

gMCP - an R package for a graphical approach to weighted multiple test procedures

Kornelius Rohmeyer

February 17, 2012

Contents

1	Introduction	3
1.1	Installation	3
1.2	Example and diving in	3
2	Basic Theoretical Background	5
3	Creating Weighted Graphs	5
3.1	Using R	5
3.2	Using the GUI	7
4	The sequentially rejective MTP	8
4.1	Using R	8
4.1.1	Adjusted p-values and simultaneous confidence intervals	9
4.2	Using the GUI	10
5	Weighted parametric tests	10
6	Correlation matrix creation	10
7	Epsilon edges	11
8	Power Simulations	12
8.1	Variable edge weights	13
9	Options and Import/Export	13
9.1	Options	13
9.1.1	Privacy	15
9.2	Import/Exports	15
9.3	Important TikZ commands for optimizing the reports	16
10	Case Studies	17
10.1	Identifying effective and/or safe doses by stepwise confidence intervals for ratios	17
10.2	Testing strategies in multi-dose experiments including active control	17
A	Appendix - Multiple Testing Basics	17
A.1	Closed testing principle	18
A.2	Partitioning principle	18
	Index	20
	Literatur	22
	Table of Symbols	22

1 Introduction

This package provides functions and graphical user interfaces for graph based multiple test procedures. By defining weighted graphs one also defines a weighting strategy for all subsets of null hypotheses. Weighted tests can be performed on these subsets and following the closed test procedure this leads to a multiple test procedure controlling the family wise error rate in the strong sense. In some cases shortcuts are available, for example the weighted Bonferroni procedure leads to a sequentially rejective multiple test procedure. At all steps either graphical user interfaces or the R Console with S4 objects and methods can be used. Please note that this is still an early release and the API will most likely still change in future versions.

1.1 Installation

Open R and type `install.packages("gMCP")`, select an arbitrary mirror and gMCP will be downloaded and installed.

Once it is installed, whenever you start R you can load the gMCP package by entering `library(gMCP)` into the R Console. The graphical user interface is started with the command `graphGUI()`.

If you run into problems, see <http://cran.r-project.org/web/packages/gMCP/INSTALL> or write us an email at help@small-projects.de.

1.2 Example and diving in

Let's start with a well-known procedure and see how it fits into this graphical approach to weighted multiple test procedures: The Bonferroni-Holm-Procedure [6].

Theorem 1.1 (Bonferroni-Holm-Procedure). *Let T_1, \dots, T_m be test statistics for $m \in \mathbb{N}$ null hypotheses H_1, \dots, H_m and p_1, \dots, p_m the associated p -values. Then the following test will control the familywise error rate at level $\alpha \in]0, 1[$ in the strong sense:*

Denote the ordered p -values by $p^{(1)} < p^{(2)} < \dots < p^{(m)}$ and the corresponding hypotheses by $H^{(1)}, H^{(2)}, \dots, H^{(m)}$. Reject $H^{(1)}, H^{(2)}, \dots, H^{(j)}$ such that

$$p^{(i)} \leq \frac{\alpha}{n - i + 1} \quad \text{for all } 1 \leq i \leq j.$$

The corresponding graph for the Bonferroni-Holm-Procedure for three hypotheses is given in Figure 1. We see a fully connected graph, where each node represents a hypothesis and the nodes and edges have weights.

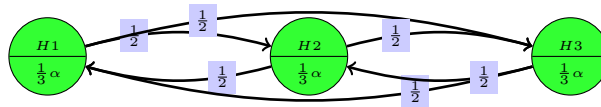


Figure 1: Graph representing the Bonferroni-Holm-Procedure for three hypotheses.

A null hypothesis can be rejected, when the p -value is less than the alpha level of the corresponding node. In this case the graph will be updated and the alpha level of this node is passed according to the edge weights.

Example 1.2. We give an example for the Bonferroni-Holm-Procedure that will be used repeatedly throughout this manual. Of course this package is made for more advanced tests (you find a selection in section 10), but since most readers are already familiar with this procedure, for a first introduction of gMCP, we stick to this simple example.

Let $p_1 = 0.01$, $p_2 = 0.07$ and $p_3 = 0.02$ be three p-values and $\alpha = 0.05$. In the first step H_1 can be rejected since $p_1 < \alpha/3$. The updated graph can be seen in figure 2 and now also H_3 can be rejected since $p_1 < \alpha/2$. Again the graph is updated, but H_2 can not be rejected.

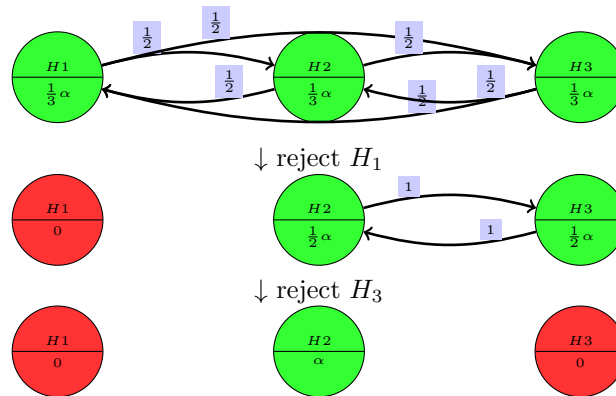


Figure 2: Example showing how two null hypotheses can be rejected with p-values $p_1 = 0.01$, $p_2 = 0.07$ and $p_3 = 0.02$.

Let's reproduce this with the gMCP package. We start R and enter:

```
> library(gMCP)
> graphGUI()
```

The GUI seen in Figure 4 is shown and we select from the menu "Example graphs" the entry "Bonferroni-Holm Test". We enter the three p-values in the respective fields on the right side. By clicking on the button with the green arrow we start the test procedure and can sequentially reject all three hypotheses.

If we don't want to use the GUI we can also use R:

```
> library(gMCP)
> graph <- BonferroniHolm(3)
> gMCP(graph, pvalues=c(0.01,0.07,0.02), alpha=0.05)
```

gMCP-Result

Initial graph:

A graphMCP graph

H1 (not rejected, weight=0.3333)

H2 (not rejected, weight=0.3333)

H3 (not rejected, weight=0.3333)

Edges:

H1 -(0.5)-> H2

H1 -(0.5)-> H3

H2 -(0.5)-> H1

H2 -(0.5)-> H3

H3 -(0.5)-> H1

H3 -(0.5)-> H2

P-values:

H1 H2 H3

0.01 0.07 0.02

Adjusted p-values:

H1 H2 H3

0.03 0.07 0.04

Alpha: 0.05

Hypothesis rejected:

H1 H2 H3

TRUE FALSE TRUE

Final graph after 2 steps:
A graphMCP graph
H1 (rejected, weight=0)
H2 (not rejected, weight=1)
H3 (rejected, weight=0)
No edges.

