Package ‘MSiP’

October 12, 2022

Type Package

Title 'MassSpectrometry' Interaction Prediction

Version 1.3.7

Description The 'MSiP' is a computational approach to predict protein-protein interactions from large-scale affinity purification mass 'spectrometry' (AP-MS) data. This approach includes both spoke and matrix models for interpreting AP-MS data in a network context. The ``spoke'' model considers only bait-prey interactions, whereas the ``matrix'' model assumes that each of the identified proteins (baits and prey) in a given AP-MS experiment interacts with each of the others. The spoke model has a high false-negative rate, whereas the matrix model has a high false-positive rate. Although, both statistical models have merits, a combination of both models has shown to increase the performance of machine learning classifiers in terms of their capabilities in discrimination between true and false positive interactions.

Depends R (>= 3.6.0)

Imports dplyr (>= 1.0.6), tibble (>= 3.1.2), tidyr (>= 1.1.3), magrittr (>= 2.0.1), plyr (>= 1.8.6), PRROC (>= 1.3.1), caret (>= 6.0.88), e1071 (>= 1.7.7), mice (>= 3.13.0), pROC (>= 1.17.0.1), ranger (>= 0.12.1)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Suggests knitr, markdown

VignetteBuilder knitr

NeedsCompilation no

Author Matineh Rahmatbakhsh [aut, cre]

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

Repository CRAN

Date/Publication 2021-06-17 08:20:05 UTC

R topics documented:

`cPASS` *cPASS*

Description

This function applies Comparative Proteomic Analysis Software Suite (CompPASS) model to score instances (e.g., bait-prey interactions (BPIs) in the data.frame. The CompPASS is a robust statistical scoring scheme for assigning confidence scores to bait-prey interactions (Sowa et al., 2009).This function was based on the source code. `https://github.com/dnusinow/cRomppass`

Usage

```
cPASS(datInput)
```

Arguments

`datInput`	Data frame with column names: Experiment.id, Replicate, Bait, Prey, and count (i.e., prey count).

Value

Data frame containing bait-prey pairs with average peptide spectrum match (PSMs), total PSMs, ratio total PSMs,Z-score,S-score,D-score and WD-score.

Author(s)

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

References

Huttlin, E. L., Ting, L., Bruckner, R. J., Gebreab, F., Gygi, M. P., Szpyt, J., et al. (2015). The BioPlex Network: A Systematic Exploration of the Human Interactome. Cell 162, 425-440.

Sowa, M. E., Bennett, E. J., Gygi, S. P., and Harper, J. W. (2009). Defining the human deubiquitinating enzyme interaction landscape. Cell 138, 389-403.

Examples

```
data(SampleDatInput)
datScoring <- cPASS(SampleDatInput)
head(datScoring)
```

diceCoefficient *diceCoefficient*

Description

This function applies Dice coefficient to score instances (e.g., bait-prey interactions (BPIs) in the data.frame.The Dice coefficient was first applied by Zhang et al., 2008 to score interactions between all identified proteins (baits and preys) in a given AP-MS experiment.

Usage

```
diceCoefficient(datInput)
```

Arguments

datInput	Data frame with column names: Experiment.id, Replicate, Bait, Prey, and count (i.e., prey count).

Value

Data frame containing bait-prey pairs with Dice coefficient score, a number between 0 and 1

Author(s)

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

References

Zhang, B., Park, B.-H., Karpinets, T., and Samatova, N. F. (2008). From pull-down data to protein interaction networks and complexes with biological relevance. Bioinformatics 24, 979-986.

Examples

```
data(SampleDatInput)
datScoring <- diceCoefficient(SampleDatInput)
head(datScoring)
```

jaccardCoefficient *jaccardCoefficient*

Description

This function computes the Jaccard similarity coefficient scores for instances (e.g., bait-prey interactions (BPIs)) in the data.frame.

