

```
rm(list = ls())
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22000)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods    base
##
## loaded via a namespace (and not attached):
## [1] compiler_4.2.1  magrittr_2.0.3  fastmap_1.1.0   cli_3.3.0
## [5] tools_4.2.1     htmltools_0.5.3 rstudioapi_0.13 yaml_2.3.5
## [9] stringi_1.7.8   rmarkdown_2.15  knitr_1.39      stringr_1.4.0
## [13] xfun_0.32       digest_0.6.29   rlang_1.0.4     evaluate_0.16
```

```
library(covdepGE)
library(loggle)
library(HeteroGGM)
library(mclust)
```

```
## Package 'mclust' version 5.4.10
## Type 'citation("mclust")' for citing this R package in publications.
```

```
library(ggplot2)
library(ggpubr)
library(kableExtra)
```

```
now <- format(Sys.time(), "%Y%m%d_%H%M%S")
```

```
# initialize storage for results, time, and progress tracking
```

```
set.seed(1)
```

```
n_trials <- 1
```

```
results <- vector("list", n_trials + 1)
```

```
names(results) <- c(paste0("trial", 1:n_trials), "time")
```

```
results$time <- Sys.time()
```

```
pb <- txtProgressBar(0, n_trials, style = 3)
```

```
## |
```

```
|
```

```
# define data dimensions
```

```
p <- 3
```

```
(n <- round(2 * 3 * p))
```

```
## [1] 18
```

```
(nj <- n %/% 3)
```

```
## [1] 6
```

```
# function for surpressing cat
quiet <- function(x) {
  sink(tempfile())
  on.exit(sink())
  invisible(force(x))
}

# get number of available workers
(num_workers <- parallel::detectCores() - 1)
```

```
## [1] 15
```

```
# function for evaluating estimated graphs compared to ground truth
eval_est <- function(est, true){

  # get true number of edges and non-edges
  num_edge <- sum(true, na.rm = T)
  num_non <- sum(true == 0, na.rm = T)

  # calculate sensitivity and specificity
  true_edge <- sum(est == true & true == 1, na.rm = T)
  true_non <- sum(est == true & true == 0, na.rm = T)
  sens <- true_edge / num_edge
  spec <- true_non / num_non

  list(sens = sens, spec = spec)
}

#`

#``{r, eval = F}

# perform trials
for (j in 1:n_trials){

  # generate the data and create storage for the models
  data <- generateData(p, nj, nj, nj)
  trial <- vector("list", 4)
  names(trial) <- c("data", "covdepGE", "loggle", "HeteroGGM")
  prec_arr_dim <- c(p, p, n)
  trial$data <- data

  # convert the true precision to an array and then to a graph; mask diagonal
  prec <- array(unlist(data$true_precision), prec_arr_dim)
  graph <- (prec != 0) * 1 + replicate(n, diag(rep(NA, p)) * 1)

  # fit each method, save details about results, time, etc.
```

```

# covdepGE
out_covdepGE <- tryCatch(suppressWarnings(covdepGE(
  data$X, data$Z, parallel = T, num_workers = num_workers)),
  error = function(e) list(error = e))
if (names(out_covdepGE)[[1]] == "error"){
  out_covdepGE <- out_covdepGE$error
}else{
  out_covdepGE$time <- as.numeric(out_covdepGE$model_details$elapsed, units = "secs")
  out_covdepGE$str <- array(unlist(out_covdepGE$graphs$graphs), dim = prec_arr_dim)
  out_covdepGE[c("sens", "spec")] <- eval_est(out_covdepGE$str, graph)[c("sens", "spec")]
}
trial$covdepGE <- out_covdepGE
rm(list = "out_covdepGE")
gc()

# loggle
start <- Sys.time()
out_loggle <- tryCatch(quiet(loggle.cv(t(data$X), num.thread = num_workers)),
  error = function(e) list(error = e))
if (names(out_loggle)[[1]] == "error"){
  out_loggle <- out_loggle$error
}else{
  out_loggle <- out_loggle$cv.select.result
  out_loggle$time <- as.numeric(Sys.time() - start, units = "secs")
  out_loggle$str <- array(unlist(lapply(lapply(
    out_loggle$adj.mat.opt, `~`~, diag(p)), as.matrix)), dim = prec_arr_dim)
  out_loggle[c("sens", "spec")] <- eval_est(out_loggle$str, graph)[c("sens", "spec")]
}
trial$loggle <- out_loggle
rm(list = "out_loggle")
gc()

# HeteroGGM
clust <- Mclust(data$Z, verbose = F)
lambda <- genelambda.obo(lambda2_min = 0.15)
start <- Sys.time()
out_hetGGM <- tryCatch(GGMPF(lambda, data$X + clust$classification * 10, clust$G),
  error = function(e) list(error = e))
if (names(out_hetGGM)[[1]] == "error"){
  out_hetGGM <- out_hetGGM$error
}else{
  out_hetGGM$time <- as.numeric(Sys.time() - start, units = "secs")
  out_hetGGM$str <- (out_hetGGM$Theta_hat.list[[out_hetGGM$Opt_num]] != 0) * 1
  out_hetGGM$str <- out_hetGGM$str - replicate(dim(out_hetGGM$str)[3], diag(p))
  out_hetGGM$str <- out_hetGGM$str[, , out_hetGGM$member.list[[out_hetGGM$Opt_num]]]
  out_hetGGM[c("sens", "spec")] <- eval_est(out_hetGGM$str, graph)[c("sens", "spec")]
}
trial$HeteroGGM <- out_hetGGM
rm(list = "out_hetGGM")
gc()

# for hp tuning lambda grid
# het_gr <- vector("list", n)

