default NULL). If NULL then all assays of the se object are saved.

Value

Nothing

Examples

```
data(tuberculosis_TMT_se)
## Not run: export_data(tuberculosis_TMT_se, out_dir = "data/",
            ain = c("IRS_on_RobNorm", "IRS_on_Median"))
## End(Not run)
```

`express_to_DT` *Helper function to transform an expression data frame to a data table*

Description

Helper function to transform an expression data frame to a data table

Usage

```
express_to_DT(expr_data, column_names, row_names)
```

Arguments

`expr_data`	Expression data frame containing the expression data
`column_names`	Column names of the expression data
`row_names`	Row names of the expression data

Value

Data table containing the expression data

`extract_consensus_DE_candidates`
Extract consensus DE candidates

Description

Extract consensus DE candidates

Usage

```
extract_consensus_DE_candidates(
  de_res,
  ain = NULL,
  comparisons = NULL,
  norm_thr = 0.8,
  per_comparison = FALSE
)
```

Arguments

`de_res`	data table resulting of run_DE
`ain`	Vector of strings of normalization methods to visualize (must be valid normalization methods saved in de_res)
`comparisons`	Vector of comparisons (must be valid comparisons saved in de_res)
`norm_thr`	Threshold for the number of normalization methods that must agree on a DE candidate
`per_comparison`	Logical indicating if the consensus should be calculated per comparison

Value

data table with consensus DE candidates

Examples

```
data(tuberculosis_TMT_de_res)
extract_consensus_DE_candidates(tuberculosis_TMT_de_res, ain = NULL,
                   comparisons = NULL, norm_thr = 0.8, per_comparison = TRUE)
```

`extract_limma_DE` *Extract the DE results from eBayes fit of perform_limma function.*

Description

Extract the DE results from eBayes fit of perform_limma function.

Usage

```
extract_limma_DE(
  fit,
  comparisons,
  logFC = TRUE,
  logFC_up = 1,
  logFC_down = -1,
  p_adj = TRUE,
  alpha = 0.05
)
```

Arguments

`fit`	eBayes object resulting from perform_limma method
`comparisons`	Vector of comparisons that are performed in the DE analysis (from specify_comparisons method)
`logFC`	Boolean specifying whether to apply a logFC threshold (TRUE) or not (FALSE)

`logFC_up`	Upper log2 fold change threshold (dividing into up regulated)
`logFC_down`	Lower log2 fold change threshold (dividing into down regulated)
`p_adj`	Boolean specifying whether to apply a threshold on adjusted p-values (TRUE) or on raw p-values (FALSE)
`alpha`	Threshold for adjusted p-values or p-values

Value

Data table with limma DE results

`filter_out_complete_NA_proteins`
Remove proteins with NAs in all samples

Description

Remove proteins with NAs in all samples

Usage

```
filter_out_complete_NA_proteins(se)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set

Value

filtered SummarizedExperiment object

Examples

```
data(tuberculosis_TMT_se)
tuberulosis_TMT_se <- filter_out_complete_NA_proteins(tuberculosis_TMT_se)
```

`filter_out_NA_proteins_by_threshold` *Filter proteins based on their NA pattern using a specific threshold*

Description

Filter proteins based on their NA pattern using a specific threshold

Usage

```
filter_out_NA_proteins_by_threshold(se, thr = 0.8)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`thr`	Threshold for the minimum fraction of valid values allowed for any protein

Value

filtered SummarizedExperiment object

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- filter_out_NA_proteins_by_threshold(tuberculosis_TMT_se,
                                                           thr = 0.8)
```

`filter_out_proteins_by_ID` *Remove proteins by their ID*

Description

Remove proteins by their ID

Usage

```
filter_out_proteins_by_ID(se, protein_ids)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`protein_ids`	Vector of protein IDs that should be kept

Value

filtered SummarizedExperiment object

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- filter_out_proteins_by_ID(tuberculosis_TMT_se,
                                protein_ids = c("P0A8V2", "P0A8V2"))
```

`filter_out_proteins_by_value`

Remove proteins by value in specific column

Description

Remove proteins by value in specific column

Usage

```
filter_out_proteins_by_value(se, column_name = "Reverse", values = c("+"))
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`column_name`	name of column of which proteins with a specific value should be removed
`values`	value of the column defining the proteins that should be removed

Value

filtered SummarizedExperiment object

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- filter_out_proteins_by_value(tuberculosis_TMT_se,
                                 column_name = "Reverse", values = c("+"))
```

`get_color_value` *Helper function to get correct value for coloration of plots (color_by parameter)*

Description

Helper function to get correct value for coloration of plots (color_by parameter)

Usage

```
get_color_value(se, color_by)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`color_by`	String specifying the column to color the samples (If NULL, the condition column of the SummarizedExperiment object is used. If "No", no color bar added.)

Value

String of column to color or NULL if no color should be applied

---

`get_complete_dt` *Function to get a long data table of all intensities of all kind of normalization*

Description

Function to get a long data table of all intensities of all kind of normalization

Usage

```
get_complete_dt(se, ain = NULL)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`ain`	String which assay should be used as input (default NULL) If NULL then all normalization of the SummarizedExperiment object are plotted next to each other (except raw).

Value

data table

`get_complete_pca_dt` *Function to get a long data table of all PCA1 and PCA2 values of all kind of normalization*

Description

Function to get a long data table of all PCA1 and PCA2 values of all kind of normalization

Usage

```
get_complete_pca_dt(se, ain = NULL)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`ain`	String which assay should be used as input (default NULL) If NULL then all normalization of the SummarizedExperiment object are plotted next to each other (except raw).

Value

data table

`get_condition_value` *Helper function to check the condition value*

Description

Helper function to check the condition value

Usage

```
get_condition_value(se, condition)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`condition`	column name of condition (if NULL, condition saved in SummarizedExperiment will be taken)

Value

String of column for condition

`get_facet_value` *Helper function to get correct value for faceting of plots (facet_by parameter)*

Description

Helper function to get correct value for faceting of plots (facet_by parameter)

Usage

```
get_facet_value(se, facet_by)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`facet_by`	String specifying the column to facet the samples (If NULL or "No", no faceting is done.)

Value

String of column to facet or NULL if no faceting should be done

---

`get_label_value` *Helper function to get correct value for sample labeling of plots (label_by parameter)*

Description

Helper function to get correct value for sample labeling of plots (label_by parameter)

Usage

```
get_label_value(se, label_by)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`label_by`	String specifying the column to label the samples (If NULL, the labels column of the SummarizedExperiment object is used. If "No", no labeling of samples done.)

Value

String of column to label or NULL if no label should be applied

---

`get_NA_overview` *Function returning some values on the numbers of NA in the data*

Description

Function returning some values on the numbers of NA in the data

Usage

```
get_NA_overview(se, ain = "log2")
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`ain`	String which data type should be used (default raw)

Value

list with total amount of values in the data, amount of NA values, and the percentage of NAs

Examples

```
data(tuberculosis_TMT_se)
get_NA_overview(tuberculosis_TMT_se, ain="log2")
```

`get_normalization_methods` *Function to return available normalization methods' identifier names*

Description

Function to return available normalization methods' identifier names

Usage

```
get_normalization_methods()
```

Value

Vector of normalization methods

Examples

```
get_normalization_methods()
```

`get_overview_DE` *Get overview table of DE results*

Description

Get overview table of DE results

Usage

```
get_overview_DE(de_res)
```

Arguments

`de_res`	data table resulting of run_DE

Value

data table of numbers of DE proteins per comparison and per normalization method

Examples

```
data(tuberculosis_TMT_de_res)
get_overview_DE(tuberculosis_TMT_de_res)
```

`get_proteins_by_value` *Get proteins by value in specific column*

Description

Get proteins by value in specific column

Usage

```
get_proteins_by_value(se, column_name = "Reverse", values = c("+"))
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`column_name`	name of column of which proteins with a specific value should be identified
`values`	value of the column defining the proteins that should be identified

Value

vector of protein IDs

Examples

```
data(tuberculosis_TMT_se)
proteins <- get_proteins_by_value(tuberculosis_TMT_se,
                     column_name = "Potential.contaminant", values = c("+"))
```

`get_shape_value` *Helper function to get correct value for shaping of plots (shape_by parameter)*

Description

Helper function to get correct value for shaping of plots (shape_by parameter)

Usage

```
get_shape_value(se, shape_by)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`shape_by`	String specifying the column to shape the samples (If NULL or "No", no shaping is done.)

Value

String of column to shape or NULL if no shaping should be done

---

`get_spiked_stats_DE` *Get performance metrics of DE results of spike-in data set.*

Description

Get performance metrics of DE results of spike-in data set.

Usage

```
get_spiked_stats_DE(se, de_res)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`de_res`	data table resulting of run_DE

Value

data table with multiple performance metrics of the DE results

Examples

```
data(spike_in_se)
data(spike_in_de_res)
stats <- get_spiked_stats_DE(spike_in_se, spike_in_de_res)
```

`globalIntNorm` *Total Intensity Normalization*

Description

Intensities of each variable in a sample are divided with the sum of intensities of all variables in the sample and multiplied with the median or mean of sum of intensities of all variables in all samples. Raw data should be taken as input (on_raw = TRUE).

Usage

```
globalIntNorm(
  se,
  ain = "raw",
  aout = "GlobalMedian",
  type = "median",
  on_raw = TRUE
)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`ain`	String which assay should be used as input
`aout`	String which assay should be used to save normalized data
`type`	String whether to use median or mean to calculate the scaling factor
`on_raw`	Boolean specifying whether normalization should be performed on raw or log2-scaled data

Value

SummarizedExperiment containing the total intensity normalized data as assay (on log2 scale)

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- globalIntNorm(tuberculosis_TMT_se, ain = "raw",
                                       aout = "GlobalMedian",
                                       type = "median",
                                       on_raw = TRUE)
```

---

globalMeanNorm *Total Intensity Normalization Using the Mean for the Calculation of Scaling Factors*

Description

Intensities of each variable in a sample are divided with the sum of intensities of all variables in the sample and multiplied with the mean of sum of intensities of all variables in all samples. Raw data should be taken as input (on_raw = TRUE).

Usage

```
globalMeanNorm(se, ain = "raw", aout = "GlobalMean", on_raw = TRUE)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`ain`	String which assay should be used as input
`aout`	String which assay should be used to save normalized data
`on_raw`	Boolean specifying whether normalization should be performed on raw or log2-scaled data

