

# Package ‘gMCPmini’

March 3, 2022

**Type** Package

**Title** Graph Based Multiple Comparison Procedures (Mini Version)

**Version** 0.8-15

**Maintainer** Kornelius Rohmeyer <rohrmeyer@small-projects.de>

**Description**

Selected Functions From gMCP package to support gsDesign and remove JAVA dependency. The package version is the same as the gMCP package version.

**Depends** R (>= 3.0.0),  
methods

**License** GPL (>= 2)

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**Suggests** testthat (>= 3.0.0),  
stats

**Config/testthat/edition** 3

## R topics documented:

BonferroniHolm . . . . .	1
gMCP . . . . .	2
hGraph . . . . .	4
matrix2graph . . . . .	7
rejectNode . . . . .	8
substituteEps . . . . .	9

<b>Index</b>	<b>10</b>
--------------	-----------

---

BonferroniHolm	<i>Example Graph</i>
----------------	----------------------

---

**Description**

Example Graph

**Usage**

BonferroniHolm(n, weights = rep(1/n, n))

gMCP

*Graph based Multiple Comparison Procedures***Description**

Performs a graph based multiple test procedure for a given graph and unadjusted p-values.

**Usage**

```
gMCP(
  graph,
  pvalues,
  test,
  correlation,
  alpha = 0.05,
  approxEps = TRUE,
  eps = 10^(-3),
  ...,
  upscale = ifelse(missing(test) && !missing(correlation) || !missing(test) && test ==
    "Bretz2011", TRUE, FALSE),
  useC = FALSE,
  verbose = FALSE,
  keepWeights = FALSE,
  adjPValues = TRUE
)
```

**Arguments**

graph	A graph of class <a href="#">graphMCP</a> .
pvalues	A numeric vector specifying the p-values for the graph based MCP. Note the assumptions in the details section for the parametric tests, when a correlation is specified.
test	Should be either "Bonferroni", "Simes" or "parametric". If not specified by default the Bonferroni-based test procedure is used if no correlation is specified or the algorithm from Bretz et al. 2011 if a correlation is specified. If test is set to "Simes" the weighted Simes test will be performed for each subset of hypotheses.
correlation	Optional correlation matrix. If the weighted Simes test is performed, it is checked whether type I error rate can be ensured and a warning is given if this is not the case. For parametric tests the p-values must arise from one-sided tests with multivariate normal distributed test statistics for which the correlation is (partially) known. In that case a weighted parametric closed test is performed (also see <a href="#">generatePvals</a> ). Unknown values can be set to NA. (See details for more information)
alpha	A numeric specifying the maximal allowed type one error rate.
approxEps	A boolean specifying whether epsilon values should be substituted with the value given in the parameter eps.
eps	A numeric scalar specifying a value for epsilon edges.
...	Test specific arguments can be given here.

upscale	Logical. If upscale=FALSE then for each intersection of hypotheses (i.e. each subgraph) a weighted test is performed at the possibly reduced level alpha of $\text{sum}(w) \cdot \alpha$ , where $\text{sum}(w)$ is the sum of all node weights in this subset. If upscale=TRUE all weights are upscaled, so that $\text{sum}(w)=1$ . For backward compatibility the default value is TRUE if a the parameter test is missing, but parameter correlation is specified or if <code>test=="Bretz2011"</code> .
useC	Logical scalar. If TRUE neither adjusted p-values nor intermediate graphs are returned, but the calculation is sped up by using code written in C. THIS CODE IS NOT FOR PRODUCTIVE USE YET! If approxEps is FALSE and the graph contains epsilon edges, a warning is thrown and useC will be ignored.
verbose	Logical scalar. If TRUE verbose output is generated.
keepWeights	Logical scalar. If FALSE the weight of a node without outgoing edges is set to 0 if it is removed. Otherwise it keeps its weight.
adjPValues	Logical scalar. If FALSE no adjusted p-values will be calculated. Especially for the weighted Simes test this will result in significantly less calculations in most cases.

## Details

For the Bonferroni procedure the p-values can arise from any statistical test, but if you improve the test by specifying a correlation matrix, the following assumptions apply:

It is assumed that under the global null hypothesis  $(\Phi^{-1}(1 - p_1), \dots, \Phi^{-1}(1 - p_m))$  follow a multivariate normal distribution with correlation matrix `correlation` where  $\Phi^{-1}$  denotes the inverse of the standard normal distribution function.

