Package 'HPiP'

July 12, 2023

Type Package

Title Host-Pathogen Interaction Prediction

Version 1.7.0

Description HPiP (Host-Pathogen Interaction Prediction) uses an ensemble learning algorithm for prediction of host-pathogen protein-protein interactions (HP-PPIs) using structural and physicochemical descriptors computed from amino acid-composition of host and pathogen proteins.The proposed package can effectively address data shortages and data unavailability for HP-PPI network reconstructions. Moreover, establishing computational frameworks in that regard will reveal mechanistic insights into infectious diseases and suggest potential HP-PPI targets, thus narrowing down the range of possible candidates for subsequent wet-lab experimental validations.

Depends R (>= 4.1)

Imports dplyr (>= 1.0.6), httr (>= 1.4.2), readr, tidyr, tibble, utils, stringr, magrittr, caret, corrplot, ggplot2, pROC, PRROC, igraph, graphics, stats, purrr, grDevices, protr, MCL

License MIT + file LICENSE

Encoding UTF-8

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

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

VignetteBuilder knitr

Suggests rmarkdown, colorspace, e1071, kernlab, ranger, SummarizedExperiment, Biostrings, randomForest, gprofiler2, gridExtra, ggthemes, BiocStyle, BiocGenerics, RUnit, tools, knitr

biocViews Proteomics, SystemsBiology, NetworkInference, StructuralPrediction, GenePrediction, Network

RoxygenNote 7.2.0

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

git_branch devel

git_last_commit 9fb6dd0

git_last_commit_date 2023-04-25

Date/Publication 2023-07-12

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

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

R topics documented:

`calculateAAC` *Calculate Amino Acid Composition (AAC) Descriptor*

Description

This function calculates Amino Acid Composition (AAC) descriptor for the data input.

Usage

```
calculateAAC(x)
```

Arguments

`x`	A data.frame containing gene/protein names and their fasta sequences.

Details

calculateAAC

Value

A length 20 named vector for the data input.

Author(s)

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

References

Dey, L., Chakraborty, S., and Mukhopadhyay, A. (2020). Machine learning techniques for sequence-based prediction of viral–host interactions between SARS-CoV-2 and human proteins. *Biomed. J.* 43, 438–450.

See Also

See `calculateDC` and `calculateTC` for Dipeptide Composition and Tripeptide Composition descriptors.

Examples

```
data(UP000464024_df)
x_df <- calculateAAC(UP000464024_df)
head(x_df, n = 2L)
```

`calculateAutocor` *Calculate Autocorrelation Descriptors*

Description

This function calculates autocorrelation descriptors:

- `moran` - moran autocorrelation, (Dim: `length(target.props) * nlag`).
- `geary` - geary autocorrelation, (Dim: `length(target.props) * nlag`).
- `moreaubroto` - moreau-broto autocorrelation, (Dim: `length(target.props) * nlag`).

Usage

```
calculateAutocor(
  x,
  target.props = c("CIDH920105", "BHAR880101", "CHAM820101", "CHAM820102",
    "CHOC760101", "BIGC670101", "CHAM810101", "DAYM780201"),
  nlag = 30L,
  type = c("moran", "geary", "moreaubroto")
)
```

Arguments

`x`	A data.frame containing gene/protein names and their fasta sequences.
`target.props`	A character vector, specifying the accession number of the target properties. 8 properties are used by default, as listed below: **AccNo. CIDH920105** Normalized average hydrophobicity scales (Cid et al., 1992) **AccNo. BHAR880101** Average flexibility indices (Bhaskaran-Ponnuswamy, 1988) **AccNo. CHAM820101** Polarizability parameter (Charton-Charton, 1982) **AccNo. CHAM820102** Free energy of solution in water, kcal/mole (Charton-Charton, 1982) **AccNo. CHOC760101** Residue accessible surface area in tripeptide (Chothia, 1976) **AccNo. BIGC670101** Residue volume (Bigelow, 1967) **AccNo. CHAM810101** Steric parameter (Charton, 1981) **AccNo. DAYM780201** Relative mutability (Dayhoff et al., 1978b)
`nlag`	Maximum value of the lag parameter. Default is `30`.
`type`	The autocorrelation type: `moran`, `geary`, or `moreaubroto`.

Details

calculateAutocor

Value

A length nlag named vector for data input.

Author(s)

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

References

AAindex: Amino acid index database. http://www.genome.ad.jp/dbget/aaindex.html

Feng, Z.P. and Zhang, C.T. (2000) Prediction of membrane protein types based on the hydrophobic index of amino acids. *Journal of Protein Chemistry*, 19, 269-275.

