Block 3: Clustering

Huizhi Lin (88112518130) & Jan Orsel (950608630010)

November 14, 2017

In this report we apply hierarchical clustering, and apply K-means clustering to the dataset containing skin tissue. This data contains samples derived from normal and tumour samples. Clustering is preformed separately for the genes and the samples. The genes are clustered based on their expression in different samples and the samples are clustered based on the expression levels of the genes.

To start with we read in the data in two variables, one for the original data and one for the transposed version.

```
skin.df <- read.table('get_normal_vs_tumor2_RAW_Skin.out',sep='
',header=TRUE,stringsAsFactors=FALSE)

skin.tdf <- data.frame(t(skin.df[,-2562]))
colnames(skin.tdf)<-paste0(skin.df$tissue,1:72)
```

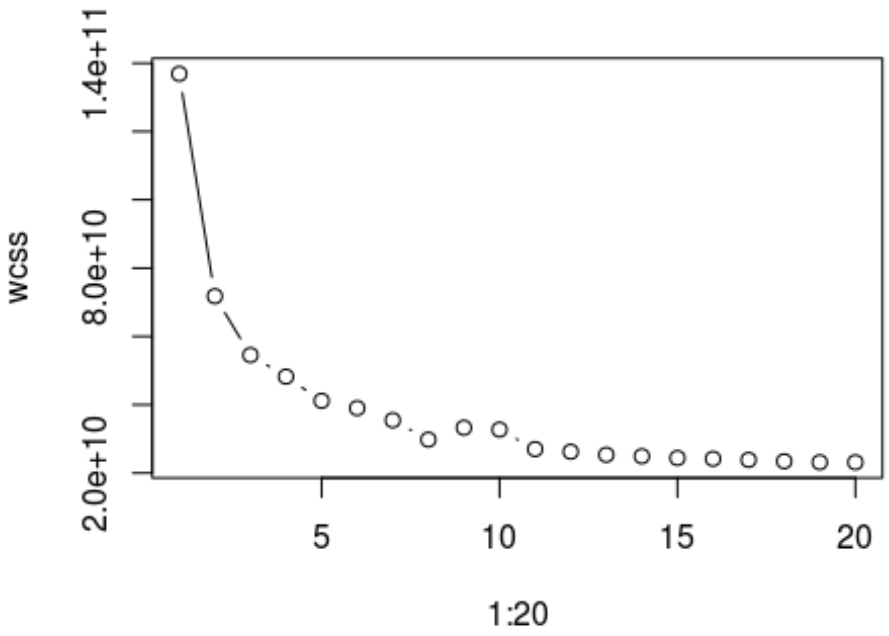
Task 1: Clustering the genes

We used the transposed data frame for K-means clustering of the genes. First we tried k-mean clustering with k=2. For nstart we chose the value 20, this means that of 20 attempts to create 2 clusters the best attempt is shown.

```
dim(skin.tdf)

## [1] 2561   72

km.gene <- kmeans(skin.tdf,2,nstart=20)
```

Then we used Elbow Method to check the optimal k value. This method visualises when a clustering attempt does not get any "better" by a flattening of a curve.

```
set.seed(6)
wcss <- vector()
for (i in 1:20) wcss[i] <- sum(kmeans(skin.tdf,i)$withinss)
plot(1:20,wcss,type='b')
```



Based on this plot, we chose k=8 as the optimal k value. Because after point 8 there is no significant improvement on the clustering. Then we proformed k-mean clustering again with k=8.

```
km.gene8 <- kmeans(skin.tdf,8,nstart=20)
```

In order to check whether this elbow method actually delivered on it's promises we compared the total value between of the k=2 attempt to the k=8 attempt.

```
km.gene$tot.withinss

## [1] 71774033699

km.gene8$tot.withinss

## [1] 29741890462
```

