

Package ‘efs’

November 25, 2006

Title estimate false discovery rate in sequential genome scan

Version 0.1

Author wei zou <wzou@statgen.ncsu.edu> and Sheng-Mao Chang<schang2@unity.ncsu.edu>

Description This package is to estimate FDR for multiple QTL genetic models obtained using sequential genome scans, with genome-wide threshold

Maintainer wei zou <wzou@statgen.ncsu.edu>

Depends R (>= 1.7.0), gss, spline, qvalue

R topics documented:

attach.locfdr	1
compute.FDR	2
efs-package	3
find.QTL	4
find.locfdr	5
find.value	6
getP0	7
getP0.efron	8
getP0.storey	9
getTailProb	10
locfdr.sm	10

Index **12**

`attach.locfdr` *attach the probability that a QTL is true for each QTL*

Description

This function combines the result from `find.locfdr` with `genome.scan`.

Arguments

`genome.scan` A same data frame as describe in the documentation for `getP`)
`lptable` The output of `find.locfdr`

Details

It is possible to make further modification in *lptable* before it is supplied to this function. As documented in `find.locfdr`, by default, we will enforce a monotony relationship between test statistics and the probabilities at the two ends of a splined curve, before it is output by `find.locfdr`.

Its result will be provided to `compute.FDR`.

Value

It returns *genome.scan* with an additional column 'p.true'.

References

possibly secondary sources and usages

Examples

```
# load data
data(o.test,o.null);
test <- o.test;
null <- o.null;

p <- getP0 (test,null, p0.method ="storey");
# p will contain the prior probablity for each cycle.

f <- find.locfdr (test, null, P0=p);
# find the relationship between the test statistics and the probability
# that the QTL is true
#
test <- attach.locfdr(test,f);
# Apply the relationship stored in 'f' to the sequential genome scan results.
#
```

`compute.FDR`

compute FDR for sequential genome scans

Description

FDR for sequential genome scans is computed as the average rate of declaring a false multiple genetic model.

Usage

```
compute.FDR(genome.scan)
```

Arguments

`genome.scan` A same data frame as describe in the documentation for `getP`). If there is a column 'isQTL', this function will calculate the FDR in the fixed rejection region. In the column 'isQTL', 1 is interpreted as the trait in the same row has a QTL declared in the cycle indicated in the column 'cyc'; 0 for otherwise. If there is no column 'isQTL', this function will calculate the probability that a multiple QTL genetic model is true for all traits in this input variable.

Details

If QTL have been specified, this function gives the final result: FDR in sequential genome scans. If QTL have not been specified, this function is to be called by `find.QTL`. In the later case, we will be interested in the 'qList' element of the output.

Value

A list containing two elements. A 'FDR' element is the FDR if QTL have been specified in the argument *genome.scan*. A 'qList' element is a data frame with a 'trait' column and a 'q' column. The 'q' column contains the probability that a multiple QTL genetic model is FALSE for the trait in the 'trait' column of the same row. The multiple QTL genetic model contains QTL found in all cycles of sequential genome scans for a trait.

References

Storey, J. D. and Akey, J. M. and Kruglyak, L. (2005) Multiple locus linkage analysis of genome-wide expression in yeast. *PLoS Biol*, 3: e267.

Examples

```
# load data
data(gs.null,gs.test)
test <- gs.test;
null <- gs.null;

p <- getP0 (test,null,p0.method="efron",p0.quantile=0.25);
# p will contain the prior probability for each cycle.

f <- find.locfdr <- (test, null, P0=p);
# find the relationship between the test statistics and the probability
# that the QTL is true
#
test <- attach.locfdr(test,f);
# Apply the relationship stored in 'f' to the sequential genome scan results.
#
compute.FDR(test);
```

 efs-package

What the package does (short line) package title

Description

More about what it does (maybe more than one line) A concise (1-5 lines) description of the package

Details

Package: efs
 Type: Package
 Version: 1.0
 Date: 2006-11-14
 License: What license is it under?

