Package ‘POMA’

April 10, 2023

Title Tools for Omics Data Analysis

Version 1.8.0

Description A reproducible and easy-to-use toolkit for visualization, pre-processing, exploration, and statistical analysis of omics datasets. The main aim of POMA is to enable a flexible data cleaning and
statistical analysis processes in one comprehensible and user-friendly R package. This package has a Shiny app
version called POMAShiny that implements all POMA functions. See https://github.com/pcastellanoescuder/POMAShiny.
See Castellano-Escuder P, González-Domínguez R, Carmona-Pontaque F, et al. (2021) <doi:10.1371/journal.pcbi.1009148>
for more details.

License GPL-3

Encoding UTF-8

LazyData true

biocViews BatchEffect, Classification, Clustering, DecisionTree, DimensionReduction, MultidimensionalScaling, Normalization, Preprocessing, PrincipalComponent, Regression, RNASeq, Software, StatisticalMethod, Visualization

Imports broom, caret, ComplexHeatmap, dbscan, dplyr, DESeq2, ggplot2, ggrepel, glasso (>= 1.11), glmnet, impute, limma, magrittr, mixOmics, randomForest, RankProd (>= 3.14), rmarkdown, SummarizedExperiment, tibble, tidyr, uwot, vegan

Suggests BiocStyle, covr, ggraph, knitr, patchwork, plotly, tidyverse, testthat (>= 2.3.2)

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

Depends R (>= 4.0)

VignetteBuilder knitr

URL https://github.com/pcastellanoescuder/POMA

BugReports https://github.com/pcastellanoescuder/POMA/issues

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

git_branch RELEASE_3_16

git_last_commit 3a1c9c6

git_last_commit_date 2022-11-01

Date/Publication 2023-04-10

Author Pol Castellano-Escuder [aut, cre]
(<https://orcid.org/0000-0001-6466-877X>)

Maintainer Pol Castellano-Escuder <polcaes@gmail.com>

R topics documented:

cor_pmat 3
flattenCorrMatrix 3
PomaBoxplots 4
PomaClust 5
PomaCorr 6
PomaDensity 7
PomaDESeq 8
PomaEDA 9
PomaHeatmap 10
PomaImpute 11
PomaLasso 12
PomaLimma 14
PomaMultivariate 15
PomaNorm 17
PomaOddsRatio 18
PomaOutliers 19
PomaPCR 20
PomaRandForest 21
PomaRankProd 22
PomaSummarizedExperiment 24
PomaUMAP 25
PomaUnivariate 26
PomaVolcano 27
st000284 29
st000336 30

Index **31**

`cor_pmat` *Correlation P-Values*

Description

Compute correlation p-values.

Usage

```
cor_pmat(x, method)
```

Arguments

`x` A data matrix.

`method` Character indicating which correlation coefficient has to be computed. Options are "pearson" (default), "kendall" and "spearman".

`flattenCorrMatrix` *Flatten Correlation Matrix*

Description

Flatten Correlation Matrix

Usage

```
flattenCorrMatrix(cormat, pmat)
```

Arguments

`cormat` Output from `cor`.

`pmat` Output from `cor_pmat`.

PomaBoxplots *Classical Boxplots*

Description

PomaBoxplots() generates a boxplot for subjects or features. This boxplot can help in the comparison between pre and post normalized data and in the "validation" of the normalization process.

Usage

```
PomaBoxplots(
  data,
  group = "samples",
  jitter = FALSE,
  feature_name = NULL,
  label_size = 10,
  legend_position = "bottom"
)
```

Arguments

`data` A SummarizedExperiment object.

`group` Groupping factor for the plot. Options are "samples" and "features". Option "samples" (default) will create a boxplot for each sample and option "features" will create a boxplot of each variable.

`jitter` Logical. If it's TRUE (default), the boxplot will show all points.

`feature_name` A vector with the name/s of feature/s to plot. If it's NULL (default) a boxplot of all features will be created.

`label_size` Numeric indicating the size of x-axis labels.

`legend_position` Character indicating the legend position. Options are "none", "top", "bottom", "left", and "right".

Value

A ggplot2 object.

Author(s)

Pol Castellano-Escuder

Examples

```
data("st000284")

# samples
PomaBoxplots(st000284)

# features
PomaBoxplots(st000284, group = "features")

# concrete features
PomaBoxplots(st000284, group = "features",
             feature_name = c("ornithine", "orotate"))
```

PomaClust *Cluster Analysis*

Description

This function performs a classical multidimensional scaling (MDS) using all features in the data and computes a cluster analysis for k clusters. Then, the calculated clusters will be represented on a MDS plot.