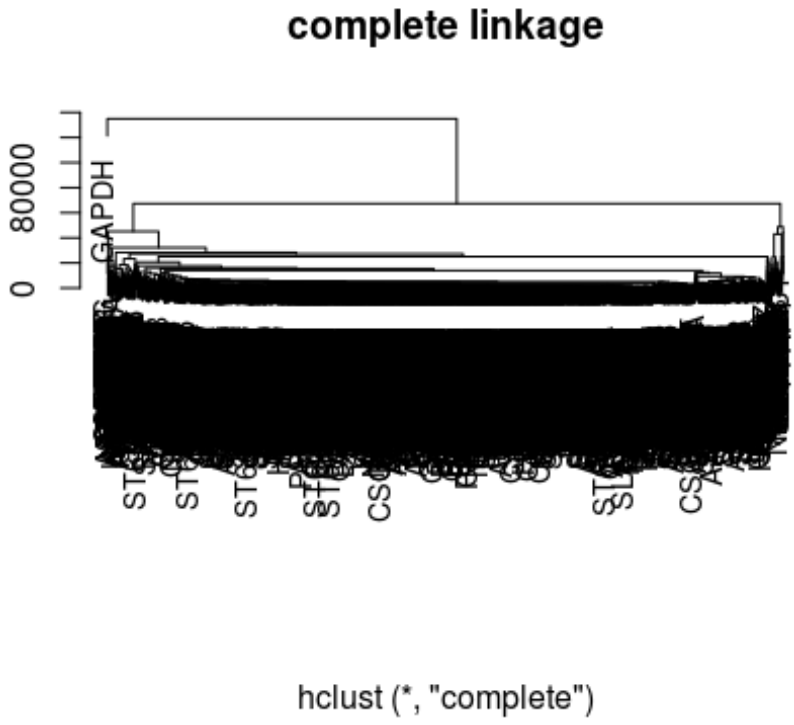
Km.gene8$ gives a far smaller number for tot.withinss. the value for tot.withinss is the sum of the within-cluster sum of squares for each cluster and can therefore be used to assess how good the clustering result is. Based on these two numbers, we conclude that 8 clusters method is far better than the 2 clusters method.

hierarchical clustering

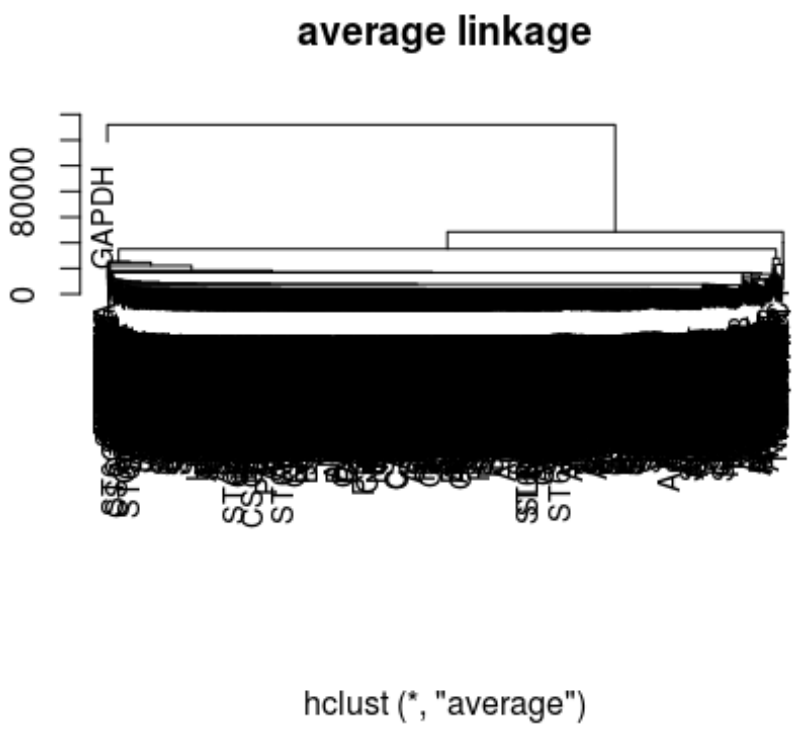
For hierarchical clustering on the genes, using the Euclidian distanc, there are two ways work with this distance. This first block of code shows how we preformed hierarchical

clustering using complete and average linkage. This block contains the code for the first plot too.

```
# use complete linkage
hc.gene.com <- hclust(dist(skin.tdf),method="complete")
# use average linkage
hc.gene.avg <- hclust(dist(skin.tdf),method="average")
#create denrogram
plot(hc.gene.com,main="complete linkage",xlab='',ylab='',cex=.9)
```

