Supplementary Material for
Accurate directional inference in Gaussian graphical models

Claudia Di Caterina, Nancy Reid and Nicola Sartori

1 Introduction

The current report reproduces the Figures and numerical results in Sections 4 and 5 of the main text. The outputs have been produced using R version 4.0.3 (R Core Team 2020). The code chunk below loads the necessary R packages.

```
library(matrixcalc)
library(Matrix)
library(mvtnorm)
library(ggm)
library(Rgraphviz)
library(gRbase)
```

We also provide code to reproduce all simulation results in the paper. The R scripts to carry out the simulation experiments, and the results from those, are provided in the `DirTestGGM_code+results.zip` archive. `res_dir` is the directory where the contents of the archive are and needs to be set appropriately.

```
res_dir <- "~/Dropbox/Directional/Supplementary/DirTestGGM_code+results"
```

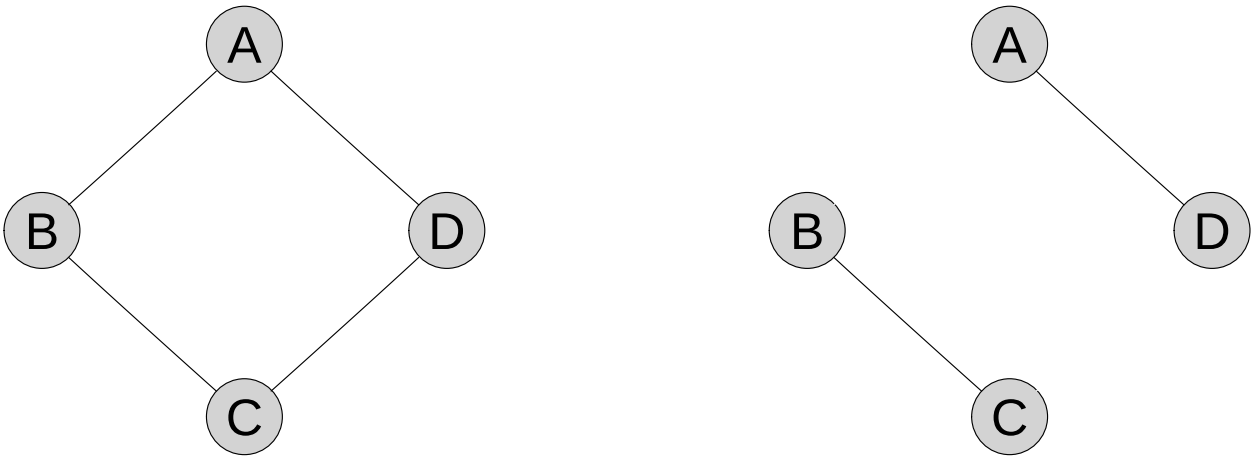
2 Simulation studies

This section provides the R code that reproduces the outputs of Section 4 in the paper.

2.1 Scenario 1

The code chunk below reproduces Figure 1.

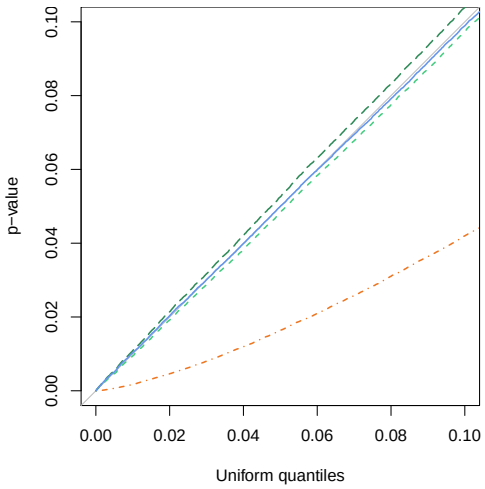
```
defAttrs <- getDefaultAttrs()
eAttrs <- list()
eAttrs$color <- c("A~B" = "white", "D~C" = "white", "C~B" = "black", "D~A" = "black")

ug0 <- ug(~ A*B + D*C)
ug1 <- ug(~ A*B + D*C + C*B + D*A)

par(mfrow = c(1, 2))
plot(ug1, attrs = list(node = list(fillcolor = "lightgrey", fontsize = "10")))
plot(ug1, edgeAttrs = eAttrs, attrs = list(node = list(fillcolor = "lightgrey",
                                                       fontsize = "10")))
```

