Silesian University of Technology

Statistical Learning

Laboratory 8

Advanced Clustering and data dimension reduction methods.

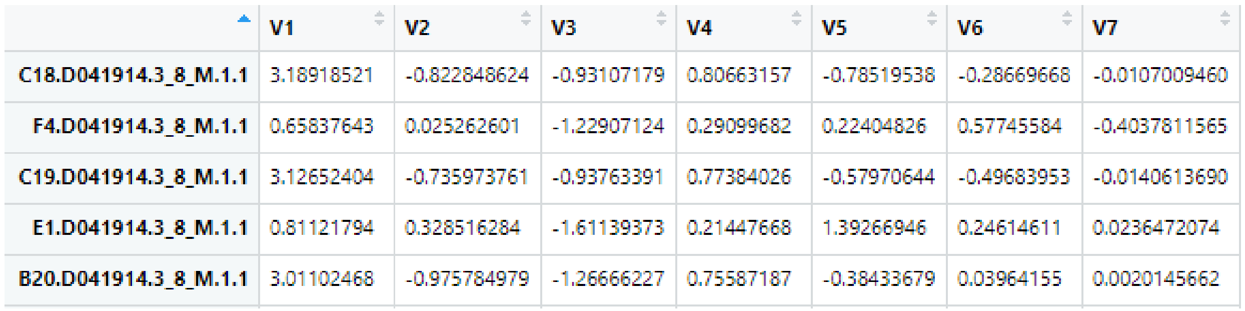
Jean Remy UWACU

Ankit Rathi

We used dataset13.

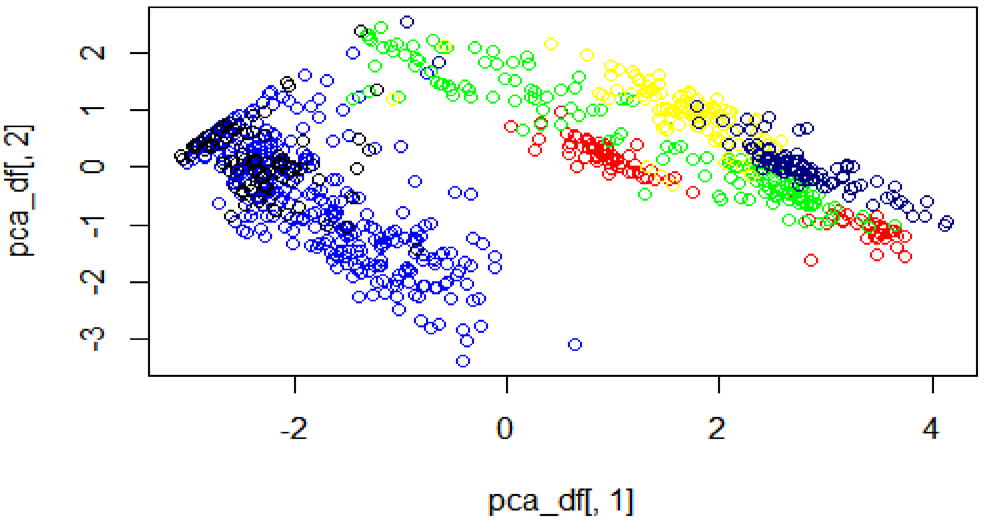
PART 1: Data Dimension Reduction methods.

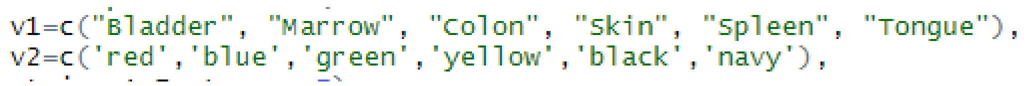
1. perform PCA step by step without the use of ready function for the part of the dataset A, consisting only expression levels for the following genes: Gsto1, Gstm1, Cd9, Prdx1, Coro1a, Rac2, Perp.



	V1	V2	V3	V4	V5	V6	V7
C18.D041914.3_8_M.1.1	3.18918521	-0.822848624	-0.93107179	0.80663157	-0.78519538	-0.28669668	-0.0107009460
F4.D041914.3_8_M.1.1	0.65837643	0.025262601	-1.22907124	0.29099682	0.22404826	0.57745584	-0.4037811565
C19.D041914.3_8_M.1.1	3.12652404	-0.735973761	-0.93763391	0.77384026	-0.57970644	-0.49683953	-0.0140613690
E1.D041914.3_8_M.1.1	0.81121794	0.328516284	-1.61139373	0.21447668	1.39266946	0.24614611	0.0236472074
B20.D041914.3_8_M.1.1	3.01102468	-0.975784979	-1.26666227	0.75587187	-0.38433679	0.03964155	0.0020145662

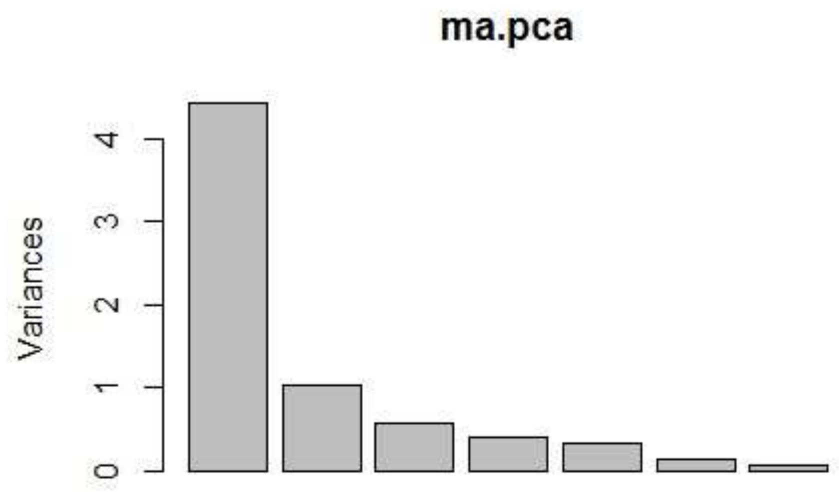
2. Draw the scatterplot of the two principal components (the ones with the highest proportion of variance). Remember to mark cells coming from various tissues differently. Comment on the results.

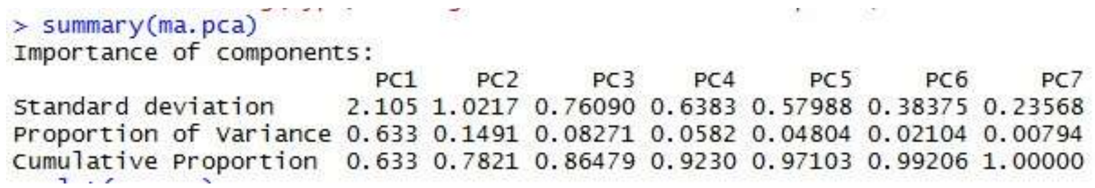




```
v1=c("Bladder", "Marrow", "Colon", "Skin", "Spleen", "Tongue"),
v2=c('red','blue','green','yellow','black','navy'),
```

It may be possible to conduct a good classification for some, but for example to distinguish properly between spleen and Marrow (black and blue) or Colon and Skin (green and yellow) is not that obvious.

