Package 'MWASTools'

November 12, 2026

Type Package

Title MWASTools: an integrated pipeline to perform metabolome-wide association studies

Version 1.36.0

Date 2021-11-18

Author Andrea Rodriguez-Martinez, Joram M. Posma, Rafael Ayala, Ana L. Neves, Maryam Anwar, Jeremy K. Nicholson, Marc-Emmanuel Dumas

Maintainer Andrea Rodriguez-Martinez <andrea.rodriguez-martinez13@imperial.ac.uk>, Rafael Ayala <rafael.ayala@oist.jp>

Description MWASTools provides a complete pipeline to perform metabolome-wide association studies. Key functionalities of the package include: quality control analysis of metabonomic data; MWAS using different association models (partial correlations; generalized linear models); model validation using non-parametric bootstrapping; visualization of MWAS results; NMR metabolite identification using STOCSY; and biological interpretation of MWAS results.

License CC BY-NC-ND 4.0

Depends R(>= 3.4)

Suggests RUnit, BiocGenerics, knitr, BiocStyle, rmarkdown

VignetteBuilder knitr

Imports glm2, ppcor, qvalue, car, boot, grid, ggplot2, gridExtra, igraph, SummarizedExperiment, KEGGgraph, RCurl, KEGGREST, ComplexHeatmap, stats, utils

biocViews Metabolomics, Lipidomics, Cheminformatics, SystemsBiology, QualityControl

NeedsCompilation no

LazyData true

Encoding UTF-8

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

git_branch RELEASE_3_23

git_last_commit 70e7bb6

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Contents

CV_filter . . . 2
JBA_binning . . . 3
JBA_corDistribution . . . 5
JBA_plotBins . . . 6
KEGG_metabolic_paths . . . 7
metabo_SE . . . 7
MS_data . . . 8
MWAS_barplot . . . 8
MWAS_bootstrapping . . . 10
MWAS_filter . . . 11
MWAS_heatmap . . . 12
MWAS_KEGG_network . . . 13
MWAS_KEGG_pathways . . . 14
MWAS_KEGG_shortestpaths . . . 15
MWAS_network . . . 17
MWAS_scatterplotMS . . . 18
MWAS_skylineNMR . . . 20
MWAS_stats . . . 22
MWAS_SummarizedExperiment . . . 23
plot_spectraNMR . . . 24
QC_CV . . . 25
QC_CV_scatterplot . . . 26
QC_CV_specNMR . . . 27
QC_PCA . . . 29
QC_PCA_scoreplot . . . 30
STOCSY_NMR . . . 31
targetMetabo_SE . . . 32

Index **34**

`CV_filter` *Filter metabolic data by CV*

Description

This function allows filtering a matrix of metabolic variables based on the coefficient of variation (CV) of each variable across the quality control (QC) samples. See also function "QC_CV()".

Usage

```
CV_filter(metabo_SE, CV_metabo, CV_th = 0.30)
```

Arguments

metabo_SE	SummarizedExperiment object. See "MWAS_SummarizedExperiment()".
CV_metabo	numeric vector containing the CVs of the metabolic variables. See function "QC_CV()".
CV_th	numeric value indicating the CV threshold. Only features with CV below CV_th will be retained in the matrix.

Value

A SummarizedExperiment object containing the CV-filtered metabolic_data.

References

Dumas ME, et al. (2006). Assessment of analytical reproducibility of 1H NMR spectroscopy based metabonomics for large-scale epidemiological research: the INTERMAP Study. Analytical Chemistry, 78, 2199-1208.

Examples

```
## Load data
data(metabo_SE)

## Calculate CVs
CV_metabo  <-  QC_CV (metabo_SE)

## Filter metabolic_data by CV
metabo_CVfiltered <- CV_filter(metabo_SE, CV_metabo, CV_th = 0.30)
metabo_CVfiltered2 <- CV_filter(metabo_SE, CV_metabo, CV_th = 0.15)
```

JBA_binning *Binning of pJRES spectra*

Description

This function performs binning of pJRES spectra using the JBA algorithm (see details).

Usage

```
JBA_binning (NMR_data, st = 4, ct = 0.85, int = "sum", cm = "pearson",
             ef = 2, merge = TRUE, mt = 0.9)
```

Arguments

NMR_data	numeric matrix containing the NMR data (i.e. NMR peak intensities). The columns of the matrix must correspond to the metabolic variables (chemical shifts) and the rows to the samples. Column and row names must contain the metabolite IDs (i.e chemical shifts) and the sample IDs, respectively.
st	numeric value indicating the minimum bin size.
ct	numeric value indicating the correlation threshold. Bins with average correlation below ct will be neglected. This value can be established by comparing the distribution of average correlations in a spectral region dominated by electronic noise, and a spectral region dominated by metabolic signals. See function "JBA_corDistribution()".
int	character vector indicating the method used to calculate the bin intensity. Possible values are: "sum", "mean", "max", "median".
cm	character vector specifying the correlation method ("pearson" or "spearman").
ef	numeric value establishing the maximum number of upfield or downfield variables that can be used to expand the seed. The maximum number of variables that can be added on each side of a given seed (size = st) is st*ef.
merge	character constant indicating whether highly correlated (correlation > mt) adjacent bins will be merged (i.e. integrated as a single bin).
mt	numeric value indicating the correlation threshold used to merge adjacent bins. This argument is ignored if merge = FALSE.

Details

JBA ("pJRES Binning Algorithm") is a new binning method designed to extend the applicability of SRV (Blaise et al., 2009) to pJRES data. The main steps of the JBA algorithm are described below:

1) The algorithm scans the NMR spectra (from low to high frequencies) and calculates the average correlation of st adjacent variables, using a sliding window of size one. This means that a given bin i starts at the NMR variable i and finishes at NMR variable with i + (st -1).

2) The vector of average correlations can be visualized as pseudo-NMR spectrum, displaying the average correlation values in the y-axis and the chemical shifts in the x-axis. This correlation-based spectrum is then scanned to identify local maxima passing the ct threshold. Each of these local maxima is used as seed that can be expanded by progressively aggregating upfield and downfield NMR variables, as long as the following criteria are met: (i) the average correlation of the bin remains equal or above ct; (ii) for a given upfield variable (vi), correlation (vi, vi+1) needs to be equal or higher than correlation (vi, vi-1); (iii) for a given downfield variable (vz), correlation (vz, vz+1) needs to be equal or lower than correlation (vz, vz-1).

3) The intensity of each bin is calculated as the mean, median, sum or maximum intensity of all variables within the bin. Notice that due to misalignments/signal overlap, it is possible that a single peak is split into several bins. These bins can be detected based on a given correlation threshold and integrated as a single bin.

Value

A list containing binned NMR data and information about the bins, as indicated below:

-The first element of the list ("all_clusters") reports the average correlation of st adjacent NMR variables along the chemical shift axis, using a sliding window of size one.

-The second element ("JBA_seeds") contains the local maxima (i.e. seeds) of the correlation-based spectrum along the chemical shift axis.

-The fourth element ("JBA_bins_expanded") indicates the bin edges after expanding the seeds by aggregating upfield and downfield NMR variables.

