Example1

```r
library(gMCP)
graph <- BonferroniHolm(3)
pvalues <- c(0.01,0.07,0.02)
gMCP(graph, pvalues, 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.
```

```r
# graphGUI(graph)
```

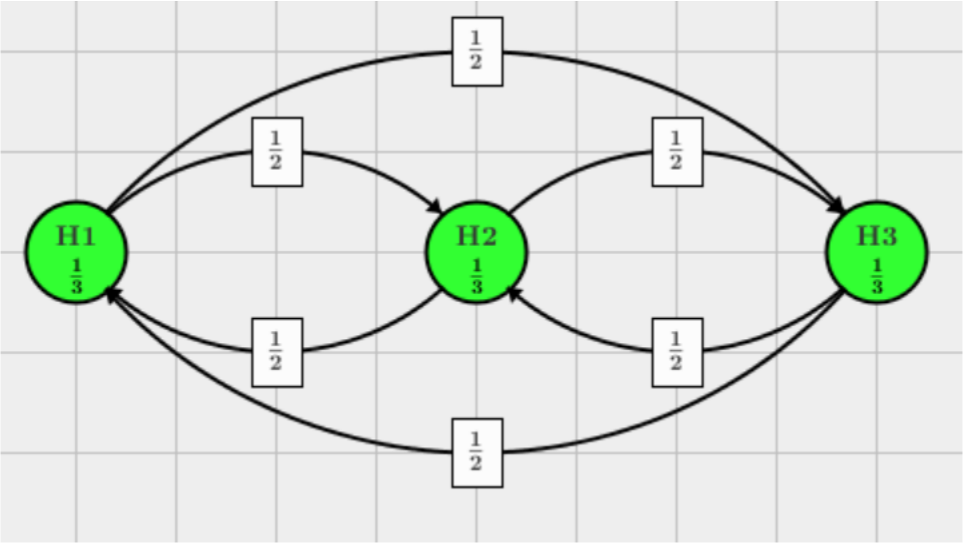


Figure 1: Example1 GUI

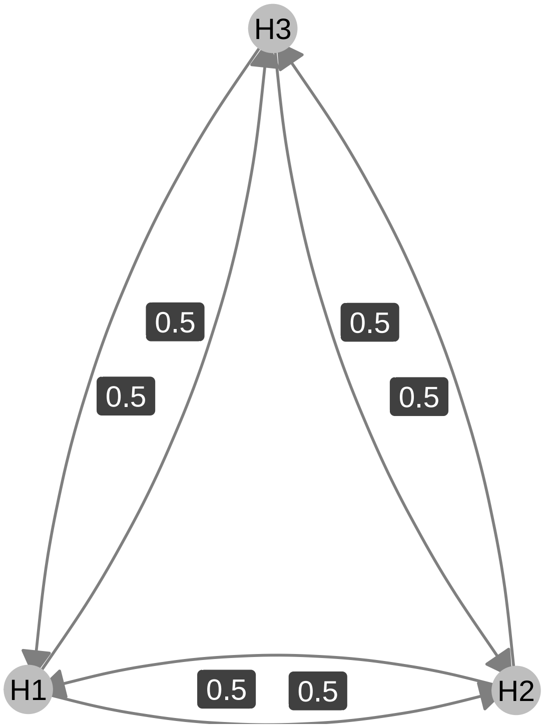
```
source("./R/gMCP_xc2.R")
weights <- graph@weights
matrix <- graph@m
gMCP_xc2(matrix,weights,
         pvalues, alpha = 0.05,fweights = F)
```

```
## $pvalues
## [1] 0.01 0.07 0.02
##
## $adjpvalues
## [1] 0.03 0.07 0.04
##
## $alpha
## [1] 0.05
##
## $rejected
##    H1    H2    H3
##  TRUE FALSE  TRUE
##
## $weights
## H1 H2 H3
##  0  1  0
##
## $G
##      [,1] [,2] [,3]
## [1,]    0    0    0
## [2,]    0    0    0
## [3,]    0    0    0
```

```
library(network)
library(ggplot2)
library(ggnetwork)
```

```
net <- network(matrix,directed = TRUE,
               names.eval = "weights",ignore.eval = FALSE)
net
```

```
##  Network attributes:
##   vertices = 3
##   directed = TRUE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##   total edges= 6
##     missing edges= 0
##     non-missing edges= 6
##
##  Vertex attribute names:
##     vertex.names
##
##  Edge attribute names:
##     weights
```



```
res <- gMCP_xc2(matrix,weights,
         pvalues, alpha = 0.05,fweights = F)
res
```

```
## $pvalues
## [1] 0.01 0.07 0.02
```