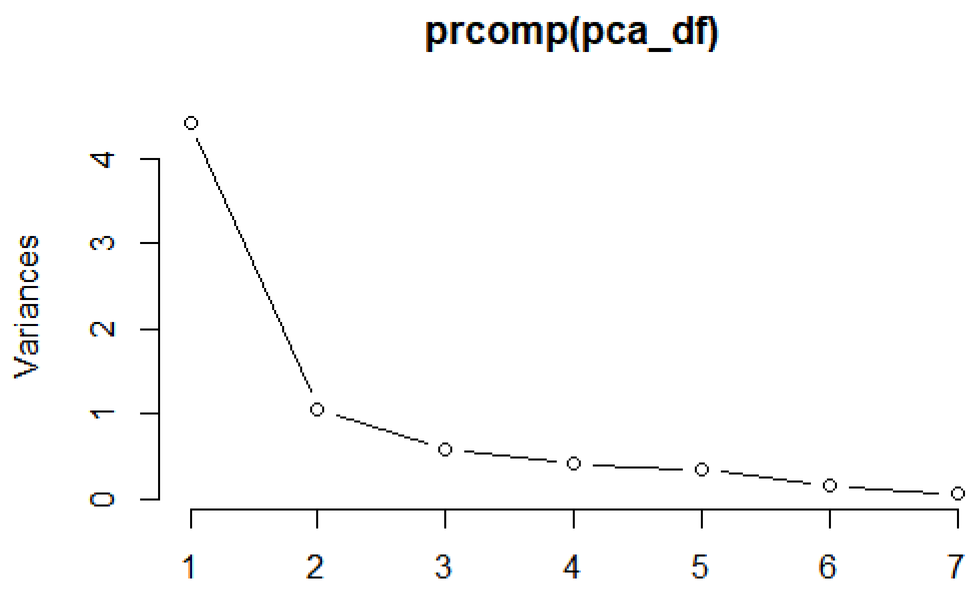




```
> summary(ma.pca)
Importance of components:
                         PC1    PC2     PC3    PC4     PC5     PC6     PC7
Standard deviation     2.105 1.0217 0.76090 0.6383 0.57988 0.38375 0.23568
Proportion of Variance 0.633 0.1491 0.08271 0.0582 0.04804 0.02104 0.00794
Cumulative Proportion  0.633 0.7821 0.86479 0.9230 0.97103 0.99206 1.00000
```

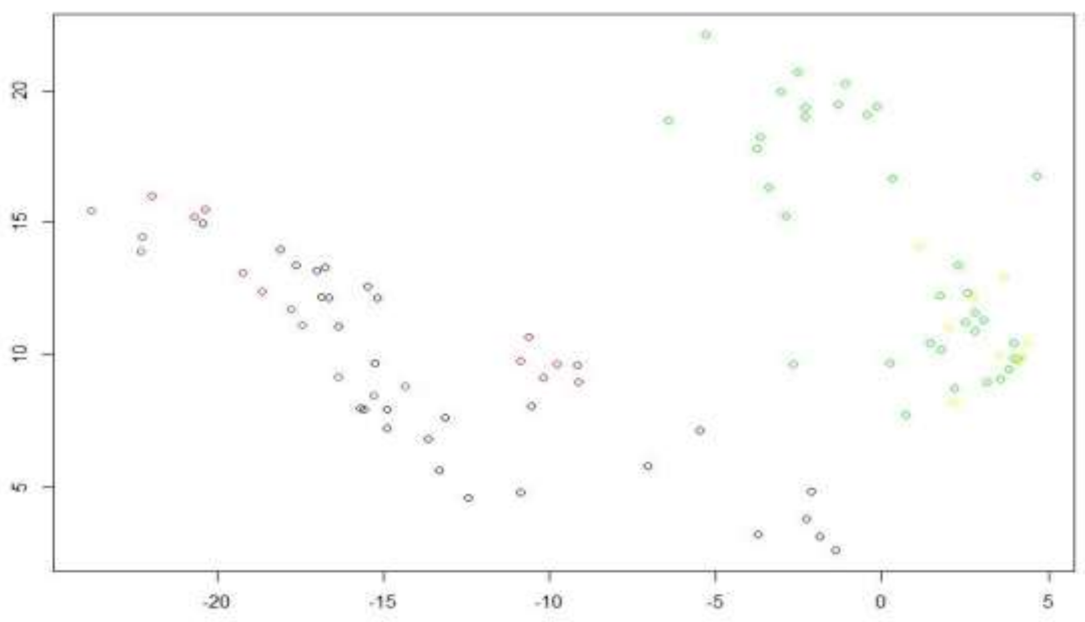
We obtain 7 principal components, which we call PC1-7. Each of these explains a percentage of the total variation in the dataset. That is to say: PC1 explains 63% of the total variance, which means that nearly two-thirds of the information in the dataset (7 variables) can be encapsulated by just that one Principal Component. PC2 explains nearly 15% of the variance. So, by knowing the position of a sample in relation to just PC1 and PC2, we can get a very accurate view on where it stands in relation to other samples, as just PC1 and PC2 can explain 78% of the variance.

3. **Draw the screeplot and comment on the results. Is the reduction to two dimensions sufficient in this case?**

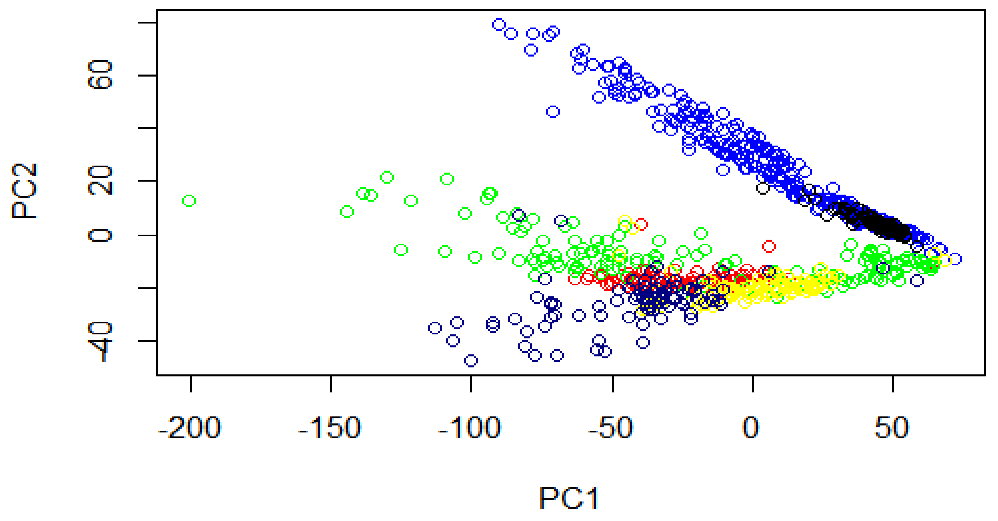


In my opinion the reduction to 2 pc is enough to give us an idea on the whole dataset and make some analysis. Because with the two principal components we have over 2/3 of the whole information from the dataset.

4. Predict the values of the principal components for the dataset B. Draw the scatterplot to visually evaluate the results of prediction. Remember to differently mark cells coming from various tissues as well as datasets A and B. Comment on the results.



