Report: Clustering, Tissue Samples and Gene-Expression levels

Activity 1

Introduction

This activity explores clustering tissue samples based on their gene-expression levels. This is relevant to analyze due to its significance in cancer discovery. Clustering is used to partition the tissue samples into distinct groups, in this case, the clustering is based on their gene-expression levels. The data set used in Activity 1 is from the file "golub-1999-v1_database.arff" which is from a study that contains 72 tissue samples diagnosed with leukemia. The expression levels used to cluster the tissue samples are from 1868 selected genes.

Firstly reading the data into a data frame in R using the function read.arff() found in the library 'foreign' can be done as follows;

```
library(foreign)
Golub_Data <- read.arff("golub-1999-v1_database.arff")
Golub_Data <- data.frame(Golub_Data)
dim.data.frame(Golub_Data)
```

```
## [1]   72 1869
```

The resulting data frame has 1869 columns rather than the required 1868 selected genes. This rightmost column is the labels (either 1 or 2) indicating the type of leukemia associated with each of the samples. This last column isn't used in the clustering and will only be used for external assessment of the results post clustering. So in order to remove it for the clustering, it must be removed and stored separately from the remaining data frame.

Question 1

The rightmost column containing the class labels from the data can be stored separately from the remaining data frame as follows:

```
#the rightmost column is assigned to the variable Classes
Classes <- as.matrix(Golub_Data[,1869])

#summary(Classes)

#The rightmost column is then removed fro the Golub_Data
Golub_Data <- Golub_Data[,-1869]

#dim(Golub_Data)
```

Now there is a data frame 'Classes' with the labels indicating the type of leukemia associated with each of the samples and a data frame 'Golub_Data' with 1868 selected genes of 72 observations of tissue samples with leukemia.

Question 2

Now using the Golub_Data 72 x 1868 data frame the 72 x 72 matrix can be generated with the Euclidean distances between observations (tissue samples). This matrix has the distances between tissue samples according to their 1868 expression levels. This matrix is of type dist() and can be achieved using the following code:

```r
#euclidean distance is calculated using dist function with method = 'euclidean'
euclid.dist <- dist(Golub_Data, method = 'euclidean')
```

Question 3

The Single-Linkage Algorithm can be used to cluster tissue samples with leukemia based on their gene expression levels, we can then use a dendrogram to visually show the hierarchical relationship between the tissue samples.

For the single-linkage Algorithm, the distance between two clusters of observations is defined as the smallest pairwise dissimilarity between a observation in the first cluster and a observation in the second cluster. The euclidean distance matrix found in question 2, will be used to define the distances between pairwise observations for which the single-linkage Algorithm will use to evaluate the dissimilarity between the clusters.

The function hclust() can be used to perform Hierarchical Clustering, setting the argument method = "single" will perform the single-linkage Algorithm which will iteratively join the two most similar clusters (cluster closest together) and continue until just a single cluster

```r
#run single-linkage Algorithm with the euclidean distances and method = "single"
SLA <- hclust(euclid.dist, method = "single")

#plot the dendrogram of the single-linking Algorithm, cex = 0.53
#changes the label size of the leaves
plot(SLA, main = "Single-Linkage", xlab = "", sub = "Figure 1: Single Linkage
     Dendrogram, Euclidean Distances", hang = -1, cex = 0.53)
```

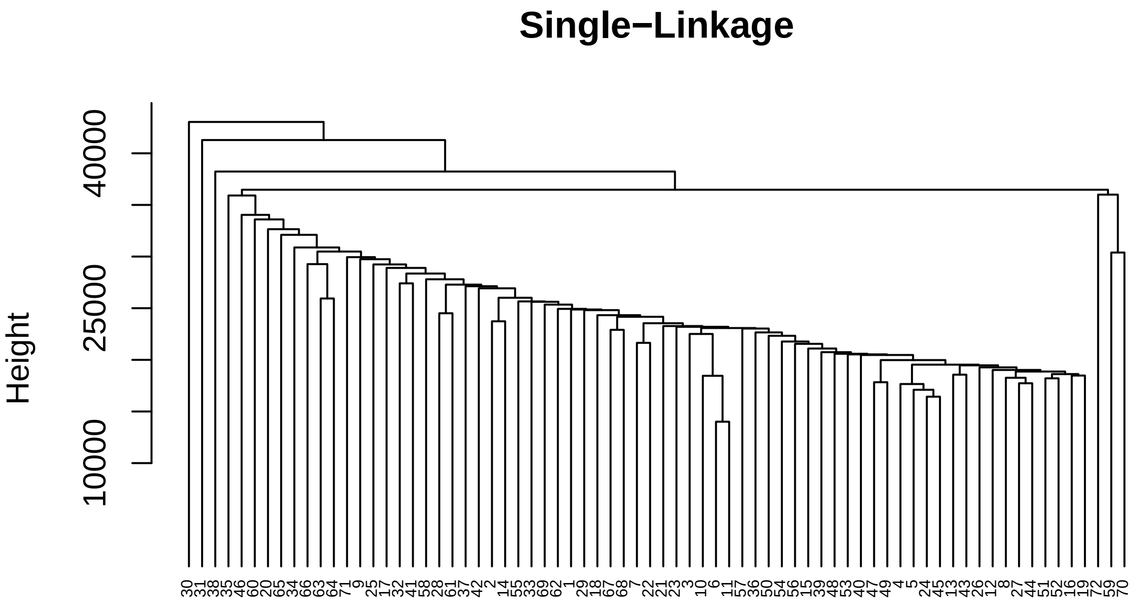


Figure 1: Single Linkage
Dendrogram, Euclidean Distances

Each leaf of the dendrogram represents one of the 72 tissue samples which are labeled by the numbers found on the x-axis. The y-axis of the dendrogram is the smallest pairwise dissimilarity between two clusters for them to merge. The height (on the y-axis) at which two clusters merge can be determined from the horizontal lines on the dendrogram and their corresponding height value on the y-axis. The higher this horizontal line is on the dendrogram, the higher dissimilarity of the clusters.

