

# Package ‘PRONE’

November 23, 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** 2024-11-22

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

**Maintainer** Lis Arend <[lis.arend@tum.de](mailto:lis.arend@tum.de)>

**Contents**

PRONE-package . . . . .	4
apply_thresholds . . . . .	5
check_DEqMS_parameter . . . . .	6
check_DE_parameters . . . . .	6
check_input_assays . . . . .	7
check_plot_DE_parameters . . . . .	8
check_stats_spiked_DE_parameters . . . . .	9
detect_outliers_POMA . . . . .	9
eigenMSNORM . . . . .	10
export_data . . . . .	11
express_to_DT . . . . .	12
extract_consensus_DE_candidates . . . . .	12
extract_limma_DE . . . . .	13
filter_out_complete_NA_proteins . . . . .	14
filter_out_NA_proteins_by_threshold . . . . .	15
filter_out_proteins_by_ID . . . . .	15
filter_out_proteins_by_value . . . . .	16
get_color_value . . . . .	17
get_complete_dt . . . . .	17
get_complete_pca_dt . . . . .	18
get_condition_value . . . . .	18
get_facet_value . . . . .	19
get_label_value . . . . .	19
get_NA_overview . . . . .	20
get_normalization_methods . . . . .	20
get_overview_DE . . . . .	21
get_proteins_by_value . . . . .	21
get_shape_value . . . . .	22
get_spiked_stats_DE . . . . .	22
globalIntNorm . . . . .	23
globalMeanNorm . . . . .	24

globalMedianNorm . . . . .	25
impute_se . . . . .	25
irsNorm . . . . .	26
limmaNorm . . . . .	27
load_data . . . . .	28
load_spike_data . . . . .	29
loessCycNorm . . . . .	30
loessFNorm . . . . .	31
meanNorm . . . . .	32
medianAbsDevNorm . . . . .	32
medianNorm . . . . .	33
normalize_se . . . . .	34
normalize_se_combination . . . . .	35
normalize_se_single . . . . .	37
normicsNorm . . . . .	38
perform_DEqMS . . . . .	39
perform_limma . . . . .	40
perform_ROTSt . . . . .	41
plot_boxplots . . . . .	42
plot_condition_overview . . . . .	43
plot_densities . . . . .	44
plot_fold_changes_spiked . . . . .	45
plot_heatmap . . . . .	46
plot_heatmap_DE . . . . .	47
plot_histogram_spiked . . . . .	48
plot_identified_spiked_proteins . . . . .	48
plot_intersection_enrichment . . . . .	49
plot_intragroup_correlation . . . . .	50
plot_intragroup_PCV . . . . .	51
plot_intragroup_PEV . . . . .	52
plot_intragroup_PMad . . . . .	53
plot_jaccard_heatmap . . . . .	53
plot_logFC_thresholds_spiked . . . . .	54
plot_markers_boxplots . . . . .	55
plot_NA_density . . . . .	56
plot_NA_frequency . . . . .	57
plot_NA_heatmap . . . . .	58
plot_nr_prot_samples . . . . .	59
plot_overview_DE_bar . . . . .	60
plot_overview_DE_tile . . . . .	61
plot_PCA . . . . .	61
plot_profiles_spiked . . . . .	63
plot_pvalues_spiked . . . . .	63
plot_ROC_AUC_spiked . . . . .	64
plot_stats_spiked_heatmap . . . . .	65
plot_tot_int_samples . . . . .	66
plot_TP_FP_spiked_bar . . . . .	66
plot_TP_FP_spiked_box . . . . .	67

plot_TP_FP_spiked_scatter . . . . .	68
plot_upset . . . . .	68
plot_upset_DE . . . . .	69
plot_volcano_DE . . . . .	70
quantileNorm . . . . .	71
readPRONE_example . . . . .	72
remove_assays_from_SE . . . . .	72
remove_POMA_outliers . . . . .	73
remove_reference_samples . . . . .	74
remove_samples_manually . . . . .	74
rlrMACycNorm . . . . .	75
rlrMANorm . . . . .	76
rlrNorm . . . . .	77
RobNorm . . . . .	77
robnormNorm . . . . .	78
run_DE . . . . .	79
run_DE_single . . . . .	80
specify_comparisons . . . . .	82
spectraCounteBayes_DEqMS . . . . .	82
spike_in_de_res . . . . .	83
spike_in_se . . . . .	83
subset_SE_by_norm . . . . .	84
tib_to_DF . . . . .	85
tmmNorm . . . . .	85
tuberculosis_TMT_de_res . . . . .	86
tuberculosis_TMT_se . . . . .	86
vsnNorm . . . . .	87
<b>Index</b>	<b>88</b>