Horne, D.S. (1988) Prediction of protein helix content from an autocorrelation analysis of sequence hydrophobicities. *Biopolymers*, 27, 451-477.

Sokal, R.R. and Thomson, B.A. (2006) Population structure inferred by local spatial autocorrelation: an Usage from an Amerindian tribal population. *American Journal of Physical Anthropology*, 129, 121-131.

Examples

```
data(UP000464024_df)
x_df <- calculateAutocor(UP000464024_df,type = 'moran')
head(x_df, n = 2L)
```

---

calculateBE *Tranform a Seqeuence into Binary Encoding (BE)*

Description

This function transform each residue in a peptide into 20 coding values.

Usage

```
calculateBE(x)
```

Arguments

x	A data.frame containing gene/protein names and their fasta sequences.

Details

calculateBE

Value

A length 400 named vector for the data input.

Author(s)

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

References

Al-Barakati, H. J., Saigo, H., and Newman, R. H. (2019). RF-GlutarySite: a random forest based predictor for glutarylation sites. *Mol. Omi.* 15, 189–204.

`calculateCTDC` *Calculate CTD Descriptors - Composition (C)*

Description

This function calculates Composition (C) descriptor for data input.

Usage

```
calculateCTDC(x)
```

Arguments

x	A data.frame containing gene/protein names and their fasta sequences.

Details

calculateCTDC

Value

A length 21 named vector for the data input.

Author(s)

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

References

Dubchak, I., Muchnik, I., Holbrook, S. R., and Kim, S.-H. (1995). Prediction of protein folding class using global description of amino acid sequence.*Proc. Natl. Acad. Sci.* 92, 8700–8704.

See Also

See `calculateCTDT` and `calculateCTDD` for Transition and Distribution descriptors.

Examples

```
data(UP000464024_df)
x_df <- calculateCTDC(UP000464024_df)
head(x_df, n = 2L)
```

`calculateCTDD` *Calculate CTD Descriptors - Distribution (D)*

Description

This function calculates Distribution (D) descriptor for data input.

Usage

```
calculateCTDD(x)
```

Arguments

x	A data.frame containing gene/protein names and their fasta sequences.

Details

calculateCTDD

Value

A length 105 named vector for the data input.

Author(s)

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

References

Dubchak, I., Muchnik, I., Holbrook, S. R., and Kim, S.-H. (1995). Prediction of protein folding class using global description of amino acid sequence.*Proc. Natl. Acad. Sci.* 92, 8700–8704.

See Also

See `calculateCTDC` and `calculateCTDT` for Composition and Transition descriptors.

Examples

```
data(UP000464024_df)
x_df <- calculateCTDD(UP000464024_df)
head(x_df, n = 1L)
```

`calculateCTDT` *Calculate CTD Descriptors - Transition (T)*

Description

This function calculates Transition (T) descriptor for data input.

Usage

```
calculateCTDT(x)
```

Arguments

x	A data.frame containing gene/protein names and their fasta sequences.

Details

calculateCTDT

Value

A length 21 named vector for the data input.

Author(s)

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

References

Dubchak, I., Muchnik, I., Holbrook, S. R., and Kim, S.-H. (1995). Prediction of protein folding class using global description of amino acid sequence. *Proc. Natl. Acad. Sci.* 92, 8700–8704.

See Also

See `calculateCTDC` and `calculateCTDD` for Composition and Distribution descriptors.

Examples

```
data(UP000464024_df)
x_df <- calculateCTDT(UP000464024_df)
head(x_df, n = 2L)
```

`calculateCTriad` *Calculate Conjoint Triad Descriptor*

Description

This function calculates Conjoint Triad descriptor for data input.

Usage

```
calculateCTriad(x)
```

Arguments

`x`	A data.frame containing gene/protein names and their fasta sequences.

Details

calculateCTriad

Value

A length 343 named vector for the data input.

Author(s)

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

References

Shen, J., Zhang, J., Luo, X., Zhu, W., Yu, K., Chen, K., et al. (2007). Predicting protein–protein interactions based only on sequences information. *Proc. Natl. Acad. Sci.* 104, 4337–4341.

Examples

```
data(UP000464024_df)
x_df <- calculateCTriad(UP000464024_df)
head(x_df, n = 2L)
```

`calculateDC` *Calculate Dipeptide Composition (DC) Descriptor*

Description

This function calculates Dipeptide Composition (DC) descriptor for data input.

Usage

```
calculateDC(x)
```

Arguments

`x`	A data.frame containing gene/protein names and their fasta sequences.

Details

calculateDC

Value

A length 400 named vector for the data input.

