Assignment3

2022-10-17

#2

library(magrittr)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:AnnotationDbi':  
##   
## select

## The following objects are masked from 'package:IRanges':  
##   
## collapse, desc, intersect, setdiff, slice, union

## The following objects are masked from 'package:S4Vectors':  
##   
## first, intersect, rename, setdiff, setequal, union

## The following object is masked from 'package:Biobase':  
##   
## combine

## The following objects are masked from 'package:BiocGenerics':  
##   
## combine, intersect, setdiff, union

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tibble)  
  
  
  
compute\_cv <- function(x) sd(x) / mean(x)  
cv <- data.frame(apply(filtered\_expression\_df,1 , compute\_cv))  
colnames(cv) <- (c("cv"))  
summary(cv)

## cv   
## Min. : 0.2934   
## 1st Qu.: 0.5832   
## Median : 0.8812   
## Mean : 1.2056   
## 3rd Qu.: 1.3807   
## Max. :11.3176

#https://jdblischak.github.io/singlecell-qtl/pca-variable.html  
  
orderedByCV <- data.frame(cv %>% arrange(desc(cv))) %>% rownames\_to\_column('gene')  
  
#(a)  
V5000 <- filtered\_expression\_df %>% filter(rownames(filtered\_expression\_df) %in% (orderedByCV[0:5000,1]))#5000 most variable genes  
V5000 <- t(V5000)

#(b)  
  
##kmeans clustering  
k2 <- kmeans(V5000, centers = 2)  
k3 <- kmeans(V5000, centers = 3)  
str(kmeans)

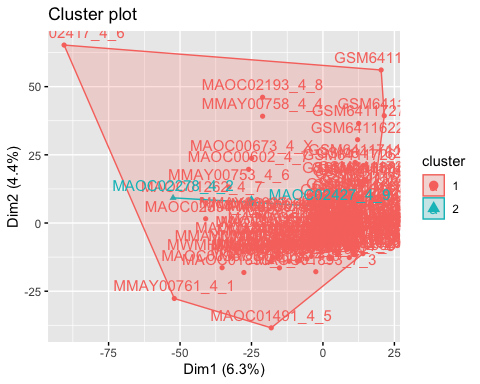
## function (x, centers, iter.max = 10L, nstart = 1L, algorithm = c("Hartigan-Wong",   
## "Lloyd", "Forgy", "MacQueen"), trace = FALSE)

#https://uc-r.github.io/kmeans\_clustering#prep  
k4 <- kmeans(V5000, centers = 4)  
  
library(factoextra)

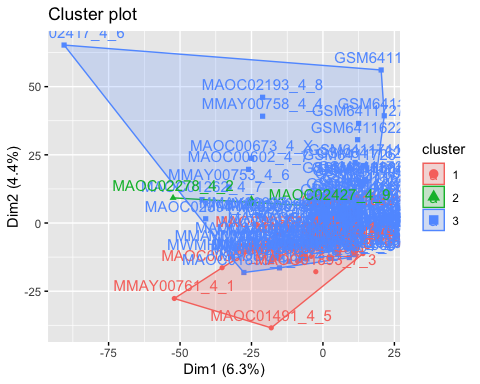
## Loading required package: ggplot2

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

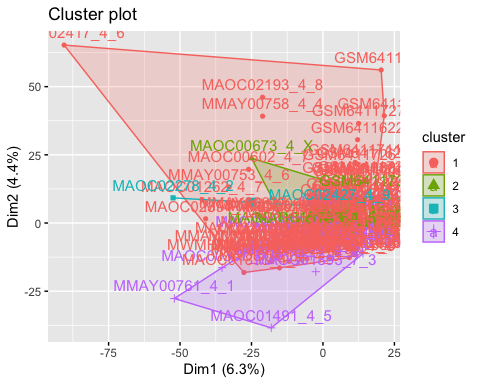
fviz\_cluster(k2, V5000)



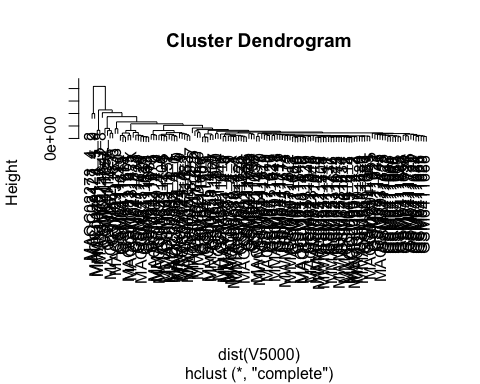
fviz\_cluster(k3, V5000)



fviz\_cluster(k4, V5000)



##hierarchical clustering  
library(stats)  
  
hclust <- hclust(dist(V5000), method = "complete", members = NULL)  
plot(hclust)



##consensus clustering   
BiocManager::install("ConsensusClusterPlus")

## 'getOption("repos")' replaces Bioconductor standard repositories, see  
## '?repositories' for details  
##   
## replacement repositories:  
## CRAN: http://cran.rstudio.com/