Usage

```
PomaClust(
  data,
  method = "euclidean",
  k = NA,
  k_max = 15,
  show_clusters = TRUE,
  labels = FALSE,
  show_group = FALSE
)
```

Arguments

data	A SummarizedExperiment object.
method	Distance measure method to perform MDS. Options are "euclidean", "maximum", "manhattan", "canberra" and "minkowski". See ?dist().
k	Number of clusters (default is NA). The optimum number of clusters will be used by default.
k_max	Number of clusters among which the optimal one will be selected.
show_clusters	Logical indicating if clusters should be plotted or not. If this parameter is set to FALSE the resultant plot will be a classical 2-dimension MDS plot.
labels	Logical indicating if sample names should be plotted or not.
show_group	Logical indicating if the original sample group from target should be plotted instead of sample ID or not. Only works if labels is set to TRUE.

Value

A list with the results.

Author(s)

Pol Castellano-Escuder

Examples

```
data("st000284")

PomaClust(st000284)
```

PomaCorr *Correlation Analysis*

Description

This function returns different correlation plots and a table with all pairwise correlations in the data.

Usage

```
PomaCorr(
  data,
  method = "pearson",
  low = "red",
  outline = "white",
  high = "blue",
  label_size = 12,
  corr_type = "cor",
  coeff = 0.7
)
```

Arguments

`data`	A SummarizedExperiment object.
`method`	Character indicating which correlation coefficient has to be computed. Options are "pearson" (default), "kendall" and "spearman".
`low`	Color for low end of the gradient in corrplot.
`outline`	Color for the outline of the gradient in corrplot.
`high`	Color for high end of the gradient in corrplot.
`label_size`	Numeric indicating label size in corrplot.
`corr_type`	Type of correlation network. Options are "cor" (for global correlations) and "glasso" (for gaussian graphical model). Default is "cor". See `glasso` R package for the second option.

`coeff` Numeric indicating correlation coefficient. Edges with absolute weight below this value will be removed from the network. If "corr_type" is set to "glasso", this parameter indicates the regularization parameter for lasso (rho = 0 means no regularization). See `glasso::glasso()`.

Value

A list with the results.

Author(s)

Pol Castellano-Escuder

References

Jerome Friedman, Trevor Hastie and Rob Tibshirani (2019). glasso: Graphical Lasso: Estimation of Gaussian Graphical Models. R package version 1.11. https://CRAN.R-project.org/package=glasso

Examples

```
data("st000284")

# Pearson correlation
PomaCorr(st000284)$correlations

## Gaussian graphical model
# library(ggraph)
# PomaCorr(st000284, corr_type = "glasso")
```

`PomaDensity` *Distribution Plot*

Description

PomaDensity() generates a density plot of not normalized and normalized MS data. This plot can help in the comparison between pre and post normalized data and in the "validation" of the normalization process.

Usage

```
PomaDensity(
  data,
  group = "samples",
  feature_name = NULL,
  legend_position = "bottom"
)
```

Arguments

`data`	A SummarizedExperiment object.
`group`	Groupping factor for the plot. Options are "samples" and "features". Option "samples" (default) will create a density plot for each group and option "features" will create a density plot of each variable.
`feature_name`	A vector with the name/s of feature/s to plot. If it's NULL (default) a density plot of all variables will be created.
`legend_position`	Character indicating the legend position. Options are "none", "top", "bottom", "left", and "right".

Value

A ggplot2 object.

Author(s)

Pol Castellano-Escuder

Examples

```
data("st000284")

# samples
PomaDensity(st000284)

# features
PomaDensity(st000284, group = "features")

# concrete features
PomaDensity(st000284, group = "features",
            feature_name = c("ornithine", "orotate"))
```

PomaDESeq *Differential expression analysis based on the Negative Binomial distribution*

Description

DESeq2 package wrapper to estimate variance-mean dependence in count data from high-throughput sequencing assays and test for differential expression based on a model using the negative binomial distribution.

Usage

```
PomaDESeq(data, adjust = "BH")
```

Arguments

`data` A SummarizedExperiment object.

`adjust` Multiple comparisons correction method. Options are: "fdr", "holm", "hochberg", "hommel", "bonferroni", "BH", and "BY".

Value

A tibble with the results.

Author(s)

Pol Castellano-Escuder

References

Love, M.I., Huber, W., Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2 Genome Biology 15(12):550 (2014)

Examples

```
data("st000284")
st000284_sub <- st000284[, st000284@colData$factors %in% c("CRC", "Healthy")] # select two groups
SummarizedExperiment::assay(st000284_sub) <- floor(SummarizedExperiment::assay(st000284_sub)) # convert all valu
st000284_sub %>%
  PomaDESeq(adjust = "fdr")
```

PomaEDA *Automatic Exploratory Data Analysis HTML Report*

Description

This function automatically generates a HTML report with different exploratory plots and tables from an `SummarizedExperiment` object.