-The fourth element ("JBA_data") contains the binned NMR data.

References

Blaise, et al. (2009). Statistical recoupling prior to significance testing in nuclear magnetic resonance based metabonomics. Analytical Chemistry, 81, 6242-6251.

Examples

```
## Not available.
```

JBA_corDistribution *Setting the ct threshold for JBA*

Description

This function compares the distribution of correlations between st adjacent variables in a spectral region dominated by noise and a spectral region dominated by metabolic signals. This function can be used to set the ct threshold for JBA binning.

Usage

```
JBA_corDistribution(NMR_data, st = 4, cm = "pearson", metabo_range = c(3.50, 3.96),
              noise_range = c(9.72, 9.99), color_scale = c("lightcoral", "honeydew3"))
```

Arguments

NMR_data	numeric matrix containing the NMR data (i.e. NMR peak intensities). The columns of the matrix must correspond to the metabolic variables (chemical shifts) and the rows to the samples. Column and row names must contain the metabolite IDs (i.e chemical shifts) and the sample IDs, respectively.
st	numeric value indicating the minimum bin size.
cm	character vector specifying the correlation method ("pearson" or "spearman").
metabo_range	numeric vector indicating the limits of a spectral region dominated by metabolic signals.
noise_range	numeric vector indicating the limits of a spectral region dominated by noise.
color_scale	character vector indicating color of the metabolic curve (first value), and the noise curve (second value).

Value

A plot comparing the distribution of average correlations between st adjacent variables in a spectral region dominated by metabolic signals (metabo_range) and in a spectral region dominated by electronic noise (noise_range). The suggested ct value corresponds to the correlation coefficient where the cumulative proportion of noise clusters is 1.

Examples

```
## Not available.
```

`JBA_plotBins` *Visualization of JBA bins*

Description

This function allows visualizing the bins generated by the JBA algorithm.

Usage

```
JBA_plotBins(NMR_JBA, NMR_data, ct = 0.85, ref_sample = 1, xlim = NULL,
             ylim = NULL)
```

Arguments

`NMR_JBA`	list corresponding to the output of "JBA_binning()".
`NMR_data`	numeric matrix containing the NMR data (i.e. NMR peak intensities). The columns of the matrix must correspond to the metabolic variables (chemical shifts) and the rows to the samples. Column and row names must contain the metabolite IDs (i.e chemical shifts) and the sample IDs, respectively.
`ct`	numeric value indicating the correlation threshold. Bins with average correlation below ct will be neglected. This value can be established by comparing the distribution of average correlations in a spectral region dominated by electronic noise, and a spectral region dominated by metabolic signals. See function "JBA_corDistribution()".
`ref_sample`	numeric value indicating the index of the reference spectrum.
`xlim`	numeric vector containing the minimum and maximum values of the x axis.
`ylim`	numeric vector containing the minimum and maximum values of the y axis.

Value

A plot with two panels. The upper panel shows the reference spectrum with the bin edges (start: dark blue, end: light blue). The lowe panel shows the corresponding correlation-based spectrum.

Examples

```
## Not available.
```

KEGG_metabolic_paths *KEGG human metabolic pathways*

Description

The first element of this list contains the KEGG identifiers (IDs) and names of 51 human metabolic pathways. By default, the function "MWAS_KEGG_network()" builds a reaction network using these KEGG IDs. The second element of the list is a matrix containing KEGG reactions with incorrect/inconsistent directionality. The directionality of these reactions has been corrected based on published literature. This matrix can be updated or edited by the user if required.

Usage

```
data(KEGG_metabolic_paths)
```

Format

List

Value

List

metabo_SE *NMR plasma metabolic profiles dataset*

Description

This SummarizedExperiment object contains the following information:

-An assay matrix containing the 1H NMR profiles (1.60 - 0.80 ppm) of 506 plasma samples from the FGENTCARD cohort and 10 identical quality control (QC) samples. The QC samples were prepared from a representative pool of the experimental samples, and were injected regularly throughout the run to ensure analytical reproducibility.

-A data.frame containing clinical information (age, gender, type II diabetes status and BMI) and sample class (i.e. experimental sample or QC sample) information for each sample row in the assay matrix.

Usage

```
data(metabo_SE)
```

Format

SummarizedExperiment

Value

SummarizedExperiment

`MS_data` *Simulated LC-MS features*

Description

A matrix with simulated LC-MS features (retention times in the first column, and mz values in the second column.)

Usage

```
data(MS_data)
```

Format

Matrix

Value

Matrix

`MWAS_barplot` *Visualize MWAS results in a bar plot*

Description

This function creates a bar plot based on the output from "MWAS_stats()". This function is designed to visualize MWAS results in the case of discrete metabolic variables (e.g. target GC/MS metabolites).

Usage

```
MWAS_barplot(MWAS_matrix, alpha_th = 0.05, width = NULL,
             scale_color = c("darkgray", "cornflowerblue", "firebrick1"),
             legend_labs =c("unchanged","downregulated","upregulated"),
             ylab = "sign*log(pFDR)", size_yaxis = 12, size_ylab = 12,
             size_names = 10, angle_names = 45, sort = TRUE)
```

Arguments

MWAS_matrix	numeric matrix resulting from the function "MWAS_stats()".
alpha_th	numeric value indicating the significance threshold.
width	numeric value indicating bar width.
scale_color	character vector corresponding to the 3-color scale that will be used to represent the association results. The first color of the scale indicates "no change", the second color indicates "downregulation", and the third color indicates "upregulation".
legend_labs	character vector containing the legend labels, according to scale_color.
ylab	character vector specifying a title for the y-axis.
size_yaxis	numeric value indicating the font size of y-axis title.
size_ylab	numeric value indicating the font size of y-axis labels.
size_names	numeric value indicating the font size of the metabolite ids displayed on the x-axis.
angle_names	numeric value indicating the angle in which the metabolite ids will be displayed on the x-axis.
sort	logical constant indicating whether the metabolites will be sorted based on MWAS results.

Value

A bar plot.

Examples

```
## Load data
data(targetMetabo_SE)

## Test for association between diabetes and target_metabolites
T2D_model <- MWAS_stats (targetMetabo_SE, disease_id = "T2D",
                         confounder_ids = c("Age", "Gender", "BMI"),
                         assoc_method = "logistic")

## Bar plot
MWAS_barplot(T2D_model)
MWAS_barplot(T2D_model, width = 0.7) # change bar width
MWAS_barplot(T2D_model, width = 0.7, angle_names = 90)
```

`MWAS_bootstrapping` *MWAS bootstrap resampling*

Description

This function generates bootstrap replicates (non-parametric resampling) of a model testing for association between a given metabolite and a disease phenotype, and calculates the confidence interval of model coefficients. Confidence intervals are calculated using the adjusted bootstrap percentile (BCa) method.