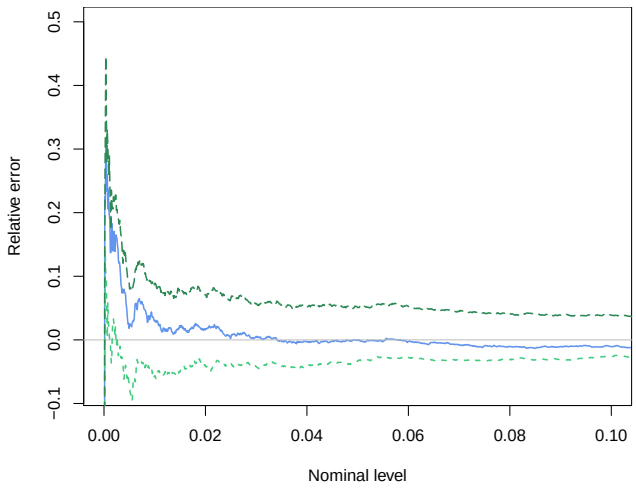


The following code chunk uses the image `nondec_res_q4_n7.rda` to reproduce Figure 2 of the main text. `nondec_res_q4_n7.rda` results by running the script `simulations1.R`, available in the supplementary code archive.

```
setwd(res_dir)
load("nondec_res_q4_n7.rda")

par(mfrow = c(1, 2), mai = c(1, 1, 0, 0.5))
old.pty <- par("pty")
par(pty = "s")
# 1st plot: empirical p-values
index <- seq(1, length(res_small$first.order), by = 10)
x <- sort(res_small$first.order)
x <- x[index]
plot(ppoints(x), x, type = "l",
     xlim= c(0, 0.1), lwd = 1.5, ylim= c(0, 0.1), xlab = "Uniform quantiles",
     ylab = "p-value", pty = "s", col = "chocolate2", lty = "dotdash", cex.lab = 1)
abline(0, 1, col ="grey")
x <- sort(res_small$skovgaard1)
x <- x[index]
lines(ppoints(x), x, col = "seagreen3",
      lwd = 1.5, lty = "dashed")
x <- sort(res_small$skovgaard2)
x <- x[index]
lines(ppoints(x), x, col = "seagreen4",
      lty = "longdash", lwd = 1.5)
x <- sort(res_small$directional)
x <- x[index]
lines(ppoints(x), x, col = "cornflowerblue", lwd = 1.5)

par(pty = old.pty)

# 2nd plot: relative error for p-values lower than 0.10 (without w)
x <- sort(res_small$directional)
x <- x[index]
plot(ppoints(x), (x - ppoints(x))/ppoints(x), xlim = c(0, 0.1), lwd = 1.5,
     ylab ="Relative error", xlab = "Nominal level", type ="l", ylim = c(-0.08, 0.5),
     cex.lab = 1, col = "cornflowerblue")
abline(h = 0, col = "grey")
```

```
x <- sort(res_small$skovgaard1)
x <- x[index]
lines(ppoints(x), (x - ppoints(x))/ppoints(x),
      col = "seagreen3", lty = "dashed", lwd = 1.5)
x <- sort(res_small$skovgaard2)
x <- x[index]
lines(ppoints(x), (x - ppoints(x))/ppoints(x), lwd = 1.5,
      col = "seagreen4", lty = "longdash")
```





2.2 Scenario 2

The code chunk below uses the image `dir_res_q11.rda` to reproduce Table 1 of the main text. Tables 2 and 3 are obtained similarly starting from the files `dir_res_q30.rda` and `dir_res_q50.rda`, respectively. All these images result by running the script `simulations2.R` appropriately adjusted according to the value of q, and available in the supplementary code archive.

```
summary.sim <- function(out, levels = c(0.01, 0.025, 0.05, 0.10, 0.25, 0.5, 0.75,
                                        0.9, 0.95, 0.975, 0.99)) {
  fo <- sapply(levels, function(x) mean(out$first.order < x))
  dir2 <- sapply(levels, function(x) mean(out$directional < x))
  sk1 <- sapply(levels, function(x) mean(out$skovgaard1 < x))
  sk2 <- sapply(levels, function(x) mean(out$skovgaard2 < x))

  sim.se <- sqrt(levels * (1 - levels)/length(out$first.order))
  res <- rbind(levels, fo, dir2, sk1, sk2, sim.se)
  rownames(res) <- c("nominal", "first-order", "directional", "Skovgaard W*",
                     "Skovgaard W**", "sim-error")
  res
}

setwd(res_dir)
load("dir_res_q11.rda")

res2md <- round(summary.sim(resMD2), 3)
res3md <- round(summary.sim(resMD3), 3)
res6md <- round(summary.sim(resMD6), 3)
res9md <- round(summary.sim(resMD9), 3)
```

```
tab1 <- rbind(res2md[c(2, 4, 5, 3),], res3md[c(2, 4, 5, 3),],
              res6md[c(2, 4, 5, 3),], res9md[c(2, 4, 5, 3),])
colnames(tab1) <- c("1", "2.5", "5", "10", "25", "50", "75", "90", "95", "97.5", "99")
tab1*100
```

