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

Kornelius Rohmeyer

V0.8.17

October 21, 2024

Contents

Chapter 1

Introduction

This package provides functions and graphical user interfaces for graph based multiple test procedures. By defining weighted directed 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, one example is the weighted Bonferroni procedure that leads to a sequentially rejective multiple test procedure.

For all steps either graphical user interfaces or the R Console with S4 objects and methods can be used.

1.1 Installation

If you don't already have R (R Core Team [2014]) on your system, you can download a bundled version of R and gMCP from `http://www.algorithm-forge.com/gMCP/bundle/`.

Otherwise 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 please write us an email at `help@small-projects.de`. We are eager to help and to learn about existing problems.

1.2 Basic Theoretical Background

Graph based multiple test procedures are closed test procedures, i.e. for a family $\{H_i \mid i \in I\}$, $I = \{1, \ldots, n\}$ of elementary hypotheses each intersection $\bigcap_{j \in J} H_j$, $J \subset I$ is tested with a local level α test. Following the closed testing principle one can derive a multiple test procedure that controls the family-wise error rate (FWER) at level α.

The local level α tests in gMCP are weighted tests, where the weights are derived from a directed weighted graph G_I. Examples of weigthed tests that are available in gMCP are the weighted Bonferroni, parametric and Simes tests.

For each intersection $\bigcap_{j \in J} H_j$, $J \subset I$ a graph G_J can be derived from G_I and the weights for the weighted local test for $\bigcap_{j \in J} H_j$ are the weights of the nodes of G_J. To derive graph G_J remove all nodes that are not in J and update the edges of the graph according to Algorithm 1 (the order does not matter). For a more detailed version please take a look at the article from Bretz et al. [2011b] that is freely available as an

---

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** $g_{lj} \cdot g_{jl} \neq 1$ **then**
         $g_{lk} \leftarrow \frac{g_{lk} + g_{lj} \cdot g_{jk}}{1 - g_{lj} \cdot g_{jl}}$
       else
         $g_{lk} \leftarrow 0$
       end if
     end for
   end for

1.3   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 from Holm [1979].

Theorem 1.3.1 (Bonferroni-Holm-Procedure)**.** *Let $T_1, \ldots, T_m$ be test statistics for $m \in \mathbb{N}$ null hypotheses $H_1, \ldots, H_m$ and $p_1, \ldots, p_m$ 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 $p^{(1)} < p^{(2)} < \ldots < p^{(m)}$ and the corresponding hypotheses by $H^{(1)}, H^{(2)}, \ldots, H^{(m)}$. Reject $H^{(1)}, H^{(2)}, \ldots, H^{(j)}$ 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 1.1. We see a fully connected graph, where each node represents a hypothesis and the nodes and edges have weights.
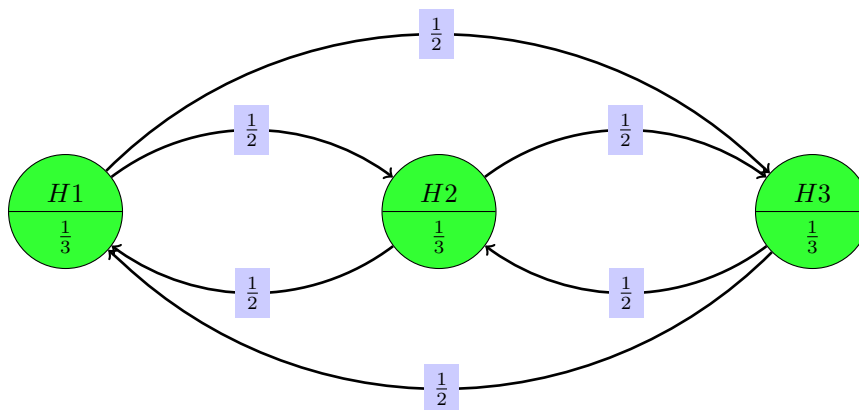
Figure 1.1: Graph representing the Bonferroni-Holm-Procedure for three hypotheses.

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 1.3.2. We give an example for the Bonferroni-Holm-Procedure. Of course this package is made for more advanced tests (you find a selection in appendix A with applied examples in section 9), 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 1.2 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.
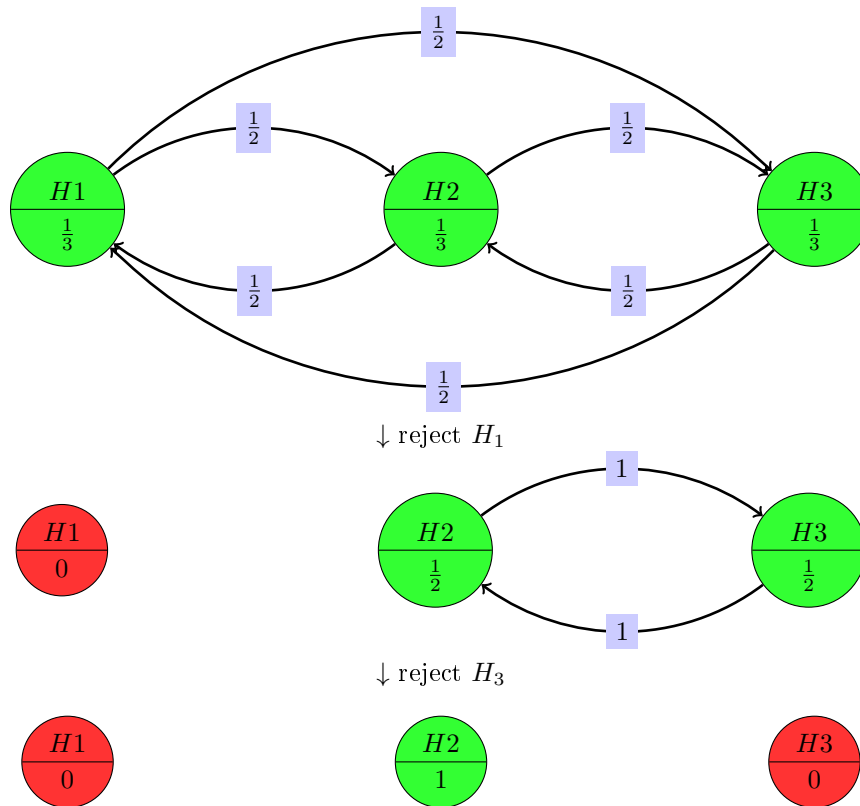
$\downarrow$ reject H_1

$\downarrow$ reject H_3

Figure 1.2:   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 2.2 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 (weight=0.3333)
## H2 (weight=0.3333)
## H3 (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 (weight=1)
## H3 (rejected, weight=0)
## No edges.
```

Chapter 2

Creating Weighted Graphs

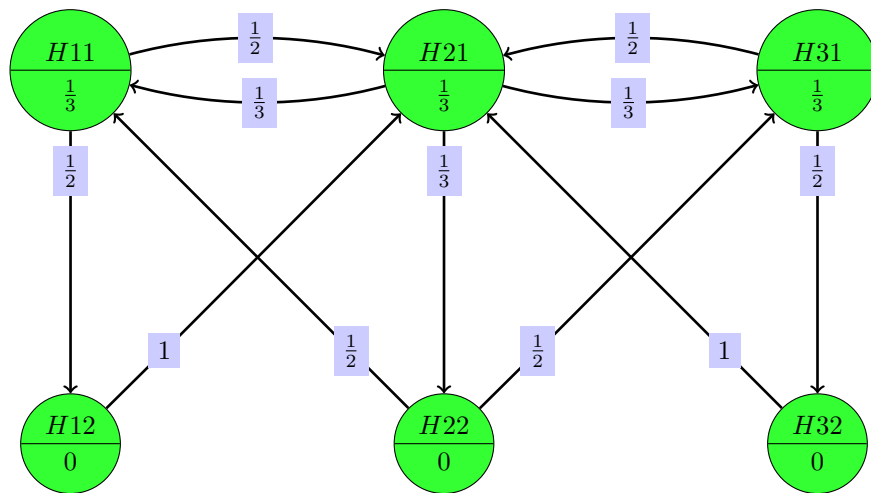In the first step a graph that describes the multiple test procedures must be created.

Figure 2.1: Example graph from Bretz et al. [2011a] that we will create in this vignette.

2.1 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. [2011a] that you can see in figure 2.1.

```
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 transition 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 (weight=0.3333)
## H21 (weight=0.3333)
## H31 (weight=0.3333)
## H12 (weight=0)
## H22 (weight=0)
## H32 (weight=0)
## Edges:
## H11  -( 0.5 )->  H21
## H11  -( 0.5 )->  H12
## H21  -( 0.333333333333333 )->  H11
## H21  -( 0.333333333333333 )->  H31
## H21  -( 0.333333333333333 )->  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 two 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 package TikZ from Tantau [2008] (figure 2.1 shows the compiled result):

```
cat(graph2latex(graph))

## \begin{tikzpicture}
##
## \node (H11) at (125bp,-125bp)[draw,circle split,fill=green!80] {$H11$ \nodepart{lower} $\frac{1}{3}$};
## \node (H21) at (325bp,-125bp)[draw,circle split,fill=green!80] {$H21$ \nodepart{lower} $\frac{1}{3}$};
## \node (H31) at (525bp,-125bp)[draw,circle split,fill=green!80] {$H31$ \nodepart{lower} $\frac{1}{3}$};
## \node (H12) at (125bp,-325bp)[draw,circle split,fill=green!80] {$H12$ \nodepart{lower} $0$};
## \node (H22) at (325bp,-325bp)[draw,circle split,fill=green!80] {$H22$ \nodepart{lower} $0$};
## \node (H32) at (525bp,-325bp)[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
```

2.2   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.
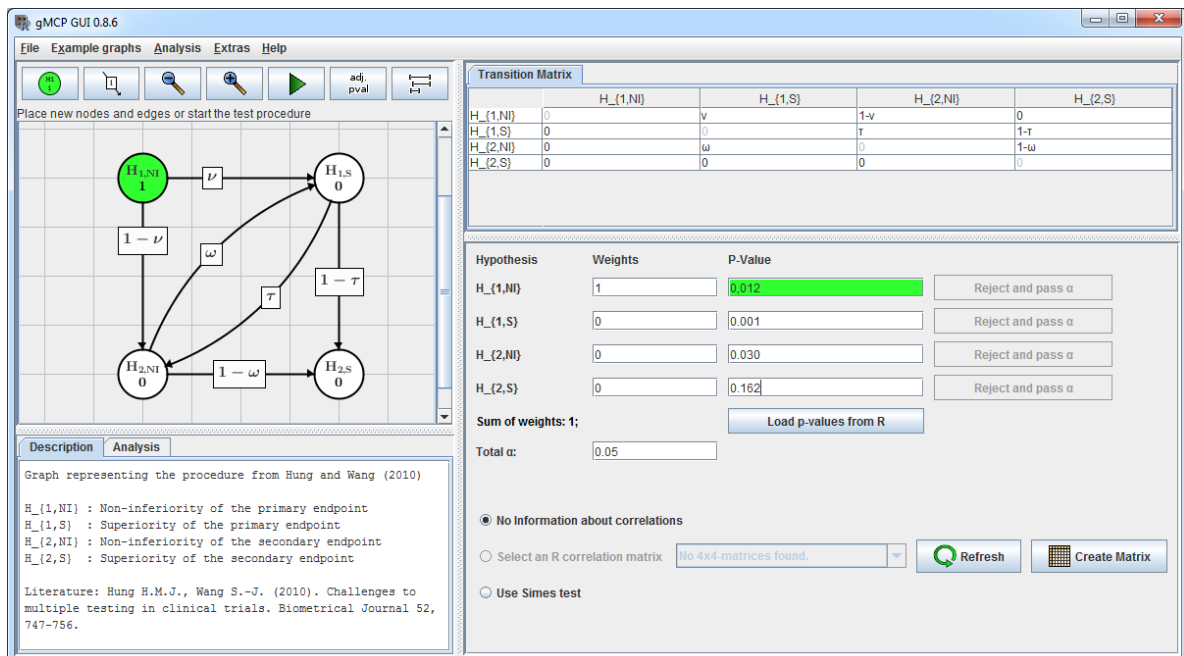
```
graphGUI("graph")
```
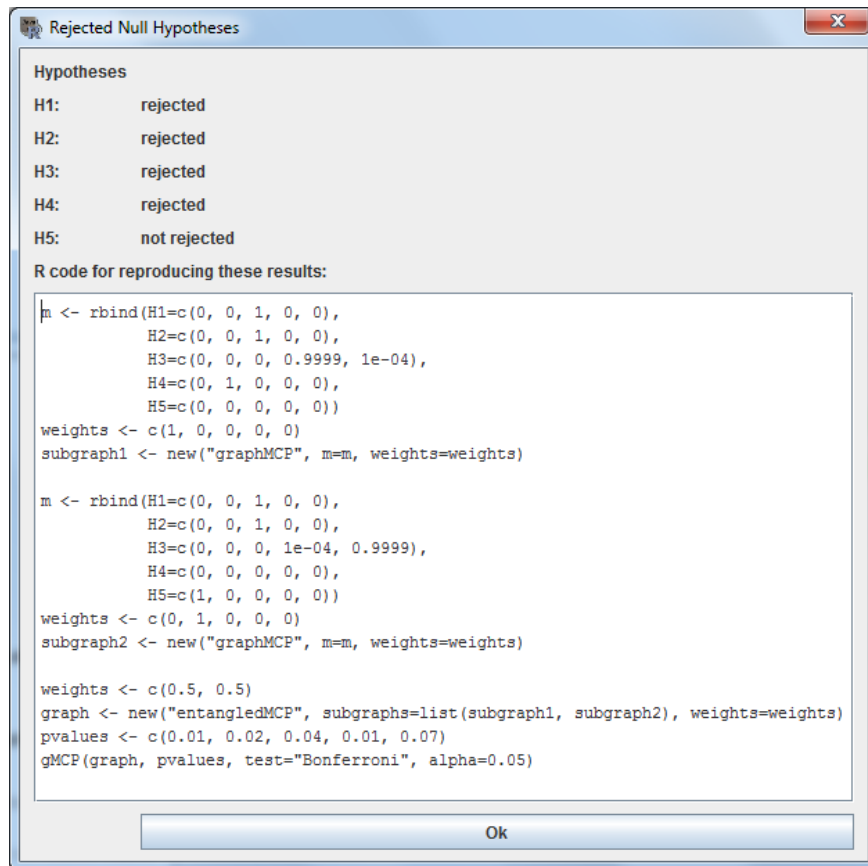

Figure 2.2:   The graphical user interface allows testing, calculation of confidence intervals and adjusted p-values.

