Package: cit (via r-universe)

September 25, 2024

Type Package

Title Causal Inference Test

Version 2.4.0

Date 2024-06-26

Description A likelihood-based hypothesis testing approach is implemented for assessing causal mediation. Described in Millstein, Chen, and Breton (2016), <DOI:10.1093/bioinformatics/btw135>, it could be used to test for mediation of a known causal association between a DNA variant, the 'instrumental variable', and a clinical outcome or phenotype by gene expression or DNA methylation, the potential mediator. Another example would be testing mediation of the effect of a drug on a clinical outcome by the molecular target. The hypothesis test generates a p-value or permutation-based FDR value with confidence intervals to quantify uncertainty in the causal inference. The outcome can be represented by either a continuous or binary variable, the potential mediator is continuous, and the instrumental variable can be continuous or binary and is not limited to a single variable but may be a design matrix representing multiple variables.

SystemRequirements gsl (with development libraries libgsl-dev)

License Artistic-2.0

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

LazyLoad yes

Suggests tinytest

Depends R (>= 3.5.0)

BugReports https://github.com/USCbiostats/cit/issues

URL https://github.com/USCbiostats/cit

LinkingTo Rcpp

Imports Rcpp

Repository https://mrcieu.r-universe.dev

RemoteUrl https://github.com/USCbiostats/cit

RemoteRef HEAD

RemoteSha 620f42dc9195d73b391497b6f1fccf6794756d67

Contents

`cit-package` *Causal Inference Test*

Description

This package implements a formal statistical hypothesis test for causal mediation. For example, it could be used to test for mediation of a known causal association between a DNA variant, the 'instrumental variable', and a clinical outcome or phenotype by gene expression or DNA methylation, the potential mediator. Another example would be testing mediation of the effect of a drug on a clinical outcome by the molecular target. The hypothesis test generates a p-value or permutation-based false discovery rate (FDR) value with confidence intervals to quantify uncertainty in the causal inference. The outcome can be represented by either a continuous or binary variable, the potential mediator is continuous, and the instrumental variable can be continuous or binary and is not limited to a single variable but may be a design matrix representing multiple variables.

Details

Package:	cit
Type:	Package
Version:	2.4.0
Date:	2024-06-26
License:	Artistic-2.0
LazyLoad:	yes

This package implements a novel statistical framework in which existing notions of causal mediation are formalized into a hypothesis test, thus providing a standard quantitative measure of uncertainty in the form of a p-value and optionally, permutation-based FDR. We treat the causal inference as a 'chain' of mathematical conditions that must be satisfied to conclude that the potential mediator is causal for the trait, where the inference is only as good as the weakest link in the chain. P-values are computed for the component conditions. The Intersection-Union Test, in which a series of statistical tests are combined to form an omnibus test, is then employed to generate the overall test result as the maximum of the component p-values. If we let L denote a locus of interest or other instrumental variable, T denote an outcome trait, and G denote a potential mediator, then the four component conditions are, 1) L and G are associated, 2) L and T are associated, 3) L is associated with G|T, and 4) L is independent of T|G. Test 4 requires an equivalence test and is implemented here using a permutation based approach (as described in Millstein et al. 2009).

Author(s)

Joshua Millstein

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

References

Millstein J, Chen GK, Breton CV. 2016. cit: hypothesis testing software for mediation analysis in genomic applications. Bioinformatics. btw135. PMID: 27153715. Millstein J, Zhang B, Zhu J, Schadt EE. 2009. Disentangling molecular relationships with a causal inference test. BMC Genetics, 10:23.

`cit.bp` *Causal Inference Test for a Binary Outcome*

Description

This function implements a formal statistical hypothesis test, resulting in a p-value, to quantify uncertainty in a causal inference pertaining to a measured factor, e.g. a molecular species, which potentially mediates a known causal association between a locus or other instrumental variable and a trait or clinical outcome. If the number of permutations is greater than zero, then the results can be used with fdr.cit to generate permutation-based FDR values (q-values) that are returned with confidence intervals to quantify uncertainty in the estimate. The outcome is binary, the potential mediator is continuous, and the instrumental variable can be continuous, discrete (such as coding a SNP 0, 1, 2), or binary and is not limited to a single variable but may be a design matrix representing multiple variables.

