Package ‘newFocus’

July 22, 2025

Type Package

Title True Discovery Guarantee by Combining Partial Closed Testings

Version 1.1

Date 2021-06-22

Author Ningning Xu

Maintainer Ningning Xu <`n.xu@lumc.nl`>

Description Closed testing has been proved powerful for true discovery guarantee. The computation of closed testing is, however, quite burdensome. A general way to reduce computational complexity is to combine partial closed testings for some prespecified feature sets of interest. Partial closed testings are performed at Bonferroni-corrected alpha level to guarantee the lower bounds for the number of true discoveries in prespecified sets are simultaneously valid. For any post hoc chosen sets of interest, coherence property is used to get the lower bound. In this package, we implement closed testing with globaltest to calculate the lower bound for number of true discoveries, see Ningning Xu et.al (2021) <doi:10.48550/arXiv.2001.01541> for detailed description.

License GPL (>= 2)

Depends ctgt

NeedsCompilation no

Repository CRAN

Date/Publication 2021-07-05 15:50:06 UTC

Contents

`newFocus-package` *True Discovery Guarantee by Combining Partial Closed Testings*

Description

Closed testing has been proved powerful for true discovery guarantee. The computation of closed testing is, however, quite burdensome. A general way to reduce computational complexity is to combine partial closed testings for some prespecified feature sets of interest. Partial closed testings are performed at Bonferroni-corrected alpha level to guarantee the lower bounds for the number of true discoveries in prespecified sets are simultaneously valid. For any post hoc chosen sets of interest, coherence property is used to get the lower bound. In this package, we implement closed testing with globaltest to calculate the lower bound for number of true discoveries, see Ningning Xu et.al (2021) <arXiv:2001.01541> for detailed description.

Details

The DESCRIPTION file:

Package:	newFocus
Type:	Package
Title:	True Discovery Guarantee by Combining Partial Closed Testings
Version:	1.1
Date:	2021-06-22
Author:	Ningning Xu
Maintainer:	Ningning Xu <n.xu@lumc.nl>
Description:	Closed testing has been proved powerful for true discovery guarantee. The computation of closed testin
License:	GPL (>= 2)
Depends:	ctgt
NeedsCompilation:	no
Packaged:	2021-06-22 09:50:32 UTC; nxu
Repository:	CRAN

Index of help topics:

`choosepath`	`A set of focus set index`
`ctbab`	`Closed testing with branch and bound`
`discov`	`True discoveries`
`newFocus`	`The new focus level procedure`
`newFocus-package`	`True Discovery Guarantee by Combining Partial Closed Testings`
`pick`	`True discoveries for non-focus level node`

For the GO (Gene Ontology) terms chosen as focus level nodes, newFocus function will return the minimum number of true discoveries. For GO terms that are non-focus level nodes, we use pick to count the number of true discoveries based on the result of newFocus.

Author(s)

Ningning Xu

Maintainer: Ningning Xu <n.xu@lumc.nl>

References

Ningning Xu, Aldo solari, Jelle Goeman, Clsoed testing with global test, with applications on metabolomics data, arXiv:2001.01541, https://arxiv.org/abs/2001.01541

Jelle J. Goeman, Sara A. van de Geer, Floor de Kort, Hans C. van Houwelingen, A global test for groups of genes: testing association with a clinical outcome, Bioinformatics, Volume 20, Issue 1, 1 January 2004, Pages 93-99, https://doi.org/10.1093/bioinformatics/btg382

`choosepath` *A set of focus set index*

Description

The function aims to find out the focus set index for which the true discoveries is the most and all other focus sets that are disjoint with it .

Usage

```
choosepath(startingindex = 1, fsets, lowdv)
```

Arguments

`startingindex`	The index of focus set that has the first largest number of true discovereis
`fsets`	A list of focus level gene sets,or GO (Gene Ontology) terms
`lowdv`	A non-negative integer vector, which are the number of true discovereis, the length of the vector is the same as the list of focus level sets

Value

The function will return an integer or a numeric vector.