The Single-Linkage Algorithm defines similarity of clustered tissue samples by the distance between the two clusters. The Single-Linkage Algorithm defines this distance by taking the smallest pairwise distances between observations within the two clusters. Tissue samples that combine together at the bottom of the dendrogram have very similar gene-expression levels and form larger clusters. For example the tissue samples labeled 6 and 11 have very similar gene-expression levels since they are clustered together at the bottom of the dendrogram at a height of approximately 13500. Also note that the height at which tissue sample 30 merged with the other clustered tissue samples is very large (at the top of the dendrogram) meaning there is a large dissimilarity between observation 30 and the other clustered tissue samples. The height at which the tissue sample 30 merged with the other tissue samples is approximately 42000.

To identify clusters, a horizontal cut across the dendrogram is made. The distinct sets of observations beneath the cut are clusters. This graph doesn't suggest any prominent clustering since individual observations are constantly being merged with a larger cluster rather than sub-clusters of multiple observations merging together. Their is no occurrence of dense clustering at low heights which indicates the Single-Linkage Algorithm has been unable to identify a lot of observations that are close to each other at these low heights. For well defined clusters, we would also expect to see long vertical lines as the number of clusters decreases which suggests there is a large separation between the main clusters. This is not occuring in our dendrogram. For the majority of the dendrogram, tissue samples are merged one-by-one into a larger cluster rather than forming smaller sub-clusters, this further emphasizes the lack of clustering for the single linkage algorithm.

The dendrogram does suggest that observation 30,31 and 38 are outliers in the data since they merge at the very top of the dendrogram with the larger cluster. But, the Single-Linkage Algorithm has not clearly clustered the data into prominent groupings meaning these observations indicated as outliers could just be caused by the algorithm not going a good job a clustering.

This lack of distinct clustering may be caused by Single-Linkage Algorithm being very sensitive to noisy data but since we are doing unsupervised analysis, we cannot clearly determine if we have noisy data. The Single-Linkage Algorithm has caused chain like linkage rather than clear distinct clusters.

Question 4

In the complete-linkage Algorithm, the distance between two clusters of observations is defined as the largest pairwise distance between a observation in the first cluster and a observation in the second cluster. The euclidean distance matrix found in question 2, will be used to define the distances between pairwise observations for which the complete-linkage Algorithm will use to evaluate the dissimilarity between clusters.

The function hclust() can be used to perform Hierarchical Clustering, setting the argument method = "complete" will perform the complete-linkage Algorithm which will iteratively join the two most similar clusters and continue until just a single cluster

```r
#run complete-linkage Algorithm with the euclidean distances and method = "complete"
CLA <- hclust(euclid.dist, method = "complete")

#plot the dendrogram of the complete-linking Algorithm, cex = 0.53 changes
#the label size of the leaves
plot(CLA, main = "Complete-Linkage", xlab = "", sub = "Figure 2: Complete Linkage Algorithm,
     euclidean distances", hang = -1, cex = 0.53)
```

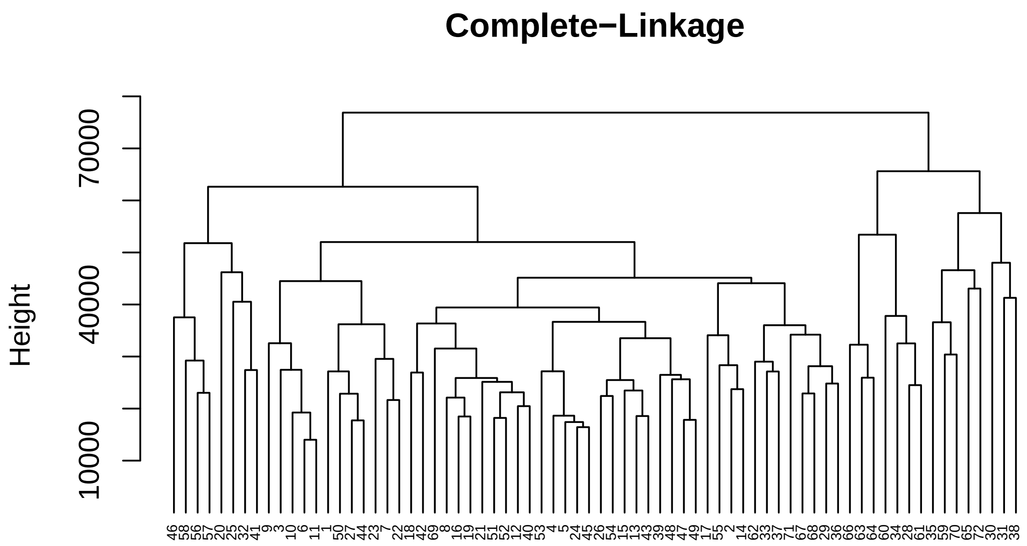


Figure 2: Complete Linkage Algorithm,
euclidean distances

The y-axis labeled 'height' is the measure of the smallest pairwise dissimilarity between the two clusters for them to merge. This dissimilarity is determined by the largest pairwise distance between an observation in each cluster. Tissue samples that combine together at the bottom of the tree have very similar gene-expression levels and form larger clusters. For example the tissue samples labeled 6 and 11 have very similar gene-expression levels since they are clustered together at the bottom of the tree at a height approximately 15000. Conclusions about the similarity of clusters is based on the location of the horizontal line on the vertical axis where the two clusters combine to make a larger cluster. The higher this horizontal line is on the dendrogram, the higher dissimilarity of the clusters.

