Package 'DrugClust'

April 23, 2016

Type Package

Title Implementation of a Machine Learning Framework for Predicting
Drugs Side Effects

Version 0.2

Date 2016-04-23

Author Giovanna Maria Dimitri

Maintainer Giovanna Maria Dimitri <gmd43@cam.ac.uk>

Description
An implementation of a Machine Learning Framework for prediction of new drugs Side Effects.
Firstly drugs are clustered with respect to their features description and secondly predictions are made, according to Bayesian scores.
Moreover it can perform protein enrichment considering the proteins clustered together in the first step of the algorithm.
This last tool is of extreme interest for biologist and drug discovery purposes, given the fact that it can be used either as a validation of the clusters obtained, as well as for the possible discovery of new interactions between certain side effects and non targeted pathways.
Clustering of the drugs in the feature space can be done using K-Means, PAM or K-Seeds (a novel clustering algorithm proposed by the author).

License GPL-2

LazyData TRUE

RoxygenNote 5.0.0

NeedsCompilation no

Imports ROCR, MESS, cclust, cluster, e1071, utils, base

Repository CRAN

Date/Publication 2016-04-23 14:19:13

R topics documented:

AUC 2
AUPR 3

CreateFolds 4
DrugClustKMeans 4
DrugClustKMeansEnrichment 5
DrugClustKSeeds 6
DrugClustKSeedsEnrichment 6
DrugClustPAM 7
DrugClustPAMEnrichment 8
Enrichment_Proteins 8
InitFeatures 9
InitSideEffect 10
KMeansClusteringAlgorithm 10
KMeansModel 11
KSeedsClusters 12
KSeedsScores 12
PAM 13
PAM_Model 14
PredictionKMeans 15
PredictionKSeeds 15
Prediction_PAM 16
RandomSeedGenerator 17
SeedSelection 18

Index 19

AUC *AUC*

Description

Function Implementing metrics calculation of AUC

Usage

```
AUC(predizioni, testpharmat, vectorAUC, name)
```

Arguments

`predizioni`	matrix of predictions
`testpharmat`	matrix of test for the side effects
`vectorAUC`	empty vector where the AUC values will be saved
`name`	string stating the name of the clustering Algorithm used, KSeeds, Kmeans or PAM

Value

vectorAUC vector containing the various AUC values for the various folds. Moreover the function draw the graph of AUC

Examples

```
#' #Function for obtaining AUC
#Once you have obtained predizioni with the Prediction function you can apply
#this AUC function using the following command (testpharmat sideeffects test matrix)
#vectorAUC<-numeric()
#vectorAUC<-AUC(predizioni,testpharmat,vectorAUC,"KSeeds")
```

AUPR *AUPR*

Description

Function Implementing metrics calculation of AUPR

Usage

```
AUPR(predizioni, testpharmat, vectorAUPR, name)
```

Arguments

`predizioni`	matrix of predictions
`testpharmat`	matrix of test for the Side Effects
`vectorAUPR`	empty vector to store AUPR
`name`	name of the clustering algorithm used (KSeeds, KMeans,PAM)

Value

vectorAUPR vector containing AUPR values for the various folds, the function also draws AUPR graphs

Examples

```
#Function for obtaining AUC
#Once you have obtained predizioni with the Prediction function you can apply
#this AUPR function using the following command (testpharmat sideeffects test matrix)
#vectorAUPR<-numeric()
#vectorAUPR<-AUPR(predizioni,testpharmat,vectorAUPR,"KSeeds")
```

CreateFolds *CreateFolds*

Description

Create the folds given the features matrix

Usage

```
CreateFolds(features, num_folds)
```

Arguments

features	is the features matrix that has to be divided in folds for performing cross validation
num_folds	number of folds desired

Value

folds: the elements divided in folds

Examples

```
r <- 8
c <- 10
m0 <- matrix(0, r, c)
features<-apply(m0, c(1,2), function(x) sample(c(0,1),1))
folds<-CreateFolds(features,4)
```

DrugClustKMeans *DrugClustKMeans*

Description

Function Implementing DrugClust with KMeans algorithm

Usage

```
DrugClustKMeans(num_folds, num_clusters, num_iterations, features, side_effects)
```

Arguments

num_folds	number of folds
num_clusters	number of clusters
num_iterations	number of iterations
features	features matrix
side_effects	side_effects matrix

Value

(list(AUCFinal,AUPRFinal)) first value is the mean AUC on the various folders, second value is the mean AUPR on the various folders

Examples

```
# num_folds=3
# num_clusters=4
# num_iterations= 5
#features is the features matrix (see InitFeatures function)
# side effects is the matrix containing side effects (see InitSideEffects function)
#result<-DrugClustKMeans(num_folds,num_clusters,num_iterations,features,side_effects)
```

