Example2

```
library(gMCP)
graph <- BretzEtAl2011()
pvalues <- c(0.1, 0.008, 0.005, 0.15, 0.04, 0.006)
gMCP(graph, pvalues,alpha = 0.05)
```

```
## 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
```

```
# graphGUI(graph)
```

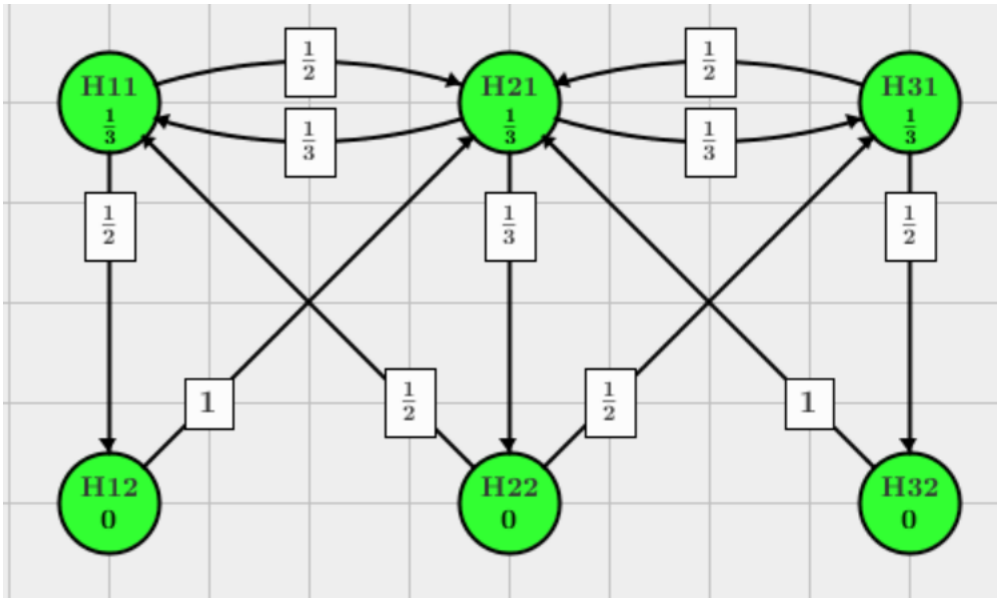


Figure 1: Example1 GUI

```
source("./R/gMCP_xc2.R")
weights <- graph@weights
matrix <- graph@m
matrix <- round(matrix,digits = 2)
gMCP_xc2(matrix,weights,
         pvalues, alpha = 0.05,fweights = F)
```

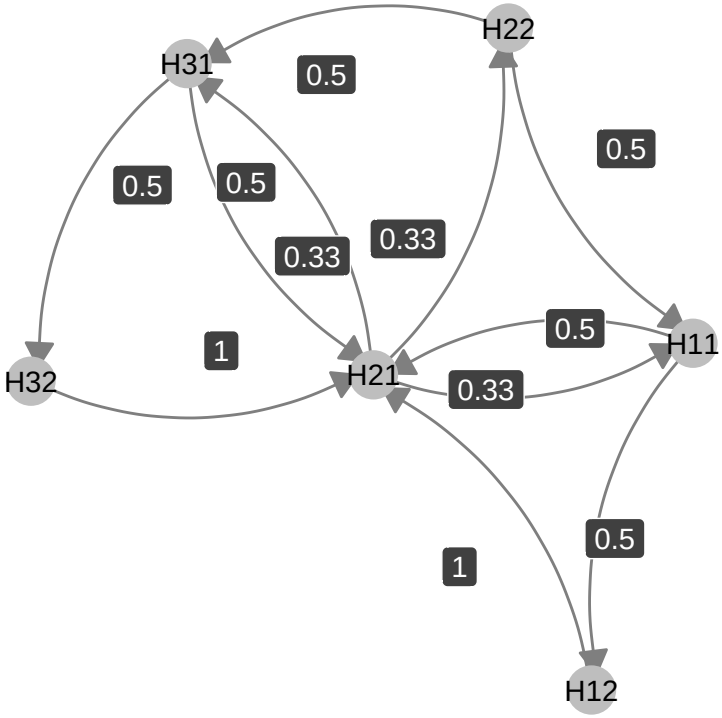
```
## $pvalues
## [1] 0.100 0.008 0.005 0.150 0.040 0.006
##
## $adjpvalues
## [1] 0.200 0.032 0.030 0.200 0.120 0.030
##
## $alpha
## [1] 0.05
```

```
##
## $rejected
##    H1    H2    H3    H4    H5    H6
## FALSE  TRUE  TRUE FALSE FALSE  TRUE
##
## $weights
##       H11       H21       H31       H12       H22       H32
## 0.6616915 0.0000000 0.0000000 0.0000000 0.3283582 0.0000000
##
## $G
##           [,1] [,2] [,3]      [,4]      [,5] [,6]
## [1,] 0.0000000    0    0 0.6633663 0.3267327    0
## [2,] 0.0000000    0    0 0.0000000 0.0000000    0
## [3,] 0.0000000    0    0 0.0000000 0.0000000    0
## [4,] 0.4925373    0    0 0.0000000 0.4925373    0
## [5,] 0.9900990    0    0 0.0000000 0.0000000    0
## [6,] 0.0000000    0    0 0.0000000 0.0000000    0
```

