Package 'timma'

February 28, 2015

Type Package

Title Target Inhibition Interaction using Maximization and Minimization Averaging

Version 1.2.1

Date 2015-02-24

Author Liye He, Krister Wennerberg, Tero Aittokallio and Jing Tang

Maintainer Jing Tang <jing.tang@helsinki.fi>

Description Prediction and ranking of drug combinations based on their drug-target interaction profiles and single-drug sensitivities in a given cancer cell line or patient-derived sample.

License Artistic License 2.0

Imports Rcpp (>= 0.11.2), QCA (>= 1.1-3), reshape2

LinkingTo Rcpp, RcppArmadillo

Suggests R.rsp

VignetteBuilder R.rsp

biocViews Software, StatisticalMethod

Depends R (>= 2.10)

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-02-28 12:17:41

R topics documented:

timma-package *Target Inhibition inference using Maximization and Minimization Averaging*

Description

Due to the exponentially increasing number of potential drug and target combinations, it is meaningful to select the most promising combinations based on computational models. The TIMMA model was proposed to utilize drug-target interaction data and drug sensitivity data to infer the effects of drug combinations. This R package TIMMA is the implementation of the TIMMA model. It consists of the following components: (a) model selection using the sffs algorithm; (b) model construction using the maximization and minimization averaging rules; (c) ranking of drug combinations according to their synergy scores and a target inhibition network.

Details

Package:	TIMMA
Type:	Package
Version:	0.99.0
Date:	2014-10-07
License:	Artistic License 2.0

Author(s)

Liye He <liye.he@helsinki.fi>

References

Tang J, Karhinen L, Xu T, Szwajda A, Yadav B, Wennerberg K, Aittokallio T. Target inhibition networks: predicting selective combinations of druggable targets to block cancer survival pathways. PLOS Computational Biology 2013; 9: e1003226.

Examples

```
## Not run:
data(tyner_interaction_binary)
data(tyner_sensitivity)
median_sensitivity<-tyner_sensitivity[, 1]
results<-timma(tyner_interaction_binary, median_sensitivity)

## End(Not run)
```

binarizeDrugTargets *Binarize the drug target profile data*

Description

A function for binarizing the drug target profile data.

Usage

```
binarizeDrugTargets(profile, method = "universal", threshold = "100nM")
```

Arguments

profile	a matrix with non-binary entries. The rows are drugs and the columns are targets.
method	a string to specify the methods used for binarizing the data. When it is "universal", an universal threshold is used. In such case, another parameter threshold can only be one of "100nM", "1000nM", and "10000nM". When it is "drug-specific", the threshold used for binarization depends on each drug and the parameter threshold can be only one of "10fold", "50fold", and "100fold".
threshold	a string to specify the threshold.

Value

A matrix contains the binarized drug target data.

Author(s)

Jing Tang <jing.tang@helsinki.fi>

Examples

```
data(davis)
profile<-binarizeDrugTargets(davis, method="drug-specific", threshold="50fold")
```

binarySet *Search for supersets and subsets*

Description

A function for searching the supersets and subsets of the binary drug-target interaction data.

Usage

```
binarySet(profile_data)
```

Arguments

profile_data	the binary drug-target interaction matrix with row indexes as drugs and column indexes as targets.

Value

A list contains the following components:

superset	all the possible supersets of the input drug-target interaction data
subset	all the possible subsets of the input drug-target interaction data

Author(s)

Liye He <liye.he@helsinki.fi>

Examples

```
data(tyner_interaction_binary)
sets<-binarySet(tyner_interaction_binary[1, 1:3])
```

---

ci *The combination index extracted from Figure 1B of the Miller study*

Description

The combination index extracted from Figure 1B of the Miller study

---

davis *Drug-target profile for 72 drugs and 442 targets.*

Description

Binding results (Kd's in nM) for 72 drugs vs 442 kinase assays. Blank fields indicate interactions that were not detected in a 10 uM primary screen.

References

Davis et al. Comprehensive analysis of kinase inhibitor selectivity. Nat. Biotechnol. 2011 29, 1046-51.

dec2bin *Convert decimal values to binary values*

Description

A function to convert decimal values to binary values

Usage

```
dec2bin(number, bits)
```

Arguments

number	the decimal number that needs to be converted.
bits	the number of bits of the result

Value

a vector contains of binary values 0 and 1.

Author(s)

Liye He <liye.he@helsinki.fi>

Examples

```
dec2bin(8, 5)
```

drawGraph *Draw graph function*

Description

A function to draw the target inhibition network.

Usage

```
drawGraph(draw_data)
```

Arguments

draw_data	a data frame combining drug-target interaction data with drug sensitivity. The column names must be upper case.

Value

An image in both pdf and nnf format of the estimated target inhibition network.

Author(s)

Jing Tang <jing.tang@helsinki.fi>

References

Tang J, Karhinen L, Xu T, Szwajda A, Yadav B, Wennerberg K, Aittokallio T. Target inhibition networks: predicting selective combinations of druggable targets to block cancer survival pathways. PLOS Computational Biology 2013; 9: e1003226.

Examples

```
## Not run:
data(tyner_interaction_binary)
data(tyner_sensitivity)
y<-tyner_sensitivity[,1]
k_selected<-sffs(tyner_interaction_binary, y)$k_sel
x<-data.frame(tyner_interaction_binary[, k_selected])
#binarize the sensitivity data
one<-which(y>0.5)
zero<-which(y<=0.5)
SENS<-y
SENS[one]<-1
SENS[zero]<-0
draw_data<-cbind(x, SENS)
drawGraph(draw_data)

## End(Not run)
```

---

drugRank *Generate the list of ranked drug combinations*

Description

A function to provide a list of drug combinations ranked by their synergy scores

Usage

```
drugRank(profile_select, predicted_matrix, sens)
```

Arguments

profile_select	the selected drug-target interaction data
predicted_matrix	the predicted efficacy matrix
sens	the drug sensitivity vector.

Value

a matrix contains the information about the list of drug combinations ranked by their synergy scores.

Author(s)

Jing Tang <jing.tang@helsinki.fi>

References

Tang J, Karhinen L, Xu T, Szwajda A, Yadav B, Wennerberg K, Aittokallio T. Target inhibition networks: predicting selective combinations of druggable targets to block cancer survival pathways. PLOS Computational Biology 2013; 9: e1003226.