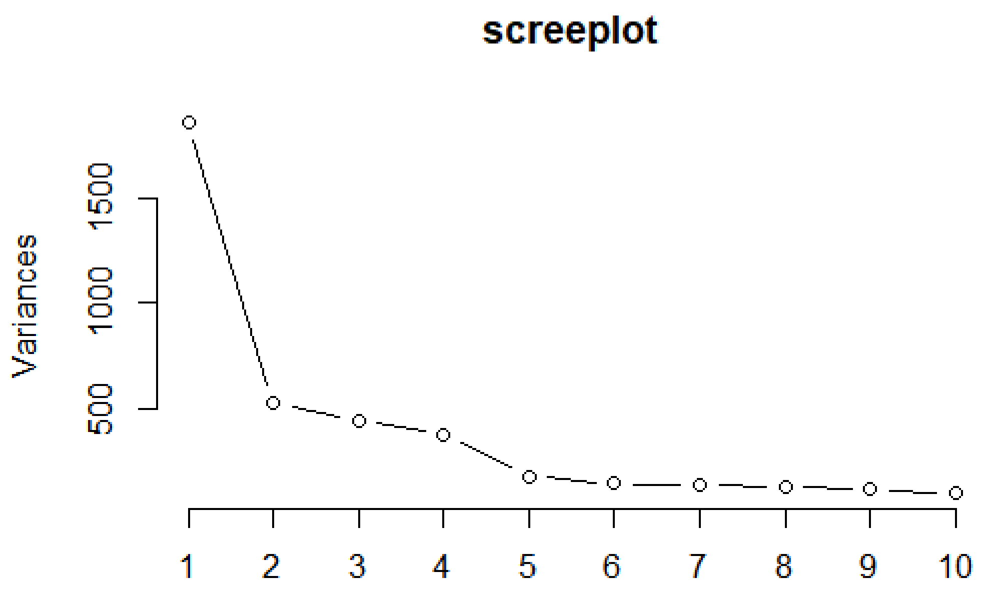
5. Perform PCA for the whole dataset A (with the use of the ready function). Draw the scatterplot of the two principal components (the ones with the highest proportion of variance). Remember to mark cells coming from various tissues differently. Comment on the results.



```
v1=c("Bladder", "Marrow", "colon", "skin", "spleen", "Tongue"),
v2=c('red','blue','green','yellow','black','navy'),
```

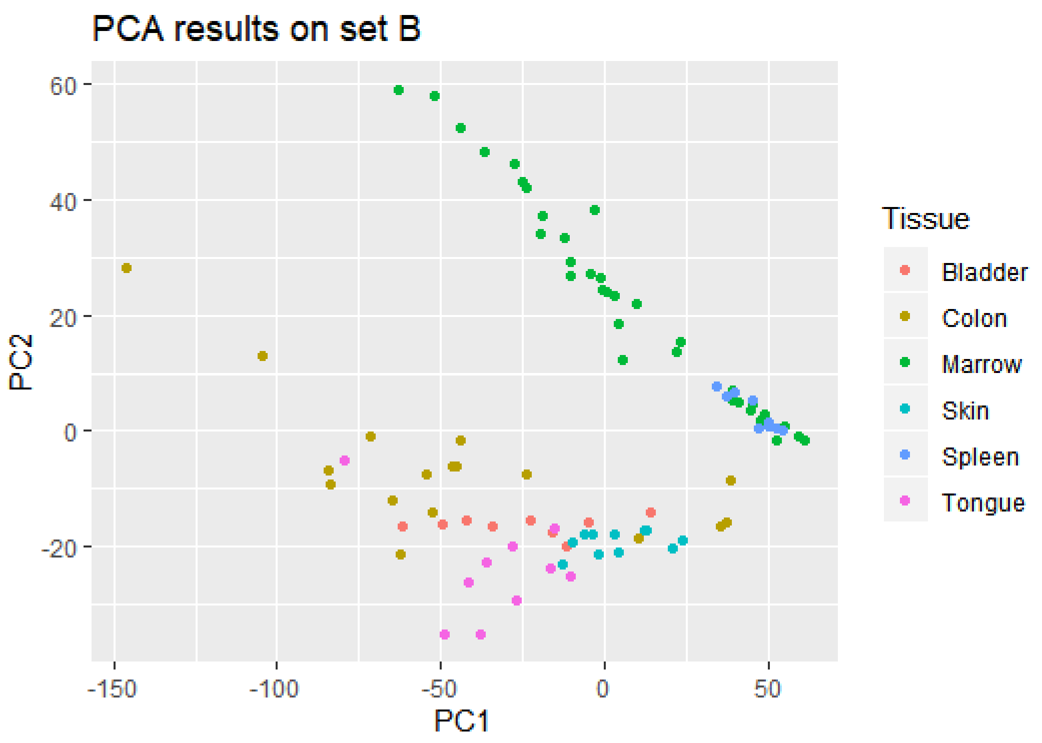
As we can see colon and skin(green and yellow), spleen and marrow(blue and black) it's hard to form a classification with them, we can see for some it may be easy to form classification.

6. Draw the screeplot for the first 20 components and comment on the results. Is the reduction to two dimensions sufficient in this case



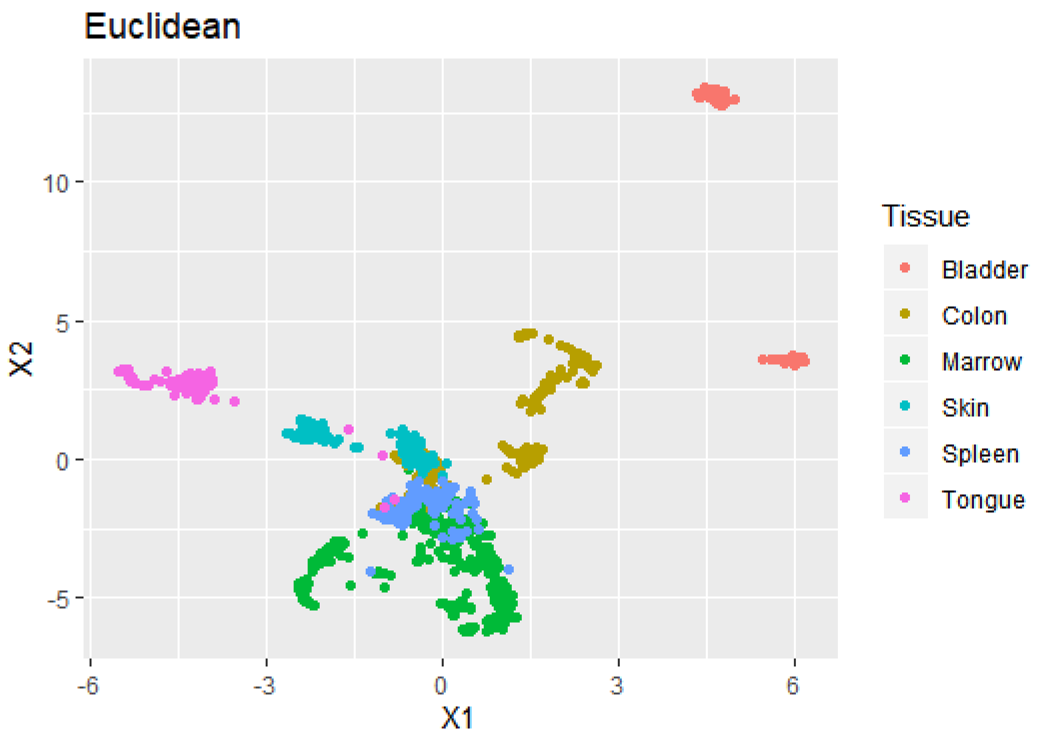
In my opinion, this time, reduction to 2 dimensions wouldn't be a good idea.At least 5 or 6 should be taken into consideration..

7. Predict the values of the principal components for the dataset B. Draw the scatterplot to visually evaluate the results of prediction. Remember to differently mark cells coming from various tissues as well as datasets A and B. Comment on the results.