2 Basic Theoretical Background

Algorithm 1 Removing node i , passing the weight and updating the graph edges

```

for  $l \in I$  do
   $w_l \leftarrow w_l + w_i \cdot g_{il}$ 
  for  $k \in I$  do
    if  $l \neq k$  and  $g_{lj} \cdot g_{jk} \neq 1$  then
       $g_{lk} \leftarrow \frac{g_{lk} + g_{lj} \cdot g_{jk}}{1 - g_{lj} \cdot g_{jk}}$ 
    else
       $g_{lk} \leftarrow 0$ 
    end if
  end for
end for
end for

```

3 Creating Weighted Graphs

In the first step a graph that describes the multiple test procedures must be created.

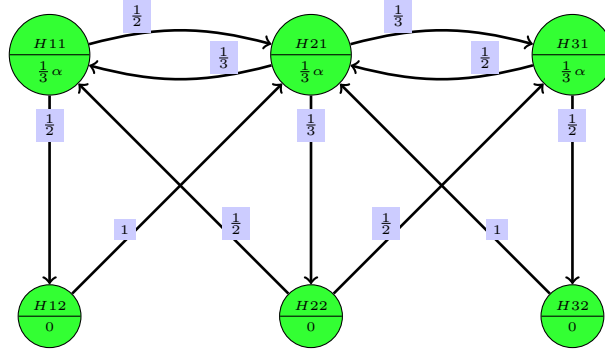


Figure 3: Example graph from [4] that we will create in this vignette.

3.1 Using R

The most convenient way to create a graph in R is to use the functions `matrix2graph` and `setWeights`. As an example we create the graph from Bretz et al. [4] that you can see in figure 3.

```

> m <- rbind(H11=c(0, 0.5, 0, 0.5, 0, 0),
+           H21=c(1/3, 0, 1/3, 0, 1/3, 0),
+           H31=c(0, 0.5, 0, 0, 0, 0.5),
+           H12=c(0, 1, 0, 0, 0, 0),
+           H22=c(0.5, 0, 0.5, 0, 0, 0),
+           H32=c(0, 1, 0, 0, 0, 0))
> graph <- matrix2graph(m)
> graph <- setWeights(graph, c(1/3, 1/3, 1/3, 0, 0, 0))

```

For accessing the weights and adjacency matrix of an existing graph the functions `getWeights` and `getMatrix` are provided.

Let's print the newly created graph:

```
> print(graph)
A graphMCP graph
H11 (not rejected, weight=0.3333)
H21 (not rejected, weight=0.3333)
H31 (not rejected, weight=0.3333)
H12 (not rejected, weight=0)
H22 (not rejected, weight=0)
H32 (not rejected, weight=0)
Edges:
H11 -( 0.5 )-> H21
H11 -( 0.5 )-> H12
H21 -( 0.3333333333333333 )-> H11
H21 -( 0.3333333333333333 )-> H31
H21 -( 0.3333333333333333 )-> H22
H31 -( 0.5 )-> H21
H31 -( 0.5 )-> H32
H12 -( 1 )-> H21
H22 -( 0.5 )-> H11
H22 -( 0.5 )-> H31
H32 -( 1 )-> H21
```

Since we also want to visualize the graph, we set two node attributes `X` and `Y` (for further information see the manual pages of method `nodeAttr`).

```
> graph@nodeAttr$X <- c(H11=100, H21=300, H31=500, H12=100, H22=300, H32=500)
> graph@nodeAttr$Y <- c(H11=100, H21=100, H31=100, H12=300, H22=300, H32=300)
```

For placement of the nodes in a matrix pattern, the function `placeNodes` is helpful. The following code does the same as the three lines of R code above.

```
> graph <- placeNodes(graph, nrow=2)
```

Coordinates are interpreted as pixels in the GUI and big points in \LaTeX (72 bp = 1 inch).

Let's take a look at the graph in \LaTeX rendered with TikZ [8] (you can see the compiled result in figure 3):

```
> cat(graph2latex(graph))
\begin{tikzpicture}[scale=1]
\node (H11) at (100bp,-100bp)[draw,circle split,fill=green!80] {$H11$ \nodepart{lower}  $\frac{1}{3}\alpha$ };
\node (H21) at (300bp,-100bp)[draw,circle split,fill=green!80] {$H21$ \nodepart{lower}  $\frac{1}{3}\alpha$ };
\node (H31) at (500bp,-100bp)[draw,circle split,fill=green!80] {$H31$ \nodepart{lower}  $\frac{1}{3}\alpha$ };
\node (H12) at (100bp,-300bp)[draw,circle split,fill=green!80] {$H12$ \nodepart{lower}  $0$ };
\node (H22) at (300bp,-300bp)[draw,circle split,fill=green!80] {$H22$ \nodepart{lower}  $0$ };
\node (H32) at (500bp,-300bp)[draw,circle split,fill=green!80] {$H32$ \nodepart{lower}  $0$ };
\draw [->,line width=1pt] (H11) to[bend left=15] node[near start,above,fill=blue!20] {$\frac{1}{2}$} (H21);
\draw [->,line width=1pt] (H11) to[auto] node[near start,above,fill=blue!20] {$\frac{1}{2}$} (H12);
\draw [->,line width=1pt] (H21) to[bend left=15] node[near start,above,fill=blue!20] {$\frac{1}{3}$} (H11);
\draw [->,line width=1pt] (H21) to[bend left=15] node[near start,above,fill=blue!20] {$\frac{1}{3}$} (H31);
\draw [->,line width=1pt] (H21) to[auto] node[near start,above,fill=blue!20] {$\frac{1}{3}$} (H22);
\draw [->,line width=1pt] (H31) to[bend left=15] node[near start,above,fill=blue!20] {$\frac{1}{2}$} (H21);
\draw [->,line width=1pt] (H31) to[auto] node[near start,above,fill=blue!20] {$\frac{1}{2}$} (H32);
\draw [->,line width=1pt] (H12) to[auto] node[near start,above,fill=blue!20] {$1$} (H21);
\draw [->,line width=1pt] (H22) to[auto] node[near start,above,fill=blue!20] {$\frac{1}{2}$} (H11);
\draw [->,line width=1pt] (H22) to[auto] node[near start,above,fill=blue!20] {$\frac{1}{2}$} (H31);
\draw [->,line width=1pt] (H32) to[auto] node[near start,above,fill=blue!20] {$1$} (H21);
\end{tikzpicture}
```

We can even change the position of the edge labels for further fine tuning of the graphical representation. With the following command we place the label for the edge from H1 to H2 at position (200, 80):

```
> edgeAttr(graph, "H11", "H21", "labelX") <- 200
> edgeAttr(graph, "H11", "H21", "labelY") <- 80
```

3.2 Using the GUI

The creation of **graphMCP** objects as seen in the last section with basic R commands is very straight forward, but still takes some time and typos may occur. More convenient for the most users is the use of the graphical user interface for creating and editing MCP graphs that the **gMCP** package includes.

