236523 - Introduction to Bioinformatics - Spring 21 - HW2

Alon Hacohen | 311587653 & Roy Bernea |316315829

5/27/2021

# Question 2

Following are answers for the questions under the “Question 2” Section - as instructed in the assignment:

### Q2.1

The mapped reads are mapped preferentially to exons since during the process of splicing the introns are being removed from the RNA.  
In the final product (the mRNA molecule) there are only exons, therefore the read are mapped to the exons.

### Q2.2

A biological reason for reads being mapped to the intronic region during the RNA-seq process could be *Alternative Splicing*. In case alternative splicing had occurred in the pre mRNA molecule, a known phenomenon is IR (intron retention) in which a sequence might be spliced out as an intron, or simply retained.  
This time, in the final product, there will also be introns, therefore the read could be mapped to them as well.

### Q2.3

*Q2.3.1*  
Additional names (aliases) for the gene *MARCH7* Are:  
Membrane Associated Ring-CH-Type Finger  
Axotrophin  
MARCH-VII  
RNF177  
Membrane-Associated Ring Finger (C3HC4) 7, E3 Ubiquitin Protein Ligase  
RING-Type E3 Ubiquitin Transferase MARCHF7  
Membrane-Associated RING Finger Protein 7  
Membrane-Associated RING-CH Protein VII  
E3 Ubiquitin-Protein Ligase MARCHF7  
RING Finger Protein 177  
MARCH7  
AXOT  
Membrane-Associated Ring Finger (C3HC4)  
RING-Type E3 Ubiquitin Transferase MARCH7  
E3 Ubiquitin-Protein Ligase MARCH7  
Membrane Associated Ring Finger 7  
EC 2.3.2.27  
EC 6.3.2  
MARCHF7  
AXO

These gene names were extracted from the following databases:  
HGNC - the HUGO Gene Nomenclature Committee database  
Ensembl genome database  
UniProt database  
NCBI’s Entrez molecular sequence database  
OMIM - Online Mendelian Inheritance in Man database

*Q2.3.2*  
If we paste the name of this gene manually into an Excel sheet, meaning we paste “MARCH7” into a cell inside the sheet, a possible problem we might encounter is that Excel would recognize the name as a date - March 7 - and refer to it as one.  
In such case, instead of referring to the gene’s name “MARCH7” as its ID (or as a unique identifier), Excel would mistakenly refer to it as a date - which might cause us problem later on in our data analysis process (e.g. if we want to compare the gene in our excel sheet to its appearance in a database - but R would recognize the identifiers as two different ones because of their type etc.).

# Question 3

Following are the code and answers for the questions under the “Question 3” Section - as instructed in the assignment:

### Q3.1

Before we begin to work, we upload the neccesarry libraries:

library(ggplot2)  
library(RColorBrewer)  
library(factoextra)  
library(tidyverse)  
library(Rtsne)  
library(dbscan)  
library(fossil)

We begin by preparing and cleaning the data in order to use it correctly - meaning we load the data from the file, remove NA rows and scale the data. We also use dist() function in order to calculate the distances between each sample and create a distance matrix we will use later on.

# Data Preperation  
#change to relative path?  
sdy <- readRDS("C:/CS\_Technion/236523-Intro\_to\_Bioinformatics/Spr21/Intro\_to\_BioInf-Spr21-HW2/sdy420.rds")  
data <- na.omit(t(sdy$expr))  
data.scaled <- scale(data)  
data.dist <- dist(data.scaled, method="euclidean")  
  
# preparing tsne for later  
tsne = Rtsne(data.scaled)

Now, we move on to do the real work - and cluster our cleaned data in 3 different techniques (with arbitrary parameters): K-Means, Hierarchical clustering and DBSCAN.

set.seed(555)  
  
# K-Means  
km.result <- kmeans(data.scaled, centers = 3, nstart = 25)  
# hClust  
hc.result <- hclust(data.dist, "ward.D")  
# DBSCAN  
dbscan.result <- dbscan(tsne$Y, eps = 1, minPts = 4)

