Package ‘eQTLMAPT’

November 2, 2019

Type Package

Title eQTL Mediaiton Analysis with accelerated Permutation Testing approaches

Version 0.1.0

Author Qidi Peng

Maintainer The package maintainer <13753103470@163.com>

Description Do genome-intermediary analysis, use:
whether to use a Adaptive Permutation scheme,
whether to use the GPD fit to calculate the empirical P-value,
whether to use the Adaptive Confunding adjustments.

License GPL

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Depends parallel, readr

NeedsCompilation no

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

`child.p` *This function uses stratified fdr to figure out, for each locus, the list of covariates that do not play roles as child/intermediate mediator.*

Description

This function uses stratified fdr to figure out, for each locus, the list of covariates that do not play roles as child/intermediate mediator.

Usage

```
child.p(i, tripletmatrix, covariates)
```

`conf.fdr` *This function uses stratified fdr to figure out, for each covariate, the list of trios where the covariate plays a role as a confounder.*

Description

This function uses stratified fdr to figure out, for each covariate, the list of trios where the covariate plays a role as a confounder.

Usage

```
conf.fdr(i, tripletmatrix, covariates, conf_candidates, fdr)
```

dat *Example data*

Description

This simulated data list is for demonstration.

Value

A list containing

known.conf The known confounders matrix which is adjusted in all mediation tests. Each row is a confounder, each column is a sample.

cov.pool The pool of candidate confounding variables from which potential confounders are adaptively selected to adjust for each mediation test. Each row is a covariate, each column is a sample.

fea.dat The gene expression matrix. Each row is for one gene, each column is a sample.

snp.dat The eQTL genotype matrix. Each row is an eQTL, each column is a sample.

trios.idx The matrix of selected trios indexes (row numbers) for mediation tests. Each row consists of the index (i.e., row number) of the eQTLs in snp.dat, the index of cis-gene transcript in fea.dat, and the index of trans-gene transcript in fea.dat. The dimension is the number of trios by three.

Examples

```
data(example)
```

get.cov *This function get the Adaptive Confunding adjustments*

Description

This function get the Adaptive Confunding adjustments

Usage

```
get.cov(cl = NULL, cov.pool = NULL, fea.dat = NULL, triomatrix,
  fdr = 0.05, fdr_filter = 0.1)
```

get.pc *normalize fearession data so that every gene contributes equally to the construction of PCs*

Description

normalize fearession data so that every gene contributes equally to the construction of PCs

Usage

```
get.pc(fea.dat)
```

getp.func *This function caculate P-value for every trio*

Description

This function calculates the P-value for each trios. If Minperm=0, only the nominal P-value is calculated. If Minperm=Maxperm, the empirical P-value is calculated using a fixed number of permutation statistics; otherwise, the empirical P-value is calculated using the adaptive permutation scheme. The user can specify whether to use the GPD fit to estimate a more accurate empirical P-value, and at the same time specify how small the empirical P-value is for GPD fitting.

Usage

```
getp.func(i, triomatrix, confounders, Minperm = 100, Maxperm = 10000,
  use.gpd = FALSE, gpd.perm = 0.01, pool_cov = NULL,
  est_conf_pool_idx = NULL, use.PC = FALSE)
```

Arguments

i	Trios index in triomatrix
triomatrix	A three-dimensional matrix of size: samples number * trios number * 3. Triomatrix[i,j,1] represents the genotype of the j-th trios at the i-th sample,and triomatrix[i,j,2] represents the feature1 data of the j-th trios at the i-th sample, triomatrix[i,j,3] represents the feature2 data of the j-th trios at the i-th sample.
Minperm	Decide whether to use the parameters of the GPD fit. If the value is 0, only the nominal P-value is calculated. If the proportion of the permutation statistic better than the original statistic to the total number of permutations exceeds this value, a more accurate empirical P value is estimated using the GPD fit. If Minperm=0, only the nominal P-value is calculated. We set Minperm=100 as default.
Maxperm	Maximum number of permutation. We set Maxperm=10000 as default.
use.gpd	Whether to use the GPD fit to estimate a more accurate empirical P-value. We set use.gpd=NULL as default.
gpd.perm	The proportion parameter for estimating the empirical P-value when using GPD fit. When the proportion of permutation better than the original statistic is greater than par, the GPD is fitted to estimate the empirical P-value. We set gpd.perm=0.01 as default.
pool_cov	Candidate Confusion Variable Pool. We set pool_cov=NULL as default, which use.PC requied.
est_conf_pool_idx	The index of the adaptively selected confunding variable. We set pool_cov=NULL as default, at this time pool_cov=NULL too.
use.PC	Whether the candidate confusion variable pool is PCs.
confunders	A confounders matrix which is adjusted in all mediation tests.

Value

The algorithm will return a list of nperm, empirical.p, empirical.p.gpd, nominal.p, std.error, t_stat, beta, beta.total, beta.change.

`nperm` The actual number of permutations for testing mediation, equal to the input parameter nperm.

`empirical.p` The mediation Empirical P-values with nperm times permutation.

`empirical.p.gpd`
The mediation Empirical P-values with nperm times permutation using GPD fit.

`nominal.p` The mediation nominal P-values. A matrix with dimension of the number of trios.

`std.error` The return std.error value of feature1 for fit liner models.

`t_stat` The return t_stat value of feature1 for fit liner models.

`beta` The return beta value of feature2 for fit liner models in the case of feature1.

`beta.total` The return beta value of feature2 for fit liner models without considering feature1.

`beta.change` The proportions mediated.

`gmap` *Genomic Mediation analysis with Adaptive Petmutation scheme*

Description

The gmap function performs genomic mediation analysis with Adaptive Permutation scheme. It tests for mediation effects for a set of user specified mediation trios(e.g., eQTL, cis- and trans-genes) in the genome with the assumption of the presence of cis-association.

It returns the mediation p-values(nominal and empirical), the coefficient of linear models(e.g, t_stat, std.error, beta, beta.total) and the proportions mediated(e.g., the percentage of reduction in trans-effects after accounting for cis-mediation).

Usage

```
gmap(snp.dat, fea.dat, conf, trios.idx, cl = NULL, Minperm = 100,
  Maxperm = 10000)
```

Arguments

`snp.dat` The eQTL genotype matrix. Each row is an eQTL, each column is a sample.

`fea.dat` A feature profile matrix. Each row is for one feature, each column is a sample.

`conf` A confounders matrix which is adjusted in all mediation tests. Each row is a confounder, each column is a sample.