Examples

```
## Not run:
data(tyner_interaction_binary)
data(tyner_sensitivity)
float<-sffsBinary(tyner_interaction_binary, tyner_sensitivity[, 1], max_k = 8)
k_select<-float$k_sel
x<-data.frame(tyner_interaction_binary)
kinase_names <- dimnames(x)[[2]]
select_kinase_names <- findSameSet(x, k_select, kinase_names)
gc_timma <- graycode3(length(k_select))
gc_names <- graycodeNames(length(k_select), select_kinase_names, gc_timma$gc_row, gc_timma$gc_col)
nr <- gc_names$nr
nc <- t(gc_names$nc)
timma_row <- nrow(nr) + nrow(nc)
timma_col <- ncol(nr) + ncol(nc)
timma <- array("", dim = c(timma_row, timma_col))
timma[(nrow(nc) + 1):timma_row, 1:ncol(nr)] <- nr
timma[1:nrow(nc), (ncol(nr) + 1):timma_col] <- nc
timma[(nrow(nc) + 1):timma_row, (ncol(nr) + 1):timma_col] <- float$timma$dummy
profile_select<-data.frame(tyner_interaction_binary)[, k_select]
drug_combo_rank<-drugRank(profile_select, timma, tyner_sensitivity[, 1])

## End(Not run)
```

`findSameCol` *Find the same column from a matrix*

Description

A function to seek for the same column from a matrix

Usage

```
findSameCol(X, Y)
```

Arguments

`X`	a matrix
`Y`	a vector with the same length as each column in X.

Value

a vector of the column indexes which are the same as vector Y.

Author(s)

Liye He <liye.he@helsinki.fi>

Examples

```
## Not run:
data(tyner_interaction_binary)
data(tyner_sensitivity)
x<-data.frame(tyner_interaction_binary)
kinase_names<-dimnames(tyner_interaction_binary)
float<-sffsBinary(tyner_interaction_binary, tyner_sensitivity[,1])
k_select <- float$k_sel
select_kinase_names <- findSameSet(x, k_select, kinase_names)

## End(Not run)
```

---

findSameSet *Find the same columns from two matrices*

Description

A function to find the same columns from two matrices

Usage

```
findSameSet(profile, selected_list, kinase_name)
```

Arguments

profile	the drug-target interaction data matrix
selected_list	the selected drug-target matrix
kinase_name	the names of the targets

Value

a vector of combined selected target names

Author(s)

Liye He <liye.he@helsinki.fi>

Examples

```
## Not run:
data(tyner_interaction_binary)
data(tyner_sensitivity)
x<-data.frame(tyner_interaction_binary)
kinase_names<-dimnames(tyner_interaction_binary)
float<-sffsBinary(tyner_interaction_binary, tyner_sensitivity[,1])
k_select <- float$k_sel
select_kinase_names <- findSameSet(x, k_select, kinase_names)

## End(Not run)
```

`floating2` *Filter targets*

Description

A function to filter targets based on their corration with the drug sensitivity

Usage

```
floating2(profile, sens, sp = 1, max_k = 2, verbosity = FALSE)
```

Arguments

`profile`	drug-target interaction data
`sens`	drug sensitivity data
`sp`	an integer to specify the starting point for the sffs search algorithm. The number cannot be larger than the total number of targets in the drug-target interaction data. By default, the starting point is the first target, namely, sp = 1.
`max_k`	an integer to specify the maximal number of targets that can be selected by the sffs algorithm. In practice it is advised to keep it under 10 as the number of sensitivities to be predicted will increase exponentially. By default, max_k = 2.
`verbosity`	a boolean value to decide if the information should be displayed. If it is TRUE, the information will be displayed while the model is running. Otherwise, the information will not be displayed. By default, it is FALSE.

Details

The major difference between original and modified averaging method is the averaging methods for the case where the minimization and maximization rules are not simultaneously satisfied. For example, for a queried target set there are supersets but not subsets in the training data, the original algorithm will take the prediction from these supersets data using the minimization rule. However, the modified algorithm will further adjust the prediction using the average between such a prediction and 0.

Value

A list containing the following components:

timma	a list contains: the predicted efficacy matrix, prediction error and predicted drug sensitivity
k_sel	the indexes for selected targets

Author(s)

Liye He <liye.he@helsinki.fi>

Examples

```
## Not run:
data(tyner_interaction_binary)
data(tyner_sensitivity)
result<-floating2(tyner_interaction_binary, tyner_sensitivity[,1], sp = 1, max_k = 5)

## End(Not run)
```

---

getBinary *Binary set for multiclass data*

Description

A function to get the supersets and subsets for multiclass data

Usage

```
getBinary(input, data)
```

Arguments

input	a vector of multiclass data
data	a matrix of multiclass data as training data

Value

a list of the following components:

superset	the supersets of the input data from the training data
subset	the subsets of the input data from the training data

Author(s)

Liye He <liye.he@helsinki.fi>

Examples

```
data(tyner_interaction_multiclass)
sets<-getBinary(tyner_interaction_multiclass[1,], tyner_interaction_multiclass)
```

`getBinary1` *Weighted binary set for multiclass data*

Description

A function to get the weighted supersets and subsets for multiclass data

Usage

```
getBinary1(input, data)
```

Arguments

`input`	a vector of multiclass data
`data`	a matrix of multiclass data as training data

Value

a list of the following components:

`superset`	the weighted supersets of the input data from the training data
`subset`	the weighted subsets of the input data from the training data

Author(s)

Liye He <liye.he@helsinki.fi>

Examples

```
data(tyner_interaction_multiclass)
sets<-getBinary1(tyner_interaction_multiclass[1,], tyner_interaction_multiclass)
```

graycode2 *Graycode Function*

Description

A function to generate decimal graycode

Usage

```
graycode2(a)
```

Arguments

a	the number of targets

Value

A list contains the following components:

rows	the number of rows
cols	the number of columns
dec	the decimal graycode results

Author(s)

Liye He <liye.he@helsinki.fi>

References

Dah jyh Guan. (Scientific Note) Generalized Gray Codes with Applications. 1998

Examples

```
code<-graycode2(5)
```

graycode3 *Gray code function for matrix indexes*

Description

A function to generate gray code used for matrix row and column names

Usage

```
graycode3(m)
```

Arguments

m	an integer to specify the number of bits

Value

a list of the following components:

gc_row	binary gray code as row names of the predicted sensitivity matrix
gc_col	binary gray code as column names of the predicted sensitivity matrix
dec_row	decimal gray code as row names of the predicted sensitivity matrix
dec_col	decimal gray code as column names of the predicted sensitivity matrix

Author(s)

Liye He <liye.he@helsinki.fi>

Examples

```
names<-graycode3(3)
```

graycodeNames *Names for the predicted sensitivity matrix*

Description

A function to make the target names in the format of gray code for the predected sensitivity matrix

Usage

```
graycodeNames(m, names, gc_row, gc_col)
```

Arguments

m	an integer to specify the number of targets
names	a vector of the names of the targets
gc_row	the gray code as row indexes. It can be returned by graycode3.
gc_col	the gray code as column indexes. It can be returned by graycode3.