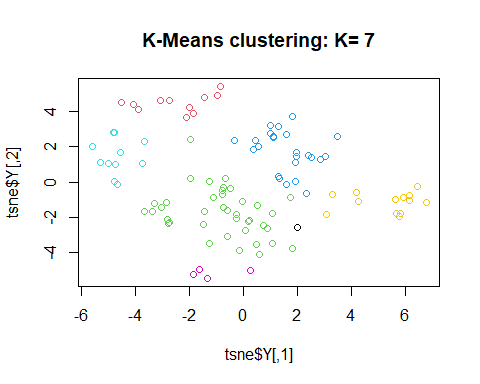
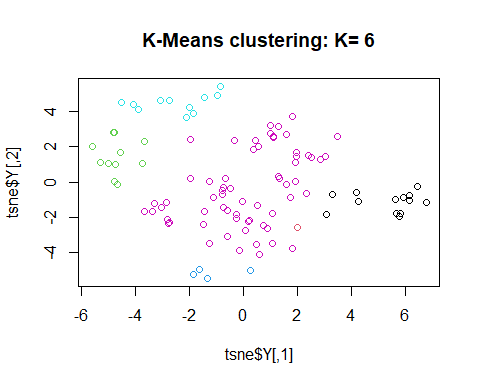
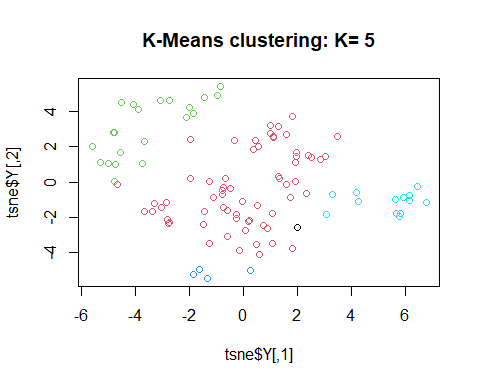
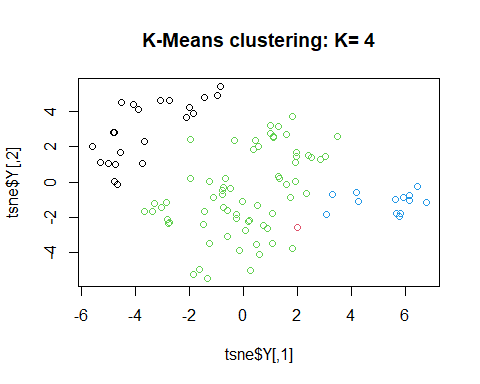
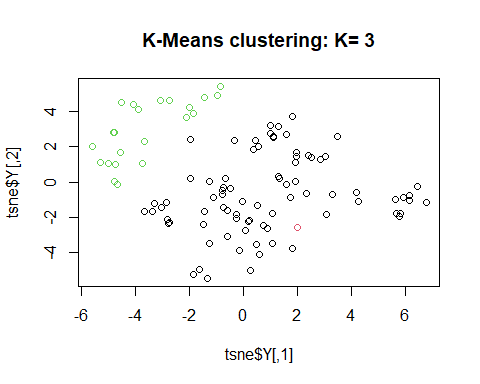
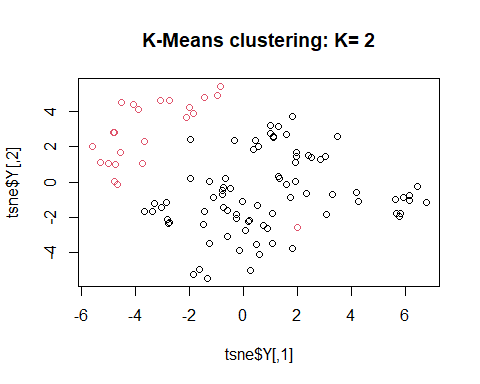
### Q3.2

You may see below our efforts to find the best parameters for each of the clustering methods. That includes parameters tuning (using different parameters for each method), and plotting our results.

Note: as opposed to different methods to set our parameters we found along the way, we decided that we would like to try the exploratory approach - meaning we tried to find the best parameters for each clustering method by looking at our data mainly visually.