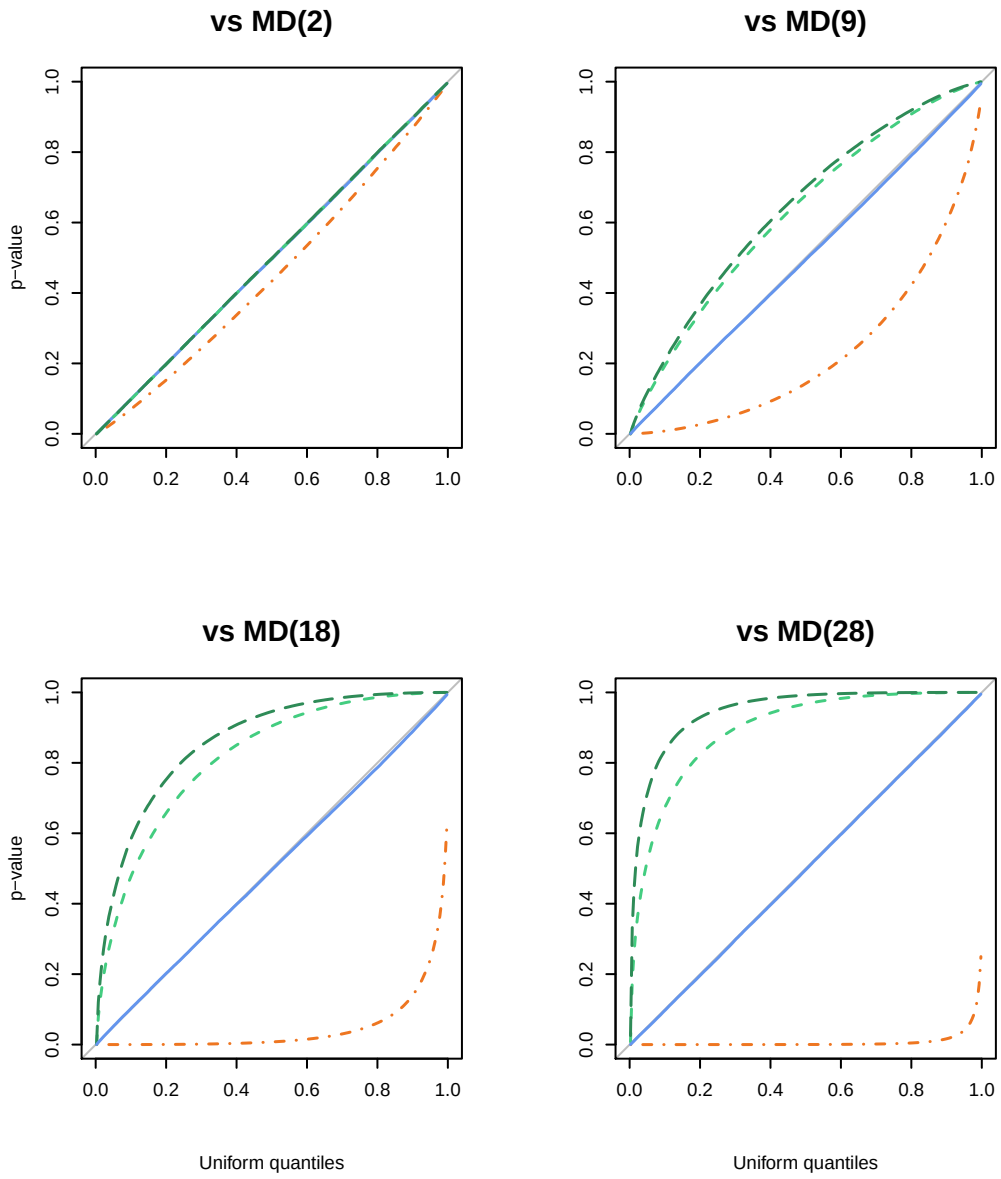
```
##                  1 2.5    5   10   25   50   75   90   95 97.5   99
## first-order    1.4 3.4  6.5 12.2 28.6 54.0 77.8 91.2 95.7 97.8 99.1
## Skovgaard W*   1.0 2.5  5.0 10.0 25.0 50.1 75.0 89.9 94.9 97.4 98.9
## Skovgaard W** 1.0 2.5  5.0 10.0 25.0 50.1 75.0 89.9 94.9 97.4 98.9
## directional    1.0 2.5  5.0 10.0 25.1 50.3 75.3 90.1 95.0 97.5 99.0
## first-order    1.8 3.9  7.2 13.5 30.4 56.1 79.3 92.0 96.0 98.1 99.2
## Skovgaard W*   1.1 2.6  5.0 10.0 24.6 49.6 74.6 89.6 94.7 97.3 98.9
## Skovgaard W** 1.0 2.5  5.0  9.9 24.5 49.5 74.5 89.5 94.7 97.2 98.9
## directional    1.0 2.6  5.1 10.1 25.0 50.3 75.4 90.2 95.0 97.5 99.0
## first-order    2.5 5.5  9.8 17.4 36.2 62.2 83.3 94.0 97.2 98.6 99.5
## Skovgaard W*   0.8 2.1  4.3  8.8 22.4 46.4 71.7 87.8 93.6 96.6 98.5
## Skovgaard W** 0.8 2.1  4.2  8.6 22.0 45.9 71.2 87.5 93.4 96.4 98.5
## directional    1.0 2.5  4.9 10.0 25.0 50.3 75.3 90.2 95.1 97.5 99.0
## first-order    3.3 6.9 12.0 20.6 40.8 66.2 85.9 95.2 97.8 99.0 99.6
## Skovgaard W*   0.7 1.8  3.7  7.8 20.7 43.7 69.1 86.3 92.6 96.1 98.2
## Skovgaard W** 0.7 1.8  3.6  7.5 20.1 42.8 68.2 85.7 92.2 95.8 98.1
## directional    1.0 2.4  4.9  9.9 25.2 50.0 75.0 90.1 95.1 97.5 99.0
```

