

Package ‘TMTI’

August 13, 2021

Type Package

Title 'Too Many, Too Improbable' (TMTI) Test Procedures

Version 0.1.0

Author Phillip B. Mogensen

Maintainer Phillip B. Mogensen <pbm@math.ku.dk>

Description Methods for computing joint tests, controlling the Familywise Error Rate (FWER) and getting lower bounds on the number of false hypotheses in a set. The methods implemented here are described in Mogensen and Markussen (2021) <[arXiv:2108.04731](#)>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.1

Imports dplyr, stats

Suggests parallel

NeedsCompilation no

Repository CRAN

Date/Publication 2021-08-13 09:10:02 UTC

R topics documented:

.GetMinima	2
gamma_bootstrapper	2
localTest_CTP	3
make_Y	4
rtTMTI_CDF	5
TestSet_localTest	6
TestSet_TMTI	7
TMTI	8
TMTI_CDF	9
TMTI_CTP	10
TopDown_localTest	11
TopDown_TMTI	12
tTMTI_CDF	13

<code>.GetMinima</code>	<i>Helper function to return the first local minimum that is strictly smaller than the following n entries in a vector.</i>
-------------------------	--

Description

Helper function to return the first local minimum that is strictly smaller than the following n entries in a vector.

Usage

```
.GetMinima(Y, n)
```

Arguments

<code>Y</code>	A vector of transformed p-values
<code>n</code>	The number describing the kind of local minimum to return; $n=1$ returns the first local minimum, $n=Inf$ returns the global minimum.

Value

The index of the n 'th local minimum

Examples

```
## Simulate some p-values
p <- runif(10)
## Generate the transformations
Y <- pbeta(sort(p), 1:10, 10 + 1 - 1:10)
## Get the first local minimum
.GetMinima(Y, 1)
## Get the global minimum:
.GetMinima(Y, Inf) # Note, equivalent to which.min(Y)
```

<code>gamma_bootstrapper</code>	<i>Function to bootstrap the Cumulative Distribution Functions (CDFs) of the TMTI statistics.</i>
---------------------------------	---

Description

Function to bootstrap the Cumulative Distribution Functions (CDFs) of the TMTI statistics.

Usage

```
gamma_bootstrapper(
  m,
  n = Inf,
  B = 1000,
  log.p = TRUE,
  mc.cores = 1L,
  tau = NULL,
  K = NULL
)
```

Arguments

m	Number of tests
n	Number (or Inf) indicating what kind of minimum to consider. Defaults to Inf, corresponding to the global minimum.
B	Number of bootstrap replicates. Rule of thumb is to use at least $10 * m$
log.p	Logical indicating whether to calculate p-values on log-scale. Defaults to TRUE
mc.cores	Integer denoting the number of cores to use when using parallelization, Defaults to 1, corresponding to single-threaded computations
tau	Numerical (in (0,1)); threshold to use in tTMTI. If set to NULL, then either TMTI (default) or rtTMTI is used.
K	Integer; Number of smallest p-values to use in rtTMTI. If set to NULL, then either TMTI (default) or tTMTI is used.

Value

An approximation of the function $\gamma^m(x)$ under the assumption that all p-values are independent and exactly uniform

Examples

```
## Get an approximation of gamma
gamma_function <- gamma_bootstrapper(10)
## Evaluate it in a number, say .2
gamma_function(.2)
```

localTest_CTP	<i>A Closed Testing Procedure for any local test satisfying the conditions of Mogensen and Markussen (2021) using an $O(n^2)$ shortcut.</i>
---------------	--

Description

A Closed Testing Procedure for any local test satisfying the conditions of Mogensen and Markussen (2021) using an $O(n^2)$ shortcut.

Usage

```
localTest_CTP(localTest, pvals, alpha = 0.05, ...)
```

Arguments

localTest	A function which defines the choice of local test to use.
pvals	A vector of p-values
alpha	Level to perform each intersection test at. Defaults to 0.05
...	Additional arguments

Value

A data.frame containing: * p_adjust: The CTP adjusted p-value, controlling the FWER strongly. * Index: The original index of the unsorted p-value inputs.