`trios.idx` A matrix of selected trios indexes (row numbers) for mediation tests. Each row consists of the index (i.e., row number) of the eQTL in `snp.dat`, the index of cis-gene feature in `fea.dat`, and the index of trans-gene feature in `fea.dat`. The dimension is the number of trios by three.

`cl` Parallel backend if it is set up. It is used for parallel computing. We set `cl`=NULL as default.

`Minperm` The minimum number of permutations. When the number of permutation statistics better than the original statistic is greater than `Minperm`, stop permutation and directly calculate the empirical P value. If `Minperm`=0, only the nominal P-value is calculated. We set `Minperm`=100 as default.

`Maxperm` Maximum number of permutation. We set `Maxperm`=10000 as default.

Details

The function performs genomic mediation analysis with Adaptive Permutation scheme. `Adaptive Permutation scheme` using Fixed Permutation scheme, good estimation of insignificant adjusted P-values can be achieved with few permutations while many more are needed to estimate highly significant ones. Therefore, we implemented an alternative permutation scheme that adapts the number of permutations to the significance level of the variant–phenotype pairs.

Value

The algorithm will return a list of nperm, empirical.p, nominal.p, beta, std.error, t_stat, beta.total, beta.change.

`nperm` The actual number of permutations for testing mediation.

`empirical.p` The mediation empirical P-values with nperm times permutation. A matrix with dimension of the number of trios.

`nominal.p` The mediation nominal P-values. A matrix with dimension of the number of trios.

`std.error` The return std.error value of feature1 for fit liner models. A matrix with dimension of the number of trios.

`t_stat` The return t_stat value of feature1 for fit liner models. A matrix with dimension of the number of trios.

`beta` The return beta value of feature2 for fit liner models in the case of feature1. A matrix with dimension of the number of trios.

`beta.total` The return beta value of feature2 for fit liner models without considering feature1. A matrix with dimension of the number of trios.

`beta.change` The proportions mediated. A matrix with dimension of the number of trios.

References

Ongen H, Buil A, Brown AA, Dermitzakis ET, Delaneau O. (2016) Fast and efficient QTL mapper for thousands of molecular phenotypes. Bioinformatics. 2016;32:1479–1485. doi: 10.1093/bioinformatics/btv722

Examples

```
output <- gmap(conf = dat$known.conf, fea.dat = dat$fea.dat, snp.dat = dat$snp.dat,
               trios.idx = dat$trios.idx[1:10,], Minperm = 100, Maxperm = 10000)

## Not run:
  ## generate a cluster with 2 nodes for parallel computing
  cl <- makeCluster(2)
  output <- gmap(conf = dat$known.conf, fea.dat = dat$fea.dat, snp.dat = dat$snp.dat,
               trios.idx = dat$trios.idx[1:10,], cl = cl, Minperm = 100, Maxperm = 10000)
  stopCluster(cl)
```

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

gmap.ac *Genomic Mediation analysis with Adaptive Petmutation scheme and Adaptive Confunders*

Description

The gmap.ac function performs genomic mediation analysis with Adaptive Permutation scheme and Adaptive Confunders. It tests for mediation effects for a set of user specified mediation trios(e.g., eQTL, cis- and trans-genes) in the genome with the assumption of the presence of cis-association. The gmap.ac function considers either a user provided pool of potential confounding variables, real or constructed by other methods, or all the PCs based on the feature data as the potential confounder pool.

It returns the mediation p-values(nominal and empirical), the coefficient of linear models(e.g, t_stat, std.error, beta, beta.total), and the proportions mediated(e.g., the percentage of reduction in trans-effects after accounting for cis-mediation) based on the mediation tests i) adjusting for known confounders only, and ii) adjusting for known confounders and adaptively selected potential confounders for each mediation trio.

Usage

```
gmap.ac(snp.dat, fea.dat, known.conf, trios.idx, cl = NULL,
  cov.pool = NULL, Minperm = 100, Maxperm = 10000, fdr = 0.05,
  fdr_filter = 0.1)
```

Arguments

snp.dat The eQTL genotype matrix. Each row is an eQTL, each column is a sample.

fea.dat A feature profile matrix. Each row is for one feature, each column is a sample.

known.conf A known confounders matrix which is adjusted in all mediation tests. Each row is a confounder, each column is a sample.

trios.idx A matrix of selected trios indexes (row numbers) for mediation tests. Each row consists of the index (i.e., row number) of the eQTL in snp.dat, the index of cis-gene feature in fea.dat, and the index of trans-gene feature in fea.dat. The dimension is the number of trios by three.

cl Parallel backend if it is set up. It is used for parallel computing. We set cl=NULL as default.

cov.pool The pool of candidate confounding variables from which potential confounders are adaptively selected to adjust for each trio. Each row is a covariate, each column is a sample. We set cov.pool=NULL as default, which will calculate PCs of features as cov.pool.

Minperm The minimum number of permutations. When the number of permutation statistics better than the original statistic is greater than Minperm, stop permutation and directly calculate the empirical P value. If Minperm=0, only the nominal P-value is calculated. We set Minperm=100 as default.

Maxperm Maximum number of permutation. We set Maxperm=10000 as default.

`fdr` The false discovery rate to select confounders. We set `fdr`=0.05 as default.

`fdr_filter` The false discovery rate to filter common child and intermediate variables. We set `fdr_filter`=0.1 as default.

Details

The function performs genomic mediation analysis with Adaptive Permutation scheme and Adaptive Confunders. `Adaptive Permutation scheme`When using Fixed Permutation scheme, good estimation of insignificant adjusted P-values can be achieved with few permutations while many more are needed to estimate highly significant ones. Therefore, we implemented an alternative permutation scheme that adapts the number of permutations to the significance level of the variant–phenotype pairs `Adaptive Confunding adjustment` One challenge in mediation test in genomic studies is how to adjust unmeasured confounding variables for the cis- and trans-genes (i.e., mediator-outcome) relationship.The current function adaptively selects the variables to adjust for each mediation trio given a large pool of constructed or real potential confounding variables. The function allows the input of variables known to be potential cis- and trans-genes (mediator-outcome) confounders in all mediation tests (`known.conf`), and the input of the pool of candidate confounders from which potential confounders for each mediation test will be adaptively selected (`cov.pool`). When no pool is provided (`cov.pool = NULL`), all the PCs based on feature profile (`fea.dat`) will be constructed as the potential confounder pool. `calculate Empirical P-values using GPD fitting`The use of a fixed number of permutations to calculate empirical P-values has the disadvantage that the minimum empirical P-value that can be calculated is 1/N. This makes a larger number of permutations needed to calculate a smaller P-value. Therefore, we model the tail of the permutation value as a Generalized Pareto Distribution(GPD), enabling a smaller empirical P-value with fewer permutation times.