Author(s)

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

References

Bhasin, M., and Raghava, G. P. S. (2004). Classification of nuclear receptors based on amino acid composition and dipeptide composition. *J. Biol. Chem.* 279, 23262–23266.

See Also

See `calculateAAC` and `calculateTC` for Amino Acid Composition and Tripeptide Composition descriptors.

Examples

```
data(UP000464024_df)
x_df <- calculateDC(UP000464024_df)
head(x_df, n = 2L)
```

`calculateF` *Calculate F1 or F2 Descriptors*

Description

This function calculates F1 or F2 descriptors:

- `F1` - sum of squared length of Single Amino Acid Repeats (SARs) in the entire protein sequence.
- `F2` - maximum of the sum of Single Amino Acid Repeats (SARs) in a window of 6 residues.

Usage

```
calculateF(x, type = c("F1", "F2"))
```

Arguments

`x`	A data.frame containing gene/protein names and their fasta sequences.
`type`	The descriptor type: `F1` or `F2`.

Details

calculateF

Value

A length 20 named vector for the data input.

Author(s)

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

References

Alguwaizani, S., Park, B., Zhou, X., Huang, D.-S., and Han, K. (2018). Predicting interactions between virus and host proteins using repeat patterns and composition of amino acids. *J. Healthc. Eng.* 2018.

Examples

```
data(UP000464024_df)
x_df <- calculateF(UP000464024_df, type = "F1")
head(x_df, n = 2L)
```

`calculateKSAAP` *Calculate k-spaced Amino Acid Pairs (KSAAP) Descriptor*

Description

This function calculates k-spaced Amino Acid Pairs (KSAAP) Descriptor for data input. This function is adapted from the `CkSAApair` function in the ftrCOOL package.

Usage

```
calculateKSAAP(x, spc = 3)
```

Arguments

`x`	A data.frame containing gene/protein names and their fasta sequences.
`spc`	A number of spaces separating two adjacent residues by a distance of spc, which can be any number up to two less than the length of the peptide; default to 3.

Details

calculateKSAAP

Value

A length 400 named vector for the data input.

Author(s)

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

References

Kao, H.-J., Nguyen, V.-N., Huang, K.-Y., Chang, W.-C., and Lee, T.-Y. (2020).SuccSite: incorporating amino acid composition and informative k-spaced amino acid pairs to identify protein succinylation sites. *Genomics. Proteomics Bioinformatics* 18, 208–219.

Examples

```
data(UP000464024_df)
x_df <- calculateKSAAP(UP000464024_df)
head(x_df, n = 2L)
```

`calculateQD_Sm` *Calculate Quadruples Composition (QC) Descriptor from Biochemical Similarity Classes*

Description

This function calculates Quadruples Composition (QC) descriptor from biochemical similarity classes.

Usage

```
calculateQD_Sm(x)
```

Arguments

x	A data.frame containing gene/protein names and their fasta sequences.

Details

calculateQD_Sm

Value

A length 1296 named vector for the data input.

Author(s)

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

References

Ahmed, I., Witbooi, P., and Christoffels, A. (2018). Prediction of human-Bacillus anthracis protein–protein interactions using multi-layer neural network.*Bioinformatics* 34, 4159–4164.

Examples

```
data(UP000464024_df)
x_df <- calculateQD_Sm(UP000464024_df)
head(x_df, n = 2L)
```

`calculateTC` *Calculate Tripeptide Composition (TC) Descriptor*

Description

This function calculates Tripeptide Composition (TC) descriptor for data input.

Usage

```
calculateTC(x)
```

Arguments

`x`	A data.frame containing gene/protein names and their fasta sequences.

Details

calculateTC

Value

A length 8,000 named vector for the data input.

Author(s)

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

References

Liao, B., Jiang, J.-B., Zeng, Q.-G., and Zhu, W. (2011). Predicting apoptosis protein subcellular location with PseAAC by incorporating tripeptide composition. *Protein Pept. Lett.* 18, 1086–1092

See Also

See `calculateAAC`,`calculateDC` and `calculateTC_Sm` for Amino Acid Composition, Dipeptide Composition and Tripeptide Composition (TC) Descriptor from Biochemical Similarity Classes.

Examples

```
data(UP000464024_df)
x_df <- calculateTC(UP000464024_df)
head(x_df, n = 2L)
```

`calculateTC_Sm` *Calculate Tripeptide Composition (TC) Descriptor from Biochemical Similarity Classes*

Description

This function calculates Tripeptide Composition (TC) descriptor from biochemical similarity classes.

Usage

```
calculateTC_Sm(x)
```

Arguments

`x`	A data.frame containing gene/protein names and their fasta sequences.

Details

calculateTC_Sm

Value

A length 216 named vector for the data input.

