# 2023 SISBID Unsupervised Lab

Genevera I. Allen & Yufeng Liu

## Data Description

### gdat is Gene Expression Data, n = 445 patients x p = 353 genes

- Only 353 genes with somatic mutations from COSMIC are retained

### Data is Level III TCGA BRCA RNA-Sequencing gene expression data that have already been pre-processed according to the following steps:

- Reads normalized by RPKM
- Corrected for overdispersion by a log-transformation (1 + data)
- Short gene name labels are given as the column names

### cdat is Clinical Data, n = 445 patients x q = 6 clinical features

- Subtype - denotes 5 PAM50 subtypes including Basal-like, Luminal A, Luminal B, HER2-enriched, and Normal-like
- ER-Status - estrogen-receptor status
- PR-Status - progesterone-receptor status
- HER2-Status - human epidermal growth factor receptor 2 status
- Node - number of lymph nodes involved
- Metastasis - indicator for whether the cancer has metastasized

## Problems

### Problem 1 - Dimension reduction

1a - Apply PCA, NMF, ICA and MDS, UMAP, and tSNE to this dataset. Compare and contrast the results using these methods.
1b - Relate the dimension reduction results with the clinical data. Is any clinical information reflected in the lower dimensional spaces?

1c - Overall, which dimension reduction method do you recommend for this data set and why?

### Problem 2 - Clustering

2a - Apply various clustering algorithms such as K-means (explore different K), hierarchical clustering (explore different linkages), NMF, and biclustering. Compare the clustering results using these methods.

2b - Relate the clustering results with the clinical data. Can the clustering algorithm recover some of the clinical information such as cancer subtypes?

2c - (Optional) Validate your cluster findings.

2c - Overall, which clustering method(s) do you recommend for this data set and why?

### Problem 3 - Multiple comparisons

3a - Identify important genes to differetiate ER postive and negative, PR postive and negative, HER2 postive and negative, metastasis status.

3b - Try different procedures to adjust for multiple comparisons.

3c - Examine the lists of genes identified using different methods for each clinical response. Which method is best? Why?

### Problem 5 - Graphical models

5a - Use graphical models to explore interactions among genes. Are any of the well-connected genes related to patterns previously identified?

### Problem 6 - Visulaization

6a - Visualize this data using multiple approaches.

6b - Prepare the "best" visual summary of this data.

### Problem 7 - Exploratory Data Analysis Summary

7a - What is the most interesting finding?

7b - Is this finding consistent and stable?

7c - Prepare a visual summary that best illustrates this interesting finding.

## R scripts to help out with the BRCA case study Lab

### Don't peek at this if you want to practice coding on your own!!

Load Data

```
load("UnsupL_SISBID_2023.Rdata")
library(ggplot2)
library(kknn)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(umap)
library(Rtsne)
library(igraph)
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

```r
library(huge)
```

Explore Data

```r
dim(gdat)
```

```
## [1] 445 353
```

```r
dim(cdat)
```

```
## [1] 445   6
```

```r
# clinical data
table(cdat$Subtype)
```

```
##
##    Basal-like HER2-enriched     Luminal A     Luminal B   Normal-like
##            79            53           200           106             7
```

```r
table(cdat$ER)
```

```
##
##               Indeterminate                    Negative
##                           2                         100
##               Not Performed Performed but Not Available
##                           2                           2
##                    Positive
##                         339
```

```r
table(cdat$PR)
```

```
##
##               Indeterminate                    Negative
##                           3                         147
##               Not Performed Performed but Not Available
##                           2                           2
##                    Positive
##                         291
```

```r
table(cdat$HER2)
```

```
##
##     Equivocal      Negative Not Available      Positive
##             5           370             4            65
```

```r
table(cdat$Node)
```

```
##
##   0   1   2   3
## 221 146  54  23
```

```r
table(cdat$Metastasis)
```

```
##
##   0   1
## 427  11
```

```r
table(cdat$ER,cdat$PR)
```

```
##
##                              Indeterminate Negative Not Performed
```

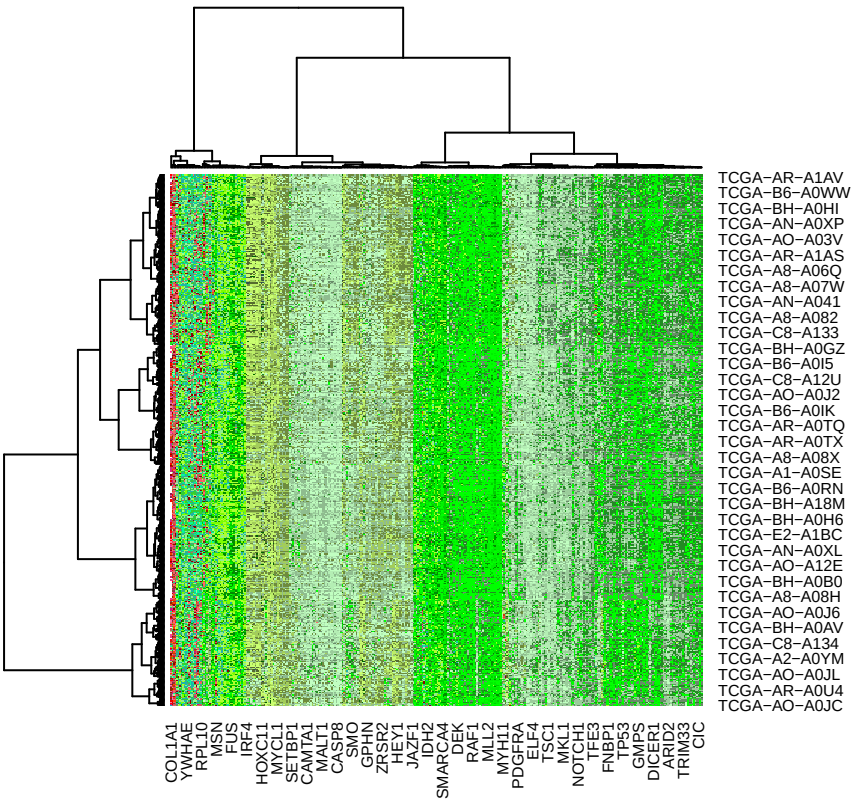
```
##    Indeterminate                          0        1             0
##    Negative                               1       93             0
##    Not Performed                          0        0             2
##    Performed but Not Available            0        0             0
##    Positive                               2       53             0
##
##                               Performed but Not Available Positive
##    Indeterminate                                        0        1
##    Negative                                             0        6
##    Not Performed                                        0        0
##    Performed but Not Available                          2        0
##    Positive                                             0      284
```

## Cluster Heatmap - biclustering