Value

The algorithm will return a list of nperm, empirical.p, nominal.p, beta, std.error, t_stat, beta.total, beta.change.

`nperm` The actual number of permutations for testing mediation.

`empirical.p` The mediation empirical P-values with nperm times permutation. A matrix with dimension of the number of trios.

`nominal.p` The mediation nominal P-values. A matrix with dimension of the number of trios.

`std.error` The return std.error value of feature1 for fit liner models. A matrix with dimension of the number of trios.

`t_stat` The return t_stat value of feature1 for fit liner models. A matrix with dimension of the number of trios.

`beta` The return beta value of feature2 for fit liner models in the case of feature1. A matrix with dimension of the number of trios.

`beta.total` The return beta value of feature2 for fit liner models without considering feature1. A matrix with dimension of the number of trios.

`beta.change` The proportions mediated. A matrix with dimension of the number of trios.

`pc.matrix` PCs will be returned if the PCs based on expression data are used as the pool of potential confounders. Each column is a PC.

`sel.conf.ind` An indicator matrix with dimension of the number of trios by the number of covariates in `cov.pool` or `pc.matrix`if the principal components (PCs) based on expression data are used as the pool of potential confounders.

References

Ongen H, Buil A, Brown AA, Dermitzakis ET, Delaneau O. (2016) Fast and efficient QTL mapper for thousands of molecular phenotypes. Bioinformatics. 2016;32:1479–1485. doi: 10.1093/bioinformatics/btv722

Yang F, Wang J, Consortium G, Pierce BL, Chen LS. (2017) Identifying cis-mediators for trans-eQTLs across many human tissues using genomic mediation analysis. Genome Research. 2017;27:1859–1871. doi: 10.1101/gr.216754.116

Examples

```
output <- gmap.ac(known.conf = dat$known.conf, fea.dat = dat$fea.dat, snp.dat = dat$snp.dat,
               trios.idx = dat$trios.idx[1:10,], Minperm = 100, Maxperm = 10000)

## Not run:
  ## generate a cluster with 2 nodes for parallel computing
  cl <- makeCluster(2)

  ## Use the specified candidate confusion variable pool
 output <- gmap.ac(known.conf = dat$known.conf, fea.dat = dat$fea.dat, snp.dat = dat$snp.dat,
                 trios.idx = dat$trios.idx[1:10,], cl = cl, cov.pool = dat$cov.pool,
                 Minperm = 100, Maxperm = 10000)

  stopCluster(cl)

## End(Not run)
```

---

gmap.ac.gpd — *Genomic Mediation analysis with Adaptive Petmutation scheme and Adaptive Confunders and Generalized Pareto Distribution(GPD)*

Description

The gmap.ac.gpd function performs genomic mediation analysis with Adaptive Permutation scheme and Adaptive Confunders. It tests for mediation effects for a set of user specified mediation trios(e.g., eQTL, cis- and trans-genes) in the genome with the assumption of the presence of cis-association. The gmap.ac.gpd function considers either a user provided pool of potential confounding variables, real or constructed by other methods, or all the PCs based on the feature data as the potential confounder pool. When the empirical P-value is small enough, the GPD fit is used to estimate a more accurate empirical P value.

It returns the mediation p-values(nominal P-value, empirical P-values obtained from ordinary calculations and empirical P-values estimated using GPD fitting), the coefficient of linear models(e.g, t_stat, std.error, beta, beta.total) and the proportions mediated(e.g., the percentage of reduction in trans-effects after accounting for cis-mediation) based on the mediation tests i) adjusting for known confounders only, and ii) adjusting for known confounders and adaptively selected potential confounders for each mediation trio.

Usage

```
gmap.ac.gpd(snp.dat, fea.dat, known.conf, trios.idx, cl = NULL,
  cov.pool = NULL, Minperm = 100, Maxperm = 10000, gpd.perm = 0.01,
  fdr = 0.05, fdr_filter = 0.1)
```

Arguments

`snp.dat` The eQTL genotype matrix. Each row is an eQTL, each column is a sample.

`fea.dat` A feature profile matrix. Each row is for one feature, each column is a sample.

`known.conf` A known confounders matrix which is adjusted in all mediation tests. Each row is a confounder, each column is a sample.

`trios.idx` A matrix of selected trios indexes (row numbers) for mediation tests. Each row consists of the index (i.e., row number) of the eQTL in `snp.dat`, the index of cis-gene feature in `fea.dat`, and the index of trans-gene feature in `fea.dat`. The dimension is the number of trios by three.

`cl` Parallel backend if it is set up. It is used for parallel computing. We set `cl`=NULL as default.

`cov.pool` The pool of candidate confounding variables from which potential confounders are adaptively selected to adjust for each trio. Each row is a covariate, each column is a sample. We set `cov.pool`=NULL as default, which will calculate PCs of features as cov.pool.

`Minperm` The minimum number of permutations. When the number of permutation statistics better than the original statistic is greater than `Minperm`, stop permutation and directly calculate the empirical P value. If `Minperm`=0, only the nominal P-value is calculated. We set `Minperm`=100 as default.

`Maxperm` Maximum number of permutation. We set `Maxperm`=10000 as default.

`gpd.perm` Decide when to use GPD to fit estimation parameters. When the proportion of permutation better than the original statistic is greater than par, the GPD is fitted to estimate the empirical P-value. We set `gpd.perm`=0.01 as default.

`fdr` The false discovery rate to select confounders. We set `fdr`=0.05 as default.

`fdr_filter` The false discovery rate to filter common child and intermediate variables. We set `fdr_filter`=0.1 as default.