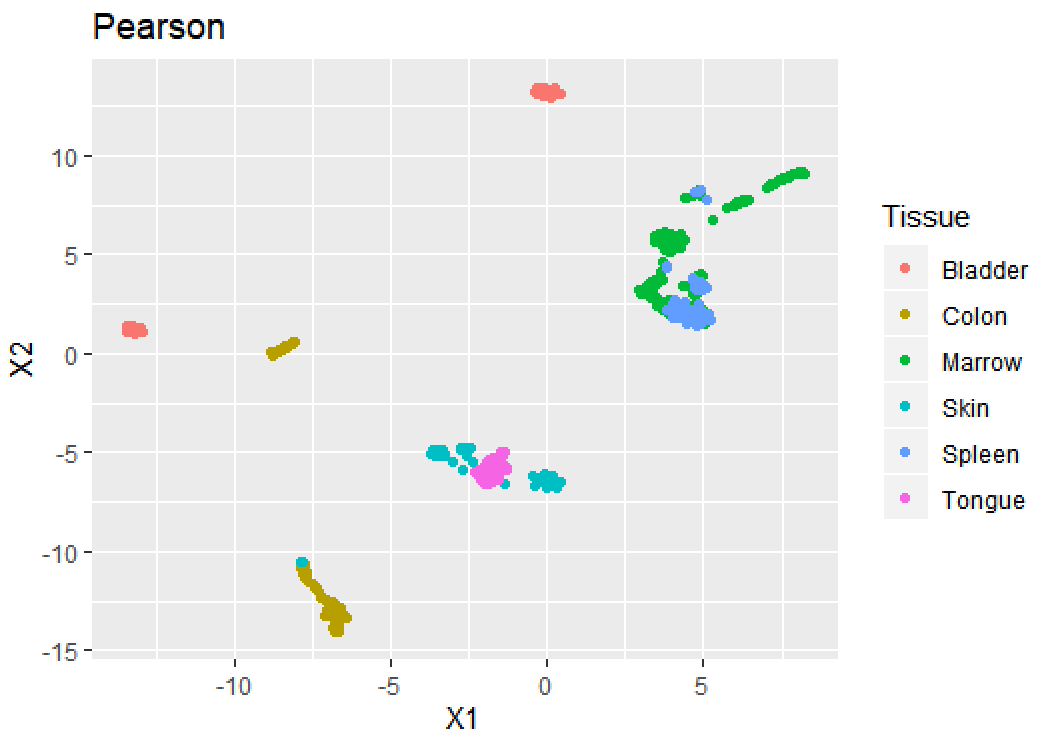


As we can see from the principal component result on set B, marrow and spleen are hard to distinguish so it will be hard to from a classification for them, whereas for colon is well segregated and it’s more convient to form a classification with it.

8. Perform UMAP for the whole dataset A. Compare two distance metrics: Euclidean and Pearson. For each of them draw the scatterplot of the results with cells from various tissues marked differently. Comment on the results.



While performing Umap for euclidean metrics we can see that bladder is segregated from rest of the cells whereas its hard to make a classification between marrow and spleen and some tongue cells are overlapping with skin, overall we can see it doesn’t makes good cluster which are easy to classify.



In here we had to perform Pearson metrics for UMAP, as we can see by the result skin and spleen are more corelated to each other whereas tongue and skin has much similarity to each other as we can see by the cluster formation Pearson Performs better task than that's of Euclidean.

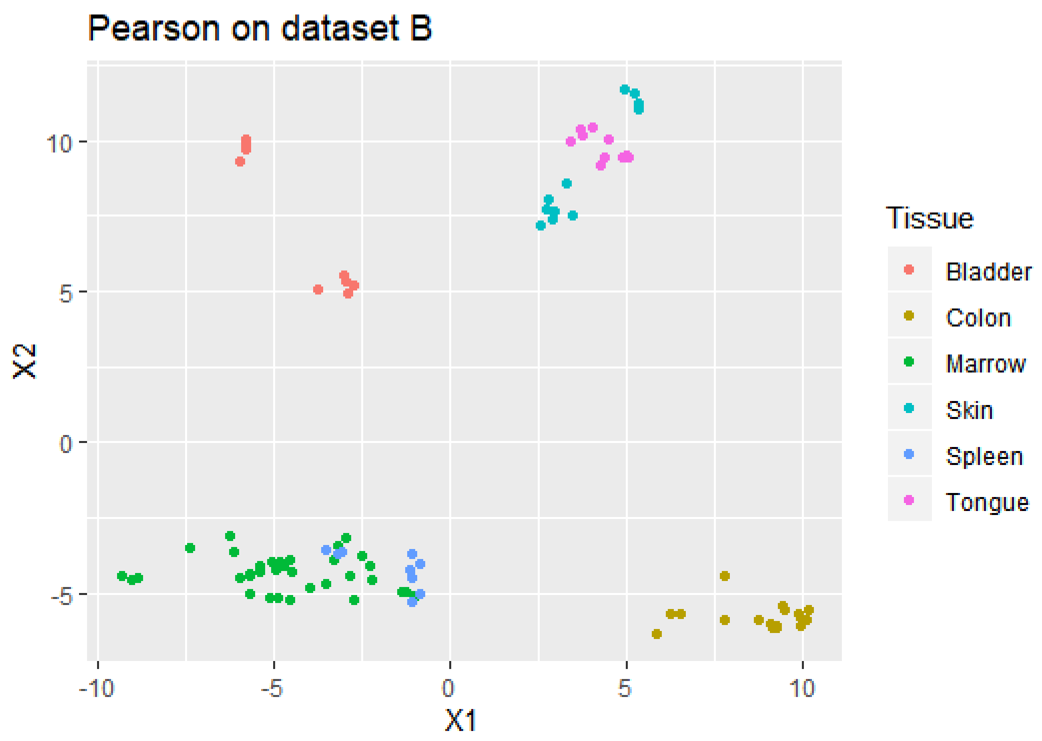
In this case Pearson performs better.

Note : On the left side of the plots above, the legend have different colors, from the prime definition. It is simply because we used different methods and definitions to try to see which one get us better results. From the code it is observable.

9. Choose the metric that allows for better tissue separation.

Pearson is better in this case.

10. Perform the prediction on the dataset B. Draw the scatterplot to visually evaluate the results of prediction. Remember to differently mark cells coming from various tissues as well as datasets A and B. Comment on the results.



We had to perform Pearson on dataset B, and after drawing the scatterplot we can see spleen and marrow tends to have similar classification whereas skin and tongue tends to go as same but they can be easily distinguished as they don't overlap each other.

11. Conclusions: compare results provided by PCA and UMAP.

UMAP gives us better results than normal PCA(clearly it can be seen that UMAP Pearson does a better job than PCA for dimensionality reduction).

12. Conclusions: what can you say about differences between various tissues based on your results?

    Skin and tongue tends to have similar classification whereas for marrow and spleen it's hard to distinguish between the cells/tissues.

    Code used for part 1 :