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. Double clicking the nodes or edges will open a property dialog that can also be accessed via the right-click context menu. These dialogs and context menus also give you the option to delete the selected node/edge.

As seen in figure 2.3 the GUI will show you R code to reproduce results.

Figure 2.3: R Code generated and shown by the GUI to reproduce the results.

Chapter 3

The sequentially rejective MTP

For a full description of the sequentially rejective multiple testing procedure take a look at Bretz et al. [2009].

3.1 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 (weight=0.5)
## H31 (weight=0.3333)
## H12 (weight=0.1667)
## H22 (weight=0)
## H32 (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 (weight=0.3333)
```

```
## H21 (weight=0.3333)
## H31 (weight=0.3333)
## H12 (weight=0)
## H22 (weight=0)
## H32 (weight=0)
## Edges:
## H11  -( 0.5 )->  H21
## H11  -( 0.5 )->  H12
## H21  -( 0.333333333333333 )->  H11
## H21  -( 0.333333333333333 )->  H31
## H21  -( 0.333333333333333 )->  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 (weight=0.6667)
## H21 (rejected, weight=0)
## H31 (rejected, weight=0)
## H12 (weight=0)
## H22 (weight=0.3333)
## H32 (rejected, weight=0)
## Edges:
## H11  -( 0.666666666666667 )->  H12
## H11  -( 0.333333333333333 )->  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 3.1.

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

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

Adjusted p-values and simultaneous confidence intervals

Also adjusted p-values and simultaneous confidence intervals can be computed.
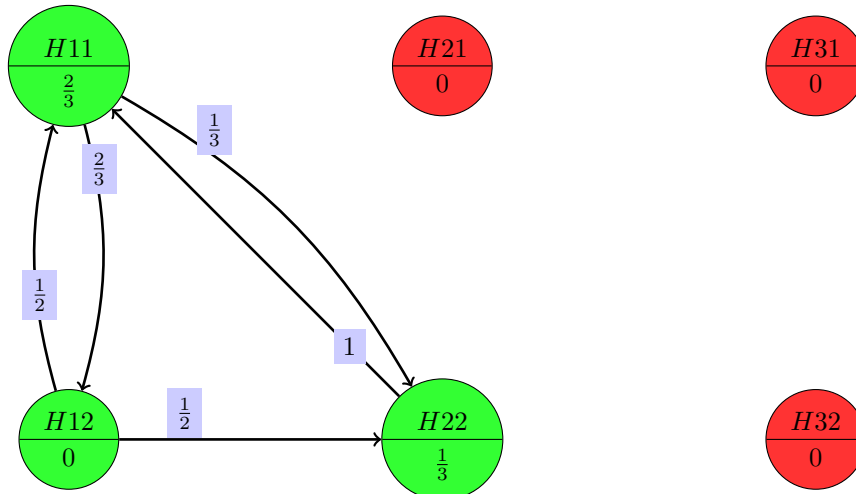
Figure 3.1: Final graph from the test procedure after rejection of H_{21}, H_{31} and H_{32}.

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
```

3.2   Using the GUI
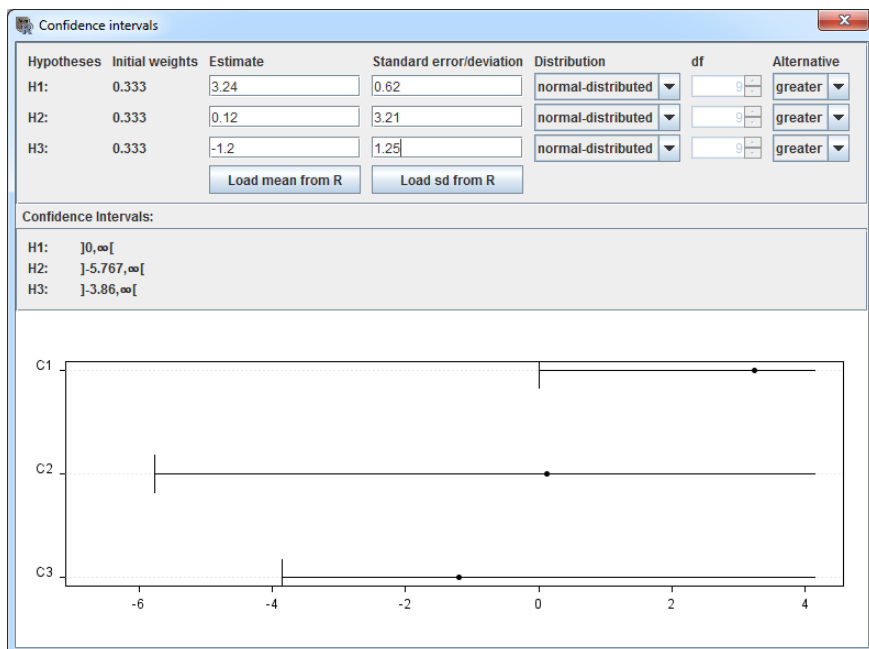
Use the following two buttons:

See Bretz et al. [2011b].

Figure 3.2: For normal and t-distributions simultaneous CI can be calculated by the GUI.

Chapter 4

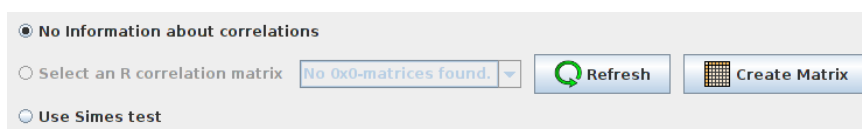# Weighted parametric and Simes tests

Figure 4.1: 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 the original test statistics (see figure 4.1). Either you can perform a Simes test or a weighted parametric tests as described in Bretz et al. [2011b]. For the later it is assumed that under the global null hypothesis $(\Phi^{-1}(1 - p_1), \ldots, \Phi^{-1}(1 - p_m))$ follow a multivariate normal distribution with correlation matrix Σ where Φ^{-1} denotes the inverse of the standard normal distribution function. For example, this is the case if $p_1, \ldots, p_m$ 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.

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

4.1 Correlation matrix creation

The GUI features a dialog for easy creation of correlation matrices (see figure 4.2).

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

This dialog is perhaps useful on its own and can be opened by calling the function `corMatWizard`.
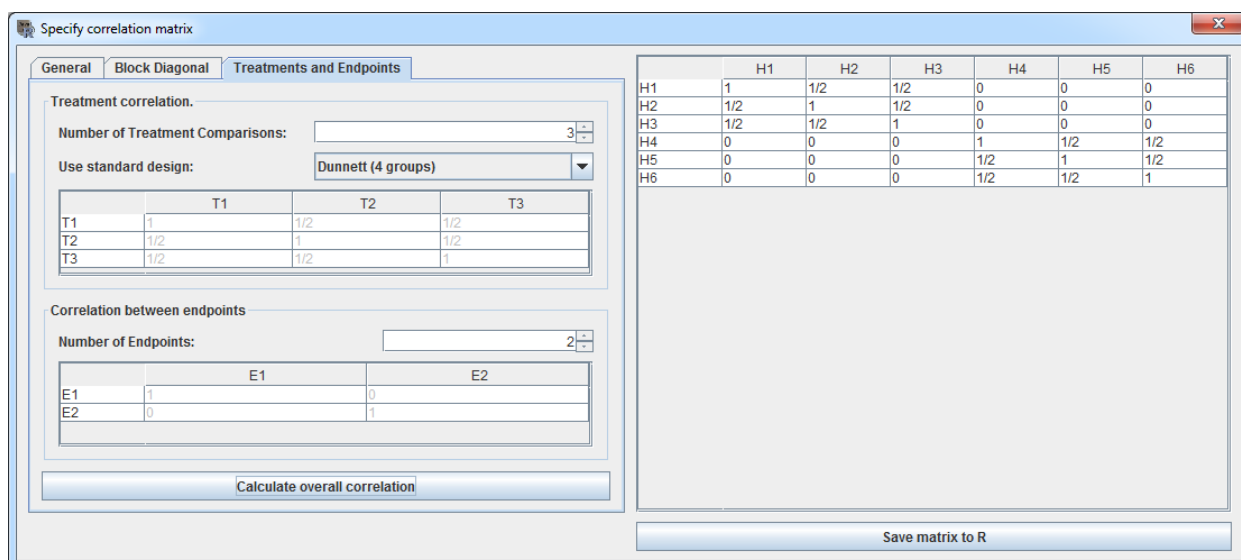
Figure 4.2: Dialog for specifying a correlation matrix.

Chapter 5

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 $1 - 2\varepsilon + \frac{1}{3}\varepsilon^2$.
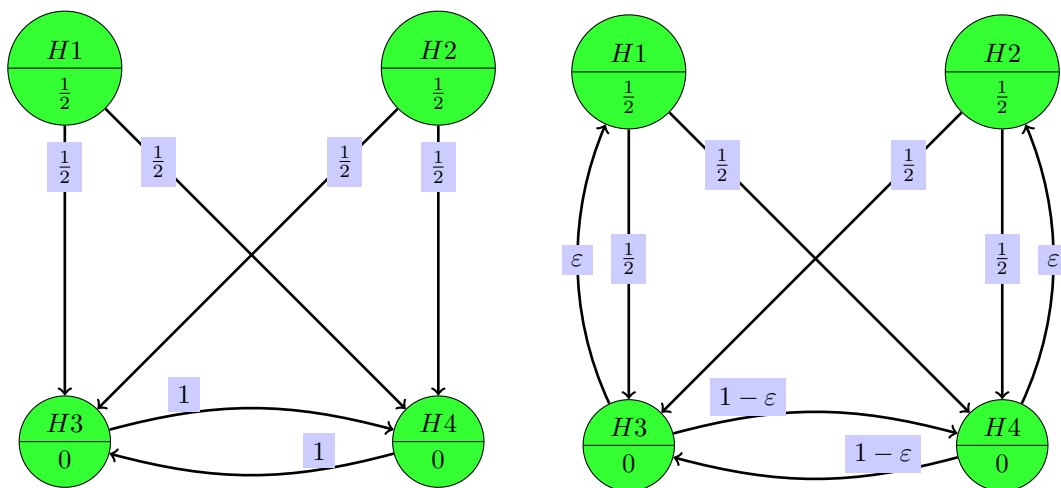
Figure 5.1: The Parallel Gatekeeping and the Improved Parallel Gatekeeping Procedure.