## Bioconductor version 3.15 (BiocManager 1.30.18), R 4.2.1 (2022-06-23)

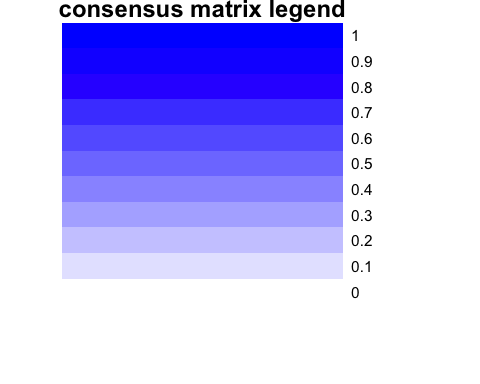
## Warning: package(s) not installed when version(s) same as current; use `force = TRUE` to  
## re-install: 'ConsensusClusterPlus'

## Old packages: 'BiocManager', 'BiocParallel', 'car', 'commonmark', 'cpp11',  
## 'data.table', 'DT', 'fontawesome', 'jsonlite', 'maptools', 'markdown',  
## 'mgcv', 'minqa', 'nlme', 'openssl', 'processx', 'ragg', 'rpart', 'shiny',  
## 'sys', 'vctrs', 'xfun', 'yaml', 'zip'

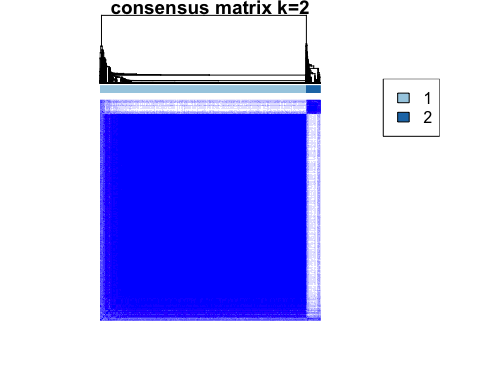
library(ConsensusClusterPlus)  
  
v5000matrix <-data.matrix(V5000, rownames.force = NA)  
results <- ConsensusClusterPlus(v5000matrix,maxK=4,reps=50,pItem=0.8,pFeature=1,title='title',clusterAlg="hc",distance="pearson",plot="screen")

## end fraction

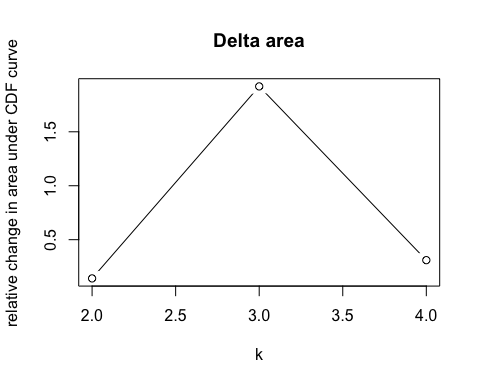
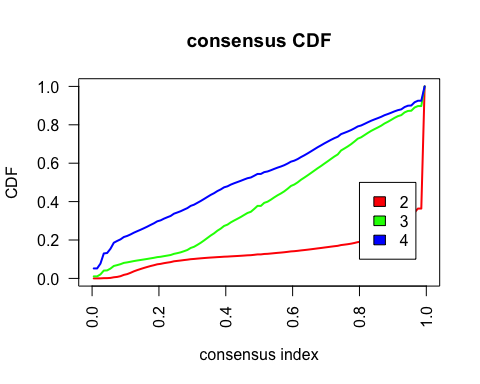
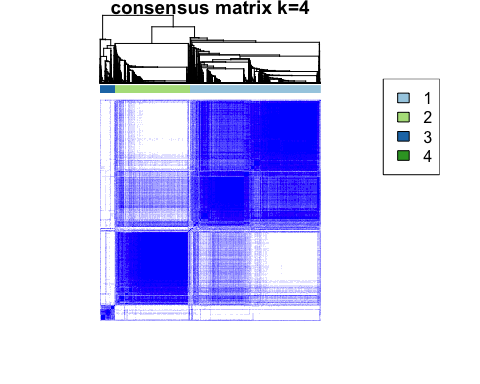
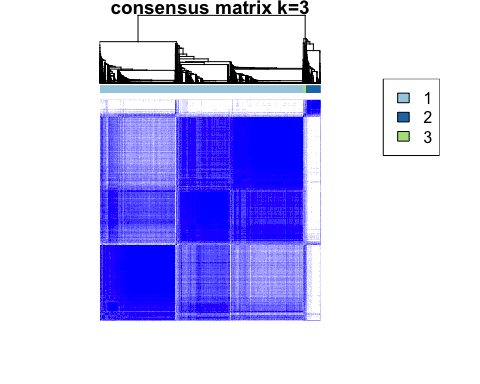
## clustered



## clustered



## clustered

 ##d Using kmeans and an initial cluster number of 2, one small cluster is nested within a cluster that was much larger in terms of area and membership. With k=3, the small nested cluster persists, but the larger cluster is split into two. As in the previous case, there is a cluster with a membership that is much larger than the other clusters. The same pattern can be oberved for k=4.

The hierarchical clustering function doesn’t appear to have chosen a k or been determined by any input for k. Instead it llooks like the clustering resolved down to each individual sample while providing a metric for cluster proximity with its height axis. In this case, the three or four primary clusters still maintain an extreme inequality in terms of sample membership, similar to our kmeans clustering results.