An overview of how to use the package, including the most important functions

Author(s)

Who wrote it

Maintainer: Who to complain to <yourfault@somewhere.net> The author and/or maintainer of the package

References

Literature or other references for background information

See Also

Optional links to other man pages, e.g. <pkg>

Examples

~~ simple examples of the most important functions ~~

```
find.QTL          declare QTL models while controlling FDR
```

Description

This function implements the algorithm by Storey in his 2005 PLoS Biology paper. It can find a rejection region so that the corresponding FDR is controlled at a specified level.

Usage

```
find.QTL (genome.scan = NULL ,FDR = 0.05, n.QTL = 0);
```

Arguments

genome.scan	A same data frame as describe in the documentation for <code>getP</code> . It must include a 'p.true' column, for the probability that an individual QTL is true. This column can be generated by calling <code>attach.locfdr</code> . It must not have a 'isQTL' column.
FDR	The desired level of FDR.
n.QTL	Number of QTL in a genetic model. If $n.QTL > 0$, only those traits with $n.QTL$ QTL will be considered: the function will find a subset of these traits with average error rates controlled at the specified level

Details

This function relies on `compute.FDR`.

Value

A data frame with a 'trait' column and a 'q' column.

References

Storey, J. D. and Akey, J. M. and Kruglyak, L. (2005) Multiple locus linkage analysis of genome-wide expression in yeast. PLoS Biol, 3: e267.

Examples

```
# load data
data(o.null,o.test)
test <- o.test
null <- o.null

p <- getP0 (test,null, p0.method ="storey");
# p will contain the prior probability for each cycle.
#
f <- find.locfdr (test, null, P0=p);
# find the relationship between the test statistics and the probability
# that the QTL is true
#
test <- attach.locfdr(test,f);
# Apply the relationship stored in 'f' to the sequential genome scan results.
#
result <- find.QTL (test,FDR=0.1,n.QTL=2);
# Find a rejection region while FDR is controlled at certain level.
```

find.locfdr

Find the probability that a test statistic is from a true QTL

Description

find a function that maps a test statistic to the probability that it is associated with a true QTL for each cycle of sequential genome scan.

Usage

```
find.locfdr (genome.scan, null, P0=p, smoothing.method = "logistic", dof=7,minP=
find.locfdr (genome.scan, null, P0=p)
```

Arguments

genome.scan	A same data frame as describe in the documentation for <code>getP</code>
null	A same data frame as describe in the documentation for <code>getP</code>
smoothing.method	If it is "logistic", this function will use natural spline technique in ns and logistic regressions to find a smooth curve. The method is described in Efron's 2001 JASA paper <i>Empirical Bayes analysis of a microarray experiment</i> . If it is "gss", this function will use general smoothing splines in ssden to find a smooth curve.
dof	Degree of freedom is a parameter for spline. Usually, small values, like the default value 7, generate small curves; larger values will result in rougher curves. It is not required if using general smoothing spline to fit the curve.

minP	It is not guaranteed that the local fdr will be bounded within [0,1]. Since local fdr should also be interpreted as a probability measure, all values smaller than <i>minP</i> is set to <i>minP</i>
maxP	It is not guaranteed that the local fdr will be bounded within [0,1]. Since local fdr should also be interpreted as a probability measure, all values larger than <i>maxP</i> is set to <i>maxP</i>
force.monotony	If it is set to TRUE, after obtaining the function that maps a test statistic to the probability that it is associated with a true QTL, this function will try to enforce a monotony relationship between test statistics and the probabilities at the two ends of a splined curve. Usually, we would like to believe that a larger test statistic will be more likely from a true QTL. A plot will be generated to show the curves before and after the modification for each cycle of sequential genome scan.

Details

The smoothing method based on natural splines seem more robust, though it requires a subjective argument: *dof*. Its speed is also faster than the method "gss". This function calls `locfdr.sm` to do the smoothing work.

The returned data frame will be supplied to `attach.locfdr`.