It is called by the command **graphGUI()** and takes as optional argument a variable name, given as a character string, of the graph to edit.

```
> graphGUI("graph")
```

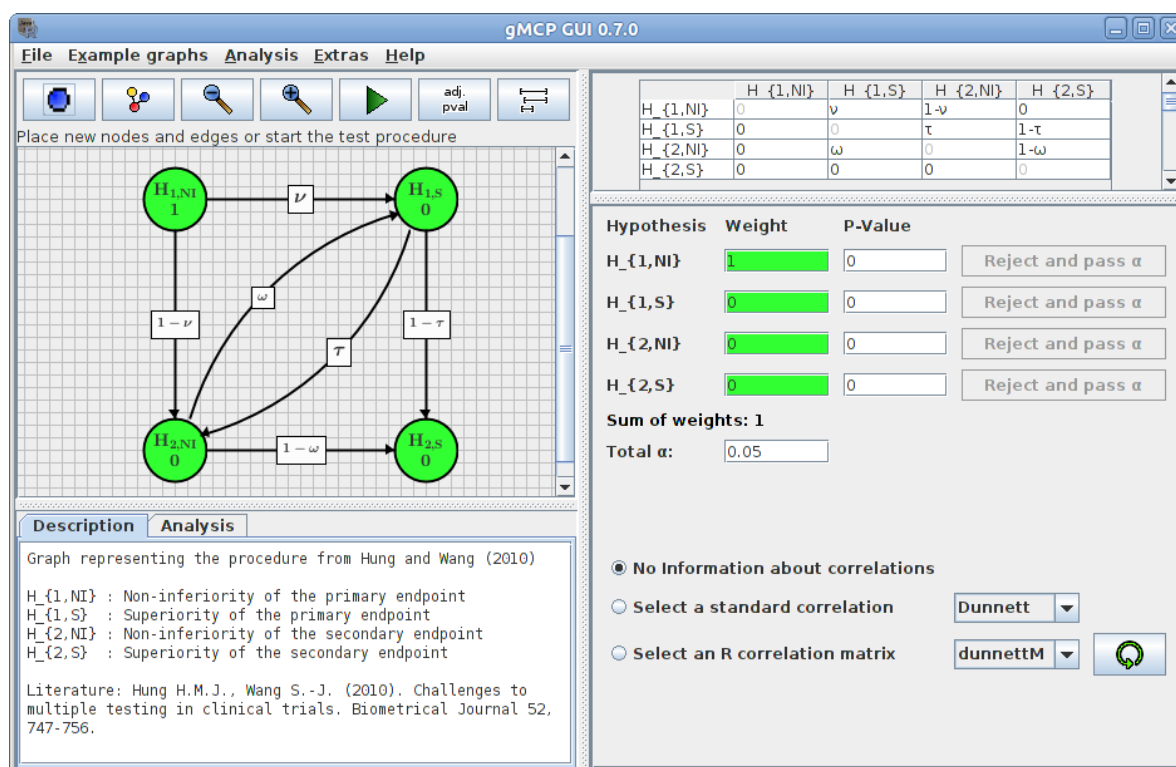


Figure 4: The graphical user interface allows testing, calculation of confidence intervals and adjusted p-values.

Let's take a look at the icon panel:



This button lets you add a new node to the graph. After pressing the button click somewhere on the graph panel and a new node will appear at this place.



This button lets you add a new edge between two nodes. After pressing the button click on the node the edge should start and after that on the node the edge should end.



For really big graphs the ability to zoom in and out is usefull.



Starts the testing procedure / goes back to the graph modification.



Calculates the adjusted p-values.



Calculates simultaneous confidence intervals.

With drag and drop you can move nodes and also adjust edges.

4 The sequentially rejective MTP

For a full description of the sequentially rejective multiple testing procedure take a look at Bretz et al. [3].

4.1 Using R

You can either specify each rejection step yourself or simply use the method `gMCP`:

```
> graph <- BretzEtAl2011()
> # We can reject a single node:
> print(rejectNode(graph, "H11"))

A graphMCP graph
H11 (rejected, weight=0)
H21 (not rejected, weight=0.5)
H31 (not rejected, weight=0.3333)
H12 (not rejected, weight=0.1667)
H22 (not rejected, weight=0)
H32 (not rejected, weight=0)
Edges:
H21 -( 0.4 )-> H31
H21 -( 0.2 )-> H12
H21 -( 0.4 )-> H22
H31 -( 0.5 )-> H21
H31 -( 0.5 )-> H32
H12 -( 1 )-> H21
H22 -( 0.25 )-> H21
H22 -( 0.5 )-> H31
H22 -( 0.25 )-> H12
H32 -( 1 )-> H21

> # Or given a vector of pvalues let the function gMCP do all the work:
> pvalues <- c(0.1, 0.008, 0.005, 0.15, 0.04, 0.006)
> result <- gMCP(graph, pvalues)
> print(result)

gMCP-Result

Initial graph:
A graphMCP graph
H11 (not rejected, weight=0.3333)
H21 (not rejected, weight=0.3333)
H31 (not rejected, weight=0.3333)
H12 (not rejected, weight=0)
H22 (not rejected, weight=0)
H32 (not rejected, weight=0)
Edges:
H11 -( 0.5 )-> H21
H11 -( 0.5 )-> H12
H21 -( 0.3333333333333333 )-> H11
H21 -( 0.3333333333333333 )-> H31
H21 -( 0.3333333333333333 )-> H22
H31 -( 0.5 )-> H21
H31 -( 0.5 )-> H32
H12 -( 1 )-> H21
H22 -( 0.5 )-> H11
H22 -( 0.5 )-> H31
H32 -( 1 )-> H21

P-values:
  H11  H21  H31  H12  H22  H32
0.100 0.008 0.005 0.150 0.040 0.006

Adjusted p-values:
  H11  H21  H31  H12  H22  H32
0.1200 0.0160 0.0150 0.1500 0.1200 0.0225

Alpha: 0.05
```



```
Hypothesis rejected:
  H11  H21  H31  H12  H22  H32
FALSE TRUE TRUE FALSE FALSE TRUE
```

```
Final graph after 3 steps:
A graphMCP graph
H11 (not rejected, weight=0.6667)
H21 (rejected, weight=0)
H31 (rejected, weight=0)
H12 (not rejected, weight=0)
H22 (not rejected, weight=0.3333)
H32 (rejected, weight=0)
Edges:
H11 -( 0.6666666666666667 )-> H12
H11 -( 0.3333333333333333 )-> H22
H12 -( 0.5 )-> H11
H12 -( 0.5 )-> H22
H22 -( 1 )-> H11
```

We can create a TikZ graphic from the last graph with `graph2latex(result@graphs[[4]])` that is shown in figure 5.