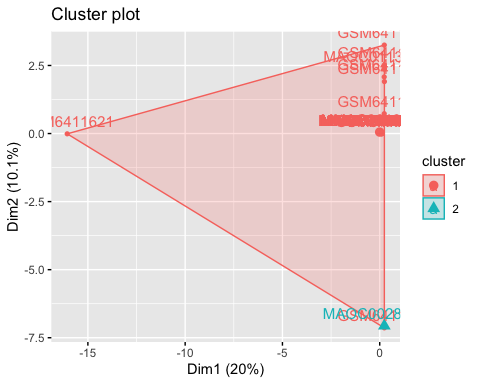
Using consensus clustering, our delta area graph no appreciable difference in consensus beyond k-3. The consensus clustering graph shows a majority of membership in one group with a smaller group and an even smaller group next to that. With an increase to a cluster number of 4, the appearance of a larger group emerges. This group is possibly analogous to the nested group that appeared in the k=4 kmeans clustering.

In each of the methods, the proportion of cluster membership does not seem to reflect that of the control groups for our experiment, which is roughly half and half.

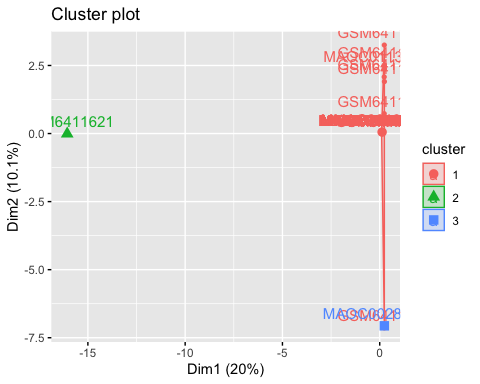
##(e) rerun each clustering method for 10, 100, 1000 and 10000 genes

V10 <- filtered\_expression\_df %>% filter(rownames(filtered\_expression\_df) %in% (orderedByCV[0:10,1]))  
V10 <- t(V10)  
V100 <- filtered\_expression\_df %>% filter(rownames(filtered\_expression\_df) %in% (orderedByCV[0:100,1]))  
V100 <- t(V100)  
V1000 <- filtered\_expression\_df %>% filter(rownames(filtered\_expression\_df) %in% (orderedByCV[0:1000,1]))  
V1000 <- t(V1000)  
V10000 <- filtered\_expression\_df %>% filter(rownames(filtered\_expression\_df) %in% (orderedByCV[0:10000,1]))  
V10000 <- t(V10000)

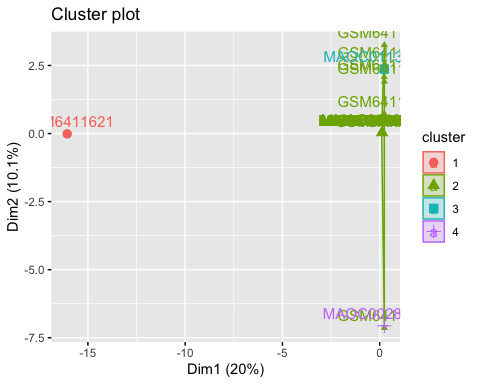
##10  
k2\_10 <- kmeans(V10, centers = 2)  
k3\_10 <- kmeans(V10, centers = 3)  
k4\_10 <- kmeans(V10, centers = 4)  
  
fviz\_cluster(k2\_10, V10)



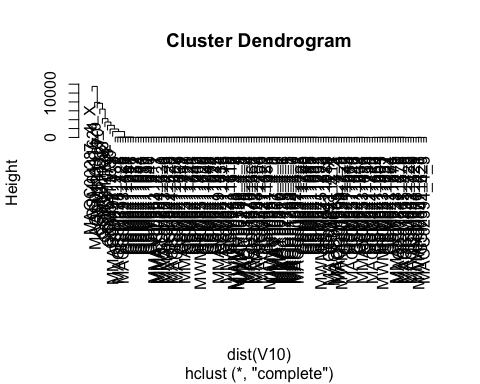
fviz\_cluster(k3\_10, V10)



fviz\_cluster(k4\_10, V10)



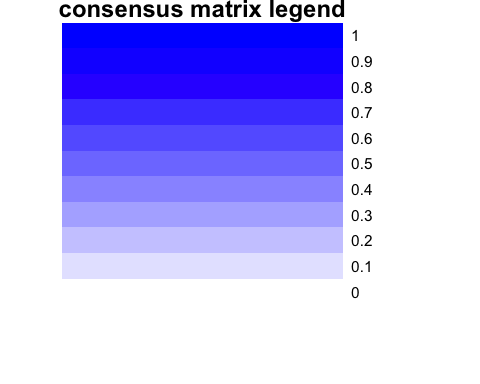
hclust10 <- hclust(dist(V10), method = "complete", members = NULL)  
plot(hclust10)



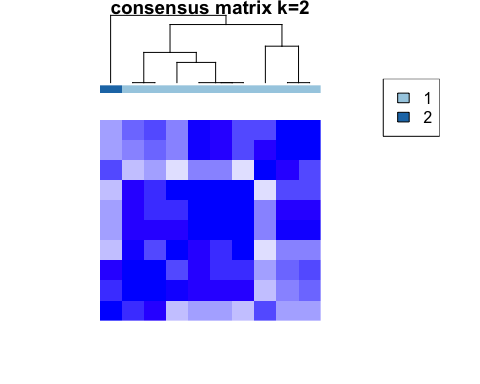
v10matrix <-data.matrix(V10, rownames.force = NA)  
results10 <- ConsensusClusterPlus(v10matrix,maxK=4,reps=50,pItem=0.8,pFeature=1,title='title',clusterAlg="hc",distance="pearson",plot="screen")