## 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	<i>Helper function to check whether all given assays are in Summarized-Experiment object</i>
--------------------	--

---

### 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**

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

**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	<i>Export the SummarizedExperiment object, the meta data, and the normalized data.</i>
-------------	--

---

**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	<i>Helper function to transform an expression data frame to a data table</i>
---------------	--

---

**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	<i>Extract consensus DE candidates</i>
---------------------------------	--

---

**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	<i>Extract the DE results from eBayes fit of perform_limma function.</i>
------------------	--

---

**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)
tuberculosis_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**

<code>se</code>	SummarizedExperiment containing all necessary information of the proteomics data set
<code>thr</code>	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**

<code>se</code>	SummarizedExperiment containing all necessary information of the proteomics data set
<code>protein_ids</code>	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	<i>Helper function to get correct value for coloration of plots (color_by parameter)</i>
-----------------	--

---

**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	<i>Function to get a long data table of all intensities of all kind of normalization</i>
-----------------	--

---

**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	<i>Helper function to get correct value for faceting of plots (facet_by parameter)</i>
-----------------	--

---

**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	<i>Helper function to get correct value for sample labeling of plots (label_by parameter)</i>
-----------------	---

---

**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	<i>Helper function to get correct value for shaping of plots (shape_by parameter)</i>
-----------------	---

---

**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	<i>Get performance metrics of DE results of spike-in data set.</i>
---------------------	--

---

**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	<i>Total Intensity Normalization</i>
---------------	--------------------------------------

---

**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)





---

globalMedianNorm	<i>Total Intensity Normalization Using the Median for the Calculation of Scaling Factors</i>
------------------	--

---

### 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	<i>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).</i>
-----------	---

---

### 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



---

load_data	<i>Load real-world proteomics data into a SummarizedExperiment</i>
-----------	--

---

**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 protein-Groups.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	<i>Load spike-in proteomics data into a SummarizedExperiment</i>
-----------------	--

---

### 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 protein-Groups.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. Log<sub>2</sub>-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 log<sub>2</sub>-scaled data

**Value**

SummarizedExperiment containing the loessCyc normalized data as assay (on log<sub>2</sub> scale)



---

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

Subtracts 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	<i>Normalize SummarizedExperiment object using single normalization methods or specified combinations of normalization methods</i>
--------------	--

---

**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 <code>get_normalization_methods()</code> )
combination_pattern	String specifying how normalization methods are combined. For instance, <code>methods = c("IRS", "Median_on_IRS")</code> , <code>combination_pattern = "_on_"</code> .
on_raw	Logical indicating if the normalization should be performed on the raw data or on log2-transformed data. If <code>on_raw = NULL</code> (default), the normalization is performed on the default method specific <code>on_raw</code> 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 <code>get_normalization_methods()</code> )
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 <code>on_raw = NULL</code> (default), the normalization is performed on the default method specific <code>on_raw</code> setting (suggestion based on publications).
combination_pattern	String to give name to combination of methods (e.g. <code>IRS_on_Median</code> → <code>"_on_"</code> )
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 <code>vsn2</code> <code>Its.quantile</code> )

### 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 <code>get_normalization_methods()</code> )
on_raw	Logical indicating if the normalization should be performed on the raw data or on log2-transformed data. If <code>on_raw = NULL</code> (default), the normalization is performed on the default method specific <code>on_raw</code> 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**

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

---

### Description

Performing ROTSt

### Usage

```

perform_ROTSt(
  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	<i>Plot the distributions of the normalized data as boxplots</i>
---------------	--

---

**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	<i>Plot the densities of the normalized data</i>
----------------	--

---

**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	<i>Plot a heatmap of the sample intensities with optional column annotations for a selection of normalization methods</i>
--------------	---

---

### 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	<i>Heatmap of DE results</i>
-----------------	------------------------------

---

**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	<i>Plot intragroup pooled estimate of variance (PEV) of the normalized data</i>
---------------------	---

---

**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\_P MAD *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_P MAD(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_P MAD(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)
```

---

<code>plot_NA_frequency</code>	<i>Plot protein identification overlap (x = identified in number of Samples, y=number of proteins)</i>
--------------------------------	--