Figure 5: Final graph from the test procedure after rejection of H_{21} , H_{31} and H_{32} .

The command `gMCPReport` generates a full report of the testing procedure:

```
> gMCPReport(result, "Report.tex")
```

4.1.1 Adjusted p-values and simultaneous confidence intervals

Also adjusted p-values and simultaneous confidence intervals can be computed.

Let's assume the tests for hypotheses $H1 : \theta_1 \leq 0$, $H2 : \theta_2 \leq 0$ and $H3 : \theta_3 \leq 0$ are three t-tests with degree of freedom 9. The estimates are $\hat{\theta}_1 = 0.981$, $\hat{\theta}_2 = 1.089$ and $\hat{\theta}_3 = 0.8706$, the sample standard deviations $s_1 = 0.876$, $s_2 = 1.291$ and $s_3 = 0.8571$ the t-statistics 3.541, 2.666 and 3.212 and the corresponding p-values 0.0063, 0.02577 and 0.01062. We want to adjust for multiple testing by using the Bonferroni-Holm-Procedure with $\alpha = 0.025$.

```
> # Estimates:
> est <- c("H1"=0.860382, "H2"=0.9161474, "H3"=0.9732953)
> # Sample standard deviations:
> ssd <- c("H1"=0.8759528, "H2"=1.291310, "H3"=0.8570892)
> pval <- c(0.01260, 0.05154, 0.02124)/2
> simConfint(BonferroniHolm(3), pvalues=pval,
+             confint=function(node, alpha) {
+               c(est[node]-qt(1-alpha,df=9)*ssd[node]/sqrt(10), Inf)
+             }, estimates=est, alpha=0.025, mu=0, alternative="greater")
  lower bound estimate upper bound
H1      0.0000    0.8604      Inf
H2     -0.0076    0.9161      Inf
H3      0.0000    0.9733      Inf
> # Note that the sample standard deviations in the following call
> # will be calculated from the pvalues and estimates.
> simConfint(BonferroniHolm(3), pvalues=pval,
+             confint="t", df=9, estimates=est, alpha=0.025, alternative="greater")
  lower bound estimate upper bound
[1,]  0.000000    0.8604      Inf
[2,] -0.007581    0.9161      Inf
[3,]  0.000000    0.9733      Inf
```

4.2 Using the GUI

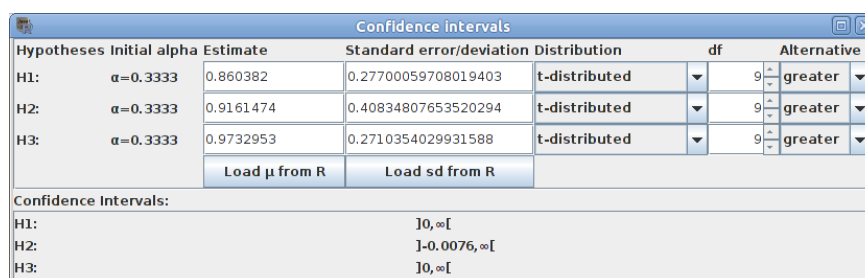


Figure 6: For normal and t-distributions simultaneous CI can be calculated by the GUI.

Use the following two buttons:

adj.
pval

↔

See [5].

5 Weighted parametric tests

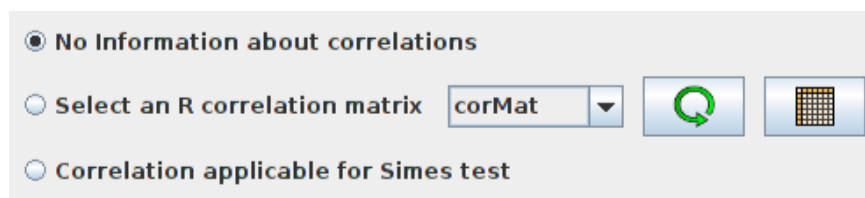


Figure 7: You can also specify a correlation between the tests.

In the lower right panel with p-values, it is also possible to specify a known correlation between these values (see figure 7).

For further information please take a look at the vignette "*Weighted parametric tests defined by graphs*".

6 Correlation matrix creation

The GUI features a dialog for easy creation of correlation matrices (see figure 8).

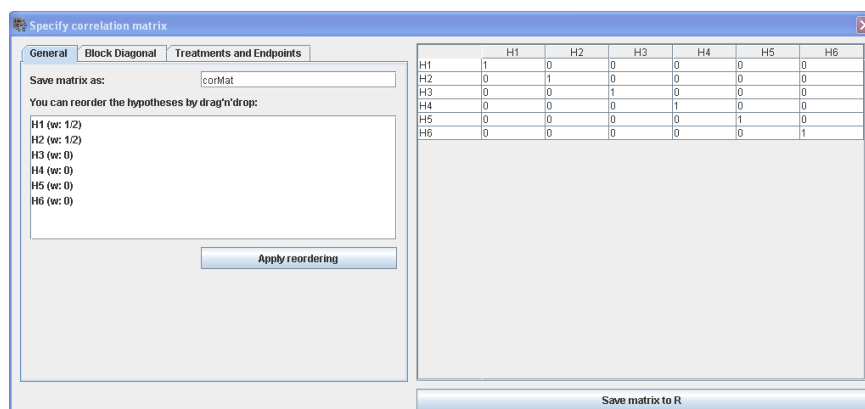


Figure 8: You can configure many things in the option dialog.

If the entered matrix is not positive semidefinite, i.e. negative eigen values exist, a warning is given.

$$A \otimes B$$

Graph from Bretz et al. (2009) - Figure 14

Literature: Bretz F., Maurer W., Brannath W., Posch M.: A graphical approach to sequentially rejective multiple test procedures. Statistics in Medicine 2009; 28:586-604. Figure 14.

7 Epsilon edges

The GUI supports epsilon edges. You can enter the weights in R syntax, e.g. $1-2*\epsilon+1/3*\epsilon^2$ for $1 - 2\epsilon + \frac{1}{3}\epsilon^2$.