## end fraction

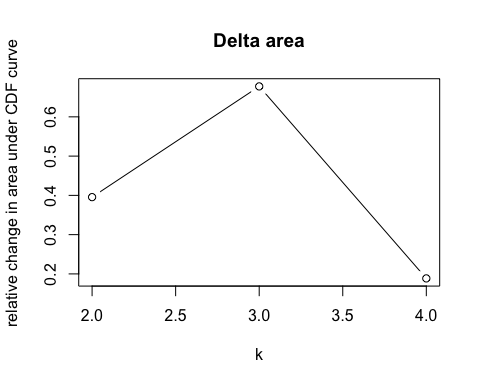
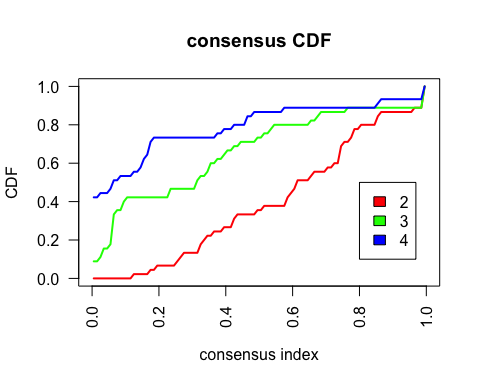
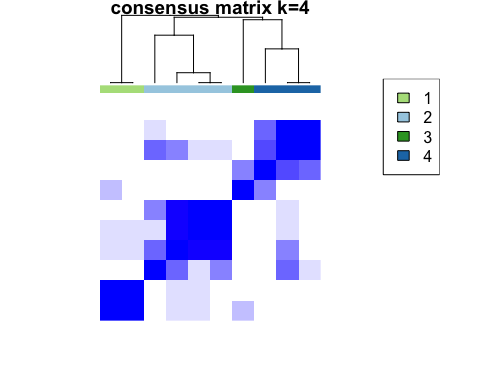
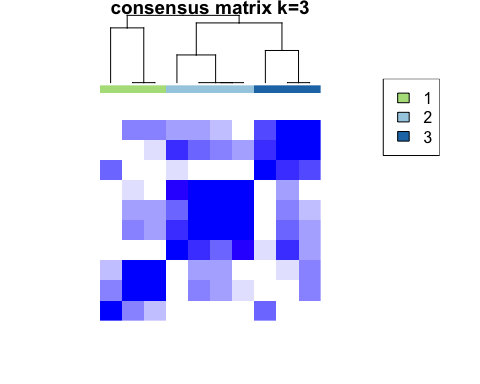
## clustered



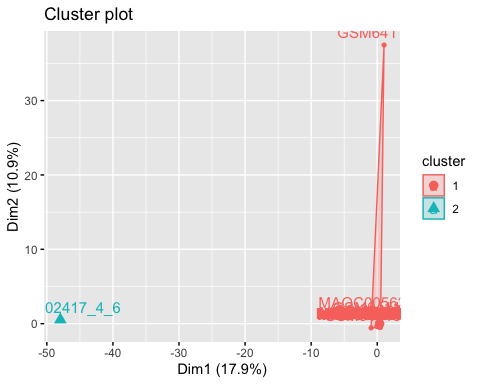
## clustered



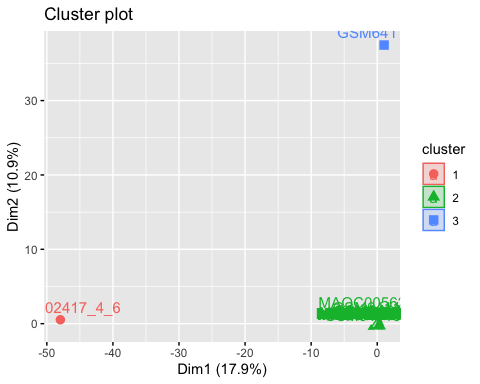
## clustered



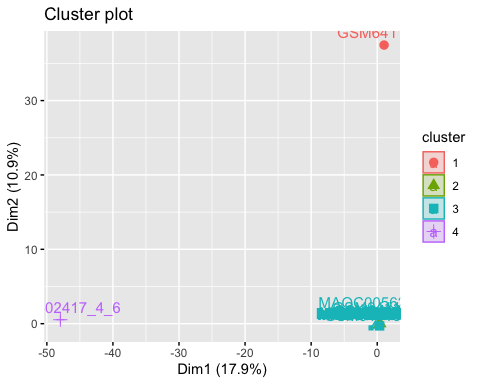
##100  
k2\_100 <- kmeans(V100, centers = 2)  
k3\_100 <- kmeans(V100, centers = 3)  
k4\_100 <- kmeans(V100, centers = 4)  
  
fviz\_cluster(k2\_100, V100)



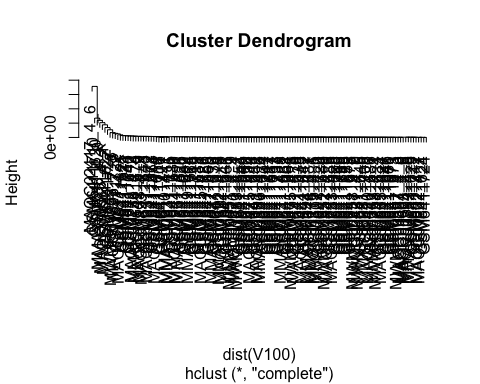
fviz\_cluster(k3\_100, V100)



fviz\_cluster(k4\_100, V100)