To identify clusters, a horizontal cut across the dendrogram is made. The distinct sets of observations beneath the cut are clusters. This graph does suggest a clustering in the data, there are two distinct clustering of tissue samples when making a horizontal line on the dendrogram at height 60000. Reading left to right of the labels found on the x-axis, the first cluster contains samples 46 - 36 and the second is the cluster 66 - 38 which are then merged together as one at a height of 75000. Looking at all the mergers occurring at small distances, the compactness of the merges suggests that they are close together and are the gene-expression levels are determined to be similar for a lot of individual observations. Then we see that when the sub-clusters merge this occurs at high distances indicating that we have separated clusters. This compactness and long vertical lines as the number of clusters decreases indicates that we have compact and separated clusters making them clearly identifiable.

The data could also be clustered into four district clusters when a horizontal line at height 50000 is made. The first cluster (when reading left to right on the x-axis) is tissue samples 46 to 41 of the dendrogram, the second from 9 to 36, the third cluster then goes from 66 to 61 of the dendrogram and the forth from 35 to 38. Note in this dendrogram, there are no individual tissue samples merging last with a large cluster at high heights, this means the dendrogram does not contain any outliers.

In each situation of k = 2 or k = 4 a partition can be made with clear clusters. Note, there is improvements in the clustering that can be made which would make the choice of partition (k = 2 or k = 4) a clearer decision. ### Question 5

In the average-linkage Algorithm, the dissimilarity between two clusters is defined as the average pairwise distances between single observations in each cluster.

The function hclust() can be used to perform Hierarchical Clustering, setting the argument method = “average” will perform the average-linkage Algorithm which will iteratively at each stage, join the two most similar clusters (clusters closest together) and continuing until just a single cluster

```r
#run average-linkage Algorithm with the euclidean distances and method = "average"
ALA <- hclust(euclid.dist, method = "average")

#plot the dendrogram of the average-linking Algorithm,
#cex =0.53 changes the label types of the leaves
plot(ALA, main = "Average-Linkage", xlab = "", sub = "Figure 3: Average Linkage
     Dendrogram, euclidean distances", hang = -1, cex = 0.53)
```

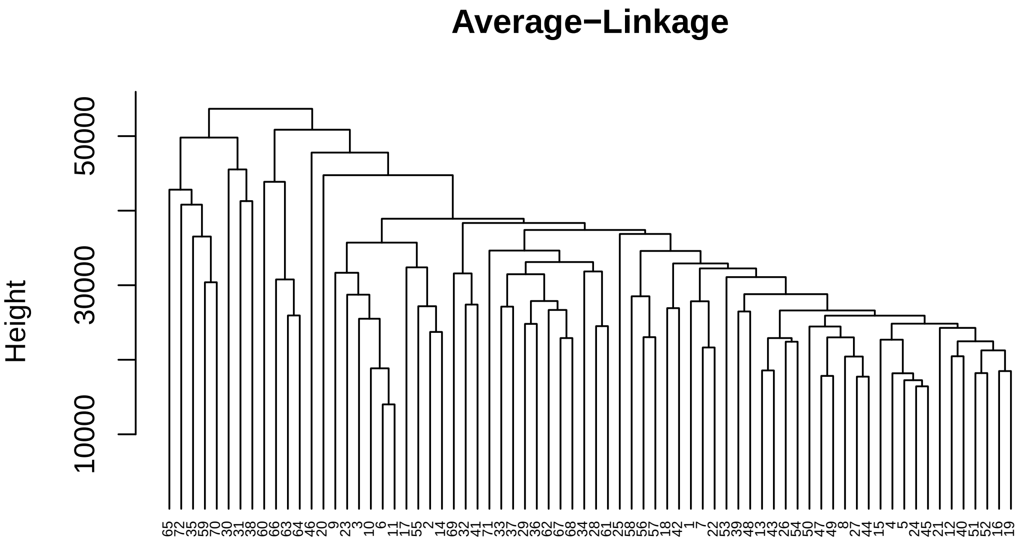


Figure 3: Average Linkage
Dendrogram, euclidean distances

Again we know that tissue samples that combine together at the bottom of the tree have very similar gene-expression levels and form clusters. The Average-Linkage Algorithm defines this distance by taking the average pairwise distances between observations within the two clusters. For example the tissue samples labeled 6 and 11 have very similar gene-expression levels since they clustered together at the bottom of the tree at a height approximately 15000.

Conclusions about the similarity of clusters is based on the location of the horizontal line on the vertical axis where the two clusters combine to make a larger cluster. The higher this horizontal line is on the dendrogram, the higher dissimilarity of the clusters.

For the Average-Linkage dendrogram it is quite hard to clearly identify well defined clusters with no particular partition being immediately clear. The partitioning is not clear from the dendrogram since there are no long vertical lines indicating large height difference between merges of the larger clusters. This means that the larger clusters, for example at a partition at k = 2 didn't stay as two unique clusters for long before they merged into one cluster. This indicates that the Average-Linkage algorithm didn't do well at clustering the data into district separated groups of tissue samples. At the very small heights, there isn't many observations merging meaning there is a small amount of observations that the algorithm was able to pick up to naturally cluster together. These two factors indicate that the algorithm was unable to form compact and separated clusters making the dendrogram partition unclear to parition.