```
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 <- improvedParallelGatekeeping()
graph

## A graphMCP graph
## H1 (weight=0.25)
## H2 (weight=0.25)
## H3 (weight=0.25)
## H4 (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 (weight=0.25)
## H2 (weight=0.25)
## H3 (weight=0.25)
## H4 (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 (weight=0.25)
## H2 (weight=0.25)
## H3 (weight=0.25)
## H4 (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
## Sum of weight: 0
## H1 (rejected, weight=0)
## H2 (rejected, weight=0)
```

```
## H3 (rejected, weight=0)
## H4 (rejected, weight=0)
## No edges.
```

Chapter 6

Entangled graphs

As a new feature we support entangled graphs (see Maurer and Bretz [2013]), which allow us to describe an even bigger class of multiple test procedures in the familiar graphical way. Most notably these test procedures can have some kind of memory when passing the local significance levels to unrejected null hypotheses, i.e. for the step-wise rejective Bonferroni based test procedure the passing of alpha levels when a null hypothesis is rejected can be different depending from which previous rejected nodes this alpha level originates.

Definition 6.0.1 (Entangled Graph). An *entangled graph* is a pair $(\mathcal{G}, w)$ with $\mathcal{G} = (\mathcal{G}_1, \ldots, \mathcal{G}_n)$ a n-tuple of graphs and w .... We call $\mathcal{G}_1, \ldots, \mathcal{G}_n$ the *component graphs* of the entangled graph $\mathcal{G}$.

You can add a second component graph by selecting "*Extras $\rightarrow$ Add entangled graph*" from the menu. Should be "*Extras $\rightarrow$ Add another component graph*"?

For each graph a new tab in the graph panel and transition matrices panel is created (see figure 6.1 for example).
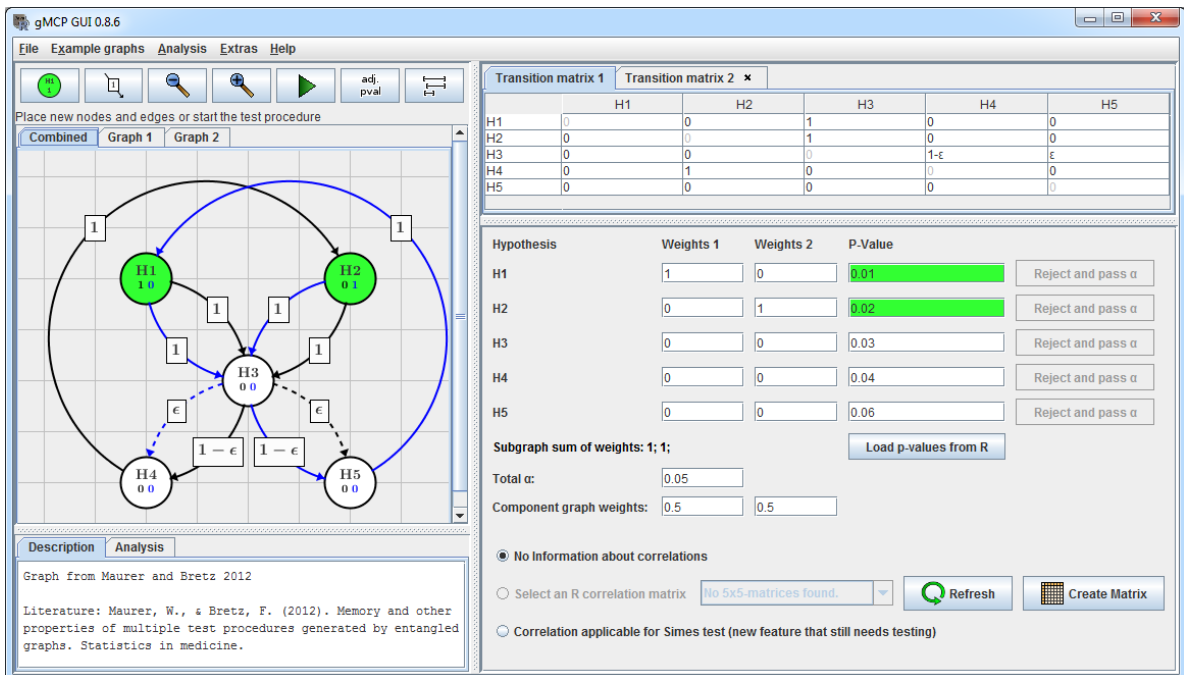

Figure 6.1: Different tabs show the different entangled graphs / transition matrices.

```
m <- rbind(H1=c(0, 0, 1, 0, 0),
           H2=c(0, 0, 1, 0, 0),
           H3=c(0, 0, 0, 0.9999, 1e-04),
           H4=c(0, 1, 0, 0, 0),
           H5=c(0, 0, 0, 0, 0))
weights <- c(1, 0, 0, 0, 0)
subgraph1 <- new("graphMCP", m=m, weights=weights)

m <- rbind(H1=c(0, 0, 1, 0, 0),
           H2=c(0, 0, 1, 0, 0),
           H3=c(0, 0, 0, 1e-04, 0.9999),
           H4=c(0, 0, 0, 0, 0),
           H5=c(1, 0, 0, 0, 0))
weights <- c(0, 1, 0, 0, 0)
subgraph2 <- new("graphMCP", m=m, weights=weights)

weights <- c(0.5, 0.5)
graph <- new("entangledMCP", subgraphs=list(subgraph1, subgraph2), weights=weights)
```
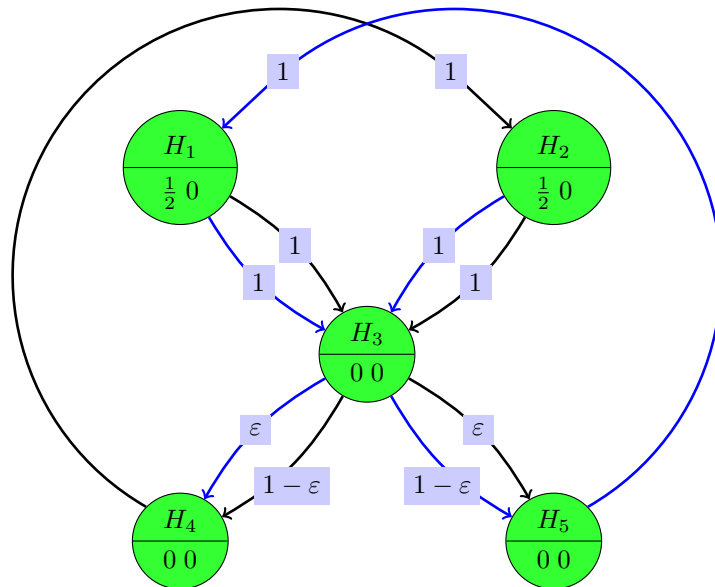

Figure 6.2: Entangled graph from Maurer et Bretz

Chapter 7

Power Simulations

The underlying distribution of test statistics in power simulations is assumed to be multivariate normal in the following with the exception of section 7.2 "Simulation for Non-normal Distributions".

7.1 Power Analysis

For the *"Options"* tab see subsection 8.1.2. The other tabs are described in the now following sections.

7.1.1 Noncentrality Parameter (NCP) Settings

The noncentrality parameter (NCP) of a Z-statistic

$$Z = \frac{\frac{1}{n}\sum_{i=1}^{n} X_i - \mu_0}{\sigma_X}\sqrt{n}, \quad \text{with } X_1, \ldots, X_n \text{ i.i.d. } \mathcal{N}(\mu_X, \sigma_X)$$

is

$$\text{NCP} = \frac{\mu_X - \mu_0}{\sigma_X}\sqrt{n}.$$

7.1.2 Correlation matrices

In the correlation matrix tab you will see the correlation matrix used for the generation of the test statistics and - if the parametric test was chosen - for the test. If the correlation matrix for the parametric test contains `NA` values, a second matrix is displayed (as can be seen in figure 7.1) were the user should specify the unknown values for the simulation.

7.1.3 User defined power functions

The user often is interested in certain combinations of rejections, for example that for one treatment the null hypotheses regarding both end points can be rejected.

The tab *"User defined power functions"* allows for exactly this kind of probability calculations. Use the R syntax and `"x[i]"` to specify the proposition that hypothesis i can be rejected. Negation (!) takes precedence over the logical conjunction ('and', &&), which takes precedence over the logical disjunction ('or', ||), but in doubt use brackets. Note that you can use all R commands, for example `any(x)` to see whether any hypotheses was rejected or `all(x[1:3])` to see whether all of the first three hypotheses were rejected.
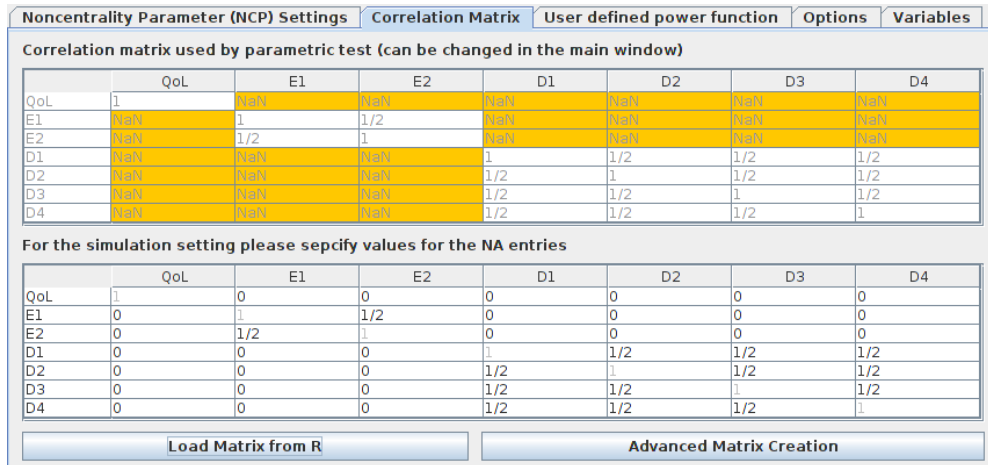
Figure 7.1: Correlation matrices for test and simulation

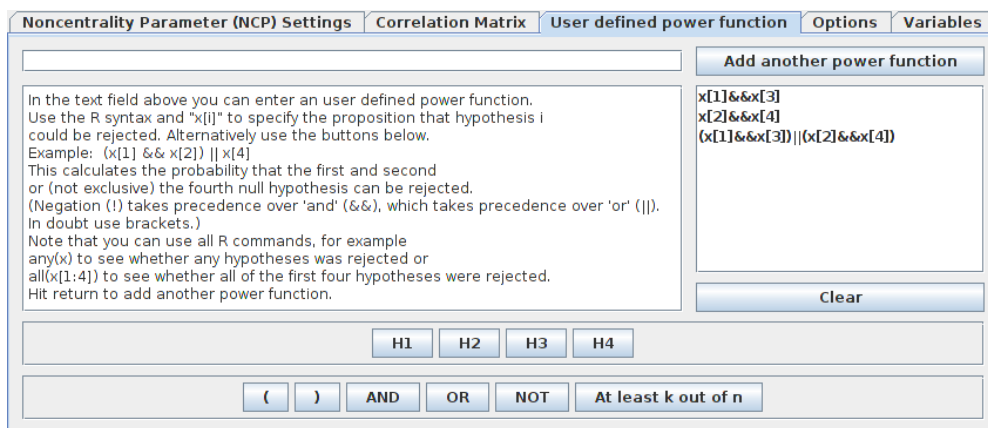You can also create weighted utility functions, where the result is then of course no longer a probability.

Figure 7.2: User defined power functions.

For example in figure 7.2 we define three power functions:

- `x[1]&&x[3]` - Efficacy for treatment 1 can be shown for both endpoints.

- `x[2]&&x[4]` - Efficacy for treatment 2 can be shown for both endpoints.

- `(x[1]&&x[3]) || (x[2]&&x[4])` - For at least one treatment efficacy can be shown for both endpoints. (We could have written this as `x[1] && x[3]||x[2] && x[4]`, but this could easily be misread as the rejection of H1 and H4 and at least one rejection of H2 or H3.)

7.2 Simulation for Non-normal Distributions

If the underlying distribution of the test statistics is not multivariate normal, but test statistics can be simulated and p-values generated, one can still use function `graphTest`.

For example let's take a look at correlated binary data. We assume we have $n_1 = 20$ observations for two binary endpoints and two treatments, where the resulting binomial distributions are not well approximated by the normal distribution. Further we assume a correlation of 0.3 between different endpoints and 0.5 between treatments, i.e. the following correlation matrix with H_1 and H_2 denoting the null hypotheses

Power Results

Scenario	LocalPower H1	LocalPower H2	LocalPower H3	LocalPower H4	ExpRejections
Scenario 1, delta=0.5, gamma=0.1	0.4118	0.1124	0.0431	0.009	0.5762
Scenario 2, delta=0.5, gamma=0.1	0.0	6.0E-4	0.0	0.0	6.0E-4
Scenario 3, delta=0.5, gamma=0.1	0.0125	0.0125	1.0E-4	1.0E-4	0.0253
Scenario 1, delta=0.9, gamma=0.1	0.4156	0.1123	0.0449	0.0043	0.5771
Scenario 2, delta=0.9, gamma=0.1	0.0	6.0E-4	0.0	0.0	6.0E-4
Scenario 3, delta=0.9, gamma=0.1	0.0126	0.0126	2.0E-4	1.0E-4	0.0255
Scenario 1, delta=0.5, gamma=0.5	0.4119	0.1212	0.0302	0.011	0.5742
Scenario 2, delta=0.5, gamma=0.5	0.0	6.0E-4	0.0	0.0	6.0E-4
Scenario 3, delta=0.5, gamma=0.5	0.0126	0.0126	1.0E-4	1.0E-4	0.0253
Scenario 1, delta=0.9, gamma=0.5	0.4156	0.1213	0.0339	0.0052	0.5759
Scenario 2, delta=0.9, gamma=0.5	0.0	6.0E-4	0.0	0.0	6.0E-4
Scenario 3, delta=0.9, gamma=0.5	0.0126	0.0126	1.0E-4	0.0	0.0253
Scenario 1, delta=0.5, gamma=0.8	0.4119	0.1277	0.0176	0.0137	0.5709
Scenario 2, delta=0.5, gamma=0.8	0.0	6.0E-4	0.0	0.0	6.0E-4
Scenario 3, delta=0.5, gamma=0.8	0.0125	0.0126	1.0E-4	1.0E-4	0.0253
Scenario 1, delta=0.9, gamma=0.8	0.4157	0.1279	0.0217	0.0077	0.573
Scenario 2, delta=0.9, gamma=0.8	0.0	5.0E-4	0.0	0.0	5.0E-4
Scenario 3, delta=0.9, gamma=0.8	0.0125	0.0126	0.0	1.0E-4	0.0252

