Package 'MACP'

February 28, 2023

Type Package

Title Macromolecular Assemblies from Co-Elution Profile (MACP)

Version 0.1.0

Description The MACP employs machine learning algorithm for automated scoring of co-fractionation mass spectrometry (CF-MS) and then systematically map multi-protein complexes from these high-confidence protein-protein interactions (PPIs) using unsupervised learning (i.e., clustering).

Depends R (>= 4.1)

Imports stats, zoo, utils, dplyr, lsa, WGCNA, tidyr, tibble, Hmisc, igraph, PRROC, pROC, ggplot2, grDevices, fmsb, stringr, caret

Maintainer Matineh Rahmatbakhsh <matinerb.94@gmail.com>

License BSD_3_clause + file LICENSE

Encoding UTF-8

URL https://github.com/mrbakhsh/MACP

BugReports https://github.com/mrbakhsh/MACP/issues

VignetteBuilder knitr

Suggests knitr, ptw, e1071, kernlab, ranger, proxy, infotheo, gridExtra, philentropy, randomForest, gprofiler2, purrr, minet, entropy, MCL, orthogene, protti, arules, rmarkdown, BiocStyle

RoxygenNote 7.2.3

NeedsCompilation no

Author Matineh Rahmatbakhsh [aut, trl, cre], Mohan Babu [led]

Repository CRAN

Date/Publication 2023-02-28 17:32:30 UTC

R topics documented:

`calculate_PPIscore` *Calculate Pairwise Protein Profile Similarity using Different Metrics*

Description

This function first removes proteins pairs for which two proteins never occurred in the same fractions, then computes pairwise protein similarity using up to 18 metrics (by default all the 18 measures are activated). This function also provides users with an option to choose an appropriate co-fractionation correlation score cut-off using the 'corr_cutoff' argument, if argument 'corr_removal' is set to TRUE.

Usage

```
calculate_PPIscore(
  x,
  pcc = TRUE,
  PCCN = TRUE,
  pcc_p = TRUE,
  spearman = TRUE,
  kendall = TRUE,
  bicor = TRUE,
  weighted_rank = TRUE,
```

```
  cosine = TRUE,
  jaccard = TRUE,
  dice = TRUE,
  apex = TRUE,
  minfo = TRUE,
  bayesian = TRUE,
  wcc = TRUE,
  euclidean = TRUE,
  manhattan = TRUE,
  canberra = TRUE,
  avg.distance = TRUE,
  rept = 10,
  corr_removal = FALSE,
  corr_cutoff = 0.5
)
```

Arguments

x	A co-elution data matrix with proteins in rows and fractions in columns.
pcc	If TRUE, computes pairwise protein profile similarity using Pearson correlation metric.
PCCN	If TRUE, computes pairwise protein profile similarity using Pearson correlation plus noise.This function is adapted from the PCCN function in the SMED package.
pcc_p	If TRUE, computes P-value of the Pearson correlation.
spearman	if TRUE, computes pairwise protein profile similarity using spearman correlation.
kendall	if TRUE, computes pairwise protein profile similarity using kendall correlation.
bicor	if TRUE, computes pairwise protein profile similarity using biweight midcorrealtion (bicor) correlation.
weighted_rank	if TRUE, computes pairwise protein profile similarity using weighted rank measure.
cosine	If TRUE, computes pairwise protein profile similarity using cosine metric.
jaccard	If TRUE, computes pairwise protein profile similarity using jaccard metric.
dice	if TRUE, computes pairwise protein profile similarity using dice measure.
apex	If TRUE, computes pairwise protein profile similarity using apex.
minfo	If TRUE, computes pairwise protein profile similarity using mutual information.
bayesian	If TRUE, computes pairwise protein profile similarity using Bayes correlation based on zero-count distribution.
wcc	If TRUE, computes pairwise protein profile similarity using weighted cross correlation.
euclidean	if TRUE, computes pairwise protein profile similarity using euclidean measure.
manhattan	if TRUE, computes pairwise protein profile similarity using manhattan measure.
canberra	if TRUE, computes pairwise protein profile similarity using canberra measure.

avg.distance	if TRUE, computes pairwise protein profile similarity using avg.distance measure.
rept	Poisson iterations, defaults to 10.
corr_removal	If TRUE, removes protein pairs with correlation scores < the user defined threshold ; defaults to FALSE.
corr_cutoff	user defined threshold for correlation similarity scores. Defaults to 0.5.