Value

SummarizedExperiment containing the total intensity normalized data as assay (on log2 scale)

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- globalMeanNorm(tuberculosis_TMT_se, ain = "raw",
                             aout = "GlobalMean", on_raw = TRUE)
```

`globalMedianNorm` *Total Intensity Normalization Using the Median for the Calculation of Scaling Factors*

Description

Intensities of each variable in a sample are divided with the sum of intensities of all variables in the sample and multiplied with the median of sum of intensities of all variables in all samples. Raw data should be taken as input (on_raw = TRUE).

Usage

```
globalMedianNorm(se, ain = "raw", aout = "GlobalMedian", on_raw = TRUE)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`ain`	String which assay should be used as input
`aout`	String which assay should be used to save normalized data
`on_raw`	Boolean specifying whether normalization should be performed on raw or log2-scaled data

Value

SummarizedExperiment containing the total intensity normalized data as assay (on log2 scale)

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- globalMedianNorm(tuberculosis_TMT_se, ain = "raw",
                                        aout = "GlobalMedian", on_raw = TRUE)
```

`impute_se` *Method to impute SummarizedExperiment. This method performs a mixed imputation on the proteins. It uses a k-nearest neighbor imputation for proteins with missing values at random (MAR) and imputes missing values by random draws from a left-shifted Gaussian distribution for proteins with missing values not at random (MNAR).*

Description

Method to impute SummarizedExperiment. This method performs a mixed imputation on the proteins. It uses a k-nearest neighbor imputation for proteins with missing values at random (MAR) and imputes missing values by random draws from a left-shifted Gaussian distribution for proteins with missing values not at random (MNAR).

Usage

```
impute_se(se, ain = NULL, condition = NULL)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics dataset
`ain`	Vector of strings which assay should be used as input (default NULL). If NULL then all normalization of the se object are plotted next to each other.
`condition`	name of column of colData(se) representing the conditions of the data

Value

SummarizedExperiment with imputed intensities

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- remove_samples_manually(tuberculosis_TMT_se,
                           column = "Label", values = c("1.HC_Pool1"))
tuberculosis_TMT_se <- impute_se(tuberculosis_TMT_se, ain = NULL,
                           condition = NULL)
```

`irsNorm` *Internal Reference Scaling Normalization*

Description

IRS makes different measurements of the same thing all exactly the same and puts all of the intensities on the same scale. Raw data should be taken as input (on_raw = TRUE)

Usage

```
irsNorm(se, ain = "raw", aout = "IRS", on_raw = TRUE)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomic dataset
`ain`	String which assay should be used as input
`aout`	String which assay should be used to save normalized data
`on_raw`	Boolean specifying whether normalization should be performed on raw or log2-scaled data

Value

SummarizedExperiment containing the IRS normalized data as assay (on log2 scale)

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- irsNorm(tuberculosis_TMT_se, ain = "raw",
                                aout = "IRS", on_raw = TRUE)
```

`limmaNorm` *limma::removeBatchEffects (limBE)*

Description

Log2-scaled data should be used as input (on_raw = FALSE).

Usage

```
limmaNorm(se, ain = "log2", aout = "limBE", on_raw = FALSE)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomic dataset
`ain`	String which assay should be used as input
`aout`	String which assay should be used to save normalized data
`on_raw`	Boolean specifying whether normalization should be performed on raw or log2-scaled data

Value

SummarizedExperiment containing the limBE normalized data as assay (on log2 scale)

See Also

`removeBatchEffect()`

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- limmaNorm(tuberculosis_TMT_se, ain = "log2",
                                  aout = "limBE", on_raw = FALSE)
```

`load_data` *Load real-world proteomics data into a SummarizedExperiment*

Description

Load real-world proteomics data into a SummarizedExperiment

Usage

```
load_data(
  data,
  md,
  protein_column = "Protein.IDs",
  gene_column = "Gene.Names",
  ref_samples = NULL,
  batch_column = NULL,
  condition_column = NULL,
  label_column = NULL
)
```

Arguments

`data`	tabular data table with rows = proteins and columns = samples (such as proteinGroups.txt of MaxQuant)
`md`	experimental design table (requires a column named "Column" for the column names of the sample intensities in data)
`protein_column`	name of the column in data containing the protein IDs
`gene_column`	name of the column in data containing the gene names
`ref_samples`	reference samples if TMT experiment provided (names of samples)
`batch_column`	name of the column in md defining the batches
`condition_column`	name of the column in md defining the condition (can still be changed afterwards)
`label_column`	name of the column in md containing simple sample names (for visualization)

Value

SummarizedExperiment object

Examples

```
data_path <- readPRONE_example("tuberculosis_protein_intensities.csv")
md_path <- readPRONE_example("tuberculosis_metadata.csv")
data <- read.csv(data_path)
md <- read.csv(md_path)
md$Column <- stringr::str_replace_all(md$Column, " ", ".")
```

```
ref_samples <- md[md$Group == "ref",]$Column
se <- load_data(data, md, protein_column = "Protein.IDs",
                gene_column = "Gene.names", ref_samples = ref_samples,
                batch_column = "Pool", condition_column = "Group",
                label_column = "Label")
```

`load_spike_data` *Load spike-in proteomics data into a SummarizedExperiment*

Description

Load spike-in proteomics data into a SummarizedExperiment

Usage

```
load_spike_data(
  data,
  md,
  spike_column,
  spike_value,
  spike_concentration,
  protein_column = "Protein.IDs",
  gene_column = "Gene.Names",
  ref_samples = NULL,
  batch_column = NULL,
  condition_column = NULL,
  label_column = NULL
)
```

Arguments

`data`	tabular data table with rows = proteins and columns = samples (such as proteinGroups.txt of MaxQuant)
`md`	experimental design table (requires a column named "Column" for the column names of the sample intensities in data)
`spike_column`	name of the column specifying which proteins are the spike-ins
`spike_value`	String value specifying the spike-in proteins in the spike-in column
`spike_concentration`	name of the column in md defining the spike-in concentration per sample
`protein_column`	name of the column in data containing the protein IDs
`gene_column`	name of the column in data containing the gene names
`ref_samples`	reference samples if TMT experiment provided (names of samples)
`batch_column`	name of the column in md defining the batches

`condition_column`	name of the column in md defining the condition (can still be changed afterwards)
`label_column`	name of the column in md containing simple sample names (for visualization)

Value

SummarizedExperiment object

Examples

```
data_path <- readPRONE_example("Ecoli_human_MaxLFQ_protein_intensities.csv")
md_path <- readPRONE_example("Ecoli_human_MaxLFQ_metadata.csv")
data <- read.csv(data_path)
md <- read.csv(md_path)
mixed <- grepl("Homo sapiens.*Escherichia|Escherichia.*Homo sapiens", data$Fasta.headers)
data <- data[!mixed,]
data$Spiked <- rep("HUMAN", nrow(data))
data$Spiked[grepl("ECOLI", data$Fasta.headers)] <- "ECOLI"
se <- load_spike_data(data, md, spike_column = "Spiked", spike_value = "ECOLI",
        spike_concentration = "Concentration", protein_column = "Protein.IDs",
        gene_column = "Gene.names", ref_samples = NULL, batch_column = NULL,
        condition_column = "Condition", label_column = "Label")
```

`loessCycNorm` *Cyclic Loess Normalization of limma*

Description

Two samples of the data are MA transformed and normalized at a time, and all pairs of samples are iterated through. Log2-scaled data should be taken as input (on_raw = FALSE).

Usage

```
loessCycNorm(se, ain = "log2", aout = "LoessCyc", on_raw = FALSE)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomic dataset
`ain`	String which assay should be used as input
`aout`	String which assay should be used to save normalized data
`on_raw`	Boolean specifying whether normalization should be performed on raw or log2-scaled data

Value

SummarizedExperiment containing the loessCyc normalized data as assay (on log2 scale)

See Also

`normalizeCyclicLoess()`

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- loessCycNorm(tuberculosis_TMT_se, ain = "log2",
                                    aout = "LoessCyc", on_raw = FALSE)
```

`loessFNorm` *Fast Loess Normalization of limma*

Description

Using mean intensities over all the samples as its reference A sample. Log2-scaled data should be used as input (on_raw = FALSE).

Usage

```
loessFNorm(se, ain = "log2", aout = "LoessF", on_raw = FALSE)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomic dataset
`ain`	String which assay should be used as input
`aout`	String which assay should be used to save normalized data
`on_raw`	Boolean specifying whether normalization should be performed on raw or log2-scaled data

Value

SummarizedExperiment containing the LoessF normalized data as assay (on log2 scale)

See Also

`normalizeCyclicLoess()`

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- loessFNorm(tuberculosis_TMT_se, ain = "log2",
                                  aout = "LoessCyc", on_raw = FALSE)
```

`meanNorm` *Mean Normalization*

Description

The intensity of each protein group in a given sample is divided by the mean of the intensities of all protein groups in that sample and then multiplied by the mean of mean of sum of intensities of all protein groups in all samples.

Usage

```
meanNorm(se, ain = "raw", aout = "Mean", on_raw = TRUE)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomic dataset
`ain`	String which assay should be used as input
`aout`	String which assay should be used to save normalized data
`on_raw`	Boolean whether normalized should be performed on raw or log2-scaled data

Value

SummarizedExperiment containing the mean normalized data as assay (on log2 scale)

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- meanNorm(tuberculosis_TMT_se, ain = "raw",
                                aout = "Mean", on_raw = TRUE)
```

`medianAbsDevNorm` *Median Absolute Deviation Normalization*

Description

Substracts the median and divides the data by the median absolute deviation (MAD). Log2-scaled data should be used as input (on_raw = FALSE).

Usage

```
medianAbsDevNorm(se, ain = "log2", aout = "MAD", on_raw = FALSE)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomic dataset
`ain`	String which assay should be used as input
`aout`	String which assay should be used to save normalized data
`on_raw`	Boolean specifying whether normalization should be performed on raw or log2-scale data

Value

SummarizedExperiment containing the MAD normalized data as assay (on log2 scale)

See Also

`performSMADNormalization()`

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- medianAbsDevNorm(tuberculosis_TMT_se, ain = "log2",
                                        aout = "MAD", on_raw = FALSE)
```

`medianNorm` *Median Normalization*

Description

The intensity of each protein group in a given sample is divided by the median of the intensities of all protein groups in that sample and then multiplied by the mean of median of sum of intensities of all protein groups in all samples.

Usage

```
medianNorm(se, ain = "raw", aout = "Median", on_raw = TRUE)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomic dataset
`ain`	String which assay should be used as input
`aout`	String which assay should be used to save normalized data
`on_raw`	Boolean specifying whether normalization should be performed on raw or log2-scaled data

Value

SummarizedExperiment containing the median normalized data as assay (on log2 scale)

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- medianNorm(tuberculosis_TMT_se, ain = "raw",
                       aout = "Median", on_raw = TRUE)
```

`normalize_se` *Normalize SummarizedExperiment object using single normalization methods or specified combinations of normalization methods*

Description

Normalize SummarizedExperiment object using single normalization methods or specified combinations of normalization methods

Usage

```
normalize_se(
  se,
  methods,
  combination_pattern = "_on_",
  on_raw = NULL,
  gamma.0 = 0.1,
  reduce_correlation_by = 1,
  NormicsVSN_quantile = 0.8,
  top_x = 50,
  VSN_quantile = 0.9
)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`methods`	Vector of normalization methods to apply for normalizing the proteomics data of the SummarizedExperiment object (identifier of normalization methods can be retrieved using get_normalization_methods())
`combination_pattern`	String specifying how normalization methods are combined. For instance, methods = c("IRS", "Median_on_IRS"), combination_pattern = "_on_".
`on_raw`	Logical indicating if the normalization should be performed on the raw data or on log2-transformed data. If on_raw = NULL(default), the normalization is performed on the default method specific on_raw setting (suggestion based on publications).