The following code chunk uses the image `dir_res_q30.rda` to reproduce Figure 3 of the main text.

```
plot3.sim <- function(out, out1, out2, out3, step = 10) {
  index <- seq(1, length(out$first.order), by = step)
  par(mfrow = c(2, 2), cex = 0.6, mai = c(0.4, 0.8, 0.15, 0), mex = 0.8,
      pty = "s", oma = c(0.2, 0.2, 0.2, 0.2), mgp = c(3, 1, 0))
  x <- sort(out$first.order)
  x <- x[index]
  plot(ppoints(x), x, xlab = ' ', ylab = '', pch = ".", axes = FALSE, type ="l",
       xlim = c(0, 1), lwd = 1.5, ylim = c(0, 1), col = "chocolate2", lty = "dotdash")
  abline(0, 1, col ="grey")
  x <- sort(out$directional)
  x <- x[index]
  lines(ppoints(x), x, col = "cornflowerblue", lwd = 1.5)
  x <- sort(out$skovgaard1)
  x <- x[index]
  lines(ppoints(x), x, col = "seagreen3",
        lwd = 1.5, lty = "dashed")
  x <- sort(out$skovgaard2)
  x <- x[index]
  lines(ppoints(x), x, col = "seagreen4",
        lty = "longdash", lwd = 1.5)
  box()
  axis(2)
  axis(1)
  title("vs MD(2)", line = 2, cex.main = 1.5)
  title(ylab = "p-value", cex.lab = 1, line = 3)

  x <- sort(out1$first.order)
  x <- x[index]
  plot(ppoints(x), x, xlab = ' ', ylab = '', pch = ".", axes = FALSE, type = "l",
```

```
     xlim = c(0, 1), lwd = 1.5, ylim = c(0, 1), lty = "dotdash", col = "chocolate2")
abline(0, 1, col ="grey")
x <- sort(out1$skovgaard1)
x <- x[index]
lines(ppoints(x), x, col = "seagreen3",
      lwd = 1.5, lty = "dashed")
x <- sort(out1$skovgaard2)
x <- x[index]
lines(ppoints(x), x, col = "seagreen4",
      lty = "longdash", lwd = 1.5)
x <- sort(out1$directional)
x <- x[index]
lines(ppoints(x), x, col = "cornflowerblue", lwd = 1.5)
box()
axis(2)
axis(1)
title("vs MD(9)", line = 2, cex.main = 1.5)

x <- sort(out2$first.order)
x <- x[index]
plot(ppoints(x), x, xlab = ' ', ylab = '', pch = ".", axes = FALSE, type = "l",
     xlim = c(0, 1), lwd = 1.5, ylim = c(0, 1), lty = "dotdash", col = "chocolate2")
abline(0, 1, col ="grey")
x <- sort(out2$skovgaard1)
x <- x[index]
lines(ppoints(x), x, col = "seagreen3",
      lwd = 1.5, lty = "dashed")
x <- sort(out2$skovgaard2)
x <- x[index]
lines(ppoints(x), x, col = "seagreen4",
      lty = "longdash", lwd = 1.5)
x <- sort(out2$directional)
x <- x[index]
lines(ppoints(x), x, col = "cornflowerblue", lwd = 1.5)
box()
axis(2)
axis(1)
title(xlab = "Uniform quantiles", cex.lab = 1, line = 5)
title("vs MD(18)", line = 2, cex.main = 1.5)
title(ylab = "p-value", cex.lab = 1, line = 3)

x <- sort(out3$first.order)
x <- x[index]
plot(ppoints(x), x, xlab = ' ', ylab = '', pch = ".", axes = FALSE, type = "l",
     xlim = c(0, 1), lwd = 1.5, ylim = c(0, 1), lty = "dotdash", col = "chocolate2")
abline(0, 1, col ="grey")
x <- sort(out3$skovgaard1)
x <- x[index]
lines(ppoints(x), x, col = "seagreen3",
      lwd = 1.5, lty = "dashed")
x <- sort(out3$skovgaard2)
x <- x[index]
lines(ppoints(x), x, col = "seagreen4",
```

```
        lty = "longdash", lwd = 1.5)
  x <- sort(out3$directional)
  x <- x[index]
  lines(ppoints(x), x, col = "cornflowerblue", lwd = 1.5)
  box()
  axis(2)
  axis(1)
  title(xlab = "Uniform quantiles", cex.lab = 1, line = 5)
  title("vs MD(28)", line = 2, cex.main = 1.5)
}

setwd(res_dir)
load("dir_res_q30.rda")

plot3.sim(resMD2, resMD9, resMD18, resMD28, step = 500)
```