*K-Means*

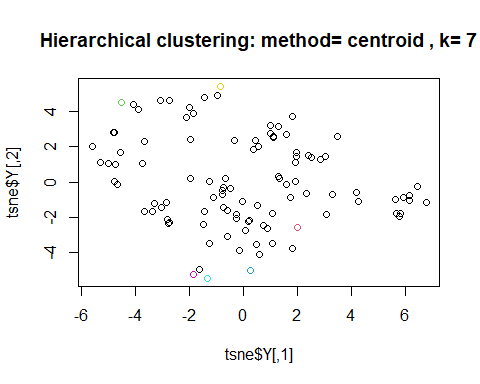
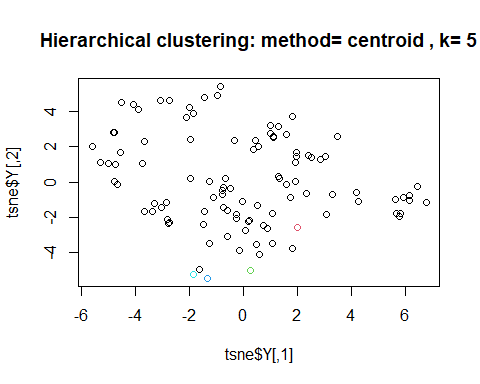
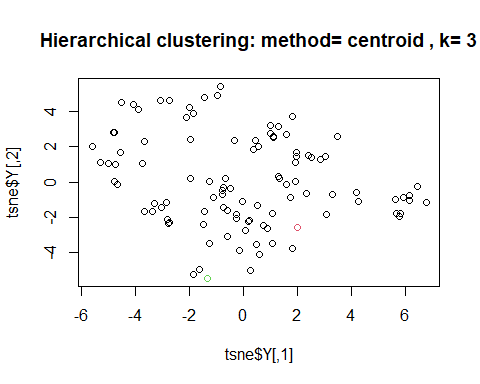
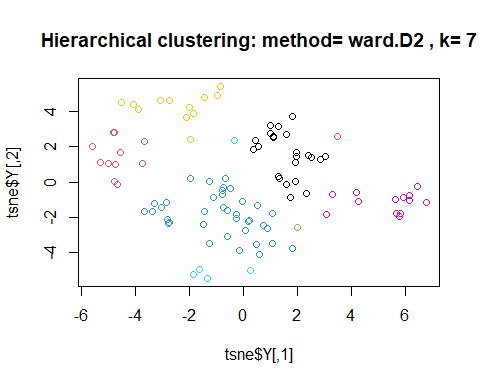
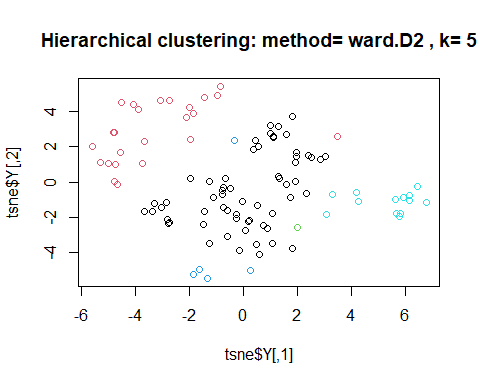
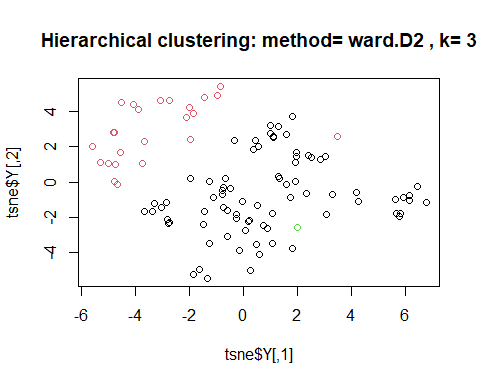
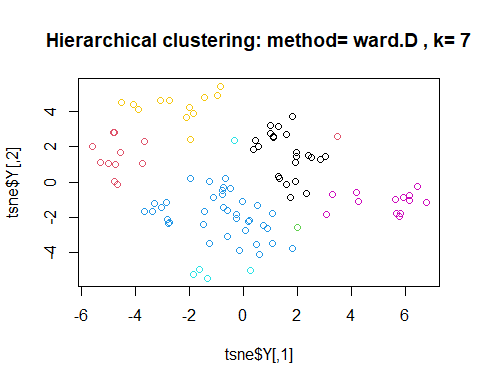
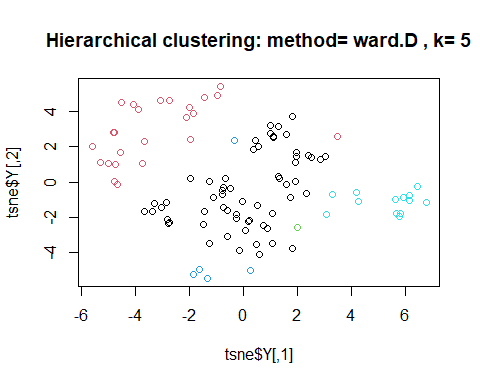