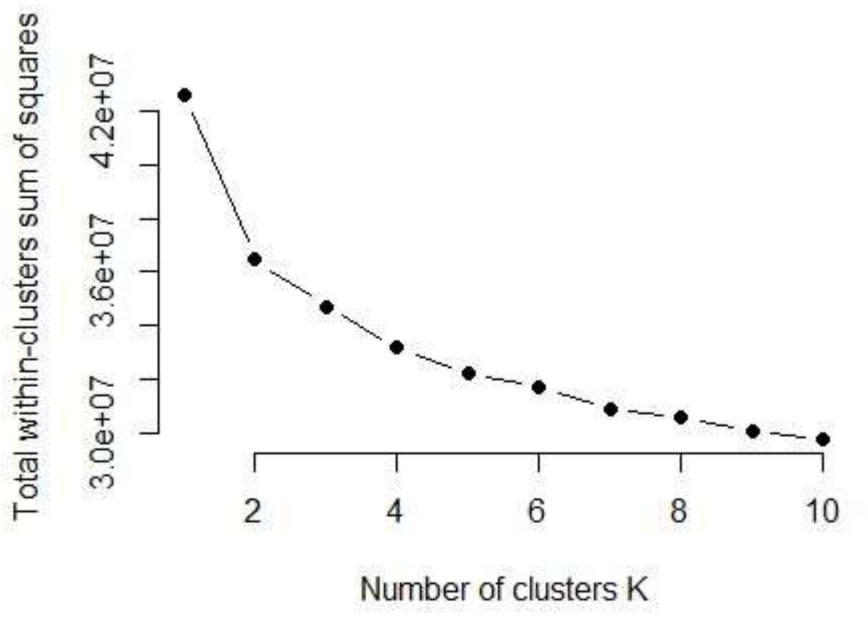
```
setwd("/Users/rathi/Downloads/Set13/Set13")
library(dplyr)
library(ggplot2)
library(gplots)
library(umap)
library(dendextend)
library(mclust)

df_1=read.delim("setA.txt",stringsAsFactors = F,sep=' ')
df_2=read.delim("setB.txt",stringsAsFactors = F,sep=' ')
#Q1
#Perform PCA step by step without the use of ready function for the part of the dataset A, consisting
#only expression levels for the following genes: Gsto1, Gstm1, Cd9, Prdx1, Coro1a, Rac2, Perp.

my_pca<-function(df){

  m=as.matrix(scale(df)) #center scaling and making it a matrix
  m=cov(m)   #covariance matrix
  eigenv=eigen(m)
  eigenv_1=-eigenv$vectors
  return(as.matrix(scale(df)) %*%eigenv_1)
}

features<-c('Gsto1', 'Gstm1', 'Cd9', 'Prdx1', 'Coro1a', 'Rac2', 'Perp')


pca_df<-my_pca(df_1[,features])

#Q2
#scatter plot for
color_map<-data.frame(
  v1=c("Bladder", "Marrow", "Colon", "Skin", "Spleen", "Tongue"),
  v2=c('red','blue','green','yellow','black','navy'),
  stringsAsFactors = F)

df_1_col<-df_1%>%left_join(color_map,
               by=c('Tissue'='v1'))

plot(pca_df[,1],pca_df[,2],
     col=df_1_col[,'v2'])
```

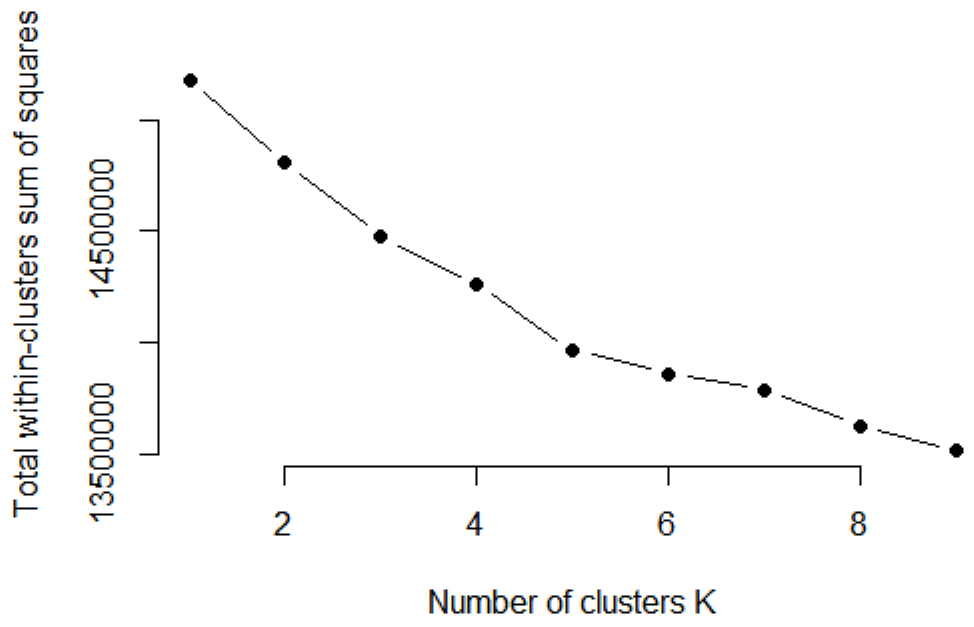
```
#Q3
#scree plot of percent variance explained

variance_explained<-pca_df%>%apply(2,sd)
perc_varianc_explained<-variance_explained/sum(variance_explained)
plot(1:length(perc_varianc_explained),
     100*cumsum(perc_varianc_explained),'l',
     xlab='component',
     ylab='variantion_explained',
      main='Percent Variance Explained')
#Q4
#predict for dataset B


pca_1<-prcomp(df_1[,features],scale. = T)
prediction_for_set_b=predict(pca_1,df_2[,features])


#Q5
#with the use of ready function

sum_of_na<-function(x){
  return(sum(is.na(x)))
}

zero_sd<-apply(df_1,2,sd)

features_having_zero_sd<-zero_sd[zero_sd==0]

#removing the features that are constant i.e. have 0 sd
# colnames(df_1)%>%
#   setdiff(features_having_zero_sd%>%names())%>%length()

non_zero_sd_feats<-colnames(df_1)%>%
  setdiff(features_having_zero_sd%>%names())


numeric_feats<-df_1%>%apply(2,is.numeric)


df_1_a <- mutate_all(df_1, function(x) as.numeric(as.character(x)))

non_numeric_cols<-dplyr::select_if(df_1, is.numeric)%>%colnames()
non_numeric_cols<-colnames(df_1)%>%setdiff(non_numeric_cols)
non_numeric_cols # Tissue

pca_overall<-prcomp(df_1[,non_zero_sd_feats%>%setdiff('Tissue')],scale. = T)

#plotting top two pc1, pc2
color_map<-data.frame(
  v1=c("Bladder", "Marrow", "Colon", "Skin", "Spleen", "Tongue"),
  v2=c('red','blue','green','yellow','black','navy'),
  stringsAsFactors = F)

df_1_col<-df_1%>%left_join(color_map,
                           by=c('Tissue'='v1'))


plot(pca_df[,1],pca_df[,2],
     col=df_1_col[,'v2'])


plot(pca_overall$x[,c(1,2)],col=df_1_col[,'v2'])
```

```
#Q6
#percent variance explained
#screeplot

variance_explained_overall<-pca_overall$sdev/sum(pca_overall$sdev)
variance_explained_overall_cum<-cumsum(variance_explained_overall)

plot(1:20,
     100*variance_explained_overall_cum[1:20],'l',
     xlab='component',
     ylab='variantion_explained',
     main='Percent Variance Explained')


#Q7 prediction on setB
setb_predicted_pca<-predict(pca_overall,
                         df_2[,non_zero_sd_feats%>%setdiff('Tissue')])


ggplot(setb_predicted_pca%>%data.frame()%>%
         dplyr::bind_cols(df_2%>%select(Tissue)), aes(x=PC1, y=PC2, color=Tissue)) +
 geom_point()+ ggplot2::ggtitle('PCA results on set B')



#Q8Perform UMAP for the whole dataset A. Compare two distance metrics: Euclidean and Pearson. For
# of them draw the scatterplot of the results with cells from various tissues marked differently.

#Euclidean
umap.defaults
config_eculidiean=umap.defaults

umpa_df_a = umap(df_1[,colnames(df_1)%>%setdiff('Tissue')],
                config=config_eculidiean)


ggplot(umpa_df_a$layout%>%
  data.frame()%>%
  bind_cols(df_1%>%select("Tissue")),
  aes(x=X1,y=X2,color=Tissue))+geom_point()+ggplot2::ggtitle('Euclidean')



#Pearson
umap.defaults
config_pearson=umap.defaults
config_pearson$metric='pearson'

umpa_df_a_pearson = umap(df_1[,colnames(df_1)%>%setdiff('Tissue')],
               config=config_pearson)


ggplot(umpa_df_a_pearson$layout%>%
         data.frame()%>%
         bind_cols(df_1%>%select("Tissue")),
       aes(x=X1,y=X2,color=Tissue))+geom_point()+ggplot2::ggtitle('Pearson')

#pearson displays better seggregation

#Q9 . Choose the metric that allows for better tissue separation
#going with pearson i.e. umpa_df_a_pearson


#.Q10 Perform the prediction on the dataset B. Draw the scatterplot to visually evaluate the results of

umpa_df_b_pearson=predict(umpa_df_a_pearson,
                        df_2[,colnames(df_2)%>%setdiff('Tissue')])


ggplot(umpa_df_b_pearson%>%data.frame()%>%
        bind_cols(df_2%>%select("Tissue")),
      aes(x=X1,y=X2,color=Tissue))+geom_point()+ggplot2::ggtitle('Pearson on dataset B')

#clearly it can be seen that UMAP Pearson does a better job than PCA for dimensionality reduction
```

