2022 SISBID Graphical Models Lab

Genevera I. Allen and Yufeng Liu

Load packages

```
library("igraph")
library("huge")
library("glasso")

library("WGCNA")
library("glmnet")
library("ggplot2")
```

Read the Sachs et al data: Flow cytometry proteomics in single cells, $p = 11$ proteins measured in $n = 6466$ cells.

```
sachscov <- as.matrix(read.table("sachscov.txt"))
sachscor <- cov2cor(sachscov)
sachsdat <- as.matrix(read.table("sachs.data.txt"))
ps <- c("praf","pmek","plcg","PIP2","PIP3","P44","pakts","PKA","PKC","P38","pjnk")
colnames(sachsdat) <- ps
p <- ncol(sachsdat)
n <- nrow(sachsdat)
dim(sachsdat)
```

```
## [1] 7466   11
```

```
dim(sachscov)
```

```
## [1] 11 11
```

```
head(sachsdat)
```

```
##            praf       pmek      plcg      PIP2       PIP3        P44     pakts
## [1,]  -97.67193 -132.18100 -46.03364 -132.8207  31.765040 -20.021190 -64.16721
## [2,]  -88.17193 -128.88100 -42.55364 -134.3207 -18.904960  -8.031193 -48.66721
## [3,]  -64.67193 -101.28100 -40.25364 -140.9207 -14.034960 -11.731190 -48.66721
## [4,]  -51.07193  -62.58096 -31.75364 -137.6207 -25.744960 -20.801190 -69.36721
## [5,]  -90.37193 -125.58100 -49.66364 -141.3907  -2.234962  -5.531193 -35.06721
## [6,] -105.27190 -141.63100 -37.25364 -129.0207 -16.134960 -14.731190 -55.46721
##             PKA       PKC       P38       pjnk
## [1,] -211.75860 -13.34166  -90.1145 -33.267500
## [2,] -273.75860 -26.97166 -118.5145 -11.767500
## [3,] -222.75860 -18.94166 -103.1145 -53.767500
## [4,]  -97.75859 -16.64166 -106.4145 -50.167500
## [5,] -320.75860 -25.68166 -109.3145   8.032497
## [6,]  -15.75859 -16.64166  -85.9145 -15.467500
```

```
load("UnsupL_SISBID_2022.Rdata")
```

Coexpression network

simple thresholding of correlations, at a cutoff chosen to give similar number of edges to partial correlation methods

- a randomly chosen threshold

```
tau <- 0.1
A1 <- abs(sachscor) > tau
diag(A1) <- 0
sum(A1)/2
```

```
## [1] 43
```

testing for nonzero correlations

- testing for nonzero correlation, using Fisher Z-transform

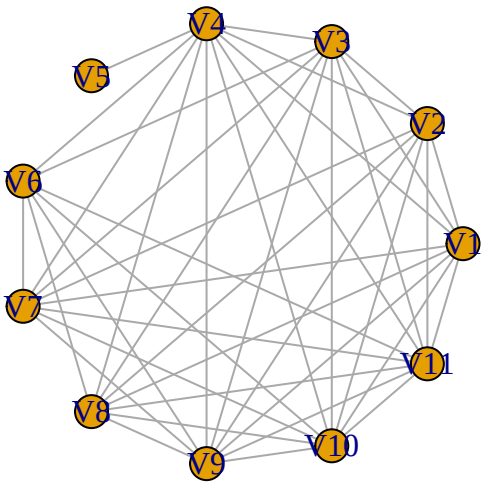
```
fisherzs <- atanh(sachscor)
fisherps  <- 2*pnorm(abs(fisherzs), 0, 1/sqrt(n-3), lower.tail=FALSE)
A2 <- fisherps < (0.01/(p*(p-1)/2))
diag(A2) <- 0
sum(A2)/2
```

```
## [1] 45
```

