Simulation Examples for Graphical Models

Package Setup

```
if (!require("devtools", quietly = TRUE))
  install.packages("devtools")
```

```
Warning: package 'usethis' was built under R version 4.4.1
```

```
devtools::install_github("anonstats123/SyNPar")
```

```
Skipping install of 'SyNPar' from a github remote, the SHA1 (e3765e99) has not changed
since last install.
  Use `force = TRUE` to force installation
```

```
library(PRROC)
library(SyNPar)
library(MASS)
```

```
Warning: package 'MASS' was built under R version 4.4.1
```

```
library(SILGGM)
```

```
Loading required package: Rcpp
```

```
Warning: package 'Rcpp' was built under R version 4.4.1
```

Simulation Data Generation

```
generate_precision_matrix <- function(p, b, graph_type) {
  # Initialize Omega matrix
  Omega_0 <- matrix(0, nrow = p, ncol = p)

  if (graph_type == "band") {
    # Band graph
    diag(Omega_0) <- 1
    for (i in 1:(p-1)) {
      for (j in (i+1):min(p, i+10)) {
        Omega_0[i, j] <- sign(b) * abs(b)^(abs(i - j) / 10)
        Omega_0[j, i] <- Omega_0[i, j]  # Symmetric matrix
      }
    }

  } else if (graph_type == "block") {
    # Block graph: 10 blocks, each block size 20
    for (k in seq(1, p, by = 20)) {
```

```r
      block_size <- min(20, p - k + 1)
      Omega_0[k:(k+block_size-1), k:(k+block_size-1)] <- 1
      Omega_0[k:(k+block_size-1), k:(k+block_size-1)] <- b
    }

  } else if (graph_type == "erdos") {
    # Erdos-Renyi graph
    diag(Omega_0) <- 1
    for (i in 1:(p-1)) {
      for (j in (i+1):p) {
        if (rbinom(1, 1, 0.1)) {
          samples_1 <- runif(n, min = -0.6, max = -0.2)  # [-0.6, -0.2]
          samples_2 <- runif(n, min = 0.2, max = 0.6)    # [0.2, 0.6]
          phi_ij = combined_samples <- sample(c(samples_1, samples_2), 1)
          # phi_ij <- sample(c(-0.6, -0.2, 0.2, 0.6), 1)
          Omega_0[i, j] <- Omega_0[j, i] <- phi_ij
        }
      }
    }

  } else if (graph_type == "cluster") {
    # Cluster graph: 5 blocks, each block size 40
    for (k in seq(1, p, by = 40)) {
      block_size <- min(40, p - k + 1)
      Omega_0[k:(k+block_size-1), k:(k+block_size-1)] <- 1
      for (i in k:(k+block_size-1)) {
        if (i < (k+block_size-1)) { # in case j exceeds range
          for (j in (i+1):(k+block_size-1)) {
            if (rbinom(1, 1, 0.5)) {
              samples_1 <- runif(n, min = -0.6, max = -0.2)  # [-0.6, -0.2]
              samples_2 <- runif(n, min = 0.2, max = 0.6)    # [0.2, 0.6]
              phi_ij = combined_samples <- sample(c(samples_1, samples_2), 1)
              Omega_0[i, j] <- Omega_0[j, i] <- phi_ij
            }
          }
        }
      }
    }
  }

  # Make the precision matrix positive definite
  lambda_min <- eigen(Omega_0)$values[p]
  Omega <- Omega_0 + (abs(lambda_min) + 0.5) * diag(p)

  return(Omega)
}

generate_data <- function(n, p, Omega) {
  # Generate the precision matrix Omega
  # Omega <- generate_precision_matrix(p, b, graph_type)
```

```
  # Generate the covariance matrix Sigma
  Sigma <- solve(Omega)

  # Generate n samples from N_p(0, Sigma)
  data <- mvrnorm(n = n, mu = rep(0, p), Sigma = Sigma)

  return(data)
}

# Function to obtain the true support of the precision matrix
obtain_true_support <- function(Omega, threshold = 1e-10) {
  # Obtain the true support of the precision matrix
  true_support = matrix(0, nrow = ncol(Omega), ncol = ncol(Omega))
  true_support[abs(Omega) > threshold] = 1
  return(true_support)
}

n <- 2000
p <- 200
edge_strength <- -0.8
Omega <- generate_precision_matrix(p, edge_strength, "band")
true_support = obtain_true_support(Omega)
Signal_index = which(true_support[upper.tri(true_support)] == 1)
true_labels <- true_support[upper.tri(true_support)]
X <- generate_data(n, p, Omega)
data_scale = scale(X)
```

Statistical Metrics Function

```
aupr = function(statistic) pr.curve(scores.class0 = statistic,
                                    weights.class0 = true_labels,
                                    curve = FALSE)$auc.integral
```