Usage

```
cit.bp( L, G, T, C=NULL, maxit=10000, n.perm=0, perm.index=NULL, rseed=NULL )
```

Arguments

L	vector or nxp design matrix representing the instrumental variable(s).
G	continuous vector representing the potential causal mediator.
T	Continuous vector representing the clinical trait or outcome of interest.
C	Vector or nxp design matrix representing adjustment covariates.
maxit	Maximum number of iterations to be conducted for the conditional independence test, test 4, which is permutation-based. The minimum number of permutations conducted is 1000, regardless of maxit. Increasing maxit will increase the precision of the p-value for test 4 if the p-value is small.
n.perm	If n.perm is set to an integer greater than 0, then n.perm permutations for each component test will be conducted (randomly permuting the data to generate results under the null).
perm.index	This item is only important when the CIT is conducted multiple times, leading to a multiple testing issue, that is addressed by computing FDR using the fdr.cit() function. To accurately account for dependencies among tests when computing FDR confidence intervals, the permutations must be the same for all tests. Thus, if 100 permutations are conducted for 500 CIT test scenarios, each of these 100 permutations are applied to all 500 tests. This is achieved by passing an argument, perm.index, which is an n row by n.perm column dataframe or matrix of permutation indices, where n is the number of observations and n.perm the number of permutations. Each column of perm.index includes a random permutation of 1:n, and this same perm.index object would passed to all 500 CIT tests. fdr.cit() would then be used to compute q-values (FDR) and q-value confidence intervals for each test.
rseed	If n.perm > 0, and multiple tests (CITs) are being conducted, setting rseed to the same integer for all tests insures that the permutations will be the same across CITs. This is important for maintaining the observed dependencies among tests for permuted data in order to compute accurate confidence intervals for FDR estimates.

Details

The omnibus p-value, p_cit, is the maximum of the component p-values, an intersection-union test, representing the probability of the data if at least one of the component null hypotheses is true. For component test 4, rather than using the semiparametric approach proposed by Millstein et al. (2009), here it is estimated completely by permutation, resulting in an exact test. If permutations are conducted by setting n.perm to a value greater than zero, then the results are provided in matrix (dataframe) form where each row represents an analysis using a unique permutation, except the first row (perm = 0), which has results from the observed or non-permuted analysis. These results can then be aggregated across multiple cit.bp tests and input to the function fdr.cit to generate component test FDR values (q-values) as well as omnibus q-values with confidence intervals that correspond to the p_cit omnibus p-values.

Value

A dataframe which includes the following columns:

perm	Indicator for permutation results. Zero indicates that the data were not permuted and subsequent rows include an integer greater than zero for each permutation conducted.
p_cit	CIT (omnibus) p-value
p_TassocL	component p-value for the test of association between T and L.
p_TassocGgvnL	component p-value for the test of association between T and G|L.
p_GassocLgvnT	component p-value for the test of association between G and L|T.
p_LindTgvnG	component p-value for the equivalence test of L ind T|G

Author(s)

Joshua Millstein

References

Millstein J, Chen GK, Breton CV. 2016. cit: hypothesis testing software for mediation analysis in genomic applications. Bioinformatics. btw135. PMID: 27153715. Millstein J, Zhang B, Zhu J, Schadt EE. 2009. Disentangling molecular relationships with a causal inference test. BMC Genetics, 10:23.

Examples

```
# Sample Size
ss = 100

# Errors
e1 = matrix(rnorm(ss),ncol=1)
e2 = matrix(rnorm(ss),ncol=1)

# Simulate genotypes, gene expression, covariates and a clinical trait
L = matrix(rbinom(ss*3,2,.5),ncol=3)
G =  matrix( apply(.4*L, 1, sum) + e1,ncol=1)
T =  matrix(.3*G + e2,ncol=1)
T = ifelse( T > median(T), 1, 0 )
C =  matrix(matrix(rnorm(ss*2),ncol=1),ncol=2)

n.perm = 5
perm.index = matrix(NA, nrow=ss, ncol=n.perm )
for( j in 1:ncol(perm.index) ) perm.index[, j] = sample( 1:ss )

results = cit.bp(L, G, T, perm.index=perm.index, n.perm=n.perm)
results

results = cit.bp(L, G, T)
results

results = cit.bp(L, G, T, C, n.perm=5)
results

results = cit.bp(L, G, T, C)
results
```

