Figure3 Plot Code

Overview

This doc shows reproducible plot code to generate the Figure 3 result in manuscript *Community Detection with Heterogeneous Block Covariance Model*.

Install packages

Install the HBCM package from github and other helper packages.

```r
# install helper package
packages_helper <- c(
  'devtools', "parallel",
  "foreach", "doParallel",
  "kernlab", "matrixcalc",
  "tidyverse", "ggpubr",
  "scales", "ggforce",
  "patchwork"
  )
install.packages(packages_helper, repos = "http://cran.us.r-project.org")

# install hbcm package
devtools::install_github("xiangli2pro/hbcm")

# load all required packages
lapply(c('hbcm', packages_helper), require, char = TRUE)

# set random seed to reproduce the result
set.seed(2022)
```

Simulation data and Gene expression data

Prepare simulation data and gene expression data for the cross-validation to estimate the cluster number K.

Gene expression data used in the manuscript is already loaded once the HBCM package is loaded. Execute the command below to see details in the Help panel on how data is processed.

```
?hbcm::data_gene_sample
?hbcm::data_gene_groups
```

Generate simulation data.

```
# data size and cluster number
centers <- 5 # number of Classes
n <- 1500 # number of Observations
p <- 500 # number of Genes

# mean vector
mu <- rep(0, centers)

# cluster level covariance matrix
off_diag <- 0.2
omega <- diag(rep(1, centers))
for (i in 1:centers) {
  for (j in 1:centers) {
    if (i != j) {
      omega[i, j] <- off_diag
    }
  }
}

# equally distributed class
ppi <- rep(1 / centers, centers)
labels <- sample(c(1:centers), size = p, replace = TRUE, prob = ppi)

# individual hlambda and hsigma
hlambda <- rep(1, p)
hsigma <- rep(6, p)

# sample data
```

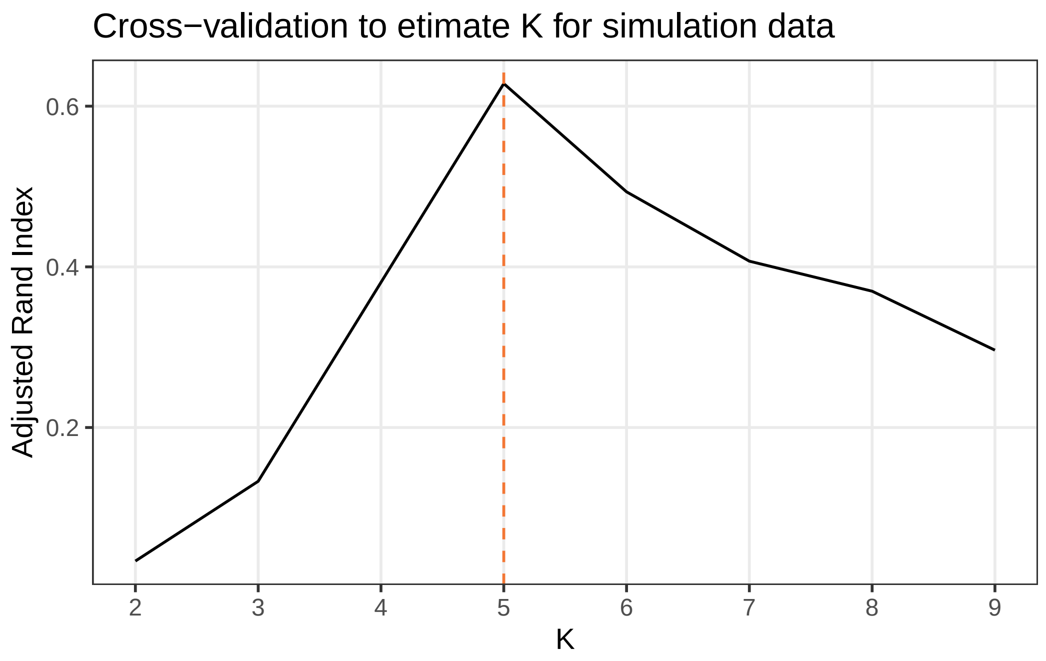
```r
sample_gen <- function(n, p, mu, omega, labels, hlambda, hsigma) {

  alpha <- MASS::mvrnorm(n, mu, omega)
  x <- matrix(rep(0, n * p), nrow = n)
  for (i in 1:n) {
    for (j in 1:p) {
      x[i, j] <- hlambda[j] * (alpha[i, labels[j]]) + hsigma[j] * rnorm(1, 0, 1)
    }
  }

  list(
    x = x, alpha = alpha,
    hlambda = hlambda, hsigma = hsigma
  )
}

data_list <- sample_gen(n, p, mu, omega, labels, hlambda, hsigma)
X <- data_list$x
```