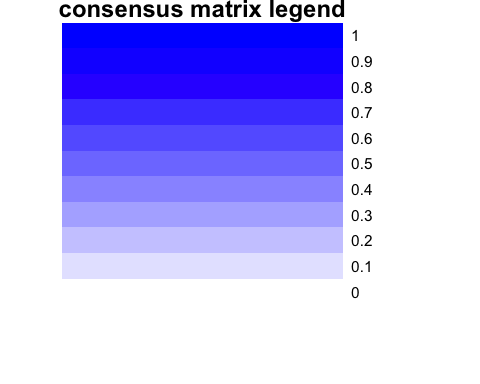
hclust100 <- hclust(dist(V100), method = "complete", members = NULL)  
plot(hclust100)



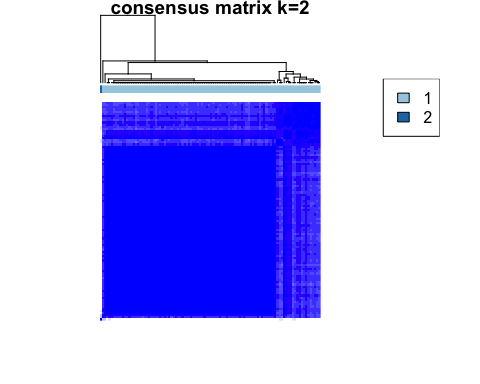
v100matrix <-data.matrix(V100, rownames.force = NA)  
results100 <- ConsensusClusterPlus(v100matrix,maxK=4,reps=50,pItem=0.8,pFeature=1,title='title',clusterAlg="hc",distance="pearson",plot="screen")

## end fraction

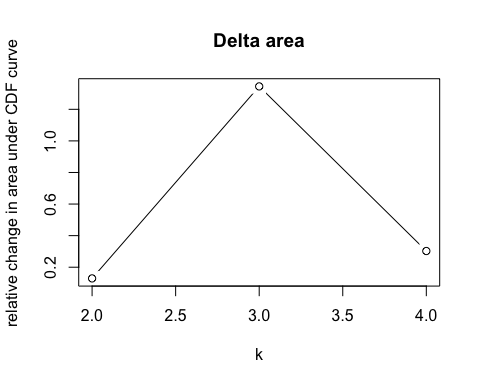
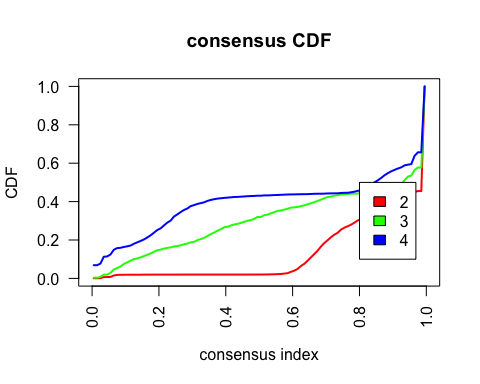
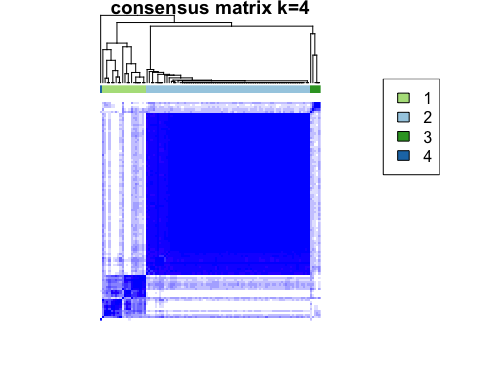
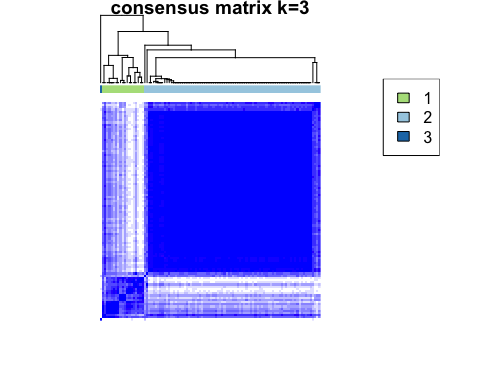
## clustered