2.3 Scenario 3

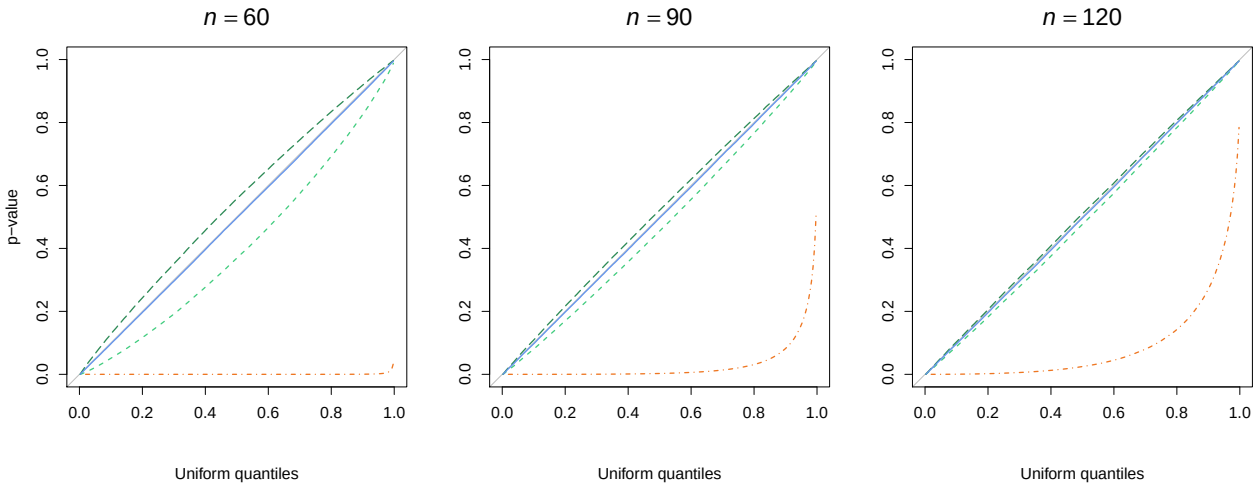
The following two code chunks use the image `dir_res_block.rda` to reproduce Table 4 and Figure 4, respectively, of the main text. The file `dir_res_block.rda` results by running the script `simulations3.R`, available in the supplementary code archive.

```
setwd(res_dir)
load("dir_res_block.rda")

res60n <- round(summary.sim(res60n), 3)
res90n <- round(summary.sim(res90n), 3)
res120n <- round(summary.sim(res120n), 3)

tab4 <- rbind(res60n[c(2, 4, 5, 3),], res90n[c(2, 4, 5, 3),],
              res120n[c(2, 4, 5, 3),])
colnames(tab1) <- c("1", "2.5", "5", "10", "25", "50", "75", "90", "95", "97.5", "99")
tab4 * 100
```

```
##               [,1] [,2] [,3] [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10] [,11]
## first-order   98.4 99.3 99.7 99.9 100.0 100.0 100.0 100.0 100.0 100.0 100.0
## Skovgaard W*   2.4  5.3  9.6 17.3  36.6  62.9  84.2  94.6  97.6  98.9  99.6
## Skovgaard W**  0.6  1.7  3.5  7.5  20.4  43.9  70.3  87.4  93.5  96.7  98.6
## directional    1.0  2.5  5.0 10.0  25.1  50.1  75.2  90.2  95.1  97.6  99.0
## first-order   65.9 77.1 85.0 91.5  97.3  99.4  99.9 100.0 100.0 100.0 100.0
## Skovgaard W*   1.3  3.2  6.1 12.0  28.5  54.2  78.2  91.7  96.0  98.1  99.2
## Skovgaard W**  0.8  2.1  4.3  8.9  23.0  47.6  73.2  89.0  94.5  97.2  98.8
## directional    0.9  2.5  5.0 10.1  25.0  50.1  75.1  90.1  95.1  97.6  99.0
## first-order   36.6 50.0 61.6 73.6  88.6  96.7  99.3  99.9 100.0 100.0 100.0
## Skovgaard W*   1.1  2.9  5.6 11.0  26.8  52.2  76.5  90.9  95.5  97.8  99.1
## Skovgaard W**  0.9  2.3  4.6  9.4  24.0  48.7  73.9  89.4  94.6  97.3  98.9
## directional    1.0  2.5  5.0 10.1  25.1  50.1  75.0  90.0  95.0  97.5  99.0
```

```
plot4.sim <- function(out, out1, out2, step = 10) {
  index <- seq(1, length(out$first.order), by = step)
  par(mfrow = c(1, 3), cex = 1.5, mai = c(0.7, 1.2, 0.15, 0.1), mex = 0.8,
      pty = "s", oma = c(0.2, 0.2, 0.2, 0.2), mgp = c(3, 1, 0))
  x <- sort(out$first.order)
  x <- x[index]
  plot(ppoints(x), x, xlab = ' ', ylab = '', pch = ".", axes = FALSE, type = "l",
       xlim = c(0, 1), lwd = 2, ylim = c(0, 1), lty = "dotdash", col = "chocolate2")
  abline(0, 1, col ="grey")
  x <- sort(out$skovgaard1)
  x <- x[index]