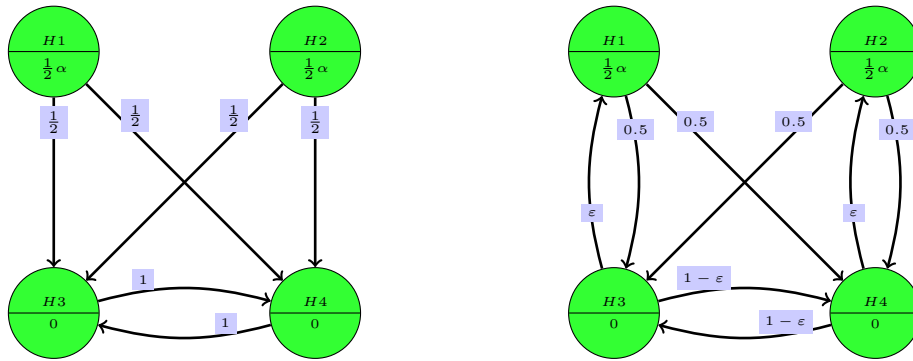


Figure 9: The Parallel Gatekeeping and the Improved Parallel Gatekeeping Procedure.

```
> m <- rbind(H1=c(0, 0, 0.5, 0.5),
+           H2=c(0, 0, 0.5, 0.5),
+           H3=c("\\epsilon", 0, 0, "1-\\epsilon"),
+           H4=c(0, "\\epsilon", "1-\\epsilon", 0))
> graph <- matrix2graph(m)
> graph
A graphMCP graph
H1 (not rejected, weight=0.25)
H2 (not rejected, weight=0.25)
H3 (not rejected, weight=0.25)
H4 (not rejected, weight=0.25)
Edges:
H1 -( 0.5 )-> H3
H1 -( 0.5 )-> H4
H2 -( 0.5 )-> H3
H2 -( 0.5 )-> H4
H3 -( \\epsilon )-> H1
H3 -( 1-\\epsilon )-> H4
H4 -( \\epsilon )-> H2
H4 -( 1-\\epsilon )-> H3
> substituteEps(graph, eps=0.001)
A graphMCP graph
H1 (not rejected, weight=0.25)
H2 (not rejected, weight=0.25)
H3 (not rejected, weight=0.25)
H4 (not rejected, weight=0.25)
Edges:
H1 -( 0.5 )-> H3
H1 -( 0.5 )-> H4
H2 -( 0.5 )-> H3
H2 -( 0.5 )-> H4
H3 -( 0.001 )-> H1
H3 -( 0.999 )-> H4
H4 -( 0.001 )-> H2
```

```

H4 -( 0.999 )-> H3
> gMCP(graph, pvalues=c(0.02, 0.04, 0.01, 0.02), eps=0.001)
gMCP-Result

Initial graph:
A graphMCP graph
H1 (not rejected, weight=0.25)
H2 (not rejected, weight=0.25)
H3 (not rejected, weight=0.25)
H4 (not rejected, weight=0.25)
Edges:
H1 -( 0.5 )-> H3
H1 -( 0.5 )-> H4
H2 -( 0.5 )-> H3
H2 -( 0.5 )-> H4
H3 -( 0.001 )-> H1
H3 -( 0.999 )-> H4
H4 -( 0.001 )-> H2
H4 -( 0.999 )-> H3

P-values:
  H1  H2  H3  H4
0.02 0.04 0.01 0.02

Adjusted p-values:
  H1  H2  H3  H4
0.04002 0.04002 0.04000 0.04002

Alpha: 0.05

Hypothesis rejected:
  H1  H2  H3  H4
TRUE TRUE TRUE TRUE

Final graph after 4 steps:
A graphMCP graph
H1 (rejected, weight=0)
H2 (rejected, weight=1)
H3 (rejected, weight=0)
H4 (rejected, weight=0)
No edges.

```

8 Power Simulations

No ε -edges are allowed.

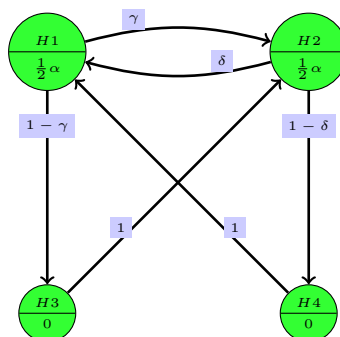


Figure 10: Graph from Bretz et al. (2009)

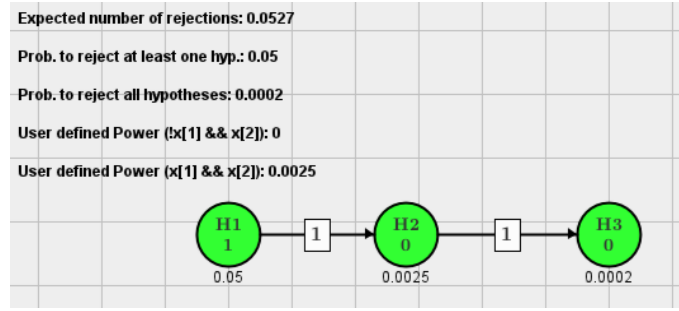


Figure 11: Local power and some (trivial) user defined power functions.

8.1 Variable edge weights

Apart from latin letters the following greek letters can be used to name a variable¹. Please enter them with a leading backslash so that they are recognized:

`\alpha`, `\beta`, `\gamma`, `\delta`, `\epsilon`, `\zeta`, `\eta`, `\theta`, `\iota`, `\kappa`, `\lambda`, `\mu`, `\nu`, `\xi`, `\pi`, `\rho`, `\sigma`, `\tau`, `\upsilon`, `\varphi`, `\chi`, `\psi` and `\omega`.

These are shown in the GUI as α , β , γ , δ , ϵ , ζ , η , θ , ι , κ , λ , μ , ν , ξ , π , ρ , σ , τ , υ , φ , χ , ψ and ω .

	H1	H2	H3	H4
H1	0	1	0	0
H2	0	0	0	0
H3	0	1	0	0
H4	1	0	0	0

```
> graph <- generalSuccessive()
> graph
A graphMCP graph
H1 (not rejected, weight=0.5)
H2 (not rejected, weight=0.5)
H3 (not rejected, weight=0)
H4 (not rejected, weight=0)
Edges:
H1 -( \gamma )-> H2
H1 -( 1-\gamma )-> H3
H2 -( \delta )-> H1
H2 -( 1-\delta )-> H4
H3 -( 1 )-> H2
H4 -( 1 )-> H1
```

9 Options and Import/Export

9.1 Options

This subsection is work in progress, but fortunately the options in figure 12 should be fairly self-explanatory.