plot the three networks

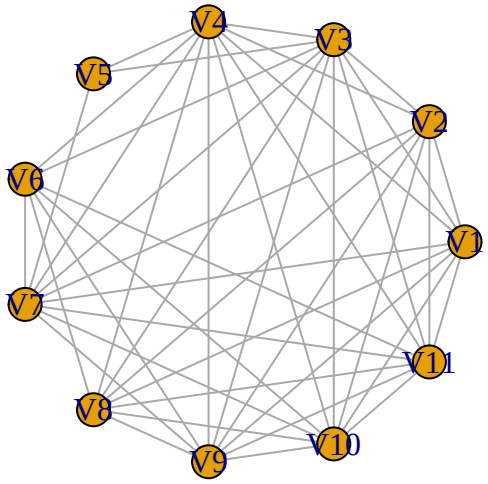
```
g1 <- graph.adjacency(A1, mode="undirected")
g2 <- graph.adjacency(A2, mode="undirected")
g0 <- g2
```

```
plot(g1,layout=layout.circle(g1), main='simple thresholding of correlations')
```




```
plot(g2,layout=layout.circle(g2), main='testing for nonzero correlations')
```

testing for nonzero correlations



Partial correlation networks(under gaussian graphical models assumption)

inverse covariance matrix

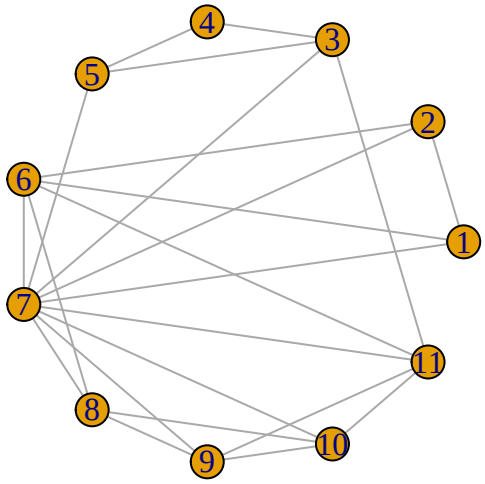
```
invcov <- abs(round(solve(sachscor),3))
invcor <- cov2cor(invcov)
A1 <- 1*(invcor > 0.05)
diag(A1) <- 0
sum(A1)/2
```

```
## [1] 23
```

```
g1 <- graph.adjacency(A1, mode="undirected")
```

```
plot(g1,layout=layout.circle(g1),main = "Partial correlation networks")
```

Partial correlation networks



Graphical lasso

Calculate lambda, based on formula in the slides (the third method)

```
alpha <- 0.01
num <- qt(p=alpha/(2*(p^2)),df=n-2, lower.tail=F)
lambda <- num / sqrt(n-2 + num)
```

Apply glasso

```
glasso.est <- glasso(s=sachscor,rho=lambda*4.2,approx=FALSE,
                     penalize.diagonal=FALSE)
A2 <- abs(glasso.est$wi) > 1E-16
diag(A2) <- 0
g2 <- graph.adjacency(A2, mode="undirected")
```

Neighborhood selection

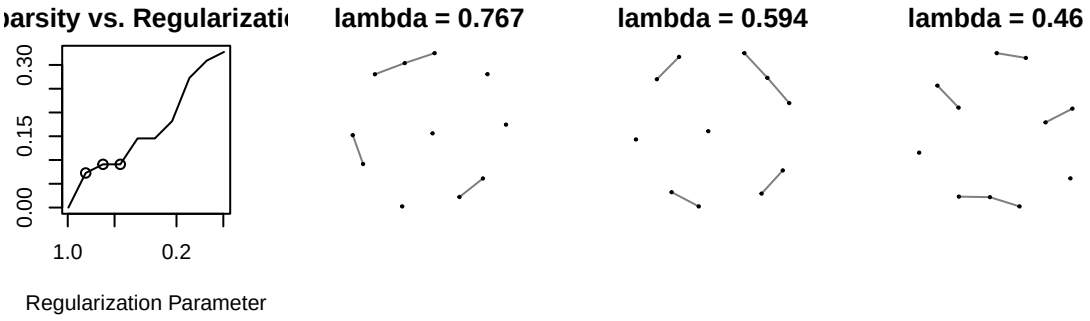
```
ns.est <- glasso(s=sachscor, rho=lambda, approx=TRUE, penalize.diagonal=FALSE)
A3 <- abs(ns.est$wi) > 1E-16; diag(A3) <- 0
g3 <- graph.adjacency(A3, mode="undirected")
```

Neighborhood selection estimate with huge (Stability selection for the value of λ)

```
X <- data.matrix(scale(sachsdat))
neth = huge(X,method="mb")
```

```
## Conducting Meinshausen & Buhlmann graph estimation (mb)....done
```

```
plot(neth)
```