  lines(ppoints(x), x, col = "seagreen3",
        lwd = 2, lty = "dashed")
  x <- sort(out$skovgaard2)
  x <- x[index]
  lines(ppoints(x), x, col = "seagreen4",
        lty = "longdash", lwd = 2)
  x <- sort(out$directional)
  x <- x[index]
  lines(ppoints(x), x, col = "cornflowerblue", lwd = 2)
  box()
  axis(2)
  axis(1)
```

```
  title(xlab = "Uniform quantiles", cex.lab = 1, line = 5)
  title(expression(italic(n)==60), line = 2, cex.main = 1.5)
  title(ylab = "p-value", cex.lab = 1, line = 3)

  x <- sort(out1$first.order)
  x <- x[index]
  plot(ppoints(x), x, xlab = ' ', ylab = '', pch = ".", axes = FALSE, type = "l",
       xlim = c(0, 1), lwd = 2, ylim = c(0, 1), lty = "dotdash", col = "chocolate2")
  abline(0, 1, col ="grey")
  x <- sort(out1$skovgaard1)
  x <- x[index]
  lines(ppoints(x), x, col = "seagreen3",
        lwd = 2, lty = "dashed")
  x <- sort(out1$skovgaard2)
  x <- x[index]
  lines(ppoints(x), x, col = "seagreen4",
        lty = "longdash", lwd = 2)
  x <- sort(out1$directional)
  x <- x[index]
  lines(ppoints(x), x, col = "cornflowerblue", lwd = 2)
  box()
  axis(2)
  axis(1)
  title(xlab = "Uniform quantiles", cex.lab = 1, line = 5)
  title(expression(italic(n)==90), line = 2, cex.main = 1.5)

  x <- sort(out2$first.order)
  x <- x[index]
  plot(ppoints(x), x, xlab = ' ', ylab = '', pch = ".", axes = FALSE, type = "l",
       xlim = c(0, 1), lwd = 2, ylim = c(0, 1), lty = "dotdash", col = "chocolate2")
  abline(0, 1, col ="grey")
  x <- sort(out2$skovgaard1)
  x <- x[index]
  lines(ppoints(x), x, col = "seagreen3",
        lwd = 2, lty = "dashed")
  x <- sort(out2$skovgaard2)
  x <- x[index]
  lines(ppoints(x), x, col = "seagreen4",
        lty = "longdash", lwd = 2)
  x <- sort(out2$directional)
  x <- x[index]
  lines(ppoints(x), x, col = "cornflowerblue", lwd = 2)
  box()
  axis(2)
  axis(1)
  title(xlab = "Uniform quantiles", cex.lab = 1, line = 5)
  title(expression(italic(n)==120), line = 2, cex.main = 1.5)
}

setwd(res_dir)
load("dir_res_block.rda")

plot4.sim(res60n, res90n, res120n, step = 500)
```



3 Applications

This section provides the R code that reproduces the outputs of Section 5 in the paper. The functions to be used in order to run the following analyses can be found in the script `simulation_functions.R` in the archive.