```
#cluster heatmap - biclustering
aa = grep("grey",colors())
bb = grep("green",colors())
cc = grep("red",colors())
gcol2 = colors()[c(aa[1:2],bb[1:25],cc[1:50])]
```

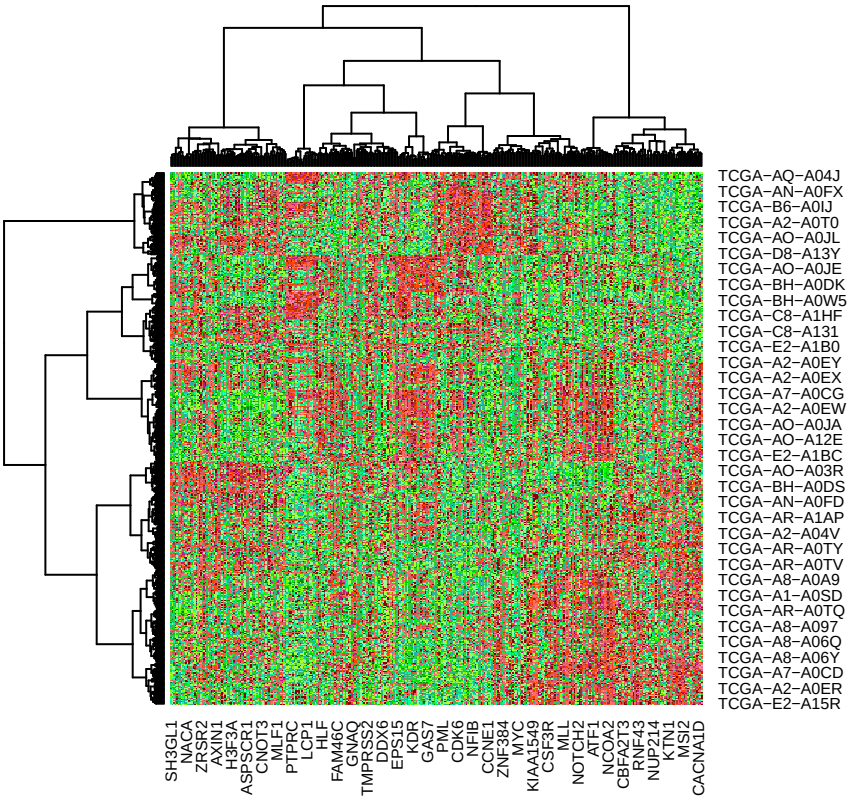
Without scaling

```
heatmap(gdat,col=gcol2,hclustfun=function(x)hclust(x,method="ward.D"))
```

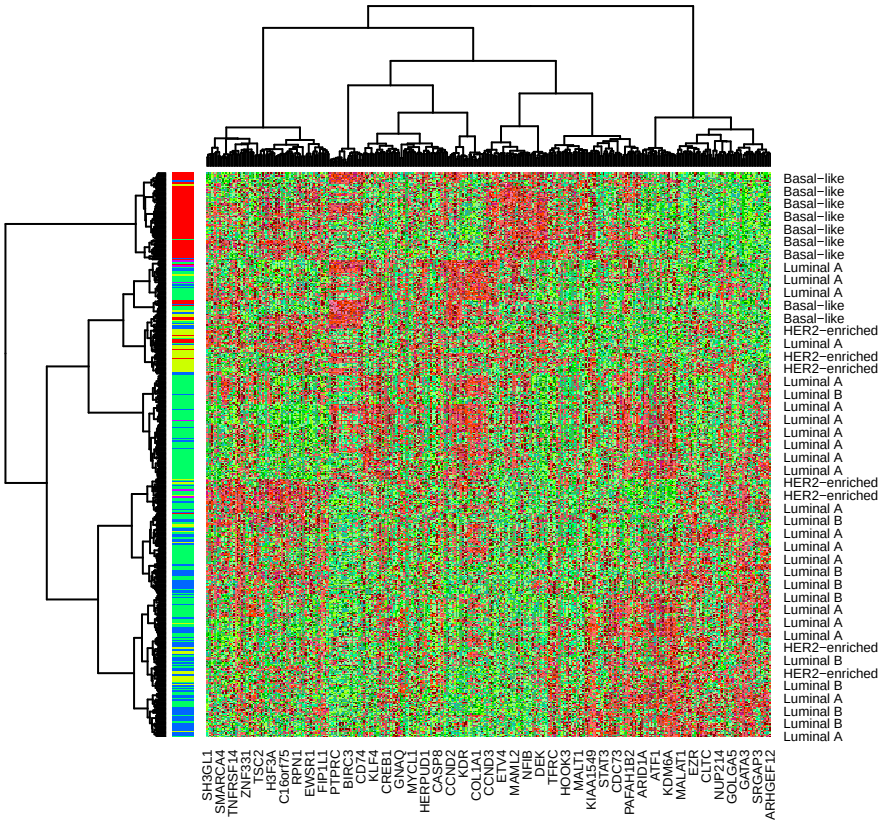



With scaling