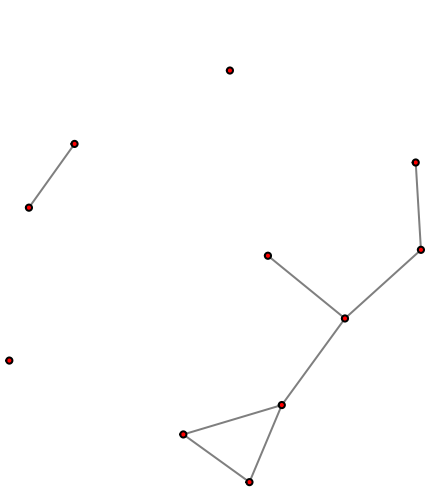

```
## stability selection with huge
net.s <- huge.select(neth, criterion="stars")
```

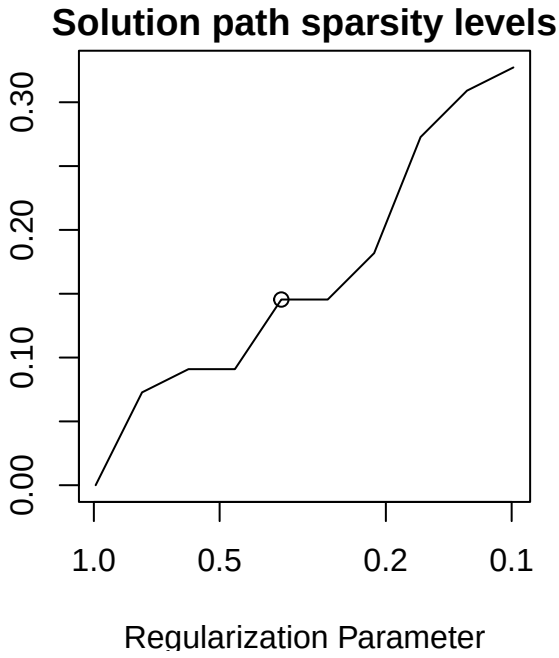
```
## Conducting Subsampling....in progress:5% Conducting Subsampling....in progress:10% Conducting Subsamp
```

```
net.s
```

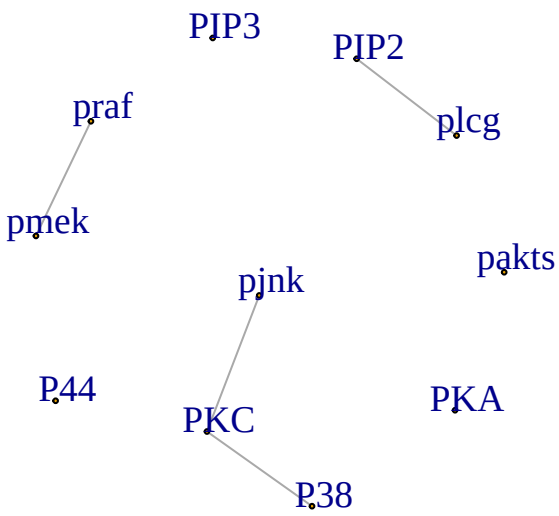
```
## Model: Meinshausen & Buhlmann Graph Estimation (mb)
## selection criterion: stars
## Graph dimension: 11
## sparsity level 0.1454545
```

```
plot(net.s)
```

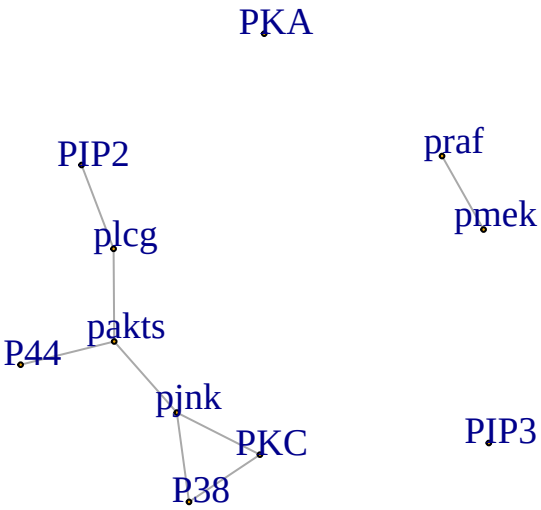





```
#larger lambda
mat <- neth$path[[2]]
neti <- as.undirected(graph_from_adjacency_matrix(mat))
plot(neti,vertex.label=colnames(X),vertex.size=2,vertex.label.cex=1.2,vertex.label.dist=1,layout=layout
```




```
#smaller lambda
mat = neth$path[[5]]
neti = as.undirected(graph_from_adjacency_matrix(mat))
plot(neti,vertex.label=colnames(X),vertex.size=2,vertex.label.cex=1.2,vertex.label.dist=1,layout=layout
```