Details

calculate_PPIscore

Value

A data frame containing the calculated features for all possible protein pairs.

Author(s)

Matineh Rahmatbakhsh, <matinerb.94@gmail.com>

Examples

```
M1<-matrix(rnorm(36),nrow=6)
M1 <- abs(M1)
rownames(M1) <- c("A","B","C","D","E","F")
scored_Data <- calculate_PPIscore(M1)
```

---

cluster_tuning *ClusterONE Hyperparameters Tuning*

Description

This function optimizes the choice of ClusterONE algorithm parameters such as density, node penalty, and overlap score by comparing clustering-derived partitions for each combination of parameters to known labels (i.e., CORUM complexes) and assess the similarity between them using quality measures including overlap score, sensitivity (Sn), clustering-wise positive predictive value (PPV), geometric accuracy (Acc), and maximum matching raio (MMR).It is recommended to first reduce redundancy in the known reference complexes via EliminateCpxRedundance,then performs parameter tuning.

Usage

```
cluster_tuning(
  refcpx,
  csize = 2,
  d = c(0.3, 0.5),
  p = c(2),
  max_overlap = c(0.5, 0.6),
  tpath = file.path(system.file("extdata", package = "MACP"))
)
```

Arguments

refcpx	A list containing reference complexes (i.e., corum complexes).
csize	An integer, the minimum size of the predicted complexes. Defualts to 2.
d	A vector of number, density of predicted complexes.
p	A vector of integer, penalty value for the inclusion of each node.
max_overlap	A vector of number, specifies the maximum allowed overlap between two clusters.
tpath	A character string indicating the path to the project directory that contains the interaction data. Interaction data must be stored as .txt file and containing id1-id2-weight triplets. Defaults to MACP/inst/extdata directory.

Details

cluster_tuning

Value

A data.frame containing clustering performance across different combination of parameters.

Author(s)

Matineh Rahmatbakhsh, <matinerb.94@gmail.com>

References

Nepusz, T., Yu, H., and Paccanaro, A. (2012a). Detecting overlapping protein complexes in protein-protein interaction networks. Nat. Methods 9, 471.

Clust_Valid *Cluster Evaluation by External Measures*

Description

This function evaluate the quality of clusters by comparing clustering-derived partitions to known labels (i.e., CORUM complexes) and assess the similarity between them using quality measures including overlap score (O), sensitivity (Sn), clustering-wise positive predictive value (PPV), geometric accuracy (Acc), and maximum matching ratio (MMR).

Usage

```
Clust_Valid(predcpx, refcpx)
```

Arguments

predcpx	A list containing predicted complexes.
refcpx	A list containing reference complexes (i.e., CORUM complexes).

Details

Clust_Valid

Value

A list containing the numerical values for each evaluation metrics.

Author(s)

Matineh Rahmatbakhsh, <matinerb.94@gmail.com>

Examples

```
# Load known reference complexes
data(refcpx)
# Select subset of complexes to be used as an instance sets for predicted
# complexes
predcpx <- refcpx[5:15]
Eval_result <- Clust_Valid(predcpx,refcpx)
```

---

data_filtering *Data Filtering*

Description

This function removes proteins for which peptide only detected in one fraction (i.e., "one-hit-wonders") across the co-elution table, common contaminants (e.g., keratins) only for mouse and human organisms and frequent flyers.

Usage

```
data_filtering(x)
```

Arguments

x	A data matrix object with rows including proteins and fractions along the columns.