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

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

```
##  Network attributes:
##   vertices = 6
##   directed = TRUE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##   total edges= 11
##     missing edges= 0
##     non-missing edges= 11
##
##  Vertex attribute names:
##     vertex.names
##
##  Edge attribute names:
##     weights
```



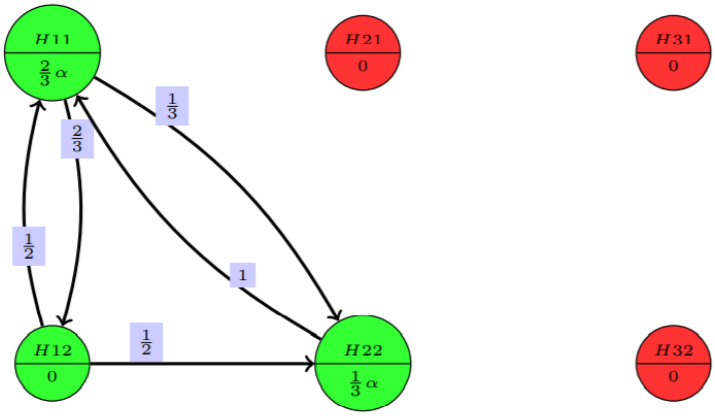


Figure 5: Final graph from the test procedure after rejection of H_{21}, H_{31} and H_{32}.

Figure 2: Example1 GUI

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

```
## $pvalues
## [1] 0.100 0.008 0.005 0.150 0.040 0.006
##
## $adjpvalues
```

```
## [1] 0.200 0.032 0.030 0.200 0.120 0.030
##
## $alpha
## [1] 0.05
##
## $rejected
##    H1    H2    H3    H4    H5    H6
## FALSE  TRUE  TRUE FALSE FALSE  TRUE
##
## $weights
##       H11       H21       H31       H12       H22       H32
## 0.6616915 0.0000000 0.0000000 0.0000000 0.3283582 0.0000000
##
## $G
##           [,1] [,2] [,3]      [,4]      [,5] [,6]
## [1,] 0.0000000    0    0 0.6633663 0.3267327    0
## [2,] 0.0000000    0    0 0.0000000 0.0000000    0
## [3,] 0.0000000    0    0 0.0000000 0.0000000    0
## [4,] 0.4925373    0    0 0.0000000 0.4925373    0
## [5,] 0.9900990    0    0 0.0000000 0.0000000    0
## [6,] 0.0000000    0    0 0.0000000 0.0000000    0
```

```
res_pvalues <- res$pvalues
res_weights <- round(res$weights,digits = 2)
res_G <- round(res$G,digits = 2)

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

e <- network.edgecount(res_net)

# res_net %v% "weights" <- res$weights
 res_net %v% "Rejection" <- res$rejected
# res_net %v% "vertex.names"  <- rownames(matrix)
res_net
```

```
##  Network attributes:
##   vertices = 6
##   directed = TRUE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##   total edges= 5
##     missing edges= 0
##     non-missing edges= 5
##
##  Vertex attribute names:
##     Rejection vertex.names
##
##  Edge attribute names:
##     weights
```

```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") +
#  scale_size_area("importance", max_size = 9) +
  theme_blank()+
  theme(legend.position = "bottom")
```