SyNPar

```
result_SyNPar <- synpar_filter(
  X = data_scale, fdr_value = 0.2, best_lambda = NULL, B_reps = NULL, model_type = "graph
)
```

```
Loading required package: Matrix

Warning: package 'Matrix' was built under R version 4.4.1

Loaded glmnet 4.1-8

Warning: package 'survival' was built under R version 4.4.1
```

```
Warning: package 'eha' was built under R version 4.4.1


Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 1%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 2%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 3%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 4%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 5%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 6%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 6%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 7%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 8%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 9%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 10%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 12%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 13%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 14%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 15%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 16%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 17%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 18%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 18%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 19%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 20%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 21%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 23%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 24%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 25%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 26%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 27%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 28%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 29%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 30%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 31%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 31%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 32%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 34%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 35%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 36%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 37%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 38%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 39%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 40%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 41%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 42%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 43%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 43%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 44%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 46%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 47%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 48%
```

```
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 49%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 50%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 51%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 52%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 53%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 54%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 55%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 56%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 57%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 58%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 59%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 60%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 61%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 62%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 63%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 64%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 65%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 65%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 67%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 68%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 69%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 70%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 71%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 72%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 73%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 74%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 75%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 76%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 77%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 78%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 79%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 80%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 81%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 82%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 83%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 84%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 85%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 86%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 87%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 88%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 89%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 90%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 91%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 92%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 93%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 94%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 95%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 96%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 97%
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 98%
```

```
Conducting the graphical lasso (glasso) wtih lossless screening....in progress: 99%
Conducting the graphical lasso (glasso)....done.
```

```
SyNPar_FDR <- length(which(result_SyNPar$statistic[setdiff(1:length(result_SyNPar$statist
                                                            Signal_index)] >=
                              result_SyNPar$threshold)) / max(length(result_SyNPar$selecte
SyNPar_Power <- (length(which(result_SyNPar$statistic[Signal_index] >=
                                     result_SyNPar$threshold))) / (length(Signal_index))
SyNPar_AUPR <- aupr(result_SyNPar$statistic)
cat("SyNPar FDR:", SyNPar_FDR, "\n")
```

```
SyNPar FDR: 0.08153348
```

```
cat("SyNPar Power:", SyNPar_Power, "\n")
```

```
SyNPar Power: 0.8745501
```

```
cat("SyNPar AUPR:", SyNPar_AUPR, "\n")
```

```
SyNPar AUPR: 0.9551264
```

GFC-L

```
result_GFCL <- SILGGM(data_scale, method = "GFC_L", alpha = 0.2,
                                    true_graph = Omega)
```

```
Use method '"GFC_L"'
Center each column.
Standardize each column.
Pre-calculate inner product matrixes.
Use default number of delta = 40
Perform global inference.
Use pre-specified level(s): 0.2
True graph is available.
Calculate Lasso of each variable with tuning parameters under each delta.
Record test statistics under each delta.
delta 1
Lasso for variable 1
Lasso for variable 101
delta 2
Lasso for variable 1
Lasso for variable 101
delta 3
Lasso for variable 1
Lasso for variable 101
delta 4
```

```
Lasso for variable 1
Lasso for variable 101
delta 5
Lasso for variable 1
Lasso for variable 101
delta 6
Lasso for variable 1
Lasso for variable 101
delta 7
Lasso for variable 1
Lasso for variable 101
delta 8
Lasso for variable 1
Lasso for variable 101
delta 9
Lasso for variable 1
Lasso for variable 101
delta 10
Lasso for variable 1
Lasso for variable 101
delta 11
Lasso for variable 1
Lasso for variable 101
delta 12
Lasso for variable 1
Lasso for variable 101
delta 13
Lasso for variable 1
Lasso for variable 101
delta 14
Lasso for variable 1
Lasso for variable 101
delta 15
Lasso for variable 1
Lasso for variable 101
delta 16
Lasso for variable 1
Lasso for variable 101
delta 17
Lasso for variable 1
Lasso for variable 101
delta 18
Lasso for variable 1
Lasso for variable 101
delta 19
Lasso for variable 1
Lasso for variable 101
delta 20
Lasso for variable 1
Lasso for variable 101
delta 21
```

```
Lasso for variable 1
Lasso for variable 101
delta 22
Lasso for variable 1
Lasso for variable 101
delta 23
Lasso for variable 1
Lasso for variable 101
delta 24
Lasso for variable 1
Lasso for variable 101
delta 25
Lasso for variable 1
Lasso for variable 101
delta 26
Lasso for variable 1
Lasso for variable 101
delta 27
Lasso for variable 1
Lasso for variable 101
delta 28
Lasso for variable 1
Lasso for variable 101
delta 29
Lasso for variable 1
Lasso for variable 101
delta 30
Lasso for variable 1
Lasso for variable 101
delta 31
Lasso for variable 1
Lasso for variable 101
delta 32
Lasso for variable 1
Lasso for variable 101
delta 33
Lasso for variable 1
Lasso for variable 101
delta 34
Lasso for variable 1
Lasso for variable 101
delta 35
Lasso for variable 1
Lasso for variable 101
delta 36
Lasso for variable 1
Lasso for variable 101
delta 37
Lasso for variable 1
Lasso for variable 101
delta 38
```

```
Lasso for variable 1
Lasso for variable 101
delta 39
Lasso for variable 1
Lasso for variable 101
delta 40
Lasso for variable 1
Lasso for variable 101
Choose delta for FDR control.
```

```
GFCL_Power = result_GFCL$power
GFCL_FDR = result_GFCL$FDR
GFCL_AUPR = aupr(abs(result_GFCL$T_stat[upper.tri(result_GFCL$T_stat)]))
cat("GFC-L FDR:", GFCL_FDR, "\n")
```

```
GFC-L FDR: 0.1602871
```

```
cat("GFC-L Power:", GFCL_Power, "\n")
```

```
GFC-L Power: 0.3609254
```

```
cat("GFC-L AUPR:", GFCL_AUPR, "\n")
```

```
GFC-L AUPR: 0.6674617
```

