Package 'ProtE'

January 20, 2025

Type Package

Title Processing Proteomics Data, Statistical Analysis and Visualization

Version 1.0.1

Maintainer Theodoros Margelos <ted.margelos02@gmail.com>

Description The 'Proteomics Eye' ('ProtE') offers a comprehensive and intuitive framework for the univariate analysis of label-free proteomics data. By integrating essential data wrangling and processing steps into a single function, 'ProtE' streamlines pairwise statistical comparisons for categorical variables. It provides quality checks and generates publication-ready visualizations, enabling efficient and robust data analysis. 'ProtE' is compatible with proteomics data outputs from 'MaxQuant' (Cox & Mann, (2008) <doi:10.1038/nbt.1511>), 'DIA-NN' (Demichev et al., (2020) <doi:10.1038/s41592-019-0638-x>), and 'Proteome Discoverer' (Thermo Fisher Scientific, version 2.5). The package leverages 'ggplot2' for visualization (Wickham, (2016) <doi:10.1007/978-3-319-24277-4>) and 'limma' for statistical analysis (Ritchie et al., (2015) <doi:10.1093/nar/gkv007>).

License MIT + file LICENSE

Encoding UTF-8

Imports dplyr, circlize, vegan, UniProt.ws, stringr, missRanger, vsn, car, openxlsx, tidyr, broom, reshape2, ggpubr, ggplot2, VIM, forcats, grDevices, grid, limma, ComplexHeatmap,

Suggests BiocManager, rmarkdown, knitr

URL https://github.com/theomargel/ProtE

BugReports https://github.com/theomargel/ProtE/issues

RoxygenNote 7.3.2

Config/testthat/edition 3

Depends R (>= 2.10)

VignetteBuilder knitr

Language en-US

NeedsCompilation no

Author Theodoros Margelos [aut, cre, cph], Rafael Stroggilos [ctb, cph]

Repository CRAN

Date/Publication 2025-01-15 18:40:02 UTC

Contents

`dianno` *DIA NN proteomics data analysis*

Description

Processes the DIA-NN proteomics output and performs exploratory statistical analysis for a single categorical variable. Accepts as input either of the files pg_matrix.tsv or unique_gene_matrix.tsv.

Usage

```
dianno(
  file,
  group_names,
  samples_per_group,
  normalization = FALSE,
  imputation = FALSE,
  global_filtering = TRUE,
  independent = TRUE,
  filtering_value = 50,
  parametric = FALSE,
  significance = "p",
  description = FALSE
)
```

Arguments

`file`	The whole path to the DIA-NN pg_matrix.tsv file (or alternatively, to the unique_genes_matrix.tsv file). The folders in the file path must be separated either with the forward slashes (/), or with the double backslashes (\). See the example for inserting correctly the file path.
`group_names`	A character vector specifying group names. The order of the names should align with the order of the sample groups in the input tsv file.
`samples_per_group`	A numerical vector giving the number of samples in each group. The order of the numbers should align with the order of the names in group_names.

`normalization` The specific method for normalizing the data.By default it is set to FALSE. As DIA-NN output has already been normalized with the MaxLFQ algorithm, we suggest to be cautious if you select any method. Options are FALSE for no normalization of the data, "log2" for a simple log2 transformation, "Quantile" for a quantiles based normalization and "Cyclic_Loess" for a Cyclic Loess normalization of the log2 data, "median" for a median one, "TIC" for Total Ion Current normalization, "VSN" for Variance Stabilizing Normalization and "PPM" for Parts per Million transformation of the data.

`imputation` Imputes all remaining missing values. Available methods: "LOD" for assigning the dataset's Limit Of Detection (lowest protein intensity identified), "LOD/2", "Gaussian_LOD" for selecting random values from the normal distribution around LOD with sd= 0.2*LOD, "zeros" for simply assigning 0 to MVs, mean" for replacing missing values with the mean of each protein across the entire dataset, "kNN" for a k-nearest neighbors imputation using 5 neighbors (from the package VIM) and "missRanger" for a random forest based imputation using predictive mean matching (from the package missRanger). By default it is set to FALSE (skips imputation).

`global_filtering` TRUE/FALSE. If TRUE, the per-protein percentage of missing values will be calculated across the entire dataset. If FALSE, it will be calculated separately for each group, allowing proteins to remain in the analysis if they meet the criteria within any group. By default it is set to TRUE.

`independent` TRUE/FALSE If TRUE, the samples come from different populations, if FALSE they come from the same population (Dependent samples). By default, it is set to TRUE. If set to FALSE, the numbers given in the samples_per_group param must be equal to each other.