Usage

```
MWAS_bootstrapping (metabo_SE, metabolite_id, disease_id, confounder_ids = NULL,
                    assoc_method, iterations = 10000)
```

Arguments

`metabo_SE`	SummarizedExperiment object. See "MWAS_SummarizedExperiment()".
`metabolite_id`	character vector corresponding to the id of the metabolite to be modeled.
`disease_id`	character vector corresponding to the id of the response to be modeled.
`confounder_ids`	optional character vector corresponding to the ids of covariates to be included in the model (e.g. age or gender).
`assoc_method`	character constant indicating the association method that will be used. Possible values for assoc_method are: "pearson" (pearson correlation), "spearman" (spearman correlation), "kendall" (kendall correlation), "linear" (linear regression) or "logistic" (logistic regression).
`iterations`	numeric value indicating the number of bootstrap replicates

Value

A list with 3 elements, each list element reporting the following information: i) object of class "boot"; ii) summary of the previous object; iii) 95-confidence interval of the metabolite model coefficient. For more details, check the function "boot()" from the "boot" package.

References

Davison AC, Hinkley, DV. (1997). Bootstrap Methods and Their Application. Cambridge University Press.

Examples

```
## Load data
data(targetMetabo_SE)

## Bootstrap model testing for association between diabetes (T2D) and 3OH-butyrate
MWAS_bootstrapping (targetMetabo_SE, metabolite_id = "3-Hydroxybutyrate",
                    disease_id = "T2D",assoc_method = "logistic",
                    iterations = 1000)
```

MWAS_filter *Filter MWAS results by p-value and/or CV*

Description

This function allows filtering the output matrix from "MWAS_stats()", by p-value and/or coefficient of variation (CV).

Usage

```
MWAS_filter(MWAS_matrix, type = "pvalue", alpha_th = 0.05, CV_th = 0.30, sort = FALSE)
```

Arguments

MWAS_matrix	numeric matrix generated by the function "MWAS_stats()".
type	character constant indicating the filtering criteria. If type = "pvalue", only metabolic variables with p-value below alpha_th will be retained in the MWAS_matrix. If type = "CV", only metabolic variables with CV below CV_th will be retained. If type = "all", only metabolic variables with CV below CV_th and p-value below alpha_th will be retained.
alpha_th	numeric value indicating the significance threshold.
CV_th	numeric value indicating the CV threshold.
sort	logical constant indicating whether the filter MWAS_matrix will be sorted based on p-values.

Value

A numeric matrix corresponding to the filtered MWAS_matrix. The matrix has an additional column, which indicates the index of each metabolic variable in the original MWAS_matrix.

Examples

```
## Load data
data(targetMetabo_SE)

## Test for association between diabetes and target_metabolites
T2D_model <- MWAS_stats (targetMetabo_SE, disease_id = "T2D",
                         assoc_method = "logistic")

## Filter T2D_model by p-value
MWAS_filter(T2D_model, type = "pvalue", alpha_th = 0.001, sort = TRUE)

## Subset targetMetabo_SE based on pvalue_filter
pvalue_filter <- MWAS_filter(T2D_model, type = "pvalue", alpha_th = 0.001)
index_features <- pvalue_filter[, 4]
targetMetabo_SE[index_features, ]
```

`MWAS_heatmap` *Visualize MWAS results as a multiple-phenotype heatmap*

Description

This function allows visualizing MWAS results generated using multiple phenotypes as a heatmap. The values of the heatmap are the individual MWAS scores: -log10 p-values (corrected for multiple-testing) adjusted for the direction of the association. The metabolites are ordered based on hierarchical cluster analysis of the auto-correlation metabolic matrix.

Usage

```
MWAS_heatmap (metabo_SE, MWAS_list, alpha_th = 0.05, display_all = TRUE, ncut = 3, ...)
```

Arguments

`metabo_SE`	SummarizedExperiment object. See "MWAS_SummarizedExperiment()".
`MWAS_list`	list of matrices generated with the function "MWAS_stats()". The names of the individual matrices must correspond to the phenotype names. The dimensions of all matrices must be the same, and consistent with metabo_SE dimensions.
`alpha_th`	numeric value indicating MWAS significance threshold. Metabolites with p-value (corrected for multiple-testing) above alpha_th will have a MWAS score of 0.
`display_all`	logical constant indicating whether all metabolites from metabo_SE will be shown in the heatmap, or only the ones significantly associated with at least one phenotype.
`ncut`	numeric value indicating where the tree will be cut.
`...`	other arguments passed to the function "Heatmap()" from the ComplexHeatmap package.

Value

A heatmap showing MWAS results generated with multiple phenotypes. The function also returns a matrix indicating the metabolic clusters.

References

Gu Z, et al. (2016). Complex heatmaps reveal patterns and correlations in multidimensional genomic data. Bioinformatics, 32, 2847-2849.

Examples

```
## Load data
data(targetMetabo_SE)

## Test for association between diabetes and target_metabolites
```

```
T2D_model <- MWAS_stats (targetMetabo_SE, disease_id = "T2D",
                         confounder_ids = c("Age", "Gender", "BMI"),
                         assoc_method = "logistic")

## Test for association between BMI and target_metabolites
BMI_model <- MWAS_stats (targetMetabo_SE, disease_id = "BMI",
                         confounder_ids = c("Age", "Gender", "T2D"),
                         assoc_method = "spearman")

## Generate MWAS_list: do not forget the names!
MWAS_list <- list(T2D = T2D_model, BMI = BMI_model)

## Generate heatmap
MWAS_heatmap (targetMetabo_SE, MWAS_list, alpha_th = 0.05)
```

`MWAS_KEGG_network` *Build a KEGG-based metabolic network*

Description

This function generates a KEGG-based metabolic network connecting substrate-product pairs. The network is formated as a four-column matrix, where each row represents an edge connecting two metabolites (from metabolite in column 1 to metabolite in column 2). The third column contains the identifiers (IDs) of the reactions performing each of the metabolic conversions, while the fourth column indicates the direction of each reaction.

Usage

```
MWAS_KEGG_network(kegg_paths = NULL)
```

Arguments

`kegg_paths`	character vector containing the KEGG IDs of the metabolic pathways of interest (organism-specific). For example, the KEGG ID for the human "glycolysis/gluconeogenesis" pathway is"hsa00010". By default, the KEGG IDs contained in the dataset "KEGG_metabolic_paths" will be used.

Value

A four-column matrix where each row represents an edge between two nodes.

Note

Like in the MetaboSignal package, reaction directionality has been cross-checked and corrected (when required) based on previous literature (Duarte et al., 2007).

References

Duarte NC, et al. (2007). Global reconstruction of the human metabolic network based on genomic and bibliomic data. Proceedings of the National Academy of Sciences, 104, 1777-1782.

Posma JM, et al. (2014). MetaboNetworks, an interactive Matlab-based toolbox for creating, customizing and exploring sub-networks from KEGG. Bioinformatics, 30, 893-895.

Rodriguez-Martinez A, et al. (2017).MetaboSignal: a network-based approach for topological analysis of metabotype regulation via metabolic and signaling pathways. Bioinformatics, 33, 773-775.

Zhang JD, Wiemann S. (2009). KEGGgraph: a graph approach to KEGG PATHWAY in R and Bioconductor. Bioinformatics, 25, 1470-1471.

http://www.kegg.jp/kegg/docs/keggapi.html

Examples

```
## Define the paths that will be used to build the network
data(KEGG_metabolic_paths)
metabo_paths =  KEGG_metabolic_paths[[1]][, 1]

## Build metabolic network-table: might take few minutes
metabolic_network = MWAS_KEGG_network(kegg_paths = metabo_paths)
```

MWAS_KEGG_pathways *Map metabolites into KEGG pathways*

Description

This function allows mapping the metabolites of interest detected by MWAS analysis onto the KEGG pathways. The function also exports a network file and an attribute file which can be imported into Cytoscape to visualize the results as a pathway-based metabolic network.