---

**Description**

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

**Usage**

```
plot_NA_frequency(se)
```

**Arguments**

<code>se</code>	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

<code>se</code>	SummarizedExperiment containing all necessary information of the proteomics data set
<code>ain</code>	Vector of strings which assay should be used as input (default NULL). If NULL then all normalization of the <code>se</code> object are plotted next to each other.
<code>color_by</code>	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.)
<code>label_by</code>	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.)
<code>shape_by</code>	String specifying the column to shape the samples (If NULL or "No", no shaping of samples is done.)
<code>facet_norm</code>	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.
<code>facet_by</code>	String specifying the column to facet the samples (If <code>facet = FALSE</code> , 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 <code>facet_by</code> is NULL or "No", no faceting is performed.)
<code>ellipse</code>	Boolean to indicate if ellipses should be drawn
<code>ncol</code>	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 comparison 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	<i>Volcano plots of DE results</i>
-----------------	------------------------------------

---

**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



---

rlrNorm	<i>Robust Linear Regression Normalization of NormalizerDE.</i>
---------	--

---

**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**

<code>se</code>	SummarizedExperiment containing all necessary information of the proteomic dataset
<code>ain</code>	String which assay should be used as input
<code>aout</code>	String which assay should be used to save normalized data
<code>on_raw</code>	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	<i>Original RobNorm Function</i>
---------	----------------------------------

---

**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**

<code>X.0</code>	The expression matrix in log scale.
<code>gamma.0</code>	The density exponent parameter gamma, in practice, taking <code>gamma.0 = 0.5</code> or <code>1</code> .
<code>tol</code>	The tolerance for iterations (default: $10^{(-4)}$ ).
<code>step</code>	The step limit (default: <code>50</code> ).

**Value**

Normalized expression data

---

robnormNorm	<i>RobNorm Normalization</i>
-------------	------------------------------

---

**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**

<code>se</code>	SummarizedExperiment containing all necessary information of the proteomic dataset
<code>ain</code>	String which assay should be used as input
<code>aout</code>	String which assay should be used to save normalized data
<code>on_raw</code>	Boolean specifying whether normalization should be performed on raw or log2-scaled data
<code>gamma.0</code>	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 log <sub>2</sub> fold change threshold (dividing into up regulated)
logFC_down	Lower log <sub>2</sub> 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

---

<code>spike_in_de_res</code>	<i>Example data.table of DE results of a spike-in proteomics data set</i>
------------------------------	---

---

**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>>.

---

<code>spike_in_se</code>	<i>Example SummarizedExperiment of a spike-in proteomics data set</i>
--------------------------	---

---

**Description**

A `SummarizedExperiment` containing the raw and  $\log_2$ -scaled data of 301 proteins measured in 20 samples. Due to size restriction, we only included the relevant columns of the original `protein-Groups.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	<i>Subset SummarizedExperiment object by normalization assays</i>
-------------------	---

---

**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	<i>Helper function to transform a tibble to a data table</i>
-----------	--

---

**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	<i>Weighted Trimmed Mean of M Values (TMM) Normalization of edgeR package.</i>
---------	--