GFC-SL

```
result_GFCS <- SILGGM(data_scale, method = "GFC_SL", alpha = 0.2,
                                  true_graph = Omega)
```

```
Use method '"GFC_SL"'
Use default lambda = sqrt(2*log(p/sqrt(n))/n)
In this case, lambda = 0.0387023
Center each column.
Standardize each column.
Pre-calculate inner product matrixes.
Calculate scaled Lasso for each variable.
scaled Lasso for variable 1
scaled Lasso for variable 101
Perform global inference.
Use pre-specified level(s): 0.2
True graph is available.
```

```
GFCS_Power = result_GFCS$power
GFCS_FDR = result_GFCS$FDR
GFCS_AUPR = aupr(abs(result_GFCS$T_stat[upper.tri(result_GFCS$T_stat)]))
cat("GFC-SL FDR:", GFCS_FDR, "\n")
```

```
GFC-SL FDR: 0.1552106
```

```r
cat("GFC-SL Power:", GFCS_Power, "\n")
```

```
GFC-SL Power: 0.3917738
```

```r
cat("GFC-SL AUPR:", GFCS_AUPR, "\n")
```

```
GFC-SL AUPR: 0.6859362
```

KO2

```r
source("./Knockoff.R")
trueNoneConnections <- which(true_support == 0)
ko.est <- GraphEstimation(data_scale, FDRtarget = 0.2, plus = FALSE)
```

```
Warning in pcor(dat): The inverse of variance-covariance matrix is calculated
using Moore-Penrose generalized matrix invers due to its determinant of zero.
```

```r
ko.adj <- ko.est$adjacency.matrix
if(dim(ko.est$edge.set)[1]==0){
  ko.err<-0
} else {
  ko.FP <- ko.adj[trueNoneConnections]
  ko.err <- length(which(ko.FP == 1))
}
ko2_FDR <- ko.err / max(1, length(which(ko.adj == 1)) )
ko2_Power <- (length(which(ko.adj == 1)) -
            ko.err) / max(1, length(which(true_support == 1)))
ko2_AUPR = aupr(abs(ko.est$continuous.raw[upper.tri(ko.est$continuous.raw)]))
cat("KO2 FDR:", ko2_FDR, "\n")
```

```
KO2 FDR: 0.17348608838
```

```r
cat("KO2 Power:", ko2_Power, "\n")
```

```
KO2 Power: 0.246943765281
```

```r
cat("KO2 AUPR:", ko2_AUPR, "\n")
```

```
KO2 AUPR: 0.580655499463
```

Data Splitting

```r
source("./DS_graph.R")
source("./fdp_power_graph.R")
```

```r
source("./analys_graph.R")
selected_ds <- DS_graph(data_scale, q = 0.2, num_split = 1)
result_ds = fdp_power_graph(selected_ds$DS_selected_edge, true_support)
ds_Power <- result_ds$power
ds_FDR <- result_ds$fdp
ds_AUPR = aupr(abs(selected_ds$DS_statistic[upper.tri(selected_ds$DS_statistic)]))
cat("Data Splitting FDR:", ds_FDR, "\n")
```

```
Data Splitting FDR: 0.172081218274
```

```r
cat("Data Splitting Power:", ds_Power, "\n")
```

```
Data Splitting Power: 0.797555012225
```

```r
cat("Data Splitting AUPR:", ds_AUPR, "\n")
```

```
Data Splitting AUPR: 0.854382616505
```