`gamma.0`	Numeric representing the exponent of the weighted density of RobNorm normalization. When the sample size is small, the fitted population of some proteins could be locally trapped such that the variance of those proteins was very small under a large gamma. To avoid this, a small gamma is recommended. When sample size smaller than 40, then set gamma to 0.5 or 0.1.
`reduce_correlation_by`	If the data is too big for the computation of the params, increase this parameter by 2,3,4.... The whole data will still be normalized, but the params are calculated on every second row etc.
`NormicsVSN_quantile`	The quantile that is used for the resistant least trimmed sum of squares regression. A value of 0.8 means focusing on the central 80% of the data, reducing the influence of outliers.
`top_x`	Number of reference proteins extracted for the calculation of parameters
`VSN_quantile`	Numeric of length 1. The quantile that is used for the resistant least trimmed sum of squares regression (see vsn2 lts.quantile)

Value

SummarizedExperiment object with normalized data saved as assays

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- normalize_se(tuberculosis_TMT_se,
          methods = c("IRS_on_GlobalMedian", "IRS_on_Median",
          "limBE_on_NormicsVSN"), on_raw = NULL,
          combination_pattern = "_on_", gamma.0 = 0.1,
          reduce_correlation_by = 1, NormicsVSN_quantile = 0.8, top_x = 50,
          VSN_quantile = 0.9)
```

`normalize_se_combination` *Normalize SummarizedExperiment object using combinations of normalization methods*

Description

Normalize SummarizedExperiment object using combinations of normalization methods

Usage

```
normalize_se_combination(
  se,
  methods,
  ains,
```

```
  on_raw = NULL,
  combination_pattern = "_on_",
  gamma.0 = 0.1,
  reduce_correlation_by = 1,
  NormicsVSN_quantile = 0.8,
  top_x = 50,
  VSN_quantile = 0.9
)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`methods`	Vector of normalization methods to apply for normalizing the proteomics data of the SummarizedExperiment object (identifier of normalization methods can be retrieved using get_normalization_methods())
`ains`	Vector of assays of SummarizedExperiment object to apply the normalization methods (e.g. if you want to perform Median normalization on IRS-normalized data)
`on_raw`	Logical indicating if the normalization should be performed on the raw data or on log2-transformed data. If on_raw = NULL(default), the normalization is performed on the default method specific on_raw setting (suggestion based on publications).
`combination_pattern`	String to give name to combination of methods (e.g. IRS_on_Median –> "_on_")
`gamma.0`	Numeric representing the exponent of the weighted density of RobNorm normalization. When the sample size is small, the fitted population of some proteins could be locally trapped such that the variance of those proteins was very small under a large gamma. To avoid this, a small gamma is recommended. When sample size smaller than 40, then set gamma to 0.5 or 0.1.
`reduce_correlation_by`	If the data is too big for the computation of the params, increase this parameter by 2,3,4.... The whole data will still be normalized, but the params are calculated on every second row etc.
`NormicsVSN_quantile`	The quantile that is used for the resistant least trimmed sum of squares regression. A value of 0.8 means focusing on the central 80% of the data, reducing the influence of outliers.
`top_x`	Number of reference proteins extracted for the calculation of parameters
`VSN_quantile`	Numeric of length 1. The quantile that is used for the resistant least trimmed sum of squares regression. (see vsn2 lts.quantile)

Value

SummarizedExperiment object with normalized data saved as assays

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- normalize_se_combination(tuberculosis_TMT_se,
          methods = c("Median","NormicsVSN"), ains = c("IRS"), on_raw = NULL,
          combination_pattern = "_on_", gamma.0 = 0.1,
          reduce_correlation_by = 1, NormicsVSN_quantile = 0.8, top_x = 50,
          VSN_quantile = 0.9)
```

`normalize_se_single` *Normalize SummarizedExperiment object using different normalization methods*

Description

Normalize SummarizedExperiment object using different normalization methods

Usage

```
normalize_se_single(
  se,
  methods = NULL,
  on_raw = NULL,
  gamma.0 = 0.1,
  reduce_correlation_by = 1,
  NormicsVSN_quantile = 0.8,
  top_x = 50,
  VSN_quantile = 0.9
)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`methods`	Vector of normalization methods to apply for normalizing the proteomics data of the SummarizedExperiment object (identifier of normalization methods can be retrieved using get_normalization_methods())
`on_raw`	Logical indicating if the normalization should be performed on the raw data or on log2-transformed data. If on_raw = NULL(default), the normalization is performed on the default method specific on_raw setting (suggestion based on publications).
`gamma.0`	Numeric representing the exponent of the weighted density of RobNorm normalization. When the sample size is small, the fitted population of some proteins could be locally trapped such that the variance of those proteins was very small under a large gamma. To avoid this, a small gamma is recommended. When sample size smaller than 40, then set gamma to 0.5 or 0.1.

`reduce_correlation_by`	If the data is too big for the computation of the params, increase this parameter by 2,3,4.... The whole data will still be normalized, but the params are calculated on every second row etc.
`NormicsVSN_quantile`	The quantile that is used for the resistant least trimmed sum of squares regression. A value of 0.8 means focusing on the central 80% of the data, reducing the influence of outliers.
`top_x`	Number of reference proteins extracted for the calculation of parameters
`VSN_quantile`	Numeric of length 1. The quantile that is used for the resistant least trimmed sum of squares regression. (see vsn2 lts.quantile)

Value

SummarizedExperiment object with normalized data saved as assays

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- normalize_se_single(tuberculosis_TMT_se,
          methods = c("RobNorm", "Median", "NormicsVSN","VSN"),
          on_raw = NULL, gamma.0 = 0.1, reduce_correlation_by = 1,
          NormicsVSN_quantile = 0.8, top_x = 50, VSN_quantile = 0.9)
```

`normicsNorm` *Normics Normalization (Normics using VSN or using Median)*

Description

Log2-scaled data should be used as input (on_raw = FALSE).

Usage

```
normicsNorm(
  se,
  ain = "raw",
  aout = "NormicsVSN",
  method = "NormicsVSN",
  on_raw = TRUE,
  reduce_correlation_by = 1,
  NormicsVSN_quantile = 0.8,
  TMT_ratio = FALSE,
  top_x = 50
)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomic dataset
`ain`	String which assay should be used as input
`aout`	String which assay should be used to save normalized data
`method`	String specifying the method to use (NORMICS or NORMICSmedian)
`on_raw`	Boolean specifying whether normalization should be performed on raw or log2-scaled data
`reduce_correlation_by`	If the data is too big for the computation of the params, increase this parameter by 2,3,4.... The whole data will still be normalized, but the params are calculated on every second row etc.
`NormicsVSN_quantile`	The quantile that is used for the resistant least trimmed sum of squares regression. A value of 0.8 means focusing on the central 80% of the data, reducing the influence of outliers.
`TMT_ratio`	Indicates if the data involves Tandem Mass Tag (TMT) ratio-based measurements (common in proteomics). If TRUE, the method may handle the data differently.
`top_x`	Number of reference proteins extracted for the calculation of parameters

Value

SummarizedExperiment containing the NormicsVSN/NormicsMedian normalized data as assay (on log2 scale)

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- normicsNorm(tuberculosis_TMT_se, ain = "raw",
                                   aout = "NormicsVSN", method = "NormicsVSN",
                                   on_raw = TRUE)
```

`perform_DEqMS` *Perform DEqMS*

Description

Perform DEqMS

Usage

```
perform_DEqMS(
  fit,
  se,
  DEqMS_PSMs_column = NULL,
  logFC = TRUE,
  logFC_up = 1,
  logFC_down = -1,
  p_adj = TRUE,
  alpha = 0.05
)
```

Arguments

`fit`	eBayes object resulting from perform_limma method
`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`DEqMS_PSMs_column`	String specifying which column name to use for DEqMS (default NULL). Any column of the rowData(se) is accepted.
`logFC`	Boolean specifying whether to apply a logFC threshold (TRUE) or not (FALSE)
`logFC_up`	Upper log2 fold change threshold (dividing into up regulated)
`logFC_down`	Lower log2 fold change threshold (dividing into down regulated)
`p_adj`	Boolean specifying whether to apply a threshold on adjusted p-values (TRUE) or on raw p-values (FALSE)
`alpha`	Threshold for adjusted p-values or p-values

Value

data.table of DE results

`perform_limma` *Fitting a linear model using limma*

Description

Fitting a linear model using limma

Usage

```
perform_limma(
  data,
  condition_vector,
  comparisons,
  covariate = NULL,
```