Details

data_filtering

Value

Filtered matrix.

Author(s)

Matineh Rahmatbakhsh, <matinerb.94@gmail.com>

Examples

```
# Load the co-elution data
data("exampleData")
# Perform raw data pre-processing
datOut <- data_filtering(exampleData)
```

`EliminateCpxRedundance` *Hierarchical Clustering of Modules*

Description

This function reduces redundancy in the reference complexes by first computing the overlap of two complexes via Jaccard index, followed by merging overlapping complexes with user-defined threshold (here is 0.2).

Usage

```
EliminateCpxRedundance(
  rawCpx,
  custom_bg = NULL,
  sim_method = "euclidean",
  linkage = "average",
  h = 0.2
)
```

Arguments

`rawCpx`	A list containing protein complexes
`custom_bg`	Vector of proteins names to use as a background. If given, refcpx will be first mapped to the background proteisn, followed by removing redundancy in the refcpx.
`sim_method`	c(euclidean", "maximum", "manhattan", "canberra", "binary", or "minkowski"); Default is euclidean
`linkage`	c("average", "ward", "single", "complete", "mcquitty", "median", "centroid"); Default is average.
`h`	numeric scalar or vector with heights where the tree should be cut; Defaults to 0.2

Details

EliminateCpxRedundance

Value

List of unique complexes.

Author(s)

Matineh Rahmatbakhsh

Examples

```
# predicted interactions
pred_ppi <- read.table(
system.file("extdata/ppi_input_ClusterONE.txt", package = "MACP"),
header = FALSE)
# get all the proteins in the predicted network
custom_bg <- union(pred_ppi$V1, pred_ppi$V2)
# reference complexes
data("refcpx")
# reduce redundancy in reference complexes
filt_cpx <- EliminateCpxRedundance(refcpx,
custom_bg,
sim_method = "euclidean",
linkage="average",
h = 0.2)
```

`enrichmentCPX` *Functional Enrichment Analysis for Predicted Complexes*

Description

This function uses `gost` function in `gprofiler2` package to perform functional enrichment analysis for predicted modules.

Usage

```
enrichmentCPX(
  predcpx,
  threshold = 0.05,
  sources = c("GO", "KEGG", "CORUM", "REAC", "CORUM"),
  p.corrction.method = "bonferroni",
  custom_bg = NULL,
  org = "mmusculus"
)
```

Arguments

`predcpx`	A data.frame containing predicted complexes resulted from `get_clusters` or `MCL_clustering`.
`threshold`	Custom p-value threshold for significance.
`sources`	A vector of data sources to use. See `gost` for more details.
`p.corrction.method`	The algorithm used for multiple testing correction;defaults to 'bonferroni'. See `gost` for more details.

custom_bg	vector of gene names to use as a statistical background. Defaults to NULL.
org	An organism name;defaults to 'mmusculus'. See gost for more details.

Details

enrichmentCPX

Value

A data.frame with the enrichment analysis results.

Author(s)

Matineh Rahmatbakhsh, <matinerb.94@gmail.com>

---

ensemble_model *Predict Interactions via Ensemble Learning Method*

Description

This function uses individual or an ensemble of classifiers to predict interactions from CF-MS data. This ensemble algorithm combines different results generated from individual classifiers within the ensemble via average to enhance prediction.

Usage

```
ensemble_model(
  features,
  gd,
  classifier = c("glm", "svmRadial", "ranger"),
  cv_fold = 2,
  verboseIter = TRUE,
  plots = FALSE,
  filename = file.path(tempdir(), "plots.pdf")
)
```

Arguments

features	A data frame with protein-protein associations in the first column, and features to be passed to the classifier in the remaining columns.
gd	A gold reference set including true associations with class labels indicating if such PPIs are positive or negative.
classifier	The type of classifier to use. See caret for the available classifiers.
cv_fold	Number of partitions for cross-validation; defaults to 5.
verboseIter	Logical value, indicating whether to check the status of training process;defaults to FALSE.