Details

The function performs genomic mediation analysis with Adaptive Permutation scheme and Adaptive Confunders. `Adaptive Permutation  scheme`When using Fixed Permutation scheme, good estimation of insignificant adjusted P-values can be achieved with few permutations while many more are needed to estimate highly significant ones. Therefore, we implemented an alternative permutation scheme that adapts the number of permutations to the significance level of the variant–phenotype pairs `Adaptive Confunding adjustment` One challenge in mediation test in genomic studies is how to adjust unmeasured confounding variables for the cis- and trans-genes (i.e., mediator-outcome) relationship.The current function adaptively selects the variables to adjust for each mediation trio given a large pool of constructed or real potential confounding variables. The function allows the input of variables known to be potential cis- and trans-genes (mediator-outcome) confounders in all mediation tests (`known.conf`), and the input of the pool of candidate confounders from which potential confounders for each mediation test will be adaptively selected (`cov.pool`). When no pool is provided (`cov.pool = NULL`), all the PCs based on feature profile (`fea.dat`) will be constructed as the potential confounder pool. `calculate Empirical P-values using GPD fitting`The use of a fixed number of permutations to calculate empirical P-values has the disadvantage that the minimum empirical P-value that can be calculated is 1/N. This makes a larger number of permutations needed to calculate a smaller P-value. Therefore, we model the tail of the permutation value as a Generalized Pareto Distribution(GPD), enabling a smaller empirical P-value with fewer permutation times. `calculate Empirical P-values using GPD fitting`The use of a fixed number of permutations to calculate empirical P-values has the disadvantage that the minimum empirical P-value that can be calculated is 1/N. This makes a larger number of permutations needed to calculate a smaller P-value. Therefore, we model the tail of the permutation value as a Generalized Pareto Distribution(GPD), enabling a smaller empirical P-value with fewer permutation times.

Value

The algorithm will return a list of nperm, empirical.p, empirical.p.gpd, nominal.p, beta, std.error, t_stat, beta.total, beta.change.

`nperm` The actual number of permutations for testing mediation.

`empirical.p` The mediation empirical P-values with nperm times permutation. A matrix with dimension of the number of trios.

`empirical.p.gpd`
The mediation empirical P-values with nperm times permutation using GPD fit. A matrix with dimension of the number of trios.

`nominal.p` The mediation nominal P-values. A matrix with dimension of the number of trios.

`std.error` The return std.error value of feature1 for fit liner models. A matrix with dimension of the number of trios.

`t_stat` The return t_stat value of feature1 for fit liner models. A matrix with dimension of the number of trios.

`beta` The return beta value of feature2 for fit liner models in the case of feature1. A matrix with dimension of the number of trios.

`beta.total` The return beta value of feature2 for fit liner models without considering feature1. A matrix with dimension of the number of trios.

`beta.change` The proportions mediated. A matrix with dimension of the number of trios.

`pc.matrix` PCs will be returned if the PCs based on expression data are used as the pool of potential confounders. Each column is a PC.

`sel.conf.ind` An indicator matrix with dimension of the number of trios by the number of covariates in `cov.pool` or `pc.matrix`if the principal components (PCs) based on expression data are used as the pool of potential confounders.

References

Ongen H, Buil A, Brown AA, Dermitzakis ET, Delaneau O. (2016) Fast and efficient QTL mapper for thousands of molecular phenotypes. Bioinformatics. 2016;32:1479–1485. doi: 10.1093/bioinformatics/btv722

Yang F, Wang J, Consortium G, Pierce BL, Chen LS. (2017) Identifying cis-mediators for trans-eQTLs across many human tissues using genomic mediation analysis. Genome Research. 2017;27:1859–1871. doi: 10.1101/gr.216754.116

Knijnenburg TA, Wessels LFA, Reinders MJT, Shmulevich I. (2009) Fewer permutations, more accurate P-values. Bioinformatics. 2009;25:i161–i168. doi: 10.1093/bioinformatics/btp211

Examples

```
output <- gmap.ac.gpd(known.conf = dat$known.conf, fea.dat = dat$fea.dat, snp.dat = dat$snp.dat,
                      trios.idx = dat$trios.idx[1:10,], Minperm = 100, Maxperm = 10000)

## Not run:
  ## generate a cluster with 2 nodes for parallel computing
  cl <- makeCluster(2)

  ## Use the specified candidate confusion variable pool
  ## When the empirical P-value is less than 0.02, a more accurate
     empirical P-value is estimated using the GPD fit.
 output <- gmap.ac.gpd(known.conf = dat$known.conf, fea.dat = dat$fea.dat, snp.dat = dat$snp.dat,
                    trios.idx = dat$trios.idx[1:10,], cl = cl, cov.pool = dat$cov.pool,
                      Minperm = 100, Maxperm = 10000, gpd.perm = 0.02)

  stopCluster(cl)

## End(Not run)
```

gmap.gpd *Genomic Mediation analysis with Fixed Permutation scheme and Generalized Pareto Distribution(GPD)*

Description

The gmap.gpd function performs genomic mediation analysis with Fixed Permutation scheme. It tests for mediation effects for a set of user specified mediation trios(e.g., eQTL, cis- and trans-genes) in the genome with the assumption of the presence of cis-association. When the empirical P-value is small enough, the GPD fit is used to estimate a more accurate empirical P value.

It returns the mediation p-values(nominal P-value, empirical P-values obtained from ordinary calculations and empirical P-values estimated using GPD fitting), the coefficient of linear models(e.g, t_stat, std.error, beta, beta.total) and the proportions mediated(e.g., the percentage of reduction in trans-effects after accounting for cis-mediation).

Usage

```
gmap.gpd(snp.dat, fea.dat, conf, trios.idx, cl = NULL, Minperm = 100,
  Maxperm = 10000, gpd.perm = 0.01)
```

Arguments

`snp.dat` The eQTL genotype matrix. Each row is an eQTL, each column is a sample.

`fea.dat` A feature profile matrix. Each row is for one feature, each column is a sample.

`conf` A confounders matrix which is adjusted in all mediation tests. Each row is a confounder, each column is a sample.

`trios.idx` A matrix of selected trios indexes (row numbers) for mediation tests. Each row consists of the index (i.e., row number) of the eQTL in `snp.dat`, the index of cis-gene feature in `fea.dat`, and the index of trans-gene feature in `fea.dat`. The dimension is the number of trios by three.

`cl` Parallel backend if it is set up. It is used for parallel computing. We set `cl`=NULL as default.

`Minperm` The minimum number of permutations. When the number of permutation statistics better than the original statistic is greater than `Minperm`, stop permutation and directly calculate the empirical P value. If `Minperm`=0, only the nominal P-value is calculated. We set `Minperm`=100 as default.

`Maxperm` Maximum number of permutation. We set `Maxperm`=10000 as default.

`gpd.perm` Decide when to use GPD to fit estimation parameters. When the proportion of permutation better than the original statistic is greater than par, the GPD is fitted to estimate the empirical P-value. We set `gpd.perm`=0.01 as default.