```
  trend = TRUE,
  robust = TRUE
)
```

Arguments

`data`	Data table of intensities (rows = proteins, cols = samples)
`condition_vector`	Vector of experimental design specifying the condition(s) to compare
`comparisons`	Vector of comparisons that are performed in the DE analysis (from specify_comparisons method)
`covariate`	String specifying which column to include as covariate into limma
`trend`	logical, should an intensity-dependent trend be allowed for the prior variance? If FALSE then the prior variance is constant. Alternatively, trend can be a row-wise numeric vector, which will be used as the covariate for the prior variance.
`robust`	logical, should the estimation of df.prior and var.prior be robustified against outlier sample variances?

Value

eBayes object

`perform_ROTS` *Performing ROTS*

Description

Performing ROTS

Usage

```
perform_ROTS(
  data,
  condition,
  comparisons,
  condition_name,
  coldata,
  logFC = TRUE,
  logFC_up = 1,
  logFC_down = -1,
  p_adj = TRUE,
  alpha = 0.05,
  B = 100,
  K = 500
)
```

Arguments

`data`	Data table of intensities (rows = proteins, cols = samples)
`condition`	Vector of experimental design specifying the condition(s) to compare
`comparisons`	Vector of comparisons that are performed in the DE analysis (from specify_comparisons method)
`condition_name`	String of name of condition in colData
`coldata`	colData of the SummarizedExperiment
`logFC`	Boolean specifying whether to apply a logFC threshold (TRUE) or not (FALSE)
`logFC_up`	Upper log2 fold change threshold (dividing into up regulated)
`logFC_down`	Lower log2 fold change threshold (dividing into down regulated)
`p_adj`	Boolean specifying whether to apply a threshold on adjusted p-values (TRUE) or on raw p-values (FALSE)
`alpha`	Threshold for adjusted p-values or p-values
`B`	Number of bootstrapping for ROTS
`K`	Number of top-ranked features for reproducibility optimization

Value

Data table with DE results

`plot_boxplots` *Plot the distributions of the normalized data as boxplots*

Description

Plot the distributions of the normalized data as boxplots

Usage

```
plot_boxplots(
  se,
  ain = NULL,
  color_by = NULL,
  label_by = NULL,
  facet_norm = TRUE,
  ncol = 3
)
```

Arguments

se	SummarizedExperiment containing all necessary information of the proteomics data set
ain	Vector of strings which assay should be used as input (default NULL). If NULL then all normalization of the se object are plotted next to each other.
color_by	String specifying the column to color the samples (If NULL, the condition column of the SummarizedExperiment object is used. If "No", no color bar added.)
label_by	String specifying the column to label the samples (If NULL, the labels column of the SummarizedExperiment object is used. If "No", no labeling of samples done.)
facet_norm	Boolean specifying whether to facet by normalization methods (default TRUE). If FALSE, list of plots of the different normalized data is returned.
ncol	Number of columns in plot (for faceting)

Value

if facet_norm = TRUE, ggplot object, else list of ggplot objects

Examples

```
data(tuberculosis_TMT_se)
plot_boxplots(tuberculosis_TMT_se, ain = NULL, color_by = NULL, label_by = NULL,
              facet_norm = TRUE, ncol = 3)
plot_boxplots(tuberculosis_TMT_se, ain = c("log2", "IRS_on_RobNorm"), color_by = "Pool",
              label_by = "Label", facet_norm = FALSE)
```

---

plot_condition_overview

Barplot showing the number of samples per condition

Description

Barplot showing the number of samples per condition

Usage

```
plot_condition_overview(se, condition = NULL)
```

Arguments

se	SummarizedExperiment containing all necessary information of the proteomics data set
condition	column name of condition (if NULL, condition saved in SummarizedExperiment will be taken)

Value

ggplot object

Examples

```
data(tuberculosis_TMT_se)
plot_condition_overview(tuberculosis_TMT_se, condition = NULL)
```

`plot_densities` *Plot the densities of the normalized data*

Description

Plot the densities of the normalized data

Usage

```
plot_densities(se, ain = NULL, color_by = NULL, facet_norm = TRUE, ncol = 3)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`ain`	Vector of strings which assay should be used as input (default NULL). If NULL then all normalization of the se object are plotted next to each other.
`color_by`	String specifying the column to color the samples (If NULL, the condition column of the SummarizedExperiment object is used. If "No", no color bar added.)
`facet_norm`	Boolean specifying whether to facet by normalization methods (default TRUE). If FALSE, list of plots of the different normalized data is returned.
`ncol`	Number of columns in plot (for faceting)

Value

if facet_norm = TRUE, ggplot object, else list of ggplot objects

Examples

```
data(tuberculosis_TMT_se)
plot_densities(tuberculosis_TMT_se, ain = NULL, color_by = NULL,
                          facet_norm = TRUE, ncol = 3)
plot_densities(tuberculosis_TMT_se, ain = c("log2", "IRS_on_RobNorm"),
                          color_by = "Label",
               facet_norm = FALSE)
```

`plot_fold_changes_spiked` *Boxplot of log fold changes of spike-in and background proteins for specific normalization methods and comparisons. The ground truth (calculated based on the concentrations of the spike-ins) is shown as a horizontal line.*

Description

Boxplot of log fold changes of spike-in and background proteins for specific normalization methods and comparisons. The ground truth (calculated based on the concentrations of the spike-ins) is shown as a horizontal line.

Usage

```
plot_fold_changes_spiked(se, de_res, condition, ain = NULL, comparisons = NULL)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`de_res`	data table resulting of run_DE
`condition`	column name of condition (if NULL, condition saved in SummarizedExperiment will be taken)
`ain`	Vector of strings of normalization methods to visualize (must be valid normalization methods saved in stats)
`comparisons`	Vector of comparisons (must be valid comparisons saved in stats)

Value

ggplot object

Examples

```
data(spike_in_se)
data(spike_in_de_res)
plot_fold_changes_spiked(spike_in_se, spike_in_de_res,
                         condition = "Condition", ain = NULL,
                         comparisons = NULL)
```

`plot_heatmap` *Plot a heatmap of the sample intensities with optional column annotations for a selection of normalization methods*

Description

Plot a heatmap of the sample intensities with optional column annotations for a selection of normalization methods

Usage

```
plot_heatmap(
  se,
  ain = NULL,
  color_by = c("Group", "Pool"),
  label_by = NULL,
  only_refs = FALSE
)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`ain`	Vector of strings which assay should be used as input (default NULL). If NULL then all normalization of the se object are plotted next to each other.
`color_by`	Vector of strings specifying the columns to color the samples (If NULL, the condition column of the SummarizedExperiment object is used. If "No", no color bars added.)
`label_by`	String specifying the column in the metadata used to label the samples for the UpSet plot
`only_refs`	Logical, if TRUE, only reference samples (ComRef) are included in the plot

Value

list of ggplot objects

Examples

```
data(tuberculosis_TMT_se)
plot_heatmap(tuberculosis_TMT_se, ain = c("log2"), color_by = NULL,
            label_by = NULL, only_refs = FALSE)
```

`plot_heatmap_DE` *Heatmap of DE results*

Description

Heatmap of DE results

Usage

```
plot_heatmap_DE(
  se,
  de_res,
  ain,
  comparison,
  condition = NULL,
  label_by = NULL,
  pvalue_column = "adj.P.Val",
  col_vector = NULL
)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set (including the normalized intensities)
`de_res`	data table resulting of run_DE
`ain`	Vector of strings of normalization methods to visualize (must be valid normalization methods saved in de_res)
`comparison`	String of comparison (must be a valid comparison saved in de_res)
`condition`	column name of condition (if NULL, condition saved in SummarizedExperiment will be taken)
`label_by`	String specifying the column to label the samples (If NULL, the labels column of the SummarizedExperiment object is used. If "No", no labeling of samples done.)
`pvalue_column`	column name of p-values in de_res
`col_vector`	Vector of colors to use for the heatmap. If NULL, default colors are used.

Value

list of ComplexHeatmaps for each method

Examples

```
data(tuberculosis_TMT_se)
data(tuberculosis_TMT_de_res)
plot_heatmap_DE(tuberculosis_TMT_se, tuberculosis_TMT_de_res, ain = NULL,
```

```
comparison = "PTB-HC",
condition = NULL, label_by = NULL,
pvalue_column = "adj.P.Val", col_vector = NULL)
```

`plot_histogram_spiked` *Plot histogram of the spike-in and background protein intensities per condition.*

Description

Plot histogram of the spike-in and background protein intensities per condition.

Usage

```
plot_histogram_spiked(se, condition = NULL)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`condition`	column name of condition (if NULL, condition saved in SummarizedExperiment will be taken)

Value

ggplot object

Examples

```
data(spike_in_se)
plot_histogram_spiked(spike_in_se, condition = NULL)
```

`plot_identified_spiked_proteins` *Plot number of identified spike-in proteins per sample.*

Description

Plot number of identified spike-in proteins per sample.

Usage

```
plot_identified_spiked_proteins(se, color_by = NULL, label_by = NULL)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`color_by`	String specifying the column to color the samples (If NULL, the condition column of the SummarizedExperiment object is used. If "No", no color bar added.)
`label_by`	String specifying the column to label the samples (If NULL, the labels column of the SummarizedExperiment object is used. If "No", no labeling of samples done.)#'

Value

ggplot object

Examples

```
data(spike_in_se)
plot_identified_spiked_proteins(spike_in_se, color_by = NULL,
                                label_by = NULL)
```

`plot_intersection_enrichment` *Intersect top N enrichment terms per normalization method*

Description

Intersect top N enrichment terms per normalization method

Usage

```
plot_intersection_enrichment(
  se,
  de_res,
  ain = NULL,
  comparisons = NULL,
  id_column = "Gene.Names",
  organism = "hsapiens",
  per_comparison = TRUE,
  sources = c("GO:BP", "GO:MF", "GO:CC"),
  top = 10
)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`de_res`	data table resulting of run_DE
`ain`	Vector of strings of normalization methods to visualize (must be valid normalization methods saved in de_res)
`comparisons`	Vector of comparisons (must be valid comparisons saved in de_res)
`id_column`	String specifying the column of the rowData of the SummarizedExperiment object which includes the gene names
`organism`	Organism name (gprofiler parameter)
`per_comparison`	Boolean specifying whether the enrichment analysis should be performed per comparison (TRUE) or on all given comparisons together (FALSE)
`sources`	Vector of data sources to use (gprofiler parameter)
`top`	Number of enrichment terms to extract for each normalization method

Value

list of ggplot objects or single ggplot object

Examples

```
data(tuberculosis_TMT_se)
data(tuberculosis_TMT_de_res)
plot_intersection_enrichment(tuberculosis_TMT_se, tuberculosis_TMT_de_res,
              ain = c("IRS_on_RobNorm", "IRS_on_Median"),
              comparisons = NULL, id_column = "Gene.Names",
              organism = "hsapiens", per_comparison = TRUE,
              sources = c("GO:BP", "GO:MF", "GO:CC"), top = 10)
```