Value

a list of the following components:

nr	the gray code format target names as row names.
nc	the gray code format target names as row names.

Author(s)

Liye He <liye.he@helsinki.fi>

Examples

```
## Not run:
data(tyner_interaction_binary)
data(tyner_sensitivity)
k_select<-sffsBinary(tyner_interaction_binary, tyner_sensitivity[, 1])$k_sel
gc_timma<-graycode3(length(k_select))
select_kinase_names<-dimnames(tyner_interaction_binary)[[2]][k_select]
gc_names<-graycodeNames(length(k_select), select_kinase_names, gc_timma$gc_row, gc_timma$gc_col)

## End(Not run)
```

---

grays *Generate gray code*

Description

A function to generate gray code

Usage

```
grays(n)
```

Arguments

n	an integer to specify the number of bits.

Value

a vector of the decimal gray code.

Author(s)

Liye He <liye.he@helsinki.fi>

References

Dah jyh Guan. (Scientific Note) Generalized Gray Codes with Applications. 1998

Examples

```
code<-grays(3)
```

---

kiba *Kiba interaction data*

Description

The curated drug-target interactions data including 52498 ChEMBL compounds and 467 targets from the Tang et al. (2014) study

Format

A data frame with column names as target ID and the first column as CHEMBL ID.

---

maxcpp *Search for the max values of 3D matrix in cpp*

Description

Search for the max values of 3D matrix in cpp

Author(s)

Liye He <liye.he@helsinki.fi>

---

maxcpp1 *Search for the max values of 2D matrix in cpp*

Description

A function to search for the max values of 2D matrix in cpp

Author(s)

Liye He <liye.he@helsinki.fi>

miller_drugs *A drug list from Miller study*

Description

The drug list from the Miller study

Format

A data frame with drug information from the Miller study

miller_drug_response *The single drug does-response data from the Miller study*

Description

The single drug does-response data from the Miller study.

Format

A data frame contains the drug response from Miller study

miller_interaction_binary *The binarized drug-target data for the Miller drugs*

Description

The binarized drug-target data for the Miller drugs

Format

A data frame contains drug names, target names, and binding affities

miller_sensitivity *The scaled drug sensitivity data for the Miller drugs*

Description

The scaled drug sensitivity data for the Miller drugs

Format

A matrix with drugs as row indexes and entries are drug sensitivities

miller_targets *The curated drug-target data for the Miller drugs*

Description

The curated drug-target data for the Miller drugs

Format

A data frame contains 234 targets information for 14 drugs

mincpp *Search for the min values of 3D matrix in cpp*

Description

A function to search for the min values of 3D matrix by one dimension in cpp

Author(s)

Liye He <liye.he@helsinki.fi>

mincpp1 *Search for the min values of 2D matrix in cpp*

Description

A function to search for the min values of 2D matrix in cpp

Author(s)

Liye He <liye.he@helsinki.fi>

normalizeSensitivity *Normalize the drug sensitivity data*

Description

A function to normalize the drug sensitivity data to [0,1]

Usage

```
normalizeSensitivity(IC50, method = "minMax")
```

Arguments

IC50	a vector contains the drug sensitivity in the form of IC50.
method	a string to specify the method used to normalize the sensitivity data. If it is "minMax", the sensitivity is scaled by (Max_IC50-IC50)/(Max_IC50-Min_IC50). If it is "logistic", it is scaled by 1/(1+exp(-1/IC50)). If it is "hyperbolic", it is scaled by tanh(1/IC50).

Value

A vector contains the normalized drug sensitivity data.

Author(s)

Jing Tang <jing.tang@helsinki.fi>

Examples

```
data(tyner_sensitivity)
normalizedSensitivity<-normalizeSensitivity(tyner_sensitivity[,1])
```

searchSpace *Generate search space*

Description

A function to generate the search space for sffs

Usage

```
searchSpace(drug_number, k_set, profile_data, y_actual)
```

Arguments

drug_number	an integer to specify the number of drugs
k_set	a vector to specify the selected target set
profile_data	drug-target interaction data
y_actual	the drug sensitivity data

Value

a list of the following components:

IM_d	search space of identical sets
IM_superset	search space of supersets
IM_subset	search space of subsets

Author(s)

Liye He <liye.he@helsinki.fi>

Examples

```
data(tyner_interaction_binary)
data(tyner_sensitivity)
num<-length(tyner_sensitivity[,1])
k_set<-rep(0, dim(tyner_interaction_binary)[2])
k_set[1]<-1
space<-searchSpace(num, k_set, tyner_interaction_binary, tyner_sensitivity[,1])
```

---

sffs *SFFS switch function*

Description

A function to choose which sffs function to run. There are six sffs algorithms for choosing.

Usage

```
sffs(profile_data, sens, sp = 1, max_k = 2, loo = TRUE, class = 2,
  averaging = "one.sided", weighted = FALSE, verbosity = FALSE)
```

Arguments

profile_data	drug-target interaction data which is a matrix with drugs as row indexes and targets as column indexes.
sens	a drug sensitivity vector.
sp	an integer to specify the starting point for the sffs search algorithm. The number cannot exceed the total number of targets in the drug-target interaction data. By default, the starting point is the first target, namely, sp = 1.
max_k	an integer to sepcify the maximum number of targets that can be selected by the sffs algorithm. By default, max_k = 2. In practice it should not be over than 10 as the number of target combinations will increase exponentially.
loo	a logical value indicating whether to use the leave-one-out cross-validation in the model selection process. By default, loo = TRUE.
class	an integer to specify the number of classes in the drug-target interaction data. For a binary drug-target interaction data, class = 2. For a multi-class drug-target interaction data, class should be the number of classes.
averaging	a parameter to specify which one of the averaging algorithms will be applied in the model construction. By default, averaging = "one.sided", which is the original model construction algorithm. When averaging = "two.sided", a modified averaging algorithm will be used. These two variants only differ for the case where the minimization and maximization rules are not simultaneously satisfied. For example, for a queried target set if the supersets but not the subsets can be found in the training data, the one.sided algorithm will take the prediction from the averages on the supersets sensitivities using the minimization rule. The two.sided algorithm, however, will lower the predicted sensitivity by averaging it with 0, which is the theoretical lower boundary of the sensitivities that could be obtained in the subsets.
weighted	a parameter to specify if the similarity between the queried target set and its subsets/supersets is considered as a weight factor in the averaging. When weighted =T RUE, the similarity is considered as a weight factor such that those related target sets will be weighted more in the final predictions.
verbosity	a boolean value to decide if the information should be displayed. If it is TRUE, the information will be displayed while the model is running. Otherwise, the information will not be displayed. By default, it is FALSE.