Usage

```
PomaEDA(
  data,
  imputation = "knn",
  normalization = "log_pareto",
  clean_outliers = TRUE,
  coeff_outliers = 1.5,
  username = NULL,
  institution = NULL
)
```

Arguments

`data` A SummarizedExperiment object.

`imputation` Imputation method. See `?POMA::PomaImpute()`.

`normalization` Normalization method. See `?POMA::PomaNorm()`.

`clean_outliers` Logical. If it's set to TRUE, outliers will be removed from EDA.

`coeff_outliers` This value corresponds to the classical 1.5 in $Q3 + 1.5 * IQR$ formula to detect outliers. See `?POMA::PomaOutliers()`.

`username` Author name in the report.

`institution` Institution name in the report.

Value

An exploratory data analysis HTML report.

Author(s)

Pol Castellano-Escuder

`PomaHeatmap` *Classical Heatmap*

Description

This function returns a basic heatmap plot made with base R.

Usage

```
PomaHeatmap(
  data,
  cols = 1,
  sample_names = TRUE,
  feature_names = FALSE,
  show_legend = TRUE
)
```

Arguments

`data` A SummarizedExperiment object.

`cols` Numerical vector indicating the column index of variables in `colData` to be displayed. Default is 1 (main group).

`sample_names` Logical indicating if sample names should be plotted or not. Default is TRUE.

`feature_names` Logical indicating if feature names should be plotted or not. Default is FALSE.

`show_legend` Logical indicating if legend should be plotted or not. Default is TRUE.

Value

A heatmap plot.

Author(s)

Pol Castellano-Escuder

Examples

```
data("st000284")

st000284 %>%
  PomaNorm() %>%
  PomaHeatmap()
```

PomaImpute *Collection of Imputation Methods for Mass Spectrometry Data*

Description

PomaImpute() offers different methods to impute missing values in MS data.

Usage

```
PomaImpute(
  data,
  ZerosAsNA = FALSE,
  RemoveNA = TRUE,
  cutoff = 20,
  method = "knn"
)
```

Arguments

`data` A SummarizedExperiment object.

`ZerosAsNA` Logical that indicates if the zeros in the data are missing values. Default is FALSE.

`RemoveNA` Logical that indicates if those features with more than selected cutoff missing values in each group have to be removed. Default is TRUE.

`cutoff` Numeric that indicates the percentage of missing values allowed in each group. If one of the groups have less missing values than selected cutoff value, these feature will not be removed.

`method` Imputation method. Options are: "none", "half_min", "median", "mean", "min", "knn" and "rf". If "none", all missing values will be replaced by zero.

Value

A SummarizedExperiment object with cleaned data.

Author(s)

Pol Castellano-Escuder

References

Armitage, E. G., Godzien, J., Alonso-Herranz, V., López-Gonzálvez, Á., & Barbas, C. (2015). Missing value imputation strategies for metabolomics data. Electrophoresis, 36(24), 3050-3060.

Examples

```
data("st000336")

PomaImpute(st000336, method = "knn")
```

PomaLasso *Lasso, Ridge and Elasticnet Regularized Generalized Linear Models for Binary Outcomes*

Description

PomaLasso() is an implementation of the lasso, ridge and elasticnet regression from `glmnet` package for binary outcomes.

Usage

```
PomaLasso(
  data,
  alpha = 1,
  ntest = NULL,
  nfolds = 10,
  lambda = NULL,
  labels = FALSE
)
```

Arguments

`data`	A SummarizedExperiment object.
`alpha`	Elasticnet mixing parameter. alpha = 1 is the lasso penalty and alpha = 0 is the ridge penalty. This value must be between 0 and 1.
`ntest`	Numeric indicating the percentage of observations that will be used as test set. Default is NULL (no test set).

`nfolds` Number of folds for CV (default is 10). Although nfolds can be as large as the sample size (leave-one-out CV), it is not recommended for large datasets. Smallest value allowable is nfolds = 3.

`lambda` A user supplied lambda sequence. Typical usage is to have the program compute its own lambda sequence based on `nlambda` and `lambda.min.ratio`. See `?glmnet::glmnet()`.

`labels` Logical indicating if feature names should be plotted in coefficient plot or not. Default is FALSE.

Value

A list with all results including plots, tables and the resulting prediction model.