`plot_intragroup_correlation` *Plot intragroup correlation of the normalized data*

Description

Plot intragroup correlation of the normalized data

Usage

```
plot_intragroup_correlation(
  se,
  ain = NULL,
  condition = NULL,
  method = "pearson"
)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`ain`	Vector of strings which assay should be used as input (default NULL). If NULL then all normalization of the se object are plotted next to each other.
`condition`	column name of condition (if NULL, condition saved in SummarizedExperiment will be taken)
`method`	String specifying the method for correlation calculation (pearson, spearman or kendall)

Value

ggplot object (boxplot)

Examples

```
data(tuberculosis_TMT_se)
plot_intragroup_correlation(tuberculosis_TMT_se, ain = NULL,
                            condition = NULL, method = "pearson")
```

`plot_intragroup_PCV` *Plot intragroup pooled coefficient of variation (PCV) of the normalized data*

Description

Plot intragroup pooled coefficient of variation (PCV) of the normalized data

Usage

```
plot_intragroup_PCV(se, ain = NULL, condition = NULL, diff = FALSE)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`ain`	Vector of strings which assay should be used as input (default NULL). If NULL then all normalization of the se object are plotted next to each other.
`condition`	Column name of condition (if NULL, condition saved in SummarizedExperiment will be taken)
`diff`	Boolean indicating whether to visualize the reduction of intragroup variation (PCV) compared to "log" (TRUE) or a normal boxplot of intragroup variation (PCV) for each normalization method (FALSE).

Value

ggplot object (boxplot)

Examples

```
data(tuberculosis_TMT_se)
plot_intragroup_PCV(tuberculosis_TMT_se, ain = NULL,
                            condition = NULL, diff = FALSE)
```

`plot_intragroup_PEV` *Plot intragroup pooled estimate of variance (PEV) of the normalized data*

Description

Plot intragroup pooled estimate of variance (PEV) of the normalized data

Usage

```
plot_intragroup_PEV(se, ain = NULL, condition = NULL, diff = FALSE)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`ain`	Vector of strings which assay should be used as input (default NULL). If NULL then all normalization of the se object are plotted next to each other.
`condition`	column name of condition (if NULL, condition saved in SummarizedExperiment will be taken)
`diff`	Boolean indicating whether to visualize the reduction of intragroup variation (PEV) compared to "log" (TRUE) or a normal boxplot of intragroup variation (PEV) for each normalization method (FALSE).

Value

ggplot object (boxplot)

Examples

```
data(tuberculosis_TMT_se)
plot_intragroup_PEV(tuberculosis_TMT_se, ain = NULL,
                            condition = NULL, diff = FALSE)
```

`plot_intragroup_PMAD` *Plot intragroup pooled median absolute deviation (PMAD) of the normalized data*

Description

Plot intragroup pooled median absolute deviation (PMAD) of the normalized data

Usage

```
plot_intragroup_PMAD(se, ain = NULL, condition = NULL, diff = FALSE)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`ain`	Vector of strings which assay should be used as input (default NULL). If NULL then all normalization of the se object are plotted next to each other.
`condition`	column name of condition (if NULL, condition saved in SummarizedExperiment will be taken)
`diff`	Boolean indicating whether to visualize the reduction of intragroup variation (PMAD) compared to "log" (TRUE) or a normal boxplot of intragroup variation (PMAD) for each normalization method (FALSE).

Value

ggplot object (boxplot)

Examples

```
data(tuberculosis_TMT_se)
plot_intragroup_PMAD(tuberculosis_TMT_se, ain = NULL,
                            condition = NULL, diff = FALSE)
```

`plot_jaccard_heatmap` *Jaccard similarity heatmap of DE proteins of the different normalization methods*

Description

Jaccard similarity heatmap of DE proteins of the different normalization methods

Usage

```
plot_jaccard_heatmap(
  de_res,
  ain = NULL,
  comparisons = NULL,
  plot_type = "single"
)
```

Arguments

`de_res`	data table resulting of run_DE
`ain`	Vector of strings of normalization methods to visualize (must be valid normalization methods saved in de_res)
`comparisons`	Vector of comparisons (must be valid comparisons saved in de_res)
`plot_type`	String indicating whether to plot a single plot per comparison ("single"), facet by comparison ("facet_comp"), or include all comparisons in a single plot ("all")

Value

ggplot object (list of objects if plot_type == "single)

Examples

```
data(tuberculosis_TMT_de_res)
plot_jaccard_heatmap(tuberculosis_TMT_de_res, ain = NULL,
                     comparisons = NULL, plot_type = "all")
```

`plot_logFC_thresholds_spiked` *Line plot of number of true and false positives when applying different logFC thresholds*

Description

Line plot of number of true and false positives when applying different logFC thresholds

Usage

```
plot_logFC_thresholds_spiked(
  se,
  de_res,
  condition,
  ain = NULL,
  comparisons = NULL,
  nrow = 2,
  alpha = 0.05
)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`de_res`	data table resulting of run_DE
`condition`	column name of condition (if NULL, condition saved in SummarizedExperiment will be taken)
`ain`	Vector of strings of normalization methods to visualize (must be valid normalization methods saved in stats)
`comparisons`	Vector of comparisons (must be valid comparisons saved in stats)
`nrow`	number of rows for facet wrap
`alpha`	threshold for adjusted p-values

Value

list of ggplot objects

Examples

```
data(spike_in_se)
data(spike_in_de_res)
plot_logFC_thresholds_spiked(spike_in_se, spike_in_de_res,
                             condition = "Condition", ain = NULL,
                             comparisons = NULL, nrow = 2, alpha = 0.05)
```

`plot_markers_boxplots` *Boxplots of intensities of specific markers*

Description

Boxplots of intensities of specific markers

Usage

```
plot_markers_boxplots(
  se,
  markers,
  ain = NULL,
  id_column = "Protein.IDs",
  color_by = NULL,
  shape_by = NULL,
  facet_norm = TRUE,
  facet_marker = FALSE
)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`markers`	Vector of the IDs of the markers to plot
`ain`	Vector of strings of normalization methods to visualize (must be valid normalization methods saved in de_res)
`id_column`	String specifying the column of the rowData of the SummarizedExperiment object which includes the IDs of the markers
`color_by`	String specifying the column to color the samples (If NULL, the condition column of the SummarizedExperiment object is used. If "No", no color bar added.)
`shape_by`	String specifying the column to shape the samples (If NULL or "No", no shaping of samples is done.)
`facet_norm`	Boolean indicating whether to facet by normalization method (TRUE) or not (FALSE)
`facet_marker`	Boolean indicating whether to facet by comparison (TRUE) or not (FALSE). Only valid if facet_norm = FALSE.

Value

ggplot object

Examples

```
data(tuberculosis_TMT_se)
plot_markers_boxplots(tuberculosis_TMT_se, markers = c("Q7Z7F0", "Q13790"),
                      ain = c("log2"), id_column = "Protein.IDs",
                      color_by = NULL,
                      shape_by = "Pool",
                      facet_norm = FALSE,
                      facet_marker = TRUE)
```

`plot_NA_density` *Plot the intensity distribution of proteins with and without NAs*

Description

Plot the intensity distribution of proteins with and without NAs

Usage

```
plot_NA_density(se)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set

Value

ggplot object

Examples

```
data(tuberculosis_TMT_se)
plot_NA_density(tuberculosis_TMT_se)
```

`plot_NA_frequency` *Plot protein identification overlap (x = identified in number of Samples, y=number of proteins)*

Description

Plot protein identification overlap (x = identified in number of Samples, y=number of proteins)

Usage

```
plot_NA_frequency(se)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set

Value

ggplot object

Examples

```
data(tuberculosis_TMT_se)
plot_NA_frequency(tuberculosis_TMT_se)
```

`plot_NA_heatmap` *Plot heatmap of the NA pattern*

Description

Plot heatmap of the NA pattern

Usage

```
plot_NA_heatmap(
  se,
  color_by = NULL,
  label_by = NULL,
  cluster_samples = TRUE,
  cluster_proteins = TRUE,
  show_row_dend = TRUE,
  show_column_dend = FALSE,
  col_vector = NULL
)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`color_by`	String specifying the column to color the samples (If NULL, the condition column of the SummarizedExperiment object is used. If "No", no color bar added.)
`label_by`	String specifying the column to label the samples (If NULL, the labels column of the SummarizedExperiment object is used. If "No", no labeling of samples done.)
`cluster_samples`	Boolean. TRUE if samples should be clustered, else FALSE.
`cluster_proteins`	Boolean. TRUE if proteins should be clustered, else FALSE.
`show_row_dend`	Boolean. TRUE if row dendrogram should be shown.
`show_column_dend`	Boolean. TRUE if column dendrogram should be shown.
`col_vector`	Vector of colors for the color bar. If NULL, default colors are used.

Value

ComplexHeatmap plot (only showing proteins with at least one missing value)

Examples

```
data(tuberculosis_TMT_se)
plot_NA_heatmap(tuberculosis_TMT_se, color_by = NULL,
                label_by = NULL, cluster_samples = TRUE,
                cluster_proteins = TRUE, show_row_dend = TRUE,
                show_column_dend = FALSE,
                col_vector = NULL)
```

`plot_nr_prot_samples` *Plot number of non-zero proteins per sample*

Description

Plot number of non-zero proteins per sample

Usage

```
plot_nr_prot_samples(se, ain = "raw", color_by = NULL, label_by = NULL)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`ain`	String which data type should be used (default raw)
`color_by`	String specifying the column to color the samples (If NULL, the condition column of the SummarizedExperiment object is used. If "No", no color bar added.)
`label_by`	String specifying the column to label the samples (If NULL, the labels column of the SummarizedExperiment object is used. If "No", no labeling of samples done.)