Value

A list containing the following components:

timma	a list contains: the predicted efficacy for target combinations, prediction error and predicted drug sensitivity
k_sel	the indexes for selected targets

Author(s)

Liye He <liye.he@helsinki.fi>

Examples

```
## Not run:
data(tyner_interaction_binary)
data(tyner_sensitivity)
results<-sffs(tyner_interaction_binary, tyner_sensitivity[, 1], max_k = 8)

## End(Not run)
```

`sffsBinary` *Model selection with sffs for the binary drug-target interaction data*

Description

A function to select the most predictive targets with sffs for the binary drug-target interaction data using orignal maximization and minimization rules

Usage

```
sffsBinary(profile_data, sens, sp = 1, max_k = 2, loo = TRUE,
  verbosity = FALSE)
```

Arguments

`profile_data`	drug-target interaction data which is a matrix with drugs as row indexes and targets as column indexes.
`sens`	a drug sensitivity vector.
`sp`	an integer to specify the starting point for sequential forward floating search (sffs) search algorithm to navigate the target set space. By default, sp = 1.
`max_k`	an integer to sepcify the maximum number of targets that can be selected by the sffs algorithm. By default, max_k = 2. In practice it should not be over than 10 as the number of target combinations will increase exponentially.
`loo`	a logical value indicating whether to use the leave-one-out cross-validation in the model selection process. By default, loo = True.
`verbosity`	a boolean value to decide if the information should be displayed. If it is TRUE, the information will be displayed while the model is running. Otherwise, the information will not be displayed. By default, it is FALSE.

Value

A list containing the following components:

`timma`	a list contains: the predicted efficacy matrix, prediction error and predicted drug sensitivity
`k_sel`	the indexes for selected targets

Author(s)

Jing Tang <jing.tang@helsinki.fi>

References

Tang J, Karhinen L, Xu T, Szwajda A, Yadav B, Wennerberg K, Aittokallio T. Target inhibition networks: predicting selective combinations of druggable targets to block cancer survival pathways. PLOS Computational Biology 2013; 9: e1003226.

Examples

```
## Not run:
data(tyner_interaction_binary)
data(tyner_sensitivity)
results<-sffsBinary(tyner_interaction_binary, tyner_sensitivity[, 1], max_k = 2)

## End(Not run)
```

---

sffsBinary1 *Model selection with sffs for the binary drug-target interaction data using two.sided TIMMA model*

Description

A function to select the most predictive targets with sffs for the binary drug-target interaction data using two.sided TIMMA model

Usage

```
sffsBinary1(profile_data, sens, sp = 1, max_k = 2, loo = TRUE,
  verbosity = FALSE)
```

Arguments

profile_data	drug-target interaction data which is a matrix with drugs as row indexes and targets as column indexes.
sens	a drug sensitivity vector.
sp	an integer to specify the starting point for sequential forward floating search (sffs) search algorithm to navigate the target set space. By default, sp = 1.
max_k	an integer to sepcify the maximum number of targets that can be selected by the sffs algorithm. By default, max_k = 2. In practice it should not be over than 10 as the number of target combinations will increase exponentially.
loo	a logical value indicating whether to use the leave-one-out cross-validation in the model selection process. By default, loo = TRUE.
verbosity	a boolean value to decide if the information should be displayed. If it is TRUE, the information will be displayed while the model is running. Otherwise, the information will not be displayed. By default, it is FALSE.

Details

The major difference between original and modified averaging method is the averaging methods for the case where the minimization and maximization rules are not simultaneously satisfied. For example, for a queried target set there are supersets but not subsets in the training data, the original algorithm will take the prediction from these supersets data using the minimization rule. However, the modified algorithm will further adjust the prediction using the average between such a prediction and 0.

Value

A list containing the following components:

`timma`	a list contains: the predicted efficacy matrix, prediction error and predicted drug sensitivity
`k_sel`	the indexes for selected targets

Author(s)

Liye He <liye.he@helsinki.fi>

References

Tang J, Karhinen L, Xu T, Szwajda A, Yadav B, Wennerberg K, Aittokallio T. Target inhibition networks: predicting selective combinations of druggable targets to block cancer survival pathways. PLOS Computational Biology 2013; 9: e1003226.

Examples

```
## Not run:
data(tyner_interaction_binary)
data(tyner_sensitivity)
results<-sffsBinary1(tyner_interaction_binary, tyner_sensitivity[, 1], max_k = 2)

## End(Not run)
```

---

sffsBinary2 *Model selection with filtered binary drug-target interaction data*

Description

A function to run sffs for model selection with filtered binary drug-target interaction data

Usage

```
sffsBinary2(profile_data, sens, sp = 1, max_k = 5, loo = TRUE,
  new_initial_list, verbosity = FALSE)
```

Arguments

profile_data	drug-target interaction data which is a matrix with drugs as row indexes and targets as column indexes.
sens	a drug sensitivity vector.
sp	an integer to specify the starting point for sequential forward floating search (sffs) search algorithm to navigate the target set space. By default, sp = 1.
max_k	an integer to sepcify the maximum number of targets that can be selected by the sffs algorithm. By default, max_k = 5. In practice it should not be over than 10 as the number of target combinations will increase exponentially.
loo	a logical value indicating whether to use the leave-one-out cross-validation in the model selection process. By default, loo = TRUE.
new_initial_list	a vector of the filtered targets indexes.
verbosity	a boolean value to decide if the information should be displayed. If it is TRUE, the information will be displayed while the model is running. Otherwise, the information will not be displayed. By default, it is FALSE.

Details

The major difference between original and modified averaging method is the averaging methods for the case where the minimization and maximization rules are not simultaneously satisfied. For example, for a queried target set there are supersets but not subsets in the training data, the original algorithm will take the prediction from these supersets data using the minimization rule. However, the modified algorithm will further adjust the prediction using the average between such a prediction and 0.