Author(s)

Pol Castellano-Escuder

References

Jerome Friedman, Trevor Hastie, Robert Tibshirani (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software, 33(1), 1-22. URL http://www.jstatsoft.org/v33/i01/.

Examples

```
data("st000336")

# lasso
st000336 %>%
  PomaImpute() %>%
  PomaNorm() %>%
  PomaOutliers() %>%
  PomaLasso()

# elasticnet
st000336 %>%
  PomaImpute() %>%
  PomaNorm() %>%
  PomaOutliers() %>%
  PomaLasso(alpha = 0.5)

# ridge
st000336 %>%
  PomaImpute() %>%
  PomaNorm() %>%
  PomaOutliers() %>%
  PomaLasso(alpha = 0)
```

PomaLimma *Easy Implementation of* `limma` *Bioconductor Package*

Description

PomaLimma() uses the classical `limma` package.

Usage

```
PomaLimma(
  data,
  contrast = NULL,
  covariates = FALSE,
  covs = NULL,
  adjust = "fdr",
  weights = FALSE,
  cutoff = NULL
)
```

Arguments

`data`	A SummarizedExperiment object.
`contrast`	A character with the limma comparison. For example, "Group1-Group2" or "control-intervention".
`covariates`	Logical. If it's set to `TRUE` all metadata variables stored in `colData` will be used as covariables. Default = FALSE.
`covs`	Character vector indicating the name of `colData` columns that will be included as covariates. Default is NULL (all variables).
`adjust`	Multiple comparisons correction method. Options are: "fdr", "holm", "hochberg", "hommel", "bonferroni", "BH" and "BY".
`weights`	Logical indicating whether the limma model should estimate the relative quality weights for each group. See `?limma::arrayWeights()`.
`cutoff`	Default is NULL. If this value is replaced for a numeric value, the resultant table will contains only those features with an adjusted p-value below selected value.

Value

A tibble with limma results.

Author(s)

Pol Castellano-Escuder

References

Matthew E. Ritchie, Belinda Phipson, Di Wu, Yifang Hu, Charity W. Law, Wei Shi, Gordon K. Smyth, limma powers differential expression analyses for RNA-sequencing and microarray studies, Nucleic Acids Research, Volume 43, Issue 7, 20 April 2015, Page e47, https://doi.org/10.1093/nar/gkv007

Examples

```
data("st000284")

st000284 %>%
  PomaNorm() %>%
  PomaLimma(contrast = "Healthy-CRC", adjust = "fdr")
```

`PomaMultivariate` *Multivariate Statistical Methods for Mass Spectrometry Data*

Description

PomaMultivariate() allows users to perform different multivariate statistical analysis on MS data.

Usage

```
PomaMultivariate(
  data,
  method = "pca",
  components = 5,
  center = FALSE,
  scale = FALSE,
  labels = FALSE,
  load_length = 1,
  ellipse = TRUE,
  validation = "Mfold",
  folds = 5,
  nrepeat = 10,
  vip = 1.5,
  num_features = 10,
  legend_position = "bottom"
)
```

Arguments

`data` A SummarizedExperiment object.

`method` A multivariate method. Options are: "pca", "plsda" and "splsda".

`components` Numeric. Number of components to include in the model. Default is 5.

`center` Logical that indicates whether the variables should be shifted to be zero centered. Default is FALSE.

`scale` Logical that indicates whether the variables should be scaled to have unit variance before the analysis takes place. Default is FALSE.

`labels` Logical indicating if sample names should be plotted or not.

`load_length` Numeric between 1 and 2. Define the length of biplot loadings. Default is 1.

`ellipse` Logical that indicates whether a 95%CI ellipse should be plotted in scores plot. Default is TRUE.

`validation` (Only for "plsda" and "splsda" methods) Validation method. Options are "Mfold" and "loo".

`folds` (Only for "plsda" and "splsda" methods) Numeric. Number of folds for Mfold validation method (default is 5). If the validation method is loo, this value will become to 1.

`nrepeat` (Only for "plsda" and "splsda" methods) Numeric. Number of iterations for the validation method selected.

`vip` (Only for "plsda" method) Numeric indicating VIP cutoff to select features that will be displayed in vip plot.

`num_features` (Only for "splsda" method) Numeric. Number of variables selected to discriminate groups.

`legend_position`

Character indicating the legend position. Options are "none", "top", "bottom", "left", and "right".

Value

A list with all results for multivariate statistical analysis including plots and tables.