`plots` Logical value, indicating whether to plot the performance of ensemble learning algorithm as compared to individual classifiers; defaults to FALSE.If the argument set to TRUE, plots will be saved in the current working directory. These plots are :

- pr_plot - Precision-recall plot of ensemble classifier vs selected individual classifiers.
- roc_plot - ROC plot of ensemble classifier vs selected individual classifiers.
- points_plot - Plot accuracy, F1-score ,positive predictive value (PPV),sensitivity (SE), and Matthews correlation coefficient (MCC) of ensemble classifier vs selected individual classifiers.

.

`filename` character string, indicating the location and output pdf filename for for performance plots. Defaults to tempdir().

Details

ensemble_model

Value

Ensemble_training_output

- prediction score - Prediction scores for whole dataset from each individual classifier.
- Best - Selected hyper parameters.
- Parameter range - Tested hyper parameters.
- prediction_score_test - Scores probabilities for test data from each individual classifier.
- class_label - Class probabilities for test data from each individual classifier.

classifier_performance

- cm - A confusion matrix.
- ACC - Accuracy.
- SE - Sensitivity.
- SP - Specificity.
- PPV - Positive Predictive Value.
- F1 - F1-score.
- MCC - Matthews correlation coefficient.
- Roc_Object - A list of elements. See `roc` for more details.
- PR_Object - A list of elements. See `pr.curve` for more details.

predicted_interactions - The input data frame of pairwise interactions, including classifier scores averaged across all models.

Author(s)

Matineh Rahmatbakhsh, <matinerb.94@gmail.com>

exampleData *Demo CF-MS data*

Description

Co-elution profiles derived from mitochondrial (mt) extracts of mouse brain culture,fractionated by (size-exclusion chromatography, SEC)

Usage

```
data(exampleData)
```

Format

A matrix with 284 rows and 83 columns, with proteins in rows and biochemical fractions in columns

generate_refInt *Generate Class Labels for Data Input Based on Gold Reference Set*

Description

This function creates class labels for protein pairs in the same order in the data input based on gold reference set.

Usage

```
generate_refInt(x, refcpx)
```

Arguments

x	A data frame with interacting proteins in the first two columns.
refcpx	A list containing gold standard protein complexes.

Details

generate_refInt

Value

A Data frame containing class labels for protein pairs in the data input. If protein pairs involve in same protein complexes are assigned to Positive, otherwise Negative.

Author(s)

Matineh Rahmatbakhsh, <matinerb.94@gmail.com>

getCPX *Fetch Complexes from the CORUM Database*

Description

This function retrieves protein complexes directly from the CORUM database.

Usage

```
getCPX(org = "Mouse", tpath = tempdir())
```

Arguments

org	Mammalian model organisms including: "Human", "Mouse", "Pig", "Bovine", "Rat", "Mammalia", "Rabbit", "Dog", "Hamster", and "MINK". Defaults to "Mouse".
tpath	A character string indicating the path to the project directory. If the directory is missing, it will be stored in the temp directory.

Details

getCPX

Value

A list containing protein complexes for mammalian organisms.

Author(s)

Matineh Rahmatbakhsh, <matinerb.94@gmail.com>

get_clusters *Predict Complexes*

Description

This function partitions high-confidence network to putative complexes via ClusterONE clustering algorithm to identify protein complex membership.

Usage

```
get_clusters(
  csize = 2,
  d = 0.3,
  p = 2,
  max_overlap = 0.8,
  tpath = file.path(system.file("extdata", package = "MACP"))
)
```

Arguments

csize	An integer, the minimum size of the predicted complexes. Defaults to 2.
d	A number, density of predicted complexes. Defaults to 0.3.
p	An integer, penalty value for the inclusion of each node. Defaults to 2.
max_overlap	A number, specifies the maximum allowed overlap between two clusters. Defaults to 0.8.
tpath	A character string indicating the path to the project directory that contains the interaction data. Interactions data must be stored as .txt file and containing id1-id2-weight triplets.