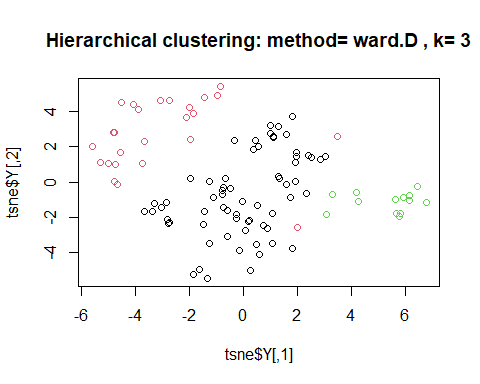
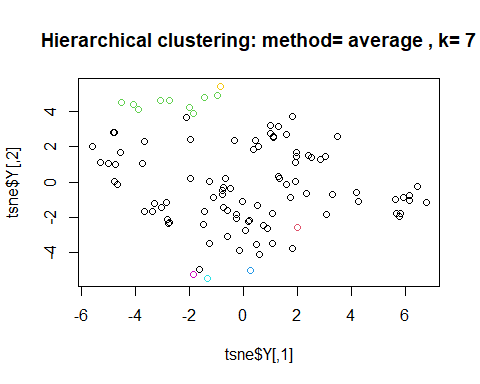
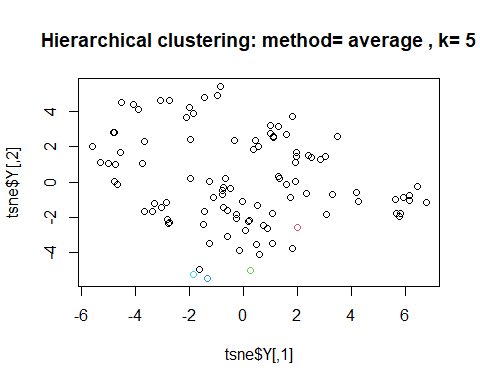
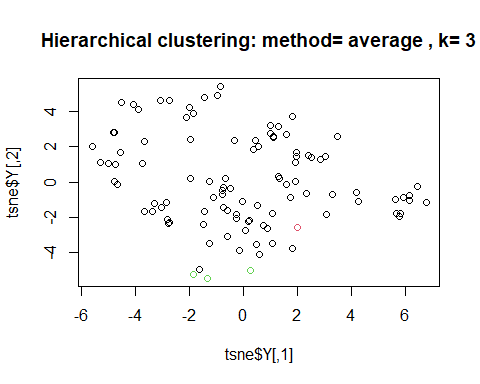
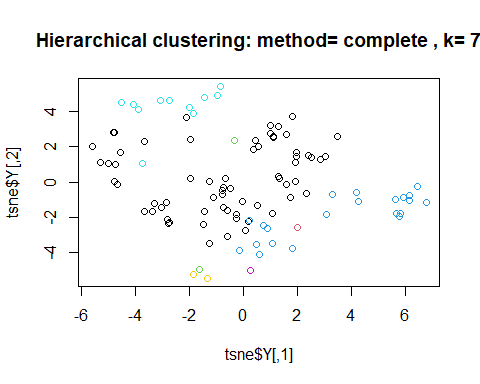
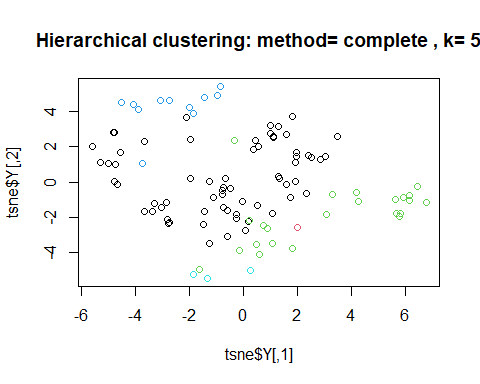
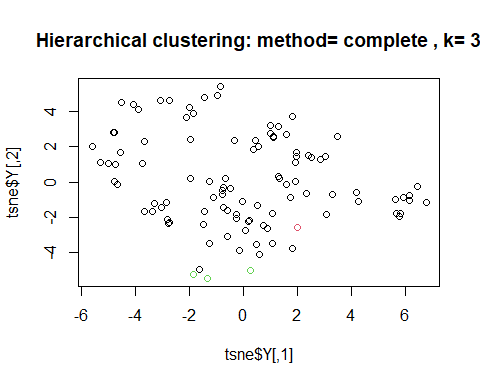
set.seed(555)  
  