Author(s)

Pol Castellano-Escuder

Examples

```
data("st000336")

# PCA
st000336 %>%
  PomaImpute() %>%
  PomaNorm() %>%
  PomaOutliers() %>%
  PomaMultivariate(method = "pca")

# PLSDA
st000336 %>%
  PomaImpute() %>%
  PomaNorm() %>%
  PomaOutliers() %>%
  PomaMultivariate(method = "plsda", vip = 1)

# sPLSDA
```

```
st000336 %>%
  PomaImpute() %>%
  PomaNorm() %>%
  PomaOutliers() %>%
  PomaMultivariate(method = "splsda")
```

PomaNorm *Collection of Normalization Methods for Mass Spectrometry Data*

Description

PomaNorm() offers different methods to normalize MS data. This function contains both centering and scaling functions to normalize the data.

Usage

```
PomaNorm(data, method = "log_pareto", round = 3)
```

Arguments

`data`	A SummarizedExperiment object.
`method`	Normalization method. Options are: "none", "auto_scaling", "level_scaling", "log_scaling", "log_transformation", "vast_scaling" and "log_pareto".
`round`	Numeric. Number of decimal places (Default is 3).

Value

A SummarizedExperiment object with normalized data.

Author(s)

Pol Castellano-Escuder

References

van den Berg, R. A., Hoefsloot, H. C., Westerhuis, J. A., Smilde, A. K., & van der Werf, M. J. (2006). Centering, scaling, and transformations: improving the biological information content of metabolomics data. BMC genomics, 7(1), 142.

Examples

```
data("st000284")

PomaNorm(st000284, method = "log_pareto")
```

PomaOddsRatio *Logistic Regression Model Odds Ratios*

Description

PomaOddsRatio() calculates the Odds Ratios for each feature from a logistic regression model using the binary outcome (group/type must be a binary factor) as a dependent variable.

Usage

```
PomaOddsRatio(
  data,
  feature_name = NULL,
  covariates = FALSE,
  covs = NULL,
  showCI = TRUE
)
```

Arguments

`data`	A SummarizedExperiment object.
`feature_name`	A vector with the name/s of feature/s that will be used to fit the model. If it's NULL (default), all variables will be included in the model.
`covariates`	Logical that indicates if covariates will be included in logistic regression model. Default is `FALSE`.
`covs`	Character vector indicating the name of `colData` columns that will be included as covariates. Default is NULL (all variables).
`showCI`	Logical that indicates if the 95% confidence intervals will be plotted. Default is `TRUE`.

Value

A tibble with the Odds Ratios for all features with their 95% confidence intervals and a ggplot2 object.

Author(s)

Pol Castellano-Escuder

Examples

```
data("st000336")

st000336 %>%
  PomaImpute() %>%
  PomaNorm() %>%
  PomaOddsRatio(feature_name = c("glutamic_acid", "glutamine",
                                 "glycine", "histidine"))
```

PomaOutliers *Remove and Analyze Outliers*

Description

This function allows users to analyze outliers by different plots and remove them from an SummarizedExperiment object.

Usage

```
PomaOutliers(
  data,
  do = "clean",
  method = "euclidean",
  type = "median",
  coef = 1.5,
  labels = FALSE
)
```

Arguments

`data`	A SummarizedExperiment object.
`do`	Action to do. Options are "clean" (to remove detected outliers) and "analyze" (to analyze data outliers). Note that the output of this function will be different depending on this parameter.
`method`	Distance measure method to perform MDS. Options are "euclidean", "maximum", "manhattan", "canberra" and "minkowski". See `?dist()`.
`type`	Type of outliers analysis to perform. Options are "median" (default) and "centroid". See `vegan::betadisper`.
`coef`	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.
`labels`	Logical indicating if sample IDs should to be plotted or not.

Value

A SummarizedExperiment object without outliers OR an exploratory outlier analysis including both plots and tables (depending on "do" parameter).

Author(s)

Pol Castellano-Escuder

Examples

```
data("st000336")

# clean outliers
st000336 %>%
  PomaImpute() %>%
  PomaNorm() %>%
  PomaOutliers()

# analyze outliers
st000336 %>%
  PomaImpute() %>%
  PomaNorm() %>%
  PomaOutliers(do = "analyze")
```

PomaPCR *Principal Components Regression*

Description

Fits a Linear Model using the indicated variable as the dependent variable (outcome) and the indicated number of Principal Components as independent variables.

Usage

```
PomaPCR(
  data,
  n_components = 5,
  scale = TRUE,
  center = TRUE,
  outcome = NULL,
  intercept = TRUE
)
```

Arguments

data	A SummarizedExperiment object.
n_components	The number of Principal Components used in the PCR model. Defaults is 5.
scale	A logical value indicating whether the variables should be scaled to have unit variance before the analysis takes place.
center	A logical value indicating whether the variables should be shifted to be zero centered.
outcome	Character string indicating the variable name in `colData` to be used as the outcome.
intercept	A logical value indicating whether intercept should be included in the LM. Default is TRUE.