```

```

# for (l in 1:n){
#   het_gr[[l]] <- out_hetGGM$str[ , , l]
# }
#
# # find the unique graphs
# unique_graphs <- unique(het_gr)
#
# # create a list where the j-th element is the j-th unique graph and the
# # indices of the observations corresponding to this graph
# unique_sum <- vector("list", length(unique_graphs))
# names(unique_sum) <- paste0("graph", 1:length(unique_graphs))
#
# # iterate over each of the unique graphs
# for (j in 1:length(unique_graphs)){
#   # fix the unique graph
#   graphj <- unique_graphs[[j]]
#
#   # find indices of the observations corresponding to this graph
#   graph_inds <- which(sapply(het_gr, identical, graphj))
#
#   # split up the contiguous subsequences of these indices
#   cont_inds <- split(sort(graph_inds), cumsum(c(1, diff(sort(graph_inds))
#                                               != 1)))
#
#   # create a character summary for each of the contiguous sequences
#   inds_sum <- sapply(cont_inds, function(idxs_seq) ifelse(length(
#     idxs_seq) > 3, paste0(min(idxs_seq), "...", max(idxs_seq)),
#     paste0(idxs_seq, collapse = ",")))
#
#   # combine the summary
#   inds_sum <- paste0(inds_sum, collapse = ",")
#
#   # add the graph, indices, and summary to the unique graphs summary list
#   unique_sum[[j]] <- list(graph = graphj, indices = graph_inds,
#                           ind_sum = inds_sum)
# }
# for (graphj in unique_sum){
#   print(matViz(graphj$graph) + ggtitle(graphj$ind_sum))
# }
# out_hetGGM$Opt_lambda
# lambda

# save the trial and update the progress bar
trial$elapsed <- sum(unlist(sapply(trial, `[`, "time")))
results$time <- c(results$time, Sys.time())
results[[j]] <- trial
setTxtProgressBar(pb, j)
save(results, file = paste0("results_", now, ".Rda"))
}

```

```
## Warning in log(d): NaNs produced
```

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

```
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(d): NaNs produced
```

[illegible]

[illegible]

```
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
```

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

```
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
```

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
```



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
```

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
```

[illegible]

[illegible]



```
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(d): NaNs produced
```

[illegible]

```
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(det(Omega)): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
```

[illegible]

```
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
## Warning in log(d): NaNs produced
```

```
## |
```

```
|=====
```

```
# save the results
save(results, file = paste0("results_", now, ".Rda"))
```

## ## Analysis