Nonparanormal Models: rank-based correlation

```
scor <- cor(sachsdat,method='spearman')
scor <- 2*sin(scor*pi/6)
npn.est <- glasso(s=scor, rho=lambda, approx=FALSE, penalize.diagonal=FALSE)
A4 <- abs(npn.est$wi) > 1E-16
diag(A4) <- 0
g4 <- graph.adjacency(A4, mode="undirected")
```

Nonparanormal Models – alternative estiamtion

```
npn.cor <- huge.npn(x=sachsdat, npn.func="skeptic", npn.thresh=NULL, verbose=FALSE)
npn.est <- glasso(s=npn.cor, rho=lambda, penalize.diagonal=FALSE)
A5 <- abs(npn.est$wi) > 1E-16
diag(A5) <- 0
g5 <- graph.adjacency(A5, mode="undirected")
```

binary network estimation

```
sachsbin <- 1*(sachsdat > 0) + -1*(sachsdat <= 0)
head(sachsbin)
```

```
##      praf pmek plcg PIP2 PIP3 P44 pakts PKA PKC P38 pjnk
## [1,]   -1   -1   -1   -1    1  -1    -1  -1  -1  -1   -1
## [2,]   -1   -1   -1   -1   -1  -1    -1  -1  -1  -1   -1
## [3,]   -1   -1   -1   -1   -1  -1    -1  -1  -1  -1   -1
## [4,]   -1   -1   -1   -1   -1  -1    -1  -1  -1  -1   -1
## [5,]   -1   -1   -1   -1   -1  -1    -1  -1  -1  -1    1
## [6,]   -1   -1   -1   -1   -1  -1    -1  -1  -1  -1   -1
```

```
bin.est <- matrix(0,p,p)
## estiamte the neighborhood for each node
for(j in 1:p){
  ## this is the same method used in neighborhood selection, the only difference is 'family'
  nbr <- glmnet(x=sachsbin[,-j], y=sachsbin[,j], family='binomial', lambda=lambda)
  bin.est[j,-j] <- 1*(abs(as(nbr$beta,"matrix")) > 0)   #store the estimates in jth row of matrix
```

```
}
A6 <- bin.est
diag(A6) <- 0
sum(A6)/2
```

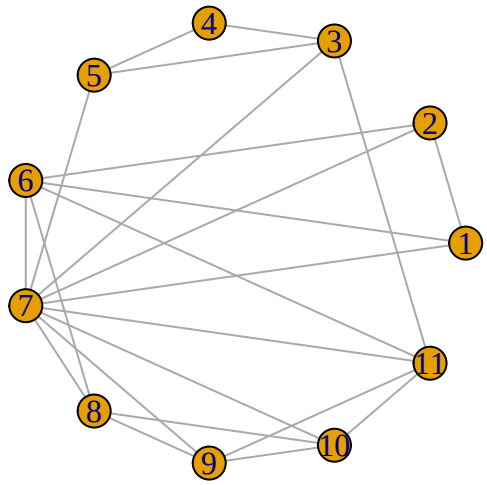
```
## [1] 20
```

```
g6 <- graph.adjacency(A6, mode="undirected")
```

plot the networks

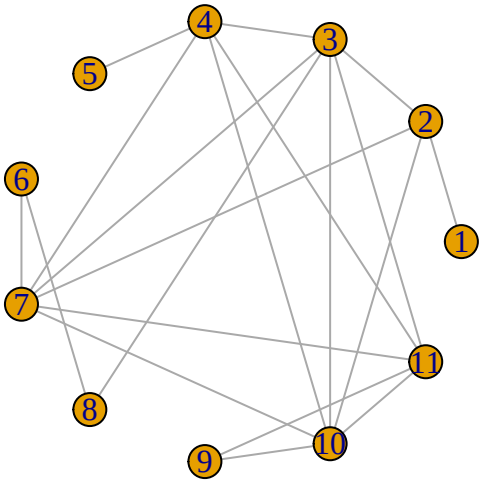
```
plot(g1,layout=layout.circle(g1), main='Partial correlation networks')
```