References

Efron, Bradley and Tibshirani, Robert and Storey, John D. and Tusher, Virginia (2001) Empirical Bayes analysis of a microarray experiment. *Journal of the American Statistical Association*, 96: 1151-1160.

Examples

```
# load data
data(o.null,o.test)
test <- o.test
null <- o.null

p <- getP0 (test,null, p0.method ="storey");
# p will contain the prior probability for each cycle.

f <- find.locfdr (test, null, P0=p);
# find the relationship between the test statistics and the probability
# that the QTL is true
#
```

find.value *find a value in a vector*

Description

find the vector element whose value is the closest to an input number

Usage

```
find.value(x.query = NULL, x.lib = NULL)
```

Arguments

`x.query` It must be a numeric vector. These values are to be searched in `x.lib`.
`x.lib` It must be a numeric vector.

Details

It is not intended to be called by users.

Value

A vector of indices of `x.lib` elements which have similar values as `x.query`.

<code>getP0</code>	<i>estimate prior probability</i>
--------------------	-----------------------------------

Description

This function is to find the prior probability that a statistic is associated with a false QTL.

Usage

```
getP0 (genome.scan, null, p0.method="efron", p0.quantile=0.25);
getP0 (genome.scan, null, p0.method = "storey", ...);
```

Arguments

(genome.scan=NULL, null.stat=NULL, p0.method="efron", p0.quantile=0.25, ...) normal-bracket17bracket-normal

`genome.scan` A data frame contains the complete result of sequential genome scan. It must have a column called 'cyc' to indicate the test statistic in the same row is obtained in which cycle of sequential genome scan. Please number the cycles as 1,2,3... It is recommended to pool QTL declared in the later cycles together if the number of QTL is small. Test statistics must be stored in a column called 'LR'. It also has to have a column called 'trait' to indicate the trait names

`null` A data frame stores test statistics obtained in permutation. It has a similar structure as `genome.scan`, but the 'trait' column is not required, since in this package, we will assume a common null distribution for test statistics of all traits in a certain cycle

`p0.method` The default value is 'efron'. It uses formula 6.7 in Efron's 2001 JASA paper *Empirical Bayes analysis of a microarray experiment*. In that case, `p0.quantile` is needed. Another option is 'storey', which will call [qvalue](#)

`p0.quantile` It specifies a quantile in the distribution of test statistics in a certain cycle: all test statistics that are less than that quantile are considered from false QTL.

... Addition arguments for [qvalue](#)

. For example, we can add `pi0.method="smoother"`.

Details

For each cycle of sequential genome scan, a series of tests will be performed for each of the thousands of traits in the data-set. The resulting list of test statistics are assumed to be sampled from a mixed distribution of statistics for true QTL and false QTL. This function is to find the prior probability that a statistic is associated with a false QTL for each cycle.

The 'cyc' values in *genome.scan* might range from 1 to J. The 'cyc' values in *null* might range from 1 to I. Due to the limitation of computational power, we sometime assume a common null distribution for test statistics collected in different cycles of sequential genome scans. This assumption allows $J > I$ (in our practice, $I=1$). In that case, the null statistics for the I-th cycle will be used to estimate the null distributions for cycle $[I+1, J]$.

In sequential genome scan like MIM, there will be less and less traits analyzed in later cycles. When we use the null statistics from cycle I to estimate the null distribution for cycle k ($k>I$), it is recommended to use a subset of the null statistics which are associated with the traits being analyzed in cycle k.

The resulting prior probabilities will be supplied to `find.locfdr`.

Value

A vector of length equal to number of cycles of sequential genome scans. The i-th element is the prior provability that a test statistic is from a false QTL in cycle i.

References

Storey JD and Tibshirani R. (2003) Statistical significance for genome-wide experiments. *Proceedings of the National Academy of Sciences*, 100: 9440-9445.

Efron, Bradley and Tibshirani, Robert and Storey, John D. and Tusher, Virginia (2001) Empirical Bayes analysis of a microarray experiment. *Journal of the American Statistical Association*, 96: 1151-1160.