Usage

```
jaccardCoefficient(datInput)
```

Arguments

`datInput`	Data frame with column names: Experiment.id, Replicate, Bait, Prey, and count (i.e., prey count).

Value

Data frame containing bait-prey pairs with the Jaccard coefficient score, a number between 0 and 1

Author(s)

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

Examples

```
data(SampleDatInput)
datScoring <- jaccardCoefficient(SampleDatInput)
head(datScoring)
```

overlapScore *overlapScore*

Description

This function computes the overlap similarity scores for instances (e.g., bait-prey interactions (BPIs)) in the data.frame.

Usage

```
overlapScore(datInput)
```

Arguments

`datInput`	Data frame with column names: Experiment.id, Replicate, Bait, Prey, and count (i.e., prey count).

Value

Data frame containing bait-prey pairs with the overlap score, a number between 0 and 1

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

```
data(SampleDatInput)
datScoring <- overlapScore(SampleDatInput)
head(datScoring)
```

---

rfTrain *rfTrain*

Description

The labeled feature matrix can be used as input for Random Forest (RF) classifier. The classifier then assigns each bait-prey pair a confidence score, indicating the level of support for that pair of proteins to interact. Hyperparameter optimization can also be performed to select a set of parameters that maximizes the model's performance. This function also computes the areas under the precision-recall (PR) and ROC curve to evaluate the performance of the classifier.

Usage

```
rfTrain(
  dtInput,
  impute = TRUE,
  p = 0.3,
  parameterTuning = TRUE,
  mtry = seq(from = 1, to = 10, by = 2),
  min_node_size = seq(from = 1, to = 9, by = 2),
  splitrule = c("gini"),
  metric = "Accuracy",
  resampling.method = "repeatedcv",
  iter = 5,
  repeats = 5,
  pr.plot = TRUE,
  roc.plot = TRUE
)
```

Arguments

dtInput Data frame containing instances with class labels

impute Logical value, indicating whether to impute missing values

p The percentage of data that goes to training; defaults to 0.3

parameterTuning Logical value; indicating whether to tune rf hyper parameters

mtry Number of variables to possibly split at in each node and it is bound by the number of variables in your model

min_node_size Minimal node size

splitrule Splitrule rule for classification: 'gini', 'extratrees' or 'hellinger' with default 'gini'

metric A string that specifies what summary metric will be used to select the optimal model; default to Accuracy

resampling.method The resampling method:'boot', 'boot632', 'optimism_boot', 'boot_all', 'cv', 'repeatedcv', 'LOOCV', 'LGOCV'; defaults to repeatedcv

iter Number of resampling iterations; defaults to 5

repeats for repeated k-fold cross validation only; defaults to 5

pr.plot Logical value, indicating whether to plot precision-recall (PR) curve

roc.plot Logical value, indicating whether to plot ROC curve

Value

Data frame containing a classification results for all instances in the data set, where positive confidence score corresponds to the level of support for the pair of proteins to be true positive, whereas negative score corresponds to the level of support for the pair of proteins to be true negative.

Author(s)

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

Examples

```
data(testdfClassifier)

predidcted_RF <-
rfTrain(testdfClassifier,impute = FALSE, p = 0.3, parameterTuning = FALSE,
mtry  = seq(from = 1, to = 5, by = 1),
min_node_size = seq(from = 1, to = 5, by = 1),
splitrule =c("gini"),metric = "Accuracy",
resampling.method = "cv",iter = 2,repeats = 2,
pr.plot = TRUE, roc.plot = FALSE)
head(predidcted_RF)
```

SampleDatInput *Test data for scoring*

Description

Bait-Prey Interactions (BPIs)

Usage

```
data(SampleDatInput)
```

Details

- Experiment ID
- Replicate
- Bait
- Prey
- counts

Author(s)

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

simpsonCoefficient *simpsonCoefficient*

Description

This function computes the Simpson similarity coefficient scores for instances (e.g., bait-prey interactions (BPIs)) in the data.frame.

