Package 'fdrci'

November 15, 2016

Type Package

Title Permutation-Based FDR Point and Confidence Interval Estimation

Version 2.1

Date 2016-11-14

Author Joshua Millstein

Maintainer Joshua Millstein <joshua.millstein@usc.edu>

Description FDR functions for permutation-based estimators, including pi0 as well as FDR confidence intervals. The confidence intervals account for dependencies between tests by the incorporation of an overdispersion parameter, which is estimated from the permuted data.

License Artistic-2.0

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2016-11-15 19:45:37

R topics documented:

`fdrci-package` *Permutation-Based FDR Point and Confidence Interval Estimation*

Description

FDR functions for permutation-based estimators, including pi0 as well as FDR confidence intervals. The confidence intervals account for dependencies between tests by the incorporation of an overdispersion parameter, which is estimated from the permuted data.

Details

Package:	fdrci
Type:	Package
Version:	2.1
Date:	2016-11-14
License:	Artistic-2.0
LazyLoad:	yes

This method is designed to compute FDR when a permutation-based approach has been utilized. The objective here is to identify a subset of positive tests that have corresponding statistics with a more exteme distribution than the permuted results, which are assumed to represent the null. The significance of the subset is described in terms of the FDR and uncertainty in the FDR estimate by computing a confidence interval. Say a set of p-values(or simply a set of test statistics) were recorded for a set of hypothesis tests, and data were permuted B times with test results generated for each permutation. The function fdr_od() can be used to estimate FDR and and a confidence interval along with pi0, the proportion of true null hypotheses, given a selected significance threshold. The function fdrTbl()uses fdr_od() to create a table of results over a sequence of possible significance thresholds. Finally, the function FDRplot will plot results from fdrTbl(), facilitating the selection of a final significance threshold.

Author(s)

Joshua Millstein

Maintainer: Joshua Millstein <millsteinjoshua@gmail.com> Joshua Millstein

References

Millstein J, Volfson D. 2013. Computationally efficient permutation-based confidence interval estimation for tail-area FDR. Frontiers in Genetics | Statistical Genetics and Methodology 4(179):1-11.

FDRplot *Plot FDR Table Results*

Description

This function plots FDR point and CI estimates over a sequence of possible significance thresholds. Results from fdrTbl() can be plotted directly as input to FDRplot.

Usage

```
FDRplot(plotdat, lowerbound, upperbound, mn, lpos = "bottomleft", outfile = FALSE)
```

Arguments

plotdat	a table that is returned from fdrTbl(), or results formated in the same way.
lowerbound	-log10(p-value) lower bound for the x-axis of the plot.
upperbound	-log10(p-value) upper bound for the x-axis of the plot.
mn	text for main title
lpos	legend position, one of, "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right" and "center".
outfile	Either 'FALSE' or the path/name of a file. If FALSE then the plot is displayed rather than saved to a file.

Author(s)

Joshua Millstein

References

Millstein J, Volfson D. 2013. Computationally efficient permutation-based confidence interval estimation for tail-area FDR. Frontiers in Genetics | Statistical Genetics and Methodology 4(179):1-11.

Examples

```
nrow_=100
ncol_=100
X = as.data.frame(matrix(rnorm(nrow_*ncol_),nrow=nrow_,ncol=ncol_))
Y = as.data.frame(matrix(rnorm(nrow_*ncol_),nrow=nrow_,ncol=ncol_))
nperm = 10

myanalysis = function(X,Y){
ntests = ncol(X)
rslts = as.data.frame(matrix(NA,nrow=ntests,ncol=2))
names(rslts) = c("ID","pvalue")
rslts[,"ID"] = 1:ntests
for(i in 1:ntests){
fit = cor.test(X[,i],Y[,i],na.action="na.exclude",
```

```
alternative="two.sided",method="pearson")
rslts[i,"pvalue"] = fit$p.value
}
return(rslts)
} # End myanalysis

## Generate observed results
obs = myanalysis(X,Y)

## Generate permuted results
perml = vector('list',nperm)
for(p_ in 1:nperm){
X1 = X[order(runif(ncol_)),]
perml[[p_]] = myanalysis(X1,Y)
}

## FDR results table
myfdrtbl = fdrTbl(obs$pvalue,perml,"pvalue",ncol_,0,3)
## Plot results
FDRplot(myfdrtbl,0,3,"My FDR Plot",lpos = "bottomleft")
```

`fdrTbl` *FDR Estimate and Confidence Interval Sequence Table*

Description

Computes FDR estimates and confidence intervals for a sequence of potential significance thresholds.

Usage

```
fdrTbl(obs_vec, perm_list, pname, ntests, lowerbound, upperbound, incr=.1,cl=.95,c1=NA)
```

Arguments

`obs_vec`	observed vector of p-values.
`perm_list`	list of dataframes that include a column of permutation p-values (or statistics) in each. The length of the list permp = number of permutations.
`pname`	name of column in each list component dataframe that includes p-values (or statistics).
`ntests`	total number of observed tests, which is usually the same as the length of obs_vec and the number of rows in each perm_list dataframe. However, this may not be the case if results were filtered by a p-value threshold or statistic threshold. If filtering was conducted then lowerbound must be greater (more extreme) than the filtering criterion.
`lowerbound`	lowerbound refers to the range of -log10(p-value) over which fdr is computed for a sequence of thresholds

upperbound	upperbound refers to the range of -log10(p-value) over which fdr is computed for a sequence of thresholds
incr	value by which to increment the sequence from lowerbound to upperbound on a -log10(p-value) scale. Default is 0.1.
cl	confidence level (default is .95).
c1	overdispersion parameter. If this parameter is not specified (default initial value is NA), then the parameter is estimated from the data. If all tests are known to be independent, then this parameter should be set to 1.