Value

A list with PCR results.

Author(s)

Pol Castellano-Escuder

Examples

```
data("st000284")

st000284 %>%
  PomaPCR(outcome = "age_at_consent")

data("st000336")

st000336 %>%
PomaImpute() %>%
PomaPCR(n_components = 2,
        outcome = "steroids",
        intercept = FALSE)
```

`PomaRandForest` *Classification Random Forest*

Description

PomaRandForest() allows users to perform a classification Random Forest using the classical `randomForest` R package.

Usage

```
PomaRandForest(
  data,
  ntest = NULL,
  ntree = 500,
  mtry = floor(sqrt(ncol(t(SummarizedExperiment::assay(data))))),
  nodesize = 1,
  nvar = 20
)
```

Arguments

`data`	A SummarizedExperiment object.
`ntest`	Numeric indicating the percentage of observations that will be used as test set. Default is NULL (no test set).
`ntree`	Number of trees to grow.

`mtry` Number of variables randomly sampled as candidates at each split. This value is set sqrt(p) (where p is number of variables in data) by default.

`nodesize` Minimum size of terminal nodes. By default is equal to 1.

`nvar` Number of variables to show in the Gini plot.

Value

A list with all results for Random Forest including plots and tables.

Author(s)

Pol Castellano-Escuder

References

A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. R News 2(3), 18–22.

Examples

```
data("st000336")

st000336 %>%
  PomaImpute() %>%
  PomaRandForest()
```

PomaRankProd *Rank Product/Rank Sum Analysis for Mass Spectrometry Data*

Description

PomaRankProd() performs the Rank Product method to identify differential feature concentration/intensity.

Usage

```
PomaRankProd(
  data,
  logged = TRUE,
  logbase = 2,
  paired = NA,
  cutoff = 0.05,
  method = "pfp"
)
```

Arguments

`data` A SummarizedExperiment object.

`logged` If "TRUE" (default) data have been previously log transformed.

`logbase` Numerical. Base for log transformation.

`paired` Number of random pairs generated in the function, if set to NA (default), the odd integer closer to the square of the number of replicates is used.

`cutoff` The pfp/pvalue threshold value used to select features.

`method` If cutoff is provided, the method needs to be selected to identify features. "pfp" uses percentage of false prediction, which is a default setting. "pval" uses p-values which is less stringent than pfp.

Value

A list with all results for Rank Product analysis including tables and plots.

Author(s)

Pol Castellano-Escuder

References

Breitling, R., Armengaud, P., Amtmann, A., and Herzyk, P.(2004) Rank Products: A simple, yet powerful, new method to detect differentially regulated genes in replicated microarray experiments, FEBS Letter, 57383-92

Hong, F., Breitling, R., McEntee, W.C., Wittner, B.S., Nemhauser, J.L., Chory, J. (2006). RankProd: a bioconductor package for detecting differentially expressed genes in meta-analysis Bioinformatics. 22(22):2825-2827

Del Carratore, F., Jankevics, A., Eisinga, R., Heskes, T., Hong, F. & Breitling, R. (2017). RankProd 2.0: a refactored Bioconductor package for detecting differentially expressed features in molecular profiling datasets. Bioinformatics. 33(17):2774-2775

Examples

```
data("st000336")

st000336 %>%
  PomaImpute() %>%
  PomaRankProd()
```

PomaSummarizedExperiment

Convert data frames to a SummarizedExperiment Object

Description

This function converts data frame objects to a SummarizedExperiment object.

Usage

```
PomaSummarizedExperiment(target, features)
```

Arguments

`target`	Metadata variables structured in columns. Sample ID must be the first column.
`features`	Matrix of features. Each feature in one column.

Value

A SummarizedExperiment object.

Author(s)

Pol Castellano-Escuder

References

Morgan M, Obenchain V, Hester J, Pagès H (2021). SummarizedExperiment: SummarizedExperiment container. R package version 1.24.0, https://bioconductor.org/packages/SummarizedExperiment.

Examples

```
data(iris)

# create target: two column (or more) data frame with IDs and Group factor
target <- data.frame(ID = 1:150, Group = iris$Species)

# create features: p column data frame (or matrix) with features
features <- iris[,1:4]

# create an SummarizedExperiment object with POMA
object <- PomaSummarizedExperiment(target = target, features = features)
```

PomaUMAP *Dimensionality Reduction with UMAP*

Description

Dimension reduction of the data using the Uniform Manifold Approximation and Projection (UMAP) method. See `?uwot::umap()` for more.