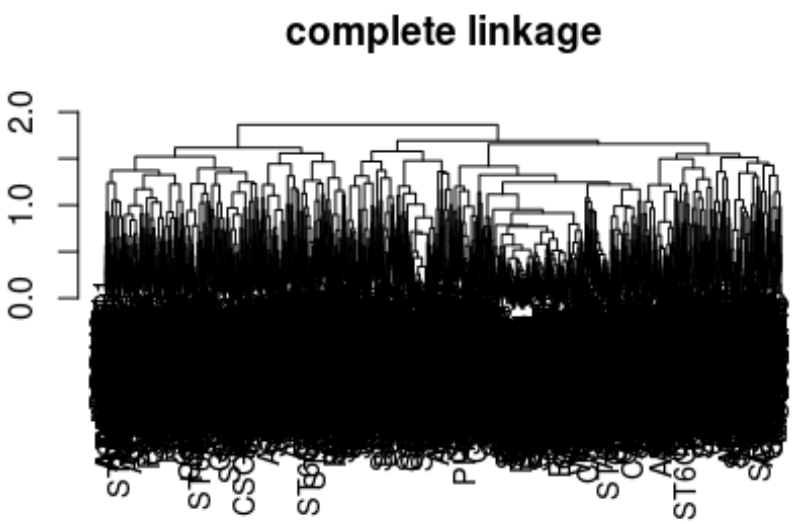



```
plot(hc.gene.avg,main="average linkage",xlab='',ylab='',cex=.9)
```



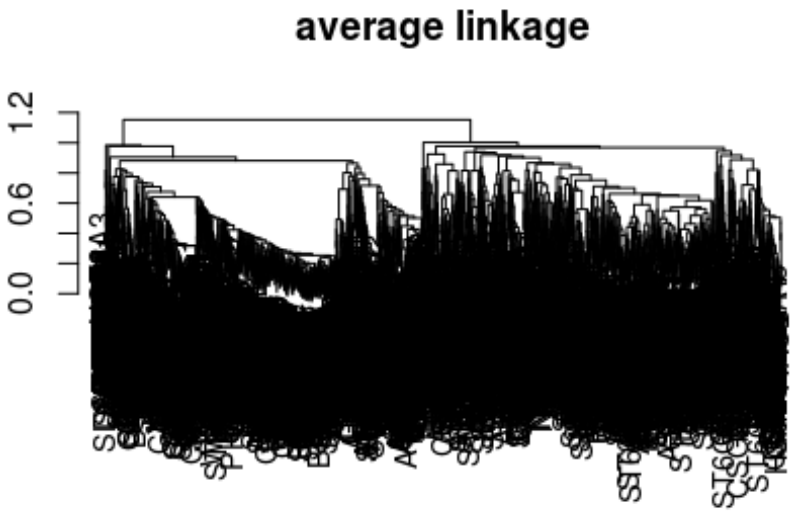
Base on the denrograms we conduced that Euclidean distance might not be suitable here. Both plots show one odd duckling (GAPDH) and the rest of the genes seem to be clustered rather peculiarly together. Therefore we decided to try using a correlation based distance for hc clustering. This time, like we did before, we used both complete and average linkage.

```
# calculate the correlation based distance
skin.tdf.cordist <- as.dist(1-cor(skin.df[,1:2561]))
# use complete linkage
hccor.gene.com <- hclust(skin.tdf.cordist,method="complete")
# use average linkage
hccor.gene.avg <- hclust(skin.tdf.cordist,method="average")
#create denrogram
plot(hccor.gene.com,main="complete linkage",xlab='',ylab='',cex=.9)
```



hclust (*, "complete")

```
plot(hccor.gene.avg,main="average linkage",xlab='',ylab='',cex=.9)
```



hclust (*, "average")

There are big differences in results from using different definitions of distance. Based on the denrograms we got, we concluded that using complete linkage is more suitable here. It prevents outliers from getting their own cluster and improves the overall readability of the dendrogram. This is because by using complete linage, there are less variances (differences) within the same cluster group.