Usage

```
MWAS_KEGG_pathways(metabolites, MWAS_matrix = NULL, file_name = "KeggPaths")
```

Arguments

metabolites	character vector containing the KEGG IDs of the metabolites of interest detected by MWAS. The order of the metabolite IDs in this vector must match the order in MWAS_matrix. Compound KEGG IDs can be obtained using the function "MS_keggFinder()" from the MetaboSignal package.
MWAS_matrix	numeric matrix generated with the function "MWAS_stats()". It can also be a submatrix containing only the significant metabolites, generated with the function "MWAS_filter()".
file_name	character vector that allows customizing the name of the exported files.

Value

A six-column matrix indicating the KEGG pathways where each metabolite was mapped. The results are formatted as a six-column matrix containing the following information: metabolite KEGG ID (column 1), metabolite name (column 2), pathway KEGG ID (column 3), pathway name (column 4), pathway class (column 5), pathway organism (i.e. "Human"/"Not_human") (column 6).

The function also exports a network file ("KeggPaths_NetworkFile.txt") and an attribute file ("KeggPaths_AttributeFile.txt") that can be imported into Cytoscape to visualize the results as a network. The attribute file allows customizing the metabolites of interest based on a score reflecting the degree of association with the phenotype under study (i.e. log10(pvalue) adjusted for the sign of the association).

References

Rodriguez-Martinez A, et al. (2017).MetaboSignal: a network-based approach for topological analysis of metabotype regulation via metabolic and signaling pathways. Bioinformatics, 33, 773-775.

Shannon P, et al. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Research, 13, 2498-2504.

Tenenbaum D. (2017). KEGGREST: Client-side REST access to KEGG. R package.

Examples

```
## Test for association between diabetes and target_metabolites
T2D_model <- MWAS_stats (targetMetabo_SE, disease_id = "T2D",
                         confounder_ids = c("Age", "Gender", "BMI"),
                         assoc_method = "logistic")

## Select the metabolites of interest and get their corresponding KEGG IDs
T2D_model_subset = T2D_model[1:5, ]
kegg_metabolites = c("cpd:C00186", "cpd:C01089", "cpd:C00123", "cpd:C00183",
                     "cpd:C00407")

## Map metabolites into KEGG pathways
kegg_pathways = MWAS_KEGG_pathways(metabolites = kegg_metabolites,
                                   MWAS_matrix = T2D_model_subset)
```

MWAS_KEGG_shortestpaths

Build a shortest-path metabolic subnetwork

Description

This function allows calculating the shortest paths between the metabolites of interest detected by MWAS analysis, and representing them as a network. The function also generates a network file and an attribute file, which can be easily imported into Cytoscape to visualize the network.

Usage

```
MWAS_KEGG_shortestpaths(network_table, metabolites, MWAS_matrix = NULL,
                        type = "all", distance_th = "Inf", names = TRUE,
                        file_name = "KeggSP")
```

Arguments

network_table	four-column matrix where each row represents an edge between two nodes. See function "MWAS_KEGG_network()".
metabolites	character vector containing the KEGG IDs of the metabolites of interest detected by MWAS. The order of the metabolite IDs in this vector must match the order in MWAS_matrix. Compound KEGG IDs can be obtained using the function "MS_keggFinder()" from the MetaboSignal package.
MWAS_matrix	numeric matrix generated with the function "MWAS_stats()". It can also be a submatrix containing only the significant metabolites, generated with the function "MWAS_filter()".
type	character constant indicating whether all shortest paths (type = "all") or a single shortest path (type = "first") will be considered when there are several shortest paths between a given source metabolite and a given target metabolite.
distance_th	establishes a shortest path length threshold. Only shortest paths with length below this threshold will be included in the network.
names	logical scalar indicating whether the metabolite KEGG IDs will be transformed into common metabolite names.
file_name	character vector that allows customizing the name of the exported files.

Value

A four-column matrix where each row represents an edge connecting two metabolites (from metabolite in column 1 to metabolite in column 2). The reactions involved in each metabolic conversion as well as the reaction type (i.e. reversible or irreversible) are reported in the third and fourth columns, respectively. This network can be visualized in R using the igraph package or similar packages.

The function also exports a network file ("KeggSP_NetworkFile.txt") and an attribute file ("KeggSP_AttributeFile.txt"), which can be easily imported into Cytoscape to visualize the network. The attribute file allows customizing the metabolites of interest based on a score reflecting the degree of association with the phenotype under study (i.e. log10(pvalue) adjusted for the sign of the association).

References

Csardi G, Nepusz T. (2006). The igraph software package for complex network research. InterJournal, Complex Systems, 1695.

Posma JM, et al. (2014). MetaboNetworks, an interactive Matlab-based toolbox for creating, customizing and exploring sub-networks from KEGG. Bioinformatics, 30, 893-895.

Rodriguez-Martinez A, et al. (2017).MetaboSignal: a network-based approach for topological analysis of metabotype regulation via metabolic and signaling pathways. Bioinformatics, 33, 773-775.

Shannon P, et al. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Research, 13, 2498-2504.

Examples

```
## Build metabolic network: might take few minutes
metabolic_network = MWAS_KEGG_network(kegg_paths = KEGG_metabolic_paths[[1]][, 1])

## Test for association between diabetes and target_metabolites
T2D_model <- MWAS_stats (targetMetabo_SE, disease_id = "T2D",
                         confounder_ids = c("Age", "Gender", "BMI"),
                         assoc_method = "logistic")

## Select the metabolites of interest and get their corresponding KEGG IDs
T2D_model_subset = T2D_model[1:5, ]
kegg_metabolites = c("cpd:C00186", "cpd:C01089", "cpd:C00123", "cpd:C00183",
                     "cpd:C00407")

## Build shortest-path subnetwork
keggSP_subnetwork = MWAS_KEGG_shortestpaths(network_table = metabolic_network,
                                            metabolites = kegg_metabolites,
                                            MWAS_matrix = T2D_model_subset)
```

MWAS_network *Visualize MWAS results in a correlation-based metabolic network*

Description

This function allows visualizing MWAS results in a correlation-based metabolic network. The network is an undirected graph where the nodes represent the metabolites, and the edges represent a co-abundance relationship between pairs of nodes. Different node parameters (e.g. color, size) can be customized based on MWAS results.

Usage

```
MWAS_network (metabo_SE, MWAS_matrix, alpha_th = 0.05, cor_th = 0.25,
              file_name = "Correlation", res_cor = 2)
```

Arguments

metabo_SE	SummarizedExperiment object. See "MWAS_SummarizedExperiment()".
MWAS_matrix	numeric matrix generated by the function "MWAS_stats()".
alpha_th	numeric value indicating MWAS significance threshold.
cor_th	numeric value indicating the co-abundance similarity threshold. Thus, two metabolites will be linked in the network if the absolute correlation (Pearson) between them exceeds cor_th.
file_name	character string indicating the name given to the cytoscape files that will be exported to the working directory.
res_cor	numeric value restricting the number of decimals of the correlation of coefficients used to build the edges of the network.