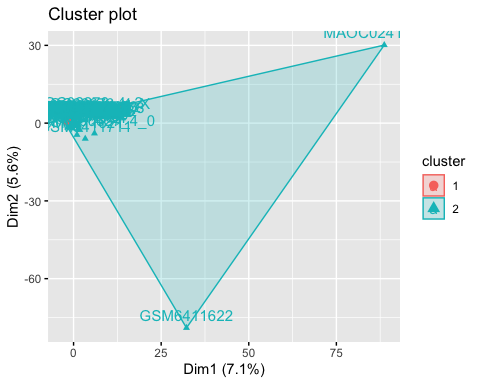
## clustered



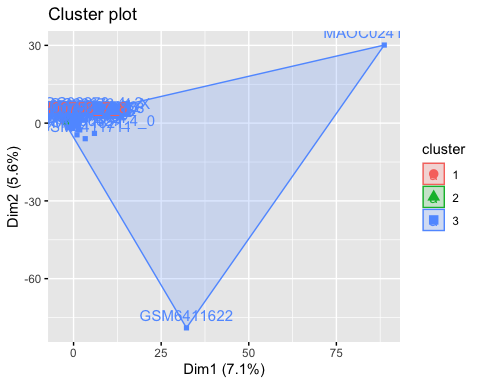
## clustered



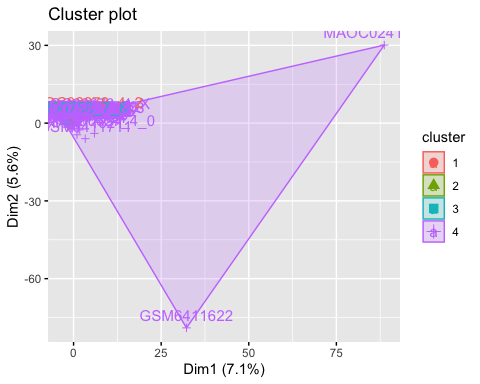
#1000  
k2\_1000 <- kmeans(V1000, centers = 2)  
k3\_1000 <- kmeans(V1000, centers = 3)  
k4\_1000 <- kmeans(V1000, centers = 4)  
  
fviz\_cluster(k2\_1000, V1000)



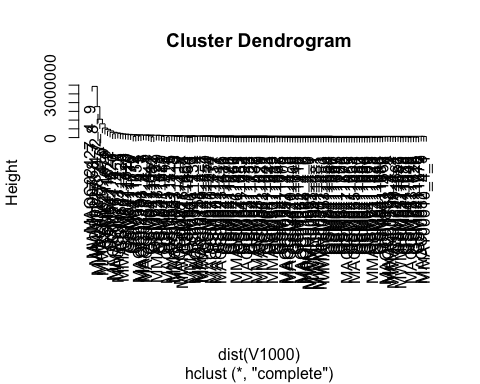
fviz\_cluster(k3\_1000, V1000)



fviz\_cluster(k4\_1000, V1000)



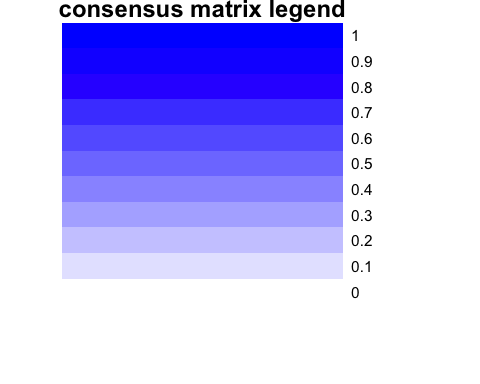
hclust1000 <- hclust(dist(V1000), method = "complete", members = NULL)  
plot(hclust1000)



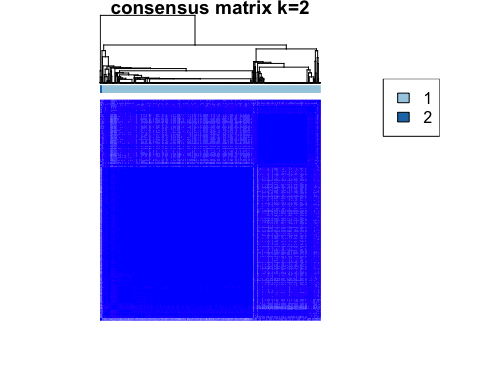
v1000matrix <-data.matrix(V1000, rownames.force = NA)  
results1000 <- ConsensusClusterPlus(v1000matrix,maxK=4,reps=50,pItem=0.8,pFeature=1,title='title',clusterAlg="hc",distance="pearson",plot="screen")

## end fraction

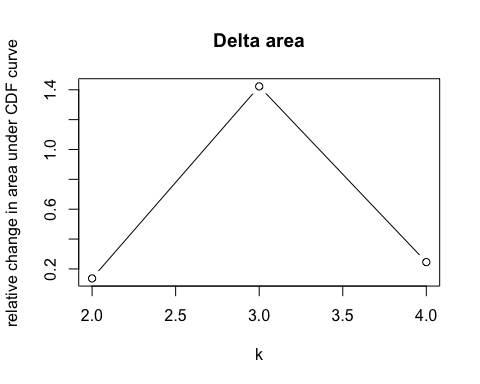
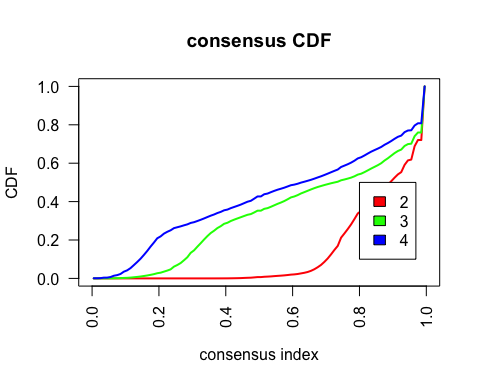
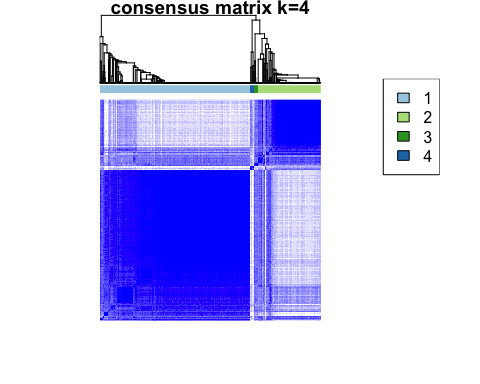
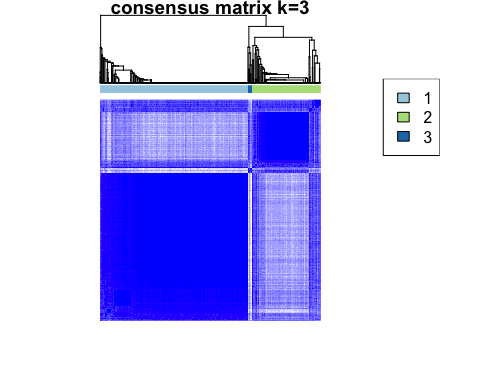
## clustered