Details

The function performs genomic mediation analysis with Fixed Permutation scheme. `Adaptive Permutation schemeWh` using Fixed Permutation scheme, good estimation of insignificant adjusted P-values can be achieved with few permutations while many more are needed to estimate highly significant ones. Therefore, we implemented an alternative permutation scheme that adapts the number of permutations to the significance level of the variant–phenotype pairs `calculate  Empirical P-values using GPD fittingThe` use of a fixed number of permutations to calculate empirical P-values has the disadvantage that the minimum empirical P-value that can be calculated is 1/N. This makes a larger number of permutations needed to calculate a smaller P-value. Therefore, we model the tail of the permutation value as a Generalized Pareto Distribution(GPD), enabling a smaller empirical P-value with fewer permutation times.

Value

The algorithm will return a list of nperm, empirical.p, empirical.p.gpd, nominal.p, beta, std.error, t_stat, beta.total, beta.change.

`nperm` The actual number of permutations for testing mediation.

`empirical.p` The mediation empirical P-values with nperm times permutation. A matrix with dimension of the number of trios.

`empirical.p.gpd`
The mediation Empirical P-values with nperm times permutation using GPD fit. A matrix with dimension of the number of trios.

`nominal.p` The mediation nominal P-values. A matrix with dimension of the number of trios.

`std.error` The return std.error value of feature1 for fit liner models. A matrix with dimension of the number of trios.

`t_stat` The return t_stat value of feature1 for fit liner models. A matrix with dimension of the number of trios.

`beta` The return beta value of feature2 for fit liner models in the case of feature1. A matrix with dimension of the number of trios.

`beta.total` The return beta value of feature2 for fit liner models without considering feature1. A matrix with dimension of the number of trios.

`beta.change` The proportions mediated. A matrix with dimension of the number of trios.

References

Ongen H, Buil A, Brown AA, Dermitzakis ET, Delaneau O. (2016) Fast and efficient QTL mapper for thousands of molecular phenotypes. Bioinformatics. 2016;32:1479–1485. doi: 10.1093/bioinformatics/btv722

Knijnenburg TA, Wessels LFA, Reinders MJT, Shmulevich I. (2009) Fewer permutations, more accurate P-values. Bioinformatics. 2009;25:i161–i168. doi: 10.1093/bioinformatics/btp211

Examples

```
output <- gmap.gpd(conf = dat$known.conf, fea.dat = dat$fea.dat, snp.dat = dat$snp.dat,
              trios.idx = dat$trios.idx[1:10,], Minperm = 100, Maxperm = 10000)

## Not run:
  ## generate a cluster with 2 nodes for parallel computing
  cl <- makeCluster(2)

  ## When the empirical P-value is less than 0.02, a more accurate
     empirical P-value is estimated using the GPD fit.
  output <- gmap.gpd(conf = dat$known.conf, fea.dat = dat$fea.dat,
                    snp.dat = dat$snp.dat, trios.idx = dat$trios.idx[1:10,],
                    cl = cl, Minperm = 100, Maxperm = 10000, gpd.perm = 0.02)

  stopCluster(cl)

## End(Not run)
```

gmfp *Genomic Mediation analysis with Fixed Permutation scheme*

Description

The gmfp function performs genomic mediation analysis with fixed permutation. It tests for mediation effects for a set of user specified mediation trios(e.g., eQTL, cis- and trans-genes) in the genome with the assumption of the presence of cis-association.

It returns the mediation p-values(nominal and empirical), the coefficient of linear models(e.g, t_stat, std.error, beta, beta.total) and the proportions mediated(e.g., the percentage of reduction in trans-effects after accounting for cis-mediation).

Usage

```
gmfp(snp.dat, fea.dat, conf, trios.idx, cl = NULL, nperm = 10000)
```

Arguments

`snp.dat`	The eQTL genotype matrix. Each row is an eQTL, each column is a sample.
`fea.dat`	A feature profile matrix. Each row is for one feature, each column is a sample.
`conf`	A confounders matrix which is adjusted in mediation tests. Each row is a confounder, each column is a sample.
`trios.idx`	A matrix of selected trios indexes (row numbers) for mediation tests. Each row consists of the index (i.e., row number) of the eQTL in `snp.dat`, the index of cis-gene feature in `fea.dat`, and the index of trans-gene feature in `fea.dat`. The dimension is the number of trios by three.
`cl`	Parallel backend if it is set up. It is used for parallel computing. We set `cl`=NULL as default.
`nperm`	The number of permutations for testing mediation. If `nperm`=0, only the nominal P-value is calculated. We set `nperm`=10000 as default.

Details

The function performs genomic mediation analysis with fixed permutation. `Fixed Permutation scheme`When calculating the empirical P-value, the data is permutated by a fixed number of times, and the statistics after permutation are separately calculated. Assuming that the number of permutation is N, where the number of permutation statistics that is better than the original statistic is M, then the Empirical P-value = (M + 1) / (N + 1).

Value

The algorithm will return a list of empirical.p, nominal.p, beta, std.error, t_stat, beta.total, beta.change.

`empirical.p` The mediation empirical P-values with nperm times permutation. A matrix with dimension of the number of trios.

`nominal.p` The mediation nominal P-values. A matrix with dimension of the number of trios.

`std.error` The return std.error value of feature1 for fit liner models. A matrix with dimension of the number of trios.

`t_stat` The return t_stat value of feature1 for fit liner models. A matrix with dimension of the number of trios.

`beta` The return beta value of feature2 for fit liner models in the case of feature1. A matrix with dimension of the number of trios.

`beta.total` The return beta value of feature2 for fit liner models without considering feature1. A matrix with dimension of the number of trios.

`beta.change` The proportions mediated. A matrix with dimension of the number of trios.

Examples

```
output <- gmfp(conf = dat$known.conf, fea.dat = dat$fea.dat, snp.dat = dat$snp.dat,
               trios.idx = dat$trios.idx[1:10,], nperm = 100)

## Not run:
  ## generate a cluster with 2 nodes for parallel computing
  cl <- makeCluster(2)
  output <- gmfp(conf = dat$known.conf, fea.dat = dat$fea.dat, snp.dat = dat$snp.dat,
                 trios.idx = dat$trios.idx[1:10,], cl = cl, nperm = 100)
  stopCluster(cl)

## End(Not run)
```

`gmfp.ac` *Genomic Mediation analysis with Fixed Petmutation scheme and Adaptive Confunders*

Description

The gmfp.ac function performs genomic mediation analysis with fixed permutation and adaptive confounding adjustment. It tests for mediation effects for a set of user specified mediation trios(e.g., eQTL, cis- and trans-genes) in the genome with the assumption of the presence of cis-association. The gmfp.ac function considers either a user provided pool of potential confounding variables, real or constructed by other methods, or all the PCs based on the feautre data as the potential confounder pool.