PART 2 : K-Means Clustering

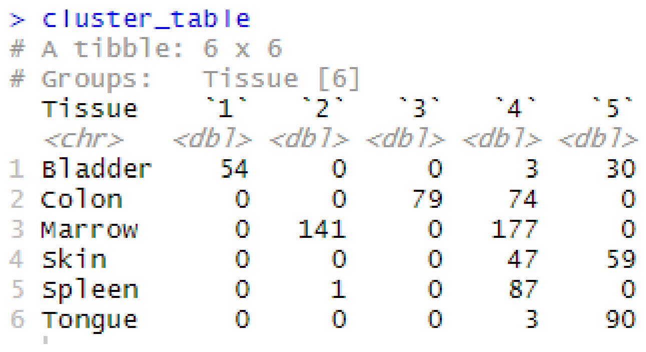
1. **Perform k-means clustering for the number of clusters varying from 2 to 10 with at least 5 random sets. Draw the elbow plot of the total within-cluster sum of squares for each number of clusters and choose the optimal number of clusters based on this plot.**





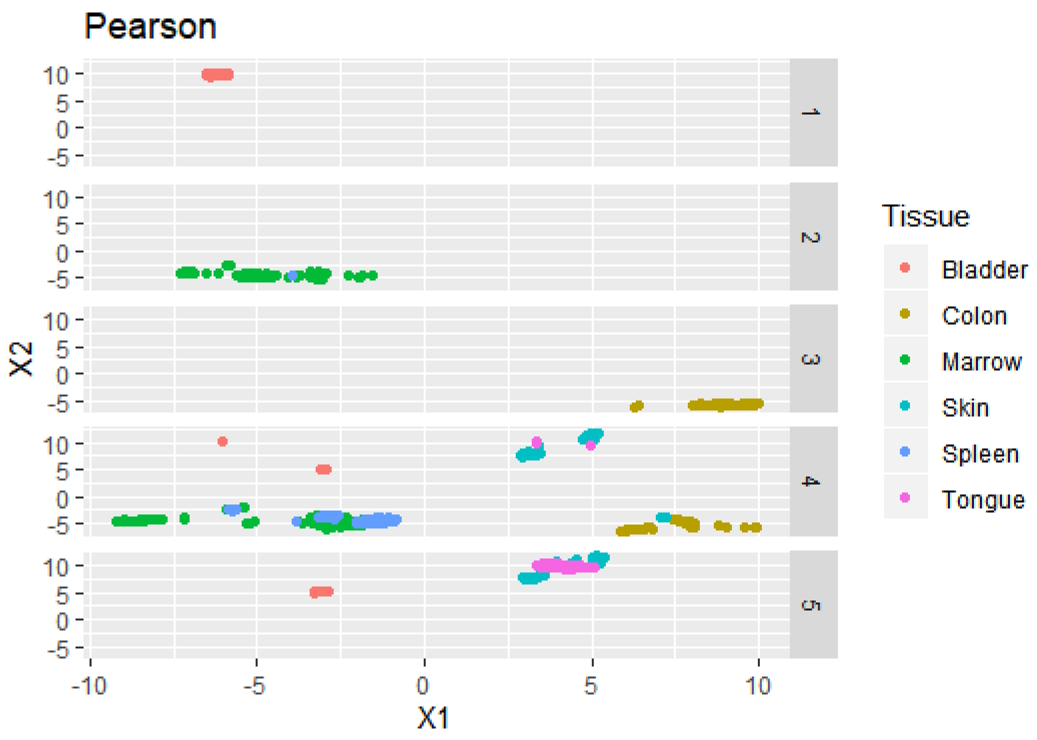
We have chose 5 as the optimal number of clusters.

2. **Perform k-means clustering for the optimal number of clusters. Investigate the composition of each cluster with regard to tissues (present the results in the table).**



```
> cluster_table
# A tibble: 6 x 6
# Groups:   Tissue [6]
  Tissue    `1`   `2`   `3`   `4`   `5`
  <chr>   <dbl> <dbl> <dbl> <dbl> <dbl>
1 Bladder    54     0     0     3    30
2 Colon       0     0    79    74     0
3 Marrow      0   141     0   177     0
4 Skin        0     0     0    47    59
5 Spleen      0     1     0    87     0
6 Tongue      0     0     0     3    90
```

3. Draw the scatterplot of the UMAP results with cells from various k-means clusters differently. Comment on the results.



Marrow and spleen (green and blue), tongue and skin(pink and sky blue) are hard to classify as they're making similar clusters which are overlapping, where as for bladder it's easy to classify and same is the case with colon.

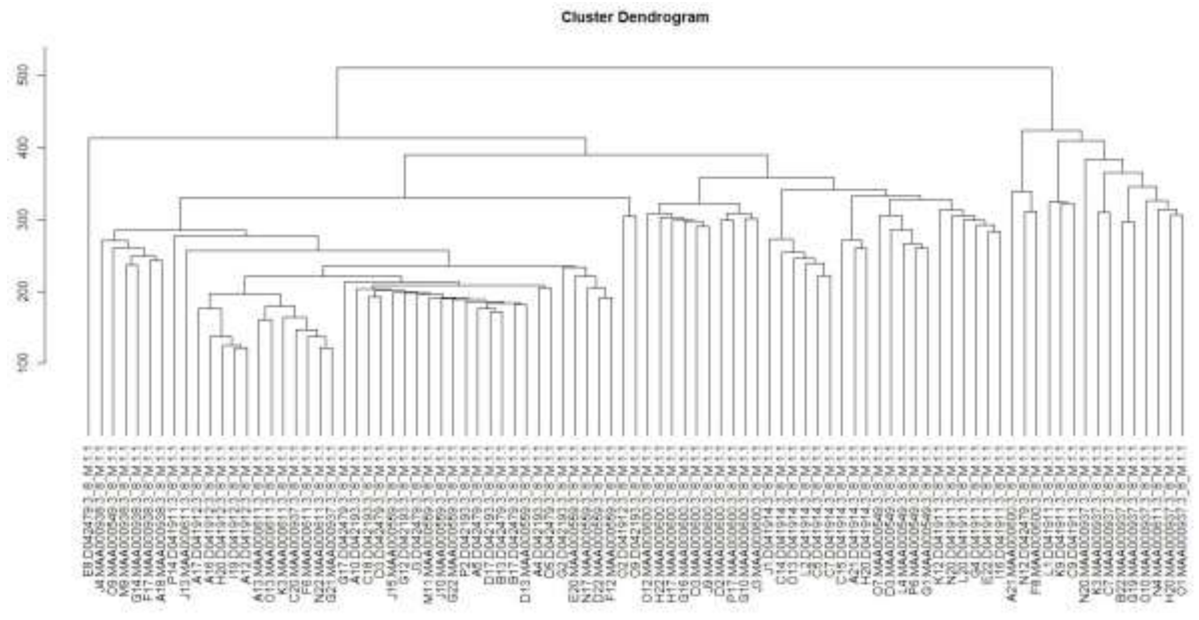
4. Comment on the results. Are cells from different tissues separated? Are clusters homo- or heterogeneous
   Heterogenous Most of them are but we can see that brown cluster containes some of Colon (originally yellow), Spleen (originally blue), Marrow (originally green)