Value

A correlation based-metabolic network formalized as a weigthed igraph object. This igraph object contains two node attributes: "score" and "color". "score" is a vector containing the MWAS score (-log10(pvalue)*estimate sign) of each metabolite. "color" is a vector indicating the color of each node based on MWAS results ("cornflowerblue": "downregulation", "gray":"no change", "firebrick1":"upregulation"). These attributes can be used to customize node parameters based on MWAS results. The function also exports a network file ("Correlation_NetworkFile.txt") and an attribute file ("Correlation_AttributeFile.txt") of MWAS scores, which can be imported into cytoscape to visualize the network.

References

Csardi G, Nepusz T. (2006). The igraph software package for complex network research. InterJournal, Complex Systems, 1695.

Examples

```
## Load data
data(targetMetabo_SE)

## Test for association between diabetes and target_metabolites
T2D_model <- MWAS_stats (targetMetabo_SE, disease_id = "T2D",
                         confounder_ids = c("Age", "Gender", "BMI"),
                         assoc_method = "logistic")

## Build correlation-based metabolic network
net_T2D <- MWAS_network(targetMetabo_SE, T2D_model, file_name = "MWAS_T2D",
                        cor_th = 0.30)

## Visualize network using the igraph package
# library(igraph)
# plot(net_T2D, vertex.size = abs(V(net_T2D)$score*6)) # node size based on scores

# plot(net_T2D, vertex.size = abs(V(net_T2D)$score*6),
#      edge.label = E(net_T2D)$weight) # show edge labels
```

MWAS_scatterplotMS *Visualize MWAS results in MS-based scatter plot*

Description

This function creates a MS-based scatter plot (rt vs mz) based on the output from "MWAS_stats()". MS-features are filtered according to a given significance threshold and only significant features are represented in the scatter plot. The color of the marker representing each significant MS feature indicates the direction of the association (i.e. downregulation or upregulation) and the size of the marker indicates the strength of the association (i.e. -log10(p-value)).

Usage

```
MWAS_scatterplotMS (rt, mz, MWAS_matrix, alpha_th = 0.05, xlab = "rt",
                ylab = "mz", pch = 20, scale_color = c("cornflowerblue", "red"),
                    xlim = NULL, ylim = NULL, size_axis = 10, size_lab = 10,
                    legend_position = "bottom")
```

Arguments

rt	numeric vector of retention time values.
mz	numeric vector of mz values.
MWAS_matrix	numeric matrix resulting from the function "MWAS_stats()". The dimensions of this matrix must be consistent with the length of rt and mz
alpha_th	numeric value indicating the significance threshold. Only variables with p-value (corrected for multiple-testing) below alpha_th will be plotted.
xlab	character vector specifying a title for the x-axis.
ylab	character vector specifying a title for the y-axis.
pch	value specifying the symbol used to represent each MS feature in the scatter plot. To see all possible symbols, check "plot()" options.
scale_color	character vector corresponding to the 2-color scale that will be used to represent the association results. The first color of the scale indicates "downregulation", and the second color indicates "upregulation".
xlim	numeric vector containing the minimum and maximum values of the x-axis.
ylim	numeric vector containing the minimum and maximum values of the y-axis.
size_axis	numeric value indicating the font size of x- and y-axis title.
size_lab	numeric value indicating the font size of x- and y-axis labels.
legend_position	character vector indicating the position of the legend: "top", "bottom", "right", "left", "none".

Value

A MS-based scatter plot where MS features are represented according to MWAS_results.

Examples

```
## Load data
data(MS_data)
rt <- MS_data[, 1]
mz <- MS_data[, 2]

## Simulate MWAS data
set.seed(100)
estimates <- runif(length(rt), -1, 1)
pvalues <- rbeta(length(estimates), 0.5, 1)
pFDR <- p.adjust(pvalues, method = "BH")
MWAS_matrix <- cbind(estimates, pvalues, pFDR)
```

```
## MS-based scatter plot
MWAS_scatterplotMS(rt, mz, MWAS_matrix)
MWAS_scatterplotMS(rt, mz, MWAS_matrix, alpha_th = 0.01)
MWAS_scatterplotMS(rt, mz, MWAS_matrix, alpha_th = 0.01,
                   scale_color = c("yellow", "blue"))
```

`MWAS_skylineNMR` *Visualize MWAS results in an NMR-skyline plot*

Description

This function generates a 2-panel figure showing the results from "MWAS_stats()" applied to NMR data. The upper panel shows an NMR-skyline plot (comparable to a GWAS-Manhattan plot), where the chemical shifts are displayed along the x-axis and the -log10 p-values (sign-adjusted for the direction of the association) are displayed on the y-axis. The lower panel shows an NMR spectrum colored according to MWAS results.

Usage

```
MWAS_skylineNMR (metabo_SE, MWAS_matrix, ref_sample, alpha_th = 0.05, output = "all",
             xlab ="ppm", ylab1 = "sign*log(pFDR)", ylab2 = "intensity", pch = 20,
              marker_size = 1, scale_color = c("black", "cornflowerblue", "red"),
                  size_lab = 12, size_axis = 12, xlim = NULL, ylim1 = NULL,
                  ylim2 = NULL, guide_type = "legend", xbreaks = waiver(),
                  xnames = waiver(), ybreaks1 = waiver(), ybreaks2 = waiver(),
                  ynames1 = waiver(), ynames2 = waiver())
```

Arguments

`metabo_SE`	SummarizedExperiment object. See "MWAS_SummarizedExperiment()".
`MWAS_matrix`	numeric matrix resulting from the function "MWAS_stats()".
`ref_sample`	character vector indicating the ID of the sample that will be used to plot the NMR spectrum.
`alpha_th`	numeric value indicating the significance threshold.
`output`	character constant indicating the outcome of the function ("skyline", "spectrum" or "all"). If outcome = "all", both the skyline and the spectrum with be plotted in a 2-panel plot.
`xlab`	character vector specifying a title for the x-axis.
`ylab1`	character vector specifying a title for the y-axis of the upper panel.
`ylab2`	character vector specifying a title for the y-axis of the lower panel.
`pch`	value specifying the symbol used to represent each ppm value in the skyline plot. To see all possible symbols, check "plot()" options.
`marker_size`	numeric value indicating the size of the symbol used to represent each ppm value in the skyline plot.

scale_color	character vector corresponding to the 3-color scale that will be used to represent the association results. The first color of the scale indicates "no change", the second color indicates "downregulation", and the third color indicates "upregulation".
size_lab	numeric value indicating the font size of x- and y-axis titles.
size_axis	numeric value indicating the font size of x- and y-axis labels.
xlim	numeric vector containing the minimum and maximum values of the x-axis. Notice that ppm is displayed in reverse scale (e.g. xlim = c(5, 2)).
ylim1	numeric vector containing the minimum and maximum values of the y-axis for the upper panel.
ylim2	numeric vector containing the minimum and maximum values of the y-axis for the lower panel.
guide_type	character constant indicating the guide ("legend" or "none") that will be added to the plots.
xbreaks	numeric vector indicating the positions of the breaks of the x-axis.
xnames	character vector (same length as xbreaks) containing the labels of each break of the x-axis.
ybreaks1	numeric vector indicating the positions of the breaks of the y-axis for the upper panel.
ybreaks2	numeric vector indicating the positions of the breaks of the y-axis for the lower panel.
ynames1	character vector (same length as ybreaks1) containing the labels of each break of the y-axis for the upper panel.
ynames2	character vector (same length as ybreaks2) containing the labels of each break of the y-axis for the lower panel.