Grid

Number of digits

Line width

Font Size

Look'n'Feel

¹Note that omicron is not allowed since it can not be distinguished from the latin character "o".

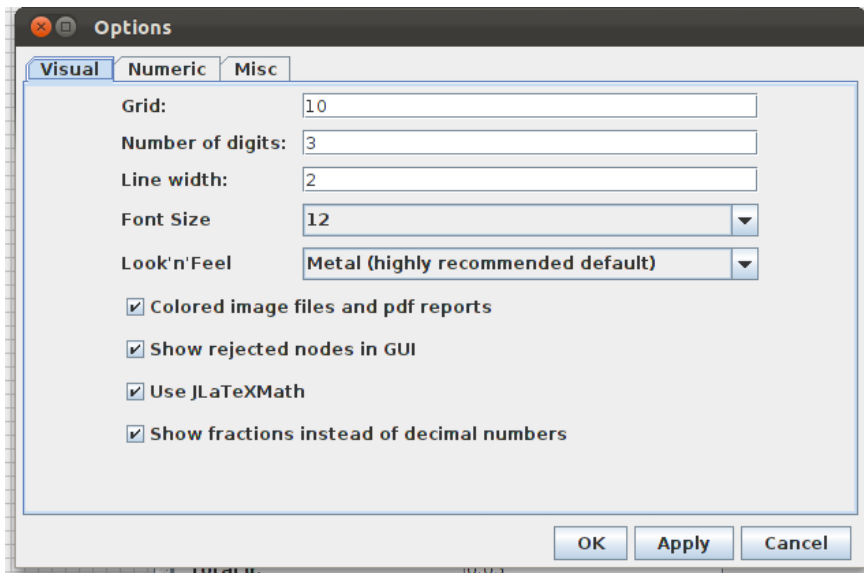


Figure 12: You can configure many things in the option dialog.

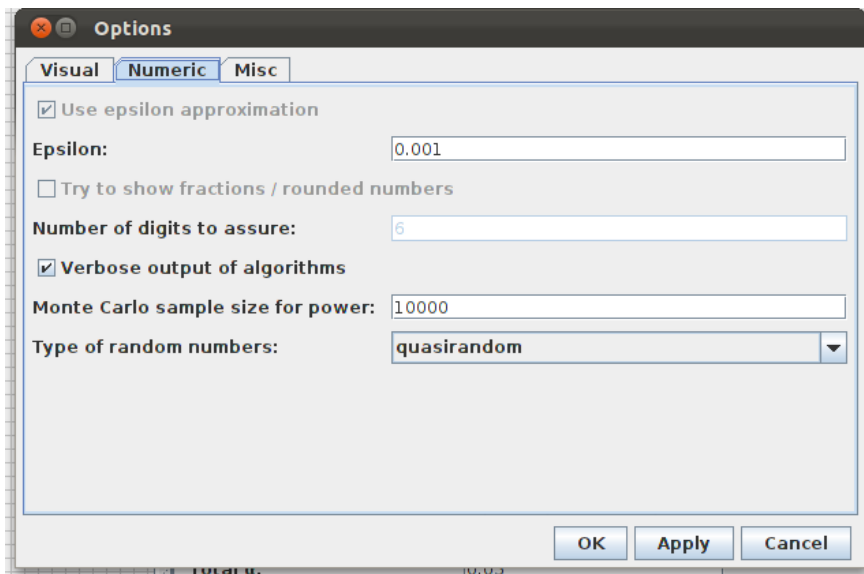


Figure 13: You can configure many things in the option dialog.

Colored image files and pdf reports

Show rejected nodes in GUI

Use JLaTeXMath There are few reasons not to use the free Java library JLaTeXMath to render numbers, symbols and formulas in the GUI. The option is mainly provided in case that errors occur displaying the numbers and formulas.

Show fractions instead of decimal numbers

Use epsilon approximation Not yet available in this version.

Epsilon

Try to show fractions / rounded numbers

Number of digits to assure

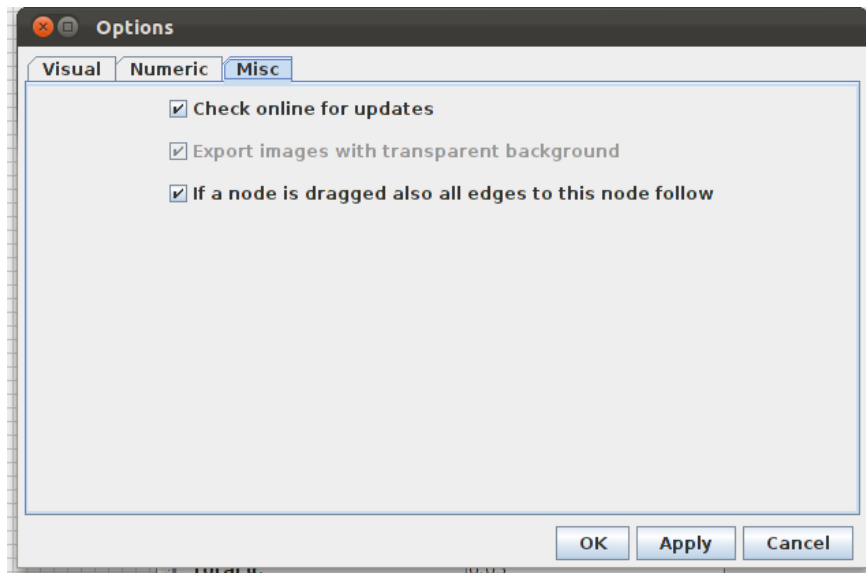


Figure 14: You can configure many things in the option dialog.

Verbose output of algorithms

Monte Carlo sample size for power

Type of random numbers

Check online for updates

Export images with transparent background

If a node is dragged also all edges to this node follow

Automatically enter the editing mode, whenever a table cell gets the focus

9.1.1 Privacy

The GUI is able to connect to our server to

1. check whether a new version of gMCP exists,
2. send a wishlist if the user chooses to,
3. send bug reports if an error occurs.

Only in the last case of a bug report some information about your computer is collected that can be reviewed by the user before sending the bug report. If you do not agree with sending this data, simply don't send a problematic bug report or if you never want to send bug reports, disable the option in the options menu.

9.2 Import/Exports

This subsection is work in progress, but fortunately the menu entries in figure 15 should be fairly self-explanatory.

You can export graphs to png files. The background of these png files will be made transparent, so that they will fit into whichever document you insert them. Note that some image viewers visualize transparency with a checkerboard pattern.

File	Example graphs	Extras	Help
New Graph			
Load Graph from R			
Load Graph from RData file			
Save Graph to R			
Save Graph to RData file			
Export Graph to PNG Image			
Export Graph to LaTeX File			
Save LaTeX Report			
Save PDF Report			
Quit			

Figure 15: Import and export of graphs.