```
m <- rbind(H1=c("0", "\\gamma", "1-\\gamma", "0"),
           H2=c("\\delta", "0", "0", "1-\\delta"),
           H3=c("0", "1.0", "0", "0"),
           H4=c("1.0", "0", "0", "0"))
weights <- c(0.5, 0.5, 0, 0)
graph <- new("graphMCP", m=m, weights=weights)

ncpL <- list('Scenario 1'=c(2, 1, 1, 1),
             'Scenario 2'=c(-2, -1, -1, -1),
             'Scenario 3'=c(0.0, 0.0, 0.0, 0.0))

f <- list('x[1] & x[2]'=function(x) {x[1] & x[2]},
          'x[3]'=function(x) {x[3]})

variables <- list(delta = c(0.5, 0.9), gamma = c(0.1, 0.5, 0.8))

corr.sim <- rbind(c(1, 0, 0, 0),
                  c(0, 1, 0, 0),
                  c(0, 0, 1, 0),
                  c(0, 0, 0, 1))

set.seed(1234)

df <- c()
for (g in replaceVariables(graph, variables, list=TRUE)) {
  result <- calcPower(graph=g, mean=ncpL, f=f, type="quasirandom", corr.sim=corr.sim, alpha=0.025, n.sim=10000)
  df <- rbind(df, matrix(unlist(result), nrow=3, byrow=TRUE))
}
print(round(df,4))
```

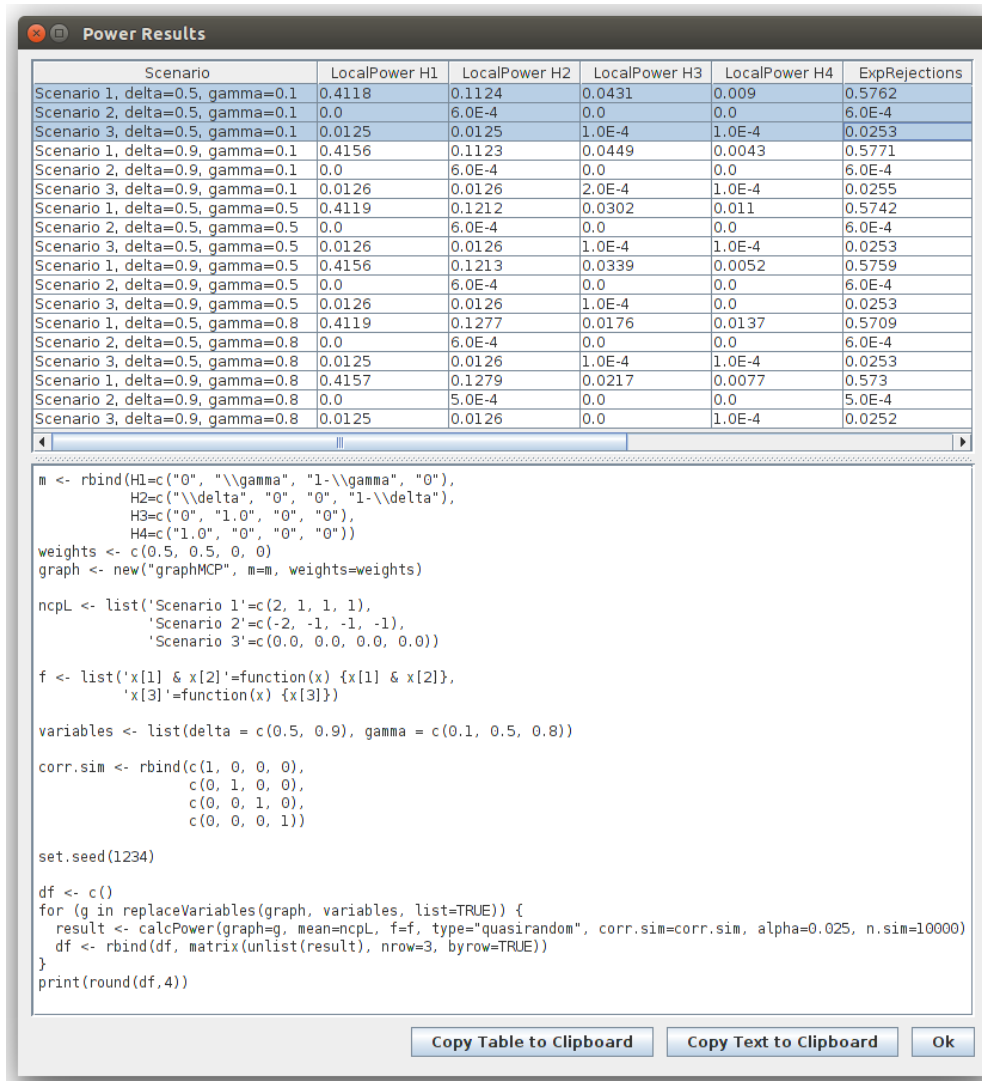Copy Table to Clipboard   Copy Text to Clipboard   Ok

Figure 7.3: Power results

that under treatment 1 and 2 we see the problematic outcome with probability not less than the known probability of $p_0 = 0.5$ in an untreated group. The hypotheses H_3 and H_4 denote the same for the secondary endpoint.

```
cr <- rbind(H1=c(1    , 0.5 , 0.3 , 0.15),
            H2=c(0.5 , 1    , 0.15, 0.3 ),
            H3=c(0.3 , 0.15, 1    , 0.5 ),
            H4=c(0.15, 0.3 , 0.5 , 1   ))
```

We simulate $n_2 = 1000$ times p-values with function `rmvbin` from package `bindata` (Leisch et al. [2012]):

```
library(bindata)

n1 <- 20
n2 <- 1e4

pvals <- t(replicate(n2,
               sapply(colSums(rmvbin(n1, margprob = c(0.35, 0.4, 0.25, 0.3), bincorr = cr)),
                   function(x, ...) {binom.test(x, ...)$p.value}, n=n1, alternative="less")
               ))
```

And then apply function `graphtest` and `extractPower`:

```
graph <- generalSuccessive(gamma=0, delta=0)
out <- graphTest(pvalues=pvals, graph = graph)
extractPower(out)

## $LocalPower
##     H1     H2     H3     H4
## 0.2479 0.1292 0.1869 0.0824
##
## $ExpRejections
## [1] 0.6464
##
## $PowAtlst1
## [1] 0.3043
##
## $RejectAll
## [1] 0.0393
```

7.3   Using the R command line

```
# Bonferroni adjustment
G <- diag(2)
weights <- c(0.5,0.5)
corMat <- diag(2)+matrix(1,2,2)
theta <- c(1,2)
calcPower(weights, alpha=0.025, G, theta, corMat)

## $LocalPower
##     H1     H2
## 0.1898 0.4321
##
## $ExpRejections
## [1] 0.6219
##
## $PowAtlst1
## [1] 0.4841
##
## $RejectAll
## [1] 0.1379

calcPower(weights, alpha=0.025, G, 2*theta, 2*corMat)

## $LocalPower
##     H1     H2
## 0.4521 0.8103
##
## $ExpRejections
## [1] 1.262
##
## $PowAtlst1
## [1] 0.8436
##
## $RejectAll
## [1] 0.4188
```
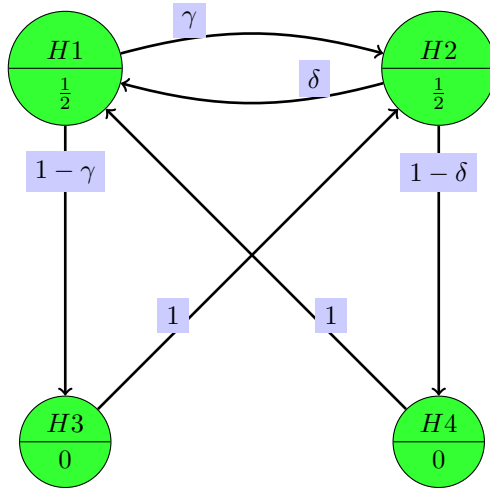
Figure 7.4: Graph from Bretz et al. (2009)

7.4 Variable edge weights

Apart from latin letters the following greek letters can be used to name a variable[1]. 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, \phi, \chi, \psi, \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	γ	1-\gamma	0
H2	δ	0	0	1-δ
H3	0	1	0	0
H4	1	0	0	0

```
graph <- generalSuccessive()
graph

## A graphMCP graph
## H1 (weight=0.5)
## H2 (weight=0.5)
## H3 (weight=0)
## H4 (weight=0)
## Edges:
## H1  -( \gamma )->  H2
## H1  -( 1-\gamma )->  H3
## H2  -( \delta )->  H1
## H2  -( 1-\delta )->  H4
## H3  -( 1 )->  H2
## H4  -( 1 )->  H1
```

---

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

Chapter 8

Options and Import/Export

8.1   Options

8.1.1   Visual

Grid For easier placement of nodes a grid can be used that aligns the nodes to its intersections. You can specify a positive integer that sets the grid size, i.e. the width in pixels between two proximate parallel lines. If you set the grid size to 1 this would allow unrestricted placement and therefore disables the grid.

Number of digits Number of digits to be shown at various places. In this version not every part of the GUI will use this value, but this will improve in further versions.

Line width Especially if you want to use exported PNG graphics in other documents, you may want to adjust the line width of edges and nodes, when borders look to thin or thick.

Font Size Font size of the text in the GUI widgets.

Look'n'Feel The way the widgets of a GUI look and how they behave is called "look and feel" in Java. Depending on your operating system and classpath several Look'n'Feel implementations may be available (e.g. Metal (Java default), Windows, Mac OS, Motif and/or System/GTK). If you are used to a particular Look'n'Feel, you can select it here. But if you have problems with the graphical interface, please try to use the default Metal theme to check whether it could be a problem with the selected Look'n'Feel.

Colored image files and pdf reports Colors are used to highlight different conditions in the graph like hypotheses that could be rejected. While these colors are helpful in the GUI, you perhaps prefer black and white PNG image files and LaTeX/docx reports.

Show rejected nodes in GUI When using the GUI to for stepwise rejection of hypotheses, this options determines whether rejected nodes should "disappear" or whether they remain on the screen and are only marked as rejected.

Use JLaTeXMath There are not many 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 Floating point numbers are used for all calculations and values like 1/3 would be normally shown as 0.3333333. When this option is active the method

27

fractions from package `MASS` is used to display fractions whenever the floating point numbers are close to a fraction that looks right.

Show epsilon edges as dashed lines You can set whether epsilon edges should been shown as dashed or solid lines.

8.1.2 Numeric

Use epsilon approximation In this version this option can not be changed. No calculations with infinitesimal small values are done but instead the epsilon is approximated by a small real number.

Epsilon The small real value that should be used to approximate the infinitesimal small epsilon. Default is 10^{-3}.

Verbose output of algorithms If selected the algorithms produce a verbose output that is shown in the GUI. For example the Simes test specifies for each intersection of elementar hypotheses whether and why it could be rejected.

Monte Carlo sample size for power The Monte Carlo sample size for power calculations. Default is 10000.

Type of random numbers You can select quasirandom or pseudorandom numbers for power calculations. The quasirandom option uses a randomized Lattice rule, and should be more efficient than the pseudorandom option that uses ordinary (pseudo) random numbers.

Weights of subgraphs are upscaled to 1 If '*No*' is selected then for each intersection of hypotheses (i.e. each subgraph) a weighted test is performed at the possibly reduced level alpha of $\text{sum}(w) * alpha$, where $\text{sum}(w)$ is the sum of all node weights in this subset If '*Yes*' is selected all weights are upscaled, so that $\text{sum}(w) = 1$.

Use random number seed If selected a user specified seed is used to set the random number generator state to the specified value. This way all calculations involving random numbers give a reproducable result.

Random number seed Integer seed value to use for random number generation.

8.1.3 Miscellaneous

Check online for updates On start-up gMCP can check automatically whether a new version of gMCP is available. Only your version of R (like 3.2.1), the version of gMCP (like 0.8-7) and a random number (to distinguish different requests) are transmitted.

Export images with transparent background If checked the background of exported PNG graphics will be transparent. Otherwise the graphs are displayed on a white background.

If a node is dragged also all edges to this node follow If selected the edges will always repositioned whenever a node is dragged. Otherwise only newly added eges behave that way and edges that have been dragged themselves are considered "anchored" and will stay with the edge weight label at the same position.

Automatically enter the editing mode, whenever a table cell gets the focus People are used to different behaviour of tables (mostly depending on which spreadsheet applications they use regularly). If this option is set to true it is easy to change the values of the cells but navigating with arrow keys

is hard since in the editing mode the right and left key will move the cursor only in the currently selected cell.

Enable highly experimental feature The gMCP GUI often contains new features that are not that well tested. If you want to use or take a look at them, activate this option. But be prepared that things might go wrong.

Show R code to reproduce results in R After performing a test in the GUI the result dialog does not only show the pure results, but also R code to reproduce these results in R. If you are not interested in this feature you can disable it here.

Save config files Many settings and options are automatically saved and restored between different gMCP sessions (using Java's Properties API). But more complicated settings (e.g. for power or sample size calculations) need to be saved as files. This can also be done automatically, but the user must first agree that non-temporary files are created and has to specify the directory, where these files should be placed.

8.2 Import/Exports

This subsection is work in progress, but fortunately the menu entries in figure 8.1 should be fairly self-explanatory.
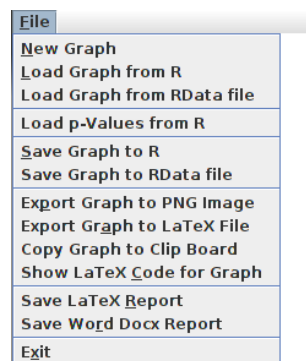
Figure 8.1: Import and export of graphs.

8.2.1 PNG Image Export

You can export graphs to PNG files with the dialog shown in figure 8.2.

The background of these png files can (and normally should) 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. In some Windows application the transparent background may be wrongly shown as black - if this happens or you generally prefer a non-transparent white background, you can toggle the option *"transparent background"*.

Further you can choose between an black and white or colored graph.

The options *"draw names of hypotheses"*, *"draw weights of hypotheses"* and *"draw weights of edges"* should be self-explanatory. Also the live preview at left side of the options shows their effects when selected/deselected.

The node radius can be changed, which is especially useful if you disable the previous options and add all labels by yourself afterwards (which might have a different size).
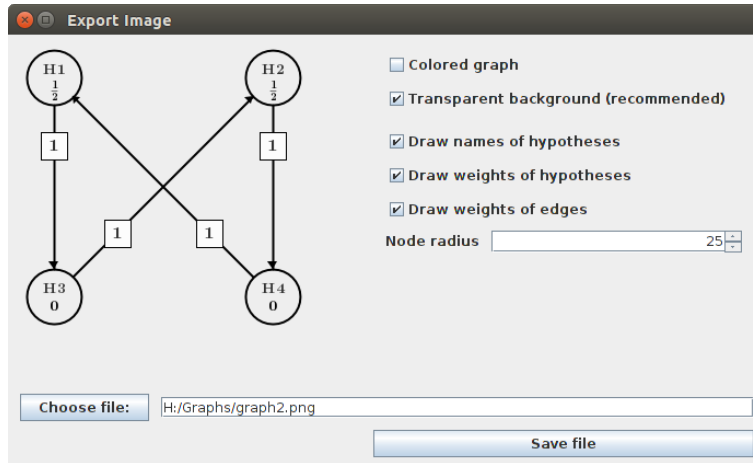
Figure 8.2: Export options for PNG image files.

8.2.2 Word Docx Reports

Since version 0.8-7 gMCP also can create word documents (Office Open XML / docx) with an example page seen in figure 8.3.

Figure 8.3: Example Word export - first page.

8.3 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 automatically. If the result of `graph2latex` does not give you an acceptable layout, simply load the graph into the GUI and use the mouse to drag the edge labels around until you are satisfied with the placement. Save the graph and the TikZ output will be pretty close to the graph seen in the GUI.

PGF/TikZ is very useful LaTeX-package, we recommend it for many purposes and it's totally worth reading its 560 pages manual (Tantau [2008]), but if you don't have the time right now, we understand and therefore will give you a short overview of the most important commands for our kind of graphs, so that you can easily adapt the output from `graph2latex`.

You also perhaps want to use an TikZ editor with a preview pane like qtikz or ktikz[1].

Let's start with this graph in figure 8.4:

```
\begin{tikzpicture}[scale=1]
\node (H11) at (200bp,200bp) [draw,circle split,fill=green!80] {$H11$ \nodepart{lower} $0.0333$};
...
\draw [->,line width=1pt] (H11) to[bend left=15] node[near start,above,fill=blue!20] {0.667} (H12);
...
\end{tikzpicture}
```
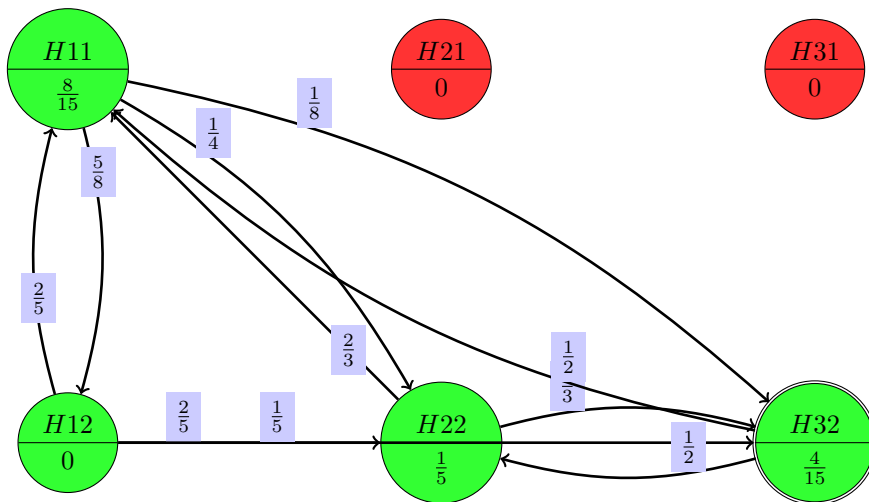

Figure 8.4: Graph from `graph2latex` that does not look optimal.

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 from Kern [2007].

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 (Tantau [2008]).

---

[1] http://www.hackenberger.at/blog/ktikz-editor-for-the-tikz-language/

Chapter 9

Case Studies

This section is work in progress.

9.1 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 from Bretz et al. [2003] with the same title.

9.2 Testing strategies in multi-dose experiments including active control

Bauer et al. [1998]