Details

get_clusters

Value

A data.frame containing predicted complexes

Author(s)

Matineh Rahmatbakhsh, <matinerb.94@gmail.com>

References

Nepusz, T., Yu, H., and Paccanaro, A. (2012a). Detecting overlapping protein complexes in protein-protein interaction networks. Nat. Methods 9, 471.

Examples

```
predcpx <-
get_clusters(csize = 3, d = 0.3, p = 2,
max_overlap = 0.8,
tpath = file.path(system.file("extdata", package = "MACP")))
```

---

get_DenoisedNet *Denoising Predicted Protein-Protein Interactions*

Description

This function removes the noise in the form of false positive edges in the predicted networks using network topology.

Usage

```
get_DenoisedNet(ppi)
```

Arguments

ppi	Interactions data containing id1-id2-weight triplets.

Details

get_DenoisedNet

Value

A data.frame containing denoised network.

Author(s)

Matineh Rahmatbakhsh, <matinerb.94@gmail.com>

Examples

```
# high-confidence network as input
ppi <-
read.table(system.file("extdata/ppi_input_ClusterONE.txt",
package="MACP"),
quote="\"", comment.char="")
# Perform network denoising
denoisetNet <- get_DenoisedNet(ppi)
```

---

impute_MissingData *Impute missing Values in Elution Profile Matrix*

Description

This function imputes missing values in protein elution profile matrix via average of adjacent rows. This function is not applicable for missing values present in the first or last column.

Usage

```
impute_MissingData(x)
```

Arguments

x	A data matrix with rows including proteins and fractions along the columns, while some fractions may contain missing values.

Details

impute_MissingData

Value

Imputed matrix.

Author(s)

Matineh Rahmatbakhsh, <matinerb.94@gmail.com>

Examples

```
# Load the co-elution data
data("exampleData")
# Replace the values with NAs in the 10th column
exampleData[, 10] <- NA
# Impute missing value
datOut <- impute_MissingData(exampleData)
```

---

keepMT *Keep Mitochondrial (mt) Proteins*

Description

This function removes all the non-mitochondrial proteins by mapping the co-eluted proteins from chromatography fractions to MitoCarta database. Note that this function is only applicable to mouse or human organisms.

Usage

```
keepMT(x)
```

Arguments

x	A data matrix object with rows including proteins and fractions along the columns.

Details

keepMT

Value

Matrix containing mt proteins.

Author(s)

Matineh Rahmatbakhsh, <matinerb.94@gmail.com>

Examples

```
# Load the co-elution data
data("exampleData")
# Removes non-mitochondtial proteins
datOut <- keepMT(exampleData)
```

MCL_clustering *MCL clustering*

Description

This function applies MCL clustering to further refine the predicted subnetworks produced by ClusterONE.

Usage

```
MCL_clustering(hc_ppi, predcpx, inflation = 9, csize = 2)
```

Arguments

hc_ppi	High-confidence interactions data containing id1-id2-weight triplets.
predcpx	A data.frame containing predicted complexes resulted from get_clusters.
inflation	MCL inflation parameter. Defaults to 9.
csize	An integer, the minimum size of the predicted complexes. Defaults to 2.

Details

MCL_clustering

Value

List of refined complexes.

Author(s)

Matineh Rahmatbakhsh

Examples

```
# open high-confidence network
hc_ppi <-
read.delim(
system.file("extdata/ppi_input_ClusterONE.txt", package = "MACP"),
header = FALSE)
# predict complexes by ClusterONE
predcpx <-
get_clusters(csize = 3, d = 0.3, p = 2,
max_overlap = 0.8,
tpath = file.path(system.file("extdata", package = "MACP")))
# Break down big complexes by MCL
MCL_clusters <- MCL_clustering(hc_ppi, predcpx, inflation = 4, csize = 2)
```

MCL_tuning *MCL Hyperparameters Tuning*

Description

This function optimize the choice of MCL algorithm parameter (inflation) by comparing clustering-derived partitions for each paramter values to known labels (i.e., CORUM complexes) and assess the similarity between them using quality measures including overlap score, sensitivity (Sn), clustering-wise positive predictive value (PPV), geometric accuracy (Acc), and maximum matching raio (MMR). It is recommended to first reduce redundancy in the known reference complexes via `EliminateCpxRedundance`, then performs parameter tuning.