Value

ggplot object

Examples

```
data(tuberculosis_TMT_se)
plot_nr_prot_samples(tuberculosis_TMT_se, ain="raw", color_by = "Group",
                     label_by = "Label")
```

`plot_overview_DE_bar` *Overview plots of DE results*

Description

Overview plots of DE results

Usage

```
plot_overview_DE_bar(
  de_res,
  ain = NULL,
  comparisons = NULL,
  plot_type = "single"
)
```

Arguments

`de_res`	data table resulting of run_DE
`ain`	Vector of strings of normalization methods to visualize (must be valid normalization methods saved in de_res)
`comparisons`	Vector of comparisons (must be valid comparisons saved in de_res)
`plot_type`	String indicating whether to plot a single plot per comparison ("single"), facet by comparison ("facet_comp"), stack the number of DE per comparison ("stacked"), or stack the number of DE per comparison but facet by up- and down-regulated ("facet_regulation")

Value

list of ggplot objects or single object if plot_type = facet or stacked

Examples

```
data(tuberculosis_TMT_de_res)
plot_overview_DE_bar(tuberculosis_TMT_de_res, ain = NULL, comparisons = NULL,
                     plot_type = "facet_regulation")
```

`plot_overview_DE_tile` *Overview heatmap plot of DE results*

Description

Overview heatmap plot of DE results

Usage

```
plot_overview_DE_tile(de_res, ain = NULL, comparisons = NULL)
```

Arguments

`de_res`	data table resulting of run_DE
`ain`	Vector of strings of normalization methods to visualize (must be valid normalization methods saved in de_res)
`comparisons`	Vector of comparisons (must be valid comparisons saved in de_res)

Value

ggplot object

Examples

```
data(tuberculosis_TMT_de_res)
plot_overview_DE_tile(tuberculosis_TMT_de_res, ain = NULL,
                      comparisons = NULL)
```

`plot_PCA` *PCA plot of the normalized data*

Description

PCA plot of the normalized data

Usage

```
plot_PCA(
  se,
  ain = NULL,
  color_by = NULL,
  label_by = NULL,
  shape_by = NULL,
  facet_norm = TRUE,
```

```
  facet_by = NULL,
  ellipse = FALSE,
  ncol = 3
)
```

Arguments

se	SummarizedExperiment containing all necessary information of the proteomics data set
ain	Vector of strings which assay should be used as input (default NULL). If NULL then all normalization of the se object are plotted next to each other.
color_by	String specifying the column to color the samples (If NULL, the condition column of the SummarizedExperiment object is used. If "No", no color bar added.)
label_by	String specifying the column to label the samples (If NULL, the labels column of the SummarizedExperiment object is used. If "No", no labeling of samples done.)
shape_by	String specifying the column to shape the samples (If NULL or "No", no shaping of samples is done.)
facet_norm	Boolean specifying whether to facet by normalization methods (default TRUE). If FALSE, list of plots of the different normalized data is returned. However, then you can also facet by any column of the metadata.
facet_by	String specifying the column to facet the samples (If facet = FALSE, the plot will not be faceted by the normalization methods, but instead a list of plots of each normalization method is returned. Then, the PCA plot can be faceted by any column of the metadata, for instance by "Batch". If facet_by is NULL or "No", no faceting is performed.)
ellipse	Boolean to indicate if ellipses should be drawn
ncol	Number of columns in plot (for faceting)

Value

if facet_norm = TRUE, ggplot object, else list of ggplot objects

Examples

```
data(tuberculosis_TMT_se)
plot_PCA(tuberculosis_TMT_se, ain = NULL, color_by = NULL, label_by = NULL,
         shape_by = "Pool",
         facet_norm = TRUE, ncol = 3)
plot_PCA(tuberculosis_TMT_se, ain = c("IRS_on_RobNorm"), color_by = "Group",
              label_by = "Label", facet_norm = FALSE, facet_by = "Pool")
```

`plot_profiles_spiked` *Plot profiles of the spike-in and background proteins using the log2 average protein intensities as a function of the different concentrations.*

Description

Plot profiles of the spike-in and background proteins using the log2 average protein intensities as a function of the different concentrations.

Usage

```
plot_profiles_spiked(se, xlab = "Concentration")
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`xlab`	String for the x-label of the plot

Value

ggplot object

Examples

```
data(spike_in_se)
plot_profiles_spiked(spike_in_se, xlab = "Concentration")
```

`plot_pvalues_spiked` *Boxplot of p-values of spike-in and background proteins for specific normalization methods and comparisons. The ground truth (calculated based on the concentrations of the spike-ins) is shown as a horizontal line.*

Description

Boxplot of p-values of spike-in and background proteins for specific normalization methods and comparisons. The ground truth (calculated based on the concentrations of the spike-ins) is shown as a horizontal line.

Usage

```
plot_pvalues_spiked(se, de_res, ain = NULL, comparisons = NULL)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`de_res`	data table resulting of run_DE
`ain`	Vector of strings of normalization methods to visualize (must be valid normalization methods saved in stats)
`comparisons`	Vector of comparisons (must be valid comparisons saved in stats)

Value

ggplot object

Examples

```
data(spike_in_se)
data(spike_in_de_res)
plot_pvalues_spiked(spike_in_se, spike_in_de_res, ain = NULL,
                    comparisons = NULL)
```

`plot_ROC_AUC_spiked` *Plot ROC curve and barplot of AUC values for each method for a specific comparion or for all comparisons*

Description

Plot ROC curve and barplot of AUC values for each method for a specific comparion or for all comparisons

Usage

```
plot_ROC_AUC_spiked(se, de_res, ain = NULL, comparisons = NULL)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`de_res`	data table resulting of run_DE
`ain`	Vector of strings of normalization methods to visualize (must be valid normalization methods saved in stats)
`comparisons`	Vector of comparisons (must be valid comparisons saved in stats)

Value

list of ggplot objects

Examples

```
data(spike_in_se)
data(spike_in_de_res)
plot_ROC_AUC_spiked(spike_in_se, spike_in_de_res)
```

---

plot_stats_spiked_heatmap

Heatmap of performance metrics for spike-in data sets

Description

Heatmap of performance metrics for spike-in data sets

Usage

```
plot_stats_spiked_heatmap(
  stats,
  ain = NULL,
  comparisons = NULL,
  metrics = c("Accuracy", "Precision", "F1Score")
)
```

Arguments

`stats`	data table with multiple metrics of the DE results (resulting of get_spiked_stats_DE)
`ain`	Vector of strings of normalization methods to visualize (must be valid normalization methods saved in stats)
`comparisons`	Vector of comparisons (must be valid comparisons saved in stats)
`metrics`	vector of Strings specifying the metrics (must be colnames of stats)

Value

ggplot object

Examples

```
data(spike_in_se)
data(spike_in_de_res)
stats <- get_spiked_stats_DE(spike_in_se, spike_in_de_res)
plot_stats_spiked_heatmap(stats, ain = NULL, comparisons = NULL,
                          metrics = c("F1Score", "Accuracy"))
```

`plot_tot_int_samples` *Plot total protein intensity per sample*

Description

Plot total protein intensity per sample

Usage

```
plot_tot_int_samples(se, ain = "raw", color_by = NULL, label_by = NULL)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`ain`	String which data type should be used (default raw)
`color_by`	String specifying the column to color the samples (If NULL, the condition column of the SummarizedExperiment object is used. If "No", no color bar added.)
`label_by`	String specifying the column to label the samples (If NULL, the labels column of the SummarizedExperiment object is used. If "No", no labeling of samples done.)

Value

list of a ggplot object and the dataframe of outliers

Examples

```
data(tuberculosis_TMT_se)
plot_tot_int_samples(tuberculosis_TMT_se, ain="raw", color_by = NULL,
                     label_by = NULL)
```

`plot_TP_FP_spiked_bar` *Barplot of true and false positives for specific comparisons and normalization methods*

Description

Barplot of true and false positives for specific comparisons and normalization methods

Usage

```
plot_TP_FP_spiked_bar(stats, ain = NULL, comparisons = NULL)
```

Arguments

`stats`	data table with multiple metrics of the DE results (resulting of get_spiked_stats_DE)
`ain`	Vector of strings of normalization methods to visualize (must be valid normalization methods saved in stats)
`comparisons`	Vector of comparisons (must be valid comparisons saved in stats)

Value

ggplot object (barplot)

Examples

```
data(spike_in_se)
data(spike_in_de_res)
stats <- get_spiked_stats_DE(spike_in_se, spike_in_de_res)
plot_TP_FP_spiked_bar(stats, ain = NULL, comparisons = NULL)
```

`plot_TP_FP_spiked_box` *Boxplot of true and false positives for specific comparisons and normalization methods*

Description

Boxplot of true and false positives for specific comparisons and normalization methods

Usage

```
plot_TP_FP_spiked_box(stats, ain = NULL, comparisons = NULL)
```

Arguments

`stats`	data table with multiple metrics of the DE results (resulting of get_spiked_stats_DE)
`ain`	Vector of strings of normalization methods to visualize (must be valid normalization methods saved in stats)
`comparisons`	Vector of comparisons (must be valid comparisons saved in stats)

Value

ggplot object (barplot)

Examples

```
data(spike_in_se)
data(spike_in_de_res)
stats <- get_spiked_stats_DE(spike_in_se, spike_in_de_res)
plot_TP_FP_spiked_box(stats, ain = NULL, comparisons = NULL)
```

`plot_TP_FP_spiked_scatter` *Scatterplot of true positives and false positives (median with errorbars as Q1, and Q3) for all comparisons*

Description

Scatterplot of true positives and false positives (median with errorbars as Q1, and Q3) for all comparisons

Usage

```
plot_TP_FP_spiked_scatter(stats, ain = NULL, comparisons = NULL)
```

Arguments

`stats`	data table with multiple metrics of the DE results (resulting of get_spiked_stats_DE)
`ain`	Vector of strings of normalization methods to visualize (must be valid normalization methods saved in stats)
`comparisons`	Vector of comparisons (must be valid comparisons saved in stats)

Value

ggplot object

Examples

```
data(spike_in_se)
data(spike_in_de_res)
stats <- get_spiked_stats_DE(spike_in_se, spike_in_de_res)
plot_TP_FP_spiked_scatter(stats, ain = NULL, comparisons = NULL)
```

`plot_upset` *Create an UpSet Plot from SummarizedExperiment Data*

Description

This function generates an UpSet plot from a given SummarizedExperiment object. It allows for the visualization of overlaps between sets defined by a specific column in the metadata. The function supports subsetting to reference samples and customizable color mapping.