Partial correlation networks

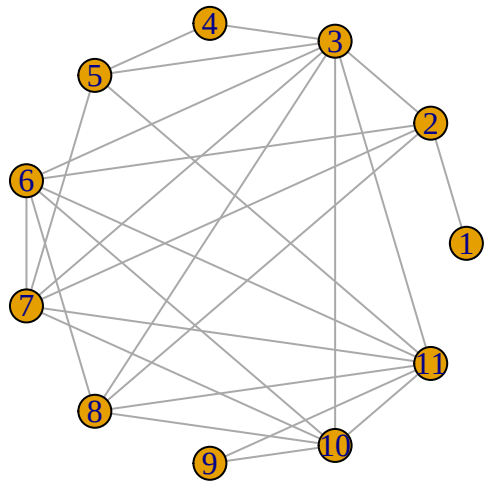



```
plot(g2,layout=layout.circle(g2), main='Glasso')
```

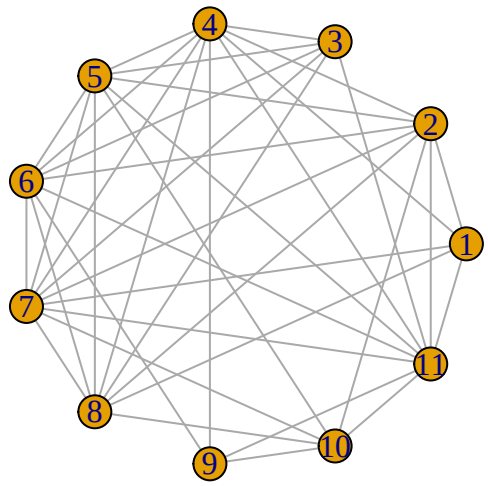
Glasso



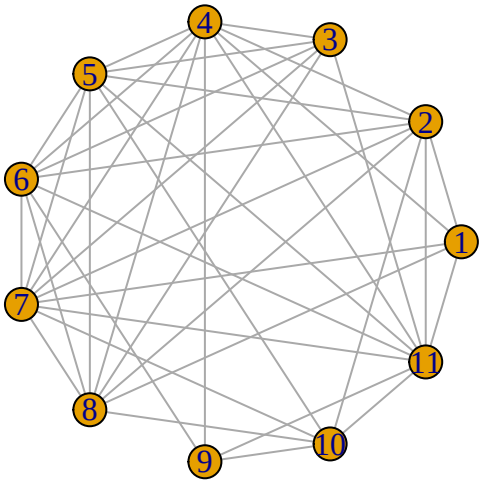
```
plot(g3,layout=layout.circle(g3), main='Neighborhood selection')
```



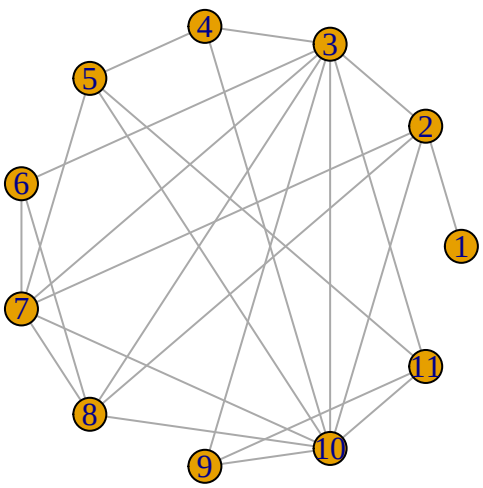

```
plot(g4,layout=layout.circle(g4), main='nonparanormal')
```




```
plot(g5,layout=layout.circle(g5), main='nonparanormal - v2')
```




```
plot(g6,layout=layout.circle(g6), main='Binary')
```