Value

By default, a plot with 2 panels, the upper panel showing an NMR-skyline plot and the lower panel showing an NMR spectrum colored based on MWAS results.

References

Elliott P, et al. (2015). Urinary metabolic signatures of human adiposity. Science Translational Medicine, 7, 285ra62.

Examples

```
## Load data
data(metabo_SE)

## Test for association between BMI and metabolic_data
BMI_model <- MWAS_stats (metabo_SE, disease_id = "BMI", assoc_method = "spearman",
                         output = "pvalues")

## Create skyline plots
MWAS_skylineNMR (metabo_SE, BMI_model, ref_sample = "QC1")
MWAS_skylineNMR (metabo_SE, BMI_model, ref_sample = "QC1", pch = "*", marker_size = 3)
```

MWAS_stats *Metabolome-Wide Associations*

Description

This function tests for association between individual metabolites and a disease phenotype.

Usage

```
MWAS_stats (metabo_SE, disease_id, confounder_ids = NULL, assoc_method, mt_method = "BH",
            output = "pvalues", CV_metabo = NULL)
```

Arguments

metabo_SE	SummarizedExperiment object. See "MWAS_SummarizedExperiment()".
disease_id	character vector corresponding to the ID of the response to be modeled.
confounder_ids	optional character vector corresponding to the IDs of the covariates to be included in the model (e.g. age or gender).
assoc_method	character constant indicating the association method that will be used. Possible values for assoc_method are: "pearson" (Pearson correlation), "spearman" (Spearman correlation), "kendall" (Kendall correlation), "linear" (linear regression) or "logistic" (logistic regression).
mt_method	character constant indicating the multiple-testing correction method that will be used. Possible values for mt_method are: "BH" (Benjamini and Hochberg), "bonferroni", "holm", "hochberg", "hommel", "BY" (Benjamini and Yekutieli), "qvalues", or "none".
output	character constant indicating the output of the function. If output = "pvalues", p-values and estimates for each metabolic variable will be returned as a matrix. If output = "models", detailed information about the statistical model fitted for each metabolic variable will be returned.
CV_metabo	optional numeric vector containing the coefficients of variation of the metabolic variables. This vector will be added as an additional column of the output matrix.

Value

By default, a matrix where each row contains the model coefficient estimate and the p-value obtained for each metabolic variable. When output = "models", the function returns a list, each list element containing detailed information about the statistic model fitted for each metabolic variable.

References

Benjamini Y, Hochberg Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society Series B, 57, 289–300.

Benjamini Y, Yekutieli D. (2001). The control of the false discovery rate in multiple testing under dependency. Annals of Statistics, 29, 1165–1188.

Dobson AJ. (1990). An Introduction to Generalized Linear Models. London: Chapman and Hall.

Kim, S. (2015). ppcor: An R Package for a Fast Calculation to Semi-partial Correlation Coefficients. Communications for Statistical Applications and Methods, 22, 665-674.

Holm S. (1979). A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics, 6, 65–70.

Hommel G. (1988). A stagewise rejective multiple test procedure based on a modified Bonferroni test. Biometrika, 75, 383–386.

Hochberg Y. (1988). A sharper Bonferroni procedure for multiple tests of significance. Biometrika, 75, 800–803.

Shaffer JP. (1995). Multiple hypothesis testing. Annual Review of Psychology, 46, 561–576.

Storey JD. (2002). A direct approach to false discovery rates. Journal of the Royal Statistical Society, Series B, 64: 479-498.

Venables WN, Ripley BD. (2002). Modern Applied Statistics with S. New York: Springer.

Examples

```
## Load data
data(metabo_SE)
data(targetMetabo_SE)

## Test for association between BMI and metabolic_data
BMI_model <- MWAS_stats (metabo_SE, disease_id = "BMI", assoc_method = "spearman",
                         mt_method = "BH", output = "pvalues")

## Test for association between diabetes and target_metabolites (age-gender adjusted)
T2D_model <- MWAS_stats (targetMetabo_SE, disease_id = "T2D",
                         confounder_ids = c("Age", "Gender"),
                         assoc_method = "logistic", mt_method = "BY",
                         output = "pvalues")
```

MWAS_SummarizedExperiment

Create a SummarizeExperiment object

Description

This function formats the metabolic and clinical data into a SummarizedExperiment object.

Usage

```
MWAS_SummarizedExperiment(metabo_matrix, clinical_matrix, sample_type)
```

Arguments

metabo_matrix	numeric matrix containing the metabolic data (e.g. NMR peak intensities or metabolite concentrations). The columns of the matrix must correspond to the metabolic variables and the rows to the samples. Column and row names must contain the metabolite IDs (e.g. chemical shifts for NMR data) and the sample IDs, respectively.
clinical_matrix	numeric matrix containing the clinical data (e.g. age, gender). The columns of the matrix must correspond to the phenotypic variables and the rows to the samples. Column and row names must contain the phenotype IDs and the sample IDs, respectively. For samples without clinical data (e.g. quality control (QC) samples), NA values must be used.
sample_type	numeric vector indicating sample type (i.e. experimental sample or QC sample). The vector must be coded as follows: experimental sample = 0, QC sample = 1. If QC samples are not available, all the elements of this vector must be 0.

Value

A SummarizedExperiment object.

References

Morgan M, et al. (2016). SummarizedExperiment: SummarizedExperiment container. R package.

Examples

```
## Load data
data(metabo_SE)

## Get metabolic_data, clinical_data, and sample_type
library(SummarizedExperiment)
metabolic_data = t(assays(metabo_SE)$metabolic_data)
clinical_data = as.matrix(colData(metabo_SE)[, -5])
sample_type = as.vector(colData(metabo_SE)[, 5])

## Reconstruct SummarizedExperiment
data_SE = MWAS_SummarizedExperiment(metabolic_data, clinical_data,
                                    sample_type)
```

---

plot_spectraNMR *Plot NMR spectra*

Description

This function generates an NMR spectra plot, with the chemical shifts displayed along the x-axis, and the peak intensities displayed on the y-axis.

Usage

```
plot_spectraNMR (metabo_SE, type = "l", lty = 1, xlab ="ppm",
                 ylab ="intensity", xlim = NULL, ...)
```

Arguments

metabo_SE	SummarizedExperiment object. See "MWAS_SummarizedExperiment()".
type	character vector indicating the type of plot for each row of metabo_matrix. For all possible types, see "plot()".
lty	character vector of line types. For all possible types, see "plot()".
xlab	character vector specifying a title for the x-axis.
ylab	character vector specifying a title for the y-axis.
xlim	numeric vector containing the minimum and maximum values of the x axis. Notice that ppm is displayed in reverse scale (e.g. xlim = c(10, 0)).
...	other arguments passed to "matplot()".