DrugClustKMeansEnrichment
DrugClustKMeansEnrichment

Description

Function Implementing DrugClust with KMeans and Enrichment

Usage

```
DrugClustKMeansEnrichment(num_clusters, features, pharmat)
```

Arguments

`num_clusters`	number of clusters desired
`features`	matrix features
`pharmat`	matrix of side effects

Value

number of pathways for various clusters

Examples

```
#features is the features matrix
#resultSeeds<-DrugClustKMeansEnrichment(4,features)
```

DrugClustKSeeds *DrugClustKSeeds*

Description

Function Implementing metrics calculation DrugClust

Usage

```
DrugClustKSeeds(num_folds, num_clusters, num_iterations, features, side_effects)
```

Arguments

num_folds	number of folds
num_clusters	number of clusters
num_iterations	number of iterations
features	features matrix
side_effects	side_effects matrix

Value

(list(AUCFinal,AUPRFinal)) first value is the mean AUC on the various folders, second value is the mean AUPR on the various folders

Examples

```
# num_folds=3
# num_clusters=4
# num_iterations= 5
# features is the features matrix (see InitFeatures function)
# side effects is the matrix containing side effects (see InitSideEffects function)
#result<-DrugClustKSeeds(num_folds,num_clusters,num_iterations,features,side_effects)
```

DrugClustKSeedsEnrichment *DrugClustKSeedsEnrichment*

Description

Function Implementing DrugClust with KSeeds and Enrichment

Usage

```
DrugClustKSeedsEnrichment(num_clusters, features, pharmat)
```

Arguments

`num_clusters`	number of clusters
`features`	matrix of features
`pharmat`	matrix of side effects

Value

number of pathways for various clusters

Examples

```
#features is the features matrix
#resultSeeds<-DrugClustKSeedsEnrichment(4,features)
```

DrugClustPAM *DrugClustPAM*

Description

Function Implementing DrugClust with PAM algorithm

Usage

```
DrugClustPAM(num_folds, num_clusters, num_iterations, features, side_effects)
```

Arguments

`num_folds`	number of folds
`num_clusters`	number of clusters
`num_iterations`	number of iterations
`features`	features matrix
`side_effects`	side_effects matrix

Value

(list(AUCFinal,AUPRFinal)) first value is the mean AUC on the various folders, second value is the mean AUPR on the various folders

Examples

```
# num_folds=3
# num_clusters=4
# num_iterations= 5
#features is the features matrix (see InitFeatures function)
# side effects is the matrix containing side effects (see InitSideEffects function)
#result<-DrugClustPAM(num_folds,num_clusters,num_iterations,features,side_effects)
```

DrugClustPAMEnrichment *DrugClustPAMEnrichment*

Description

Function Implementing DrugClust with PAM algorithm and Enrichment

Usage

```
DrugClustPAMEnrichment(num_clusters, features)
```

Arguments

num_clusters	number of clusters desired
features	matrix of features

Value

number of pathways for various clusters

Examples

```
#features is the features matrix
#resultSeeds<-DrugClustPAMEnrichment(4,features)
```

Enrichment_Proteins *Enrichment_Proteins*

Description

Function Performing Proteins Enrichment using Gene Ontology

Usage

```
Enrichment_Proteins(features, num_clusters, clusters)
```

Arguments

features	matrix of features
num_clusters	number of clusters
clusters	clusters returned from the clustering algorithms

Value

vector_numb_pathway return a vector telling in how many pathways the various clusters are involved

Examples

```
#feature is the feature matrix
# pamx is the result of the PAM function
# and pamx$clustering gives the list assigning each element to a certain cluster
#all_pathways<-Enrichment_Proteins(features,4,pamx$clustering)
```

InitFeatures *InitFeatures*

Description

Initialize the features matrix. The data needs to be binary matrices where each row is a drug, and columns represents drugs features. If the element in position ij is 1 it means that the ith drug interacts with the jth element (for example a protein). The same for the matrix where side effects are stored.

Usage

```
InitFeatures(namefeatures)
```

Arguments

namefeatures	name of the file where the features are stored.The file needs to be in the same folder where you have the code.

Value

The matrix containing drugs features

Examples

```
#Generate a sample features binary matrix
#for example you will fin the file bioma2.txt which is a sample file for feature matrix
#you can therefore type the command features<-InitFeatures("bio2mat.txt") to upload it
```

InitSideEffect *InitSideEffect*

Description

Initialize the matrix of features and Side Effects

Usage

```
InitSideEffects(nameSideEffects)
```

Arguments

nameSideEffects	name of the file where the side effects are stored. The format has to be a binary matrix, where the rows are the drugs and columns are the various side effects (1/0 meaning presence or absence of a certain side effect).