9.3 Important TikZ commands for optimizing the reports

A clear automatic placement of edges and weight labels without overlapping is a very difficult task and for complicated graphs the `gMCP` package will often fail to accomplish this. There is the possibility to adjust the edges and labels in the GUI, but since the `LATEX` graph layout is not (yet) exactly the same, there is perhaps the need for adjusting the graphs in the TikZ code. The TikZ program is very useful and we recommend it for many purposes, but perhaps you don't have the time to read the 560 pages manual [8], so here is a short overview of the most important commands for this kind of graphs.

Let's start with this graph in figure 16:

```
\begin{tikzpicture}[scale=1]
\node (H11) at (200bp,200bp) [draw,circle split,fill=green!80] {$H_{11}$ \nodepart{lower}  $\frac{2}{15}\alpha$ };
...
\draw [->,line width=1pt] (H11) to[bend left=15] node[near start,above,fill=blue!20] {0.667} (H12);
...
\end{tikzpicture}
```

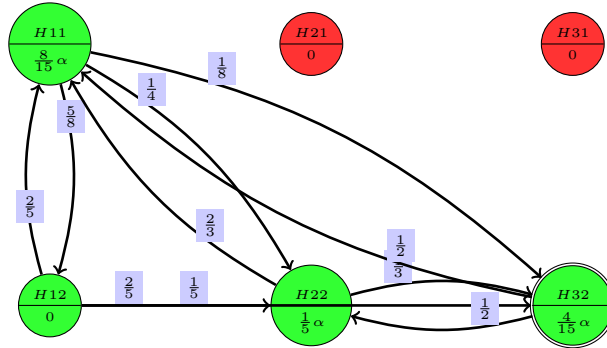


Figure 16: Graph from `graph2latex` that does not look optimal.

You can scale the TikZ graphic by changing the `[scale=1]` option. By default `graph2latex` doesn't scale TikZ graphics, but has an optional parameter `scale`.

For an explanation what `green!80` means and how you can specify other colors, please take a look at the `xcolor` manual [7].

You can choose between the following label positions `above`, `below`, `right`, `left`, `above right`, `above left`, `below right`, and `below left`. In addition these positions can take an optional dimension argument, so that for example `below=1pt` can be used to place a label below and additionally shift it 1pt downwards. You can change the position where the edge weight label is placed to `at start`, `very near start`, `near start`, `midway`, `near end`, `very near end` and `at end` or simply use something like `pos=0.5`. If you add an argument `sloped`, the text label will be rotated so that a parallel line to the base line becomes a tangent to the edge.

Often it is useful to reduce the bending angle in `[bend left=15]` below 15. You could also specify and

change `out=15` and `in=165` separately.

A powerful feature is the use of styles, since this will effect all objects of a given class. But for this please take a look directly at the TikZ manual [8].

10 Case Studies

This section is work in progress.

10.1 Identifying effective and/or safe doses by stepwise confidence intervals for ratios

In this subsection we show how to use gMCP to reproduce the results of the paper [2] with the same title.

10.2 Testing strategies in multi-dose experiments including active control

[1]

```
> data(hydroquinone)
> pvalues <- c()
> x <- hydroquinone$micronuclei[hydroquinone$group=="C-"]
> for (dose in c("30 mg/kg", "50 mg/kg", "75 mg/kg", "100 mg/kg", "C+")) {
+   y <- hydroquinone$micronuclei[hydroquinone$group==dose]
+   result <- wilcox.test(x, y, alternative="less", correct=TRUE)
+   pvalues <- c(result$p.value, pvalues)
+ }
> pvalues
[1] 0.004929 0.002634 0.002634 0.004319 0.066255
> library(coin)
> pvalues <- c()
> for (dose in c("30 mg/kg", "50 mg/kg", "75 mg/kg", "100 mg/kg", "C+")) {
+   subdata <- droplevels(hydroquinone[hydroquinone$group %in% c("C-", dose),])
+   result <- wilcox.test(micronuclei ~ group, data=subdata, distribution="exact")
+   pvalues <- c(pvalue(result), pvalues)
+ }
> pvalues
[1] 0.006061 0.001263 0.001263 0.005051 0.135101
```

A Appendix - Multiple Testing Basics

Let Θ be a parameter space indexing a family of probabilities $\{P_\theta \mid \theta \in \Theta\}$ and $(\Omega, \mathcal{F}, P_\theta)$ the associated probability spaces. For a family of null hypotheses $H_i \subset \Theta$, $i \in \{1, \dots, n\} =: I$ a multiple test procedure φ is defined as a family of $(\mathcal{F}, \mathfrak{Pot}(\{0, 1\}^n))$ -measurable functions $\{\varphi_J : \Omega \rightarrow \{0, 1\}^n \mid J \subset I\}$.

Definition A.1. Let $H_J := \bigcap_{j \in J} H_j$. The multiple test procedure φ controls the *familywise error rate at level α in the weak sense* if

$$\forall \theta \in H_I : P_\theta(\varphi_J = 1 \text{ for some } J \subset I) \leq \alpha.$$

The multiple test procedure φ controls the *familywise error rate at level α in the strong sense* if

$$\forall \theta \in \Theta : P_\theta(\varphi_J = 1 \text{ and } \theta \in H_J \text{ for some } J \subset I) \leq \alpha.$$

This section is work in progress.

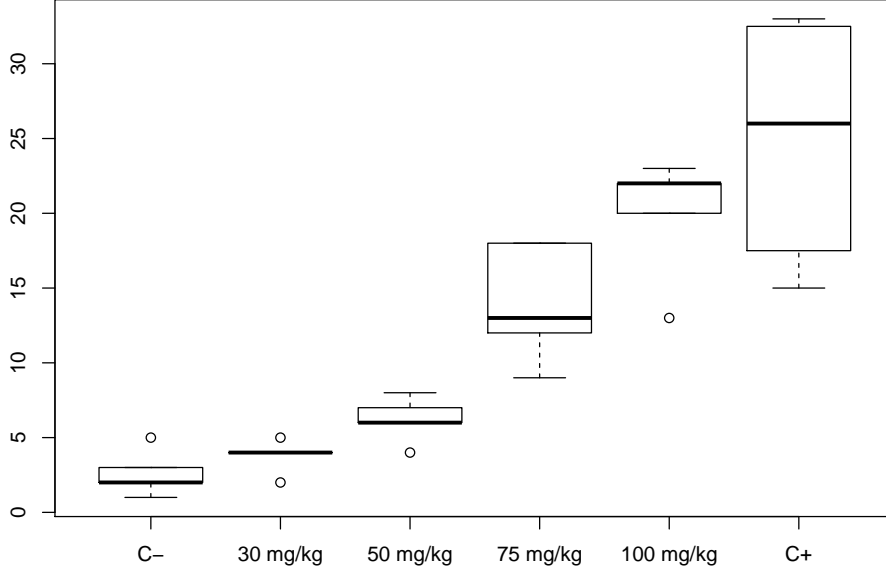


Figure 17: Boxplot of the hydroquinone data set

A.1 Closed testing principle

A.2 Partitioning principle

Definition A.2.