3.1 Cow data

The code chunk below loads the file `cowdata.txt` (Kenward 1987) from the archive and reproduces the numerical results regarding the veterinary trial on cattle reported in the main text.

```
setwd(res_dir)
source("simulation_functions.R")
cow <- read.table("cowdata.txt", skip = 7, header = TRUE)

names(cow)[-(1:2)] <- paste("T", 1:11, sep = '')
cow.times <- cow[, -(1:2)]
cow.timesA <- cow[cow$G == "A", -(1:2)]
cow.timesB <- cow[cow$G == "B", -(1:2)]

## group A
cowsA <- cow.timesA
S <- cov(cowsA)
n <- nrow(cowsA)

# unsaturated model under H1
# MD(3)
G1 <- UG(~ T1*T2*T3*T4 + T2*T3*T4*T5 + T3*T4*T5*T6 + T4*T5*T6*T7 + T5*T6*T7*T8 +
           T6*T7*T8*T9 + T7*T8*T9*T10 + T8*T9*T10*T11)
model1 <- fitConGraph(G1, S, n)

# model under H0
# MD(1)
G0 <- UG(~ T1*T2 + T2*T3 + T3*T4 + T4*T5 + T5*T6 + T6*T7 + T7*T8 + T8*T9 + T9*T10 +
           T10*T11)
```

```r
model0 <- fitConGraph(G0, S, n)

### tests
## w
lrt(model0, model1, n)
```

```
## $Woss
## [1] 28.38429
##
## $pvalue
## [1] 0.04064335
```

```r
## w* and w**
Wstar(model0, model1, n, G1 = G1)
```

```
## $wstar1
##          [,1]
## [1,] 22.97676
##
## $p.value1
##           [,1]
## [1,] 0.1500088
##
## $wstar2
##         [,1]
## [1,] 22.6913
##
## $p.value2
##           [,1]
## [1,] 0.1595728
```

```r
## directional p-value
dir_p(model0, model1, n , G1 = G1)
```

```
## [1] 0.1113944
```

```r
## group B
cowsB <- cow.timesB
S <- cov(cowsB)
n <- nrow(cowsB)

# unsaturated model under H1
# MD(3)
G1 <- UG(~ T1*T2*T3*T4 + T2*T3*T4*T5 + T3*T4*T5*T6 + T4*T5*T6*T7 + T5*T6*T7*T8 +
           T6*T7*T8*T9 + T7*T8*T9*T10 + T8*T9*T10*T11)
model1 <- fitConGraph(G1, S, n)

# model under H0
# MD(1)
G0 <- UG(~ T1*T2 + T2*T3 + T3*T4 + T4*T5 + T5*T6 + T6*T7 + T7*T8 + T8*T9 + T9*T10 +
           T10*T11)

model0 <- fitConGraph(G0, S, n)

### tests
## w
```

```
lrt(model0, model1, n)
```

```
## $Woss
## [1] 31.89548
##
## $pvalue
## [1] 0.01550424
```

```
## w* and w**
Wstar(model0, model1, n, G1 = G1)
```

```
## $wstar1
##          [,1]
## [1,] 30.05542
##
## $p.value1
##            [,1]
## [1,] 0.02594814
##
## $wstar2
##          [,1]
## [1,] 30.02809
##
## $p.value2
##            [,1]
## [1,] 0.02614321
```

```
## directional p-value
dir_p(model0, model1, n, G1 = G1)
```