Value

A list containing the following components:

timma	a list contains: the predicted efficacy matrix, prediction error and predicted drug sensitivity
k_sel	the indexes for selected targets

Author(s)

Liye He <liye.he@helsinki.fi>

Examples

```
## Not run:
data(tyner_interaction_binary)
data(tyner_sensitivity)
profile<-tyner_interaction_binary[,c(-1, -2, -5)]
num<-length(tyner_sensitivity[,1])
k_set<-rep(0, dim(profile)[2])
k_set[1]<-1
result<-sffsBinary2(profile, tyner_sensitivity[,1], new_initial_list = k_set, max_k=2)
```

```
## End(Not run)
```

`sffsCategory` *Model selection with sffs for the multi-class drug-target interaction data using one.sided TIMMA model*

Description

A function to select the most predictive targets with sffs for the multi-class drug-target interaction data using the one.sided TIMMA model

Usage

```
sffsCategory(profile_data, sens, sp = 1, max_k = 2, loo = TRUE, class,
  verbosity = FALSE)
```

Arguments

`profile_data`	drug-target interaction data which is a matrix with drugs as row indexes and targets as column indexes.
`sens`	a drug sensitivity vector.
`sp`	an integer to specify the starting point for sequential forward floating search (sffs) search algorithm to navigate the target set space. By default, sp = 1.
`max_k`	an integer to sepcify the maximum number of targets that can be selected by the sffs algorithm. By default, max_k = 2. In practice it should not be over than 10 as the number of target combinations will increase exponentially.
`loo`	a logical value indicating whether to use the leave-one-out cross-validation in the model selection process. By default, loo = TRUE.
`class`	an integer to specify the number of classes in the drug-target interaction data. For a binary drug-target interaction data, class = 2. For a multi-class drug-target interaction data, class should be the number of classes.
`verbosity`	a boolean value to decide if the information should be displayed. If it is TRUE, the information will be displayed while the model is running. Otherwise, the information will not be displayed. By default, it is FALSE.

Value

A list containing the following components:

`timma`	a list contains: the predicted efficacy matrix, prediction error and predicted drug sensitivity
`k_sel`	the indexes for selected targets

Author(s)

Jing Tang <jing.tang@helsinki.fi>

Examples

```
## Not run:
data(tyner_interaction_multiclass)
data(tyner_sensitivity)
results<-sffsCategory(tyner_interaction_multiclass, tyner_sensitivity[, 1], max_k = 2, class = 6)

## End(Not run)
```

`sffsCategory1` *Model selection with sffs for the multi-class drug-target interaction data using two.sided TIMMA model*

Description

A function to select the most predictive targets with sffs for the multi-class drug-target interaction data using two.sided TIMMA model

Usage

```
sffsCategory1(profile_data, sens, sp = 1, max_k = 2, loo = TRUE, class,
  verbosity = FALSE)
```

Arguments

`profile_data`	drug-target interaction data which is a matrix with drugs as row indexes and targets as column indexes.
`sens`	a drug sensitivity vector.
`sp`	an integer to specify the starting point for sequential forward floating search (sffs) search algorithm to navigate the target set space. By default, sp = 1.
`max_k`	an integer to sepcify the maximum number of targets that can be selected by the sffs algorithm. By default, max_k = 2. In practice it should not be over than 10 as the number of target combinations will increase exponentially.
`loo`	a logical value indicating whether to use the leave-one-out cross-validation in the model selection process. By default, loo = TRUE.
`class`	an integer number to specify the number of classes in the drug-target interaction data. For a binary drug-target interaction data, class = 2. For a multi-class drug-target interaction data, class should be the number of classes.
`verbosity`	a boolean value to decide if the information should be displayed. If it is TRUE, the information will be displayed while the model is running. Otherwise, the information will not be displayed. By default, it is FALSE.

Details

The major difference between original and modified averaging method is the averaging methods for the case where the minimization and maximization rules are not simultaneously satisfied. For example, for a queried target set there are supersets but not subsets in the training data, the original algorithm will take the prediction from these supersets data using the minimization rule. However, the modified algorithm will further adjust the prediction using the average between such a prediction and 0.

Value

A list containing the following components:

`timma`	a list contains: the predicted efficacy matrix, prediction error and predicted drug sensitivity
`k_sel`	the indexes for selected targets

Author(s)

Jing Tang <jing.tang@helsinki.fi>

Examples

```
## Not run:
data(tyner_interaction_multiclass)
data(tyner_sensitivity)
results<-sffsCategory1(tyner_interaction_multiclass, tyner_sensitivity[, 1], max_k = 2, class = 6)

## End(Not run)
```

---

sffsCategoryWeighted *Model selection with sffs for the multi-class drug-target interaction data using one.sided and weighted TIMMA model*

Description

A function to select the most predictive targets with sffs for the multi-class drug-target interaction data using one.sided and weighted TIMMA model

Usage

```
sffsCategoryWeighted(profile_data, sens, sp = 1, max_k = 2, loo = TRUE,
  class, verbosity = FALSE)
```

Arguments

profile_data	drug-target interaction data which is a matrix with drugs as row indexes and targets as column indexes.
sens	a drug sensitivity vector.
sp	an integer to specify the starting point for sequential forward floating search (sffs) search algorithm to navigate the target set space. By default, sp = 1.
max_k	an integer to sepcify the maximum number of targets that can be selected by the sffs algorithm. By default, max_k = 2. In practice it should not be over than 10 as the number of target combinations will increase exponentially.
loo	a logical value indicating whether to use the leave-one-out cross-validation in the model selection process. By default, loo = TRUE.
class	an integer number to specify the number of classes in the drug-target interaction data. For a binary drug-target interaction data, class = 2. For a multi-class drug-target interaction data, class should be the number of classes.
verbosity	a boolean value to decide if the information should be displayed. If it is TRUE, the information will be displayed while the model is running. Otherwise, the information will not be displayed. By default, it is FALSE.