`cit.cp` *Causal Inference Test for a Continuous Outcome*

Description

This function implements a formal statistical hypothesis test, resulting in a p-value, to quantify uncertainty in a causal inference pertaining to a measured factor, e.g. a molecular species, which potentially mediates a known causal association between a locus or other instrumental variable and a quantitative trait. If the number of permutations is greater than zero, then the results can be used with fdr.cit to generate permutation-based FDR values (q-values) that are returned with confidence intervals to quantify uncertainty in the estimate. The outcome is continuous, the potential mediator is continuous, and the instrumental variable can be continuous, discrete (such as coding a SNP 0, 1, 2), or binary and is not limited to a single variable but may be a design matrix representing multiple variables.

Usage

```
cit.cp( L, G, T, C=NULL, n.resampl=50, n.perm=0, perm.index=NULL, rseed=NULL )
```

Arguments

`L`	Vector or nxp design matrix representing the instrumental variable(s).
`G`	Continuous vector representing the potential causal mediator.
`T`	Continuous vector representing the clinical trait or outcome of interest.
`C`	Vector or nxp design matrix representing adjustment covariates.
`n.resampl`	The number of instances of the test statistic for conditional independence (test 4) generated by permutation (Millstein et al. 2009) under the null hypothesis of no mediation (independent effects of L on G and T). These data are used to estimate the parameters of the null distribution. The default is set to 50, which we have found to provide reasonable precision.
`n.perm`	If n.perm is set to an integer greater than 0, then n.perm permutations for each component test will be conducted (randomly permuting the data to generate results under the null).
`perm.index`	This item is only important when the CIT is conducted multiple times, leading to a multiple testing issue, that is addressed by computing FDR using the fdr.cit() function. To accurately account for dependencies among tests when computing FDR confidence intervals, the permutations must be the same for all tests. Thus, if 100 permutations are conducted for 500 CIT test scenarios, each of these 100 permutations are applied to all 500 tests. This is achieved by passing an argument, perm.index, which is an n row by n.perm column dataframe or matrix of permutation indices, where n is the number of observations and n.perm the number of permutations. Each column of perm.index includes a random permutation of 1:n, and this same perm.index object would passed to all 500 CIT tests. fdr.cit() would then be used to compute q-values (FDR) and q-value confidence intervals for each test.

`rseed` If n.perm > 0, and multiple tests (CITs) are being conducted, setting rseed to the same integer for all tests insures that the permutations will be the same across CITs. This is important for maintaining the observed dependencies among tests for permuted data in order to compute accurate confidence intervals for FDR estimates.

Details

Increasing n.resampl will increase the precision of the component test 4, the conditional independence test. This may be useful if a very small p-value is observed and high precision is desired, however, it will increase run time. The omnibus p-value, p_cit, is the maximum of the component p-values, an intersection-union test, representing the probability of the data if at least one of the component null hypotheses is true. If permutations are conducted by setting n.perm to a value greater than zero, then the results are provided in matrix (dataframe) form, where each row represents an analysis using a unique permutation, except the first row (perm = 0), which has results from the observed or non-permuted analysis. These results can then be aggregated across multiple cit.cp tests and input to the function fdr.cit to generate component test FDR values (q-values) as well as omnibus q-values with confidence intervals that correspond to the p_cit omnibus p-values.

Value

A dataframe which includes the following columns:

`perm`	Indicator for permutation results. Zero indicates that the data were not permuted and subsequent rows include an integer greater than zero for each permutation conducted.
`p_cit`	CIT (omnibus) p-value
`p_TassocL`	component p-value for the test of association between T and L.
`p_TassocGgvnL`	component p-value for the test of association between T and G|L.
`p_GassocLgvnT`	component p-value for the test of association between G and L|T.
`p_LindTgvnG`	component p-value for the equivalence test of L ind T|G

Author(s)

Joshua Millstein

References

Millstein J, Chen GK, Breton CV. 2016. cit: hypothesis testing software for mediation analysis in genomic applications. Bioinformatics. btw135. PMID: 27153715. Millstein J, Zhang B, Zhu J, Schadt EE. 2009. Disentangling molecular relationships with a causal inference test. BMC Genetics, 10:23.