Usage

```
plot_upset(
  se,
  color_by = NULL,
  label_by = NULL,
  mb.ratio = c(0.7, 0.3),
  only_refs = FALSE
)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`color_by`	String specifying the column to color the samples (If NULL, the condition column of the SummarizedExperiment object is used.)
`label_by`	String specifying the column in the metadata used to label the samples for the UpSet plot
`mb.ratio`	A numeric vector of length 2, specifying the barplot and matrix area ratios
`only_refs`	Logical, if TRUE, only reference samples (ComRef) are included in the plot

Value

ggplot object

Examples

```
data(tuberculosis_TMT_se)
plot_upset(tuberculosis_TMT_se, color_by = NULL, label_by = NULL,
           mb.ratio = c(0.7, 0.3), only_refs = FALSE)
```

`plot_upset_DE` *Upset plots of DE results of the different normalization methods*

Description

Upset plots of DE results of the different normalization methods

Usage

```
plot_upset_DE(
  de_res,
  ain = NULL,
  comparisons = NULL,
  min_degree = 2,
  plot_type = "single"
)
```

Arguments

`de_res`	data table resulting of run_DE
`ain`	Vector of strings of normalization methods to visualize (must be valid normalization methods saved in de_res)
`comparisons`	Vector of comparisons (must be valid comparisons saved in de_res)
`min_degree`	Minimal degree of an intersection for it to be included
`plot_type`	String indicating whether to plot a single plot per comparison ("single") or stack the number of DE per comparison ("stacked)

Value

list of plots and intersection tables (split by comparison if plot_type == "single)

Examples

```
data(tuberculosis_TMT_de_res)
plot_upset_DE(tuberculosis_TMT_de_res,
              ain = c("IRS_on_RobNorm", "IRS_on_Median"),
              comparisons = NULL, min_degree = 2,
              plot_type = "stacked")
```

`plot_volcano_DE` *Volcano plots of DE results*

Description

Volcano plots of DE results

Usage

```
plot_volcano_DE(
  de_res,
  ain = NULL,
  comparisons = NULL,
  facet_norm = TRUE,
  facet_comparison = FALSE
)
```

Arguments

`de_res`	data table resulting of run_DE
`ain`	Vector of strings of normalization methods to visualize (must be valid normalization methods saved in de_res)
`comparisons`	Vector of comparisons (must be valid comparisons saved in de_res)

`facet_norm`	Boolean indicating whether to facet by normalization method (TRUE) or not (FALSE)
`facet_comparison`	Boolean indicating whether to facet by comparison (TRUE) or not (FALSE). Only valid if facet_norm = FALSE.

Value

list of ggplot objects

Examples

```
data(tuberculosis_TMT_de_res)
plot_volcano_DE(tuberculosis_TMT_de_res, ain = NULL,
                comparisons = NULL, facet_norm = TRUE,
                facet_comparison = FALSE)
```

`quantileNorm` *Quantile Normalization of preprocessCore package.*

Description

Forces distributions of the samples to be the same on the basis of the quantiles of the samples by replacing each protein of a sample with the mean of the corresponding quantile. Log2-scaled data should be taken as input (on_raw = FALSE)

Usage

```
quantileNorm(se, ain = "log2", aout = "Quantile", on_raw = FALSE)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomic dataset
`ain`	String which assay should be used as input
`aout`	String which assay should be used to save normalized data
`on_raw`	Boolean specifying whether normalization should be performed on raw or log2-scaled data

Value

SummarizedExperiment containing the quantile normalized data as assay (on log2 scale)

See Also

`normalize.quantiles()`

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- quantileNorm(tuberculosis_TMT_se, ain = "log2",
                                    aout = "Quantile", on_raw = FALSE)
```

`readPRONE_example` *Helper function to read example data*

Description

Helper function to read example data

Usage

```
readPRONE_example(path = NULL)
```

Arguments

`path`	NULL to get all example data set files, otherwise specify the file name

Value

If path=NULL a character vector with the file names, otherwise the path to the specific file

Examples

```
readPRONE_example()
```

`remove_assays_from_SE` *Remove normalization assays from a SummarizedExperiment object*

Description

Remove normalization assays from a SummarizedExperiment object

Usage

```
remove_assays_from_SE(se, assays_to_remove)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`assays_to_remove`	Character vector of assay names to remove from the SummarizedExperiment object

Value

SummarizedExperiment object with the normalization assays removed

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- remove_assays_from_SE(tuberculosis_TMT_se,
                                  assays_to_remove = c("IRS_on_RobNorm"))
```

`remove_POMA_outliers` *Remove outliers samples detected by the detect_outliers_POMA function*

Description

Remove outliers samples detected by the detect_outliers_POMA function

Usage

```
remove_POMA_outliers(se, poma_res_outliers)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`poma_res_outliers`	Outliers data.table returned by the detect_outliers_POMA function

Value

filtered SummarizedExperiment object

Examples

```
data(tuberculosis_TMT_se)
poma_res <- detect_outliers_POMA(tuberculosis_TMT_se)
tuberculosis_TMT_se <- remove_POMA_outliers(tuberculosis_TMT_se, poma_res$outliers)
```

---

remove_reference_samples

Remove reference samples of SummarizedExperiment object (reference samples specified during loading)

Description

Remove reference samples of SummarizedExperiment object (reference samples specified during loading)

Usage

```
remove_reference_samples(se)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set

Value

filtered SummarizedExperiment object

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- remove_reference_samples(tuberculosis_TMT_se)
```

---

remove_samples_manually

Remove samples with specific value in column manually

Description

Remove samples with specific value in column manually

Usage

```
remove_samples_manually(se, column, values)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`column`	String specifying the column of the meta data (samples with the specified value in this column will be removed)
`values`	Vector of Strings specifying the value for the removal of samples (samples with this value in the specified column will be removed)

Value

filtered SummarizedExperiment object

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- remove_samples_manually(tuberculosis_TMT_se,
                                column = "Label", values = c("1.HC_Pool1"))
```

`rlrMACycNorm` *Cyclic Linear Regression Normalization on MA Transformed Data*

Description

No reference, but MA transformation and normalization of samples is done pairwise between two samples with A = average of two samples and M = difference. The process is iterated through all samples pairs. Log2 data should be taken as input (on_raw = FALSE).

Usage

```
rlrMACycNorm(
  se,
  ain = "log2",
  aout = "RlrMACyc",
  on_raw = FALSE,
  iterations = 3
)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomic dataset
`ain`	String which assay should be used as input
`aout`	String which assay should be used to save normalized data
`on_raw`	Boolean specifying whether normalization should be performed on raw or log2-scaled data
`iterations`	Number of cyclic iterations to be performed

Value

SummarizedExperiment containing the RlrMACyc normalized data as assay (on log2 scale)

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- rlrMACycNorm(tuberculosis_TMT_se, ain = "log2",
                            aout = "RlrMACyc", on_raw = FALSE, iterations=3)
```

`rlrMANorm` *Linear Regression Normalization on MA Transformed Data*

Description

Similar to Rlr, but data are MA transformed before normalization, (A = median sample, M = difference of that sample to A). Log2 data should be taken as input (on_raw = FALSE).

Usage

```
rlrMANorm(se, ain = "log2", aout = "RlrMA", on_raw = FALSE)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomic dataset
`ain`	String which assay should be used as input
`aout`	String which assay should be used to save normalized data
`on_raw`	Boolean specifying whether normalization should be performed on raw or log2-scaled data

Value

SummarizedExperiment containing the RlrMA normalized data as assay (on log2 scale)

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- rlrMANorm(tuberculosis_TMT_se, ain = "log2",
                                  aout = "RlrMA", on_raw = FALSE)
```

`rlrNorm` *Robust Linear Regression Normalization of NormalyzerDE.*

Description

Uses median values over all samples as reference sample to which all the other samples in the data are normalized to. Log2 data should be taken as input (on_raw = FALSE).

Usage

```
rlrNorm(se, ain = "log2", aout = "Rlr", on_raw = FALSE)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomic dataset
`ain`	String which assay should be used as input
`aout`	String which assay should be used to save normalized data
`on_raw`	Boolean specifying whether normalization should be performed on raw or log2-scaled data

Value

SummarizedExperiment containing the rlr normalized data as assay (on log2 scale)

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- rlrNorm(tuberculosis_TMT_se, ain = "log2",
                                           aout = "Rlr", on_raw = FALSE)
```

`RobNorm` *Original RobNorm Function*

Description

To robustly normalize expression data (Author: Meng Wang, https://github.com/mwgrassgreen/RobNorm).

Usage

```
RobNorm(X.0, gamma.0 = 0.5, tol = 10^(-4), step = 200)
```

Arguments

`X.0`	The expression matrix in log scale.
`gamma.0`	The density exponent parameter gamma, in practice, taking gamma.0 = 0.5 or 1.
`tol`	The tolerance for interations (default: 10^(-4)).
`step`	The step limit (default: 50).

Value

Normalized expression data

`robnormNorm` *RobNorm Normalization*

Description

Log2-scaled data should be used as input (on_raw = FALSE).

Usage

```
robnormNorm(se, ain = "log2", aout = "RobNorm", on_raw = FALSE, gamma.0 = 0.1)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomic dataset
`ain`	String which assay should be used as input
`aout`	String which assay should be used to save normalized data
`on_raw`	Boolean specifying whether normalization should be performed on raw or log2-scaled data
`gamma.0`	Numeric representing the exponent of the weighted density. When the sample size is small, the fitted population of some proteins could be locally trapped such that the variance of those proteins was very small under a large gamma. To avoid this, a small gamma is recommended. When sample size smaller than 40, then set gamma to 0.5 or 0.1.

Value

SummarizedExperiment containing the RobNorm normalized data as assay (on log2 scale)

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- robnormNorm(tuberculosis_TMT_se, ain = "log2",
                             aout = "RobNorm", on_raw = FALSE, gamma.0 = 0.1)
```