`filtering_value` The maximum allowable percentage of missing values for a protein. Proteins with missing values exceeding this percentage will be excluded from the analysis. By default it is set to 50.

`parametric` TRUE/FALSE. Specifies the statistical tests that will be taken into account for creating the PCA plots and heatmap. By default it is set to FALSE (non-parametric).

`significance` "p" or "BH" Specifies which of the p-values (nominal vs BH adjusted for multiple hypothesis) will be taken into account for creating the PCA plots and the heatmap. By default it is set to "p" (nominal p-value).

`description` TRUE/FALSE. If TRUE, establishes connection to the Uniprot database (via the Uniprot.ws package) and adds the "Description" annotation in the data. This option requires protein Accession IDs and is thus applicable only to the pg.matrix file. It requires also internet access. By default it is set to FALSE (No description fetching).

Value

Returns the complete output of the exploratory analysis: i) The processed, or filtered/normalized data ii) Statistical output containing results for the parametric (limma+ANOVA) and non-parametric tests (Wilcoxon_p_+Kruskal-Wallis+PERMANOVA), along with statistical tests for heteroscedasticity, iii) Quality metrics for the input samples iv) QC plots and exploratory visualizations.

Examples

```
#Example of running the function with paths for two groups.
# The file path is a placeholder, replace it with an actual file.

jittered.pg_matrix.tsv <- system.file("extdata", "report.pg_matrix.tsv", package = "ProtE")
dianno(file = jittered.pg_matrix.tsv,
       group_names = c("Healthy","Control"),
       samples_per_group = c(5,5), filtering_value = 80)
```

`maximum_quantum` *MaxQuant proteomics data analysis*

Description

Processes the MaxQuant proteomics dataset and performs exploratory statistical analysis for a single categorical variable. Accepts as input the ProteinGroups.txt file.

Usage

```
maximum_quantum(
  file,
  group_names,
  samples_per_group,
  imputation = FALSE,
  global_filtering = TRUE,
  independent = TRUE,
  filtering_value = 50,
  normalization = FALSE,
  parametric = FALSE,
  significance = "p"
)
```

Arguments

`file`	The whole path to the MaxQuant ProteinGroups.txt file. The folders in the file path must be separated either with the forward slashes (/), or with the double backslashes (\). See the example for inserting correctly the file path.
`group_names`	A character vector specifying group names. The order of the names should align with the order of the sample groups in the input tsv file.
`samples_per_group`	A numerical vector giving the number of samples in each group. The order of the numbers should align with the order of the names in group_names.
`imputation`	Imputes all remaining missing values. Available methods: "LOD" for assigning the dataset's Limit Of Detection (lowest protein intensity identified), "LOD/2", "Gaussian_LOD" for selecting random values from the normal distribution around

LOD with sd= 0.2*LOD, "zeros" for simply assigning 0 to MVs, mean" for replacing missing values with the mean of each protein across the entire dataset, "kNN" for a k-nearest neighbors imputation using 5 neighbors (from the package VIM) and "missRanger" for a random forest based imputation using predictive mean matching (from the package missRanger). By default it is set to FALSE (skips imputation).

`global_filtering`
: TRUE/FALSE. If TRUE, the per-protein percentage of missing values will be calculated across the entire dataset. If FALSE, it will be calculated separately for each group, allowing proteins to remain in the analysis if they meet the criteria within any group. By default it is set to TRUE.

`independent`
: TRUE/FALSE If TRUE, the samples come from different populations, if FALSE they come from the same population (Dependent samples). By default, it is set to TRUE. If set to FALSE, the numbers given in the samples_per_group param must be equal to each other.

`filtering_value`
: The maximum allowable percentage of missing values for a protein. Proteins with missing values exceeding this percentage will be excluded from the analysis. By default it is set to 50.

`normalization`
: The specific method for normalizing the data.By default it is set to FALSE. Options are FALSE for no normalization of the data, "log2" for a simple log2 transformation, "Quantile" for a quantiles based normalization and "Cyclic_Loess" for a Cyclic Loess normalization of the log2 data, "median" for a median one, "TIC" for Total Ion Current normalization, "VSN" for Variance Stabilizing Normalization and "PPM" for Parts per Million transformation of the data.

`parametric`
: TRUE/FALSE. Specifies the statistical tests that will be taken into account for creating the PCA plots and heatmap. By default it is set to FALSE (non-parametric).

`significance`
: "p" or "BH" Specifies which of the p-values (nominal vs BH adjusted for multiple hypothesis) will be taken into account for creating the PCA plots and the heatmap. By default it is set to "p" (nominal p-value).