```
heatmap(scale(gdat),col=gcol2,hclustfun=function(x)hclust(x,method="ward.D"))
```



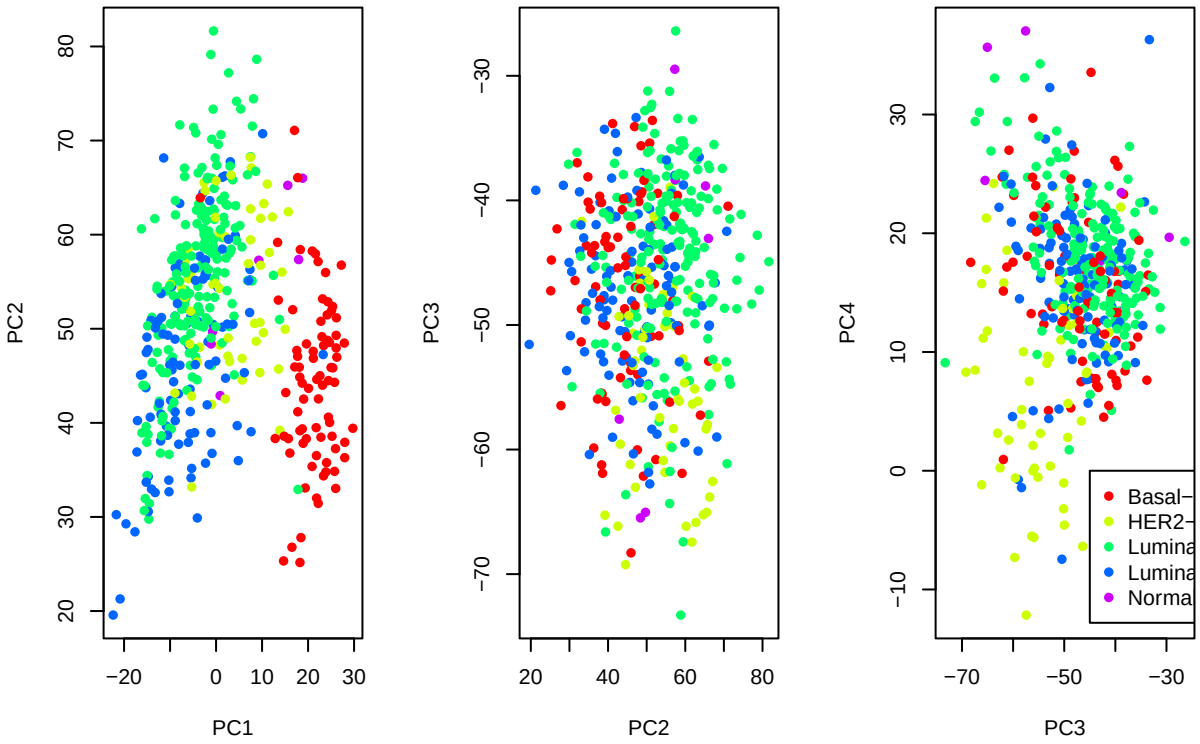

```
Cols=function(vec){cols=rainbow(length(unique(vec)))
                   return(cols[as.numeric(as.factor(vec))])}

heatmap(scale(gdat),col=gcol2,hclustfun=function(x)hclust(x,method="ward.D"),labRow=cdat$Subtype,RowSid
```




## Dimension Reduction

PCA

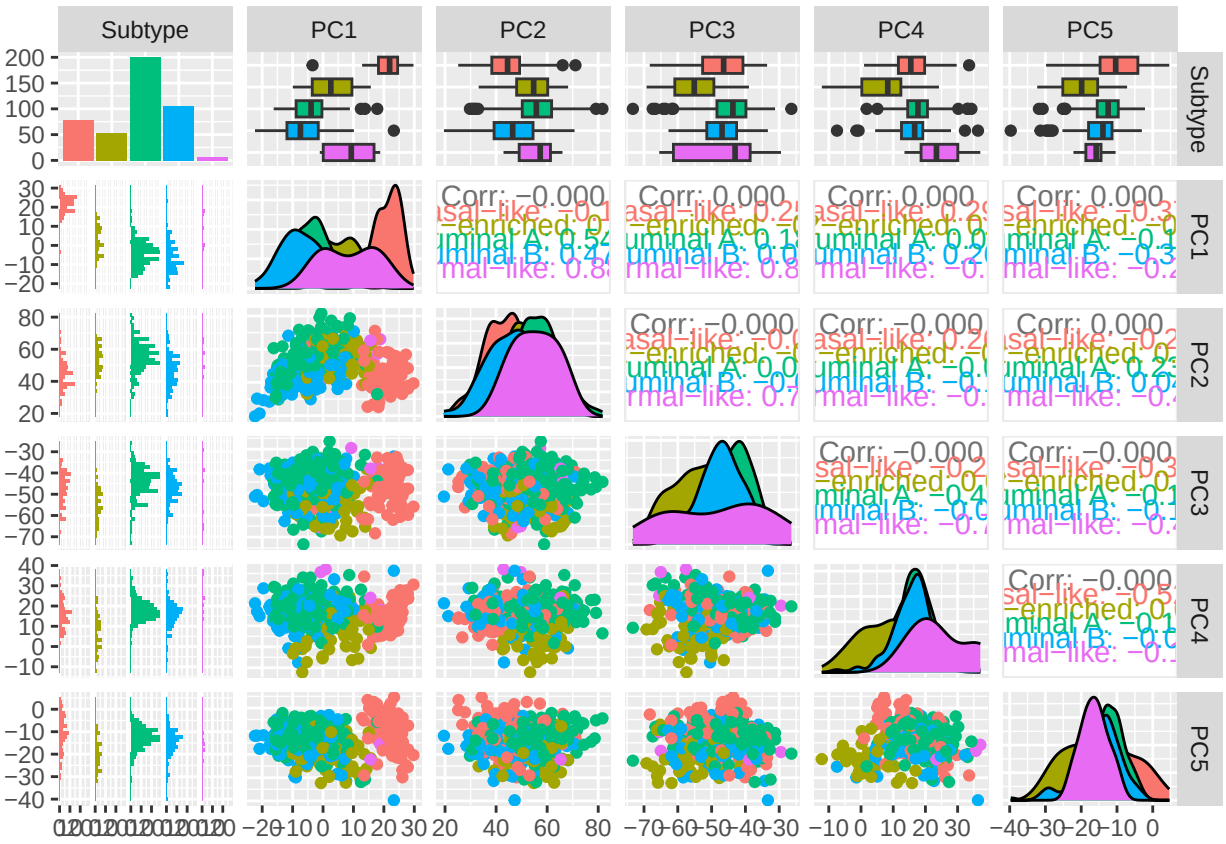
```
Cols=function(vec){cols=rainbow(length(unique(vec)))
                   return(cols[as.numeric(as.factor(vec))])}