Examples

```
## Simulate some p-values
## The first 10 are from false hypotheses, the next 10 are from true
pvals <- c (
  rbeta(10, 1, 20), ## Mean value of .05
  runif(10)
)
## Perform the CTP using a local Bonferroni test
localTest_CTP(function(x) {min(c(length(x) * min(x), 1))}, pvals)
```

make_Y	<i>Returns the transformed p-values to be used in the TMTI, tTMTI or rtTMTI. Internal function.</i>
--------	---

Description

Returns the transformed p-values to be used in the TMTI, tTMTI or rtTMTI. Internal function.

Usage

```
make_Y(pvals, tau = NULL, K = NULL, log.p = TRUE)
```

Arguments

pvals	A vector of p-values
tau	A numeric between 0 and 1 indicating the truncation level. Defaults to NULL
K	An integer > 1 indicating the the truncation index. Defaults to NULL
log.p	Logical, indicating whether transformations are on log scale. Defaults to TRUE.

Value

A vector, Y, of transformed p-values

Examples

```
## Simulate p-values
p <- runif(10)
make_Y(p) # Normal Y transformation
make_Y(p, tau = 0.5) # Using only the p-values below .5
make_Y(p, K = 5) # Using only the five smallest p-values.
```

rtTMTI_CDF	<i>Computes the analytical version of the rtMTI_infty CDF. When $m > 100$, this should not be used.</i>
------------	---

Description

Computes the analytical version of the rtTMTI_infty CDF. When $m > 100$, this should not be used.

Usage

```
rtTMTI_CDF(x, m, K)
```

Arguments

x	Point in which to evaluate the CDF
m	Number of independent tests to combine
K	Integer; the truncation point to use.

Value

The probability that the test statistic is at most x assuming independence under the global null hypothesis.

Examples

```
rtTMTI_CDF(0.05, 100, 10)
```

TestSet_localTest	<i>Test a subset of hypotheses in its closure using a user-specified local test</i>
-------------------	---

Description

Test a subset of hypotheses in its closure using a user-specified local test

Usage

```
TestSet_localTest(
  localTest,
  pvals,
  subset,
  alpha = 0.05,
  earlyStop = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

localTest	Function which defines a combination test.
pvals	Numeric vector of p-values
subset	Numeric vector; the subset to be tested
alpha	Numeric; the level to test at, if stopping early. Defaults to 0.05
earlyStop	Logical; set to TRUE to stop as soon as a hypothesis can be accepted at level alpha. This speeds up the procedure, but now only provides lower bounds on the p-values for the global test.
verbose	Logical; set to TRUE to print progress.
...	Additional arguments to be passed onto TMTI()

Value

The adjusted p-value for the test of the hypothesis that there are no false hypotheses among the selected subset.

Examples

```
## Simulate p-values; 10 from false hypotheses, 10 from true
pvals <- sort(c (
  rbeta(10, 1, 20), # Mean value of .1
  runif(10)
))
## Test whether the highest 10 contain any false hypotheses using a Bonferroni test
TestSet_localTest(function(x) {min(c(1, length(x) * min(x)))}, pvals, subset = 11:20)
```

TestSet_TMTI *Test a subset of hypotheses in its closure using the TMTI*

Description

Test a subset of hypotheses in its closure using the TMTI

Usage

```
TestSet_TMTI(
  pvals,
  subset,
  alpha = 0.05,
  tau = NULL,
  K = NULL,
  earlyStop = FALSE,
  verbose = FALSE,
  gammalist = NULL,
  ...
)
```

Arguments

pvals	Numeric vector of p-values
subset	Numeric vector; the subset to be tested
alpha	Numeric; the level to test at, if stopping early. Defaults to 0.05
tau	Numeric; the treshhold to use if using rTMTI. Set to NULL for TMTI or rTMTI. Defaults to NULL
K	Integer; The number of p-values to use if using rTMTI. Set to NULL for TMTI or tTMTI. Defaults to NULL.
earlyStop	Logical; set to TRUE to stop as soon as a hypothesis can be accepted at level alpha. This speeds up the procedure, but now only provides lower bounds on the p-values for the global test.
verbose	Logical; set to TRUE to print progress.
gammalist	List of functions. Must be such that the i'th element is the gamma function for sets of size i. Set to NULL to bootstrap the functions assuming independence. Defaults to NULL.
...	Additional arguments to be passed onto TMTI()

Value

The adjusted p-value for the test of the hypothesis that there are no false hypotheses among the selected subset.

Examples