Theorem A.3 (Simes-Procedure). *Let T_1, \dots, T_m be test statistics for $m \in \mathbb{N}$ null hypotheses H_1, \dots, H_m and p_1, \dots, p_m the associated p -values. Then the following test will control the familywise error rate at level $\alpha \in]0, 1[$ in the strong sense:*

Denote the ordered p -values by $p^{(1)} < p^{(2)} < \dots < p^{(m)}$ and the corresponding hypotheses by $H^{(1)}, H^{(2)}, \dots, H^{(m)}$.

Reject H_0 if

$$p^{(j)} \leq \frac{j\alpha}{n} \quad \text{for some } 1 \leq j \leq m.$$

For independent tests the FWER is controlled at level α .

Theorem A.4 (Weighted Simes Procedure). *Benjamini and Hochberg (1997)*

Let $\sum_{k=1}^m w_k = m$ and reject H_0 if

$$p^{(j)} \leq \frac{\sum_{k=1}^j w_{(k)}}{m} \cdot \alpha \quad \text{for some } 1 \leq j \leq m.$$

Adjusted p-Values in the Simes Test

For each set $J \subset I$ we calculate

$$m_J := \min_{j \in J} \left(\frac{p_j}{\sum_{i \in J_j} w_i} \right), \quad J_j = \{k \in J \mid p_k \leq p_j\}.$$

The weighted Simes Test rejects H_J iff $m_J \leq \alpha$.

In a closed testing procedure a hypothesis H_j is rejected iff H_J is rejected for each $J \subset I$ with $j \in J$.
An adjusted p-value p'_j is defined as the minimal α so that the test to global level α rejects H_j .
Therefore $p'_j = \max(m_J \mid j \in J)$.

Index

A	
adjusted p-values	9
B	
Bonferroni-Holm-Procedure	3
C	
Closed testing principle	18
coordinates	6
correlation matrix	10
E	
edge weights	
variable	13
epsilon edges	11
export	15
G	
gatekeeping	
improved parallel	11
parallel	11
graph2latex	16
I	
import	15
M	
matrix2graph	5
O	
options	13
P	
parallel gatekeeping	11
Partitioning principle	18
power simulation	12
R	
report generation	9
S	
setWeights	5
Simes-Procedure	18
simultaneous confidence intervals	9
T	
TikZ	6, 16
W	
Weighted Simes Procedure	18

List of Algorithms

1	Removing node i , passing the weight and updating the graph edges	5
---	---	---

List of Figures

1	Graph representing the Bonferroni-Holm-Procedure for three hypotheses.	3
2	Example showing how two null hypotheses can be rejected with p-values $p_1 = 0.01$, $p_2 = 0.07$ and $p_3 = 0.02$	4
3	Example graph from [4] that we will create in this vignette.	5
4	The graphical user interface allows testing, calculation of confidence intervals and adjusted p-values.	7
5	Final graph from the test procedure after rejection of H_{21} , H_{31} and H_{32}	9
6	For normal and t-distributions simultaneous CI can be calculated by the GUI.	10
7	You can also specify a correlation between the tests.	10
8	You can configure many things in the option dialog.	10
9	The Parallel Gatekeeping and the Improved Parallel Gatekeeping Procedure.	11
10	Graph from Bretz et al. (2009)	12
11	Local power and some (trivial) user defined power functions.	13
12	You can configure many things in the option dialog.	14
13	You can configure many things in the option dialog.	14
14	You can configure many things in the option dialog.	15
15	Import and export of graphs.	16
16	Graph from <code>graph2latex</code> that does not look optimal.	16
17	Boxplot of the hydroquinone data set	18

List of Tables

References

- [1] P. Bauer, J. Röhmel, W. Maurer, and L. Hothorn. Testing strategies in multi-dose experiments including active control. *Statistics in Medicine*, 17(18):2133–2146, 1998. ISSN 1097-0258.
- [2] F. Bretz, L.A. Hothorn, and J.C. Hsu. Identifying effective and/or safe doses by stepwise confidence intervals for ratios. *Statistics in medicine*, 22(6):847–858, 2003. ISSN 1097-0258.
- [3] F. Bretz, W. Maurer, W. Brannath, and M. Posch. A graphical approach to sequentially rejective multiple test procedures. *Statistics in medicine*, 28(4):586–604, 2009. URL www.meduniwien.ac.at/fwf_adaptive/papers/bretz_2009_22.pdf.
- [4] F. Bretz, W. Maurer, and G. Hommel. Test and power considerations for multiple endpoint analyses using sequentially rejective graphical procedures. *Statistics in medicine*, 2010.
- [5] F. Bretz, M. Posch, E. Glimm, F. Klinglmueller, W. Maurer, and K. Rohmeyer. Graphical approaches for multiple comparison problems using weighted bonferroni, simes or parametric tests. *Biometrical Journal*, 53(6):894–913, 2011. URL <http://onlinelibrary.wiley.com/doi/10.1002/bimj.201000239/full>.
- [6] S. Holm. A simple sequentially rejective multiple test procedure. *Scand. J. Statist.*, 6:65–70, 1979.
- [7] U. Kern. *Extending LaTeX’s color facilities: the xcolor package*, 2007. URL <http://www.ctan.org/tex-archive/macros/latex/contrib/xcolor/>.
- [8] T. Tantau. *The Tik Z and PGF Packages Manual for version 2.00*, 2008. URL <http://www.ctan.org/tex-archive/graphics/pgf/base/doc/generic/pgf/pgfmanual.pdf>.

Table of Symbols

Sets

\mathbb{R}	set of real numbers
\mathbb{N}_0	set of natural numbers (including 0)
$\mathfrak{P}\text{ot}(X)$	power set of set X , i.e. the set of all subsets of X

Functions

$\langle \cdot, \cdot \rangle$	standard direct product $\langle x, y \rangle = \sum_{j=1}^n x_j \cdot y_j$ for $x, y \in \mathbb{R}^n$
id_X	identity on X , i.e. $\text{id}_X : X \rightarrow X, x \mapsto x$