Code used for this part :

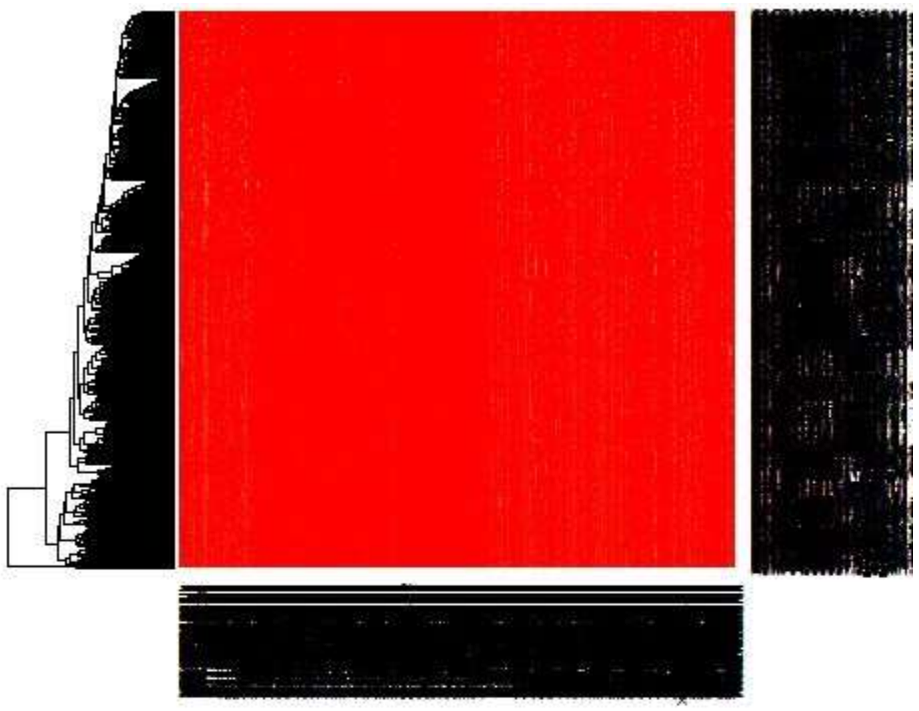
```
k.max <- 10 df_a_scaled <- df_1%>%select(-Tissue)%>%scale() na_cols<-
df_a_scaled%>%apply(2,sum_of_na) non_na_cols<-na_cols[na_cols==0]%>%names()
wss_clusters<-c() for(k in 2:k.max){ print(k) wss<-kmeans(df_a_scaled[,non_na_cols], k,
nstart=5,iter.max = 10 )$tot.withinss  wss_clusters<-c(wss_clusters,wss)}
plot(1:length(wss_clusters), wss_clusters,  type="b", pch = 19, frame = FALSE,  xlab="Number
of clusters K", ylab="Total within-clusters sum of squares" k=5 #optimal
clusterskmeans_optimal<-kmeans(df_a_scaled[,non_na_cols], k, nstart=5,iter.max = 20)
df_a_scaled_kmeans<-df_1 df_a_scaled_kmeans<-
df_a_scaled_kmeans%>%bind_cols(data.frame(cluster=kmeans_optimal$cluster))
cluster_table<-df_a_scaled_kmeans%>% dplyr::group_by(Tissue,cluster)%>%
dplyr::summarise(count=n())%>%           tidyr::spread(key=cluster,value=count,fill=0)
cluster_table  umap_set_a<-data.frame(umpa_df_a_pearson$layout
umap_set_a$cluster=kmeans_optimal$cluster
ggplot(umap_set_a%>%bind_cols(df_1%>%select("Tissue")),  aes(x=X1,y=X2,color=Tissue))
geom_point()+  ggplot2::ggtitle('Pearson')+ facet_grid(cluster~.)
```

PART 3:

1. **Perform hierarchical clustering and draw the dendrogram. Comment on the results.**



2. Draw the heatmap of the first 1000 features with the dendrogram corresponding to cells (not genes). Comment on the results. Do expression levels of all genes vary across different cells?

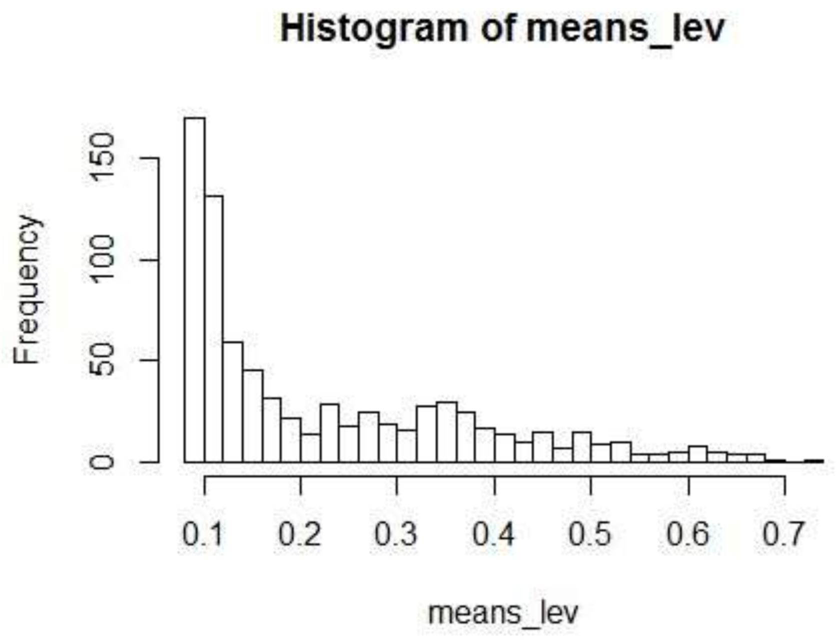


They don't.

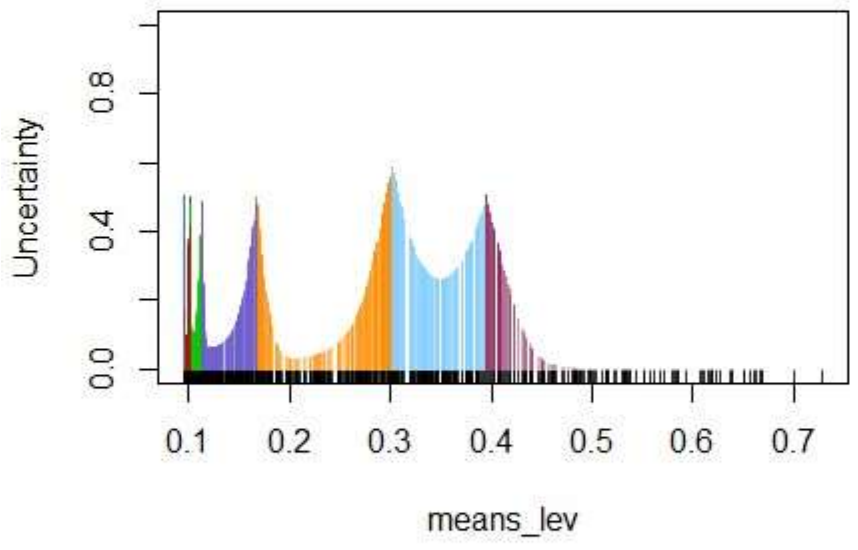
3.