Author(s)

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

References

Ahmed, I., Witbooi, P., and Christoffels, A. (2018). Prediction of human-Bacillus anthracis protein–protein interactions using multi-layer neural network.*Bioinformatics* 34, 4159–4164.

Cui, G., Fang, C., and Han, K. (2012). Prediction of protein-protein interactions between viruses and human by an SVM model. *BMC bioinformatics*, 1–10.

See Also

See `calculateTC` for Tripeptide Composition descriptor.

Examples

```
data(UP000464024_df)
x_df <- calculateTC_Sm(UP000464024_df)
head(x_df, n = 2L)
```

corr_plot *Plot Correlation Matrix between Input Features*

Description

A graphical display of a correlation matrix.

Usage

```
corr_plot(cormat, method = "number", cex = 0.9)
```

Arguments

cormat	A correlation matrix.
method	The visualization method of correlation matrix; defaults to number. See corrplot for more details.
cex	The size of x/y axis label.

Details

corr_plot

Value

A correlation plot.

Author(s)

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

Examples

```
data('example_data')
x <- na.omit(example_data)
#perform feature selection
s <- FSmethod(x, type = 'both',
cor.cutoff = 0.7, resampling.method = "repeatedcv",
iter = 5, repeats = 3, metric = "ROC", verbose = TRUE)
corr_plot(s$cor.result$corProfile, method = 'square' , cex = 0.5)
```

enrich.df *Enrichment Result*

Description

Input data for enrichplot

Usage

```
data(enrich.df)
```

Details

To construct this dataset, predicted interactions, generated from pred_ensembel was used as data input for enrichplot.

enrichfindP *Functional Enrichment Analysis for Pathogen Interactors in the High-Confidence Network.*

Description

This function uses gost function in gprofiler2 package to perfrom functional enrichment analysis for pathogen interactors in the high-confidence network.

Usage

```
enrichfindP(
  ppi,
  threshold = 0.05,
  sources = c("GO", "KEGG"),
  p.corrction.method = "bonferroni",
  org = "hsapiens"
)
```

Arguments

ppi	A data.frame containing pathogen proteins in the first column and host proteins in the second column.
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.
org	An organism name;defaults to 'hsapiens'. See gost for more details.

Details

enrichfindP

Value

A data.frame with the enrichment analysis results.

Author(s)

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

See Also

See enrichplot for plotting enrichment analysis.

Examples

```
data('predicted_PPIs')
#perform enrichment
enrich.df <- enrichfindP(predicted_PPIs,
threshold = 0.05,
sources = c("GO", "KEGG"),
p.corrction.method = "bonferroni",
org = "hsapiens")
```

---

enrichfind_cpx *Functional Enrichment Analysis for Predicted Modules*

Description

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

Usage

```
enrichfind_cpx(
  predcpx,
  threshold = 0.05,
  sources = c("GO", "KEGG"),
  p.corrction.method = "bonferroni",
  org = "hsapiens"
)
```

Arguments

predcpx	Predicted modules resutled from run_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.
org	An organism name;defaults to 'hsapiens'. See gost for more details.

Details

enrichfind_cpx

Value

A data.frame with the enrichment analysis results.

Author(s)

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

---

enrichfind_hp *Functional Enrichment Analysis of all Host Proteins*

Description

This function uses gost function in gprofiler2 package to perfrom functional enrichment analysis for all predicted host proteins in the high-confidence network.

Usage

```
enrichfind_hp(
  ppi,
  threshold = 0.05,
  sources = c("GO", "KEGG"),
  p.corrction.method = "bonferroni",
  org = "hsapiens"
)
```

Arguments

ppi	A data.frame containing pathogen proteins in the first column and host proteins in the second column.
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.

org
: An organism name;defaults to 'hsapiens'. See gost for more details.

Details

enrichfind_hp

Value

A data.frame with the enrichment analysis results.

Author(s)

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

See Also

See enrichplot for plotting enrichment analysis.

Examples

```
data('predicted_PPIs')
#perform enrichment
enrich.df <- enrichfind_hp(predicted_PPIs,
threshold = 0.05,
sources = c("GO", "KEGG"),
p.corrction.method = "bonferroni",
org = "hsapiens")
```

---

enrichplot *Plot the Enrichment Reuslt*

Description

This function plots the enrichment result.

Usage

```
enrichplot(x, low = "blue", high = "red", cex.size = 15)
```

Arguments

x
: A data.frame with the enrichment analysis results.

low
: Colours for low.

high
: Colours for high.

cex.size
: Text size.

Details

enrichplot

Value

An enrichment plot.