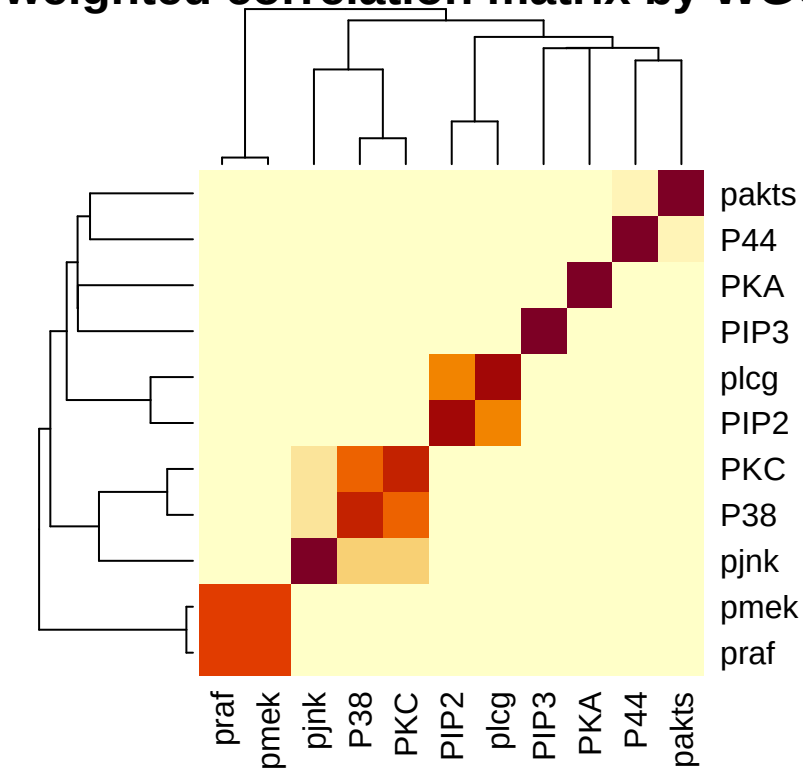


WGCNA package

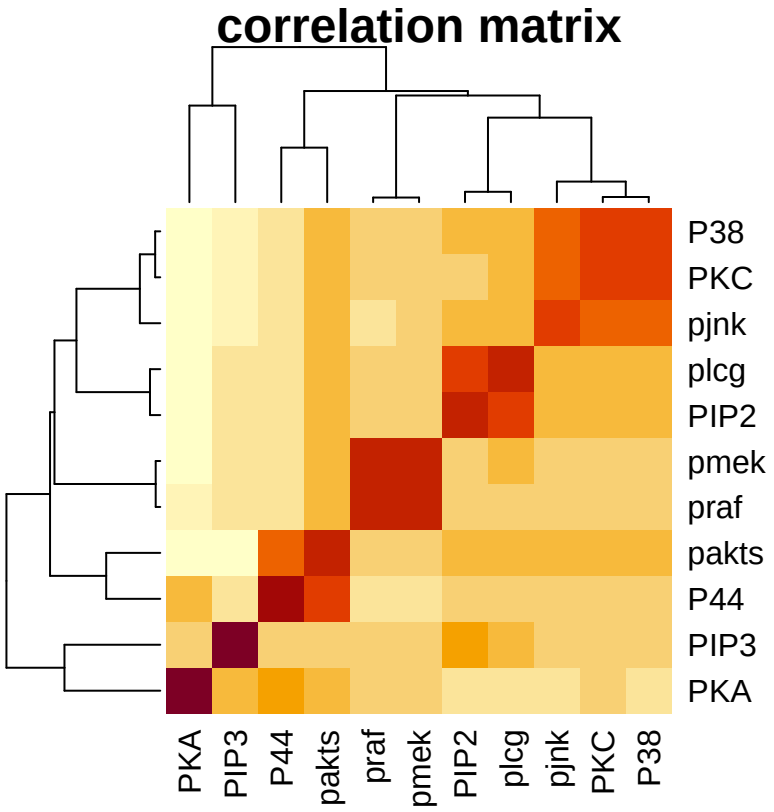
```
# construct a weighted network
adj_wgcna <- adjacency(sachsdat,power = 6)
```