---

**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**

Biadlegne 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**

Biadlegne 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**

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

**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

- \* **datasets**
  - spike\_in\_de\_res, [83](#)
  - spike\_in\_se, [83](#)
  - tuberculosis\_TMT\_de\_res, [86](#)
  - tuberculosis\_TMT\_se, [86](#)
- \* **internal**
  - check\_DE\_parameters, [6](#)
  - check\_DEqMS\_parameter, [6](#)
  - check\_input\_assays, [7](#)
  - check\_plot\_DE\_parameters, [8](#)
  - check\_stats\_spiked\_DE\_parameters, [9](#)
  - express\_to\_DT, [12](#)
  - get\_color\_value, [17](#)
  - get\_condition\_value, [18](#)
  - get\_facet\_value, [19](#)
  - get\_label\_value, [19](#)
  - get\_shape\_value, [22](#)
  - PRONE-package, [4](#)
  - RobNorm, [77](#)
  - tib\_to\_DF, [85](#)
- apply\_thresholds, [5](#)
- calcNormFactors, [86](#)
- check\_DE\_parameters, [6](#)
- check\_DEqMS\_parameter, [6](#)
- check\_input\_assays, [7](#)
- check\_plot\_DE\_parameters, [8](#)
- check\_stats\_spiked\_DE\_parameters, [9](#)
- detect\_outliers\_POMA, [9](#)
- eigenMSNorm, [10](#)
- export\_data, [11](#)
- express\_to\_DT, [12](#)
- extract\_consensus\_DE\_candidates, [12](#)
- extract\_limma\_DE, [13](#)
- filter\_out\_complete\_NA\_proteins, [14](#)
- filter\_out\_NA\_proteins\_by\_threshold, [15](#)
- filter\_out\_proteins\_by\_ID, [15](#)
- filter\_out\_proteins\_by\_value, [16](#)
- get\_color\_value, [17](#)
- get\_complete\_dt, [17](#)
- get\_complete\_pca\_dt, [18](#)
- get\_condition\_value, [18](#)
- get\_facet\_value, [19](#)
- get\_label\_value, [19](#)
- get\_NA\_overview, [20](#)
- get\_normalization\_methods, [20](#)
- get\_overview\_DE, [21](#)
- get\_proteins\_by\_value, [21](#)
- get\_shape\_value, [22](#)
- get\_spiked\_stats\_DE, [22](#)
- globalIntNorm, [23](#)
- globalMeanNorm, [24](#)
- globalMedianNorm, [25](#)
- impute\_se, [25](#)
- irsNorm, [26](#)
- limmaNorm, [27](#)
- load\_data, [28](#)
- load\_spike\_data, [29](#)
- loessCycNorm, [30](#)
- loessFNorm, [31](#)
- meanNorm, [32](#)
- medianAbsDevNorm, [32](#)
- medianNorm, [33](#)
- normalize.quantiles, [71](#)
- normalize\_se, [34](#)
- normalize\_se\_combination, [35](#)
- normalize\_se\_single, [37](#)
- normalizeCyclicLoess, [37](#)
- normalizeVSN, [87](#)
- normicsNorm, [38](#)



perform\_DEqMS, 39  
perform\_limma, 40  
perform\_ROTTS, 41  
performSMADNormalization, 33  
plot\_boxplots, 42  
plot\_condition\_overview, 43  
plot\_densities, 44  
plot\_fold\_changes\_spiked, 45  
plot\_heatmap, 46  
plot\_heatmap\_DE, 47  
plot\_histogram\_spiked, 48  
plot\_identified\_spiked\_proteins, 48  
plot\_intersection\_enrichment, 49  
plot\_intragroup\_correlation, 50  
plot\_intragroup\_PCV, 51  
plot\_intragroup\_PEV, 52  
plot\_intragroup\_PPAD, 53  
plot\_jaccard\_heatmap, 53  
plot\_logFC\_thresholds\_spiked, 54  
plot\_markers\_boxplots, 55  
plot\_NA\_density, 56  
plot\_NA\_frequency, 57  
plot\_NA\_heatmap, 58  
plot\_nr\_prot\_samples, 59  
plot\_overview\_DE\_bar, 60  
plot\_overview\_DE\_tile, 61  
plot\_PCA, 61  
plot\_profiles\_spiked, 63  
plot\_pvalues\_spiked, 63  
plot\_ROC\_AUC\_spiked, 64  
plot\_stats\_spiked\_heatmap, 65  
plot\_tot\_int\_samples, 66  
plot\_TP\_FP\_spiked\_bar, 66  
plot\_TP\_FP\_spiked\_box, 67  
plot\_TP\_FP\_spiked\_scatter, 68  
plot\_upset, 68  
plot\_upset\_DE, 69  
plot\_volcano\_DE, 70  
PRONE (PRONE-package), 4  
PRONE-package, 4  
  
quantileNorm, 71  
  
readPRONE\_example, 72  
remove\_assays\_from\_SE, 72  
remove\_POMA\_outliers, 73  
remove\_reference\_samples, 74  
remove\_samples\_manually, 74  
removeBatchEffect, 27  
  
rlrMACycNorm, 75  
rlrMANorm, 76  
rlrNorm, 77  
RobNorm, 77  
robnormNorm, 78  
run\_DE, 79  
run\_DE\_single, 80  
  
specify\_comparisons, 82  
spectraCounteBayes\_DEqMS, 82  
spike\_in\_de\_res, 83  
spike\_in\_se, 83  
subset\_SE\_by\_norm, 84  
  
tib\_to\_DF, 85  
tmmNorm, 85  
tuberculosis\_TMT\_de\_res, 86  
tuberculosis\_TMT\_se, 86  
  
vsNorm, 87