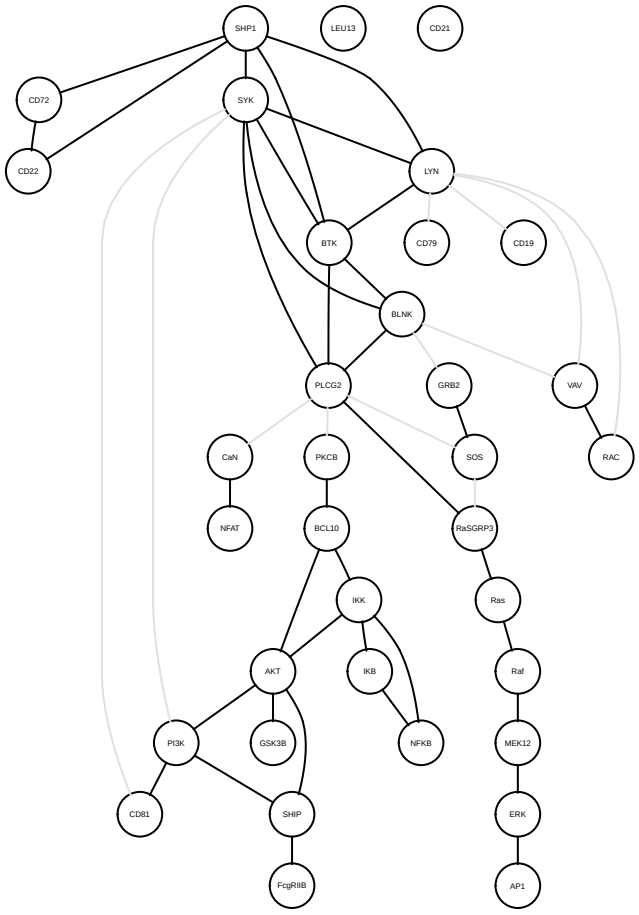
```
## [1] 0.02887994
```

3.2 Leukemia data

The next code chunks load from the archive the file `BCRdata.rda`, also accessible via the R package `topologyGSA` (Massa and Sales 2016), and reproduces Figure 5 and the numerical results regarding the genetic study on lymphocytic leukemia reported in the main text.

```
setwd(res_dir)
load("BCRdata.rda")

ug_bcell <- moralize(dag_bcell)
defAttrs <- getDefaultAttrs()
eAttrs <- list()
eAttrs$color <- c("SYK~PI3K" = "gray88", "SYK~CD81" = "gray88", "LYN~CD79" = "gray88",
                  "LYN~CD19" = "gray88", "LYN~VAV" = "gray88", "LYN~RAC" = "gray88",
                  "BLNK~VAV" = "gray88", "BLNK~GRB2" = "gray88", "PLCG2~CaN" = "gray88",
                  "PLCG2~PKCB" = "gray88", "PLCG2~SOS" = "gray88",
                  "SOS~RaSGRP3" = "gray88")

eAttrs$style <- c("SYK~PI3K" = "dotted", "SYK~CD81" = "dotted", "LYN~CD79" = "dotted",
                  "LYN~CD19" = "dotted", "LYN~VAV" = "dotted", "LYN~RAC" = "dotted",
                  "BLNK~VAV" = "dotted", "BLNK~GRB2" = "dotted", "PLCG2~CaN" = "dotted",
                  "PLCG2~PKCB" = "dotted", "PLCG2~SOS" = "dotted", "SOS~RaSGRP3" = "dotted")
```

```
plot(ug_bcell, edgeAttrs = eAttrs, attrs = list(node = list(fontsize = "13")))
```




```
G1 <- as(ug_bcell,"matrix")

n <- nrow(y2)
S <- cov(y2)

# H1: unsaturated model
model1 <- fitConGraph(G1, S, n)

# H0: simpler BCR signaling pathway
G0 <- G1
G0["SYK", "PI3K"] <- G0["SYK", "CD81"] <- G0["PI3K", "SYK"] <-
  G0["CD81", "SYK"] <- 0
G0["LYN", "CD79"] <- G0["LYN", "CD19"] <- G0["LYN", "VAV"] <- G0["LYN", "RAC"] <-
  G0[ "CD79", "LYN"] <- G0[ "CD19", "LYN"] <- G0["VAV", "LYN"] <-
  G0["RAC", "LYN"] <- 0
G0["BLNK", "VAV"] <- G0["BLNK", "GRB2"] <- G0["VAV", "BLNK"] <-
  G0["GRB2", "BLNK"] <- 0
G0["PLCG2", "CaN"] <- G0["PLCG2", "PKCB"] <- G0["PLCG2", "SOS"] <-
  G0["CaN", "PLCG2"] <- G0["PKCB", "PLCG2"] <- G0["SOS", "PLCG2"] <- 0
G0["RaSGRP3", "SOS"] <- G0["SOS", "RaSGRP3"] <- 0

model0 <- fitConGraph(G0, S, n)
```

```
## w
lrt(model0, model1, n)
```

```
## $Woss
## [1] 33.51951
##
## $pvalue
## [1] 0.0008027688
```

```
## w* and w**
Wstar(model0, model1, n, G1 = G1)
```

```
## $wstar1
##          [,1]
## [1,] 32.17165
##
## $p.value1
##              [,1]
## [1,] 0.001301832
##
## $wstar2
##          [,1]
## [1,] 32.15782
##
## $p.value2
##              [,1]
## [1,] 0.001308256
```

```
## directional p-value
dir_p(model0, model1, n, G1 = G1)
```

```
## [1] 0.001394133
```

References

Kenward, Michael G. 1987. "A Method for Comparing Profiles of Repeated Measurements." *J. R. Statist. Soc. C* 36: 296–308.

Massa, Sofia, and Gabriele Sales. 2016. *TopologyGSA: Gene Set Analysis Exploiting Pathway Topology.* https://CRAN.R-project.org/package=topologyGSA.

R Core Team. 2020. *R: A Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.