compare with correlation matrix and thresholded correlation matrix

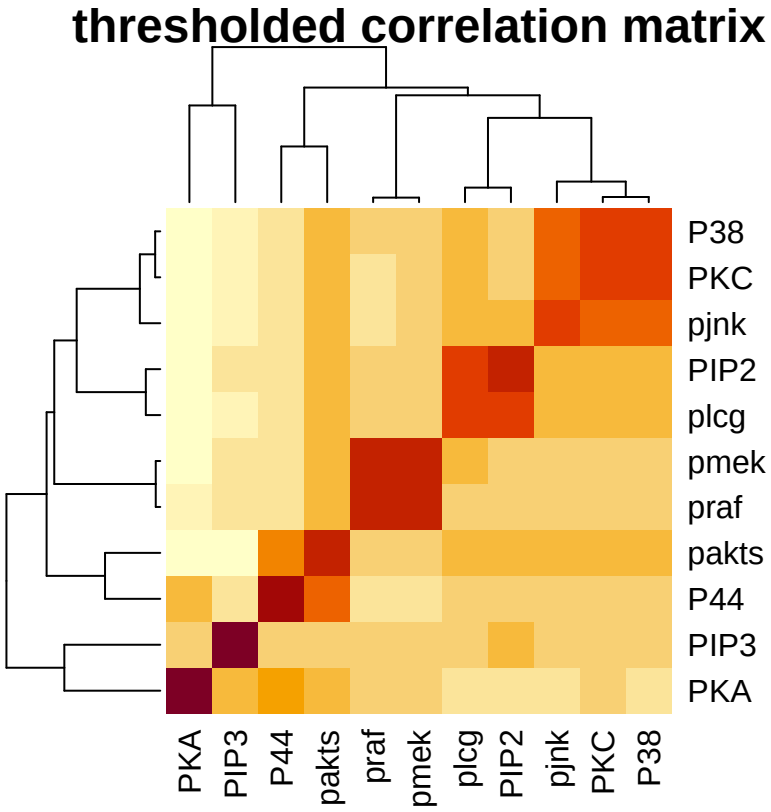
```
heatmap(adj_wgcna,main = "weighted correlation matrix by WGCNA")
```




```
colnames(sachscor) = colnames(sachsdat)
rownames(sachscor) = colnames(sachsdat)
heatmap(sachscor,main = "correlation matrix")
```




```
thresholded_correlation <- sachscor*(abs(sachscor) > 0.1)
colnames(thresholded_correlation) = colnames(sachsdat)
rownames(thresholded_correlation) = colnames(sachsdat)
heatmap(thresholded_correlation, main = "thresholded correlation matrix")
```



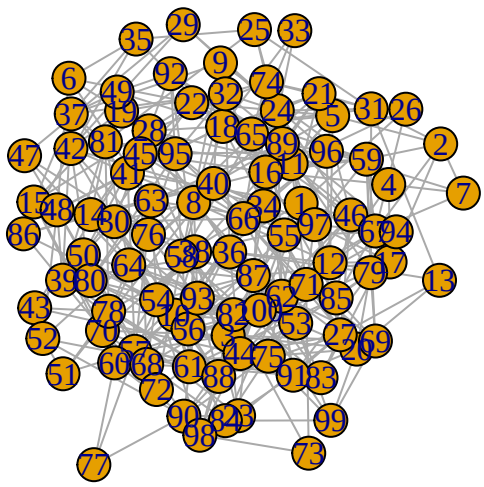
Community detection with stochastic block models

```
# generation function
gen.A.from.B <- function(B,n,c,undirected=TRUE){
  g <- vector()
  K <- length(c)
  for(i in 1:(K-1)){
    g <- c(g,rep(i,c[i]*n))
  }
  g <- c(g,rep(K,n - length(g)))
  Z <- matrix(0,n,K)
  Z[cbind(1:n,g)] <- 1
  P <- Z%*%B%*%t(Z)
  n <- nrow(P)
  if(undirected){
    upper.tri.index <- which(upper.tri(P))
    tmp.rand <- runif(n=length(upper.tri.index))
    #A <- matrix(0,n,n)
    A <- rsparsematrix(n,n,0)
    A[upper.tri.index[tmp.rand<P[upper.tri.index]]] <- 1
    A <- A+t(A)
    diag(A) <- 0
    return(list(A=A,g=g))
  }else{
    A <- matrix(0,n,n)
    r.seq <- runif(n=length(P))
    A[r.seq < as.numeric(P)] <- 1
```

```
    diag(A) <- 0
    return(list(A=A,g=g))
  }
}
```

visualization with a small network

```
n <- 100
K <- 2
B <- matrix(0.05,K,K)
diag(B) <- 0.1
c <-  rep(1/K,K)
graph <- gen.A.from.B(B,n,c)
A <- graph$A
true_label <- graph$g
neti = as.undirected(graph_from_adjacency_matrix(A))
plot(neti,layout=layout_with_kk, main='stochastic block models')
```