```
#load("simulation_study/results.Rda")
#load("/home/jacob.a.helwig/covdepGE/simulation_study/results_20220817_150045.Rda")

# extract each of the models
results <- results[setdiff(names(results), "time")]
covdepGE_models <- lapply(results, `[[`, "covdepGE")
loggle_models <- lapply(results, `[[`, "loggle")
hetGGM_models <- lapply(results, `[[`, "HeteroGGM")

# extract the sensitivities, specificities, and time
covdepGE_sens <- sapply(covdepGE_models, `[[`, "sens")
covdepGE_spec <- sapply(covdepGE_models, `[[`, "spec")
covdepGE_times <- sapply(covdepGE_models, `[[`, "time")
loggle_sens <- sapply(loggle_models, `[[`, "sens")
loggle_spec <- sapply(loggle_models, `[[`, "spec")
loggle_times <- sapply(loggle_models, `[[`, "time")
hetGGM_sens <- sapply(hetGGM_models, `[[`, "sens")
```

```
hetGGM_spec <- sapply(hetGGM_models, `[`, "spec")
hetGGM_times <- sapply(hetGGM_models, `[`, "time")
```

```
# sensitivity analysis
summary(covdepGE_sens)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.3333 0.3333 0.3333 0.3333 0.3333 0.3333
```

```
summary(loggle_sens)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.7381 0.7381 0.7381 0.7381 0.7381 0.7381
```

```
summary(hetGGM_sens)
```

```
##      Length Class  Mode
## trial1 0      -none- NULL
```

```
summary(covdepGE_sens - loggle_sens)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.4048 -0.4048 -0.4048 -0.4048 -0.4048 -0.4048
```

```
summary(covdepGE_sens - hetGGM_sens)
```

```
## Error in covdepGE_sens - hetGGM_sens: non-numeric argument to binary operator
```

```
# specificity analysis
summary(covdepGE_spec)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1      1      1      1      1      1
```

```
summary(loggle_spec)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.6667 0.6667 0.6667 0.6667 0.6667 0.6667
```

```
summary(hetGGM_spec)
```

```
##      Length Class  Mode
## trial1 0      -none- NULL
```

```
summary(covdepGE_spec - loggle_spec)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.3333 0.3333 0.3333 0.3333 0.3333 0.3333
```

```
summary(covdepGE_spec - hetGGM_spec)
```

```
## Error in covdepGE_spec - hetGGM_spec: non-numeric argument to binary operator
```

```
# time analysis
```

```
summary(covdepGE_times)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.917   1.917   1.917   1.917   1.917   1.917
```

```
summary(loggle_times)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      80.23   80.23   80.23   80.23   80.23   80.23
```

```
summary(hetGGM_times)
```

```
##      Length Class  Mode
## trial1 0      -none- NULL
```

```
# table of results
```

```
perf <- apply(cbind(covdepGE_sens, loggle_sens, hetGGM_sens,
                    covdepGE_spec, loggle_spec, hetGGM_spec,
                    covdepGE_times, loggle_times, hetGGM_times),
              2, function(x) paste0(round(mean(x), 3), " (", round(sd(x), 3), ")"))
```

```
## Warning in mean.default(x): argument is not numeric or logical: returning NA
```

```
## Error in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm = na.rm): is.atomic(x) is not
```

```
perf_df <- data.frame(t(matrix(perf, 3, 3)))
```

```
## Error in matrix(perf, 3, 3): object 'perf' not found
```

```
row.names(perf_df) <- c("Sensitivity", "Specificity", "Time(s)")
```

```
## Error in row.names(perf_df) <- c("Sensitivity", "Specificity", "Time(s)": object 'perf_df' not found
```

```
colnames(perf_df) <- c("covdepGE", "loggle", "HeteroGGM")
```

```
## Error in colnames(perf_df) <- c("covdepGE", "loggle", "HeteroGGM)": object 'perf_df' not found
```