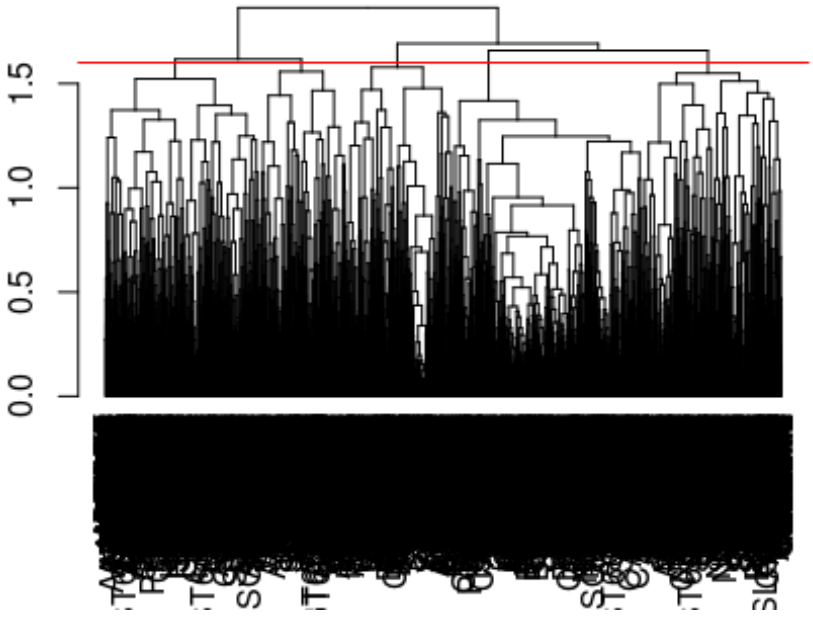
Therefore we used correlation based distance with complete linkage for further analysis. When looking at the plot we decided to cut the tree into 5 clusters, this is shown in the block of code below

```
 hc.gene=cutree(hccor.gene.com,5)
table(hc.gene)
```

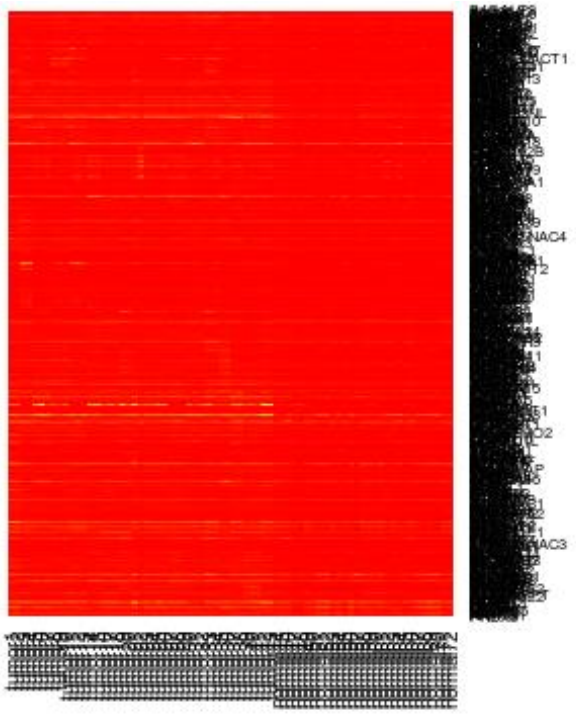
```
## hc.gene
##   1   2   3   4   5
## 527 589 717 403 325
```

In order to visualise the difference in gene expression between the 5 clusters we produced heatmaps. First we visualise the clusters we used by drawing a line in the plot where we cut the dendrogram. We create 3 heatmaps that can be compared with each other for difference in expression. This correlation of expression within clusters and the difference in expression between clusters is the reason the genes were assigned to different clusters. The last two heatmaps were not produced to increase the readability of this document.

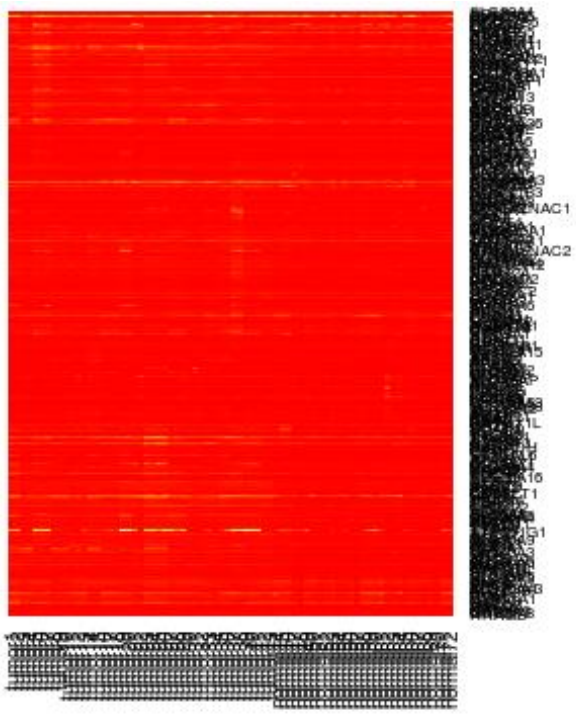
```
hcld.hccor <- as.dendrogram(hccor.gene.com)
plot(hcld.hccor, cex=.3)
abline(h=1.6, col="red")
```



