Package 'RAP'

July 28, 2016

Type Package

Title Rank aggregation-based data fusion for gene prioritization

Version 1.0

Date 2016-07-28

Author Jingjing Zhai and Chuang Ma

Maintainer Jingjing Zhai <zhaijingjing603@gmail.com>

Depends R (>= 3.3.0)

Imports randomForest, protr, missForest, RobustRankAggreg, snowfall, pROC

Description RAP can be used to perform the gene prioritization in Arabidopsis thaliana and 28 non-plant species. The input of RAP are a set of genes of interest and the network-based gene prioritization results from AraNet v2 system, while the output are the ranks of candidate genes. RAP has been successfully applied to prioritize flowering-time genes in Arabidopsis. The genome-wide experimental results indicate that RAP can be used as a complement to the network-based gene prioritization algorithm (e.g., AraNet v2) for accurately prioritizing candidate genes underlying biological processes or agricultural traits.

LazyData true

License GPL (>= 2)

URL http://bioinfo.nwafu.edu.cn/software

NeedsCompilation no

Repository CRAN

RoxygenNote 5.0.1

R topics documented:

`AAindex` *Physiochemical Properties of Amino Acid*

Description

533 physiochemical properties of 20 amino acids.

Usage

```
data(AAindex)
```

Author(s)

Jingjing Zhai, Chuang Ma

`CrossValidation` *Cross validation*

Description

Performance evaluation of the integrative random forest-based gene prioritization algorithm RafSee in distinguishing positives and negatives.

Usage

```
CrossValidation(seed = 1, featureMat, positives,
                negatives, cross = 10, cpus = 1)
```

Arguments

`seed`	An integer number specifying a random seed for randomly partitioning dataset.
`featureMat`	A numeric feature matrix.
`positives`	A character vector representing positive samples.
`negatives`	A character vector representing negative samples.
`cross`	Number of fold for cross validation.
`cpus`	An integer number specifying the number of cpus to be used for parallel computing.

Value

A list containing results from each fold cross validation including:

`positives.train`
: positive samples used to train prediction model.

`negatives.train`
: negative samples used to train prediction model.

`positives.test`
: positive samples used to test prediction model.

`negatives.test`
: negative samples used to test prediction model.

`positives.train.score`
: scores of positive samples in training dataset predicted by random foreste.

`negatives.train.score`
: scores of negative samples in training dataset predicted by random forest.

`positives.test.score`
: scores of positive samples in testing dataset predicted by random forest.

`negatives.test.score`
: scores of negative samples in testing dataset predicted by random forest.

`train.AUC`
: AUC value of random forest on training dataset.

`test.AUC`
: AUC value of random forest on testing dataset.s

Author(s)

Jingjing Zhai, Chuang Ma

Examples

```
## Not run:
  positives <- c("AT1G01060", "AT1G09530", "AT1G09570", "AT1G12610")
  cvRes <- CrossValidation(featureMat = featureMat, positives = positives,
                   negatives  = negatives, cpus = 1)
  ## featureMat can be calculated by function FeatureExtract
  ## negatives can be calculated by function selectNegSamples

## End(Not run)
```

`FeatureExtract` *Extracting sequence-based features.*

Description

This function generates sequence-based features from protein sequences using five scoring schemes.

Usage

```
FeatureExtract(ProteinSeq, feature = c("AAC", "DAAC", "PAAC", "APAAC", "PCP"),
               lambda = 5, w = 0.05)
```

Arguments

ProteinSeq	A list of protein sequences.
feature	A vector of encoding schemes.
lambda	The lambda parameter for the PAAC and APAAC-related features, default is 5.
w	The weighting parameter for the PAAC and APAAC-related features, default is 0.05.

Value

A feature matrix with genes in rows, features in columns

Author(s)

Jingjing Zhai, Chuang Ma

Examples

```
## Not run:

  ##generate a list of protein sequence
 exampleSeq1 <- "MVQYNFKRITVVPNGKEFVDIILSRTQRQTPTVVHKGYKINRLRQFYMRKVKYTQTNFHAKLSAIIDEFP"
  exampleSeq2 <- "MDSESESKLISFISQLVSRNNTDSENISCMIQTISLVSSMDLKSQPKPESKLMSLVTQTISLFNSM"
 featureMat <- FeatureExtract(ProteinSeq = list(exampleSeq1, exampleSeq2), feature = "AAC")

## End(Not run)
```

---

geneDescriptionInfo *Description information of 27,416 protein-coding genes in Arabidopsis thaliana.*

Description

Descriptive information including locus, description and Wiki description in Arabidopsis thaliana.

Usage

```
data(geneDescriptionInfo)
```

Author(s)

Jingjing Zhai, Chuang Ma

LOOCV *leave-one-out cross-validation*

Description

Leave-one-out cross-validation algorithm is performed to train and test the integrative random-forest gene prioritization algorithm RafSee.