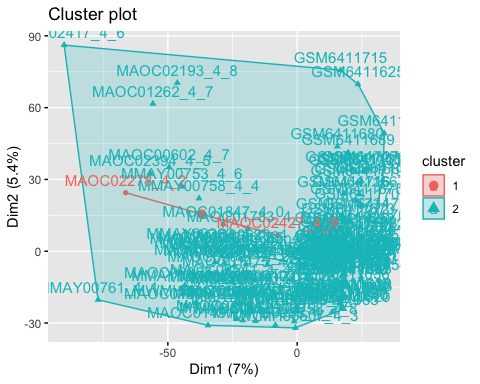
## clustered



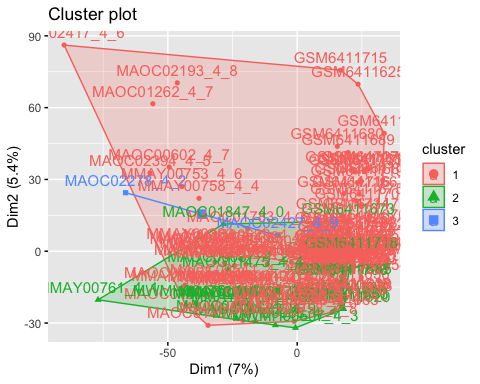
## clustered



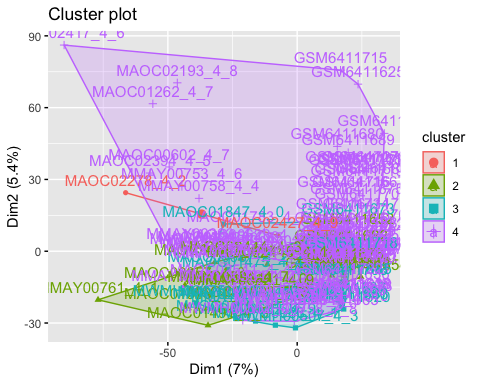
#10000  
k2\_10000 <- kmeans(V10000, centers = 2)  
k3\_10000 <- kmeans(V10000, centers = 3)  
k4\_10000 <- kmeans(V10000, centers = 4)  
  
fviz\_cluster(k2\_10000, V10000)



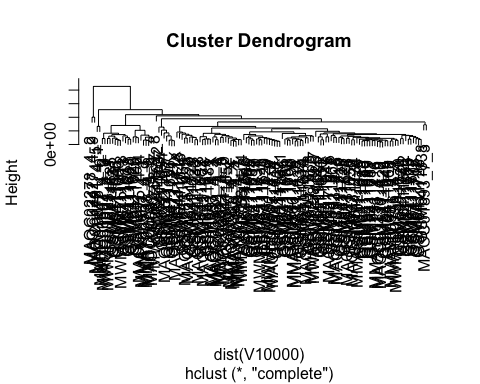
fviz\_cluster(k3\_10000, V10000)



fviz\_cluster(k4\_10000, V10000)



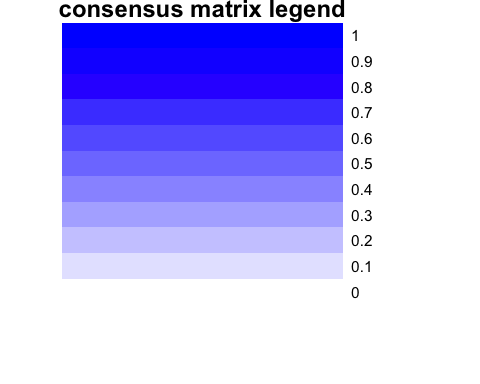
hclust10000 <- hclust(dist(V10000), method = "complete", members = NULL)  
plot(hclust10000)



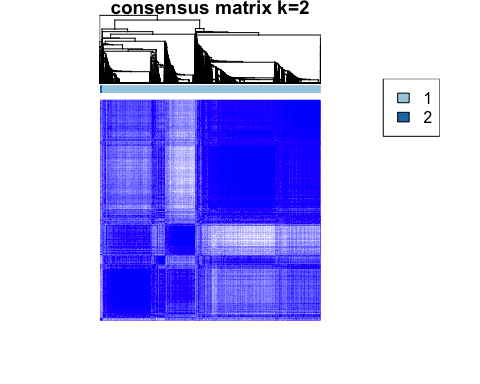
v10000matrix <-data.matrix(V10000, rownames.force = NA)  
results10000 <- ConsensusClusterPlus(v10000matrix,maxK=4,reps=50,pItem=0.8,pFeature=1,title='title',clusterAlg="hc",distance="pearson",plot="screen")