sv = svd(scale(gdat,center=TRUE,scale=FALSE))
V = sv$v
Z = gdat%*%V

K = 3
pclabs = c("PC1","PC2","PC3","PC4")
par(mfrow=c(1,K))
for(i in 1:K){
  j = i+1
  plot(Z[,i],Z[,j],pch=16,xlab=pclabs[i],ylab=pclabs[j],col=Cols(cdat$Subtype))
}
legend(-45,0,pch=16,col=rainbow(5),levels(cdat$Subtype))
```




Pairs Plot

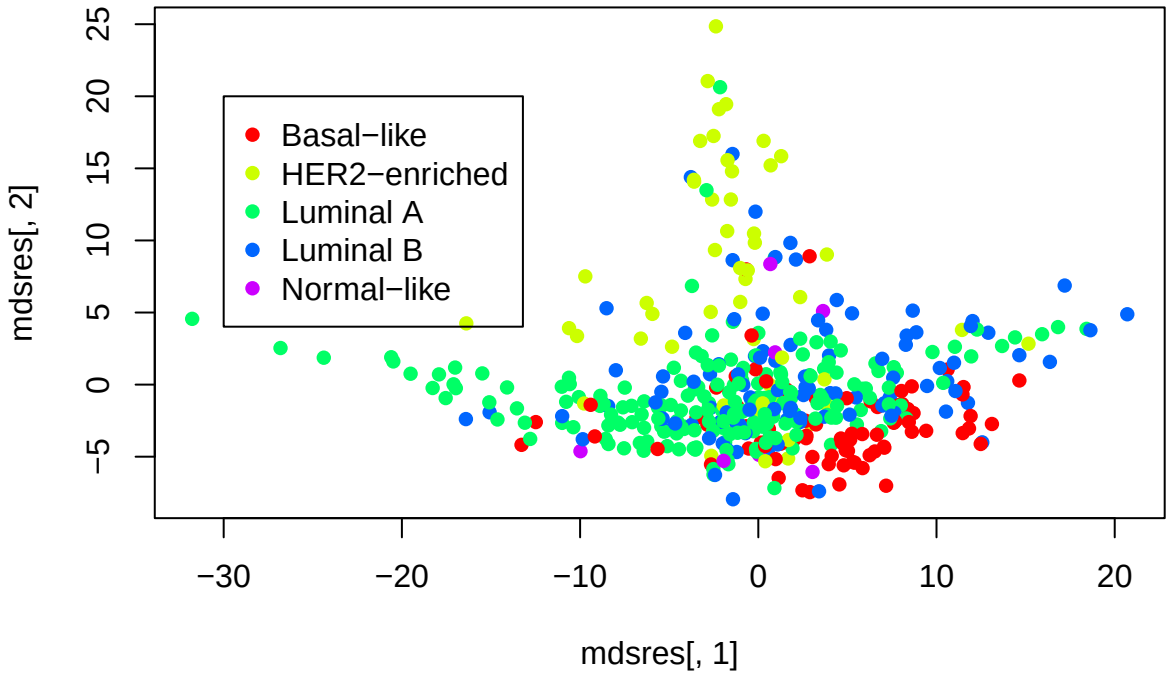
```
PC1<-as.matrix(Z[,1])
PC2<-as.matrix(Z[,2])
PC3<-as.matrix(Z[,3])
PC4<-as.matrix(Z[,4])
PC5<-as.matrix(Z[,5])

pc.df.cdat<-data.frame(Subtype = cdat$Subtype, PC1, PC2, PC3, PC4, PC5)
ggpairs(pc.df.cdat, mapping = aes(color = Subtype))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




MDS

```
Dmat = dist(gdat,method="maximum")
mdsres = cmdscale(Dmat,k=2)
plot(mdsres[,1],mdsres[,2],pch=16,col=Cols(cdat$Subtype), main = "Dimension Reduction MDS")
legend(-30,20,pch=16,col=rainbow(5),levels(cdat$Subtype))
```