Examples

```
# Sample Size
ss = 100

# Errors
```

```
e1 = matrix(rnorm(ss),ncol=1)
e2 = matrix(rnorm(ss),ncol=1)

# Simulate genotypes, gene expression, covariates, and clinical trait matrices
L = matrix(rbinom(ss*3,2,.5),ncol=3)
G =  matrix( apply(.3*L, 1, sum) + e1,ncol=1)
T =  matrix(.3*G + e2,ncol=1)
C =  matrix(matrix(rnorm(ss*2),ncol=1),ncol=2)

n.perm = 5
perm.index = matrix(NA, nrow=ss, ncol=n.perm )
for( j in 1:ncol(perm.index) ) perm.index[, j] = sample( 1:ss )

results = cit.cp(L, G, T)
results

results = cit.cp(L, G, T, perm.index=perm.index, n.perm=5)
results

results = cit.cp(L, G, T, C)
results

results = cit.cp(L, G, T, C, n.perm=5)
results
```

`fdr.cit` *Omnibus FDR Values for CIT*

Description

Results including permutation results from cit.cp or cit.bp from multiple tests are used in a permutation-based approach to compute false discovery rates (q-values) for each test. q-values are computed for the omnibus test as well as all component tests. q-values are returned with confidence intervals to quantify uncertainty in the estimate.

Usage

```
fdr.cit( cit.perm.list, cl=.95, c1=NA )
```

Arguments

`cit.perm.list`	List, where each item is a dataframe output from cit.cp or cit.bp, which must be run with n.perm > 0.
`cl`	Confidence level for the q-value confidence interval (default is .95).
`c1`	Over-dispersion parameter. Setting the over-dispersion parameter to one corresponds to the assumption that all tests (referring to multiple omnibus tests conducted using different sets of variables) are independent. The default (c1=NA) is to use an empirically estimated over-dispersion parameter based on the permutation results.

Details

For the initial cit.cp and cit.bp test computations, we suggest setting n.perm to 100 or greater for moderate numbers of tests, say 10s or 100s. Larger numbers of permutations will generate more precise FDR estimates but require substantial computation time. If a large number of tests, say 1000s, have been conducted, then n.perm can be as small as 10 and still generate good FDR estimates and confidence intervals. FDR confidence intervals are especially useful to quantify uncertainty when FDR value > 0.05. The over-dispersion parameter should be set to one if it is known that all tests are independent, however, this is rarely the case in typical multiple testing settings. The fdr.cit function can also be used with simulated data in order to generate power calculations. In this case the over-dispersion parameter may be estimated from pilot data or other datasets and fixed to the prior estimate using the c1 argument. The omnibus CIT FDR value is generated as the maximum FDR across the four component tests, in an intesection-union type of approach analogous to the method used to generate the omnibus CIT p-value in cit.cp and cit.bp.

It is important to be aware that there are certain conditions under which this permutation-based FDR is not stable. This includes the case where all observed tests achieve the significance level and the case where there are zero positive tests among the permuted results. In the latter case, we take the conservative approach of setting the number of positive permutation tests to one. Both these cases produce an NA for the upper confidence limit. The latter problem may be alleviated by increasing the number of permutations conducted.

Value

A dataframe which includes the following columns:

`p.raw`	CIT (omnibus intersection/union test, IUT) p-value, max of component p-values
`q.cit`	CIT (omnibus FDR intersection/union test, IUT) q-value, permutation-based FDR, function of component test FDR values
`q.ll.cit`	Lower 95 percent confidence limit (if cl=.95) for q.cit estimate
`q.ul.cit`	Upper 95 percent confidence limit (if cl=.95) for q.cit estimate
`q.TaL`	component q-value (FDR) for the test of association between T and L
`q.ll.TaL`	Lower 95 percent confidence limit (if cl=.95) for q.TaL estimate
`q.ul.TaL`	Upper 95 percent confidence limit (if cl=.95) for q.TaL estimate
`q.TaGgvL`	component q-value (FDR) for the test of association between T and G|L
`q.ll.TaGgvL`	Lower 95 percent confidence limit (if cl=.95) for q.TaGgvL estimate
`q.ul.TaGgvL`	Upper 95 percent confidence limit (if cl=.95) for q.TaGgvL estimate
`q.GaLgvT`	component q-value (FDR) for the test of association between G and L|T
`q.ll.GaLgvT`	Lower 95 percent confidence limit (if cl=.95) for q.GaLgvT estimate
`q.ul.GaLgvT`	Upper 95 percent confidence limit (if cl=.95) for q.GaLgvT estimate
`q.LiTgvG`	component q-value (FDR) for the equivalence test of L ind T|G
`q.ll.LiTgvG`	Lower 95 percent confidence limit (if cl=.95) for the equivalence test of L ind T|G
`q.ul.LiTgvG`	Upper 95 percent confidence limit (if cl=.95) for the equivalence test of L ind T|G