Author(s)

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

See Also

See enrichfindP for functional enrichment analysis.

Examples

```
data('enrich.df')
#select enrichment for one of the example (e.g., E protein)
enrich.df <-
enrich.df[enrich.df$id == "E:P0DTC4", ]
enrichplot(enrich.df, low = "blue", high = "red", cex.size = 10)
```

---

example_data *Input Data for Prediction Algorithm*

Description

Input data for pred_ensembel

Usage

```
data(example_data)
```

Format

a data.frame containing unlabeled or labeled HP-PPIs and pre-computed numerical features.

filter_missing_values *Drop the Missing Values Above a Certain Threshold*

Description

Given an input matrix, compute the missingness rate for each features and keep only features with missing rate more than user-defined percentage.

Usage

```
filter_missing_values(x, max_miss_rate = 20)
```

Arguments

x	A numeric matrix as input.
max_miss_rate	Maximal missing rate allowed for a feature;default is 20.

Details

filter_missing_values

Value

A dataframe with features with missingness rate of more than user-defined threshold.

Author(s)

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

Examples

```
x <- matrix(1:10, ncol = 2)
x[, 2] <- NA
filter_missing_values(x, 30)
```

FreqInteractors *Plot the Pathogen Proteins' frequency of Interactions with Host Proteins*

Description

This function plots the pathogen proteins' Frequency of interactions with host proteins

Usage

```
FreqInteractors(ppi, cex.size = 12)
```

Arguments

ppi	A data.frame containing pathogen proteins in the first column and host proteins in the second column.
cex.size	Text size.

Details

FreqInteractors

Value

A frequency plot.

Author(s)

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

Examples

```
ppi <- data.frame(
    node1 = c("A", "A", "A", "B", "B", "B", "B"),
    node2 = c("C", "E", "D", "F", "G", "H", "I")
)
FreqInteractors(ppi)
```

FSmethod *Feature Selection via Matrix Correlation and Recursive Feature Elimination (RFE)*

Description

This function performs feature selections via two approaches

- `filter.corr` - compute matrix correlation between features and filter using a threshold.
- `rfeFS` - perform recursive feature elimination (RFE) method wrapped with a Random Forest (RF) algorithm for feature importance evaluation.

Usage

```
FSmethod(
  x,
  type = c("cor", "rfe", "both"),
  cor.cutoff = 0.7,
  resampling.method = "cv",
  iter = 2,
  repeats = 3,
  metric = "Accuracy",
  verbose = TRUE
)
```

Arguments

x	A data.frame containing protein-protein interactions, class labels and features.
type	The feature selection type, one or two of filter.corr and rfeFS.
cor.cutoff	Correlation coefficient cutoff used for filtering. See filter.corr for more details.
resampling.method	The resampling method for RFE :'boot', 'boot632', optimism_boot',boot_all', 'cv', 'repeatedcv', 'LOOCV', 'LGOCV';defaults to cv. See rfeFS and rfeControl for more details.
iter	Number of partitions for cross-validation; defaults to 2. See rfeFS and rfeControl for more details.
repeats	For repeated k-fold cross validation only; defaults to 3.See rfeFS and rfeControl for more details.
metric	A string that specifies what summary metric will be used to select the optimal feature ; default to ROC.See rfeFS and rfe for more details.
verbose	Make the output verbose.See rfeFS and rfeControl for more details.

Details

FSmethod

Value

If the type set to filter.corr , the output includes the following elements:

- corProfile - A correlation matrix.
- corSelectedFeatures - Name of features that retained after the correlation analysis.
- cordf - A data.frame filtered.

If the type set to rfeFS , the output includes the following elements:

- rfProfile - A list of elements. See rfe for more details.
- rfSelectedFeatures - Name of features that retained in the feature selection process.
- rfdf - A data.frame filtered.

If type set to both the output includes the following elements:

- rfdf - The final data.frame that includes the selected features retained after both filter.corr and rfeFS analysis.

Author(s)

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

Examples

```
data('example_data')
x <- na.omit(example_data)
s <- FSmethod(x, type = 'both',
cor.cutoff = 0.7, resampling.method = "repeatedcv",
iter = 5, repeats = 3, metric = "ROC", verbose = TRUE)
```

---

getFASTA *Fetch FASTA Sequence from the UniProt Database*

Description

This function retrieves protein sequences in FASTA format directly from the UniProt database via UniProt protein IDs. This function also checks if the amino-acid composition of protein sequences is in the 20 default types.

Usage

```
getFASTA(uniprot.id, filename = "FASTA.RData", path = "FASTASeq")
```

Arguments

uniprot.id	A character vector of UniProt identifiers.
filename	A character string, indicating the output filename as an RData object to store the retrieved sequences.
path	A character string indicating the path to the project directory that contains the interaction data. If the directory is missing, it will be stored in the current directory. Default is FASTASeq.