It returns the mediation p-values(nominal and empirical), the coefficient of linear models(e.g, t_stat, std.error, beta, beta.total), and the proportions mediated(e.g., the percentage of reduction in trans-effects after accounting for cis-mediation) based on the mediation tests i) adjusting for known confounders only, and ii) adjusting for known confounders and adaptively selected potential confounders for each mediation trio.

Usage

```
gmfp.ac(snp.dat, fea.dat, known.conf, trios.idx, cl = NULL,
  cov.pool = NULL, nperm = 10000, fdr = 0.05, fdr_filter = 0.1)
```

Arguments

`snp.dat` The eQTL genotype matrix. Each row is an eQTL, each column is a sample.

`fea.dat` A feature profile matrix. Each row is for one feature, each column is a sample.

`known.conf` A confounders matrix which is adjusted in all mediation tests. Each row is a confounder, each column is a sample.

`trios.idx` A matrix of selected trios indexes (row numbers) for mediation tests. Each row consists of the index (i.e., row number) of the eQTL in `snp.dat`, the index of cis-gene feature in `fea.dat`, and the index of trans-gene feature in `fea.dat`. The dimension is the number of trios by three.

`cl` Parallel backend if it is set up. It is used for parallel computing. We set `cl`=NULL as default.

`cov.pool` The pool of candidate confounding variables from which potential confounders are adaptively selected to adjust for each trio. Each row is a covariate, each column is a sample. We set `cov.pool`=NULL as default, which will calculate PCs of features as cov.pool.

`nperm` The number of permutations for testing mediation. If `nperm`=0, only the nominal P-value is calculated. We set `nperm`=10000 as default.

`fdr` The false discovery rate to select confounders. We set `fdr`=0.05 as default.

`fdr_filter` The false discovery rate to filter common child and intermediate variables. We set `fdr_filter`=0.1 as default.

Details

The funciton performs genomic mediation analysis with fixed permutation and adaptive confounding adjustment. `Fixed Permutation scheme`When calculating the empirical P-value, the data is permutated by a fixed number of times, and the statistics after permutation are separately calculated. Assuming that the number of permutation is N, where the number of permutation statistics that is better than the original statistic is M, then the Empirical P-value = (M + 1) / (N + 1). `Adaptive Confunding adjustment` One challenge in mediation test in genomic studies is how to adjust unmeasured confounding variables for the cis- and trans-genes (i.e., mediator-outcome) relationship.The current function adaptively selects the variables to adjust for each mediation trio given a large pool of constructed or real potential confounding variables. The function allows the input of variables known to be potential cis- and trans-genes (mediator-outcome) confounders in all mediation tests (`known.conf`), and the input of the pool of candidate confounders from which potential confounders for each mediation test will be adaptively selected (`cov.pool`). When no pool is provided (`cov.pool = NULL`), all the PCs based on feature profile (`fea.dat`) will be constructed as the potential confounder pool.

Value

The algorithm will return a list of empirical.p, nominal.p, beta, std.error, t_stat, beta.total, beta.change.

`empirical.p` The mediation Empirical P-values with nperm times permutation. A matrix with dimension of the number of trios.

`nominal.p` The mediation nominal P-values. A matrix with dimension of the number of trios.

`std.error` The return std.error value of feature1 for fit liner models. A matrix with dimension of the number of trios.

`t_stat` The return t_stat value of feature1 for fit liner models. A matrix with dimension of the number of trios.

`beta` The return beta value of feature2 for fit liner models in the case of feature1. A matrix with dimension of the number of trios.

`beta.total` The return beta value of feature2 for fit liner models without considering feature1. A matrix with dimension of the number of trios.

`beta.change` The proportions mediated. A matrix with dimension of the number of trios.

`pc.matrix` PCs will be returned if the PCs based on expression data are used as the pool of potential confounders. Each column is a PC.

`sel.conf.ind` An indicator matrix with dimension of the number of trios by the number of covariates in `cov.pool` or `pc.matrixif` the principal components (PCs) based on expression data are used as the pool of potential confounders.

References

Yang F, Wang J, Consortium G, Pierce BL, Chen LS. (2017) Identifying cis-mediators for trans-eQTLs across many human tissues using genomic mediation analysis. Genome Research. 2017;27:1859–1871. doi: 10.1101/gr.216754.116

Examples

```
output <- gmfp.ac(known.conf = dat$known.conf, fea.dat = dat$fea.dat, snp.dat = dat$snp.dat,
                trios.idx = dat$trios.idx[1:10,], nperm = 100)

## Not run:
  ## generate a cluster with 2 nodes for parallel computing
  cl <- makeCluster(2)

  ## Use the specified candidate confusion variable pool
 output <- gmfp.ac(known.conf = dat$known.conf, fea.dat = dat$fea.dat, snp.dat = dat$snp.dat,
            trios.idx = dat$trios.idx[1:10,], cl = cl, cov.pool = dat$cov.pool, nperm = 100)

  stopCluster(cl)
```

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

gmfp.ac.gpd *Genomic Mediation analysis with Fixed Permutation scheme and Adaptive Confunders and Generalized Pareto Distribution(GPD)*

Description

The gmfp.ac.gpd function performs genomic mediation analysis with Fixed Permutation scheme and Adaptive Confunders. It tests for mediation effects for a set of user specified mediation trios(e.g., eQTL, cis- and trans-genes) in the genome with the assumption of the presence of cis-association. The gmfp.ac.gpd function considers either a user provided pool of potential confounding variables, real or constructed by other methods, or all the PCs based on the feature data as the potential confounder pool. When the empirical P-value is small enough, the GPD fit is used to estimate a more accurate empirical P value.

It returns the mediation p-values(nominal P-value, empirical P-values obtained from ordinary calculations and empirical P-values estimated using GPD fitting), the coefficient of linear models(e.g, t_stat, std.error, beta, beta.total), and the proportions mediated(e.g., the percentage of reduction in trans-effects after accounting for cis-mediation) based on the mediation tests i) adjusting for known confounders only, and ii) adjusting for known confounders and adaptively selected potential confounders for each mediation trio.