For example, this is the case if  $p_1, \dots, p_m$  are the raw p-values from one-sided z-tests for each of the elementary hypotheses where the correlation between z-test statistics is generated by an overlap in the observations (e.g. comparison with a common control, group-sequential analyses etc.). An application of the transformation  $\Phi^{-1}(1 - p_i)$  to raw p-values from a two-sided test will not in general lead to a multivariate normal distribution. Partial knowledge of the correlation matrix is supported. The correlation matrix has to be passed as a numeric matrix with elements of the form: `correlation[i, i] = 1` for diagonal elements, `correlation[i, j] =  $\rho_{ij}$` , where  $\rho_{ij}$  is the known value of the correlation between  $\Phi^{-1}(1 - p_i)$  and  $\Phi^{-1}(1 - p_j)$  or NA if the corresponding correlation is unknown. For example `correlation[1,2]=0` indicates that the first and second test statistic are uncorrelated, whereas `correlation[2,3] = NA` means that the true correlation between statistics two and three is unknown and may take values between -1 and 1. The correlation has to be specified for complete blocks (ie.: if `cor(i,j)`, and `cor(i,j')` for  $i \neq j \neq j'$  are specified then `cor(j,j')` has to be specified as well) otherwise the corresponding intersection null hypotheses tests are not uniquely defined and an error is returned.

For further details see the given references.

## Value

An object of class `gMCPResult`, more specifically a list with elements

`graphs` list of graphs

`pvalues` p-values

`rejected` logical whether hypotheses could be rejected

`adjPValues` adjusted p-values

**Author(s)**

Kornelius Rohmeyer <rohmeyer@small-projects.de>

**References**

Frank Bretz, Willi Maurer, Werner Brannath, Martin Posch: A graphical approach to sequentially rejective multiple test procedures. *Statistics in Medicine* 2009 vol. 28 issue 4 page 586-604. [http://www.meduniwien.ac.at/fwf\\_adaptive/papers/bretz\\_2009\\_22.pdf](http://www.meduniwien.ac.at/fwf_adaptive/papers/bretz_2009_22.pdf)

Bretz F., Posch M., Glimm E., Klinglmueller F., Maurer W., Rohmeyer K. (2011): Graphical approaches for multiple endpoint problems using weighted Bonferroni, Simes or parametric tests. *Biometrical Journal* 53 (6), pages 894-913, Wiley. <http://onlinelibrary.wiley.com/doi/10.1002/bimj.201000239/full>

Strassburger K., Bretz F.: Compatible simultaneous lower confidence bounds for the Holm procedure and other Bonferroni based closed tests. *Statistics in Medicine* 2008; 27:4914-4927.

Hommel G., Bretz F., Maurer W.: Powerful short-cuts for multiple testing procedures with special reference to gatekeeping strategies. *Statistics in Medicine* 2007; 26:4063-4073.

Guilbaud O.: Simultaneous confidence regions corresponding to Holm's stepdown procedure and other closed-testing procedures. *Biometrical Journal* 2008; 50:678-692.

**See Also**

[graphMCP](#) [graphNEL](#)

**Examples**

```
g <- BonferroniHolm(5)
gMCP(g, pvalues=c(0.01, 0.02, 0.04, 0.04, 0.7))
# Simple Bonferroni with empty graph:
g2 <- matrix2graph(matrix(0, nrow=5, ncol=5))
gMCP(g2, pvalues=c(0.01, 0.02, 0.04, 0.04, 0.7))
# With 'upscale=TRUE' equal to BonferroniHolm:
gMCP(g2, pvalues=c(0.01, 0.02, 0.04, 0.04, 0.7), upscale=TRUE)
```

---

hGraph

---

*Create multiplicity graphs using ggplot2*


---

**Description**

hGraph() plots a multiplicity graph defined by user inputs. The graph can also be used with the **\*\*\*gMCP\*\*\*** package to evaluate a set of nominal p-values for the tests of the hypotheses in the graph