Value

A list containing the following components:

timma	a list contains: the predicted efficacy matrix, prediction error and predicted drug sensitivity
k_sel	the indexes for selected targets

Author(s)

Jing Tang <jing.tang@helsinki.fi>

Examples

```
## Not run:
data(tyner_interaction_multiclass)
data(tyner_sensitivity)
results<-sffsCategoryWeighted(tyner_interaction_multiclass, tyner_sensitivity[, 1], class = 6)

## End(Not run)
```

`sffsCategoryWeighted1` *Model selection with sffs for the multi-class drug-target interaction data using two.sided and weighted TIMMA model*

Description

A function to select the most predictive targets with sffs for the multi-class drug-target interaction data using two.sided and weighted TIMMA model

Usage

```
sffsCategoryWeighted1(profile_data, sens, sp = 1, max_k = 2, loo = TRUE,
  class, verbosity = FALSE)
```

Arguments

`profile_data`	drug-target interaction data which is a matrix with drugs as row indexes and targets as column indexes.
`sens`	a drug sensitivity vector.
`sp`	an integer to specify the starting point for sequential forward floating search (sffs) search algorithm to navigate the target set space. By default, sp = 1.
`max_k`	an integer to sepcify the maximum number of targets that can be selected by the sffs algorithm. By default, max_k = 2. In practice it should not be over than 10 as the number of target combinations will increase exponentially.
`loo`	a logical value indicating whether to use the leave-one-out cross-validation in the model selection process. By default, loo = TRUE.
`class`	an integer number to specify the number of classes in the drug-target interaction data. For a binary drug-target interaction data, class = 2. For a multi-class drug-target interaction data, class should be the number of classes.
`verbosity`	a boolean value to decide if the information should be displayed. If it is TRUE, the information will be displayed while the model is running. Otherwise, the information will not be displayed. By default, it is FALSE.

Details

The major difference between original and modified averaging method is the averaging methods for the case where the minimization and maximization rules are not simultaneously satisfied. For example, for a queried target set there are supersets but not subsets in the training data, the original algorithm will take the prediction from these supersets data using the minimization rule. However, the modified algorithm will further adjust the prediction using the average between such a prediction and 0.

Value

A list containing the following components:

timma	a list contains: the predicted efficacy matrix, prediction error and predicted drug sensitivity
k_sel	the indexes for selected targets

Author(s)

Jing Tang <jing.tang@helsinki.fi>

Examples

```
## Not run:
data(tyner_interaction_multiclass)
data(tyner_sensitivity)
results<-sffsCategoryWeighted1(tyner_interaction_multiclass, tyner_sensitivity[, 1], class = 6)

## End(Not run)
```

---

sumcpp *Sum for 3D matrix*

Description

A function to get the sum of 3D matrix by one dimension

Author(s)

Liye He <liye.he@helsinki.fi>

---

sumcpp1 *Sum for 2D matrix in cpp*

Description

A function to get the sum of 2D matrix in cpp

Author(s)

Liye He <liye.he@helsinki.fi>

`targetRank` *Generate the list of ranked target combinations*

Description

A function to provide a list of target combiantions ranked by their predicted synergy scores

Usage

```
targetRank(profile_select, predicted_matrix)
```

Arguments

`profile_select`	the drug-target interaction profile for the selected targets
`predicted_matrix`	the predicted efficacy matrix

Value

a matrix containing the list of target combinations

Author(s)

Jing Tang <jing.tang@helsinki.fi>

References

Tang J, Karhinen L, Xu T, Szwajda A, Yadav B, Wennerberg K, Aittokallio T. Target inhibition networks: predicting selective combinations of druggable targets to block cancer survival pathways. PLOS Computational Biology 2013; 9: e1003226.

Examples

```
## Not run:
data(tyner_interaction_binary)
data(tyner_sensitivity)
float<-sffsBinary(tyner_interaction_binary, tyner_sensitivity[, 1], max_k = 8)
k_select<-float$k_sel
x<-data.frame(tyner_interaction_binary)
kinase_names <- dimnames(x)[[2]]
select_kinase_names <- findSameSet(x, k_select, kinase_names)
gc_timma <- graycode3(length(k_select))
gc_names <- graycodeNames(length(k_select), select_kinase_names, gc_timma$gc_row, gc_timma$gc_col)
nr <- gc_names$nr
nc <- t(gc_names$nc)
timma_row <- nrow(nr) + nrow(nc)
timma_col <- ncol(nr) + ncol(nc)
timma <- array("", dim = c(timma_row, timma_col))
timma[(nrow(nc) + 1):timma_row, 1:ncol(nr)] <- nr
```

```
timma[1:nrow(nc), (ncol(nr) + 1):timma_col] <- nc
timma[(nrow(nc) + 1):timma_row, (ncol(nr) + 1):timma_col] <- float$timma$dummy
profile_select<-data.frame(tyner_interaction_binary)[, k_select]
target_combo_rank<-targetRank(profile_select, timma)

## End(Not run)
```

`timma` *Main function for the timma package*

Description

Target inhibition inference using maximization and minimization averaging

Usage

```
timma(x, y, sp = 1, max_k = 5, filtering = FALSE, class = 2,
  averaging = "one.sided", weighted = FALSE, verbosity = FALSE,
  use = "observed")
```

Arguments

`x`	a drug-target interaction matrix. Row names are drug names and column names are target names.
`y`	a normalized drug sensitivity vector.
`sp`	an integer to specify the starting point for the sffs search algorithm. The number cannot be larger than the total number of targets in the drug-target interaction data. By default, the starting point is the first target, namely, sp = 1.
`max_k`	an integer to specify the maximal number of targets that can be selected by the sffs algorithm. In practice it is advised to keep it under 10 as the number of sensitivities to be predicted will increase exponentially. By default, max_k = 5.
`filtering`	a logical parameter to determine whether the targets should be filtered before the model selection. By default, the value is FALSE, meaning that all the available targets will be considered in the model selection. If the value is TRUE, those targets that are negatively correlated with the drug sensitivity data will be removed.
`class`	an integer to specify the number of classes in the drug-target interaction data. For a binary drug-target interaction data, class = 2. For a multi-class drug-target interaction data, class should be the number of classes.
`averaging`	a parameter to specify which one of the averaging algorithms will be applied in the model construction. By default, averaging = "one.sided", which is the original model construction algorithm. When averaging = "two.sided", a modified averaging algorithm will be used. These two variants only differ for the case where the minimization and maximization rules are not simultaneously satisfied. For example, for a queried target set if the supersets but not the subsets

can be found in the training data, the one.sided algorithm will take the prediction from the averages on the supersets sensitivities using the minimization rule. The two.sided algorithm, however, will lower the predicted sensitivity by averaging it with 0, which is the theoretical lower boundary of the sensitivities that could be obtained in the subsets.

`weighted` When averaging = "weighted", the similarity between the queried target set and its subsets/supersets is considered as a weight factor in the averaging, such that those related target sets will be more weighted in the final predictions.