```
## Simulate p-values; 10 from false hypotheses, 10 from true
pvals <- sort(c (
  rbeta(10, 1, 20), # Mean value of .1
  runif(10)
))
## Test whether the highest 10 contain any false hypotheses
TestSet_TMTI(pvals, subset = 11:20)
```

TMTI

*Functions for computing the TMTI tests***Description**

Functions for computing the TMTI tests

Usage

```
TMTI(
  pvals,
  n = Inf,
  tau = NULL,
  K = NULL,
  gamma = NULL,
  log.p = TRUE,
  B = 1000,
  m_max = 100,
  ...
)
```

Arguments

<code>pvals</code>	A vector of pvalues
<code>n</code>	A positive number (or <code>Inf</code>) indicating which type of local minimum to consider. Defaults to <code>Inf</code> , corresponding to the global minimum.
<code>tau</code>	Number between 0 and 1 or <code>NULL</code> , describing the truncation level.
<code>K</code>	Integer between >1 and <code>m</code> describing the truncation index.
<code>gamma</code>	Function; function to be used as the gamma approximation. If <code>NULL</code> , then the gamma function will be bootstrapped assuming independence. Defaults to <code>NULL</code>
<code>log.p</code>	Logical; indicating whether to compute <code>Y</code> 's on log-scale. Defaults to <code>TRUE</code>
<code>B</code>	Numeric; number of bootstrap replicates to be used when estimating the gamma function. If a gamma is supplied, this argument is ignored. Defaults to <code>1e3</code> .
<code>m_max</code>	Integer; the highest number of test for which the analytical computation of the TMTI CDF is used. When <code>m</code> is above <code>m_max</code> it will be bootstrapped or user supplied instead.
<code>...</code>	Additional parameters

Value

A p-value from the TMTI test

Examples

```
## Simulate some p-values
## The first 10 are from false hypotheses, the next 10 are from true
pvals <- c (
  rbeta(10, 1, 20), ## Mean value of .05
  runif(10)
)
TMTI(pvals)
```

TMTI_CDF	<i>Computes the analytical version of the TMTI_infty CDF. When $m > 100$, this should not be used.</i>
----------	--

Description

Computes the analytical version of the TMTI_infty CDF. When $m > 100$, this should not be used.

Usage

```
TMTI_CDF(x, m)
```

Arguments

x	Point in which to evaluate the CDF
m	Number of independent tests to combine

Value

The probability that the test statistic is at most x assuming independence under the global null hypothesis.

Examples

```
TMTI_CDF(0.05, 100)
```

TMTI_CTP

*A Closed Testing Procedure for the TMTI using an $O(n^2)$ shortcut***Description**

A Closed Testing Procedure for the TMTI using an $O(n^2)$ shortcut

Usage

```
TMTI_CTP(
  pvals,
  alpha = 0.05,
  B = 1000,
  gammaList = NULL,
  log.p = TRUE,
  tau = NULL,
  K = NULL,
  ...
)
```

Arguments

pvals	A vector of p-values
alpha	Level to perform each intersection test at. Defaults to 0.05
B	Number of bootstrap replications if gamma needs to be approximated. Not used if specifying a list of functions using the gammaList argument or if length(pvals) <= 100. Defaults to 1000
gammaList	A list of pre-specified gamma functions. If NULL, gamma functions will be approximated via bootstrap, assuming independence. Defaults to NULL.
log.p	Logical, indicating whether to compute Y's on log-scale. Defaults to TRUE
tau	Numerical (in (0,1)); threshold to use in tTMTI. If set to NULL, then either TMTI (default) or rtTMTI is used.
K	Integer; Number of smallest p-values to use in rtTMTI. If se to NULL, then either TMTI (default) or tTMTI is used.
...	Additional arguments

Value

A data.frame containing: * i: The sorted index of each p-value. * p_adjust: The CTP adjusted p-value, controlling the FWER strongly. * FirstAccept: The first level of the test tree at which the hypothesis could not be rejected. NA if it is never rejected. * Index: The original index of the unsorted p-value inputs.

Examples