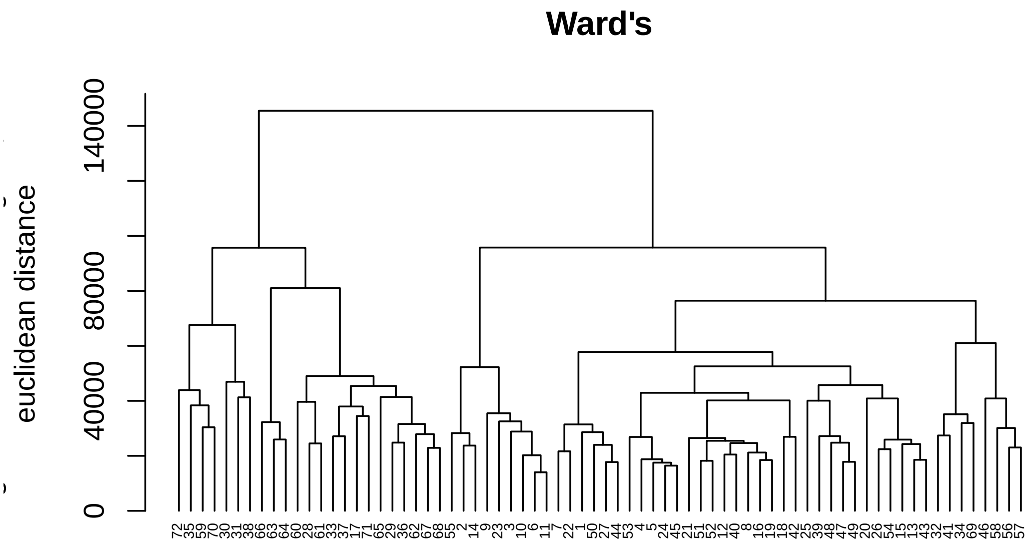
Question 6

In the Ward's method, the distance between two clusters of observations is defined as the increment that would be observed in the total within variance when two clusters are merged. Note that if two clusters are merged, then the variance of the merged group isn't smaller than the variance of the two separate clusters. Ward's method will always merge the pair of clusters that minimizes the increment observed in the total within variance. The euclidean distance matrix found in question 2, will be used to define the distances between pairwise observations for which the Ward's method will use to evaluate the dissimilarity between the clusters.

The function hclust() can be used to perform Hierarchical Clustering, setting the argument method = “ward.D2” will perform the Ward’s method which will iteratively join the two most similar clusters (the clusters that will minimize the increment in the total within variance when merged) and continue until just a single cluster.

```
#Compute the wards clustering using hclust with method = "ward.D2"
Ward <- hclust(euclid.dist, method = "ward.D2")

#Plot the corrosponding dendrogram for wards algorithm clustering
plot(Ward, main = "Ward's", xlab = "", ylab = "Figure 4: Ward's Dendrogram,
     euclidean distance", sub = "", hang = -1, cex = 0.53)
```



Firstly noting that again this dendrogram indicates that the tissue samples labeled 6 and 11 have very similar gene-expression levels since they clustered together at the bottom of the tree at a height approximately 15000.

To identify clusters, a horizontal cut across the dendrogram is made. The distinct sets of observations beneath the cut are clusters. For the Ward’s dendrogram it is easy to identify well defined clusters and a immediate partition is clear. This dendrogram indicates there is a clustering at k = 2 clusters with two long vertical lines from a height just under 100000 when the two large clusters first formed to above 140000 when the two large clusters finally combine to form one large cluster. This long height range before the clusters merge indicates these two district clusters are far apart and have a high dissimilarity. We have a lot of observations merging at smaller heights which suggests we have natural clusters with observations that are very close together.

There are two distinct clusters that can be drawn from this dendrogram by making a cut horizontally at a height of 120000. The first cluster, reading left to right on the x-axis, is made up of observations 72 to 68 and the second cluster is made up of observations 55 to 57. There are no clear outlier’s from the dendrogram and the observations are clearly divided into two distinct clusters. These two clusters merge at a height above 140000 to one complete cluster. This large height indicates that these clusters are quite distanced from each other meaning they will form a large within variance when the are merged (Ward’s Algorithm defines two most similar clusters as the clusters that will minimize the increment in the total within variance when merged).

Question 7

Each algorithm attempts to partition the tissue samples into clusters with similar gene-expression levels. The number of clusters initially is equal to the number of tissue samples (72) which is then partitioned incrementally into a single cluster (1). The algorithms attempt to always merge clusters that have similar gene-expression levels which is determined, in our case, by using the euclidean distance between each observation to calculate the distance between clusters. Each algorithm (single, complete, average and ward) measures this distance between clusters differently, and hence will produce different clustering results and different dendrograms. How the algorithms define distance affects the clearness of the clustering that can be seen in the dendrograms.

The Ward's and Complete-Linkage Algorithm have quite large distances separating the prominent clusters as indicated by the height at which those clusters finally merge on the dendrogram. Also the prominent clusters stayed as distinct clusters for a large value range of height (indicated by the vertical lines) also helping the partition of clusters be clear. Taking into account the different axis for each algorithm, this is not the case for that of the single and average algorithms.

In the Single-Linkage and Average-Linkage Algorithms, the last two cluster merges occurred quickly after they are formed making the observations of the clusters more difficult. The density at lower heights is much less than that found in Complete-Linkage and Wards algorithms suggesting that the Single and Average linkage algorithms are struggling to identify natural clusters in the data. Also when the smaller clusters finally merge into larger clusters, for both the single linkage and average linkage algorithms, this occurs right after their formation which suggests the algorithms cannot detect that the clusters are far apart and distinct like what is achieved by wards and complete.

Also, the merging of tissue samples in the Single-Linkage and Average-Linkage is occurring for the majority, one-by-one rather than whole cluster with multiple observations merging with each other larger cluster. This also makes the clusters generated by both Single and Average-Linkage dendrograms not as clear as those from Ward's and Complete-Linkage. The reasons for the disparity in prominent clusters for each algorithm is due to each algorithms benefits and drawbacks in regards to how they define dissimilarity.

Single-Linkage is not robust to noisy data and can result in extended clusters with trails for which single observations are combined one at a time. This may create so called 'bridges' between different clusters causing undesired linkage. This may be what is occurring in the Single-Linkage Algorithms clustering of tissue samples. The Single-Linkage Algorithm is merging observations into clusters almost one-by-one, which may be caused by this bridging affect between clusters caused by noisy data. It is worth remarking that Single-Linkage Algorithms are capable of detecting clusters with arbitrary shapes and can be beneficial to use in comparison to the other algorithms, but in our situation, this is not the case.