Value

Returns the complete output of the exploratory analysis: i) The processed, or filtered/normalized data ii) Statistical output containing results for the parametric (limma+ANOVA) and non-parametric tests (Wilcoxon+Kruskal-Wallis+PERMANOVA), along with statistical tests for heteroscedasticity, iii) Quality metrics for the input samples iv) QC plots and exploratory visualizations.

Examples

```
#Example of running the function with paths for two groups.
# The file path is a placeholder, replace it with an actual file.

proteinGroups.txt <- system.file("extdata", "proteinGroups.txt", package = "ProtE")
maximum_quantum(file = proteinGroups.txt,
       group_names = c("Healthy","Control"),
       samples_per_group = c(4,4), filtering_value = 80)
```

pd_multi *Proteome Discoverer (PD) multiple-files' proteomic analysis*

Description

Takes as input Proteomics Data (output of PD) in the format of a single file per sample and creates a master table with the Protein names and Abundance values. Then it performs exploratory data analysis, providing different options for data manipulation (normalization, filtering based on the missing values and imputation) It then proceeds to perform statistical analysis, while creating exploratory plots such as relative log expression boxplots and violin plots, heatmaps and PCA plots.

Usage

```
pd_multi(
  ...,
  imputation = FALSE,
  global_filtering = TRUE,
  independent = TRUE,
  filtering_value = 50,
  bugs = 0,
  normalization = FALSE,
  parametric = FALSE,
  significance = "p",
  description = FALSE
)
```

Arguments

`...`	The specific path to the folder where the samples from each group are located. They are passed as unnamed arguments via "...". Attention: Ensure paths use '/' as a directory separator.
`imputation`	Imputes all remaining missing values. Available methods: "LOD" for assigning the dataset's Limit Of Detection (lowest protein intensity identified), "LOD/2", "Gaussian_LOD" for selecting random values from the normal distribution around LOD with sd= 0.2*LOD, "zeros" for simply assigning 0 to MVs, mean" for replacing missing values with the mean of each protein across the entire dataset, "kNN" for a k-nearest neighbors imputation using 5 neighbors (from the package VIM) and "missRanger" for a random forest based imputation using predictive mean matching (from the package missRanger). By default it is set to FALSE (skips imputation).
`global_filtering`	TRUE/FALSE. If TRUE, the per-protein percentage of missing values will be calculated across the entire dataset. If FALSE, it will be calculated separately for each group, allowing proteins to remain in the analysis if they meet the criteria within any group. By default it is set to TRUE.

independent TRUE/FALSE If TRUE, the samples come from different populations, if FALSE they come from the same population (Dependent samples). By default, it is set to TRUE. If set to FALSE, the numbers given in the samples_per_group param must be equal to each other.

filtering_value The maximum allowable percentage of missing values for a protein. Proteins with missing values exceeding this percentage will be excluded from the analysis. By default it is set to 50.

bugs Either 0 to treat Proteome Discoverer bugs as Zeros (0) or "average" to convert them into the average of the protein between the samples. By default, it is set to 0. Bugs are referred to to the proteins with empty values inside a single-file analysis

normalization The specific method for normalizing the data.By default it is set to FALSE. Options are FALSE for no normalization of the data, "log2" for a simple log2 transformation, "Quantile" for a quantiles based normalization and "Cyclic_Loess" for a Cyclic Loess normalization of the log2 data, "median" for a median one, "TIC" for Total Ion Current normalization, "VSN" for Variance Stabilizing Normalization and "PPM" for Parts per Million transformation of the data.

parametric TRUE/FALSE. Specifies the statistical tests that will be taken into account for creating the PCA plots and heatmap. By default it is set to FALSE (non-parametric).

significance "p" or "BH" Specifies which of the p-values (nominal vs BH adjusted for multiple hypothesis) will be taken into account for creating the PCA plots and the heatmap. By default it is set to "p" (nominal p-value).

description TRUE/FALSE. If TRUE, establishes connection to the Uniprot database (via the Uniprot.ws package) and adds the "Description" annotation in the data. This option requires protein Accession IDs and is thus applicable only to the pg.matrix file. It requires also internet access. By default it is set to FALSE (No description fetching).

Value

Excel files with the proteomic values that are optionally processed, via normalization, imputation and filtering of proteins with a selected percentage of missing values. The result of the processing is visualized with an Protein Rank Abundance plot. PCA plots for all groups and for just their significant correlations are created. Furthermore violin and boxplots for the proteins of each sample is created and a heatmap for the significant proteins.