Author(s)

Joshua Millstein

References

Millstein J, Chen GK, Breton CV. 2016. cit: hypothesis testing software for mediation analysis in genomic applications. Bioinformatics. btw135. PMID: 27153715. 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

```
# Sample Size
ss = 100
n.perm = 20
perm.index = matrix(NA, nrow=ss, ncol=n.perm )
for( j in 1:ncol(perm.index) ) perm.index[, j] = sample( 1:ss )

n.tests = 20
myresults = vector('list', n.tests)

for( tst in 1:n.tests ){

 # Errors
 e1 = matrix(rnorm(ss),ncol=1)
 e2 = matrix(rnorm(ss),ncol=1)

 # Simulate genotypes, gene expression, and clinical traits
 L = matrix(rbinom(ss,2,.5),ncol=1)
 G =  matrix(.5*L + e1,ncol=1)
 T =  matrix(.3*G + e2,ncol=1)
 T = ifelse( T > median(T), 1, 0 )

 myresults[[ tst ]] = cit.bp(L, G, T, perm.index=perm.index, n.perm=n.perm)
}
fdr.cit( myresults )
```

`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, od=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 for FDR confidence interval (default is .95).
od	Over-dispersion parameter. Setting the over-dispersion parameter to one corresponds to the assumption that all tests (referring to multiple omnibus tests conducted using different sets of variables) are independent. The default (c1=NA) is to use an empirically estimated over-dispersion parameter based on the permutation results.

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.

If there are zero positive tests among any of the permuted results (s.perm = 0) at the specified significance threshold (thres), then a conservative approximation is conducted by setting s.perm = 1 for the FDR calculation. This approximation is necessary for the permutation-based FDR to be estimable. This situation can be identified by noting that s.perm = 0 in the output. Increasing the number of permutations conducted may result in positive permutation tests, and thereby overcome the problem. 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 vector which includes:

FDR	FDR point estimate
ll	lower confidence limit
ul	upper confidence limit

pi0	proportion of true null hypotheses
od	overdispersion parameter
s.obs	observed number of positive tests
s.perm	total number of positive tests summed across all permuted result sets

Author(s)

Joshua Millstein

References

Millstein J, Chen GK, Breton CV. 2016. cit: hypothesis testing software for mediation analysis in genomic applications. Bioinformatics. btw135. PMID: 27153715. 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)
```

`fdr.q.para` *Parametric tail-area FDR Values, q-values*

Description

Given a vector of p-values, this function will use a parametric expression for tail-area FDR to assign q-values, minimum tail-area FDR, for each p-value. This function is used by the function fdr.cit to generate q-values for the forth test, LindTgvnG.

Usage

```
fdr.q.para( pvals )
```

Arguments

`pvals`	Vector of p-values.

Details

The parametric expression is constructed by setting the average number of positive tests among permuted results to p*m, where p is the p-value and m is the number of tests, the length of vector pvals. Following Storey and Tibshirani (2003), the q-value is set to the minimum FDR for p-values greater than or equal to the observed p-value.

Value

A vector of q-values.

Author(s)

Joshua Millstein

References

Millstein J, Chen GK, Breton CV. 2016. cit: hypothesis testing software for mediation analysis in genomic applications. Bioinformatics. btw135. 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. Storey DJ, Tibshirani R. 2003. Statistical significance for genomewide studies. PNAS 100(16):9440-9445.

Examples

```
# Sample Size
ss = 100

n.tests = 20
mypvals = rep(NA, n.tests)

for( tst in 1:n.tests ){
```

```
 # Errors
 e1 = matrix(rnorm(ss),ncol=1)
 e2 = matrix(rnorm(ss),ncol=1)