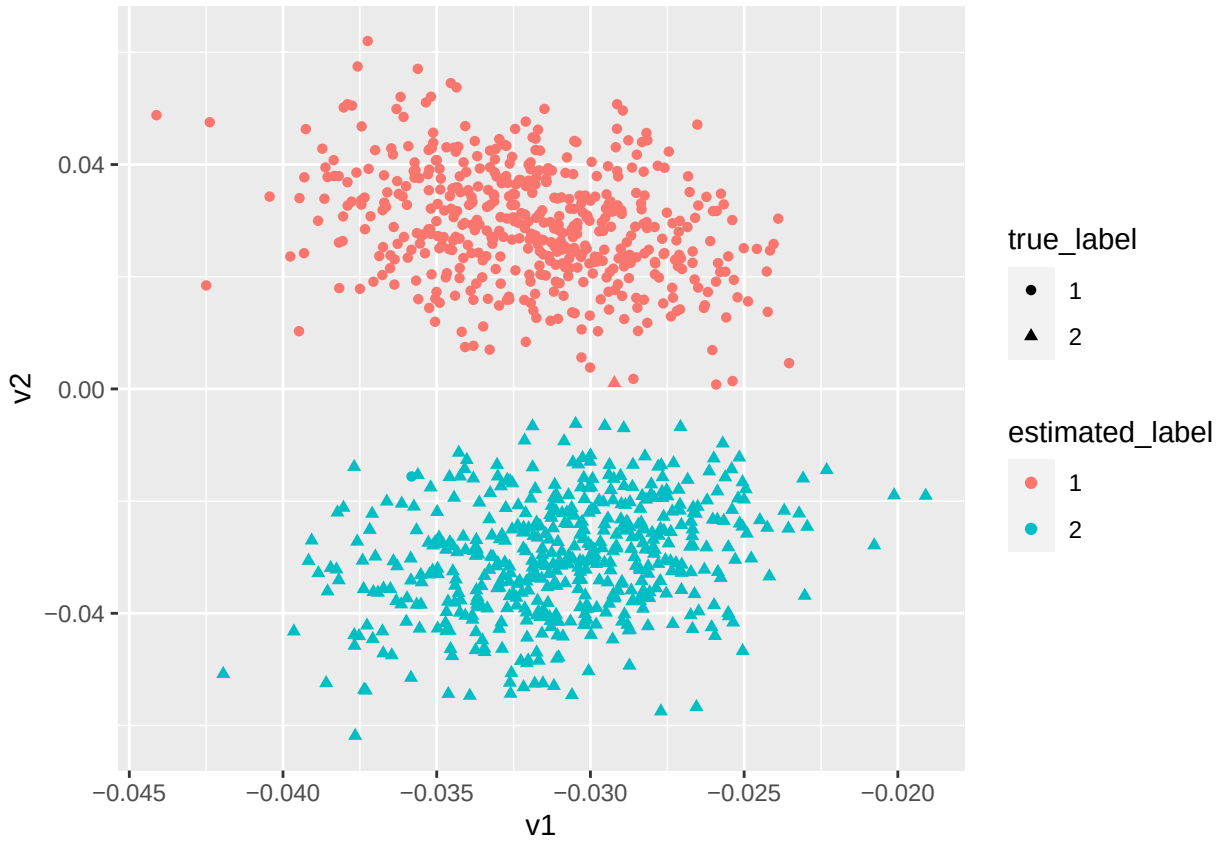


community detection with stochastic block models

```
n <- 1000
K <- 2
B <- matrix(0.05,K,K)
diag(B) <- 0.1
c <-  rep(1/K,K)
graph <- gen.A.from.B(B,n,c)
A <- graph$A
true_label <- graph$g
evA <- RSpectra::eigs(A,k = K)
clusterA <- kmeans(evA$vectors,K)
estimated_label <- clusterA$cluster
```

plot

```
newdata <- data.frame(v1 = evA$vectors[,1], v2 = evA$vectors[,2], true_label = as.factor(true_label), es
ggplot(newdata)+
  geom_point(aes(x = v1,y = v2,colour = estimated_label, shape = true_label))
```


0.04
0.00
−0.04
v2
−0.045
−0.040
−0.035
−0.030
−0.025
−0.020
v1
true_label
1
2
estimated_label
1
2