```
kbl(perf_df, format = "latex", booktabs = T)
```

```
## Error in knitr::kable(x = x, format = format, digits = digits, row.names = row.names, : object 'perf
```

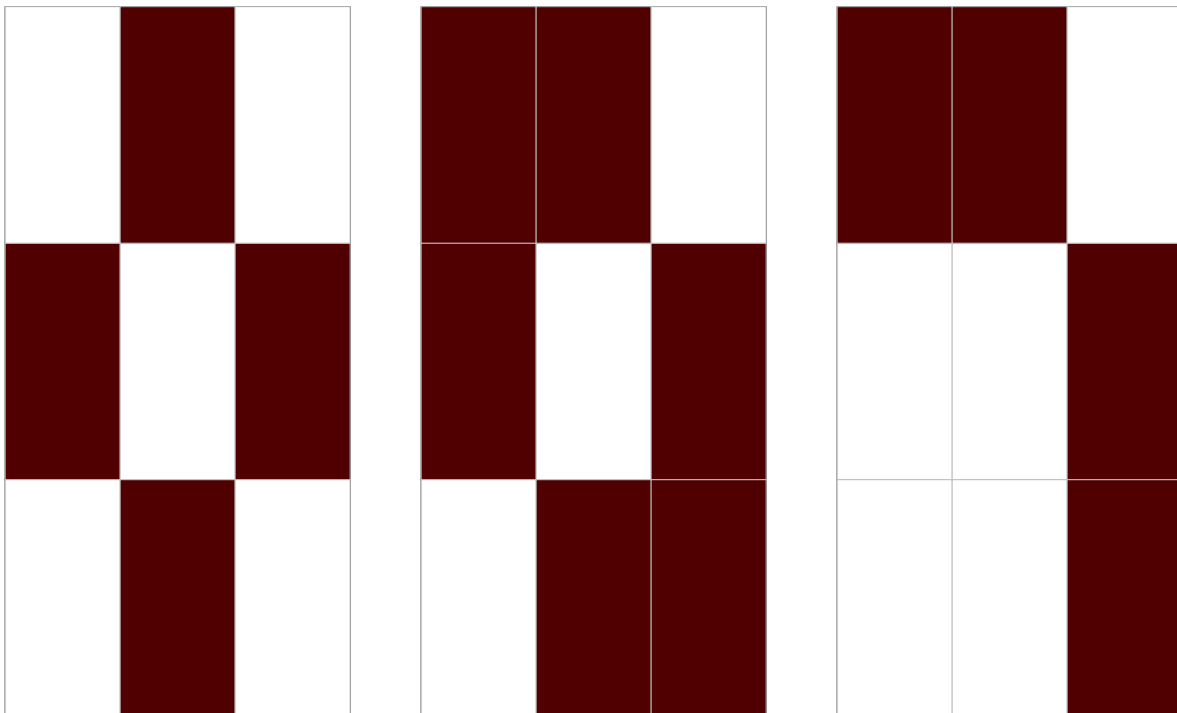
```

# visualize graphs
# pred_graphs <- plot(covdepGE_models[[12]])
true_graphs <- unique(lapply(lapply(lapply(
  results$trial1$data$true_precision, `!=`, 0), `*`, 1), `~`, diag(p)))
graph_inds <- c(paste0(c(1, nj), collapse = "..."),
               paste0(c(nj + 1, 2 * nj), collapse = "..."),
               paste0(c(2 * nj + 1, 3 * nj), collapse = "..."))
titles <- paste0("Graph ", 1:3, ", observations ", graph_inds)
true_graphs <- lapply(1:3, function(j)
  matViz(true_graphs[[j]]) +
  ggtitle(titles[[j]]) +
  geom_tile(color = "grey", size = 1e-10) +
  theme(axis.line = element_blank(),
        axis.text.x=element_blank(),
        axis.text.y = element_blank(),
        axis.ticks = element_blank(),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        legend.position = "none")
)

# pred_graphs <- ggarrange(plotlist = pred_graphs, nrow = 1, legend = F)
true_graphs <- ggarrange(plotlist = true_graphs, nrow = 1, legend = F)
true_graphs

```

Graph 1, observations 1,...,Graph 2, observations 7,...,Graph 3, observations 13,...;





```
# ggsave("simulation_study/plots/preds.pdf", pred_graphs, height = 4, width = 11)
# ggsave("simulation_study/plots/true.pdf", true_graphs, height = 4, width = 11)
```