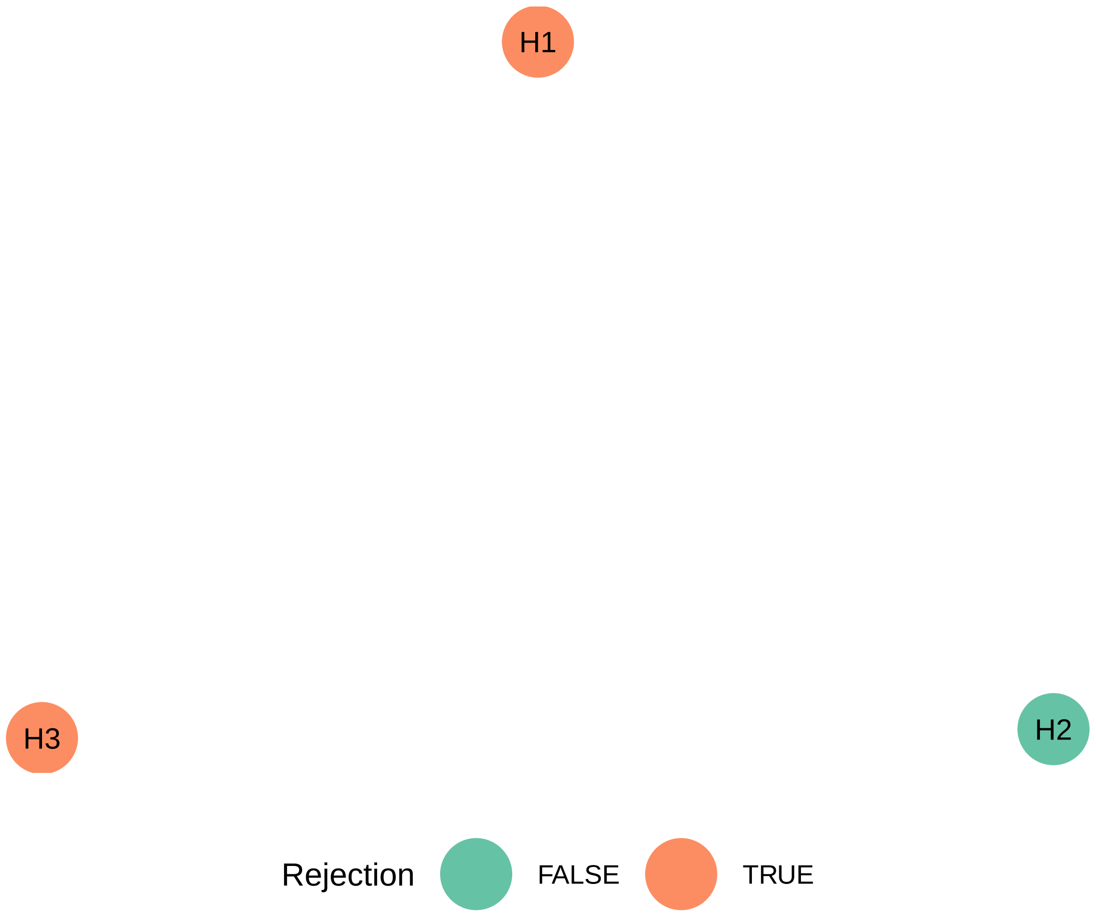
```
##
## $adjpvalues
## [1] 0.03 0.07 0.04
##
## $alpha
## [1] 0.05
##
## $rejected
##    H1    H2    H3
##  TRUE FALSE  TRUE
##
## $weights
## H1 H2 H3
##  0  1  0
##
## $G
##      [,1] [,2] [,3]
## [1,]    0    0    0
## [2,]    0    0    0
## [3,]    0    0    0
```

```r
res_pvalues <- res$pvalues
res_weights <- res$weights
res_G <- res$G

res_net <- network(res_G,directed = TRUE,
                   names.eval = "weights",ignore.eval = FALSE)
res_net %v% "weights" <- res$weights
res_net %v% "Rejection" <- res$rejected
res_net %v% "vertex.names"  <- rownames(matrix)
res_net
```

```
##  Network attributes:
##   vertices = 3
##   directed = TRUE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##   total edges= 0
##     missing edges= 0
##     non-missing edges= 0
##
##  Vertex attribute names:
##     Rejection vertex.names weights
##
## No edge attributes
```

```r
# set.edge.attribute(res_net, "color", ifelse(res_net %e% "reject" == FALSE, "red", "grey75"))
ggplot(res_net, aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(arrow = arrow(length = unit(15, "pt"), type = "closed"),
             color = "grey50",
             curvature = 0.15) +
  geom_nodes(aes(x, y,color = Rejection), size = 12) +
  geom_nodetext(aes(label = vertex.names)) +
  geom_edgetext_repel(aes(label = weights), color = "white", fill = "grey25",
                      box.padding = unit(0.25, "line")) +
  scale_color_brewer(palette = "Set2") +
  theme_blank()+
  theme(legend.position = "bottom")
```


H1
H3
H2
Rejection
FALSE
TRUE