Examples

```
#Example of running the function with paths for three groups.

T1_path <- system.file("extdata", "PDexports(multiple_files)",
 "T1_BLCA", package = "ProtE")
T2_path <- system.file("extdata", "PDexports(multiple_files)",
"T2_BLCA", package = "ProtE")

pd_multi(T1_path, T2_path,
         normalization = FALSE,
```

```
global_filtering = TRUE,  imputation = FALSE,
independent = TRUE)
```

---

pd_single *Proteome Discoverer proteomic data analysis*

Description

Processes the ProteomeDiscoverer proteomics dataset and performs exploratory statistical analysis for a single categorical variable. Accepts as input a ProteomeDiscoverer generated .xlsx file.

Usage

```
pd_single(
  file,
  group_names,
  samples_per_group,
  imputation = FALSE,
  global_filtering = TRUE,
  independent = TRUE,
  filtering_value = 50,
  normalization = FALSE,
  parametric = FALSE,
  significance = "p",
  description = FALSE
)
```

Arguments

file	The whole path to the ProteomeDiscoverer .xlsx file. Ensure that the folders in the path are separated either with the forward slashes (/), or with the double backslashes (\). See the example for inserting correctly the file path.
group_names	A character vector specifying group names. The order of the names should align with the order of the sample groups in the input tsv file.
samples_per_group	A numerical vector giving the number of samples in each group. The order of the numbers should align with the order of the names in group_names.
imputation	Imputes all remaining missing values. Available methods: "LOD" for assigning the dataset's Limit Of Detection (lowest protein intensity identified), "LOD/2", "Gaussian_LOD" for selecting random values from the normal distribution around LOD with sd= 0.2*LOD, "zeros" for simply assigning 0 to MVs, mean" for replacing missing values with the mean of each protein across the entire dataset, "kNN" for a k-nearest neighbors imputation using 5 neighbors (from the package VIM) and "missRanger" for a random forest based imputation using predictive mean matching (from the package missRanger). By default it is set to FALSE (skips imputation).

`global_filtering`
: TRUE/FALSE. If TRUE, the per-protein percentage of missing values will be calculated across the entire dataset. If FALSE, it will be calculated separately for each group, allowing proteins to remain in the analysis if they meet the criteria within any group. By default it is set to TRUE.

`independent`
: TRUE/FALSE If TRUE, the samples come from different populations, if FALSE they come from the same population (Dependent samples). By default, it is set to TRUE. If set to FALSE, the numbers given in the samples_per_group param must be equal to each other.

`filtering_value`
: The maximum allowable percentage of missing values for a protein. Proteins with missing values exceeding this percentage will be excluded from the analysis. By default it is set to 50.

`normalization`
: The specific method for normalizing the data.By default it is set to FALSE. Options are FALSE for no normalization of the data, "log2" for a simple log2 transformation, "Quantile" for a quantiles based normalization and "Cyclic_Loess" for a Cyclic Loess normalization of the log2 data, "median" for a median one, "TIC" for Total Ion Current normalization, "VSN" for Variance Stabilizing Normalization and "PPM" for Parts per Million transformation of the data.

`parametric`
: TRUE/FALSE. Specifies the statistical tests that will be taken into account for creating the PCA plots and heatmap. By default it is set to FALSE (non-parametric).

`significance`
: "p" or "BH" Specifies which of the p-values (nominal vs BH adjusted for multiple hypothesis) will be taken into account for creating the PCA plots and the heatmap. By default it is set to "p" (nominal p-value).

`description`
: TRUE/FALSE. If TRUE, establishes connection to the Uniprot database (via the Uniprot.ws package) and adds the "Description" annotation in the data. This option requires protein Accession IDs and is thus applicable only to the pg.matrix file. It requires also internet access. By default it is set to FALSE (No description fetching).

Value

Returns the complete output of the exploratory analysis: i) The processed, or filtered/normalized data ii) Statistical output containing results for the parametric (limma+ANOVA) and non-parametric tests (Wilcoxon+Kruskal-Wallis+PERMANOVA), along with statistical tests for heteroscedasticity, iii) Quality metrics for the input samples iv) QC plots and exploratory visualizations.

Examples

```
#Example of running the function with paths for two groups.
# The file path is a placeholder, replace it with an actual file.

PDconsesus_file.xlsx <- system.file("extdata", "PDconsesus_file.xlsx", package = "ProtE")
pd_single(file = PDconsesus_file.xlsx,
       group_names = c("Healthy","Control"),
       samples_per_group = c(4,4), filtering_value = 80)
```

Index