Value

The matrix containing drugs side effects

Examples

```
#Generate a sample features binary matrix
#for example you will fin the file pharmat.txt which is a sample file of side_effects matrix
#you can therefore type the command side_effects<-InitSideEffects("pharmat.txt") to upload it
```

KMeansClusteringAlgorithm *KMeans*

Description

KMeans clustering algorithm

Usage

```
KMeans(train, num_clusters)
```

Arguments

train	matrix of train features
num_clusters	number of clusters desired

Value

cl list containing the clusters ownerships

Examples

```
#use the initFeatures to upload train feature matrix
#see also KSeedsClusters function to see a similar example
#with a toy matrix
#cl<-KMeans(train,num_clusters)
```

KMeansModel *KMeansModel*

Description

Function finding the Bayesian Model given the KMeans clustering algorithm

Usage

```
KMeansModel(train, trainpharmat, num_clusters, cl)
```

Arguments

`train`	matrix of train features
`trainpharmat`	matrix of training of side_effects
`num_clusters`	number of clusters desired
`cl`	results of the KMeans model clustering function

Value

A Bayesian matrix of model for predictions, given the KMeans clustering

Examples

```
#First call the KMeans function and obtain cl (list of clusters)
#train is the feature matrix of train
#trainpharmat is the side effect matrix of train
#A<-KMeansModel(train,trainpharmat,4,cl)
```

KSeedsClusters *KSeedsClusters*

Description

Function Implementing KSeeds. K-Seeds, firstly randomly chooses a number of drugs (renamed Seeds) equal to the number of clusters desired. Then, the other drugs are assigned to a cluster with respect to Hamming Distance between the drug and the seed of a certain cluster. Cluster seeds are not recomputed at each iteration. This allows a speed up in terms of computational complexity and the algorithm terminates when all the drugs have been assigned.

Usage

```
KSeedsClusters(train, num_clusters, Seed, s)
```

Arguments

`train`	train matrix of features
`num_clusters`	number of clusters desired
`Seed`	subset of drugs features matrix, with just the Seeds as rows
`s`	the seeds of the clusters

Value

clusters list indicating the cluster to which each drug belongs to

Examples

```
r <- 8
c <- 10
m0 <- matrix(0, r, c)
num_clusters=4
features<-apply(m0, c(1,2), function(x) sample(c(0,1),1))
s<-RandomSeedGenerator(num_clusters,nrow(features))
Seed<-SeedSelection(features,num_clusters,s)
clusters<-KSeedsClusters (features,num_clusters,Seed,s)
```

KSeedsScores *KSeedsScores*

Description

Function for obtaining the Bayesian prediction scores using KSeeds clustering

Usage

```
KSeedsScores(train, trainpharmat, num_clusters, Seed, s, clusters)
```

Arguments

`train`	train matrix of features
`trainpharmat`	train matrix of side effects
`num_clusters`	number of clusters desired
`Seed`	subset of the features matrix containing only the Seeds drugs
`s`	the seeds of the clusters
`clusters`	the list of clusters where the various drugs are

Value

A matrix containing prediction scores for each cluster

Examples

```
r <- 8
c <- 10
m0 <- matrix(0, r, c)
num_clusters=4
features<-apply(m0, c(1,2), function(x) sample(c(0,1),1))
#Generate a sample side effects binary matrix
r1 <- 8
c1 <- 10
m1 <- matrix(0, r1, c1)
side_effects<-apply(m1, c(1,2), function(x) sample(c(0,1),1))
s<-RandomSeedGenerator(num_clusters,nrow(features))
Seed<-SeedSelection(features,num_clusters,s)
clusters<-KSeedsClusters (features,num_clusters,Seed,s)
A<-KSeedsScores(features,side_effects,num_clusters,Seed,s,clusters)
```

`PAM` *PAM*

Description

PAM clustering algorithm

Usage

```
PAM(train, num_clusters)
```

Arguments

`train`	matrix of train features
`num_clusters`	number of clusters desired

Value

pamx structure with various values resulting from PAM clustering algorithm

Examples

```
#train is the train feature matrix
#pamx<-PAM(train,4)
```

PAM_Model *PAM_Model*

Description

PAM clustering algorithm Model

Usage

```
PAM_Model(pamx, num_clusters, trainpharmat, train)
```

Arguments

`pamx`	result of pam clustering algorithm
`num_clusters`	number of clusters desired
`trainpharmat`	matrix of training for side effects
`train`	matrix of train features

Value

A matrix of model for prediction of uncharacterised drugs, given PAM clustering

Examples

```
#pamx is the result of the PAM function
#trainpharmat is the side effect train matrix
#train is the feature train matrix
#A<-PAM_Model(pamx,4,trainpharmat,train)
```

`PredictionKMeans` *PredictionKMeans*

Description

Function finding the predictions for the uncharacterized drugs given the KMeans clustering algorithm