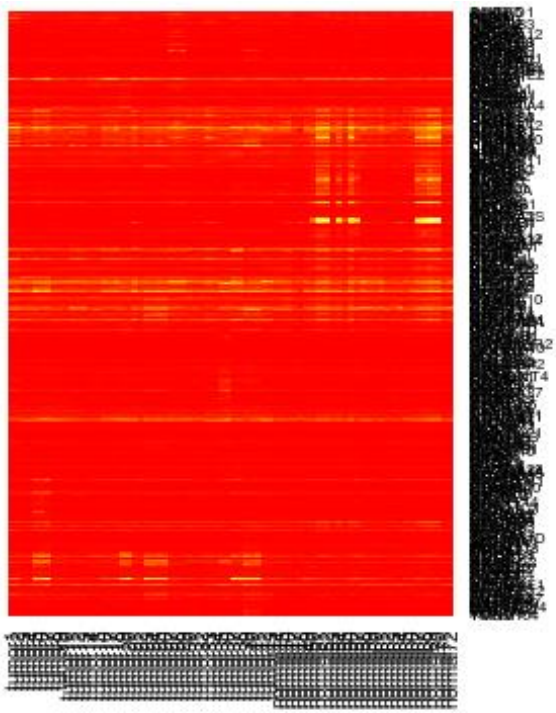
```
cuthcd = cut(hcld.hccor, h=1.6)
# Visualize the gene expression levels in the different samples in clusters 1
hc1 <- data.matrix(skin.tdf[unlist(cuthcd$lower[[1]]),])
heatmap(hc1,Rowv=NA,Colv=NA,scale='none',col=heat.colors(256),margins=c(5,10)
)
```



```
# Visualize the gene expression levels in the different samples in clusters 2
hc2 <- data.matrix(skin.tdf[unlist(cuthcd$lower[[2]]),])
heatmap(hc2,Rowv=NA,Colv=NA,scale='none',col=heat.colors(256),margins=c(5,10)
)
```



```r
# Visualize the gene expression levels in the different samples in clusters 3
hc3 <- data.matrix(skin.tdf[unlist(cuthcd$lower[[3]]),])
heatmap(hc3,Rowv=NA,Colv=NA,scale='none',col=heat.colors(256),margins=c(5,10)
)
```



Here we see recurrent patterns for lower gene expression represented by the yellow lines. Because of this patters it is clear why these genes were clustered together.

Task 2: Clustering the samples

We used the skin.df data frame to complete this task.

K-means clustering

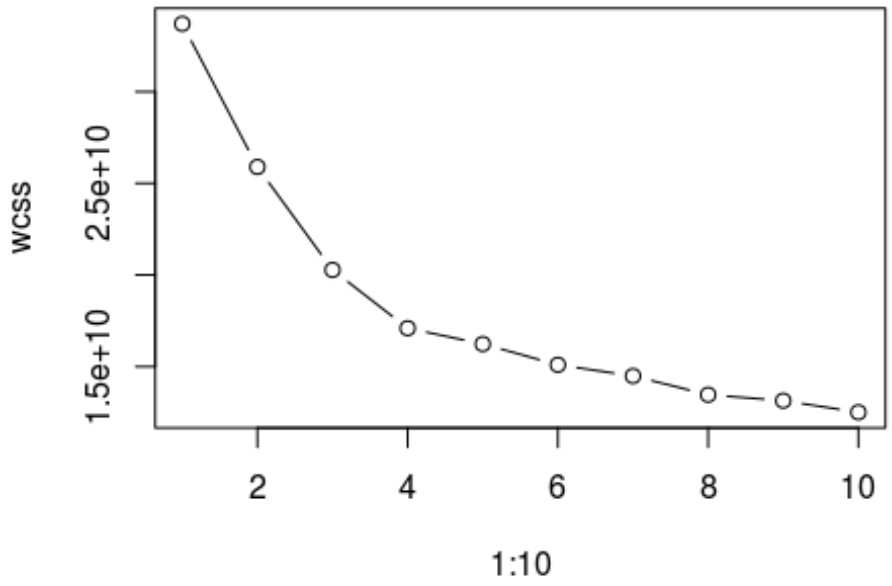
First, we used k-means clustering to separate the samples into two clusters.

```
dim(skin.df)

## [1]   72 2562

#create a data frame without the labels
skin.df.nolab <-skin.df[,1:2561]
km.sample <- kmeans(skin.df.nolab,2,nstart=20)
km.sample$cluster

##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [36] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 2 1 2 2 1 1 1 1 1 1 1 1 2 2 2 2
## [71] 1 1

table(km.sample$cluster)
```

```
##
##  1  2
## 63  9
```