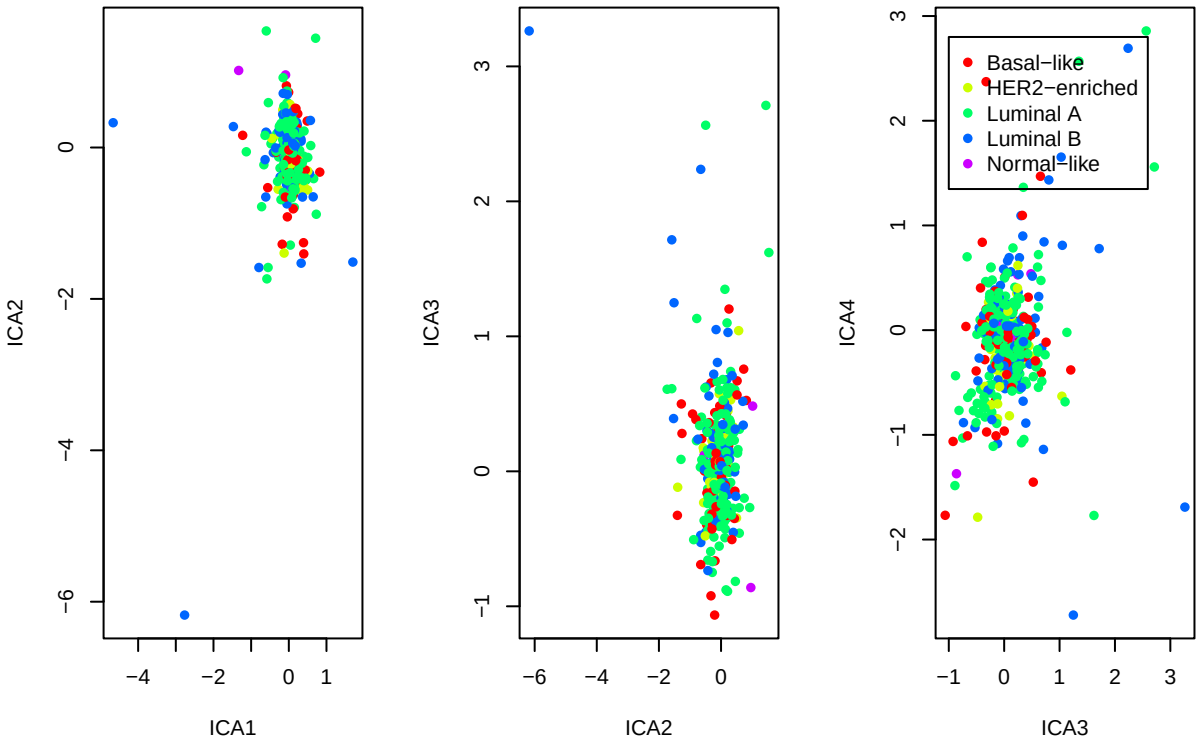



ICA

```
require("fastICA")
```

```
## Loading required package: fastICA
```

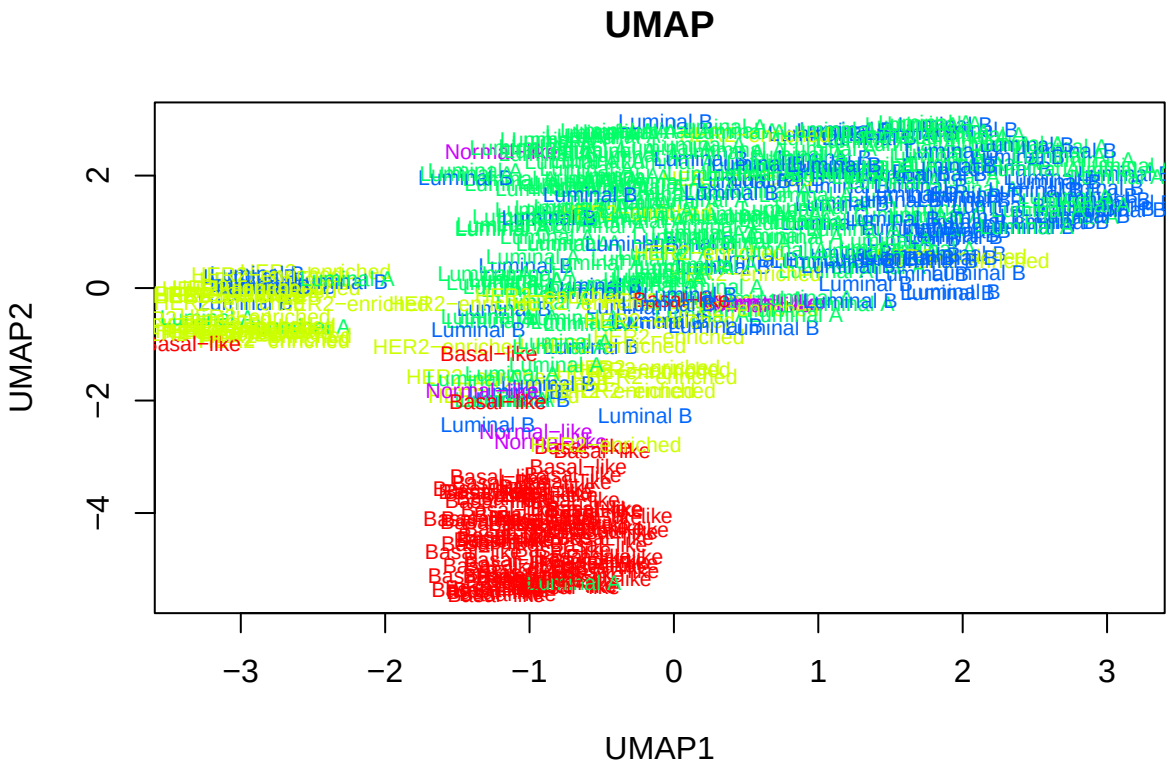
```
K = 4
icafit = fastICA(gdat,n.comp=K)

kk = 3
pclabs = c("ICA1","ICA2","ICA3","ICA4")
par(mfrow=c(1,kk))
for(i in 1:kk){
  j = i+1
  plot(icafit$A[i,],icafit$A[j,],pch=16,xlab=pclabs[i],ylab=pclabs[j],col=Cols(cdat$Subtype))
}
legend(-1,2.8,pch=16,col=rainbow(5),levels(cdat$Subtype))
```




UMAP

```
gdat.umap = umap(gdat)
plot(gdat.umap$layout[,1], y =gdat.umap$layout[,2], type = "n", main = "UMAP", xlab = "UMAP1", ylab = "U
text(gdat.umap$layout[,1], y =gdat.umap$layout[,2], type = "n", cdat$Subtype, col=Cols(cdat$Subtype), co
```

```
## Warning in text.default(gdat.umap$layout[, 1], y = gdat.umap$layout[, 2], :
## graphical parameter "type" is obsolete
```