Usage

```
MCL_tuning(hc_ppi, predcpx, refcpx, inflation = c(6, 8, 9), csize = 2)
```

Arguments

`hc_ppi`	Interactions data containing id1-id2-weight triplets.
`predcpx`	A data.frame containing predicted modules resulted from `get_clusters`.
`refcpx`	A list containing reference complexes (i.e., corum complexes).
`inflation`	A vector of integer, representing MCL inflation parameter
`csize`	An integer, the minimum size of the predicted complexes. Defaults to 2.

Details

MCL_tuning

Value

A data.frame containing clustering performance across different inflation values.

Author(s)

Matineh Rahmatbakhsh, <matinerb.94@gmail.com>

orthMappingCpx *Protein Complex Ortholog Mapping*

Description

This function uses convert_orthologs function to support ortholog mapping of protein complexes between any pair of 700+ species.

Usage

```
orthMappingCpx(
  datInput,
  input_species,
  output_species,
  input_taxid,
  output_taxid
)
```

Arguments

datInput	A list containing reference complexes (i.e., CORUM complexes). Note that the members of each complexes must be represented by UniProt accession identifier.
input_species	Name of the input species (e.g., "mouse","fly"). See map_species to return a full list of available species.
output_species	Name of the output species (e.g., "human"). See map_species to return a full list of available species.
input_taxid	A numeric value that specifies the NCBI taxonomy identifier (TaxId) for input organism (e.g., 10090).
output_taxid	A numeric value that specifies the NCBI taxonomy identifier (TaxId) for output organism.

Details

orthMappingCpx

Value

A list containing complexes, whose members converted to output_species.

Author(s)

Matineh Rahmatbakhsh, <matinerb.94@gmail.com>

predPPI_MACP *Predict Protein-Protein Interactions and Putative Complexes*

Description

This function first begins by executing several pre-processing steps to improve the quality of the raw data, followed by computing similarity between protein pairs using their co-elution profiles. Computed features and and class labels generated from reference complexes are then fed into an individual or ensemble of ML classifiers.These models then generate a weighted protein interaction network in which edge weights between protein nodes represent the ML model's probability estimate for interaction. High-confidence PPIs resulted from ROC-curve cutoff analysis is then denoised and finally are partitioned via two-stage clustering, first by ClusterONE,then by MCL clustering.

Usage

```
predPPI_MACP(
  data,
  refcpx,
  tpath = tempdir(),
  data_processing = TRUE,
  data_imputing = TRUE,
  scaling = TRUE,
  keepMT = FALSE,
  pcc = TRUE,
  PCCN = TRUE,
  pcc_p = TRUE,
  spearman = TRUE,
  kendall = TRUE,
  bicor = TRUE,
  weighted_rank = TRUE,
  cosine = TRUE,
  jaccard = TRUE,
  dice = TRUE,
  apex = TRUE,
  minfo = TRUE,
  bayesian = TRUE,
  wcc = TRUE,
  euclidean = TRUE,
  manhattan = TRUE,
  canberra = TRUE,
  avg.distance = TRUE,
  rept = 10,
  corr_removal = FALSE,
  corr_cutoff = 0.5,
  classifier = c("glm", "svmRadial", "ranger"),
  verboseIter = TRUE,
```

```
  cv_fold = 5,
  plots = FALSE,
  subcellular_mtPPI = FALSE,
  organism = "mouse",
  csize = 3,
  d = 0.3,
  p = 2,
  max_overlap = 0.8,
  inflation = 9
)
```