---

run_DE *Run DE analysis of a selection of normalized data sets*

Description

Run DE analysis of a selection of normalized data sets

Usage

```
run_DE(
  se,
  comparisons,
  ain = NULL,
  condition = NULL,
  DE_method = "limma",
  covariate = NULL,
  logFC = TRUE,
  logFC_up = 1,
  logFC_down = -1,
  p_adj = TRUE,
  alpha = 0.05,
  B = 100,
  K = 500,
  trend = TRUE,
  robust = TRUE,
  DEqMS_PSMs_column = NULL
)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`comparisons`	Vector of comparisons that are performed in the DE analysis (from specify_comparisons method)
`ain`	Vector of strings which assay should be used as input (default NULL). If NULL then all normalization of the se object are plotted next to each other.
`condition`	column name of condition (if NULL, condition saved in SummarizedExperiment will be taken)
`DE_method`	String specifying which DE method should be applied (limma, ROTS, DEqMS)
`covariate`	String specifying which column to include as covariate into limma
`logFC`	Boolean specifying whether to apply a logFC threshold (TRUE) or not (FALSE)
`logFC_up`	Upper log2 fold change threshold (dividing into up regulated)
`logFC_down`	Lower log2 fold change threshold (dividing into down regulated)
`p_adj`	Boolean specifying whether to apply a threshold on adjusted p-values (TRUE) or on raw p-values (FALSE)

`alpha`	Threshold for adjusted p-values or p-values
`B`	Number of bootstrapping for ROTS
`K`	Number of top-ranked features for reproducibility optimization
`trend`	logical, should an intensity-dependent trend be allowed for the prior variance? If FALSE then the prior variance is constant. Alternatively, trend can be a row-wise numeric vector, which will be used as the covariate for the prior variance.
`robust`	logical, should the estimation of df.prior and var.prior be robustified against outlier sample variances?
`DEqMS_PSMs_column`	String specifying which column name to use for DEqMS (default NULL). Any column of the rowData(se) is accepted.

Value

Data table of DE results of selected normalized data sets

Examples

```
data(tuberculosis_TMT_se)
comparisons <- specify_comparisons(tuberculosis_TMT_se, condition = NULL,
                           sep = NULL, control = NULL)
de_res <- run_DE(tuberculosis_TMT_se, comparisons,
              ain = NULL, condition = NULL, DE_method = "limma",
              logFC = TRUE, logFC_up = 1, logFC_down = -1, p_adj = TRUE,
              alpha = 0.05, B = 100, K = 500, trend = TRUE, robust = TRUE)
```

`run_DE_single` *Run DE analysis on a single normalized data set*

Description

Run DE analysis on a single normalized data set

Usage

```
run_DE_single(
  se,
  method,
  comparisons,
  condition = NULL,
  DE_method = "limma",
  covariate = NULL,
  logFC = TRUE,
  logFC_up = 1,
  logFC_down = -1,
  p_adj = TRUE,
```

```
  alpha = 0.05,
  B = 100,
  K = 500,
  trend = TRUE,
  robust = TRUE,
  DEqMS_PSMs_column = NULL
)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`method`	String specifying which assay should be used as input
`comparisons`	Vector of comparisons that are performed in the DE analysis (from specify_comparisons method)
`condition`	column name of condition (if NULL, condition saved in SummarizedExperiment will be taken)
`DE_method`	String specifying which DE method should be applied (limma, ROTS, DEqMS)
`covariate`	String specifying which column to include as covariate into limma
`logFC`	Boolean specifying whether to apply a logFC threshold (TRUE) or not (FALSE)
`logFC_up`	Upper log2 fold change threshold (dividing into up regulated)
`logFC_down`	Lower log2 fold change threshold (dividing into down regulated)
`p_adj`	Boolean specifying whether to apply a threshold on adjusted p-values (TRUE) or on raw p-values (FALSE)
`alpha`	Threshold for adjusted p-values or p-values
`B`	Number of bootstrapping for ROTS
`K`	Number of top-ranked features for reproducibility optimization
`trend`	logical, should an intensity-dependent trend be allowed for the prior variance? If FALSE then the prior variance is constant. Alternatively, trend can be a row-wise numeric vector, which will be used as the covariate for the prior variance.
`robust`	logical, should the estimation of df.prior and var.prior be robustified against outlier sample variances?
`DEqMS_PSMs_column`	String specifying which column name to use for DEqMS (default NULL). Any column of the rowData(se) is accepted.

Value

Data table of DE results

`specify_comparisons` *Create vector of comparisons for DE analysis (either by single condition (sep = NULL) or by combined condition)*

Description

Create vector of comparisons for DE analysis (either by single condition (sep = NULL) or by combined condition)

Usage

```
specify_comparisons(se, condition = NULL, sep = NULL, control = NULL)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`condition`	Column name of condition (if NULL, condition saved in SummarizedExperiment will be taken)
`sep`	Separator that separates both groups in the condition vector (NULL if condition composed only of single group)
`control`	String of control samples (how the control condition is named) (NULL if no control sample)

Value

Vector of comparisons for DE analysis

Examples

```
data(tuberculosis_TMT_se)
comparisons <- specify_comparisons(tuberculosis_TMT_se, condition = NULL,
                                   sep = NULL, control = NULL)
```

`spectraCounteBayes_DEqMS` *Additional function of the DEqMS package*

Description

Additional function of the DEqMS package

Usage

```
spectraCounteBayes_DEqMS(fit, coef_col)
```

Arguments

`fit`	linear model from function perform_limma
`coef_col`	an integer vector indicating the column(s) of fit$coefficients for which the function is to be performed. if not specified, all coefficients are used.

Value

list object

`spike_in_de_res` *Example data.table of DE results of a spike-in proteomics data set*

Description

A `data.table` containing the DE results of the spike_in_se data set (limma, logFC > 1, logFC < -1, p.adj < 0.05)

Usage

```
data(spike_in_de_res)
```

Format

An object of class `data.table` (inherits from `data.frame`) with 7500 rows and 10 columns.

Source

Jürgen Cox, Marco Y. Hein, Christian A. Luber, Igor Paron, Nagarjuna Nagaraj, and Matthias Mann.Accurate Proteome-wide Label-free Quantification by Delayed Normalization and Maximal Peptide Ratio Extraction, Termed MaxLFQ. Molecular & Cellular Proteomics 13.9 (Sept. 2014), pp. 2513–2526. <https://doi.org/10.1074/mcp.M113.031591>.

`spike_in_se` *Example SummarizedExperiment of a spike-in proteomics data set*

Description

A `SummarizedExperiment` containing the raw and log2-scaled data of 301 proteins measured in 20 samples. Due to size restriction, we only included the relevant columns of the original proteinGroups.txt of MaxQuant.

Usage

```
data(spike_in_se)
```

Format

An object of class `SummarizedExperiment` with 1500 rows and 6 columns.

Source

Jürgen Cox, Marco Y. Hein, Christian A. Luber, Igor Paron, Nagarjuna Nagaraj, and Matthias Mann.Accurate Proteome-wide Label-free Quantification by Delayed Normalization and Maximal Peptide Ratio Extraction, Termed MaxLFQ. Molecular & Cellular Proteomics 13.9 (Sept. 2014), pp. 2513–2526. <https://doi.org/10.1074/mcp.M113.031591>.

`subset_SE_by_norm` *Subset SummarizedExperiment object by normalization assays*

Description

Subset SummarizedExperiment object by normalization assays

Usage

```
subset_SE_by_norm(se, ain)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomics data set
`ain`	Character vector of assay names to keep in the SummarizedExperiment object

Value

SummarizedExperiment object with only the selected normalization assays

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- subset_SE_by_norm(tuberculosis_TMT_se,
                        ain = c("raw", "log2", "IRS_on_RobNorm"))
```

`tib_to_DF` *Helper function to transform a tibble to a data table*

Description

Helper function to transform a tibble to a data table

Usage

```
tib_to_DF(expr_data, column_names, row_names)
```

Arguments

`expr_data`	Tibble data frame containing the expression data
`column_names`	Column names of the expression data
`row_names`	Row names of the expression data

Value

Data table containing the expression data

`tmmNorm` *Weighted Trimmed Mean of M Values (TMM) Normalization of edgeR package.*

Description

Raw data should be taken as input (on_raw = TRUE).

Usage

```
tmmNorm(se, ain = "raw", aout = "TMM", on_raw = TRUE)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomic dataset
`ain`	String which assay should be used as input
`aout`	String which assay should be used to save normalized data
`on_raw`	Boolean specifying whether normalization should be performed on raw or log2-scaled data

Value

SummarizedExperiment containing the TMM normalized data as assay (on log2 scale)

See Also

`calcNormFactors()`

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- tmmNorm(tuberculosis_TMT_se, ain = "raw",
                                    aout = "TMM", on_raw = TRUE)
```

`tuberculosis_TMT_de_res` *Example data.table of DE results of a real-world proteomics data set*

Description

A `data.table` containing the DE results of the tuberculosis_TMT_se data set (limma, logFC > 1, logFC < -1, p.adj < 0.05)

Usage

```
data(tuberculosis_TMT_de_res)
```

Format

An object of class `data.table` (inherits from `data.frame`) with 9030 rows and 9 columns.

Source

Biadglegne et al. Mycobacterium tuberculosis Affects Protein and Lipid Content of Circulating Exosomes in Infected Patients Depending on Tuberculosis Disease State. Biomedicines 10.4 (Mar. 2022), p. 783. doi: 10.3390/ biomedicines10040783.

`tuberculosis_TMT_se` *Example SummarizedExperiment of a real-world proteomics data set*

Description

A `SummarizedExperiment` containing the raw and log2-scaled data of 301 proteins measured in 20 samples

Usage

```
data(tuberculosis_TMT_se)
```

Format

An object of class `SummarizedExperiment` with 301 rows and 20 columns.

Source

Biadglegne et al. Mycobacterium tuberculosis Affects Protein and Lipid Content of Circulating Exosomes in Infected Patients Depending on Tuberculosis Disease State. Biomedicines 10.4 (Mar. 2022), p. 783. doi: 10.3390/ biomedicines10040783.

`vsnNorm` *Variance Stabilization Normalization of limma package.*

Description

Raw data should be taken as input (on_raw = TRUE).

Usage

```
vsnNorm(se, ain = "raw", aout = "VSN", on_raw = TRUE, VSN_quantile = 0.9)
```

Arguments

`se`	SummarizedExperiment containing all necessary information of the proteomic dataset
`ain`	String which assay should be used as input
`aout`	String which assay should be used to save normalized data
`on_raw`	Boolean specifying whether normalization should be performed on raw or log2-scaled data
`VSN_quantile`	Numeric of length 1. The quantile that is used for the resistant least trimmed sum of squares regression (see vsn2 lts.quantile)

Value

SummarizedExperiment containing the vsn normalized data as assay (on log2-scale)

See Also

`normalizeVSN()`

Examples

```
data(tuberculosis_TMT_se)
tuberculosis_TMT_se <- vsnNorm(tuberculosis_TMT_se, ain = "raw",
                               aout = "VSN", on_raw = TRUE, VSN_quantile = 0.9)
```

Index