Value

An NMR spectra plot.

Examples

```
## Load data
data(metabo_SE)

## Plot first 2 spectra
plot_spectraNMR (metabo_SE[, 1:2])
plot_spectraNMR (metabo_SE[, 1:2], xlim = c(1.03, 0.85), main = "NMR spectra")
```

QC_CV *Calculate coefficients of variation*

Description

This function calculates the coefficient of variation (CV) (|sd/mean|) of each metabolic feature across the quality control (QC) samples. The CV distribution is represented in a histogram. This function can be used to assess the reproducibility of individual metabolic features. Notice that CV = 0.30 and CV = 0.15 are the thresholds established by the FDA guidelines for biomarker discovery and quantification, respectively.

Usage

```
QC_CV (metabo_SE, CV_th = 0.30, plot_hist = TRUE, hist_bw = 0.005,
       hist_col = "moccasin", size_lab = 12, size_axis = 12)
```

Arguments

metabo_SE	SummarizedExperiment object. See "MWAS_SummarizedExperiment()".
CV_th	numeric value indicating the CV threshold.
plot_hist	logical constant indicating whether a histogram showing CV distribution will be plotted.
hist_bw	numeric value indicating histogram bin width.
hist_col	character string indicating the color to be used to fill the histogram bars.
size_lab	numeric value indicating the font size of x- and y-axis titles.
size_axis	numeric value indicating the font size of x- and y-axis labels.

Value

A numeric vector containing the CV of each metabolic feature and a histogram showing CV distribution. In the histogram, CVs above 1 are set to 1.

References

Dumas ME, et al. (2006). Assessment of analytical reproducibility of 1H NMR spectroscopy based metabonomics for large-scale epidemiological research: the INTERMAP Study. Analytical Chemistry, 78, 2199-1208.

Examples

```
## Load data
data(metabo_SE)

## Calculate CVs
metabo_CV <-  QC_CV (metabo_SE)
metabo_CV2 <-  QC_CV (metabo_SE, hist_bw = 0.008, hist_col = "lightblue")
```

QC_CV_scatterplot *Plot MS-based scatter plot colored based on CV*

Description

This function creates a MS-based scatter plot (rt vs mz) where the metabolic features are colored based on their coefficient of variation (CV). See "QC_CV()".

Usage

```
QC_CV_scatterplot (rt, mz, CV_metabo, CV_th = 0.30, xlab = "rt",
                   ylab = "mz", pch = 20, marker_size = 1, xlim = NULL,
                   ylim = NULL, size_axis = 10, size_lab = 10)
```

Arguments

rt	numeric vector of retention time values.
mz	numeric vector of mz values.
CV_metabo	numeric vector containing the CV of each metabolic feature. The length of this vector should be consistent with the length of rt and mz.
CV_th	numeric value indicating the CV threshold. NMR signals with CV equal or above CV_th will be colored in red.
xlab	character vector specifying a title for the x-axis.
ylab	character vector specifying a title for the y-axis.
pch	value specifying the symbol used to represent each MS feature in the scatter plot. To see all possible symbols, check "plot()" options.
marker_size	numeric value indicating the size of the symbol used to represent each metabolic feature in the scatter plot.
xlim	numeric vector containing the minimum and maximum values of the x-axis.
ylim	numeric vector containing the minimum and maximum values of the y-axis.
size_axis	numeric value indicating the font size of x- and y-axis title.
size_lab	numeric value indicating the font size of x- and y-axis labels.

Value

A MS-based scatter plot where MS features are represented according on CV.

Examples

```
## Load data
data(MS_data)
rt <- MS_data[, 1]
mz <- MS_data[, 2]

## Simulate CV values
CV_metabo <- runif(length(rt), 0.05, 0.31)

## MS-based scatter plot
QC_CV_scatterplot(rt, mz, CV_metabo)
QC_CV_scatterplot(rt, mz, CV_metabo, xlim = c(0, 10))
QC_CV_scatterplot(rt, mz, CV_metabo, CV_th = 0.15)
```

QC_CV_specNMR *Plot NMR spectrum colored based on CV*

Description

This function allows plotting a reference NMR spectrum colored based on the coefficient of variation (CV) of each NMR signal. See function "QC_CV()".

Usage

```
QC_CV_specNMR (metabo_SE, ref_sample,  CV_th = 0.30, xlab = "ppm",
               ylab = "intensity", size_axis = 12, size_lab = 12, xlim = NULL,
               ylim = NULL, xbreaks = waiver (), xnames = waiver (),
               ybreaks = waiver (), ynames = waiver ())
```

Arguments

metabo_SE	SummarizedExperiment object. See "MWAS_SummarizedExperiment()".
ref_sample	character vector indicating the ID of the sample that will be used to plot the NMR spectrum.
CV_th	numeric value indicating the CV threshold. NMR signals with CV equal or above CV_th will be colored in red.
xlab	character vector specifying a title for the x-axis.
ylab	character vector specifying a title for the y-axis.
size_axis	numeric vector indicating the font size of x- and y-axis labels.
size_lab	numeric vector indicating the font size of x- and y-axis tittles.
xlim	numeric vector containing the minimum and maximum values of the x-axis. Notice that ppm is displayed in reverse scaled (e.g. xlim = c(10, 0)).
ylim	numeric vector containing the minimum and maximum values of the y-axis.
xbreaks	numeric vector indicating the positions of the breaks of the x-axis.
xnames	character vector (same length as xbreaks) containing the labels of each break of the x-axis.
ybreaks	numeric vector indicating the positions of the breaks of the y-axis.
ynames	character vector (same length as ybreaks) containing the labels of each break of the y-axis.

Value

An NMR spectrum plot colored based on the CV of each NMR signal.

References

Dumas ME, et al. (2006). Assessment of analytical reproducibility of 1H NMR spectroscopy based metabonomics for large-scale epidemiological research: the INTERMAP Study. Analytical Chemistry, 78, 2199-1208.

Examples

```
## Load data
data(metabo_SE)

## Plot NMR spectrum colored by CV
QC_CV_specNMR (metabo_SE, ref_sample = "QC1", CV_th = 0.30)
QC_CV_specNMR (metabo_SE, ref_sample = "QC1", CV_th = 0.30, xlim = c(1.1, 0.95))
QC_CV_specNMR (metabo_SE, ref_sample = "QC1", CV_th = 0.15)
```

QC_PCA *Principal Component Analysis*

Description

This function performs PCA on a matrix of metabolic data and returns the results as an object of class "prcomp". When quality control (QC) samples are available, "QC_PCA()" can be used to assess the stability and reproducibility of the dataset.

Usage

```
QC_PCA (metabo_SE, scale = FALSE, center = TRUE,...)
```

Arguments

metabo_SE	SummarizedExperiment object. See "MWAS_SummarizedExperiment()".
scale	logical constant indicating whether the metabolic variables will be scaled to have unit variance before the analysis. For more details, check "prcomp()".
center	logical constant indicating whether the metabolic variables will be shifted to be zero-centered before the analysis. For more details, check "prcomp()".
...	other arguments passed to "prcomp()".