```
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, quietly=TRUE)
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
```
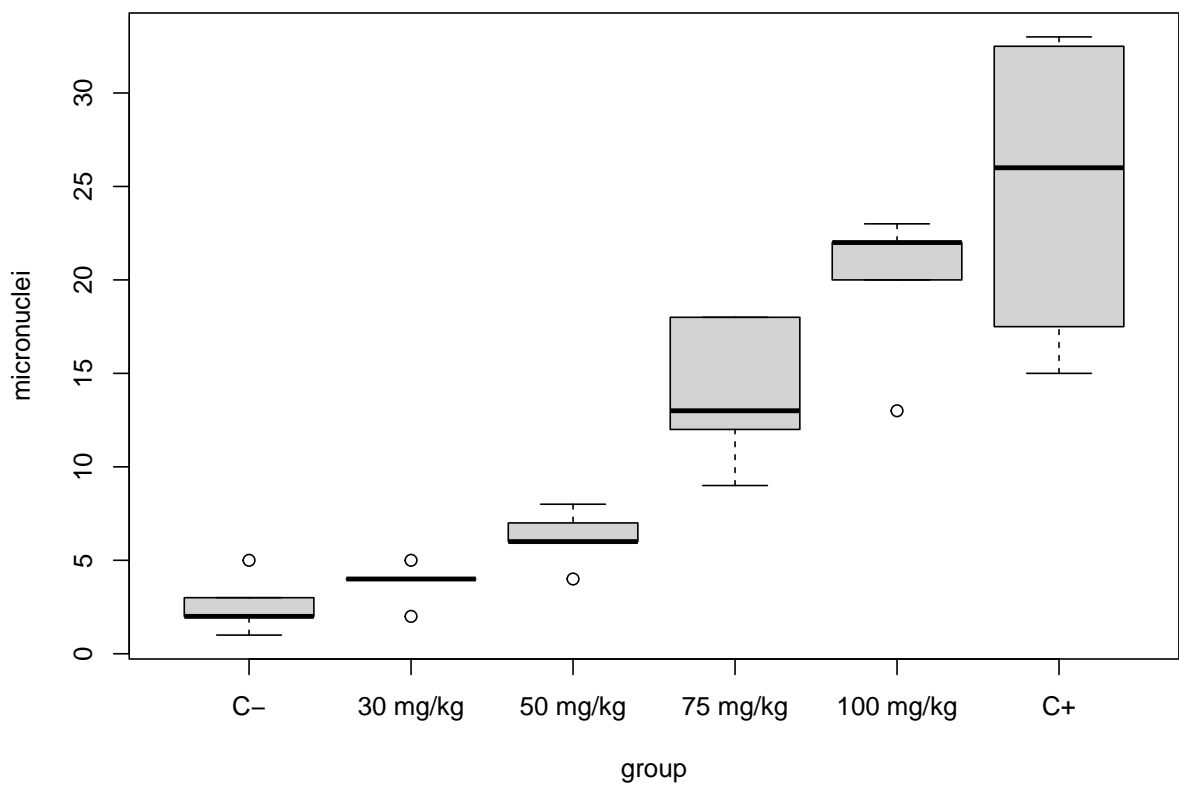
Figure 9.1: Boxplot of the hydroquinone data set

Appendix A

Appendix - Example graphs
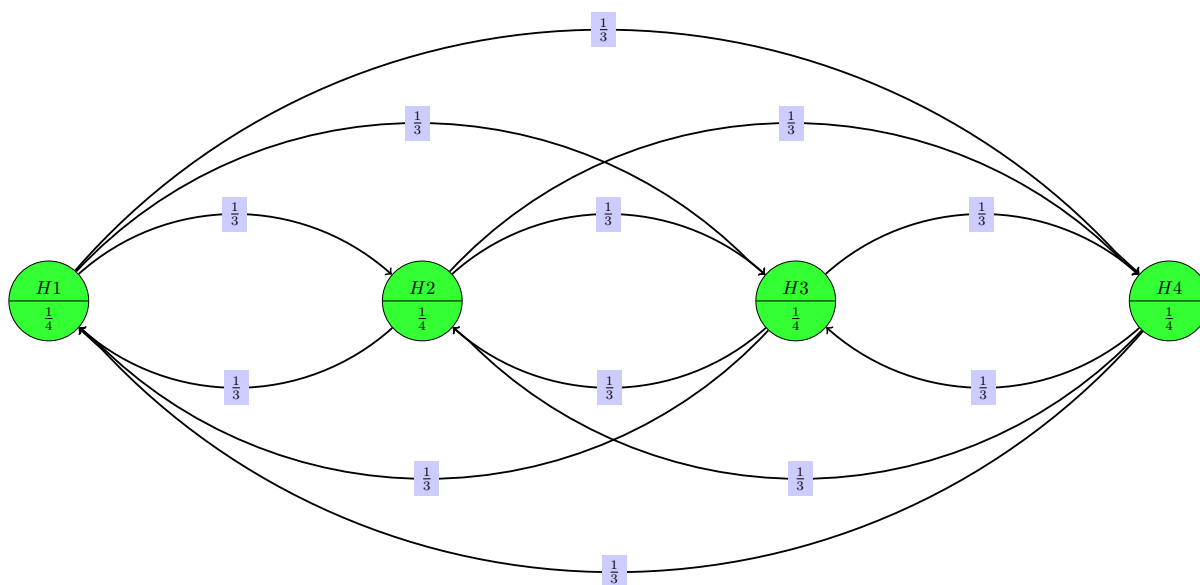
Figure A.1: Graph representing the (unweighted) Bonferroni-Holm-Procedure

The graph is a complete graph, where all nodes have the same weights and each edge weight is $1/(n-1)$.

Literature: Holm, S. (1979). A simple sequentally rejective multiple test procedure. Scandinavian Journal of Statistics 6, 65-70.
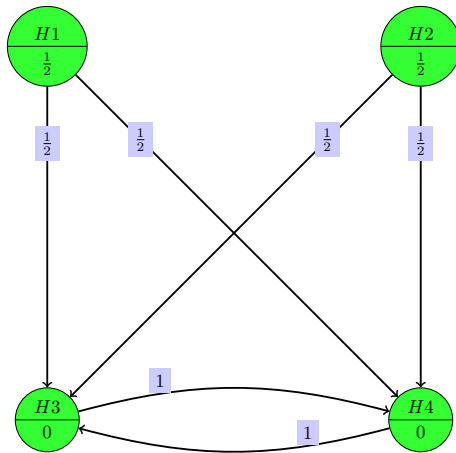
Figure A.2: Graph representing a parallel gatekeeping procedure from Dmitrienko et al. (2003) Table I

Literature: Dmitrienko, A., Offen, W., Westfall, P.H. (2003). Gatekeeping strategies for clinical trials that do not require all primary effects to be significant. Statistics in Medicine. 22, 2387-2400.
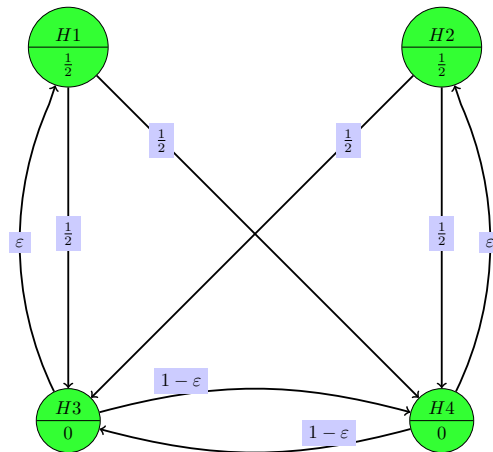
Figure A.3: Graph representing an improved parallel gatekeeping procedure

Literature: Bretz, F., Maurer, W., Brannath, W., Posch, M.: A graphical approach to sequentially rejective multiple test procedures. Statistics in Medicine 2009 vol. 28 issue 4 page 586-604. URL: https://www.meduniwien.ac.at/fwf_adaptive/papers/bretz_2009_22.pdf .
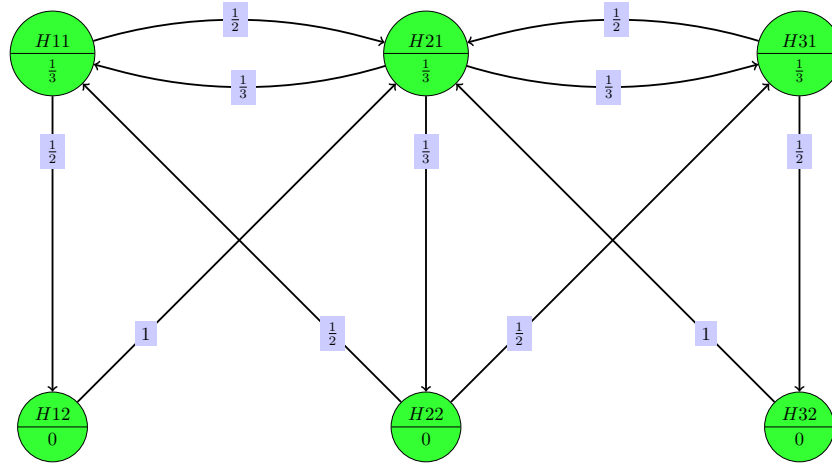
Figure A.4: Graph representing the procedure from Bretz et al. (2011) - Figure 2

H11, H21 and H31 represent three primary hypotheses and H21, H22 and H23 the associated secondary hypotheses.

A secondary hypothesis is only tested if the associated primary hypotheses is rejected.

Since in this example it is preferred to reject two adjacent hypotheses (like H11 and H21 instead of H11 and H31) there are only edges between adjacent nodes.

Literature: Bretz, F., Maurer, W. and Hommel, G. (2011), Test and power considerations for multiple endpoint analyses using sequentially rejective graphical procedures. Statistics in Medicine, 30: 1489-1501.
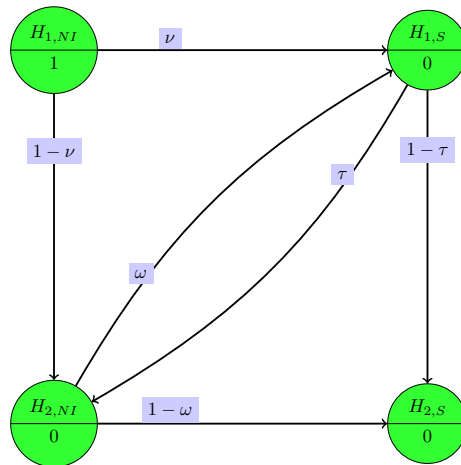
Figure A.5: Graph representing the procedure from Hung and Wang (2010)

$H_{1,NI}$: Non-inferiority of the primary endpoint
$H_{1,S}$: Superiority of the primary endpoint
$H_{2,NI}$: Non-inferiority of the secondary endpoint
$H_{2,S}$: Superiority of the secondary endpoint

Literature: Hung H.M.J., Wang S.-J. (2010). Challenges to multiple testing in clinical trials. Biometrical Journal 52, 747-756.
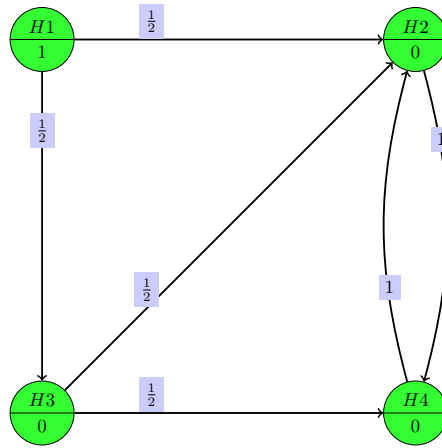
Figure A.6: Graph representing the procedure from Huque, Alosh and Bhore (2011)

Literature: Huque, M.F. and Alosh, M. and Bhore, R. (2011), Addressing Multiplicity Issues of a Composite Endpoint and Its Components in Clinical Trials. Journal of Biopharmaceutical Statistics, 21: 610-634.
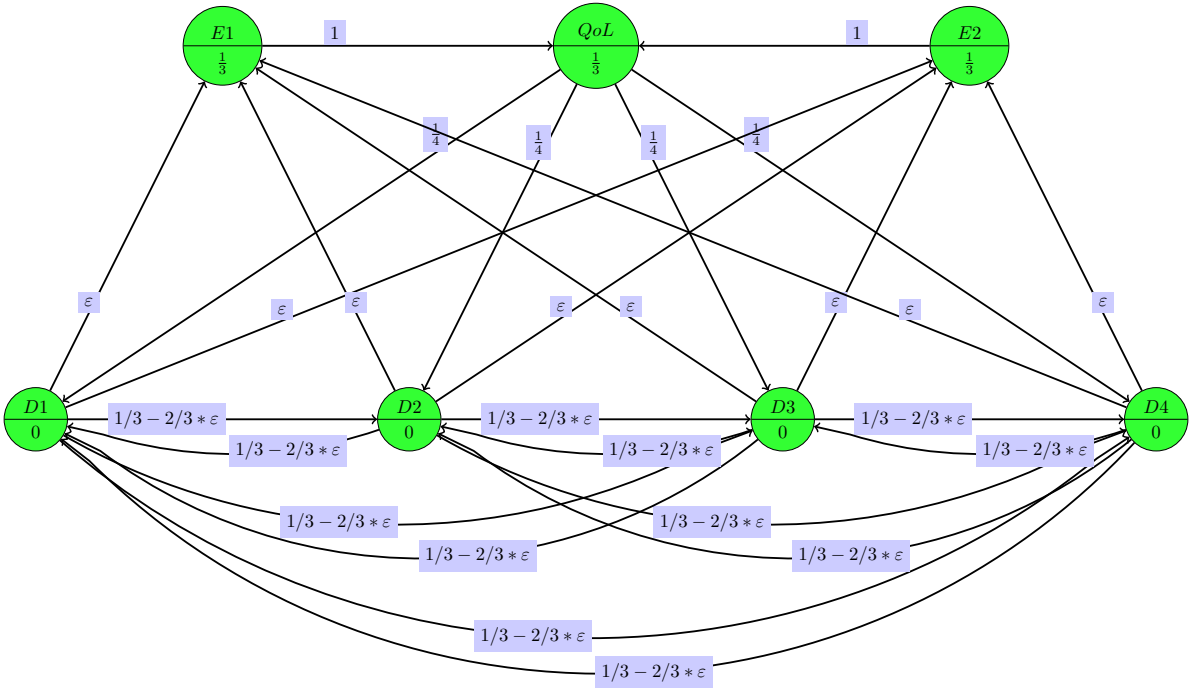
Figure A.7: Graph representing the procedure from Hommel et al. (2007)

In this clinical trial example three primary endpoints are investigated: QoL (Quality of Life score), E1 and E2.
If QoL is rejected, four secondary hypotheses D1, D2, D3 and D4 are also be tested.

Literature: Hommel, G., Bretz, F. und Maurer, W. (2007). Powerful short-cuts for multiple testing procedures with special reference to gatekeeping strategies. Statistics in Medicine, 26(22), 4063-4073.
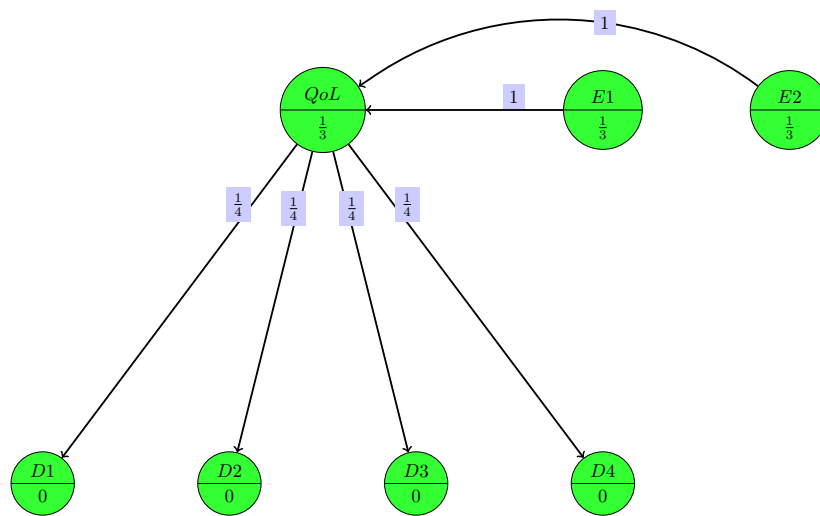
Figure A.8: Simplified graph representing the procedure from Hommel et al. (2007)

In this clinical trial example three primary endpoints are investigated: QoL (Quality of Life score), E1 and E2.
If QoL is rejected, four secondary hypotheses D1, D2, D3 and D4 are also be tested.

Literature: Hommel, G., Bretz, F. und Maurer, W. (2007). Powerful short-cuts for multiple testing procedures with special reference to gatekeeping strategies. Statistics in Medicine, 26(22), 4063-4073.
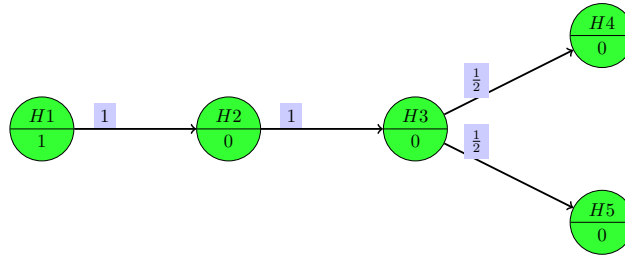
Figure A.9: Graph representing a procedure in drug clinical trials (from Maurer et al. 1995, Scenario 1)

In a univariate one-way design a drug A is compared against placebo and two positive control drugs B and C.

The order of importance is that first the sensitivity has to be shown, i.e. that drug B and C are better than placebo. Than the efficacy of A vs. placebo is tested and if this can be shown, it is tested (with Bonferroni correction) whether A is superior to drug B and/or C.

These hypotheses are represented in the graph as follows:
H1: drug B better than placebo
H2: drug C better than placebo
H3: drug A better than placebo
H4: drug A better than drug B
H5: drug A better than drug C

(Maurer et al. apply the intersection-union principle to H1 and H2 to test sensitivity, so sensitivity is shown if and only if H1 and H2 are both rejected.)

Note that you could improve the test procedure by using a Bonferroni-Holm correction instead of the Bonferroni correction in the last step by adding an edge from H4 to H5 with weight 1 and vice versa.

Literature:
W. Maurer, L. Hothorn, W. Lehmacher: Multiple comparisons in drug clinical trials and preclinical assays: a-priori ordered hypotheses. In Biometrie in der chemisch-pharmazeutischen Industrie, Vollmar J (ed.). Fischer Verlag: Stuttgart, 1995; 3-18.
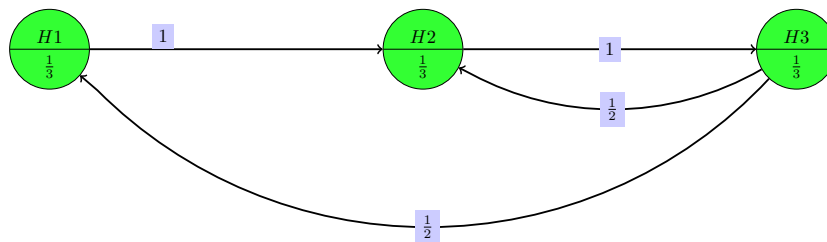
Figure A.10: Improved Fallback Method I by Wiens & Dmitrienko

Literature: B.L. Wiens, A. Dmitrienko (2005): The fallback procedure for evaluating a single family of hypotheses. Journal of Biopharmaceutical Statistics 15:929-942.