`verbosity` a boolean value to decide if the information should be displayed. If it is TRUE, the information will be displayed while the model is running. Otherwise, the information will not be displayed. By default, it is FALSE.

`use` When use = "observed", the true drug sensitivity data will be used for drawing target inhibition network. When use = "predicted", the predicted drug sensitivity data will be used for drawing target inhibition network.

Value

an R image of the input and output data.

Author(s)

Jing Tang <jing.tang@helsinki.fi>

References

Tang J, Karhinen L, Xu T, Szwajda A, Yadav B, Wennerberg K, Aittokallio T. Target inhibition networks: predicting selective combinations of druggable targets to block cancer survival pathways. PLOS Computational Biology 2013; 9: e1003226.

Examples

```
## Not run:
data(tyner_interaction_binary)
data(tyner_sensitivity)
median_sensitivity<-tyner_sensitivity[, 1]
results<-timma(tyner_interaction_binary, median_sensitivity)

## End(Not run)
```

---

timmaBinary *Predicting drug sensitivity with binary drug-target interaction data*

Description

A function to predict the drug sensitivity with binary drug-target interaction data using the original maximization and minimization rules

Usage

```
timmaBinary(drug_target_profile, sens, loo = TRUE)
```

Arguments

drug_target_profile	the drug-target interaction data. See timma.
sens	a drug sensitivity vector.
loo	a logical value indicating whether to use the leave-one-out cross-validation in the model selection process. By default, loo = TRUE.

Value

A list containing the following components:

dummy	the predicted efficacy for target combinations that can be found from the training data
error	the prediction errors
prediction	predicted drug sensitivity

Author(s)

Liye He <liye.he@helsinki.fi>

References

Tang J, Karhinen L, Xu T, Szwajda A, Yadav B, Wennerberg K, Aittokallio T. Target inhibition networks: predicting selective combinations of druggable targets to block cancer survival pathways. PLOS Computational Biology 2013; 9: e1003226.

Examples

```
data(tyner_interaction_binary)
data(tyner_sensitivity)
results<-timmaBinary(tyner_interaction_binary[, 1:6], tyner_sensitivity[,1])
```

---

timmaBinary1 *Predicting drug sensitivity with binary drug-target interaction data using modified maximization and minimization rules*

Description

A function to predict the drug sensitivity with binary drug-target interaction data using the modified maximization and minimization rules

Usage

```
timmaBinary1(drug_target_profile, sens, loo = TRUE)
```

Arguments

drug_target_profile	the drug-target interaction data. See timma.
sens	a drug sensitivity vector.
loo	a logical value indicating whether to use the leave-one-out cross-validation in the model selection process. By default, loo = TRUE.

Value

A list containing the following components:

dummy	the predicted efficacy for target combinations
error	the prediction errors
prediction	predicted drug sensitivity

Author(s)

Liye He <liye.he@helsinki.fi>

Examples

```
data(tyner_interaction_binary)
data(tyner_sensitivity)
results<-timmaBinary1(tyner_interaction_binary[, 1:6], tyner_sensitivity[,1])
```

---

timmaCategory *Predicting drug sensitivity with multi-class drug-target interaction data using one.sided TIMMA model*

Description

A function to predict the drug sensitivity with multi-class drug-target interaction data using the one.sided TIMMA model

Usage

```
timmaCategory(drug_target_profile, sens, loo = TRUE, class)
```

Arguments

drug_target_profile	the drug-target interaction data. See timma.
sens	a drug sensitivity vector.
loo	a logical value indicating whether to use the leave-one-out cross-validation in the model selection process. By default, loo = TRUE.
class	the number of classes in the drug-target interaction data

Value

A list containing the following components:

dummy	the predicted efficacy for target combinations that can be found from the training data
error	the prediction errors
prediction	predicted drug sensitivity

Author(s)

Liye He <liye.he@helsinki.fi>

Examples

```
data(tyner_interaction_multiclass)
data(tyner_sensitivity)
results<-timmaCategory(tyner_interaction_multiclass[, 1:6], tyner_sensitivity[,1], class = 6)
```

---

timmaCategory1 *Predicting drug sensitivity with multi-class drug-target interaction data using two.sided TIMMA model*

Description

A function to predict the drug sensitivity with multi-class drug-target interaction data using the two.sided TIMMA model

Usage

```
timmaCategory1(drug_target_profile, sens, loo = TRUE, class)
```

Arguments

drug_target_profile	the drug-target interaction data. See timma.
sens	a drug sensitivity vector.
loo	a logical value indicating whether to use the leave-one-out cross-validation in the model selection process. By default, loo = TRUE.
class	the number of classes in the drug-target interaction data

Value

A list containing the following components:

dummy	the predicted efficacy for target combinations that can be found from the training data
error	the prediction errors
prediction	predicted drug sensitivity

Author(s)

Liye He <liye.he@helsinki.fi>

Examples

```
data(tyner_interaction_multiclass)
data(tyner_sensitivity)
results<-timmaCategory1(tyner_interaction_multiclass[, 1:6], tyner_sensitivity[,1], class = 6)
```

---

timmaCategoryWeighted *Predicting drug sensitivity with multi-class drug-target interaction data using one.sided and weighted TIMMA model*

Description

A function to predict the drug sensitivity with multi-class drug-target interaction data using the one.sided and weighted TIMMA model

Usage

```
timmaCategoryWeighted(drug_target_profile, sens, loo = TRUE, class)
```

Arguments

drug_target_profile	the drug-target interaction data. See timma.
sens	a drug sensitivity vector.
loo	a logical value indicating whether to use the leave-one-out cross-validation in the model selection process. By default, loo = TRUE.
class	the number of classes in the drug-target interaction data

Value

A list containing the following components:

dummy	the predicted efficacy for target combinations that can be found from the training data
error	the prediction errors
prediction	predicted drug sensitivity

Author(s)

Liye He <liye.he@helsinki.fi>

Examples

```
## Not run:
profile<-data(tyner_interaction_multiclass)
sensitivity<-data(tyner_sensitivity)
results<-timmaCategoryWeighted(profile[, 1:6], sensitivity[,1], class = 6)

## End(Not run)
```

`timmaCategoryWeighted1` *Predicting drug sensitivity with multi-class drug-target interaction data using two.sided and weighted TIMMA model*

Description

A function to predict the drug sensitivity with multi-class drug-target interaction data using the two.sided and weighted TIMMA model