Value

A list with class "prcomp". For more details, check "prcomp()".

References

Mardia K, et al. (1979). Multivariate Analysis, London: Academic Press.

Examples

```
## Load data
data(metabo_SE)
data(targetMetabo_SE)


## PCA model using all metabolic data
PCA_model <- QC_PCA (metabo_SE)

## PCA model using target metabolites
PCA_subset <- QC_PCA (targetMetabo_SE)
```

`QC_PCA_scoreplot` *PCA score plot*

Description

This function generates a PCA score plot colored based on sample type (i.e. experimental or quality control (QC) sample). The plots generated with this function can be used to assess analytical reproducibility and stability. If the dataset is reproducible, all quality control samples should appear clustered in the center of the Hotelling's ellipse.

Usage

```
QC_PCA_scoreplot (PCA_model, metabo_SE, plot_labels = FALSE,  px = 1, py = 2,
                  CI_level = 0.95, pch = 20, xlim = NULL, ylim = NULL,
                  color_scale = c("cornflowerblue", "red"), grid = TRUE,...)
```

Arguments

`PCA_model`	"prcomp" object generated by the function "QC_PCA()".
`metabo_SE`	SummarizedExperiment object. See "MWAS_SummarizedExperiment()".
`plot_labels`	logical constant indicating whether the sample IDs will be displayed in the score plot.
`px`	numeric value indicating the index of the principal component that will be displayed on the x-axis.
`py`	numeric value indicating the index of the principal component that will be displayed on the y-axis.
`CI_level`	numeric value indicating the confidence interval for the Hotelling's ellipse.
`pch`	value specifying the symbol that will represent each sample in the score. To see all possible symbols, check "plot()" options.
`xlim`	numeric vector containing the minimum and maximum values of the x-axis.
`ylim`	numeric vector containing the minimum and maximum values of the y-axis.
`color_scale`	character vector corresponding to the 2-color scale that will be used to discriminate the experimental samples from the QC samples.
`grid`	logical constant indicating whether grid lines will be added to the plot.
`...`	other arguments passed to "plot()".

Value

A PCA score plot.

References

Fox J, Weisberg S. (2011). An R Companion to Applied Regression, Second Edition, Sage.

Mardia K, et al. (1979). Multivariate Analysis, London: Academic Press.

Examples

```
## Load data
data(metabo_SE)

## PCA model
PCA_model <- QC_PCA (metabo_SE)

## PCA score plots
QC_PCA_scoreplot (PCA_model, metabo_SE) # PC1 vs PC2
QC_PCA_scoreplot (PCA_model, metabo_SE, px = 3, py = 4) # PC3 vs PC4
QC_PCA_scoreplot(PCA_model, metabo_SE, plot_labels = TRUE) # show labels
QC_PCA_scoreplot (PCA_model, metabo_SE, CI_level = 0.80) # change CI
```

STOCSY_NMR *Statistical Total Correlation Spectroscopy - Academic use only*

Description

This function calculates STOCSY between an NMR signal of interest and all the NMR variables, representing a useful tool for NMR molecular identification and assignment. The results are represented in a pseudo-NMR spectrum displaying the covariance (height) and the Pearson/Spearman correlation coefficient (color) of all spectral variables with the variable of interest (driver signal).

Usage

```
STOCSY_NMR (metabo_SE, ppm_query, cor_method = "pearson", alpha_th = 0.05,
            xlab = "ppm", ylab = "covariance", size_lab = 12, size_axis = 12,
            xlim = NULL, ylim = NULL, xbreaks = waiver(), xnames = waiver(),
            ynames = waiver(), ybreaks = waiver())
```

Arguments

metabo_SE	SummarizedExperiment object. See "MWAS_SummarizedExperiment()".
ppm_query	numeric value (at least 2 decimals) corresponding to the driver ppm.
cor_method	character vector specifying the correlation method("pearson" or "spearman").
alpha_th	numeric value indicating the significance threshold. NMR variables with BH-adjusted p-value equal or above this threshold will be neglected.
xlab	character vector specifying a title for the x-axis.
ylab	character vector specifying a title for the y-axis.
size_lab	numeric value indicating the font size of x- and y- axis titles.
size_axis	numeric value indicating the font size of x- and y- axis labels.
xlim	numeric vector containing the minimum and maximum values of the x-axis. Notice that ppm is displayed in reverse scale (e.g. xlim = c(2, 1)).
ylim	numeric vector containing the minimum and maximum values of the y-axis.

xbreaks	numeric vector indicating the positions of the breaks of the x-axis.
xnames	character vector (same length as xbreaks) containing the labels of each break of the x-axis.
ybreaks	numeric vector indicating the positions of the breaks of the y-axis.
ynames	character vector (same length as ybreaks) containing the labels of each break of the y-axis.

Value

A plot displaying the Pearson correlation coefficient (color) and covariance (height) between all spectral variables and the driver signal.

References

Cloarec O, et al.(2005). Statistical total correlation spectroscopy: An exploratory approach for latent biomarker identification from metabolic 1H NMR data sets. Analytical Chemistry, 77, 1282???1289.

Nicholson JK, et al. (2005). Method for the identification of molecules and biomarkers using chemical, biochemical and biological data. US 20070043518 A1

Examples

```
## Load data
data(metabo_SE)

## STOCSY using 1.04 as driver signal
STOCSY_NMR (metabo_SE, ppm_query = 1.04)
STOCSY_NMR (metabo_SE, ppm_query = 1.04, alpha_th = 0, xlim = c(1.06, 1))
```

---

targetMetabo_SE *Target NMR metabolites dataset*

Description

This SummarizedExperiment object contains the following information:

-An assay matrix containing the levels of 8 targeted 1H NMR metabolites (lactate, 3-hydroxybutyrate, leucine, valine, isoleucine, acetate, alanine and 1,5-anhydroglucitol) across the experimental samples and the quality control (QC) samples.

-A data.frame containing clinical information (age, gender, type II diabetes status and BMI) and sample class (i.e. experimental sample or QC sample) information for each sample row in the assay matrix.

Usage

```
data(targetMetabo_SE)
```

Format

SummarizedExperiment

Value

SummarizedExperiment

Index

CV_filter, 2

JBA_binning, 3
JBA_corDistribution, 5
JBA_plotBins, 6

KEGG_metabolic_paths, 7

metabo_SE, 7
MS_data, 8
MWAS_barplot, 8
MWAS_bootstrapping, 10
MWAS_filter, 11
MWAS_heatmap, 12
MWAS_KEGG_network, 13
MWAS_KEGG_pathways, 14
MWAS_KEGG_shortestpaths, 15
MWAS_network, 17
MWAS_scatterplotMS, 18
MWAS_skylineNMR, 20
MWAS_stats, 22
MWAS_SummarizedExperiment, 23

plot_spectraNMR, 24

QC_CV, 25
QC_CV_scatterplot, 26
QC_CV_specNMR, 27
QC_PCA, 29
QC_PCA_scoreplot, 30

STOCSY_NMR, 31

targetMetabo_SE, 32