# K-Means  
k\_values <- list(2,3,4,5,6,7)  
for(k in k\_values){  
 km.result <- kmeans(data.scaled, centers = k, nstart = 25)  
 plot(tsne$Y, col=km.result$cluster, main = paste("K-Means clustering: K=",k))  
}



Looking at the plots, it seems that the number of clusters that will give us with the most accurate clustering is K=7, as it provides the best visual separation out of all of the K’s we tried.

*Hierarchical clustering*

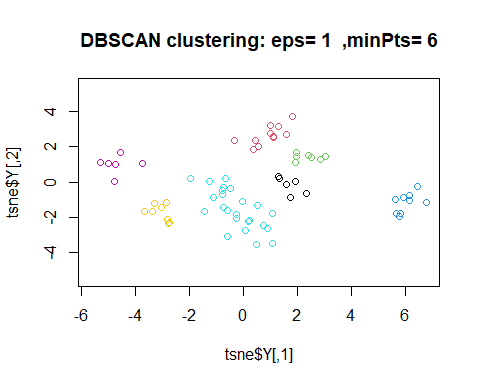
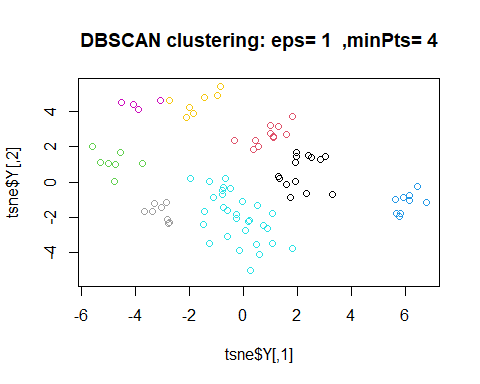
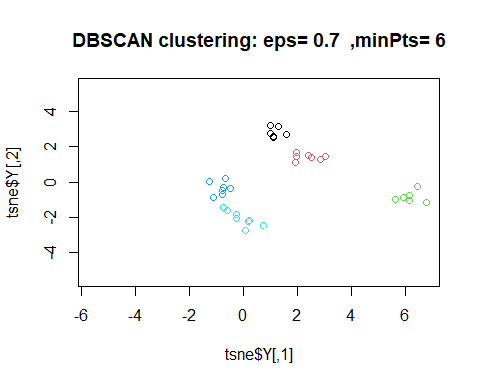
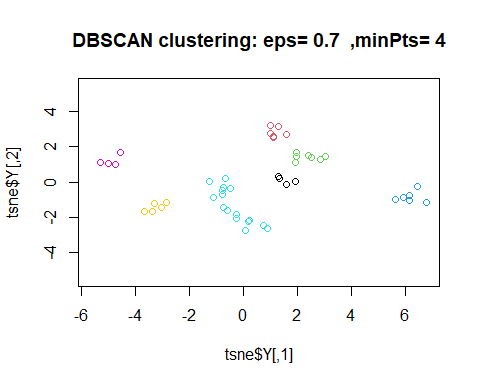
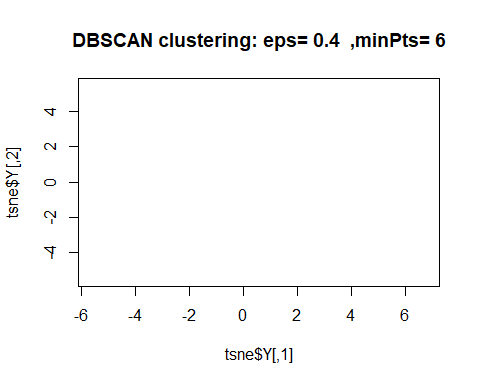
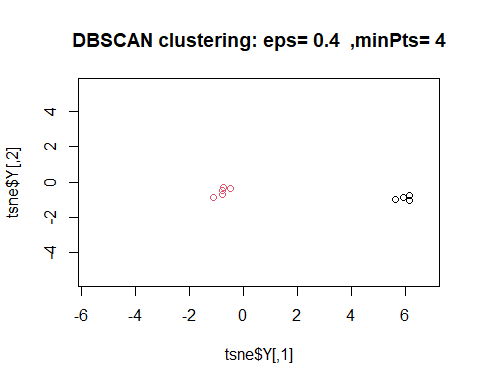
# hClust  
method\_values <- list("complete","average","ward.D","ward.D2","centroid")  
for(m in method\_values){  
 hc.result <- hclust(data.dist, method=m)  
 # had trouble with looping over cluster for some reason, so did it manually  
 cl3 <- cutree(hc.result, k=3)  
 cl5 <- cutree(hc.result, k=5)  
 cl7 <- cutree(hc.result, k=7)  
 hc.clusters <- list(cl3, cl5, cl7)  
 k <- 3  
 for(cl in hc.clusters){  
 plot(tsne$Y, col=cl, main = paste( "Hierarchical clustering: method=",m, ", k=", k))  
 k = k+2   
 }  
}