Examples

```
# load data
data(o.test,o.null);
test <- o.test.txt;
null <- o.null.txt;

p <- getP0 (test,null, p0.method ="storey");
# p will contain the prior probability for each cycle.
```

getP0.efron

find prior probabilities using efron's suggestion

Description

This function uses efron's suggestion to get the prior probability that a statistic is from a false QTL. This function is not intended to be called by users.

Arguments

<code>test.stat</code>	A vector containing test statistics in a cycle of sequential genome scan.
<code>null.stat</code>	A vector of statistics obtained using permutation tests for the same cycle of sequential genome scan.
<code>p0.quantile</code>	It specifies a quantile in the distribution of test statistics in a certain cycle: all test statistics that are less than that quantile are considered from false QTL.

Details

It uses formula 6.7 in Efron's 2001 JASA paper *Empirical Bayes analysis of a microarray experiment*.

Value

A prior probability.

References

Efron, Bradley and Tibshirani, Robert and Storey, John D. and Tusher, Virginia (2001) Empirical Bayes analysis of a microarray experiment. *Journal of the American Statistical Association*, 96: 1151-1160.

`getP0.storey` *find prior probabilities using storey's method*

Description

This function calls [qvalue](#) to get the prior probability that a statistic is from a false QTL. This function is not intended to be called by users.

Usage

```
getP0.storey (p, ...)
```

Arguments

<code>p</code>	A vector of p values. Do not supply p values from different cycles of sequential genome scan in a same vector, since there is no reason to believe that different cycles have the same prior probability.
<code>...</code>	Check the documentation of qvalue

Details

This function is a wrapper of [qvalue](#)

Value

A prior probability

References

Storey JD and Tibshirani R. (2003) Statistical significance for genome-wide experiments. *Proceedings of the National Academy of Sciences*, 100: 9440-9445.

getTailProb	<i>Find the tail probabilities for test statistics</i>
-------------	--

Description

A p value for a test statistic is computed as the frequency of observing bigger statistics under the null hypothesis. This function is not intended to be called by users.

Usage

```
getTailProb (test.stat,null.stat)
```

Arguments

test.stat	A vector containing test statistics in a cycle of sequential genome scan.
null.stat	A vector of statistics obtained using permutation tests for the same cycle of sequential genome scan. For test statistics in the second round of sequential genome scan, CET method from Gary Churchill might be used

Details

Null statistics can be obtained using permutation test. Such p values are supplied to [qvalue](#)

Value

A list of p values for the input list of test statistics

References

Doerge, R. W. and Churchill, G. A. (1996) Permutation tests for multiple loci affecting a quantitative character. *Genetics*, 142: 285-94.

locfdr.sm	<i>find local false discovery rate</i>
-----------	--

Description

Find local fdr using logistic regression or general smoothing spline

Usage

```
locfdr.sm(z.mix, z.nul, p0, opt = 2, df = 7)
```

Arguments

z.mix	test statistics from a mixture of the true H0 and the true H1 hypothesis
z.nul	test statistics from a pure H0 hypothesis
p0	the prior probability that a test statistic in <i>z.mix</i> is from the true H0 hypothesis
opt	1 for general smooth spline, 2 for logistic regression
df	degree of freedom, required by the non-parametric logistic regression

Details

This function is not intended to be called by users.

Value

a data frame If it is a LIST, use

LR It contains the test statistics in *z.mix*

p the probability that the test statistic in the same row is from H1

References

Efron, Bradley and Tibshirani, Robert and Storey, John D. and Tusher, Virginia (2001) Empirical Bayes analysis of a microarray experiment. *Journal of the American Statistical Association*, 96: 1151-1160.

Index

*Topic **package**

efs-package, 3

<pkg>, 4

attach.locfdr, 1

compute.FDR, 2

efs (*efs-package*), 3

efs-package, 3

find.locfdr, 5

find.QTL, 4

find.value, 6

getP0, 7

getP0.efron, 8

getP0.storey, 9

getTailProb, 10

locfdr.sm, 10

ns, 5

qvalue, 7, 9, 10

ssden, 5