Arguments

data	A data matrix with rows including proteins and fractions along the columns. see exampleData.
refcpx	A list of known reference complexes. see getCPX.
tpath	A character string indicating the path to the project directory. If the directory is missing, it will be stored in the Temp directory.
data_processing	If TRUE, removes proteins for which peptide only detected in one fraction (i.e., "one-hit-wonders") across the co-elution table, common contaminants (e.g., keratins) only for mouse and human organisms and frequent flyers. Defaults to TRUE. See data_filtering.
data_imputing	if TRUE, imputes missing values in protein elution profile matrix via average of adjacent rows. This function is not applicable for missing values present in the first or last column. Defaults to TRUE. See impute_MissingData.
scaling	If TRUE, performs column and row-wise normalization. Defaults to TRUE. See scaling.
keepMT	if TRUE, removes all the non-mitochondrial proteins by mapping the co-eluted proteins from chromatography fractions to MitoCarta database. Note that this function is only applicable to mouse or human organisms.Defaults to FALSE. See keepMT.
pcc	If TRUE, computes pairwise protein profile similarity using Pearson correlation metric. Defaults to TRUE. See calculate_PPIscore.
PCCN	If TRUE, computes pairwise protein profile similarity using Pearson correlation plus noise. Defaults to TRUE. See calculate_PPIscore.
pcc_p	If TRUE, computes P-value of the Pearson correlation. Defaults to TRUE. See calculate_PPIscore.
spearman	if TRUE, computes pairwise protein profile similarity using spearman correlation. Defaults to TRUE. See calculate_PPIscore.
kendall	if TRUE, computes pairwise protein profile similarity using kendall correlation. Defaults to TRUE. See calculate_PPIscore.
bicor	if TRUE, computes pairwise protein profile similarity using biweight midcorrealtion (bicor) correlation. Defaults to TRUE. See calculate_PPIscore.

`weighted_rank` if TRUE, computes pairwise protein profile similarity using weighted rank measure. Defaults to TRUE. See `calculate_PPIscore`.

`cosine` If TRUE, computes pairwise protein profile similarity using cosine metric. Defaults to TRUE. See `calculate_PPIscore`.

`jaccard` If TRUE, computes pairwise protein profile similarity using jaccard metric. Defaults to TRUE. See `calculate_PPIscore`.

`dice` if TRUE, computes pairwise protein profile similarity using dice measure. Defaults to TRUE. See `calculate_PPIscore`.

`apex` If TRUE, computes pairwise protein profile similarity using apex. Defaults to TRUE. See `calculate_PPIscore`.

`minfo` If TRUE, computes pairwise protein profile similarity using mutual information. Defaults to TRUE. See `calculate_PPIscore`.

`bayesian` If TRUE, computes pairwise protein profile similarity using Bayes correlation based on zero-count distribution. Defaults to TRUE. See `calculate_PPIscore`.

`wcc` If TRUE, computes pairwise protein profile similarity using weighted cross correlation. Defaults to TRUE. See `calculate_PPIscore`.

`euclidean` if TRUE, computes pairwise protein profile similarity using euclidean measure. Defaults to TRUE. See `calculate_PPIscore`.

`manhattan` if TRUE, computes pairwise protein profile similarity using manhattan measure. Defaults to TRUE. See `calculate_PPIscore`.

`canberra` if TRUE, computes pairwise protein profile similarity using canberra measure. Defaults to TRUE. See `calculate_PPIscore`.

`avg.distance` if TRUE, computes pairwise protein profile similarity using avg.distance measure. Defaults to TRUE. See `calculate_PPIscore`.

`rept` Poisson iterations, defaults to 10. Defaults to TRUE. See `calculate_PPIscore`.

`corr_removal` If TRUE, removes protein pairs with correlation scores < the user defined threshold; defaults to FALSE. See `calculate_PPIscore`.

`corr_cutoff` user defined threshold for correlation similarity scores. Defaults to 0.5.See `calculate_PPIscore`.