Once again, it seems that K=7 gives us the best result (more on it in Q3.3).

*DBSCAN*

# DBSCAN  
eps\_values <- list(0.4,0.7,1)  
minpts\_values <- list(4,6)  
for(e in eps\_values){  
 for(mp in minpts\_values){  
 dbscan.result <- dbscan(tsne$Y, eps = e, minPts = mp)  
 plot(tsne$Y, col=dbscan.result$cluster, main = paste("DBSCAN clustering: eps=",e," ,minPts=",mp))  
 }  
}

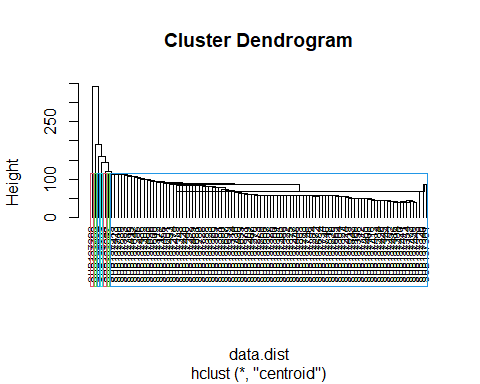


This time, the plots shows us that the number of clusters that will give us with the most accurate clustering is 6, while the parameters are: epsilon = 1, minPts = 6. Clustering using these parameters provides us with clusters that seems to be as far apart from each other, while also having the points in eacch cluster closest to each other.

# Q3.3

The method in hclust() that seems to give a clustering that is more accurate to the tSNE plot is" “ward.D2”, using k=7 (“cutting” the dendrogram into 7 different clusters). The dendrogram using these parameters will look as following:

# plotting a dendrogram - for our best k only  
best\_k <- 7  
plot(hc.result, cex = 0.6, hang = -1)  
rect.hclust(hc.result, k = best\_k, border = 2:5)



# Q3.4

In order to find the level of similarity of the 3 clustering techniques we used, we chose to use the function “rand.index” in the “fossil” library. This function allows us to understand how “close” clusters are to one another - the closer the result of the function is to 1 (on a scale from 0 to 1, while 1 means that the clusters are identical), the similar the clusters are to one another.