**Usage**

```

hGraph(
  nHypotheses = 4,
  nameHypotheses = paste("H", (1:nHypotheses), sep = ""),
  alphaHypotheses = 0.025/nHypotheses,
  m = matrix(array(1/(nHypotheses - 1), nHypotheses^2), nrow = nHypotheses) -
    diag(1/(nHypotheses - 1), nHypotheses),
  fill = 1,
  palette = grDevices::gray.colors(length(unique(fill)), start = 0.5, end = 0.8),
  labels = LETTERS[1:length(unique(fill))],
  legend.name = " ",
  legend.position = "none",
  halfWid = 0.5,
  halfHgt = 0.5,
  trhw = 0.1,
  trhh = 0.075,
  trprop = 1/3,
  digits = 5,
  trdigits = 2,
  size = 6,
  boxtextsize = 4,
  arrowsize = 0.02,
  radianStart = if ((nHypotheses)%2 != 0) { pi * (1/2 + 1/nHypotheses) } else {
    pi * (1 + 2/nHypotheses)/2 },
  offset = pi/4/nHypotheses,
  xradius = 2,
  yradius = xradius,
  x = NULL,
  y = NULL,
  wchar = if (as.character(Sys.info()[1]) == "Windows") { "w" } else { "w" }
)

```

**Arguments**

nHypotheses	number of hypotheses in graph
nameHypotheses	hypothesis names
alphaHypotheses	alpha-levels or weights for ellipses
m	square transition matrix of dimension 'nHypotheses'
fill	grouping variable for hypotheses
palette	colors for groups
labels	text labels for groups
legend.name	text for legend header
legend.position	text string or x,y coordinates for legend
halfWid	half width of ellipses
halfHgt	half height of ellipses
trhw	transition box width
trhh	transition box height

trprop	proportion of transition arrow length where transition box is placed
digits	number of digits to show for alphaHypotheses
trdigits	digits displayed for transition weights
size	text size in ellipses
boxtextsize	transition text size
arrowsize	size of arrowhead for transition arrows
radianStart	radians from origin for first ellipse; nodes spaced equally in clockwise order with centers on an ellipse by default
offset	rotational offset in radians for transition weight arrows
xradius	horizontal ellipse diameter on which ellipses are drawn
yradius	vertical ellipse diameter on which ellipses are drawn
x	x coordinates for hypothesis ellipses if elliptical arrangement is not wanted
y	y coordinates for hypothesis ellipses if elliptical arrangement is not wanted
wchar	character for alphaHypotheses in ellipses

### Details

See vignette **\*\*Multiplicity graphs formatting using ggplot2\*\*** for explanation of formatting.

### Value

A 'ggplot' object with a multi-layer multiplicity graph

### Examples

```
library(tidyr)
# Defaults: note clockwise ordering
hGraph(5)
# Add colors (default is 3 gray shades)
hGraph(3,fill=1:3)
# Colorblind palette
cbPalette <- c("#999999", "#E69F00", "#56B4E9", "#009E73",
               "#F0E442", "#0072B2", "#D55E00", "#CC79A7")
hGraph(6,fill=as.factor(1:6),palette=cbPalette)
# Use a hue palette
hGraph(4,fill=factor(1:4),palette=scales::hue_pal(l=75)(4))
# different alpha allocation, hypothesis names and transitions
alphaHypotheses <- c(.005,.007,.013)
nameHypotheses <- c("ORR","PFS","OS")
m <- matrix(c(0,1,0,
              0,0,1,
              1,0,0),nrow=3,byrow=TRUE)
hGraph(3,alphaHypotheses=alphaHypotheses,nameHypotheses=nameHypotheses,m=m)
# Custom position and size of ellipses, change text to multi-line text
# Adjust box width
# add legend in middle of plot
hGraph(3,x=sqrt(0:2),y=c(1,3,1.5),size=6,halfWid=.3,halfHgt=.3, trhw=0.6,
       palette=cbPalette[2:4], fill = c(1, 2, 2),
       legend.position = c(.6,.5), legend.name = "Legend:", labels = c("Group 1", "Group 2"),
       nameHypotheses=c("H1:\n Long name","H2:\n Longer name","H3:\n Longest name"))
```

---

matrix2graph*Matrix2Graph and Graph2Matrix*

---

## Description

Creates a graph of class `graphMCP` from a given transition matrix or vice versa.