Complete-Linkage does prevent this 'bridging' that occurs with the Single-Linkage Algorithm, but is also sensitive to outliers. This can distort the largest distance between observations and may cause problems with clustering. In our case, Complete-Linkage is one of the best dendrograms in terms of clear partitions of tissue samples, which may indicate that it isn't being dramatically affected by outliers.

Note that Average-Linkage tends to be less sensitive to noisy data and outlier's in comparison to the Single-Linkage and Complete-Linkage algorithms. In our situation, the clustering of tissue samples has not improved with the use of the Average-Linkage Algorithm.

Ward's method tends to result in nicely balanced clusters and is often used to cluster observations. Ward's method does well at separating clusters if there is noise between them but does tend to favor volumetric, globular clusters.

Overall, both complete linkage Algorithm and Ward's are the best algorithms to partition the tissue samples into distinct clusters with Single-Linkage and Average-Linkage struggling to make clear clusters.

Question 8

The algorithms that were chosen that generated clear clustering were the Complete-Linkage Algorithm and the Ward's method. The following code adds the class labels to the dendrograms:

```
#plot the dendrogram of the complete-linking Algorithm, with class labels
plot(CLA, main = "Complete-Linkage", xlab = "", sub = "Figure 5: Complete Linkage Dendrogram with
     class labels", hang = -1, labels = Classes, cex = 0.53)
```

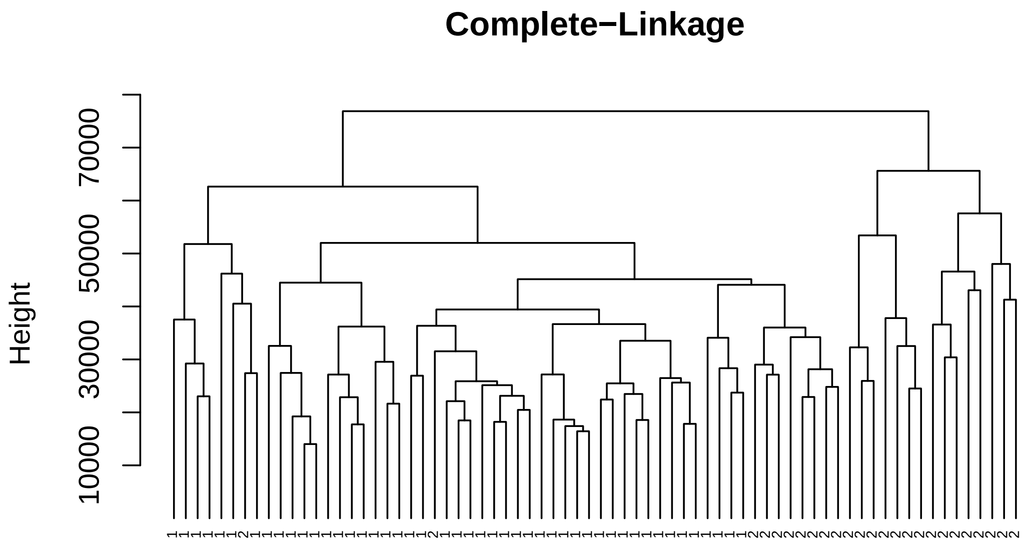


Figure 5: Complete Linkage Dendrogram with
class labels

```
#plot the dendrogram of the Ward's method with class labels added
plot(Ward, main = "Ward's", xlab = "", ylab = "", sub = "Figure 6: Ward's Dendrogram with
     Class labels", hang = -1, cex = 0.53, labels = Classes)
```

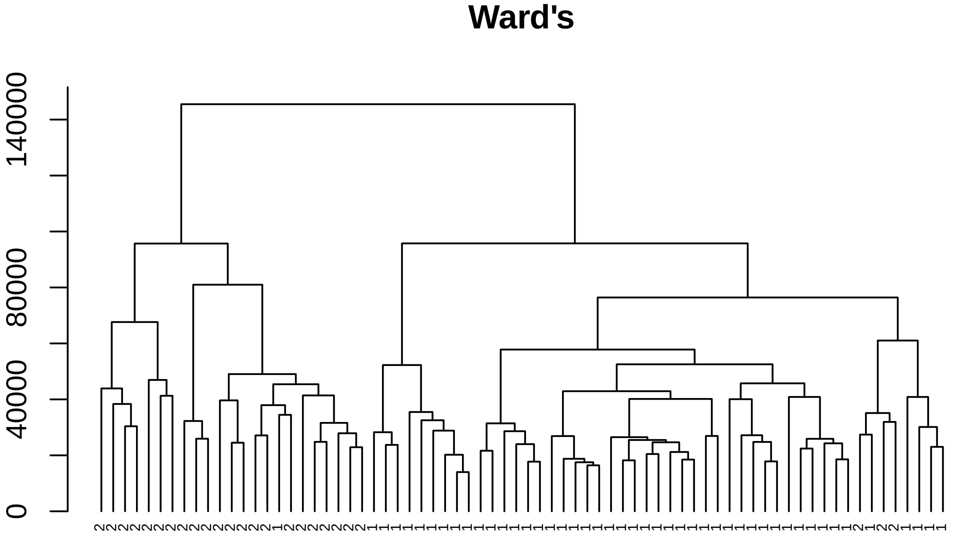


Figure 6: Ward's Dendrogram with
Class labels

As seen by the class labels on the dendrogram, there seems to be some prominent clusters in the dendrograms that correspond approximately to the classes 1 and 2. Both the Complete-Linkage Algorithm and the Ward's method have approximately clustered the two sub-types of leukemia successfully in a completely unsupervised way. This is determined by using the class labels to do an external assessment of the results. By just considering the partition of the dendrogram that has two clusters (k = 2), the following can be observed for each dendrogram:

The Complete-Linkage Algorithm has a left cluster that clearly has a predominant class label of 1 with 48 observations with the class label '1' and 10 observations with class labels '2'. The right cluster, is made up of all observations with class labels '2'. Note also the sub cluster in the left main cluster that is consistent of all observations of class label '2'. Having this occur within the cluster with predominant class label '1', may be an error in the clustering caused by the way the Complete-Linkage Algorithm defines distance. Overall, the dendrogram indicates that the left cluster corresponds to the leukemia type associated with class '1' and the right cluster corresponds to the leukemia type associated with class '2'.