Usage

```
PomaUMAP(
  data,
  n_neighbors = NULL,
  n_components = 2,
  metric = "euclidean",
  pca = NULL,
  min_dist = 0.01,
  spread = 1,
  hdbscan_minpts = NULL,
  show_clusters = FALSE,
  show_group = FALSE,
  legend_position = "bottom"
)
```

Arguments

`data`	A SummarizedExperiment object.
`n_neighbors`	The size of local neighborhood (sample points) used for manifold approximation.
`n_components`	The dimension of the space to embed into. Defaults is 2.
`metric`	Distance measure method to find nearest neighbors. Options are "euclidean", "cosine", "manhattan", "hamming" and "correlation". See `?uwot::umap()`.
`pca`	If not NULL (default), reduce data to this number of columns using PCA before UMAP.
`min_dist`	The effective minimum distance between embedded points.
`spread`	The effective scale of embedded points.
`hdbscan_minpts`	Integer; Minimum size of clusters. See `?dbscan::hdbscan()`.
`show_clusters`	Logical indicating if clusters computed with HDBSCAN method should be plotted or not.
`show_group`	Logical indicating if the original sample group from target should be plotted or not.
`legend_position`	Character indicating the legend position. Options are "none", "top", "bottom", "left", and "right".

Value

A list with results including plots and tables.

Author(s)

Pol Castellano-Escuder

References

McInnes, L., Healy, J., & Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426.

Campello, R. J., Moulavi, D., & Sander, J. (2013, April). Density-based clustering based on hierarchical density estimates. In Pacific-Asia conference on knowledge discovery and data mining (pp. 160-172). Springer, Berlin, Heidelberg.

Examples

```
data("st000284")

st000284 %>%
  PomaNorm() %>%
  PomaUMAP()
```

PomaUnivariate *Univariate Statistical Methods for Mass Spectrometry Data*

Description

PomaUnivariate() allows users to perform different univariate statistical analysis on MS data.

Usage

```
PomaUnivariate(
  data,
  covariates = FALSE,
  covs = NULL,
  method = "ttest",
  paired = FALSE,
  var_equal = FALSE,
  adjust = "fdr"
)
```

Arguments

`data`	A SummarizedExperiment object.
`covariates`	Logical. If it's set to `TRUE` all metadata variables stored in `colData` will be used as covariates. Default = FALSE.
`covs`	Character vector indicating the name of `colData` columns that will be included as covariates. Default is NULL (all variables).
`method`	Univariate statistical method. Options are: "ttest", "anova", "mann" and "kruskal".
`paired`	Logical that indicates if the data is paired or not.
`var_equal`	Logical that indicates if the data variance is equal or not.
`adjust`	Multiple comparisons correction method. Options are: "fdr", "holm", "hochberg", "hommel", "bonferroni", "BH" and "BY".

Value

A tibble with results.

Author(s)

Pol Castellano-Escuder

Examples

```
data("st000336")
data("st000284")

# ttest
st000336 %>%
  PomaImpute() %>%
  PomaNorm() %>%
  PomaOutliers() %>%
  PomaUnivariate(method = "ttest")

# ANOVA
st000284 %>%
  PomaImpute() %>%
  PomaNorm() %>%
  PomaOutliers() %>%
  PomaUnivariate(method = "anova")
```

`PomaVolcano` *Volcano Plot*

Description

PomaVolcano() generates a volcano plot. Data should not contain negative values.

Usage

```
PomaVolcano(
  data,
  method = "ttest",
  pval = "raw",
  pval_cutoff = 0.05,
  adjust = "fdr",
  log2FC = 0.6,
  xlim = 2,
  labels = FALSE,
  paired = FALSE,
  var_equal = FALSE,
  interactive = FALSE,
  plot_title = TRUE
)
```

Arguments

`data` A SummarizedExperiment object.

`method` Statistical method to compute p-values and log2FC. Options are "ttest", "limma", and "DESeq".

`pval` Select a pvalue type to generate the volcano plot. Options are: "raw" and "adjusted".

`pval_cutoff` Numeric. Define the pvalue cutoff (horizontal line).

`adjust` Multiple comparisons correction method for t test result. Options are: "fdr", "holm", "hochberg", "hommel", "bonferroni", "BH" and "BY".

`log2FC` Numeric. Define the log2 fold change cutoff (vertical lines).

`xlim` Numeric. Define the limits for x axis.

`labels` Logical that indicates if selected labels will be plotted or not. Defaul is FALSE.