The fact that the data is derived for two groups of tissue samples led us to believe that choosing k=2 would probably be the best choice. We used the Elbow Method to detected if this was true, and if not what the actual optimal k value would be.

```
set.seed(6)
wcss <- vector()
for (i in 1:10) wcss[i] <- sum(kmeans(skin.df.nolab,i)$withinss)
plot(1:10,wcss,type='b')
```



As the plot shows, the optimal k value is around 6. Then we preformed the k-means clustering again with k=6. We compared the results form clustering with k=6 to clustering with k=2 by calculating the tot.withinss value.

```
km.sample6 <- kmeans(skin.df.nolab,6,nstart=20)
km.sample6$cluster
```

```
##  [1] 4 4 3 3 2 2 2 6 6 6 3 3 3 6 6 6 6 6 2 2 4 6 2 2 2 2 4 4 4 6 6 6 6 6 6
## [36] 6 2 2 2 2 2 6 6 5 5 5 5 5 5 5 1 1 5 1 5 1 1 5 5 5 5 5 5 5 5 1 1 1 1 1
## [71] 5 5
```

```
# compare the 2 clusters and 6 clusters
km.sample$tot.withinss
```

```
## [1] 25894461717
```

```
km.sample6$tot.withinss

## [1] 14958470497

table(km.sample$cluster,km.sample6$cluster)

##
##      1  2  3  4  5  6
##   1  0 14  5  6 20 18
##   2  9  0  0  0  0  0
```

As the table shows, when comparing the two clustering methods, the smaller cluster (with 9 samples) stayed the same when changing k=2 to k=6. The bigger cluster was further divided into 4 sub clusters. Then we compared the clustering results obtained with the code above to each other whilst their original labels were assigned to them.

```
table(km.sample$cluster,skin.df$tissue)

##
##     normal tumor
##   1     20    43
##   2      9     0

table(km.sample6$cluster,skin.df$tissue)

##
##     normal tumor
##   1      9     0
##   2      0    14
##   3      0     5
##   4      0     6
##   5     20     0
##   6      0    18
```

When trying to cluster the samples into 3 clusters one cluster would contain all the tumour samples whilst the two other clusters hold all the normal samples. This could indicate that the normal group is divided in sub clusters.

```
km.sample3 <- kmeans(skin.df.nolab,3,nstart=20)
km.sample3$cluster

##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [36] 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 3 2 3 2 3 3 2 2 2 2 2 2 2 2 3 3 3 3
## [71] 2 2

table(km.sample3$cluster)
```

```
##
##  1  2  3
## 43 20  9
```

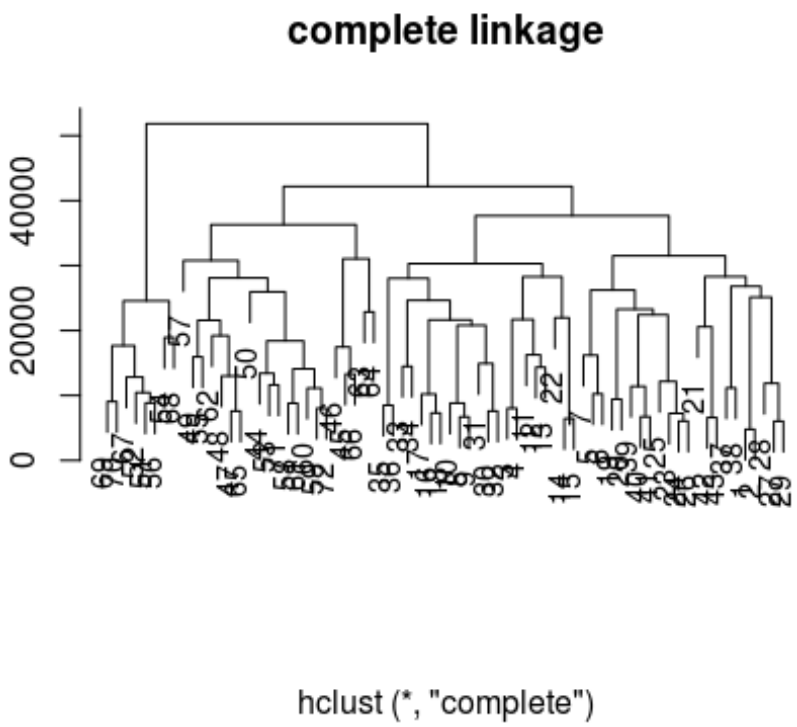
```
# compare clustering results with the known labels
table(km.sample3$cluster,skin.df$tissue)
```

```
##
##     normal tumor
##   1      0    43
##   2     20     0
##   3      9     0
```