The Ward's Algorithm, again considering two clusters, has some prominent clusters in the dendrograms that correspond approximately to the classes 1 and 2. The left cluster, is made up of 23 observations with 22 of them being of class label '2' and 1 of class label '1'. That one observation with class label '1' could be a local outlier, but this is not obvious from the dendrogram. The right cluster is made up of 49 observations with 3 of them from class label '2' and the remaining 46 with class label '1'. Again the observations labeled '2' in the right cluster could be outliers, or drawbacks from the way Wards defines distances between clusters. Note that the right cluster is made up of more observations than the left cluster, which makes sense since there was 47 tissue samples of sub-type '1' and 25 tissue samples of sub-type '2'. If the algorithm is doing well at clustering the tissue samples, then the cluster sizes will follow these values. Overall, the Ward's dendrogram indicates that the left cluster, corresponds to the leukemia type associated with class '2' and the right cluster corresponds to the leukemia type associated with class '1'.

It is worth noting that Complete-Linkage Algorithm incorrectly clustered 10 observations in total with 10 observations with class labels '2' being in the cluster that corresponded approximately to class label '1'. Whilst the Ward's Algorithm only clustered 4 tissue samples incorrectly. This is directly related to how the algorithms define distance and how this works with the natural structure of the data. Overall, the Ward's

Algorithm has clearer clusters when analysing the dendrogram unsupervised and also has less error when a external assessment of the dendrogram is done with the class labels.

Question 9

It is unclear that normalization of data in the context of unsupervised clustering improves performance. But, it has the benefit of possibly preventing variables with wider value ranges dominating decisions. It is a often occurrence that the data is analyzed both when normalised and non-normalised. Normalising the data may prevent variables with wide value ranges from dominating the clustering distance computations but can have the drawback of removing natural differences in variance and cluster shapes.

In the context of this data, we will do a z-score normalisation since we are computing the euclidean distance and compare whether noramlisation has helped with the clustering of the data. The function scale() can be used to scale the data as follows:

```r
#dim(Golub_Data)

Golub_Data_Norm <- scale(Golub_Data)

#mean(Golub_Data_Norm[, 2]) = 0

#sd(Golub_Data_Norm[, 2]) = 1

euclid.dist.normalized <- dist(Golub_Data_Norm, method = 'euclidean')
```

Question 9.a

Using this new normalised distance matrix each dendrogram from items 3-6 can be recomputed. We can compute the dendrogram for the normalised data Single-Linkage Algorithm as follows;

```r
#run single-linkage Algorithm with the euclidean distances and method = "single"
SLA.normal <- hclust(euclid.dist.normalized, method = "single")

#plot the dendrogram of the single-linking Algorithm,
#cex = 0.53 changes the label size of the leaves
plot(SLA.normal, main = "Single-Linkage", xlab = "", sub = "Figure 7: Single
     Linkage Dendrogram wtih normalised data", hang = -1, cex = 0.53)
```

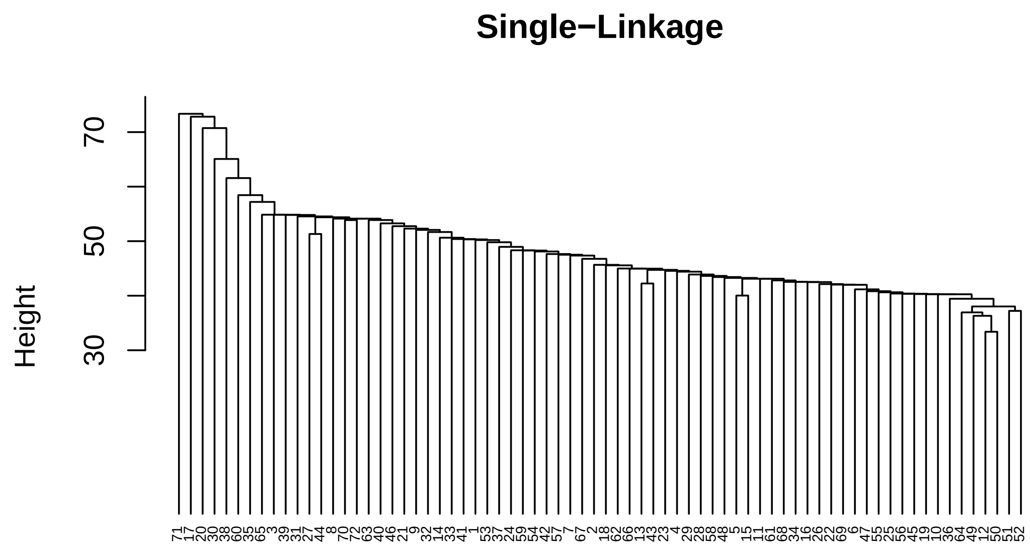


Figure 7: Single
Linkage Dendrogram wtih normalised data

We can now compare the two plots side-by-side to see if there was an improvement in the partitions of tissue samples when the data is normalised;

```r
#plotting the normalised data SLA dendrogram
plot(SLA.normal, main = "Single-Linkage Normalised", xlab = "",
  sub = "Figure 7: Single Linkage Dendrogram wtih normalised data", hang = -1, cex = 0.53)

#plotting the non-scaled data SLA dendrogram
plot(SLA, main = "Single-Linkage", xlab = "", sub = "Figure 8:
     Single Linkage Dendrogram with original data", hang = -1, cex = 0.53)
```