Use cross-validation to estimate cluster number K

Execute `?hbcm::crossValid_func_adjR` to see the details of input arguments. The input argument `pt` determines how many times data is partitioned to 2 parts for each candidate K. The larger `pt` , the more accurate yet more time consuming. The manuscript is setting `pt=20`. For demonstration purpose, we set `pt=10` .

```r
pt <- 10

# parallel process
registerDoParallel(detectCores())

# candidate K
K_candidate <- c(2:9)
# get adjRand from cross-validation
res <- foreach(
  K = K_candidate, .errorhandling = "pass",
  .packages = c("MASS", "Matrix", "matrixcalc", "kernlab", "RSpectra")
) %dopar%
  hbcm::crossValid_func_adjR(X, K, pt)
```

```r
# remove NaN
cv_res <- sapply(res, function(x) {
  if (length(x) == 1) {
    x
  } else {
    NA
  }
})

# prepare data for plot
cv_data <- cbind(K_candidate, cv_res) %>%
  `colnames<-`(c("K", "logLikelihood")) %>%
  as.data.frame()

# make plot
ggplot(cv_data) +
  geom_line(aes(x = K, y = logLikelihood)) +
  xlab("K") +
  ylab("Adjusted Rand Index") +
  ggtitle('Cross-validation to etimate K for simulation data') +
  theme_bw() +
  theme(panel.grid.minor = element_blank()) +
  scale_x_continuous(breaks = c(2:9)) +
  geom_vline(xintercept = 5, linetype = "dashed", color = "#f37735")
```



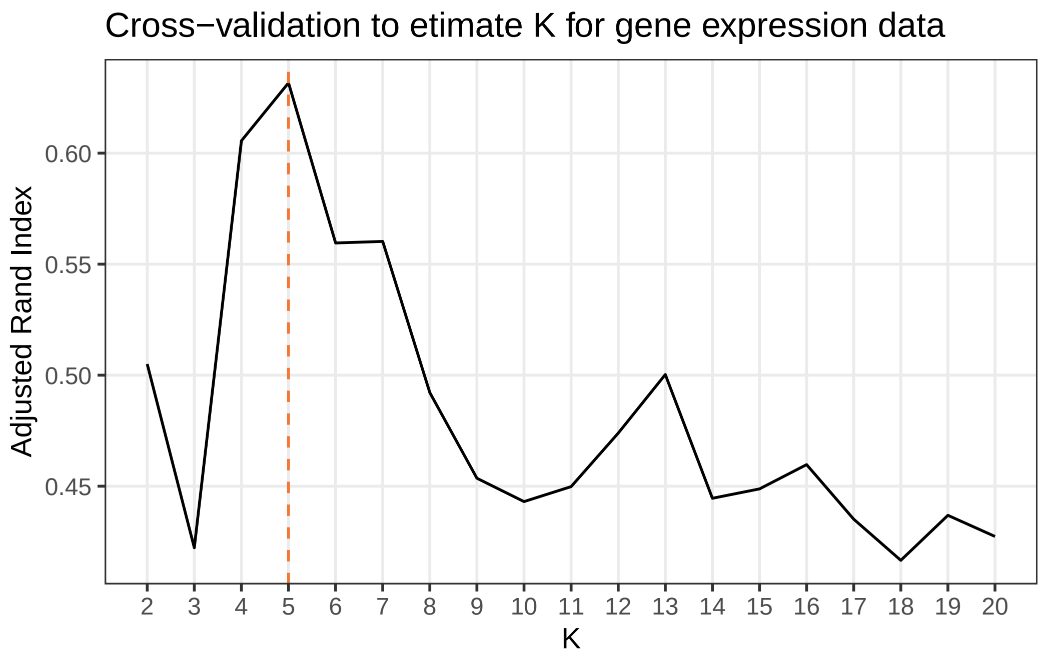
Make heat-map plot

Rearrange the genes by their HBCM estimated clusters and make heatmap plot. Note that the black line around each block is added intentionally to visually distinguish the clusters.

```
pt <- 10

# parallel computing
registerDoParallel(detectCores())

# cross-validation to calculate adjRand
K_candidate_gene <- c(2:20)
res_adjR <- foreach(
  K = K_candidate_gene, .errorhandling = "pass",
  .packages = c("MASS", "Matrix", "matrixcalc", "kernlab", "RSpectra")
) %dopar%
  crossValid_func_adjR(data_gene_sample, K, pt)
```

```
cbind(K_candidate_gene, res_adjR %>% unlist()) %>%
  `colnames<-`(c("K", "adjR_mean")) %>%
  as.data.frame() %>%
```

```r
ggplot() +
geom_line(aes(x = K, y = adjR_mean)) +
xlab("K") +
ylab("Adjusted Rand Index") +
ggtitle('Cross-validation to etimate K for gene expression data') +
theme_bw() +
theme(panel.grid.minor = element_blank()) +
scale_x_continuous(breaks = K_candidate_gene) +
geom_vline(xintercept = 5, linetype = "dashed", color = "#f37735")
```


Cross−validation to etimate K for gene expression data
Adjusted Rand Index
K