 # Simulate genotypes, gene expression, and clinical traits
 L = matrix(rbinom(ss,2,.5),ncol=1)
 G =  matrix(.5*L + e1,ncol=1)
 T =  matrix(.3*G + e2,ncol=1)
 T = ifelse( T > median(T), 1, 0 )

 mypvals[ tst ] = cit.bp(L, G, T)$p_cit
}
fdr.q.para( mypvals )
```

`fdr.q.perm` *Nonparametric permutation-based tail-area FDR Values, q-values*

Description

Given a vector of p-values, this function uses fdr.od() to compute tail-area FDR to assign q-values, minimum tail-area FDR, for each p-value. This function is used by the function fdr.cit to generate q-values for the three component tests, TassocL, TassocGgvnL, and GassocLgvnT.

Usage

```
fdr.q.perm( obs.p, perml, pname, ntests, cl=.95, od=NA )
```

Arguments

`obs.p`	observed vector of p-values.
`perml`	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 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.
`cl`	confidence level for FDR confidence interval (default is .95).
`od`	Over-dispersion parameter. Setting the over-dispersion parameter to one corresponds to the assumption that all tests (referring to multiple omnibus tests conducted using different sets of variables) are independent. The default (c1=NA) is to use an empirically estimated over-dispersion parameter based on the permutation results.

Details

Following Storey and Tibshirani (2003), the q-value is set to the minimum FDR for p-values greater than or equal to the observed p-value.

Value

A vector of q-values.

Author(s)

Joshua Millstein

References

Millstein J, Chen GK, Breton CV. 2016. cit: hypothesis testing software for mediation analysis in genomic applications. Bioinformatics. btw135. 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. Storey DJ, Tibshirani R. 2003. Statistical significance for genomewide studies. PNAS 100(16):9440-9445.

Examples

```
rowno=100
colno=100
X = as.data.frame(matrix(rnorm(rowno*colno),nrow=rowno,ncol=colno))
Y = as.data.frame(matrix(rnorm(rowno*colno),nrow=rowno,ncol=colno))
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(perm in 1:nperm){
 X1 = X[order(runif(colno)),]
 perml[[perm]] = myanalysis(X1,Y)
}
```

```
## FDR results
fdr.q.perm(obs$pvalue,perml,"pvalue",colno)
```

---

iuq *Intersection/Union Q-Value*

Description

Tail-area false discovery rates (FDRs) or 'q-values' are combined across multiple component tests to estimate a single q-value that represents the intersection of alternative hypotheses or the union of null hypotheses. In other words, this is an estimate of the rate at which tests called 'significant' at the specified omnibus q-value include at least one component condition that is called 'significant' due to chance alone.

Usage

```
iuq( qvec )
```

Arguments

qvec	vector of q-values, each corresponding to a component test.

Details

The omnibus q-value is greater than or equal to the maximum of the component q-values.

Value

A single value that represents the rate at which tests called 'significant' at the specified omnibus q-value include at least one component condition that is called 'significant' due to chance alone.

Author(s)

Joshua Millstein

References

Millstein J, Chen GK, Breton CV. 2016. cit: hypothesis testing software for mediation analysis in genomic applications. Bioinformatics. btw135. PMID: 27153715.

Examples

```
# component q-values
qvec = c( .1, .04, .02, .11 )

# omnibus q-value
iuq( qvec )
```

linreg *F Test for Linear Model*

Description

This function is used by cit.cp to compute F test given a continuous outcome and full vs reduced sets of covariates

Usage

```
linreg( nms.full, nms.redu=NULL, nm.y, mydat )
```

Arguments

nms.full	vector of variable names for all covariates included in the full model.
nms.redu	vector of variable names for all covariates included in the reduced model. If nms.redu is NULL then the reduced model is fitted with the intercept only.
nm.y	character string, which is the name of the outcome variable.
mydat	the dataframe that includes all variables with each variable in a column.

Details

An F test is conducted using the glm function by comparing the full and reduced models. This function is called by cit.cp.

Value

A single p-value is returned.

Author(s)

Joshua Millstein

References

Millstein J, Zhang B, Zhu J, Schadt EE. 2009. Disentangling molecular relationships with a causal inference test. BMC Genetics, 10:23.

Examples

```
ss = 500
cols = 6
nm.y = "y"
nms.full = paste( "x", 1:(cols-1), sep="" )
nms.redu = paste( "x", 1:2, sep="" )

mydat = as.data.frame( matrix( rnorm( ss*cols ), ncol=cols ) )
names( mydat ) = c( nm.y, nms.full )
```

```
linreg(nms.full, nms.redu, nm.y, mydat)
```

Index