Usage

```
PredictionKMeans(A, cl, test)
```

Arguments

`A`	Bayesian model given by the application of KMeansModel algorithm
`cl`	structure of clusters given by the KMeans function
`test`	test matrix of drugs

Value

predizioni matrix with a number of rows equal to the number of clusters and a number of columns equal to the features

Examples

```
# A will be the result of the previous call of KMeans model funcion
#cl will be the result of KMeans function
#test is the test feature matrix
#predizioni<-PredictionKMeans(A,cl,test)
```

`PredictionKSeeds` *PredictionKSeeds*

Description

Function implementing predictions for uncharacterized drugs

Usage

```
PredictionKSeeds(test, Seed, num_clusters, A, numcolsideffects)
```

Arguments

`test`	test drugs features matrix
`Seed`	matrix of seeds initialize in the KSeed algorithm
`num_clusters`	number of clusters desired
`A`	matrix of Naive Bayes predictions scores, result of KSeedsScores function
`numcolsideffects`	number of sideeffects

Value

predizioni matrix containing predictions for the various uncharacterized drugs

Examples

```
r <- 8
c <- 10
m0 <- matrix(0, r, c)
num_clusters=4
features<-apply(m0, c(1,2), function(x) sample(c(0,1),1))
#Generate a sample side effects binary matrix
r1 <- 8
c1 <- 15
m1 <- matrix(0, r1, c1)
side_effects<-apply(m1, c(1,2), function(x) sample(c(0,1),1))
folds<-CreateFolds(features,2)
i=0
train = features[folds != i,]
trainpharmat = side_effects[folds != i,]
test = features[folds == i,]
testpharmat = side_effects[folds == i,]
s<-RandomSeedGenerator(num_clusters,nrow(train))
Seed<-SeedSelection(train,num_clusters,s)
clusters<-KSeedsClusters (train,num_clusters,Seed,s)
A<-KSeedsScores(train,trainpharmat,num_clusters,Seed,s,clusters)
predizioni<-PredictionKSeeds(test,Seed,num_clusters,A,ncol(side_effects))
```

`Prediction_PAM` *Prediction_PAM*

Description

PAM prediction models

Usage

```
PredictionPAM(A, pamx, test, numb_sideEffects)
```

Arguments

`A`	prediction scores matrix
`pamx`	result of pam clustering algorithm
`test`	test features matrix
`numb_sideEffects`	number of side effects

Value

predizioni matrix of predictions given PAM clustering

Examples

```
#A is the result of PAM_Model function
#pamx comes from the PAM function
#test is the feature test matrix
#predizioni<-PredictionPAM(A,pamx,test)
```

`RandomSeedGenerator` *RandomSeedGenerator*

Description

Initizalize seeds for the KSeeds clustering algorithm

Usage

```
RandomSeedGenerator(num_clusters, numbrowfeatures)
```

Arguments

`num_clusters`	number of clusters desired
`numbrowfeatures`	number of rows of the features matrix

Value

s list of seeds

Examples

```
r <- 8
c <- 10
m0 <- matrix(0, r, c)
num_clusters=4
features<-apply(m0, c(1,2), function(x) sample(c(0,1),1))
s<-RandomSeedGenerator(4,nrow(features))
```

SeedSelection *SeedSelection*

Description

Given the seeds, it creates the submatrix of the features where the rows are just the seeds drugs

Usage

```
SeedSelection(features, num_clusters, s)
```

Arguments

features	train matrix of features (in the case of k-folding is the matrix of features)
num_clusters	number of clusters desired
s	the list of seeds

Value

Seed subset of the feature matrix, where rows are the Seed drugs, and columns the relative features

Examples

```
r <- 8
c <- 10
m0 <- matrix(0, r, c)
num_clusters=4
features<-apply(m0, c(1,2), function(x) sample(c(0,1),1))
s<-RandomSeedGenerator(num_clusters,nrow(features))
Seed<-SeedSelection(features,num_clusters,s)
```

Index

AUC, 2
AUPR, 3

CreateFolds, 4

DrugClustKMeans, 4
DrugClustKMeansEnrichment, 5
DrugClustKSeeds, 6
DrugClustKSeedsEnrichment, 6
DrugClustPAM, 7
DrugClustPAMEnrichment, 8

Enrichment_Proteins, 8

InitFeatures, 9
InitSideEffect, 10
InitSideEffects *(*InitSideEffect*)*, 10

KMeans *(*KMeansClusteringAlgorithm*)*, 10
KMeansClusteringAlgorithm, 10
KMeansModel, 11
KSeedsClusters, 12
KSeedsScores, 12

PAM, 13
PAM_Model, 14
Prediction_PAM, 16
PredictionKMeans, 15
PredictionKSeeds, 15
PredictionPAM *(*Prediction_PAM*)*, 16

RandomSeedGenerator, 17

SeedSelection, 18