Usage

```
LOOCV(featureMat, positives, negatives, cpus = 1, predictSample = NULL)
```

Arguments

featureMat	A numeric matrix of features where rows represent genes, cols represent features
positives	A character vector of positive samples
negatives	A character vector of negative samples
cpus	an integer number specifying the number of cpus to be used for parallel computing, the default is 1
predictSample	A vector of testing samples, if it is NULL, all genes excluding positive smaples were used

Value

Predictive score for each leave-one-out cross-validation

Author(s)

Jingjing Zhai, Chuang Ma

Examples

```
## Not run:
  positives <- c("AT1G01060", "AT1G09530", "AT1G09570", "AT1G12610")
  loocvRes <- LOOCV(featureMat = featureMat, positives = positives,
                    negatives  = negatives, cpus = 1)
  ## featureMat can be calculated by function FeatureExtract
  ## negatives can be calculated by function selectNegSamples

## End(Not run)
```

RAP *Rank Aggregation-based Data Fusion for Gene Prioritization*

Description

This function prioritize genes using sequence-based and network-based model

Usage

```
RAP(netPredResult, positives, negatives = NULL, featureSel = TRUE,
    featureMat = NULL, ProteinSeq,  PPIMat, GenomeGeneID,
    ntree = 500)
```

Arguments

`netPredResult`	Full path of gene prioritization results from the network-based gene prioritization algorithms (e.g., AraNet v2).
`featureMat`	A numeric matrix of features where rows represent genes, cols represent features.
`positives`	A character vector of positive samples.
`negatives`	A character vector of negative samples.
`featureSel`	A logical value, where TRUE selecting significant features between positive and negative samples.
`ProteinSeq`	A list of protein sequence, the parameter are not required unless patameter featureMat is NULL.
`PPIMat`	A matrix of PPI, which contains 3 coloums represent protein1, protein2, score respectively, the patameter is not required unless paramter negatives are NULL.
`GenomeGeneID`	A vector of genome ID, the parameter, the parameter is not required unless parameter negatives is NULL.
`ntree`	Number of trees to grow when using random forest, the default is 500.

Value

A matrix with the rank of genes and descriptive information

Author(s)

Jingjing Zhai, Chuang Ma

Examples

```
## Not run:
  positives <- c("AT1G01060", "AT1G09530", "AT1G09570", "AT1G12610")
  res <- RAP(netPredResult = "/home/malab/AraNetPred.txt",
             positives = positives, negatives = negatives,
             featureSel = TRUE, featureMat = featureMat)
  ## featureMat can be calculated by function FeatureExtract
  ## negatives can be calculated by function selectNegSamples
  ## The sample results of AraNet v2 (AraNetPred.txt) can be
     downloaded from http://bioinfo.nwafu.edu.cn/software

## End(Not run)
```

`selectNegSamples` *selecting negative samples for training RafSee*

Description

Negative samples are selected based on their connectivity with positive samples in protein-protein interaction network. Of note, the user can also have a try to randomly selected negative samples with a given number.

Usage

```
selectNegSamples(positives, PPIMat = NULL, balanced = TRUE,
                 ratio = 1, GenomeGeneID)
```

Arguments

`positives`	A vector of positive samples.
`PPIMat`	A matrix of PPI, which contains 3 coloums represent protein1, protein2, score respectively;Of note, negative samples will be selected randomly if this parameter is not assigned.
`balanced`	A logical value, where TRUE represents balance the positive and negative samples according to the ratio.
`ratio`	A numuric value of the the ratio between negative and positive samples.
`GenomeGeneID`	A vector of genome ID.

Value

A vector of selected negative samples

Author(s)

Jingjing Zhai, Chuang Ma

Examples

```
## Not run:
 positives <- c("AT1G01060", "AT1G09530", "AT1G09570", "AT1G12610")
GenomeID <- c("AT1G01060", "AT1G09530", "AT1G09570", "AT1G12610", "AT1G77300", "AT1G79730")
negatives <- selectNegSamples(positives = positives, PPIMat = PPIMat, GenomeID = GenomeID)

## End(Not run)
```

`sigFeatureSelection` *Selecting informative features*

Description

This function extracting informative features with feature selection algorithms including the student's t-test and chi-square test feature selection algorithms.

Usage

```
sigFeatureSelection(featureMatrix, positives, negatives, binary = FALSE, level = 0.05)
```

Arguments

`featureMatrix`	A numeric matrix of features where rows represent genes, cols represent features
`positives`	A character vector of positive samples
`negatives`	A character vector of negative samples
`binary`	A logical value, where TRUE represents the features are binary with 0 and 1, the default is FALSE
`level`	A numeric value recording the significant level, the default is 0.05

Value

A numeric feature matrix with only significant features were contained

Author(s)

Jingjing Zhai, Chuang Ma

Examples

```
## Not run:
  positives <- c("AT1G01060", "AT1G09530", "AT1G09570", "AT1G12610")
  sifFeatureMat <- sigFeatureSelection(featureMatrix = featureMat,
                                positives = positives,
                                negatives = negatives)
  ## featureMat can be calculated by function FeatureExtract
  ## negatives can be calculated by function selectNegSamples

## End(Not run)
```

testData *Example data for RAP*

Description

Positive samples, negative samples and feature matrix for RAP.

Usage

```
data(testData)
```

Author(s)

Jingjing Zhai, Chuang Ma

Index