

Package ‘mFDP’

July 22, 2025

Type Package

Title Control of the Median of the FDP

Version 0.2.1

Date 2025-01-22

Maintainer Jesse Hemerik <hemerik@ese.eur.nl>

Description Methods for controlling the median of the false discovery proportion (mFDP).
Depending on the method, simultaneous or non-simultaneous inference is provided.
The methods take a vector of p-values or test statistics as input.

License GNU General Public License

Imports methods

NeedsCompilation no

Author Jesse Hemerik [aut, cre]

Repository CRAN

Date/Publication 2025-01-22 16:00:02 UTC

Contents

get.bound	2
get.kappa.max	3
mFDP.adjust	4
mFDP.direc	5
mFDP.equiv	5

Index	7
--------------	----------

get.bound	<i>Compute a 50 percent confidence upper bound for the number of false positives</i>
-----------	--

Description

For a p-value rejection threshold t , compute the 50 percent confidence upper bound for the number of false positives. The bounds are simultaneous over t .

Usage

```
get.bound(t, c, kappa.max)
```

Arguments

<code>t</code>	The p-value threshold
<code>c</code>	The tuning parameter, which influences the intercept and slope of the envelope. Should be numeric.
<code>kappa.max</code>	This value needs to be computed based on the p-values. Together with <code>c</code> it defines the bounds.

Value

A non-negative integer, which is a median unbiased (or upward biased) estimate of the number of false positives.

Examples

```
#Suppose the envelope that has been computed is defined by c=0.002 and kappa.max=0.001.
#We can then evaluate the envelope at several thresholds t as below.
#This is equivalent to simply entering the formula floor((t+c)/kappa.max).

#50 percent confidence upper bound for nr of false positives, if p-value threshold of 0.01 is used:
get.bound(t=0.01, c=0.002, kappa.max=0.001)    #12

#50 percent confidence upper bound for nr of false positives, if p-value threshold of 0.02 is used:
get.bound(t=0.02, c=0.002, kappa.max=0.001)    #22

#50 percent confidence upper bound for nr of false positives, if p-value threshold of 0.03 is used:
get.bound(t=0.03, c=0.002, kappa.max=0.001)    #32
```

get.kappa.max	<i>Based on a vector of raw p-values, compute kappa.max, which defines the mFDP envelope</i>
---------------	--

Description

Based on a vector of unadjusted(!) p-values, compute kappa.max, which together with c defines the mFDP envelope

Usage

```
get.kappa.max(P, c="1/(2m)", s1=0, s2=0.1)
```

Arguments

P	A vector of p-values.
c	The tuning parameter, which influences the intercept and slope of the envelope. Should either be numeric or " $1/(2m)$ " or " $1/m$ ".
s1	The smallest p-value threshold of interest. Non-negative.
s2	The largest p-value threshold of interest. Should be larger than s1 and at most 1.

Value

kappa.max, which together with c defines the mFDP envelope.

Examples

```
set.seed(5193)

### Make some p-values
m=500          #the nr of hypotheses
nrfalse=100    #the nr of false hypotheses

tstats = rnorm(n=m)  #m test statistics
tstats[1:nrfalse] = tstats[1:nrfalse] + 3  #add some signal
P = 1 - pnorm(tstats)  #compute p-values

### Compute kappa.max. (Taking c to be the default value 1/(2m).)
kappa.max = get.kappa.max(P=P)
kappa.max
```

mFDP.adjust	<i>compute mFDP-adjusted p-values</i>
-------------	---------------------------------------

Description

Provides mFDP-adjusted p-values, given a vector of p-values.

Usage

```
mFDP.adjust(P, c="1/(2m)", s1=0, s2=0.1)
```

Arguments

P	A vector of (raw, i.e. unadjusted) p-values.
c	The tuning parameter, which influences the intercept and slope of the envelope. Should either be numeric or "1/(2m)" or "1/m".
s1	The smallest p-value threshold of interest.
s2	The largest p-value threshold of interest.

Value

A vector of mFDP-adjusted p-values. Some can be infinity - which can be interpreted as 1.

Examples

```
set.seed(5193)

### make some p-values
m=500      #the nr of hypotheses
nrfalse=100 #the nr of false hypotheses

tstats = rnorm(n=m) #m test statistics
tstats[1:nrfalse] = tstats[1:nrfalse] + 3 #add some signal
P = 1 - pnorm(tstats) #compute p-values

P.adjusted = mFDP.adjust(P=P) #mFDP-adjusted p-values. Be careful with interpretation.

min(P.adjusted) #0.0208
```

mFDP.direc	<i>mFDP control for directional testing</i>
------------	---

Description

Controls the median of the FDP, given a vector of test statistics.

Usage

```
mFDP.direc(Ts, delta=0, gamma=0.05)
```

Arguments

Ts	A vector containing the test statistics $T_1, \dots, T_m$
delta	Determines the null hypotheses $H_1, \dots, H_m$. For every j , H_j is the hypothesis that T_j has mean at most δ .
gamma	Desired upper bound for the mFDP

Value

The indices of the rejected hypotheses

Examples

```
set.seed(123)

tstats = rnorm(n=200)+1.5 #make some test statistics

mFDP.direc(Ts = tstats) #indices of rejected hypotheses
```

mFDP.equiv	<i>mFDP control for equivalence testing</i>
------------	---

Description

Controls the median of the FDP, given a vector of test statistics.

Usage

```
mFDP.equiv(Ts, delta, gamma=0.05)
```

Arguments

<code>Ts</code>	A vector containing the test statistics $T_1, \dots, T_m$
<code>delta</code>	Determines the null hypotheses $H_1, \dots, H_m$. Should be >0 . For every j , H_j is the null hypothesis that $ T_j $ has mean at least δ . The alternative hypothesis is that $ T_j $ has mean smaller than δ
<code>gamma</code>	Desired upper bound for the mFDP

Value

The indices of the rejected hypotheses

Examples

```
set.seed(123)

tstats = rnorm(n=200) #make some test statistics

mFDP.equiv(Ts = tstats,delta=2) #indices of rejected hypotheses
```

Index

`get.bound`, [2](#)
`get.kappa.max`, [3](#)

`mFDP.adjust`, [4](#)
`mFDP.direc`, [5](#)
`mFDP.equiv`, [5](#)