Bretz, F., Maurer, W. and Hommel, G. (2011), Test and power considerations for multiple endpoint analyses using sequentially rejective graphical procedures. Statistics in Medicine, 30: 1489-1501.
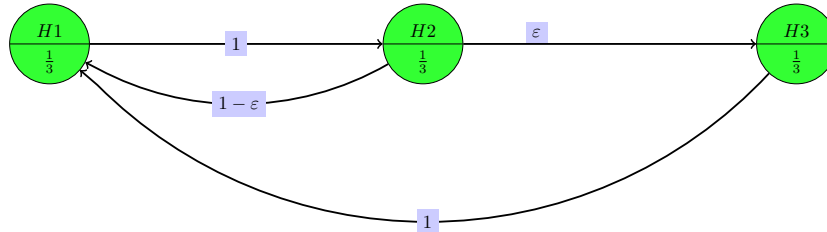
Figure A.11: Improved Fallback Method II by Hommel & Bretz

Literature: G. Hommel, F. Bretz (2008): Aesthetics and power considerations in multiple testing - a contradiction? Biometrical Journal 50:657-666.

Bretz, F., Maurer, W. and Hommel, G. (2011), Test and power considerations for multiple endpoint analyses using sequentially rejective graphical procedures. Statistics in Medicine, 30: 1489-1501.
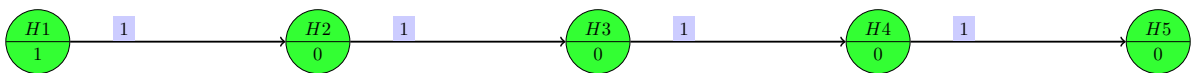
Figure A.12: Graph representing the fixed sequence test

Literature: Maurer W., Hothorn L., Lehmacher W.: Multiple comparisons in drug clinical trials and preclinical assays: a-priori ordered hypotheses. In Biometrie in der chemisch-pharmazeutischen Industrie, Vollmar J (ed.). Fischer Verlag: Stuttgart, 1995; 3-18.

Westfall P.H., Krishen A.: Optimally weighted, fixed sequence, and gatekeeping multiple testing procedures. Journal of Statistical Planning and Inference 2001; 99:25-40.
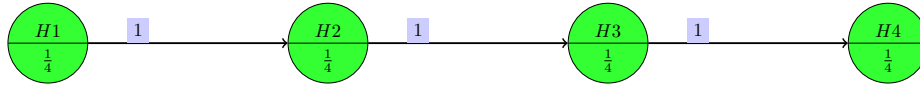
Figure A.13: Graph representing the fallback (a fixed sequence Bonferroni) procedure

Literature: Wiens B.L.: A fixed sequence Bonferroni procedure for testing multiple endpoints. Pharmaceutical Statistics 2003; 2: 211-215.

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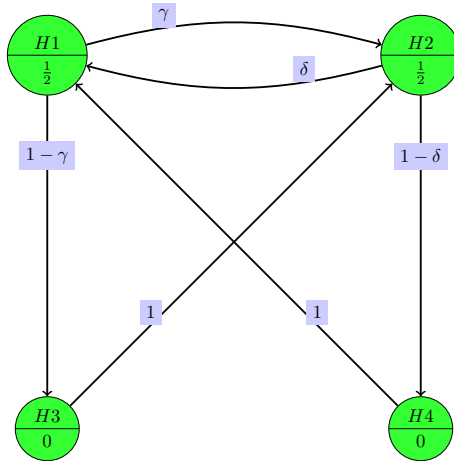

Figure A.14: General successive graph from Bretz et al. (2011), Figure 6

Literature: Bretz, F., Maurer, W. and Hommel, G. (2011), Test and power considerations for multiple endpoint analyses using sequentially rejective graphical procedures. Statistics in Medicine, 30: 1489-1501.
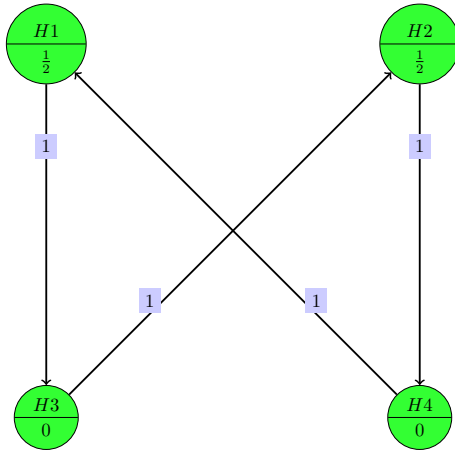
Figure A.15: Simple successive graph from Maurer et al. (2011)

Literature: Maurer W., Glimm E., Bretz F.: Multiple and repeated testing of primary, co-primary and secondary hypotheses. Statistics in Biopharmaceutical Reserach 2011; (in press).

F. Bretz, M. Posch, E. Glimm, F. Klinglmueller, W. Maurer, K. Rohmeyer (2011), Graphical approaches for multiple comparison procedures using weighted Bonferroni, Simes or parametric tests. To be published. Figure 12.
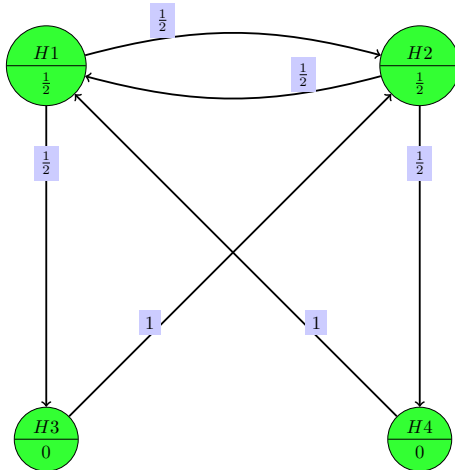
Figure A.16: Simple successive graph from Maurer et al. (2011)

Literature: Maurer W., Glimm E., Bretz F.: Multiple and repeated testing of primary, co-primary and secondary hypotheses. Statistics in Biopharmaceutical Reserach 2011; (in press).
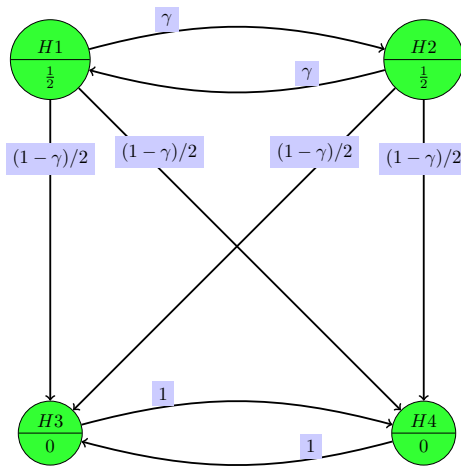
Figure A.17: Example of the Truncated Holm Procedure

Literature: Dmitrienko A, Tamhane A, Wiens B. General multi-stage gatekeeping procedures. Biometrical Journal 2008; 50:667-677.

F. Bretz, M. Posch, E. Glimm, F. Klinglmueller, W. Maurer, K. Rohmeyer (2011), Graphical approaches for multiple comparison procedures using weighted Bonferroni, Simes or parametric tests. To be published. Figure 5.
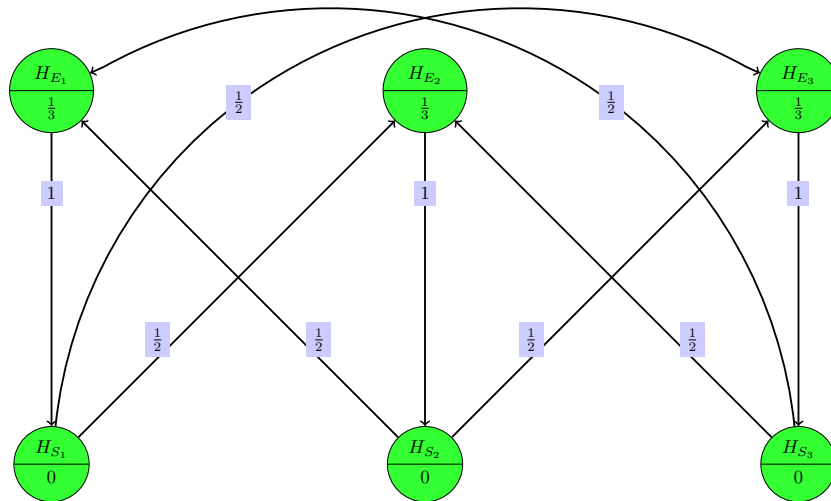
Figure A.18: Graph representing the procedure from Bretz et al. (2011) - Figure 2

Literature: Bauer P., Brannath W., Posch M.: Multiple testing for identifying effective and safe treatments. Biometrical Journal 2001; 43:605-616.

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 8.
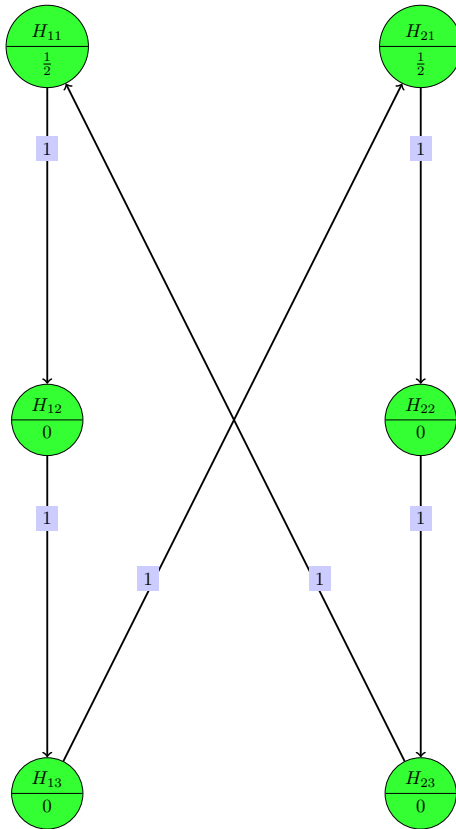
Figure A.19: Graph from Bretz et al. (2009) - Figure 14

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.
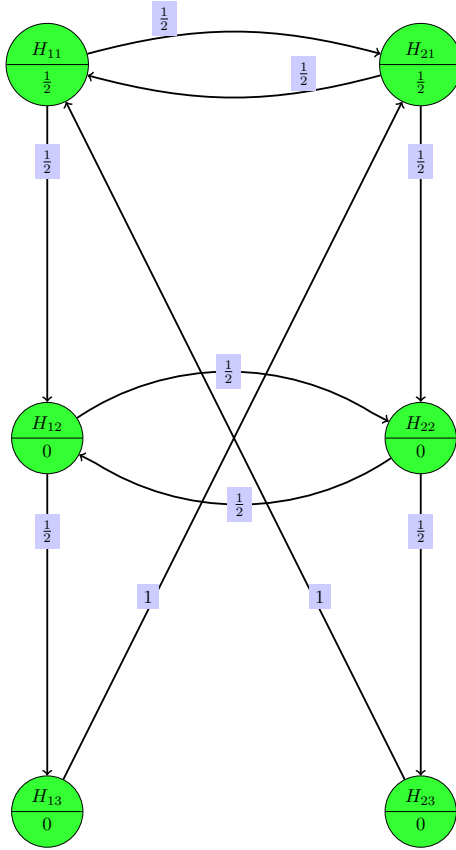
Figure A.20: Graph from Bretz et al. (2009) - Figure 14

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.
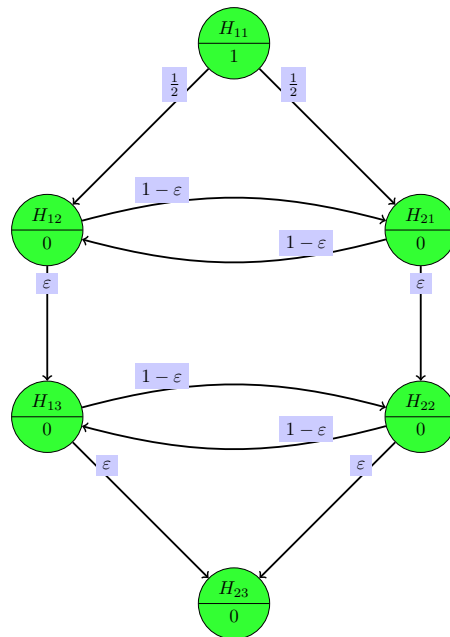
Figure A.21: Graph from Bretz et al. (2009) - Figure 15

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 15.
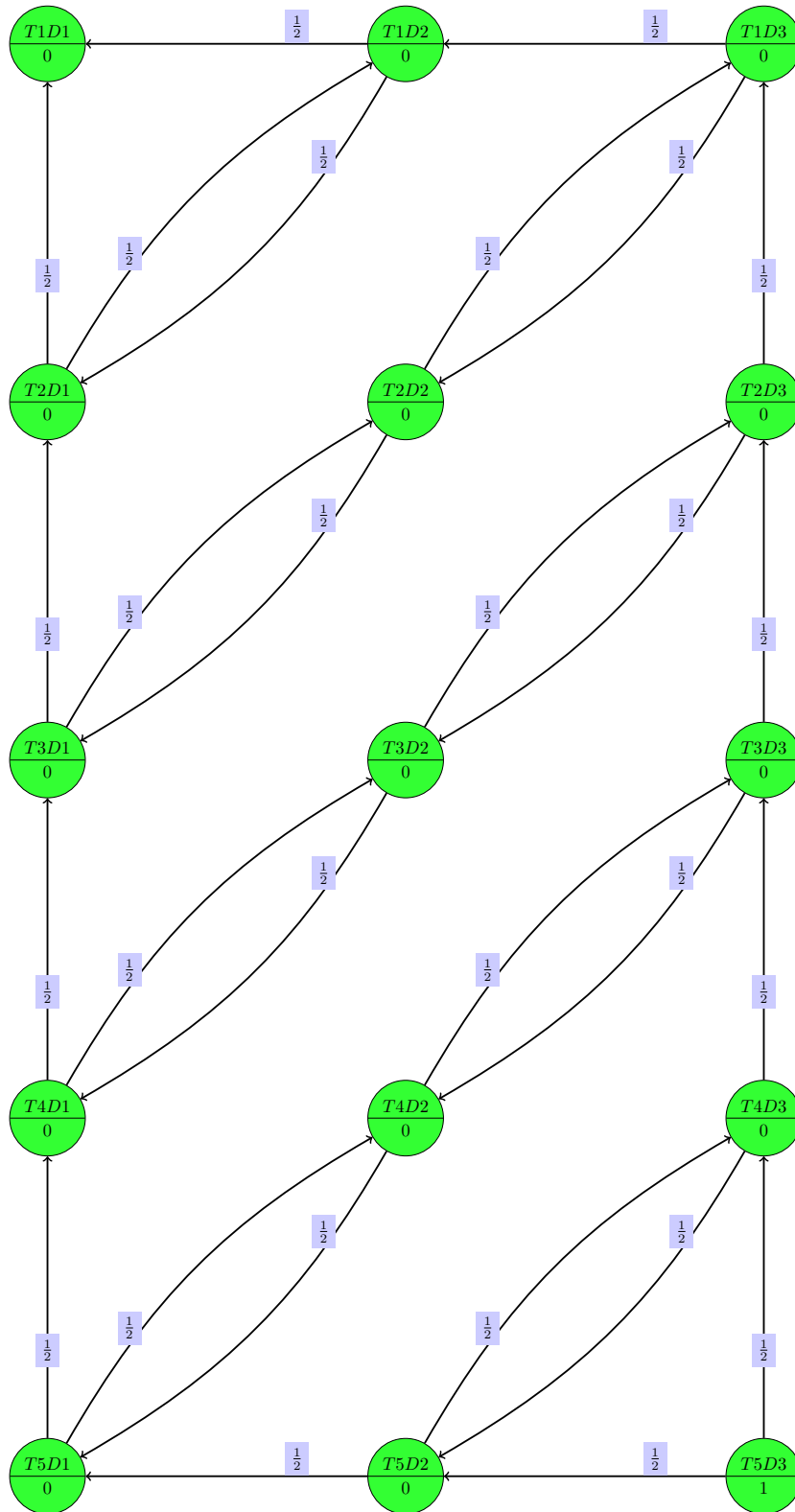
Figure A.22: Second graph from Ferber et al. 2011

Literature: G. Ferber, L. Staner and P. Boeijinga (2011): Structured multiplicity and confirmatory statistical analyses in pharmacodynamic studies using the quantitative electroencephalogram, Journal of neuroscience methods, Volume 201, Issue 1, Pages 204-212.
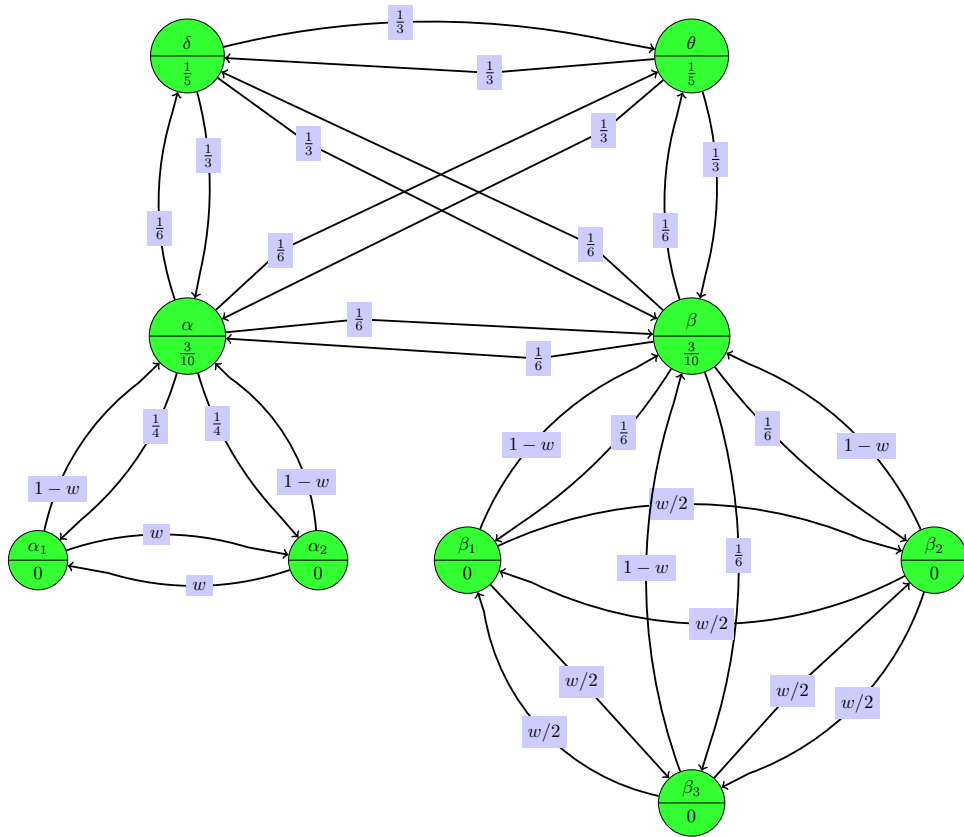
Figure A.23: Graph from Ferber et al. 2011
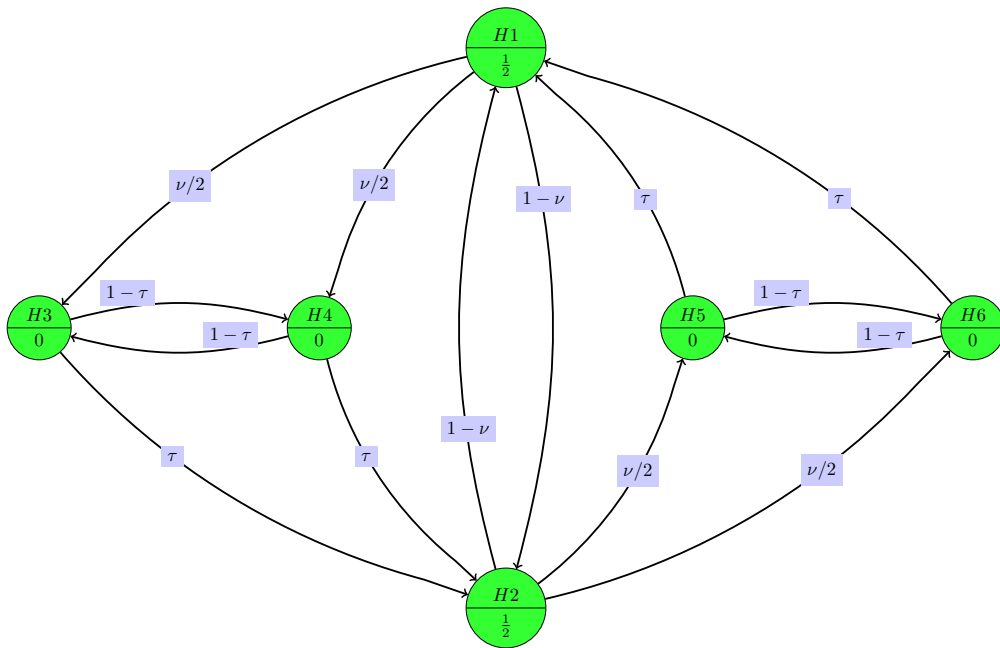
Figure A.24: Graph from Wang and Ting 2014
With ν=1, τ=0.5 graph from figure 3,
with ν=1, τ=ε graph from figure 4 and
with ν=ε, τ=ε graph from figure 5.

Literature: Wang, B., Ting, N. (2014). An Application of Graphical Approach to Construct Multiple Testing Procedures in a Hypothetical Phase III Design. Frontiers in public health, 1 (75).
URL: `https://journal.frontiersin.org/Journal/10.3389/fpubh.2013.00075/full`

Appendix B

Appendix - Multiple Testing

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 $H_i \subset \Theta$, $i \in \{1, \ldots, n\} =: I$ a multiple test procedure φ is defined as a family of $(\mathcal{F}, \mathfrak{Pot}(\{0,1\}^n))$-measurable functions $\{\varphi_J : \Omega \to \{0,1\}^n \mid J \subset I\}$. (We'll write φ_j for $\varphi_{\{j\}}$).

The family of hypotheses $\{H_i \mid i \in I\}$ is called *closed* if it is closed under intersection.

Definition B.0.1 (Familywise Error Rate). 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

$$\forall \theta \in \Theta : \ P_\theta \left(\max_{J \subset I, \theta \in H_J} \varphi_J = 1 \right) \leq \alpha.$$

This section is work in progress.

Theorem B.0.2 (Closed testing principle). *Marcus et al. [1976]*