We used this function for each clustering method using the parameters that provided us the best clustering results in Q3.2.

# Clustering with the best parameters  
km.result <- kmeans(data.scaled, centers = 7, nstart = 25)  
# hierarchical clustering  
hc.result <- hclust(data.dist, "ward.D2")  
hc.cluster <- cutree(hc.result, k=7)  
# DBSCAN  
dbscan.result <- dbscan(tsne$Y, eps = 1, minPts = 6)  
  
# Finding the similarity levels  
rand.index(km.result$cluster, hc.cluster)

## [1] 0.9686333

rand.index(km.result$cluster, dbscan.result$cluster)

## [1] 0.7963032

rand.index(dbscan.result$cluster, hc.cluster)

## [1] 0.8119866

It can be noticed that the functions results revolved around 0.8-0.9. We assume that this means that while the clusters derived from each clustering method are not identical, they are also not too different from each other.

The similarity between the clusters might be caused by the size of the dataset - it contains a little more than 100 samples, a number that is fairly small in our opinion. A larger dataset might have produces a larger difference.

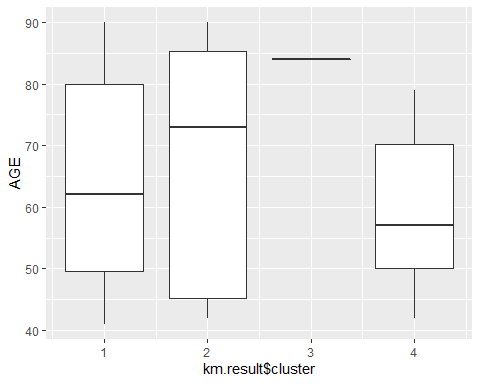
On the other hand, the difference between the clusters can be explained by the way that each clustering algorithm works. Each clustering technique has different rules on determining how data-points are divided into each cluster, and that could cause data-points that are on the “edge” of more than one cluster to be divided into different clusters using different algorithms.

# Q3.5

Last but not least, we tried to identify a connection between clusters and the demographic data provided to us in “samples\_info”.

To determine whether there is such a connection for any of the demographic variables, we experimented with different clustering methods and parameters, and finally chose to explain it with a *K-Means* clustering, as K=4.

sample\_info <- read.csv("C:/CS\_Technion/236523-Intro\_to\_Bioinformatics/Spr21/Intro\_to\_BioInf-Spr21-HW2/sample\_info.csv")  
  
# K-Means - with 4 clusters  
set.seed(555)  
km.result <- kmeans(data.scaled, centers = 4, nstart = 25)  
  
# AGE  
ggplot(sample\_info, aes(x=km.result$cluster, y=AGE, group = km.result$cluster)) + geom\_boxplot()



# RACE  
table(sample\_info$RACE, km.result$cluster)

##   
## 1 2 3 4  
## Asian 12 4 0 3  
## Black or African American 0 1 0 0  
## Native Hawaiian or Other Pacific Islander 0 1 0 0  
## Not Specified 3 1 0 0  
## Other 3 0 0 1  
## White 49 15 1 10

# GENDER  
table(sample\_info$GENDER, km.result$cluster)

##   
## 1 2 3 4  
## Female 37 15 0 7  
## Male 30 7 1 7

As can be seen in the tables, it seems that **RACE** and **GENDER** do not expalin the clusters that were chosen by the algorithm- *GENDER* - expect for **cluster number 2** , the number of samples in each cluster, divided into “Male” and “Female”, is almost identical. Even in cluster number 2 the difference doesn’t seems as significant. *RACE* - the same goes for RACE. The main difference here is that we can see that there seems to be an anomaly in **cluster number 4**, that, for some reason, does not contain the same proportion of “Asian” categorized samples as the in the other clusters. Unfortunately, we could not find a meaningful explanation for it.

When We look at the “AGE” variable, we *do* see a meaningful separation between clusters - cluster 3, which consists of one sample, is of “AGE” higher than the rest (~84). Cluster 2 samples’ median “AGE” is ~73, cluster 1’s is ~62, and cluster 4’s is ~57. As it obviously not unambiguous, we do think that this variable holds within it some explanation as to why did the 4 clusters were divided as they were.