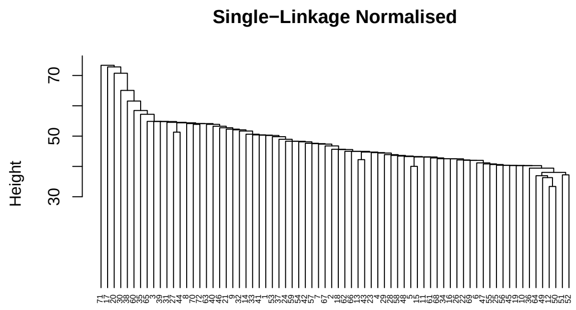


Figure 7: Single Linkage Dendrogram wtih normalised data

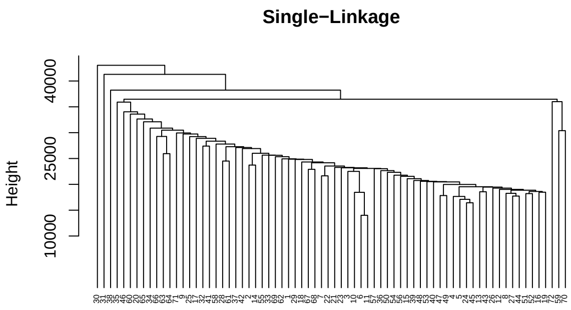


Figure 8:
Single Linkage Dendrogram with original data

Normalisation of the data didn’t improve the interpretability of the dendrogram with both the normalised and original dendrograms being difficult to form distinct clear clusters. The original dendrogram does have some evidence of sub-clusters at lower heights and normalising the data removes the majority of this. The normalised dendrogram has almost all observations merging one at a time so has actually reduced the clustering present. Similarly with the un-normalised data, their is no high density region of the dendrogram at lower heights. This indicates that the single linkage algorithm is again struggling to detect tissue samples that naturally have similar gene expression levels. Outliers are also hard to identify in the normalised dendrogram since the clustering into distinct sets of tissue samples is difficult to identify. The normalised dendrogram has all the same problems as the original dendrogram as discussed earlier.

We can compute the dendrogram for the normalised data Complete-Linkage Algorithm as follows;

```
#run complete-linkage Algorithm with the euclidean distances and method = "complete"
CLA.normal <- hclust(euclid.dist.normalized, method = "complete")

#plot the dendrogram of the complete-linking Algorithm, cex = 0.53 changes the label size of the leaves
plot(CLA.normal, main = "Complete-Linkage", xlab = "", sub = "Figure 9: Complete Linkage
     Dendrogram with normalised data", hang = -1, cex = 0.53)
```

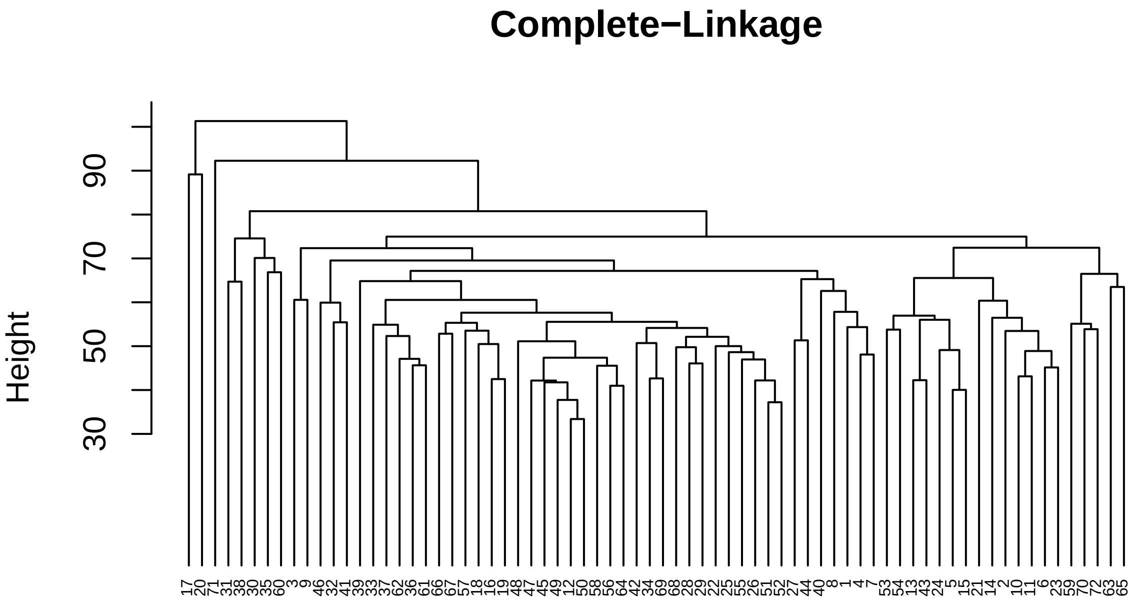


Figure 9: Complete Linkage
Dendrogram with normalised data

We can now compare the two plots side-by-side to see if there was an improvement in the dendrograms intepretability when the data is normalised;

```
#plotting normalised data CLA dendrogram
plot(CLA.normal, main = "Complete-Linkage Normalised", xlab = "", sub = "Figure 9: Complete Linkage
     Dendrogram with normalised data", hang = -1, cex = 0.53)

#plotting the non-scaled data CLA dendrogram
plot(CLA, main = "Complete-Linkage", xlab = "", sub = "Figure 10: Complete linkage
     dendrogram with original data", hang = -1, cex = 0.53)
```