`classifier` The type of classifier to use for ensemble or individual model. See `caret` for the available classifiers. Defaults to c("glm", "svmRadial", "ranger"). See `ensemble_model`.

`verboseIter` Logical value, indicating whether to check the status of training process;defaults to FALSE. See `ensemble_model`.

`cv_fold` Number of partitions for cross-validation; defaults to 5. See `ensemble_model`.

`plots` Logical value, indicating whether to plot the performance of the learning algorithm using k-fold cross-validation; defaults to FALSE. These plots are :

- pr_plot - Precision-recall PLOT
- roc_plot - ROC plot
- point_plot - Point plot showing accuracy, F1-score , positive predictive value (PPV), sensitivity (SE) and MCC.

See `ensemble_model`.

`subcellular_mtPPI`
: if TRUE, removes PPIs occurring between outer mt membrane (OMM) and matrix, between intermembrane space (IMS) and matrix, as well as between any subcellular mt compartment (except OMM) and cytosolic proteins as they deemed to be erroneous. Defaults to FALSE. See `subcellular.mtPPI`.

`organism`
: Organism under study (i.e., mouse or human). Defaults to mouse. See `subcellular.mtPPI`.

`csize`
: An integer, the minimum size of the predicted complexes. Defaults to 2. See `get_clusters`.

`d`
: A number, density of predicted complexes. Defaults to 0.3. See `get_clusters`.

`p`
: An integer, penalty value for the inclusion of each node. Defaults to 2. See `get_clusters`.

`max_overlap`
: A number, specifies the maximum allowed overlap between two clusters. Defaults to 0.8. See `get_clusters`.

`inflation`
: MCL inflation parameter. Defaults to 9.

Details

predPPI_MACP

Value

Return following data sets in the current directory including:

- unfilteredPPIs - Unfiltered interactions
- filteredPPI - High-confidence interactions defined by ROC threshold.
- High_confidence interactions_with_mt_sublocalization - if subcellular.mtPPI is TRUE, it return high-confidene PPIs with mt sublocalization status.
- predicted_cpx_clusterONE - Putative complexes generated by clusterONE.
- predicted_cpx_clusterONE_MCL - Putative complexes generated by clusterONE and MCL.
- Best_roc_curve_cutoff - Best cutoff generated from ROC curve.

`refcpx` *CORUM reference complexes*

Description

A list containing CORUM reference complexes for mouse organism.

Usage

```
data(refcpx)
```

Source

https://mips.helmholtz-muenchen.de/corum/

scaling *Column and Row-wise Normalization*

Description

This function performs column and row-wise normalization.

Usage

```
scaling(data)
```

Arguments

data	A data matrix with rows including proteins and fractions along the columns.

Details

scaling

Value

Scaled data matrix.

Author(s)

Matineh Rahmatbakhsh, <matinerb.94@gmail.com>

Examples

```
# Load the co-elution data
data("exampleData")
# Normalize the data
datOut <- scaling(exampleData)
```

subcellular.mtPPI *Keep Mitochondrial (mt) Proteins*

Description

This function removes PPIs occurring between outer mt membrane (OMM) and matrix, between intermembrane space (IMS) and matrix, as well as between any subcellular mt compartment (except OMM) and cytosolic proteins as they deemed to be erroneous

Usage

```
subcellular.mtPPI(ppi, organism = "mouse")
```

Arguments

ppi	Interactions data containing id1-id2-weight triplets.
organism	Organism under study (i.e., mouse or human). Defaults to mouse.

Details

subcellular.mtPPI

Value

Filtered PPI netwrok.

Author(s)

Matineh Rahmatbakhsh, <matinerb.94@gmail.com>

Examples

```
ppi <-
read.table(system.file("extdata/ppi_input_ClusterONE.txt",
package="MACP"),
quote="\"", comment.char="")
filtered_mtEdges <- subcellular.mtPPI(ppi)
```

Index