## Clustering

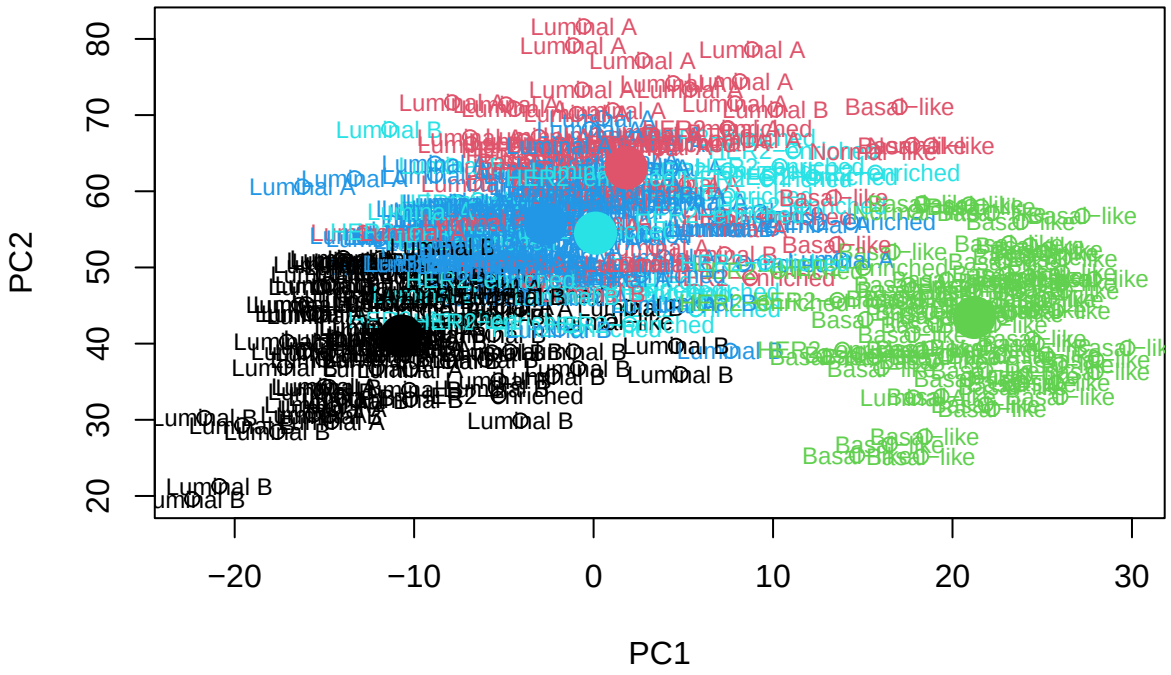
K-means

```
K = 5
km = kmeans(gdat,centers=K,nstart=25)
table(km$cluster,cdat$Subtype)
```

```
##
##     Basal-like HER2-enriched Luminal A Luminal B Normal-like
##   1          0             2        40        59           1
##   2          5             8        45        13           4
##   3         74             5         1         1           1
##   4          0             7       111        26           1
##   5          0            31         3         7           0
```

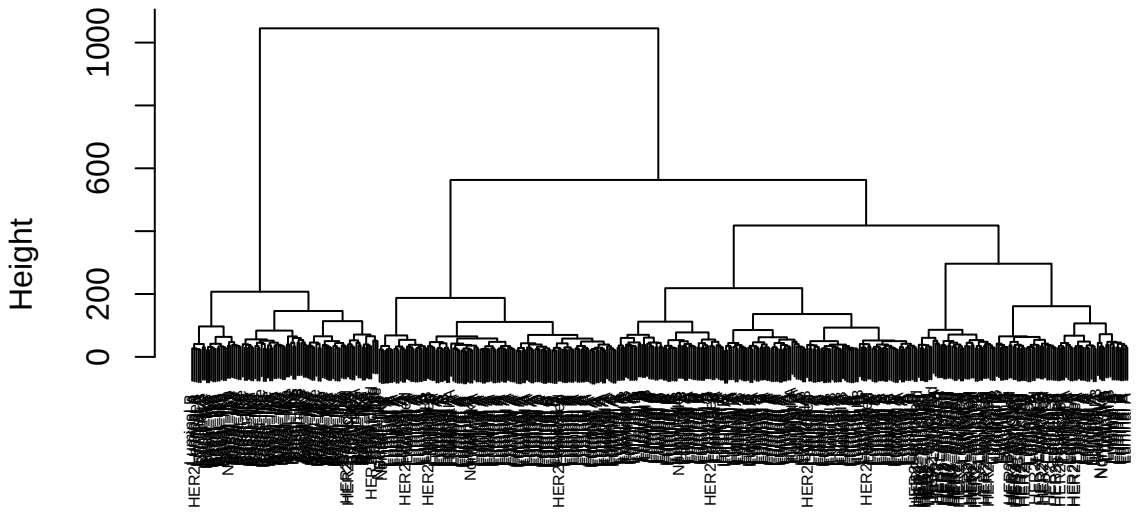
Plot Kmeans with labels

```
plot(Z[,1],Z[,2],col=km$cluster, main = "Plot Kmeans Clusters ", xlab = "PC1", ylab = "PC2")
text(Z[,1],Z[,2],cdat$Subtype,cex=.75,col=km$cluster)
cens = km$centers
points(cens%*%V[,1],cens%*%V[,2],col=1:K,pch=16,cex=3)
```




Hierarchical

```
#which linakge is the best?
#which distance metric is the best?

Dmat = dist(gdat)
com.hc = hclust(Dmat,method="ward.D")


plot(com.hc,labels=cdat$Subtype,cex=.5)
```




```
res.com = cutree(com.hc,5)
table(res.com,cdat$Subtype)
```

```
##
## res.com Basal-like HER2-enriched Luminal A Luminal B Normal-like
##       1          1             3        95        11           3
##       2          0             4        73        65           1
##       3         75             4         5         4           1
##       4          0            27         3         7           0
##       5          3            15        24        19           2
```

Consensus Clustering with Hierarchical

```
#Note that ConsensusClusterPlus not available for R version 4.0.2
#results = ConsensusClusterPlus(gdat,maxK=6,reps=500,pItem=0.8,pFeature=1,
#clusterAlg="hc",distance="pearson",plot="png")
```