Details

getFASTA

Value

A list containing protein FASTA sequences.

Author(s)

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

Examples

```
# get fasta sequences for three proteins of SARS-Cov-2
local = tempdir()
uniprot.id <- c("P0DTC4", "P0DTC5", "P0DTC9")
fasta_df <- getFASTA(uniprot.id, filename = 'FASTA.RData', path = local)
head(fasta_df)
```

getHPI *Generating Host-Pathogen Protein-Protein Interaction (HP-PPI) Descriptors*

Description

This function calculates Host-Pathogen Protein-Protein Interaction (HP-PPI) descriptors via two approaches

- `combine` - combine the two descriptor matrix, result has `(p1 + p2)` columns
- `kron.prod` - if A has m x n matrix and B is q x p matrix, then the Kronecker product is the code(pm × qn) block matrix

Usage

```
getHPI(pathogenData, hostData, type = c("combine", "kron.prod"))
```

Arguments

`pathogenData`	The pathogen descriptor matrix.
`hostData`	The host descriptor matrix.
`type`	The interaction type, one or two of `"combine"` and `"kron.prod"`.

Details

getHPI

Value

A matrix containing the Host-Pathogen Protein-Protein Interaction (HP-PPI) descriptors.

Author(s)

Matineh Rahmatbakhsh <matinerb.94@gmail.com>

Examples

```
x <- matrix(c(1, 2, 3, 1), nrow = 2, ncol = 2, byrow = TRUE)
y <- matrix(c(0, 3, 2, 1), nrow = 2, ncol = 2, byrow = TRUE)
getHPI(x, y, "combine")
getHPI(x, y, "kron.prod")
```

get_negativePPI *Construct Negative Reference Host-Pathogen Protein-Protein Interactions (HP-PPIs)*

Description

Construct true negative protein-protein interactions from the positive interactions. In the context of PPI prediction, a negative interaction is a pair of proteins that unlikely to interact. Since there is no experimentally verified non-interacting pair, the negative sampling can be used to construct the negative reference set. The negative sampling can be constructed from a set of host proteins, a set of pathogen proteins, and a list of positive reference interactions between members of host and pathogen proteins (Eid et al., 2016).

Usage

```
get_negativePPI(prot1, prot2, TPset)
```

Arguments

prot1	A character vector containing pathogen proteins.
prot2	A character vector containing host proteins.
TPset	A character vector containing positive reference interactions.

Details

get_negativePPI

Value

A Data.frame containing true negative interactions.

Author(s)

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

References

Eid, F.-E., ElHefnawi, M., and Heath, L. S. (2016). DeNovo: virus-host sequence-based protein–protein interaction prediction. Bioinformatics 32, 1144–1150.

See Also

See get_positivePPI for generating positive protein-protein interaction.

Examples

```
prot1 <- c("P0DTC4", "P0DTC5", "P0DTC9")
prot2 <- c("Q9Y679", "Q9NW15", "Q9NXF8")
TPset <- c("P0DTC4~P31948", "P0DTC8~Q13438")
TN_PPI <- get_negativePPI(prot1, prot2, TPset)
head(TN_PPI)
```

`get_positivePPI` *Fetch Positive Reference Host-Pathogen Protein-Protein Interactions (HP-PPIs) from the BioGRID Database*

Description

This function retrieves positive reference host-pathogen protein-protein interactions directly from BioGRID database.

Usage

```
get_positivePPI(
  organism.taxID,
  access.key,
  filename = "PositiveInt.RData",
  path = "PositiveInt"
)
```

Arguments

`organism.taxID`	Taxonomy identifier for the pathogen.
`access.key`	Access key for using BioGRID webpage. To retrieve interactions from the BioGRID database, the users are first required to register for access key at https://webservice.thebiogrid.org/.
`filename`	A character string, indicating the output filename as an RData object to store the retrieved interactions.
`path`	A character string indicating the path to the project directory that contains the interaction data. If the directory is missing, it will be stored in the current directory. Default is PositiveInt.

Details

get_positivePPI

Value

A Data.frame containing true positive protein-protein interactions for the selected pathogen.

Author(s)

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

See Also

See get_negativePPI for generating negative protein-protein interaction.