Usage

```
simpsonCoefficient(datInput)
```

Arguments

datInput	Data frame with column names: Experiment.id, Replicate, Bait, Prey, and count (i.e., prey count).

Value

Data frame containing bait-prey pairs with Simpson coefficient, a number between 0 and 1

Author(s)

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

Examples

```
data(SampleDatInput)
datScoring <- overlapScore(SampleDatInput)
head(datScoring)
```

svmTrain *svmTrain*

Description

The labeled feature matrix can be used as input for Support Vector Machines (SVM) classifier. The classifier then assigns each bait-prey pair a confidence score, indicating the level of support for that pair of proteins to interact. Hyperparameter optimization can also be performed to select a set of parameters that maximizes the model's performance. This function also computes the areas under the precision-recall (PR) and ROC curve to evaluate the performance of the classifier.

Usage

```
svmTrain(
  dtInput,
  impute = TRUE,
  p = 0.3,
  parameterTuning = TRUE,
  cost = seq(from = 2, to = 10, by = 2),
  gamma = seq(from = 0.01, to = 0.1, by = 0.02),
  kernel = "radial",
  ncross = 10,
  pr.plot = TRUE,
  roc.plot = TRUE
)
```

Arguments

`dtInput` Data frame containing instances with class labels

`impute` Logical value, indicating whether to impute missing values

`p` The percentage of data that goes to training; defaults to 0.3

`parameterTuning` Logical value; indicating whether to tune SVM hyper parameters

`cost` Cost of constraints violation

`gamma` Parameter needed for all kernels except linear

`kernel` Kernel type: 'linear', 'polynomial', 'sigmoid', or 'radial'; defaults to 'radial' `ncross` K-fold cross validation on the training data is performed to assess the quality of the model; defaults to 10

`pr.plot` Logical value, indicating whether to plot precision-recall (PR) curve

`roc.plot` Logical value, indicating whether to plot ROC curve

Value

Data frame containing a classification results for all instances in the data set, where positive confidence score corresponds to the level of support for the pair of proteins to be true positive, whereas negative score corresponds to the level of support for the pair of proteins to be true negative.

Author(s)

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

Examples

```
data(testdfClassifier)
predidcted_SVM <-
svmTrain(testdfClassifier,impute = FALSE,p = 0.3,parameterTuning = FALSE,
cost = seq(from = 2, to = 10, by = 2),
gamma = seq(from = 0.01, to = 0.10, by = 0.02),
kernel = "radial",ncross = 10,
pr.plot = FALSE, roc.plot = TRUE)
head(predidcted_SVM)
```

`testdfClassifier` *Test data for classifier*

Description

Scored data.frame

Usage

```
data(testdfClassifier)
```

Author(s)

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

Weighted.matrixModel *Weighted.matrixModel*

Description

This function computes the weighted matrix model for instances (e.g., bait-prey interactions (BPIs)) in the data.frame.The output of the weighted matrix model includes the number of experiments for which the pair of proteins is co-purified (i.e., k) and -1*log(P-value) of the hypergeometric test (i.e., logHG) given the experimental overlap value, each protein's total number of observed experiments, and the total number of experiments (Drew et al., 2017).

Usage

```
Weighted.matrixModel(datInput)
```

Arguments

`datInput`	Data frame with column names: Experiment.id, Replicate, Bait, Prey, and count (i.e., prey count).

Value

Data frame containing bait-prey pairs with k (i.e.,number of co-purifications) & logHG (i.e., -1*log(P-val) of the hypergeometric test)

Author(s)

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

References

Drew, K., Lee, C., Huizar, R. L., Tu, F., Borgeson, B., McWhite, C. D., et al. (2017). Integration of over 9,000 mass spectrometry experiments builds a global map of human protein complexes. Mol. Syst. Biol. 13, 932.

Examples

```
data(SampleDatInput)
datScoring <- Weighted.matrixModel(SampleDatInput)
head(datScoring)
```

Index