Details

fdrTbl calls fdr_od. Output from fdrTbl() can be used for FDRplot() input.

Value

A dataframe is returned where rows correspond to p-value thresholds in the sequence from lowerbound to upperbound and columns are: c("threshold","fdr","ll","ul","pi0","odp","S","Sp")

threshold	p-value threshold chosen to define positive tests
fdr	estimated FDR at the chosen p-value threshold
ll	estimated lower 95% confidence bound for the FDR estimate
ul	estimated upper 95% confidence bound for the FDR estimate
pi0	estimated percent of true null hypotheses
odp	estimated over-dispersion parameter
S	observed number of positive tests
Sp	total number of positive tests summed across all permuted result sets

Author(s)

Joshua Millstein

References

Millstein J, Volfson D. 2013. Computationally efficient permutation-based confidence interval estimation for tail-area FDR. Frontiers in Genetics | Statistical Genetics and Methodology 4(179):1-11.

Examples

```
nrow_=100
ncol_=100
X = as.data.frame(matrix(rnorm(nrow_*ncol_),nrow=nrow_,ncol=ncol_))
Y = as.data.frame(matrix(rnorm(nrow_*ncol_),nrow=nrow_,ncol=ncol_))
nperm = 10

myanalysis = function(X,Y){
ntests = ncol(X)
rslts = as.data.frame(matrix(NA,nrow=ntests,ncol=2))
```

```
names(rslts) = c("ID","pvalue")
rslts[,"ID"] = 1:ntests
for(i in 1:ntests){
fit = cor.test(X[,i],Y[,i],na.action="na.exclude",
alternative="two.sided",method="pearson")
rslts[i,"pvalue"] = fit$p.value
}
return(rslts)
} # End myanalysis

## Generate observed results
obs = myanalysis(X,Y)

## Generate permuted results
perml = vector('list',nperm)
for(p_ in 1:nperm){
X1 = X[order(runif(ncol_)),]
perml[[p_]] = myanalysis(X1,Y)
}

## FDR results table
fdrTbl(obs$pvalue,perml,"pvalue",ncol_,1,2)
```

`fdr_od` *Permutation-Based FDR and Confidence Interval*

Description

This function can be used to estimate FDR, corresponding confidence interval, and pi0, the proportion of true null hypotheses, given a selected significance threshold, and results from permuted data.

Usage

```
fdr_od(obsp, permp, pnm, ntests, thres, cl=.95, c1=NA)
```

Arguments

`obsp`	observed vector of p-values.
`permp`	list of dataframes that include a column of permutation p-values (or statistics) in each. The length of the list permp = number of permutations.
`pnm`	name of column in each list component dataframe that includes p-values (or statistics).
`ntests`	total number of observed tests, which is usually the same as the length of obsp and the number of rows in each permp dataframe. However, this may not be the case if results were filtered by a p-value threshold or statistic threshold. If filtering was conducted then thres must be smaller (more extreme) than the filtering criterion.

thres	significance threshold.
cl	confidence level (default is .95).
c1	overdispersion parameter. If this parameter is not specified (default initial value is NA), then the parameter is estimated from the data. If all tests are known to be independent, then this parameter should be set to 1.

Details

If a very large number of tests are conducted, it may be useful to filter results, that is, save only results of those tests that meet some relaxed nominal significance threshold. This alleviates the need to record results for tests that are clearly non-significant. Results from fdr_od() are valid as long as thres < the relaxed nomimal significance threshold for both observed and permuted results. It is not necessary for the input to fdr_od() to be p-values, however, fdr_od() is designed for statistics in which smaller values are more extreme than larger values as is the case for p-values. Therefore, if raw statistics are used, then a transformation may be necessary to insure that smaller values are more likely associated with false null hypotheses than larger values. In certain situations, for instance when a large proportion of tests meet the significance threshold, pi0 is estimated to be very small, and thus has a large influence on the FDR estimate. To limit this influence, pi0 is constrained to be .5 or greater, resulting in a more conservative estimate under these conditions.

Value

A list which includes:

FDR	FDR point estimate
ll	lower confidence limit
ul	upper confidence limit
pi0	proportion of true null hypotheses
c1	overdispersion parameter
ro	observed number of positive tests
vp1	total number of positive tests summed across all permuted result sets

Author(s)

Joshua Millstein

References

Millstein J, Volfson D. 2013. Computationally efficient permutation-based confidence interval estimation for tail-area FDR. Frontiers in Genetics | Statistical Genetics and Methodology 4(179):1-11.

Examples

```
nrow_=100
ncol_=100
X = as.data.frame(matrix(rnorm(nrow_*ncol_),nrow=nrow_,ncol=ncol_))
Y = as.data.frame(matrix(rnorm(nrow_*ncol_),nrow=nrow_,ncol=ncol_))
nperm = 10
```

```
myanalysis = function(X,Y){
ntests = ncol(X)
rslts = as.data.frame(matrix(NA,nrow=ntests,ncol=2))
names(rslts) = c("ID","pvalue")
rslts[,"ID"] = 1:ntests
for(i in 1:ntests){
fit = cor.test(X[,i],Y[,i],na.action="na.exclude",
alternative="two.sided",method="pearson")
rslts[i,"pvalue"] = fit$p.value
}
return(rslts)
} # End myanalysis

# Generate observed results
obs = myanalysis(X,Y)

## Generate permuted results
perml = vector('list',nperm)
for(p_ in 1:nperm){
X1 = X[order(runif(ncol_)),]
perml[[p_]] = myanalysis(X1,Y)
}

## FDR results
fdr_od(obs$pvalue,perml,"pvalue",ncol_,.05)
```

Index