Look at results for first 5 clusters

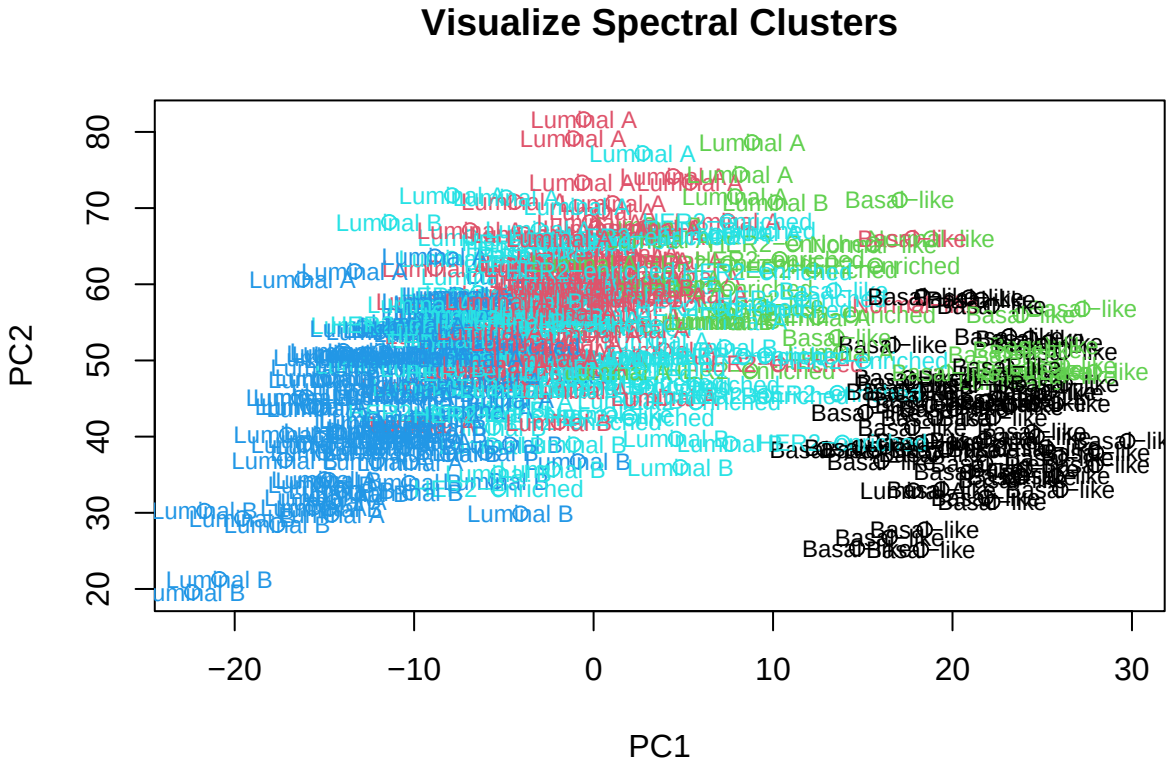
```
 #heatmap(results[[2]][["consensusMatrix"]][1:5,1:5])
```

Spectral Clustering

```
K = 5
s_gdat = specClust(gdat, centers=K, nn = 7, method = "symmetric", gmax=NULL)
```

Visualize

```
plot(Z[,1],Z[,2],col=s_gdat$cluster, main = "Visualize Spectral Clusters", xlab = "PC1", ylab = "PC2")
text(Z[,1],Z[,2],cdat$Subtype,cex=.75,col=s_gdat$cluster)
```




### Genes significantly associated with ER or PR Status, etc

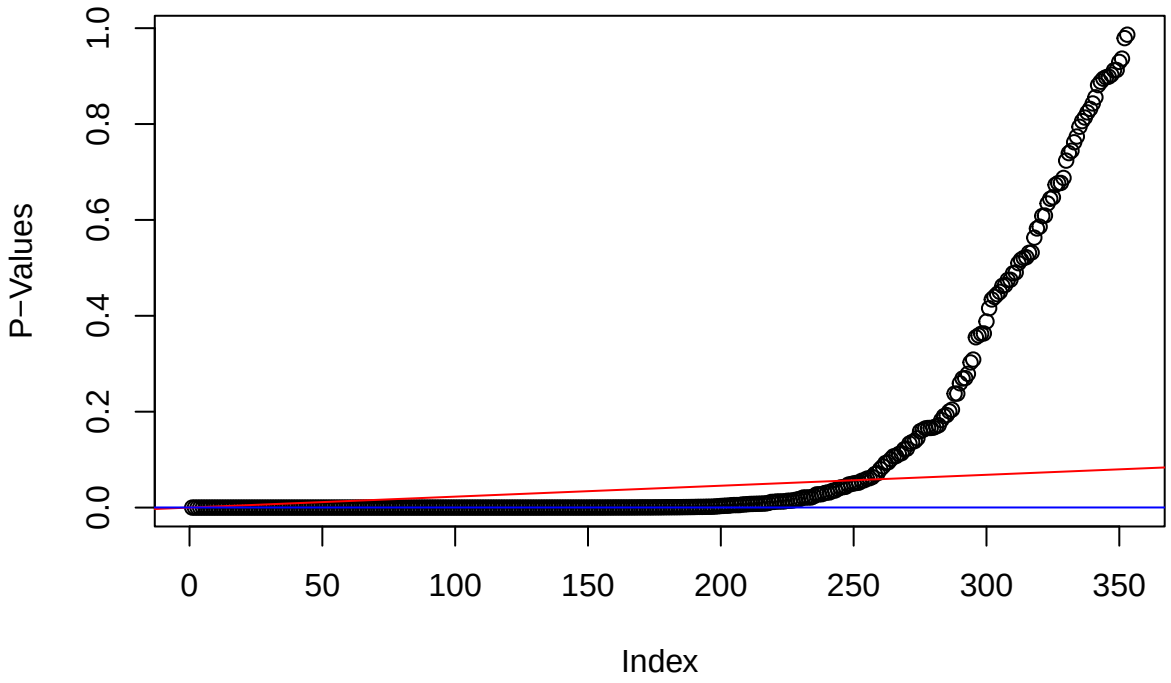
```
x = gdat[cdat$ER=="Positive" | cdat$ER=="Negative",]
y.er = cdat$ER[cdat$ER=="Positive" | cdat$ER=="Negative"]
y.label = rep(1, length(y.er))
y.label[y.er == "Positive"]=2

ps = NULL
for(i in 1:ncol(gdat)) ps = c(ps,
 t.test(x[y.label==1,i],x[y.label==2,i])$p.value)
fdrs.bh = p.adjust(ps, method="BH")

cat("Number of Tests significant with alpha=0.1 using Bonferroni correction:",
sum(ps<0.1/length(y.label)), fill=TRUE)
```

```
## Number of Tests significant with alpha=0.1 using Bonferroni correction: 165
```

```
cat("Number of Tests with FDR below 0.1:",
sum(fdrs.bh<0.1), fill=TRUE)
```

```
## Number of Tests with FDR below 0.1: 259
```

```
plot(sort(ps,decreasing=FALSE),ylab="P-Values")
#BH procedure
abline(a=0, b=0.1/length(y.label),col="red")
#Bonferroni
abline(a=0.1/length(y.label), b=0,col="blue")
```