Examples

```
local = tempdir()
try(get_positivePPI(organism.taxID = 2697049,
access.key = 'XXXX',
filename = "PositiveInt.RData",
path = local))
```

Gold_ReferenceSet *Gold-standard Reference Set of Inter-Species PPIs*

Description

This dataset consists of experimentally validated human-SARS-CoV-2 interactions (positive set) and non-interacting pairs (negative set). The following data consists of:

- PPI: SARS-CoV-2-human protein-protein interactions (PPIs)
- Official Symbol Interactor A: SARS-CoV-2 gene names
- official Symbol Interactor B: human host gene names
- Pathogen_Protein: UniProt identifiers for SARS-CoV-2 virus
- Host_Protein: UniProt identifiers for human proteins
- class: labeled examples (both positive and negative)

Usage

```
data(Gold_ReferenceSet)
```

Format

a data.frame containing 500 validated pairs (i.e., positive set) and 500 non-interacting pairs (i.e., negative set).

Details

To construct this dataset, validated interactions (positive set) were retrieved from BioGrid database and were further filtered to only include those interactions provided by (Samavarchi-Tehrani et al., 2020). In this study, the authors mapped interaction between 27 SARS-CoV-2 and human proteins via the proximity-dependent biotinylation (BioID) approach. 500 SARS-CoV-2-host interaction pairs then randomly selected from all pairs to serve as positive examples. To construct negative examples,negative sampling were used using get_negativePPI.

Source

https://www.biorxiv.org/content/10.1101/2020.09.03.282103v1

References

Samavarchi-Tehrani,P. et al. (2020) A SARS-CoV-2-host proximity interactome. BioRxiv.

---

host_se *Host SummarizedExperiment object*

Description

SummarizedExperiment object of numerical features for host proteins.

Usage

```
data(host_se)
```

Details

To construct this object, first protein sequences were converted to numerical features using (CTD) descriptors provided in the HPiP package, followed by converting each numerical features matrix to SummarizedExperiment object.Each object is then merged into one object using 'cbind()'.

---

impute_missing_data *Impute missing Values per Features (i.e., Columns)*

Description

Given an input matrix, impute the missing values via three approaches including mean, median or zero.

Usage

```
impute_missing_data(x, method = c("mean", "median", "zero"))
```

Arguments

x	A numeric matrix as input.
method	Imputation method for missing values (mean, median or zero).

Details

impute_missing_data

Value

Imputed matrix.

Author(s)

Matineh Rahmatbakhsh <matinerb.94@gmail.com>

Examples

```
x <- matrix(1:10, ncol = 2)
x[1:3, 2] <- NA
row.names(x) <- c("A", "B", "C", "D", "E")
colnames(x) <- c("col1", "col2")
impute_missing_data(x, method = "mean")
impute_missing_data(x, method = "median")
impute_missing_data(x, method = "zero")
```

---

plotPPI *Plot the Predicted PPI*

Description

Plot the predicted PPIs. This function uses the plot function of the igraph.

Usage

```
plotPPI(
  ppi,
  edge.name = "ensemble_score",
  node.color = "grey",
  edge.color = "orange",
  cex.node = 4,
  node.label.dist = 1.5
)
```

Arguments

ppi	A data.frame containing protein-protein interactions with edge score.
edge.name	A character string giving an edge attribute name.
node.color	The fill color of the node.
edge.color	The color of the edge.
cex.node	The size of the node.
node.label.dist	The distance of the label from the center of the node.

Details

plotPPI

Value

A PPI plot.

Author(s)

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

Examples

```
df <- data.frame(
    node1 = c("A", "B", "C", "D", "E"),
    node2 = c("C", "E", "E", "E", "A"),
    edge.scores = c(0.5, 0.4, 0.3, 0.2, 0.7)
)
plotPPI(df, edge.name = "edge.scores")
```

---

predicted_PPIs *Predicted HP-PPIs*

Description

Input data for enrichfindP

Usage

```
data(predicted_PPIs)
```

---

pred_ensembel *Predict Interactions via Ensemble Learning Method*

Description

This function uses an ensemble of classifiers to predict interactions from the sequence-based dataset. This ensemble algorithm combines different results generated from individual classifiers within the ensemble via average to enhance prediction.

Usage

```
pred_ensembel(
  features,
  gold_standard,
  classifier = c("avNNet", "svmRadial", "ranger"),
  resampling.method = "cv",
  ncross = 2,
  repeats = 2,
  verboseIter = TRUE,
  plots = TRUE,
  filename = "plots.pdf"
)
```

Arguments

`features`	A data frame with host-pathogen protein-protein interactions (HP-PPIs) in the first column, and features to be passed to the classifier in the remaining columns.
`gold_standard`	A data frame with gold_standard HP-PPIs and class label indicating if such PPIs are positive or negative.
`classifier`	The type of classifier to use. See `caret` for the availbale classifiers.
`resampling.method`	The resampling method:'boot', 'boot632', 'optimism_boot', boot_all', 'cv', 'repeatedcv', 'LOOCV', 'LGOCV'; defaults to cv. See `trainControl` for more details.
`ncross`	Number of partitions for cross-validation; defaults to 5.See `trainControl` for more details.
`repeats`	for repeated k-fold cross validation only; defaults to 3.See `rfeControl` for more details.
`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 TRUE.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`	A character string, indicating the output filename as an pdf object.