## Usage

```
matrix2graph(m, weights = rep(1/dim(m)[1], dim(m)[1]))
```

```
graph2matrix(graph)
```

## Arguments

<code>m</code>	A transition matrix.
<code>weights</code>	A numeric for the initial weights.
<code>graph</code>	A graph of class <code>graphMCP</code> .

## Details

The hypotheses names are the row names or if these are NULL, the column names or if these are also NULL of type H1, H2, H3, ...

If the diagonal of the matrix is unequal zero, the values are ignored and a warning is given.

## Value

A graph of class `graphMCP` with the given transition matrix for `matrix2graph`. The transition matrix of a `graphMCP` graph for `graph2matrix`.

## Author(s)

Kornelius Rohmeyer <rohrmeyer@small-projects.de>

## Examples

```
# Bonferroni-Holm:
m <- matrix(rep(1/3, 16), nrow=4)
diag(m) <- c(0, 0, 0, 0)
graph <- matrix2graph(m)
print(graph)
graph2matrix(graph)
```

---

rejectNode	<i>Rejects a node/hypothesis and updates the graph accordingly.</i>
------------	---

---

### Description

Rejects a node/hypothesis and updates the graph accordingly.

### Usage

```
rejectNode(graph, node, upscale = FALSE, verbose = FALSE, keepWeights = FALSE)
```

### Arguments

graph	A graph of class <code>graphMCP</code> or <code>entangledMCP</code> .
node	A character string specifying the node to reject.
upscale	Logical. If <code>upscale=TRUE</code> then the weights of all non-rejected nodes are scaled so that the sum is equal to 1. This forces <code>keepWeights=FALSE</code> to reduce confusion, since otherwise the sum of weights could become bigger than 1.
verbose	Logical scalar. If <code>TRUE</code> verbose output is generated during sequentially rejection steps.
keepWeights	Logical scalar. If <code>FALSE</code> the weight of a node without outgoing edges is set to 0 if it is removed. Otherwise it keeps its weight.

### Details

For details see the given references.

### Value

An updated graph of class `graphMCP` or `entangledMCP`.

### Author(s)

Kornelius Rohmeyer <rohmeier@small-projects.de>

### References

Frank Bretz, Willi Maurer, Werner Brannath, Martin Posch: A graphical approach to sequentially rejective multiple test procedures. Statistics in Medicine 2009 vol. 28 issue 4 page 586-604. [http://www.meduniwien.ac.at/fwf\\_adaptive/papers/bretz\\_2009\\_22.pdf](http://www.meduniwien.ac.at/fwf_adaptive/papers/bretz_2009_22.pdf)

### See Also

`graphMCP`



---

substituteEps	<i>Substitute Epsilon</i>
---------------	---------------------------

---

**Description**

Substitute Epsilon with a given value.

**Usage**

```
substituteEps(graph, eps = 10^(-3))
```

**Arguments**

graph	A graph of class <a href="#">graphMCP</a> or class <a href="#">entangledMCP</a> .
eps	A numeric scalar specifying a value for epsilon edges.

**Details**

For details see the given references.

**Value**

A graph where all epsilons have been replaced with the given value.

**Author(s)**

Kornelius Rohmeyer <[rohrmeyer@small-projects.de](mailto:rohrmeyer@small-projects.de)>

**See Also**

[graphMCP](#), [entangledMCP](#)

# Index

- \* **graphs**
  - gMCP, [2](#)
  - matrix2graph, [7](#)
  - rejectNode, [8](#)
  - substituteEps, [9](#)
- \* **htest**
  - gMCP, [2](#)
  - rejectNode, [8](#)
- \* **print**
  - substituteEps, [9](#)
- BonferroniHolm, [1](#)
- entangledMCP, [8](#), [9](#)
- generatePvals, [2](#)
- gMCP, [2](#)
- graph2matrix (matrix2graph), [7](#)
- graphMCP, [2](#), [4](#), [7–9](#)
- graphNEL, [4](#)
- hGraph, [4](#)
- matrix2graph, [7](#)
- rejectNode, [8](#)
- substituteEps, [9](#)