Usage

```
gmfp.ac.gpd(snp.dat, fea.dat, known.conf, trios.idx, cl = NULL,
  cov.pool = NULL, nperm = 10000, gpd.perm = 0.01, fdr = 0.05,
  fdr_filter = 0.1)
```

Arguments

snp.dat	The eQTL genotype matrix. Each row is an eQTL, each column is a sample.
fea.dat	A feature profile matrix. Each row is for one feature, each column is a sample.
known.conf	A confounders matrix which is adjusted in all mediation tests. Each row is a confounder, each column is a sample.
trios.idx	A matrix of selected trios indexes (row numbers) for mediation tests. Each row consists of the index (i.e., row number) of the eQTL in snp.dat, the index of cis-gene feature in fea.dat, and the index of trans-gene feature in fea.dat. The dimension is the number of trios by three.
cl	Parallel backend if it is set up. It is used for parallel computing. We set cl=NULL as default.
cov.pool	The pool of candidate confounding variables from which potential confounders are adaptively selected to adjust for each trio. Each row is a covariate, each column is a sample. We set cov.pool=NULL as default, which will calculate PCs of features as cov.pool.
nperm	The number of permutations for testing mediation. If nperm=0, only the nominal P-value is calculated. We set nperm=10000 as default.

`gpd.perm` Decide when to use GPD to fit estimation parameters. When the proportion of permutation better than the original statistic is greater than par, the GPD is fitted to estimate the empirical P-value. We set `gpd.perm`=0.01 as default.

`fdr` The false discovery rate to select confounders. We set `fdr`=0.05 as default.

`fdr_filter` The false discovery rate to filter common child and intermediate variables. We set `fdr_filter`=0.1 as default.

Details

The function performs genomic mediation analysis with Fixed Permutation scheme and Adaptive Confunders. `Fixed Permutation scheme`When calculating the empirical P-value, the data is permutated by a fixed number of times, and the statistics after permutation are separately calculated. Assuming that the number of permutation is N, where the number of permutation statistics that is better than the original statistic is M, then the Empirical P-value = (M + 1) / (N + 1). `Adaptive Confunding adjustment` One challenge in mediation test in genomic studies is how to adjust unmeasured confounding variables for the cis- and trans-genes (i.e., mediator-outcome) relationship.The current function adaptively selects the variables to adjust for each mediation trio given a large pool of constructed or real potential confounding variables. The function allows the input of variables known to be potential cis- and trans-genes (mediator-outcome) confounders in all mediation tests (`known.conf`), and the input of the pool of candidate confounders from which potential confounders for each mediation test will be adaptively selected (`cov.pool`). When no pool is provided (`cov.pool = NULL`), all the PCs based on feature profile (`fea.dat`) will be constructed as the potential confounder pool. `calculate Empirical P-values using GPD fitting`The use of a fixed number of permutations to calculate empirical P-values has the disadvantage that the minimum empirical P-value that can be calculated is 1/N. This makes a larger number of permutations needed to calculate a smaller P-value. Therefore, we model the tail of the permutation value as a Generalized Pareto Distribution(GPD), enabling a smaller empirical P-value with fewer permutation times.

Value

The algorithm will return a list of empirical.p, empirical.p.gpd, nominal.p, beta, std.error, t_stat, beta.total, beta.change.

`empirical.p` The mediation Empirical P-values with nperm times permutation. A matrix with dimension of the number of trios.

`empirical.p.gpd` The mediation empirical P-values with nperm times permutation using GPD fit. A matrix with dimension of the number of trios.

`nominal.p` The mediation nominal P-values. A matrix with dimension of the number of trios.

`std.error` The return std.error value of feature1 for fit liner models. A matrix with dimension of the number of trios.

`t_stat` The return t_stat value of feature1 for fit liner models. A matrix with dimension of the number of trios.

`beta` The return beta value of feature2 for fit liner models in the case of feature1. A matrix with dimension of the number of trios.

`beta.total` The return beta value of feature2 for fit liner models without considering feature1. A matrix with dimension of the number of trios.

`beta.change` The proportions mediated. A matrix with dimension of the number of trios.

`pc.matrix` PCs will be returned if the PCs based on feature data are used as the pool of potential confounders. Each column is a PC.

`sel.conf.ind` An indicator matrix with dimension of the number of trios by the number of covariates in `cov.pool` or `pc.matrix`if the principal components (PCs) based on feature data are used as the pool of potential confounders.

References

Yang F, Wang J, Consortium G, Pierce BL, Chen LS. (2017) Identifying cis-mediators for trans-eQTLs across many human tissues using genomic mediation analysis. Genome Research. 2017;27:1859–1871. doi: 10.1101/gr.216754.116

Knijnenburg TA, Wessels LFA, Reinders MJT, Shmulevich I. (2009) Fewer permutations, more accurate P-values. Bioinformatics. 2009;25:i161–i168. doi: 10.1093/bioinformatics/btp211

Examples

```
output <- gmfp.ac.gpd(known.conf = dat$known.conf, fea.dat = dat$fea.dat,
                    snp.dat = dat$snp.dat, trios.idx = dat$trios.idx[1:10,], nperm = 100)

## Not run:
  ## generate a cluster with 2 nodes for parallel computing
  cl <- makeCluster(2)

  ## Use the specified candidate confusion variable pool
  ## When the empirical P-value is less than 0.02, a more accurate
     empirical P-value is estimated using the GPD fit.
  output <- gmfp.ac.gpd(known.conf = dat$known.conf, fea.dat = dat$fea.dat,
                        snp.dat = dat$snp.dat, trios.idx = dat$trios.idx[1:10,],
                        cl = cl, cov.pool = dat$cov.pool, nperm = 100, gpd.perm = 0.02)

  stopCluster(cl)

## End(Not run)
```

gmfp.gpd *Genomic Mediation analysis with Fixed Permutation scheme and Generalized Pareto Distribution(GPD)*

Description

The gmfp.gpd function performs genomic mediation analysis with fixed permutation. It tests for mediation effects for a set of user specified mediation trios(e.g., eQTL, cis- and trans-genes) in the genome with the assumption of the presence of cis-association. When the empirical P-value is small enough, the GPD fit is used to estimate a more accurate empirical P value.

It returns the mediation p-values(nominal P-value, empirical P-values obtained from ordinary calculations and empirical P-values estimated using GPD fitting), the coefficient of linear models(e.g, t_stat, std.error, beta, beta.total) and the proportions mediated(e.g., the percentage of reduction in trans-effects after accounting for cis-mediation).

Usage

```
gmfp.gpd(snp.dat, fea.dat, conf, trios.idx, cl = NULL, nperm = 10000,
  gpd.perm = 0.01)
```

Arguments

`snp.dat` The eQTL genotype matrix. Each row is an eQTL, each column is a sample.

`fea.dat` A feature profile matrix. Each row is for one feature, each column is a sample.

`conf` A confounders matrix which is adjusted in mediation tests. Each row is a confounder, each column is a sample.