## end fraction

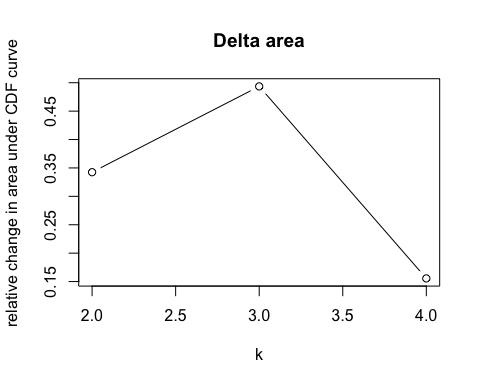
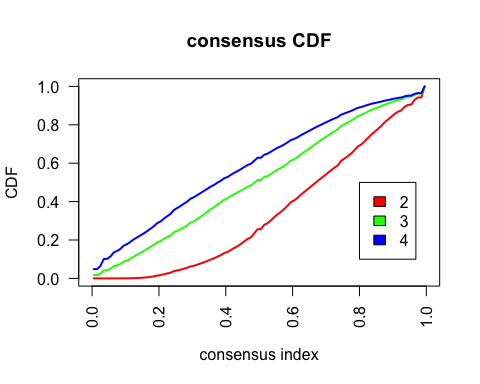
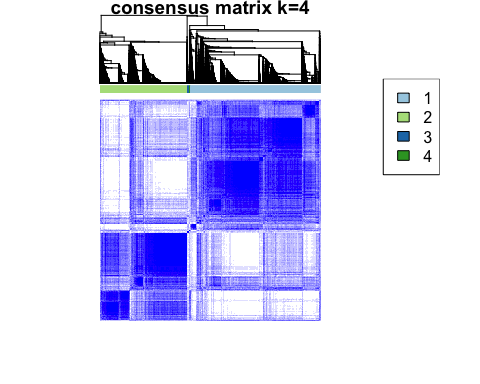
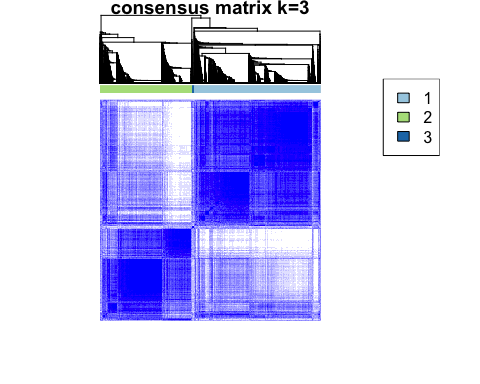
## clustered



## clustered



## clustered

 #3 Heatmaps and Dendrograms

library(devtools)

## Loading required package: usethis

install\_github("jokergoo/ComplexHeatmap")

## Skipping install of 'ComplexHeatmap' from a github remote, the SHA1 (2d6e2705) has not changed since last install.  
## Use `force = TRUE` to force installation

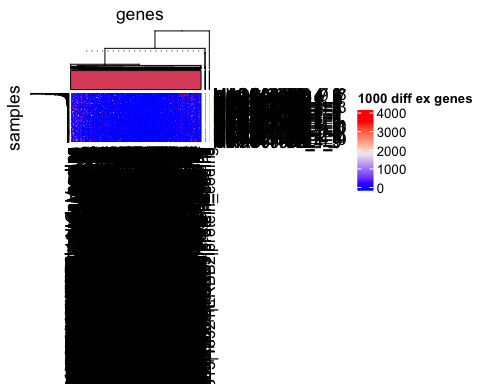
library(ComplexHeatmap)

## Loading required package: grid

## ========================================  
## ComplexHeatmap version 2.13.3  
## Bioconductor page: http://bioconductor.org/packages/ComplexHeatmap/  
## Github page: https://github.com/jokergoo/ComplexHeatmap  
## Documentation: http://jokergoo.github.io/ComplexHeatmap-reference  
##   
## If you use it in published research, please cite either one:  
## - Gu, Z. Complex Heatmap Visualization. iMeta 2022.  
## - Gu, Z. Complex heatmaps reveal patterns and correlations in multidimensional   
## genomic data. Bioinformatics 2016.  
##   
##   
## The new InteractiveComplexHeatmap package can directly export static   
## complex heatmaps into an interactive Shiny app with zero effort. Have a try!  
##   
## This message can be suppressed by:  
## suppressPackageStartupMessages(library(ComplexHeatmap))  
## ========================================

library(cluster)  
Heatmap(V1000,top\_annotation = HeatmapAnnotation(foo = anno\_block(gp = gpar(fill = 2:4))),  
 column\_km = 3, name = "1000 diff ex genes", cluster\_rows = hclust1000, column\_title = "genes", row\_title = "samples")

## The automatically generated colors map from the 1^st and 99^th of the  
## values in the matrix. There are outliers in the matrix whose patterns  
## might be hidden by this color mapping. You can manually set the color  
## to `col` argument.  
##   
## Use `suppressMessages()` to turn off this message.



##breaks w/5000 genes, so we used 1000 instead