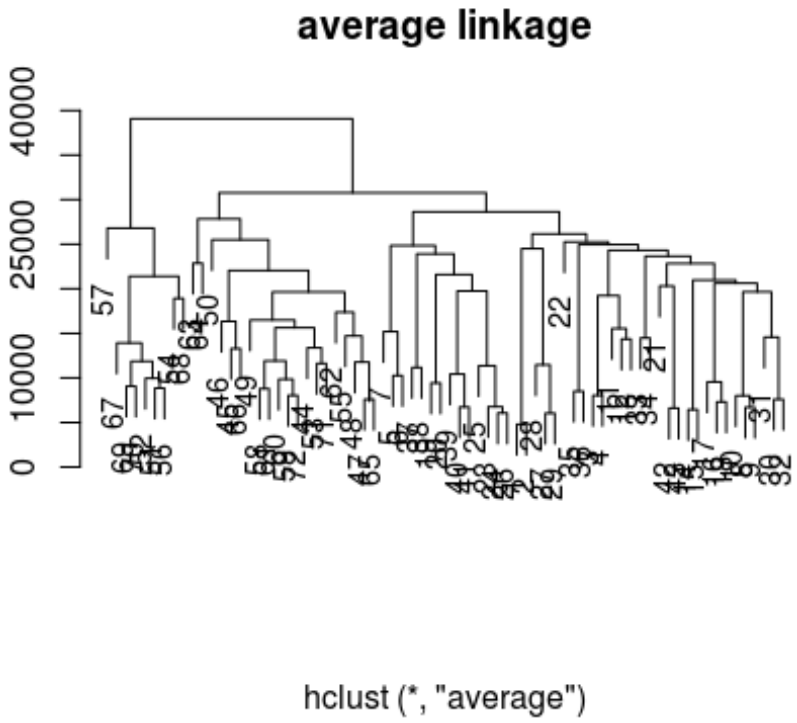
hierarchical clustering

First we performed hierarchical clustering of the samples based on Euclidean distance. We tried both complete and average linkage.

```
# use complete linkage
hce.sample.com <- hclust(dist(skin.df.nolab),method="complete")
# use average linkage
hce.sample.avg <- hclust(dist(skin.df.nolab),method="average")
#create denrogram
plot(hce.sample.com,main="complete linkage",xlab='',ylab='',cex=.9)
```

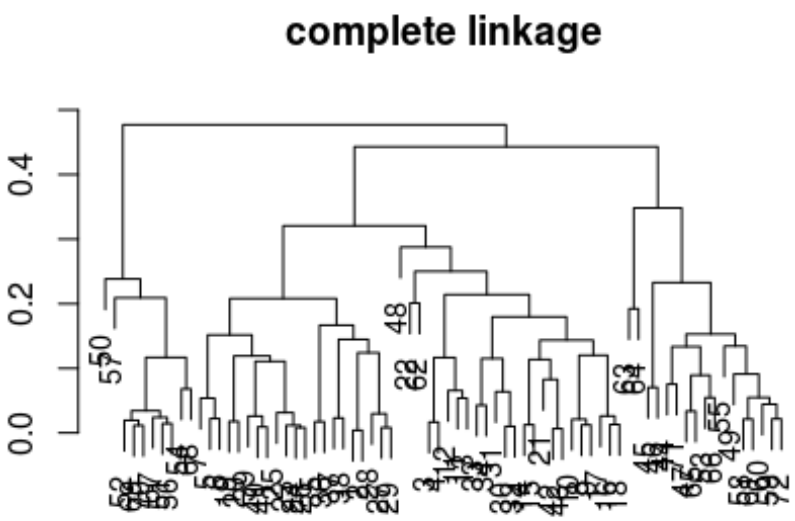



```
plot(hce.sample.avg,main="average linkage",xlab='',ylab='',cex=.9)
```