`trios.idx` A matrix of selected trios indexes (row numbers) for mediation tests. Each row consists of the index (i.e., row number) of the eQTL in `snp.dat`, the index of cis-gene feature in `fea.dat`, and the index of trans-gene feature in `fea.dat`. The dimension is the number of trios by three.

`cl` Parallel backend if it is set up. It is used for parallel computing. We set `cl`=NULL as default.

`nperm` The number of permutations for testing mediation. If `nperm`=0, only the nominal P-value is calculated. We set `nperm`=10000 as default.

`gpd.perm` Decide when to use GPD to fit estimation parameters. When the proportion of permutation better than the original statistic is greater than par, the GPD is fitted to estimate the empirical P-value. We set `gpd.perm`=0.01 as default.

Details

The function performs genomic mediation analysis with fixed permutation. `Fixed Permutation scheme`When calculating the empirical P-value, the data is permutated by a fixed number of times, and the statistics after permutation are separately calculated. Assuming that the number of permutation is N, where the number of permutation statistics that is better than the original statistic is M, then the Empirical P-value = (M + 1) / (N + 1). `calculate Empirical P-values using GPD fitting`The use of a fixed number of permutations to calculate empirical P-values has the disadvantage that the minimum empirical P-value that can be calculated is 1/N. This makes a larger number of permutations needed to calculate a smaller P-value. Therefore, we model the tail of the permutation value as a Generalized Pareto Distribution(GPD), enabling a smaller empirical P-value with fewer permutation times.

Value

The algorithm will return a list of empirical.p, empirical.p.gpd, nominal.p, beta, std.error, t_stat, beta.total, beta.change.

`empirical.p` The mediation empirical P-values with nperm times permutation. A matrix with dimension of the number of trios.

`empirical.p.gpd`
The mediation empirical P-values with nperm times permutation using GPD fit. A matrix with dimension of the number of trios.

`nominal.p` The mediation nominal P-values. A matrix with dimension of the number of trios.

`std.error` The return std.error value of feature1 for fit liner models. A matrix with dimension of the number of trios.

`t_stat` The return t_stat value of feature1 for fit liner models. A matrix with dimension of the number of trios.

`beta` The return beta value of feature2 for fit liner models in the case of feature1. A matrix with dimension of the number of trios.

`beta.total` The return beta value of feature2 for fit liner models without considering feature1. A matrix with dimension of the number of trios.

`beta.change` The proportions mediated. A matrix with dimension of the number of trios.

References

Knijnenburg TA, Wessels LFA, Reinders MJT, Shmulevich I. (2009) Fewer permutations, more accurate P-values. Bioinformatics. 2009;25:i161–i168. doi: 10.1093/bioinformatics/btp211

Examples

```
output <- gmfp.gpd(conf = dat$known.conf, fea.dat = dat$fea.dat, snp.dat = dat$snp.dat,
                   trios.idx = dat$trios.idx[1:10,], nperm = 100)

## Not run:
  ## generate a cluster with 2 nodes for parallel computing
  cl <- makeCluster(2)

  ## When the empirical P-value is less than 0.02, a more accurate
     empirical P-value is estimated using the GPD fit.
  output <- gmfp.gpd(conf = dat$known.conf, fea.dat = dat$fea.dat,
                     snp.dat = dat$snp.dat, trios.idx = dat$trios.idx[1:10,],
                     cl = cl, nperm = 100, gpd.perm = 0.02)

  stopCluster(cl)

## End(Not run)
```

gpdcdf *(1-CDF) of the generalized pareto distribution*

Description

(1-CDF) of the generalized pareto distribution

Usage

```
gpdcdf(z, a, k)
```

Value

p probability

`gpdfit` *Fitting the generalized pareto distribution using 'ML'*

Description

Fitting the generalized pareto distribution using 'ML'

Usage

```
gpdfit(z)
```

Value

Pharmat estimated shape and scale parameter

`gpdgoft` *Calculate whether the GPD fit is good enough*

Description

Goodness of fit test for the generalized pareto distribution (gpd) P-value of the null hypothesis that the data comes from (or can be modeled with) the fitted gpd. Small p-values indicate a bad fit.

Usage

```
gpdgoft(p, k)
```

Value

Pcm P-value using Cramer-von Mises statistic

Pad P-value using Anderson-Darling statistic (this gives more weight to observations in the tail of the distribution)

W2 Cramer-von Mises statistic

A2 Anderson-Darling statistic

`lfdr` *lfdr is a function to estimate the local FDR values from p-values.*

Description

The code for this function is extracted from the "qvalue" package written by Alan Dabney and John Storey.

Usage

```
lfdr(p, pi0 = NULL, trunc = TRUE, monotone = TRUE,
  transf = c("probit", "logit"), adj = 1.5, eps = 10^-8, ...)
```

Pgpd *Computing permutation test P-value of the GPD approximation using 'ML'*

Description

Computing permutation test P-value of the GPD approximation using 'ML'

Usage

```
Pgpd(y, x0, N, Nexc)
```

Value

Phat estimated P-value

pi0est *pi0est is a function to estimates the proportion of true null p-values.*

Description

The code for this function is extracted from the "qvalue" package written by Alan Dabney and John Storey.

Usage

```
pi0est(p, lambda = seq(0.05, 0.95, 0.05), pi0.method = c("smoother",
  "bootstrap"), smooth.df = 3, smooth.log.pi0 = FALSE, ...)
```

Ppermest *Estimation of P-value using Generalized Pareto Distribution(GPD)*

Description

Estimation of P-value using Generalized Pareto Distribution(GPD)

Usage

```
Ppermest(x0, y, Nexcmax = 250, proportion = 0.01)
```

Value

P Estimate of the P-value.

qvalue — *qvalue is the function to estimate q values given a vector of p values.*

Description

The code for this function is extracted from the "qvalue" package written by Alan Dabney and John Storey. We include the functions of qvalue here since it is reported than some operating system cannot isntall it, and we are using version 2.8.0.

Usage

```
qvalue(p, fdr.level = NULL, pfdr = FALSE, lfdr.out = TRUE,
  pi0 = NULL, ...)
```

SimplexMethod — *Multidimensional unconstrained nonlinear minimization (Nelder-Mead)*

Description

Multidimensional unconstrained nonlinear minimization (Nelder-Mead)

Usage

```
SimplexMethod(x, MaxFunEvals = "default", MaxIter = "default",
  TolFun = 1e-10, TolX = 1e-10, varargin)
```

Value

x Minimum objective function value point.

fval Minimum objective function value.

exitflag Return flag. If the result is found, it returns 1; if the number of function calculations or the number of iterations reaches the set value, there is still no result, and 0 is returned.

Index