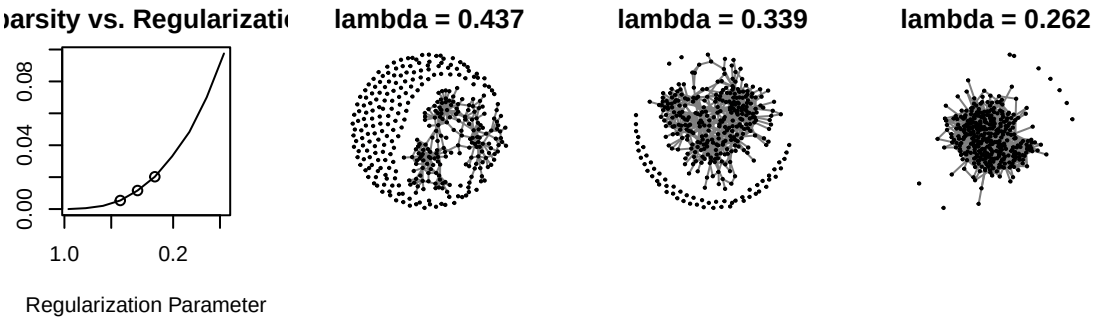



## Graphical models - how are genes related?

```
# use huge package
neth = huge(gdat,method="mb")
```

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

```
plot(neth)
```



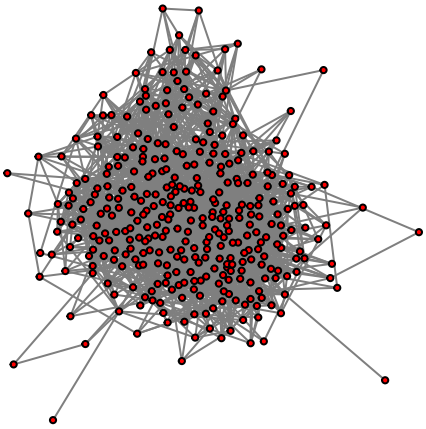

```
## stability selection with huge
net.s <- huge.select(neth, criterion="stars")
```

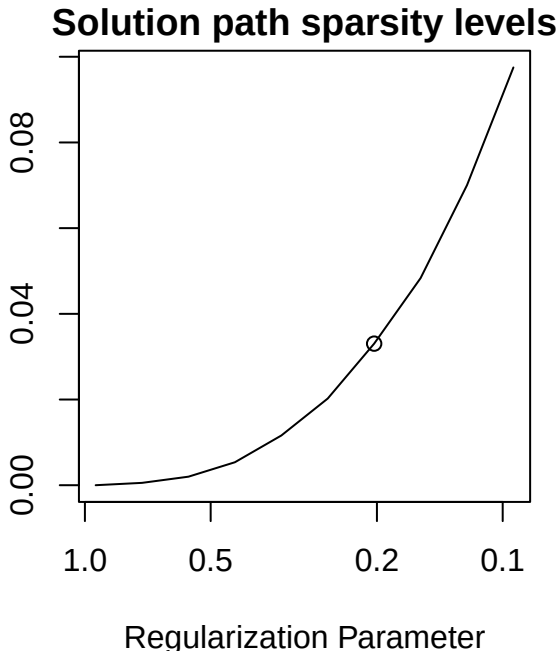
```
## Conducting Subsampling....in progress:5% Conducting Subsampling....in progress:10% Conducting Subsamp
```

```
net.s
```

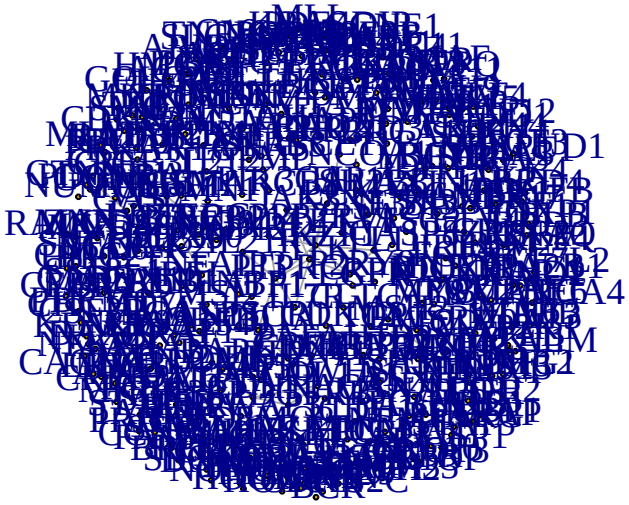
```
## Model: Meinshausen & Buhlmann Graph Estimation (mb)
## selection criterion: stars
## Graph dimension: 353
## sparsity level 0.03304468
```

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

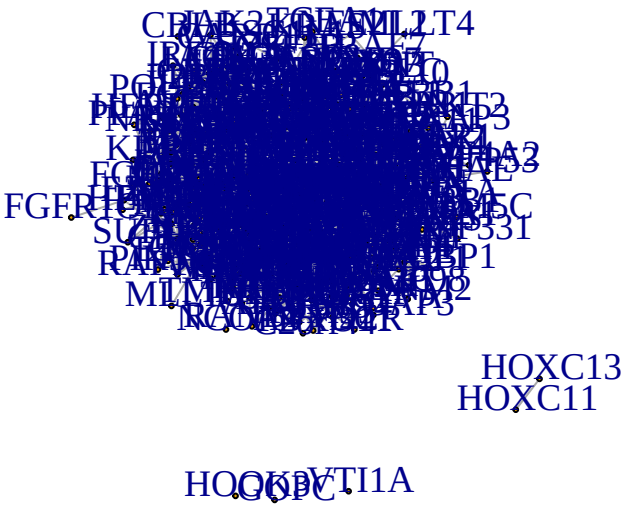





```
#larger lambda
mat <- neth$path[[2]]
neti <- as.undirected(graph_from_adjacency_matrix(mat))
plot(neti,vertex.label=colnames(gdat),vertex.size=2,vertex.label.cex=1.2,vertex.label.dist=1,layout=lay
```



```
#smaller lambda
mat = neth$path[[6]]
neti = as.undirected(graph_from_adjacency_matrix(mat))
plot(neti,vertex.label=colnames(gdat),vertex.size=2,vertex.label.cex=1.2,vertex.label.dist=1,layout=lay
```


FGFR1
HOXC13
HOXC11
VTI1A