Definition B.0.3 (Coherence and Consonance). A multiple test procedure is called *consonant* if

$$\forall J \subset I : \ (\varphi_J = 1 \ \Rightarrow \ \exists j \in J : \ \varphi_j = 1).$$

A multiple test procedure is called *coherent* if

$$\forall J, J' \subset I : \ (\varphi_J = 0 \text{ and } J' \subset J \ \Rightarrow \ \varphi_{J'} = 0).$$

The closure principle is based on enforcing coherence.

For further reading see Hochberg and Tamhane [2009] and Gabriel [1969].

Definition B.0.4.

Theorem B.0.5 (Simes-Procedure). *Let $T_1, \ldots, T_m$ be test statistics for $m \in \mathbb{N}$ null hypotheses $H_1, \ldots, H_m$ and $p_1, \ldots, p_m$ the associated p-values and $\alpha \in \ ]0,1[$.*

Denote the ordered p-values by $p^{(1)} < p^{(2)} < \ldots < p^{(m)}$ and the corresponding hypotheses by $H^{(1)}, H^{(2)}, \ldots, H^{(m)}$. 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 B.0.6 (Weighted Simes Procedure)**.** *Benjamini and Hochberg (1997)*

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

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

Theorem B.0.7 (Trimmed weighted Simes test)**.** *For two one-sided tests where the bivariate test statistic fulfills certain conditions, Brannath et al. [2009] show that the following trimmed version of the weighted Simes test with weights $w_A, w_B, w_A + w_B = 1$ is bounded by α:*

H_0: $H_A \cap H_B$ *is rejected if*

- $p_A \leq w_A \cdot \alpha$ *and* $p_B < 1 - w_B \cdot \alpha$ *or*

- $p_B \leq w_B \cdot \alpha_B$ *and* $p_A < 1 - w_A \cdot \alpha$ *or*

- $max(p_A, p_B) < \alpha$

Remark B.0.8. It is not possible to calculate adjusted p-values for the trimmed weighted Simes test, since for example with $w_A = w_B = 0.5$, $p_A = 0.01$, and $p_A = 0.97$ the test does reject H_0 for $\alpha = 0.05$ since $p_A \leq w_A \cdot \alpha = 0.025$ and $p_B < 1 - w_B \cdot \alpha = 0.975$, but not for $\alpha = 0.1$ since $p_B > 1 - w_B \cdot \alpha$ in this case.

Theorem B.0.9 (Hommel Procedure)**.** *Hommel 1988*

Theorem B.0.10 (Hochberg Procedure)**.** *Hochberg 1988*

Adjusted p-values form the Hochberg Procedure are greater as or equal to the Hommel-adjusted p-values.

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-pvalue p'_j is defined as the minimal α so that the test to global level α rejects H_j.

Therefore $p'_j = \max(m_J \mid j \in J)$.

B.1 Parametric Tests

For p-values with a known joint distribution the gMCP package provides weighted min-p test, i.e. the intersection hypotheses H_J, $J \subseteq I$ is rejected if

$$\exists j \in J : \quad p_j \leq c_J w_j(J)\alpha,$$

with c_J the largest constant satisfying

$$P_{H_J} \left(\bigcup_{j \in J} \{p_j \leq c_J w_j(J)\alpha\} \right) \leq \alpha.$$

Appendix C

Appendix - Graph Theory Basics

When we talk about graphs in the context of gMCP we always mean finite, directed, weighted graphs with no self-loops and no parallel edges:

Definition C.0.1. In our context a (valid) *graph* G is a triple $G = (V, E, w)$ of a non-empty, finite set V of nodes together with the set of edges $E \subset (V \times V) \setminus \{(v, v) \mid v \in V\}$ and a mapping $w : V \cup E \to [0, 1]$ that fulfills $\sum_{v \in V} w(v) \leq 1$ and $w(e) > 0$ for each edge $e \in E$.

Isomorphisms.

This section is work in progress.

Index

List of Algorithms

List of Figures

List of Tables

Bibliography

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.

Werner Brannath, Frank Bretz, Willi Maurer, and Sanat Sarkar. Trimmed weighted simes' test for two one-sided hypotheses with arbitrarily correlated test statistics. *Biometrical Journal*, 51(6):885–898, 2009.

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.

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 `http://www.meduniwien.ac.at/fwf_adaptive/papers/bretz_2009_22.pdf`.

F. Bretz, W. Maurer, and G. Hommel. Test and power considerations for multiple endpoint analyses using sequentially rejective graphical procedures. *Statistics in medicine*, 30(13):1489–1501, 2011a.

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, 2011b. URL `http://onlinelibrary.wiley.com/doi/10.1002/bimj.201000239/full`.

K.R. Gabriel. Simultaneous test procedures–some theory of multiple comparisons. *The Annals of Mathematical Statistics*, 40(1):224–250, 1969.

Y. Hochberg and A.C. Tamhane. *Multiple Comparison Procedures*. Wiley Series in Probability and Statistics. John Wiley & Sons, 2009. ISBN 9780470568330. URL `http://books.google.de/books?id=XC5IPwAACAAJ`.

S. Holm. A simple sequentially rejective multiple test procedure. *Scand. J. Statist.*, 6:65–70, 1979.

U. Kern. *Extending LaTeX's color facilities: the xcolor package*, 2007. URL `http://www.ctan.org/tex-archive/macros/latex/contrib/xcolor/`.

Friedrich Leisch, Andreas Weingessel, and Kurt Hornik. *bindata: Generation of Artificial Binary Data*, 2012. URL `http://CRAN.R-project.org/package=bindata`. R package version 0.9-19.

R. Marcus, P. Eric, and K.R. Gabriel. On closed testing procedures with special reference to ordered analysis of variance. *Biometrika*, 63(3):655, 1976. ISSN 0006-3444.

W. Maurer and F. Bretz. Memory and other properties of multiple test procedures generated by entangled graphs. *Statistics in medicine*, 32(10):1739–1753, 2013.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014. URL `http://www.R-project.org/`.

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)
$\mathfrak{Pot}(X)$	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. $\mathrm{id}_X : X \to X, \ x \mapsto x$