Details

pred_ensembel

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>

Examples

```
data('example_data')
features <- example_data[, -2]
gd <- example_data[, c(1,2)]
gd <- na.omit(gd)
ppi <-pred_ensembel(features,gd,
classifier = c("avNNet", "svmRadial", "ranger"),
resampling.method = "cv",ncross = 2,verboseIter = FALSE,plots = FALSE,
filename = "plots.pdf")
#extract predicted interactions
pred_interaction <- ppi[["predicted_interactions"]]
```

run_clustering *Module Detection*

Description

This function contains five module detection algorithms including fast-greedy algorithm (FC), walk-trap algorithm (RW), multi-level community algorithm (ML), label propagation algorithm (clp), and markov clustering (MCL).

Usage

```
run_clustering(
  ppi,
  method = c("FC", "RW", "ML", "clp", "MCL"),
  expan = 2,
  infla = 5,
  iter = 50
)
```

Arguments

ppi	A data.frame containing pathogen proteins in the first column,host proteins in the second column, and edge weight in the third column.
method	Module detection algorithms including: • FC - fast-greedy algorithm. • RW - walktrap algorithm. • ML - multi-level community algorithm. • clp - label propagation algorithm. • MCL - markov clustering.
expan	Numeric value > 1 for the expansion parameter. See mcl for more details.
infla	Numeric value > 0 for the inflation power coefficient. See mcl for more details.
iter	An integer, the maximum number of iterations for the MCL. See mcl for more details.

Details

run_clustering

Value

A data.frame with the enrichment analysis results.

Author(s)

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

unlabel_data *HP-PPIs with Unknown Class Labels*

Description

This dataset consists of interactions between SARS-CoV-2 and human proteins, achieved by AP-MS (affinity purification mass spectrometry).

Usage

```
data(unlabel_data)
```

Format

A data.frame containing 700 SARS-CoV-2-Human protein-protein interactions (PPIs) with pre-computed numerical features using CTD (composition/transition/distribution) descriptors.

Details

To construct this dataset, data (supplementary table 1) containing SARS-CoV-2-human PPIs was retrieved from (Gordon et al., 2020) and 700 pairs were randomly selected from total pairs, followed by converting protein sequences of host or viral proteins to numerical features and finally concatenating the computed features in order to construct host-pathogen PPIs.

Source

https://www.nature.com/articles/s41586-020-2286-9#Sec36

References

Gordon,D.E. et al. (2020) A SARS-CoV-2 protein interaction map reveals targets for drug repurposing. Nature, 583, 459–468.

UP000464024_df *Data.frame Containing SARS-CoV-2 FASTA Sequences*

Description

This data includes one protein sequence per SARS-CoV-2 gene, retrieved directly from UniProt database using getFASTA.

Usage

```
data(UP000464024_df)
```

Format

A data.frame with two columns:(1) UniprotKBID, UniProt identifier.(2) FASTASEQ, sequences per SARS-CoV-2 gene.

Source

https://www.uniprot.org/uniprot/?query=proteome:UP000464024

---

var_imp *Variable Importance Plot*

Description

A graphical display of variable importance of selected features.

Usage

```
var_imp(x, cex.x = 1, cex.y = 2)
```

Arguments

x	A list of elements returned from RFE analysis. See rfe for more details
cex.x	The size of x axis label.
cex.y	The size of y axis label.

Details

var_imp

Value

Variable Importance Plot.

Author(s)

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

Examples

```
data('example_data')
x <- na.omit(example_data)
#perform feature selection
s <- FSmethod(x, type = 'both',
cor.cutoff = 0.7, resampling.method = "repeatedcv",
iter = 5, repeats = 3, metric = "ROC", verbose = TRUE)
var_imp(s$rf.result$rfProfile, cex.x = 10, cex.y = 10)
```

viral_se *Viral SummarizedExperiment object*

Description

SummarizedExperiment object of numerical features for SARS-CoV-2 proteins.

Usage

```
data(viral_se)
```

Details

To construct this object, first protein sequences were converted to numerical features using (CTD) descriptors provided in the HPiP package, followed by converting each numerical features matrix to SummarizedExperiment object. Each object is then merged into one object using ‘cbind()‘.

Index