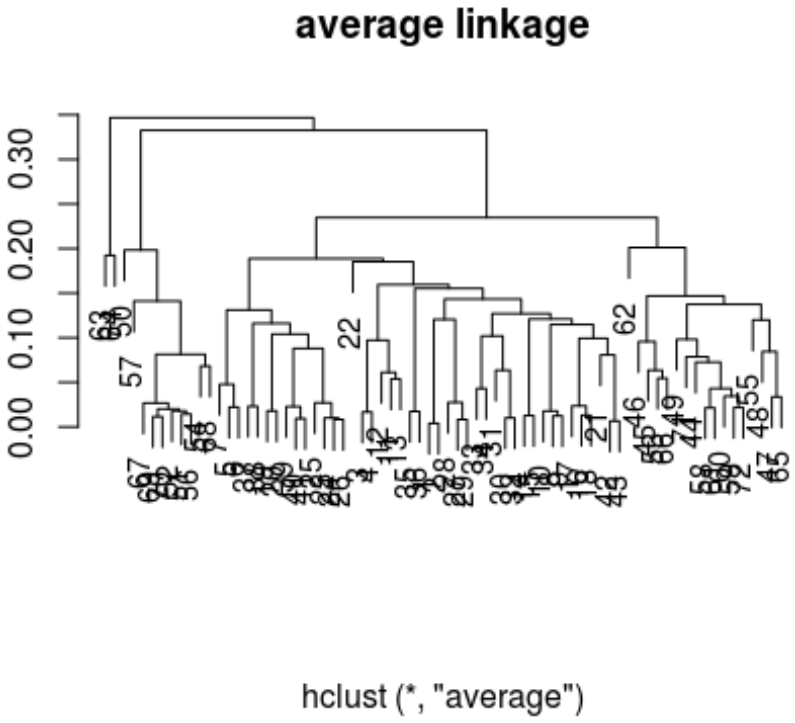


Then we used a correlation based distance for hc clustering. And this time, we also used both complete and average linkage.

```
# calculate the correlation based distance
skin.df.cordist <- as.dist(1-cor(t(skin.df.nolab)))
# use complete linkage
hccor.sample.com <- hclust(skin.df.cordist,method="complete")
# use average linkage
hccor.sample.avg <- hclust(skin.df.cordist,method="average")
#create denrogram
plot(hccor.sample.com,main="complete linkage",xlab='',ylab='',cex=.9)
```




```
plot(hccor.sample.avg,main="average linkage",xlab='',ylab='',cex=.9)
```



There are definitely differences in results from using different definitions of distance and different linkages. Based on these dendrograms it is difficult to concluded whether Euclidean distance is superior to correlation based distance. The superiority of either complete or average linkage is equally hard to determine. To make this a bit easier we we cut all 4 trees and compare the results in the following tables.

```
# Euclidean distance, complete linkage
euc.com=cutree(hce.sample.com,5)
table(euc.com,skin.df$tissue)

##
## euc.com normal tumor
##       1      0    22
##       2      0    21
##       3     16     0
##       4      5     0
##       5      8     0

# Euclidean distance, average linkage
euc.avg=cutree(hce.sample.avg,3)
table(euc.avg,skin.df$tissue)

##
## euc.avg normal tumor
##       1      0    43
##       2     20     0
##       3      9     0

# Correlation base distance, complete linkage
cor.com=cutree(hccor.sample.com,5)
table(cor.com,skin.df$tissue)

##
## cor.com normal tumor
##       1      0    21
##       2      2    22
##       3     15     0
##       4     10     0
##       5      2     0

# Correlation base distance, average linkage
cor.avg=cutree(hccor.sample.avg,4)
table(cor.avg,skin.df$tissue)

##
## cor.avg normal tumor
##       1      0    43
##       2     17     0
##       3     10     0
##       4      2     0
```

When comparing the two tables based on the Euclidian distance derived data it is hard to determine which one is better. The main difference is that when using different complete linkage the normal samples are divided into three clusters whilst using average linkage there are two clusters. This difference is not there because a smaller amount of clusters were chosen to create this table, this is visible through the difference in cluster size, amongst the two different linkage methods they don't match.

For using the correlation based distance derived data we are of the opinion that average linkage is the better option. This is the case because a few normal samples seem to be added to clusters that otherwise only contain tumour samples. When you look at the second row of the third table you'll see that there is one cluster that contains 22 tumour samples and two normal samples.

clustering samples, compare clustering results obtained with different approaches with the known labels for these samples. What do you learn from this? Can you say something on how many clusters there are in this dataset?