`paired` Only if method == "ttest". Logical that indicates if the data is paired or not.

`var_equal` Only if method == "ttest". Logical that indicates if the data variance is equal or not.

`interactive` Logical that indicates if an interactive plot will be plotted or not. Defaul is FALSE.

`plot_title` Logical that indicates if title will be plotted or not. Defaul is TRUE.

Value

A ggplot2 object.

Author(s)

Pol Castellano-Escuder

Examples

```
data("st000336")
data("st000284")

st000336 %>%
  PomaImpute() %>%
  PomaVolcano()

st000284_sub <- st000284[, st000284@colData$factors %in% c("CRC", "Healthy")] # select two groups

st000284_sub %>%
  PomaVolcano(method = "ttest")

SummarizedExperiment::assay(st000284_sub) <- floor(SummarizedExperiment::assay(st000284_sub)) # convert all valu

st000284_sub %>%
  PomaVolcano(method = "DESeq")
```

st000284 *Colorectal Cancer Detection Using Targeted Serum Metabolic Profiling*

Description

Colorectal cancer (CRC) is one of the most prevalent and deadly cancers in the world. Despite an expanding knowledge of its molecular pathogenesis during the past two decades, robust biomarkers to enable screening, surveillance, and therapy monitoring of CRC are still lacking. In this study, we present a targeted liquid chromatography-tandem mass spectrometry-based metabolic profiling approach for identifying biomarker candidates that could enable highly sensitive and specific CRC detection using human serum samples. In this targeted approach, 158 metabolites from 25 metabolic pathways of potential significance were monitored in 234 serum samples from three groups of patients (66 CRC patients, 76 polyp patients, and 92 healthy controls). Partial least squares-discriminant analysis (PLS-DA) models were established, which proved to be powerful for distinguishing CRC patients from both healthy controls and polyp patients. Receiver operating characteristic curves generated based on these PLS-DA models showed high sensitivities (0.96 and 0.89, respectively, for differentiating CRC patients from healthy controls or polyp patients); good specificities (0.80 and 0.88), and excellent areas under the curve (0.93 and 0.95) were also obtained. Monte Carlo cross validation (MCCV) was also applied, demonstrating the robust diagnostic power of this metabolic profiling approach.

Usage

```
st000284
```

Format

A `SummarizedExperiment` object: 224 samples, 113 metabolites, 4 covariables and 3 groups (CRC, Healthy and Polyp).

metabolites 113 serum metabolites.

covariables Age at consent, Gender, Smoking Condition and Alcohol Consumption.

Source

https://www.metabolomicsworkbench.org/data/DRCCMetadata.php?Mode=Study&StudyID=ST000284&StudyType=MS&ResultType=1%20target=_blank

References

Colorectal Cancer Detection Using Targeted Serum Metabolic Profiling, J. Proteome. Res., 2014, 13, 4120-4130.

---

st000336 *Targeted LC/MS of urine from boys with DMD and controls*

Description

Duchenne Muscular Dystrophy (DMD) is an X-linked recessive form of muscular dystrophy that affects males via a mutation in the gene for the muscle protein, dystrophin. Progression of the disease results in severe muscle loss, ultimately leading to paralysis and death. Steroid therapy has been a commonly employed method for reducing the severity of symptoms. This study aims to quantify the urine levels of amino acids and organic acids in patients with DMD both with and without steroid treatment. Track the progression of DMD in patients who have provided multiple urine samples.

Usage

```
st000336
```

Format

A `SummarizedExperiment` object: 57 samples, 31 metabolites, 1 covariable and 2 groups (Controls and DMD).

metabolites 31 urine metabolites.

covariables Steroid status.

Source

https://www.metabolomicsworkbench.org/data/DRCCMetadata.php?Mode=Study&DataMode=AllData&StudyID=ST000336&StudyType=MS&ResultType=1#DataTabs

Index

∗ **datasets**
  st000284, 29
  st000336, 30

cor_pmat, 3

flattenCorrMatrix, 3

PomaBoxplots, 4
PomaClust, 5
PomaCorr, 6
PomaDensity, 7
PomaDESeq, 8
PomaEDA, 9
PomaHeatmap, 10
PomaImpute, 11
PomaLasso, 12
PomaLimma, 14
PomaMultivariate, 15
PomaNorm, 17
PomaOddsRatio, 18
PomaOutliers, 19
PomaPCR, 20
PomaRandForest, 21
PomaRankProd, 22
PomaSummarizedExperiment, 24
PomaUMAP, 25
PomaUnivariate, 26
PomaVolcano, 27

st000284, 29
st000336, 30