```
## Simulate some p-values
## The first 10 are from false hypotheses, the next 10 are from true
pvals <- c (
  rbeta(10, 1, 20), ## Mean value of .05
  runif(10)
)
TMTI_CTP(pvals, earlyStop = TRUE)
```

TopDown_localTest	<i>TopDown localTest algorithm for estimating a 1-alpha confidence set for the number of false hypotheses among a set.</i>
-------------------	--

Description

TopDown localTest algorithm for estimating a 1-alpha confidence set for the number of false hypotheses among a set.

Usage

```
TopDown_localTest(
  localTest,
  pvals,
  subset = NULL,
  alpha = 0.05,
  verbose = TRUE,
  ...
)
```

Arguments

localTest	A function specifying a local test.
pvals	A vector of p-values
subset	Numeric vector specifying a subset p-values to estimate a confidence set for the number of false hypotheses for. Defaults to NULL corresponding to estimating a confidence set for the number of false hypotheses in the entire set.
alpha	Level in [0,1] at which to generate confidence set. Defaults to 0.05
verbose	Logical, indicating whether or not to write out the progress. Defaults to TRUE
...	Additional parameters

Value

A lower 1-alpha bound for the number of false hypotheses among the set of supplied p-values

Examples

```
## Simulate some p-values
## The first 10 are from false hypotheses, the next 10 are from true
pvals <- c (
  rbeta(10, 1, 20), ## Mean value of .05
  runif(10)
)
## Estimate the confidence set using a local Bonferroni test
TopDown_localTest(function(x) {min(c(1, length(x) * min(x)))}, pvals)
```

TopDown_TMTI

TopDown TMTI algorithm for estimating a 1-alpha confidence set for the number of false hypotheses among a set.

Description

TopDown TMTI algorithm for estimating a 1-alpha confidence set for the number of false hypotheses among a set.

Usage

```
TopDown_TMTI(
  pvals,
  subset = NULL,
  alpha = 0.05,
  gammaList = NULL,
  verbose = TRUE,
  log.p = TRUE,
  tau = NULL,
  K = NULL,
  ...
)
```

Arguments

pvals	A vector of p-values
subset	Numeric vector specifying a subset p-values to estimate a confidence set for the number of false hypotheses for. Defaults to NULL corresponding to estimating a confidence set for the number of false hypotheses in the entire set.
alpha	Level in [0,1] at which to generate confidence set. Defaults to 0.05
gammaList	List of pre-specified gamma functions. If NULL, the functions will be approximated by bootstrap assuming independence. Defaults to NULL
verbose	Logical, indicating whether or not to write out the progress. Defaults to TRUE
log.p	Logical, indicating whether to calculate Ys on a log scale. Defaults to TRUE
tau	Numerical (in (0,1)); threshold to use in tTMTI. If set to NULL, then either TMTI (default) or rtTMTI is used.

K Integer; Number of smallest p-values to use in rtTMTI. If se to NULL, then either TMTI (default) or tTMTI is used.

... Additional parameters

Value

A lower 1-alpha bound for the number of false hypotheses among the set of supplied p-values

Examples

```
## Simulate some p-values
## The first 10 are from false hypotheses, the next 10 are from true
pvals <- c (
  rbeta(10, 1, 20), ## Mean value of .05
  runif(10)
)
TopDown_TMTI(pvals)
```

tTMTI_CDF	<i>Computes the analytical version of the tTMTI_infty CDF. When m>100, this should not be used.</i>
-----------	--

Description

Computes the analytical version of the tTMTI_infty CDF. When m>100, this should not be used.

Usage

```
tTMTI_CDF(x, m, tau)
```

Arguments

x Point in which to evaluate the CDF

m Number of independent tests to combine

tau The truncation point of the tTMTI procedure

Value

The probability that the test statistic is at most x assuming independence under the global null hypothesis.

Examples

```
tTMTI_CDF(0.05, 100, 0.05)
```

Index

.GetMinima, [2](#)

gamma_bootstrapper, [2](#)

localTest_CTP, [3](#)

make_Y, [4](#)

rtTMTI_CDF, [5](#)

TestSet_localTest, [6](#)

TestSet_TMTI, [7](#)

TMTI, [8](#)

TMTI_CDF, [9](#)

TMTI_CTP, [10](#)

TopDown_localTest, [11](#)

TopDown_TMTI, [12](#)

tTMTI_CDF, [13](#)