Usage

```
timmaCategoryWeighted1(profile_data, sens, loo = TRUE, class)
```

Arguments

`profile_data`	the drug-target interaction data. See `timma`.
`sens`	a drug sensitivity vector.
`loo`	a logical value indicating whether to use the leave-one-out cross-validation in the model selection process. By default, loo = TRUE.
`class`	the number of classes in the drug-target interaction data

Value

A list containing the following components:

`dummy`	the predicted efficacy for target combinations that can be found from the training data
`error`	the prediction errors
`prediction`	predicted drug sensitivity

Author(s)

Liye He <liye.he@helsinki.fi>

Examples

```
## Not run:
profile<-data(tyner_interaction_multiclass)
sensitivity<-data(tyner_sensitivity)
results<-timmaCategoryWeighted1(profile[, 1:6], sensitivity[,1], class = 6)

## End(Not run)
```

`timmaModel` *Predicting drug sensitivity with binary drug-target interaction data*

Description

A function to predict the drug sensitivity with binary drug-target interaction data using the one.sided TIMMA model

Usage

```
timmaModel(drug_target_profile, sens, loo = TRUE)
```

Arguments

`drug_target_profile`	the drug-target interaction data. See `timma`.
`sens`	a drug sensitivity vector.
`loo`	a logical value indicating whether to use the leave-one-out cross-validation in the model selection process. By default, loo = TRUE.

Value

A list containing the following components:

`dummy`	the predicted efficacy matrix
`error`	the prediction errors
`prediction`	predicted drug sensitivity

The difference between `timmaModel` and `timmaBinary` is `timmaModel` returns the predicted efficacy matrix of all possible target combinations while `timmaBinary` not.

Author(s)

Liye He <liye.he@helsinki.fi>

References

Tang J, Karhinen L, Xu T, Szwajda A, Yadav B, Wennerberg K, Aittokallio T. Target inhibition networks: predicting selective combinations of druggable targets to block cancer survival pathways. PLOS Computational Biology 2013; 9: e1003226.

Examples

```
data(tyner_interaction_binary)
data(tyner_sensitivity)
results<-timmaModel(tyner_interaction_binary[, 1:6], tyner_sensitivity[,1])
```

`timmaModel1` *Predicting drug sensitivity with binary drug-target interaction data using two.sided TIMMA model*

Description

A function to predict the drug sensitivity with binary drug-target interaction data using the two.sided TIMMA model

Usage

```
timmaModel1(drug_target_profile, y_actual, loo = TRUE)
```

Arguments

`drug_target_profile`	the drug-target interaction data. See `timma`.
`y_actual`	a drug sensitivity vector.
`loo`	a logical value indicating whether to use the leave-one-out cross-validation in the model selection process. By default, loo = TRUE.

Value

A list containing the following components:

`dummy`	the predicted efficacy matrix
`error`	the prediction errors
`prediction`	predicted drug sensitivity

The difference between `timmaModel` and `timmaBinary` is `timmaModel` returns the predicted efficacy matrix of all possible target combinations while `timmaBinary` not.

Author(s)

Liye He <liye.he@helsinki.fi>

Examples

```
data(tyner_interaction_binary)
data(tyner_sensitivity)
results<-timmaModel1(tyner_interaction_binary[, 1:6], tyner_sensitivity[,1])
```

`timmaSearchBinary` *Prediction in the search space with one.sided TIMMA model*

Description

A function to return the prediction error in the search space for sffs

Usage

```
timmaSearchBinary(profile_k, space, sens, loo = TRUE)
```

Arguments

`profile_k`	current selected drug-target interaction data
`space`	the search space returned by `searchSpace` function
`sens`	drug sensitivity data
`loo`	a logical value indicating whether to use the leave-one-out cross-validation in the model selection process. By default, loo = TRUE.

Value

the prediction error

Author(s)

Liye He <liye.he@helsinki.fi>

Examples

```
data(tyner_interaction_binary)
data(tyner_sensitivity)
num<-length(tyner_sensitivity[,1])
k_set<-rep(0, dim(tyner_interaction_binary)[2])
k_set[c(1,2,3)]<-1
space<-searchSpace(num, k_set, tyner_interaction_binary, tyner_sensitivity[,1])
profile_k<-tyner_interaction_binary[, which(k_set==1)]
error<-timmaSearchBinary(profile_k, space, tyner_sensitivity[,1])
```

`timmaSearchBinary1` *Prediction in the search space with two.sided TIMMA model*

Description

A function to return the prediction error in the search space for sffs

Usage

```
timmaSearchBinary1(profile_k, space, sens, loo = TRUE)
```

Arguments

`profile_k`	current selected drug-target interaction data
`space`	the search space returned by `searchSpace` function
`sens`	drug sensitivity data
`loo`	a logical value indicating whether to use the leave-one-out cross-validation in the model selection process. By default, loo = TRUE.

Value

the prediction error

Author(s)

Liye He <liye.he@helsinki.fi>

Examples

```
data(tyner_interaction_binary)
data(tyner_sensitivity)
num<-length(tyner_sensitivity[,1])
k_set<-rep(0, dim(tyner_interaction_binary)[2])
k_set[1]<-1
space<-searchSpace(num, k_set, tyner_interaction_binary, tyner_sensitivity[,1])
profile_k<-tyner_interaction_binary[, which(k_set==1)]
error<-timmaSearchBinary1(profile_k, space, tyner_sensitivity[,1])
```

`tyner_interaction_binary`

A binary drug-target interaction data

Description

A dataset containing 65 drugs and 322 targets interaction data. The binding affinity is binary values. 0 indicates no interaction while 1 indicates true interaction.

Format

A matrix with drugs as row indexes and targets as column indexes

Source

The orinial multi-class data can be found: http://cancerres.aacrjournals.org/content/73/1/285/suppl/DC1

`tyner_interaction_multiclass`

A multi-class drug-target interaction data

Description

A dataset containing 65 drugs and 322 targets interaction data. The binding affinity is categorical values. A higher value indicates a stronger interaction.

Format

A matrix with drug names as row names and target names as column names

Source

The orinial multi-class data can be found: http://cancerres.aacrjournals.org/content/73/1/285/suppl/DC1

tyner_sensitivity *The drug sensitivity data*

Description

A dataset containing the normalized 151 patient drug sensitivity data.

Format

A matrix contains the normalized 151 patient drug sensitivity data

Source

The orinial 151 patient drug sensitivity data can be found: http://cancerres.aacrjournals.org/content/73/1/285/suppl/DC1

Index