Author(s)

Ningning Xu

`ctbab` *Closed testing with branch and bound*

Description

Closed testing with branch and bound algorithm specifically for globaltest

Usage

```
ctbab(y, Cm, Tm, upnode, level, lownode, tmin, ctrue, lf, ls, alpha, count = 0, maxIt = 0)
```

Arguments

`y`	The response variable
`Cm`	The matrix for calculating critical values of globaltest
`Tm`	The matrix for calculating test statistics of globaltest
`upnode`	The upper node that is used to bound critical values
`level`	The level that the GO term of interest
`lownode`	The lower node that is used to bound critical values
`tmin`	The minimum test statistic
`ctrue`	The true critical value corresponding to the minimum test statistic
`lf`	The lambda vector corresponding to the upper node
`ls`	The lambda vector corresponding to the lower node
`alpha`	The significance level
`count`	An integer stores the repetitions of the branch and bound, i.e. how many time branch and bound is implemented
`maxIt`	The maximal number of repetitions prespecified by user

Value

It will retrun the rejection indicator by closed testing with branch and bound algorithm.

Author(s)

Ningning Xu

References

Xu, N., & Goeman, J. (2020). Closed testing with Globaltest with applications on metabolomics data. arXiv preprint arXiv:2001.01541.

`discov` *True discoveries*

Description

True discoveries calculated by the partial closed testing

Usage

```
discov(response, alternative, null, data, maxit = 0, alpha)
```

Arguments

`response`	The response variable
`alternative`	The alternative hypothesis, which is a character vector, i.e. a set of genes
`null`	The null hypothesis
`data`	A data frame with response and all covariates included
`maxit`	The maximal number of repetitions prespecified by user
`alpha`	The significance level

Value

It will return a non-negative integer: the lower bound for the number of true discovereis of the alternative gene set.

Author(s)

Ningning Xu

`newFocus` *The new focus level procedure*

Description

The new focus level procedure for calculating true discoveries for focus level nodes

Usage

```
newFocus(response, fsets, null, data, maxit = 0, alpha = 0.05, adj = 0)
```

Arguments

`response`	The response variable
`fsets`	A list of focus level sets
`null`	The null hypothesis
`data`	The data frame with response and all covariates included
`maxit`	The maximal number of repetitions prespecified by user
`alpha`	The significance level
`adj`	The number of focus sets that are fully rejected by partial closed testing, which is used to adjust the number of focus sets, The dafault value is 0.

Value

The function will return a focus subject with the lower bound for each focus level node.

Author(s)

Ningning Xu

References

Goeman, J. J., & Mansmann, U. (2008). Multiple testing on the directed acyclic graph of gene ontology. Bioinformatics, 24(4), 537-544.

Examples

```
## example data set
n= 100
m = 5
X = matrix(0, n, m,byrow = TRUE )
for ( i in 1:n){
  set.seed(1234+i)
  X[i,] =  as.vector(arima.sim(model = list(order = c(1, 0, 0), ar = 0.2), n = m) )
}
y = rbinom(n,1,0.6)
X[which(y==1),1:3] = X[which(y==1),1:3] + 0.8
xs = paste("x",seq(1,m,1),sep="")
colnames(X) = xs

mydata = as.data.frame(cbind(X,y))

## focus level sets
fl = list(c("x1", "x2"), c("x3", "x4"), "x5")
names(fl) = c("12", "34", "5")

## get td for focus level sets
focus_subject = newFocus(response = y, fsets = fl, data = mydata)

## get td for any set of interest given the focus subject
setofinterest = c("x1", "x2","x3", "x4")
```

```
pick(focus_subject, setofinterest)
```

`pick` *True discoveries for non-focus level node*

Description

The number of true discoveries for the non-focus level GO terms is calculated given the focus subject.

Usage

```
pick(focus_obj, setofinterest)
```

Arguments

`focus_obj`	The focus subject from function newFocus
`setofinterest`	A gene set or GO term of interest

Value

It will return an integer: the lower bound for the number of true discoveries in the set of interest

Author(s)

Ningning Xu

Examples

```
## example data set
n= 100
m = 5
X = matrix(0, n, m,byrow = TRUE )
for ( i in 1:n){
  set.seed(1234+i)
  X[i,] =  as.vector(arima.sim(model = list(order = c(1, 0, 0), ar = 0.2), n = m) )
}
y = rbinom(n,1,0.6)
X[which(y==1),1:3] = X[which(y==1),1:3] + 0.8
xs = paste("x",seq(1,m,1),sep="")
colnames(X) = xs

mydata = as.data.frame(cbind(X,y))

## focus level sets
fl = list(c("x1", "x2"), c("x3", "x4"), "x5")
names(fl) = c("12", "34", "5")
```

```
## get td for focus level sets
focus_subject = newFocus(response = y, fsets = fl, data = mydata)

## get td for any set of interest given the focus subject
setofinterest = c("x1", "x2","x3", "x4")
pick(focus_subject, setofinterest)
```

Index