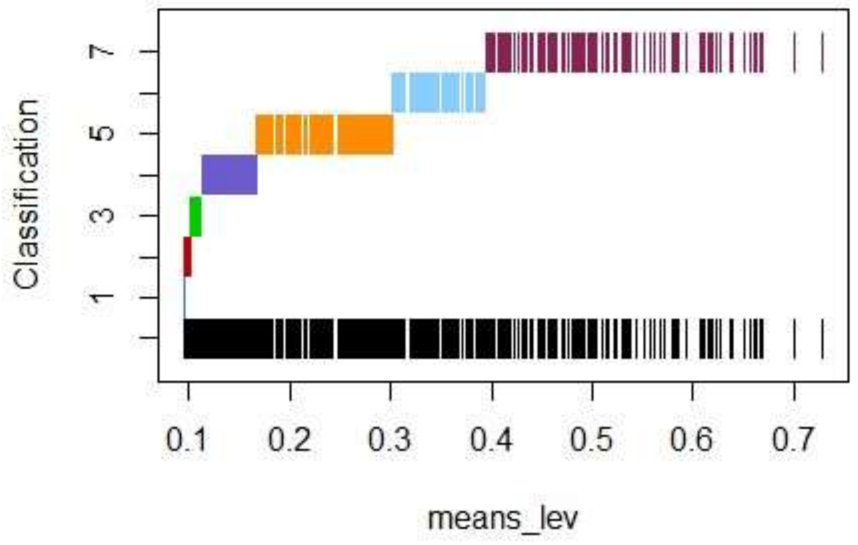
422.

4. **Compute mean expression levels in log10 scale for the remaining set of genes. Draw the histogram of those values with the number of bins equal to the square root of the number of genes**

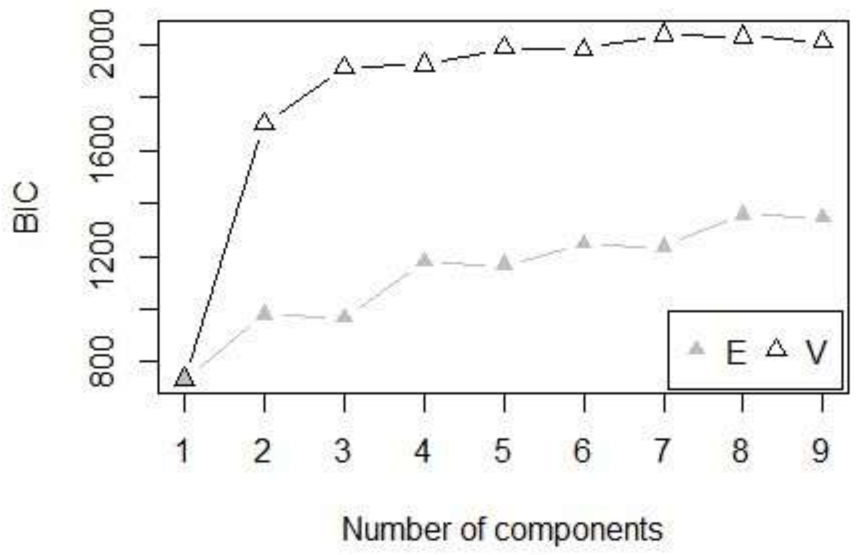


5. **Perform GMM decomposition of the mean values in log10 scale. Plot the components.**



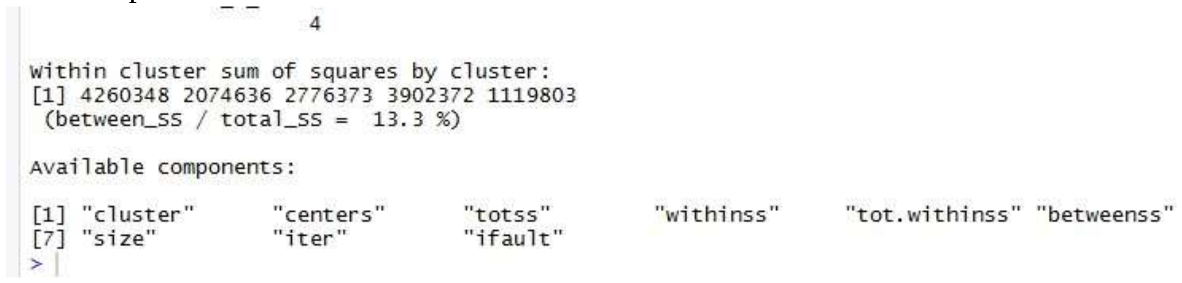


Classification
1
3
5
7
0.1
0.2
0.3
0.4
0.5
0.6
0.7
means_lev

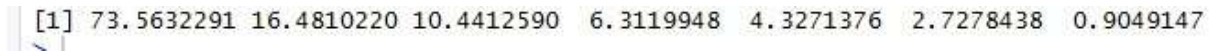


BIC
800
1200
1600
2000
1
2
3
4
5
6
7
8
9
Number of components
E
V

Kmeans optimal



```
                        4

Within cluster sum of squares by cluster:
[1] 4260348 2074636 2776373 3902372 1119803
 (between_SS / total_SS =  13.3 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"
[7] "size"         "iter"         "ifault"
> 
```



```
[1] 73.5632291 16.4810220 10.4412590  6.3119948  4.3271376  2.7278438  0.9049147
> 
```

Eigen values

Code used for this part :

```
df_b_scaled <- df_2%>%select(-Tissue)%>%scale()
na_cols<-df_b_scaled%>%apply(2,sum_of_na)
non_na_cols<-na_cols[na_cols==0]%>%names()


dist_mat <- dist(df_b_scaled[,non_na_cols], method = 'euclidean')
hclust_avg <- hclust(dist_mat, method = 'average')
plot(hclust_avg)

#Draw the heatmap of the first 1000 features with the dendrogram corresponding to cells (not
genes).

dend_r <- df_b_scaled[,non_na_cols] %>% dist(method = "euclidean") %>% hclust() %>%
as.dendrogram

dend_c <- t(df_b_scaled[,non_na_cols]) %>% dist(method = "euclidean") %>% hclust() %>%
as.dendrogram %>%
  color_branches(k=1000)

gplots::heatmap.2(as.matrix(df_b_scaled[,non_na_cols]),
                  srtCol = 35,
                  Colv = dend_c,
                  trace="row", hline = NA, tracecol = "darkgrey",
                  margins =c(6,3),
                  key.xlab = "no / yes",
                  denscol = "grey",
                  density.info = "density",
)
hc <- hclust(dist(df_2))
plot(hc)
plot(hc, hang = -1)


data_labels <- data[,23434]
data_unlabeled <- data [,1:23433]
data_1st_1000 <- data[,1:1000]


new_data_mat <- as.matrix(data_1st_1000)
heatmap(new_data_mat, Colv=NA)


x <- c()


for( i in 1:1000){
  l<-length(unique(new_data_mat[,i]))
  if (l==1){
    x<-append(x,i)
  }
}


new_df<-data_1000[-x]

means<-colMeans(new_df)
means_lev<-log10(means)
View(means_lev)
num_of_bins=as.integer(sqrt(length(means_lev)))
hist(means_lev, breaks=num_of_bins)



our_gmm <- function(mu1,mu2,s1,s2, p1, p2)
  {
  a=(s1^2-s2^2)/(2*s1^2*s2^2)
  b=(mu1*s2^2-mu2*s1^2)/(s1^2*s2^2)
  c=(-mu1^2*s2^2+mu2^2*s1^2)/(2*s1^2*s2^2)-log((s1*p2)/(s2*p1))


  delta=b^2-4*a*c

  x_1 = (-b+sqrt(delta))/(2*a)
  x_2 = (-b-sqrt(delta))/(2*a)
  result = c(x_1,x_2)

  return(result)
   }


z <- Mclust(means_lev)
our_booststrap <- MclustBootstrap(z)
plot(z)
```