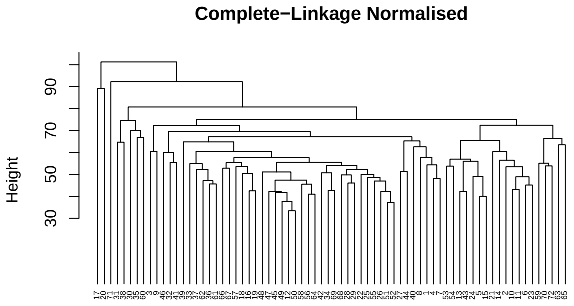


Figure 9: Complete Linkage
Dendrogram with normalised data

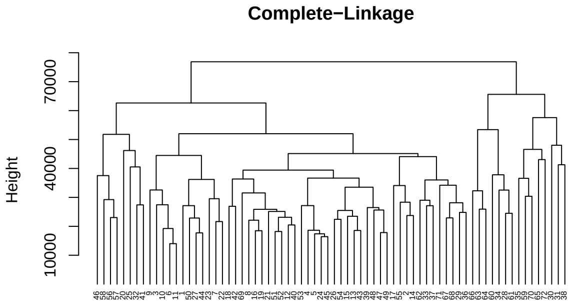


Figure 10: Complete linkage
dendrogram with original data

Again, the normalisation of the data has not improved the interpretability of the Complete-Linkage Algorithm dendrogram and is actually making the clustering of the data more unclear than the original. On the left, we can see the Complete-Linkage Algorithm with the normalised data, this dendrogram has clusters that can be found by taking partitions of the data at certain height levels. These partitions are not as clear than taking a partition with the same cluster size of the right dendrogram with the un-normalised data. For example, taking a partition at k = 2 forms clear clusters when using the original dendrogram to the right. When making the same partition at k = 2, we get clearly imbalanced clusters with one having only 2 observations.

The density of the cluster merges higher up in the dendrogram for the normalised data. This indicates that the Complete-Linkage algorithm is struggling to identify tissue samples with similar gene expression levels when the data is normalised. It is clear that normalisation of the data has not improved the amount of clear partitions that can be seen from the dendrogram.

We can plot the dendrogram of the normalised data to make a comparison again with the Average-Linkage Algorithm:

```
#run average-linkage Algorithm with the euclidean distances and method = "average"
ALA.normal <- hclust(euclid.dist.normalized, method = "average")

#plot the dendrogram of the average-linking Algorithm, cex = 0.53 changes the label types of the leaves
plot(ALA.normal, main = "Average-Linkage", xlab = "", sub = "Figure 11: Average Linkage
     Dendrogram with normalised data", hang = -1, cex = 0.53)
```

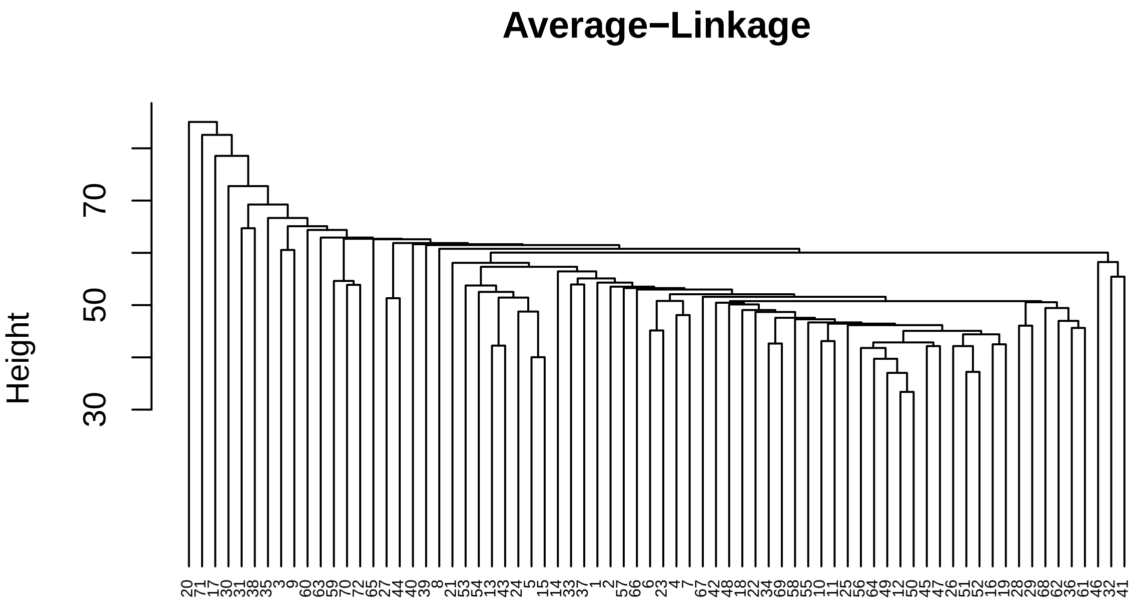


Figure 11: Average Linkage
Dendrogram with normalised data

Then the normalised data dendrogram can be plotted side-by-side with the original data dendrogram for the Average-Linkage Algorithm:

```
#plotting normalised data CLA dendrogram
plot(ALA.normal, main = "Average-Linkage Normalised", xlab = "Figure 11: Average Linkage
     Dendrogram with normalised data", sub = "", hang = -1, cex = 0.53)

#plotting the non-scaled data CLA dendrogram
plot(ALA, main = "Average-Linkage", xlab = "", sub = "Figure 12: Average Linkage
     Dendrogram with original data", hang = -1, cex = 0.53)
```

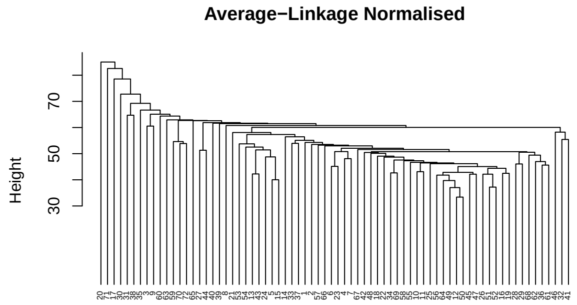


Figure 11: Average Linkage
Dendrogram with normalised data

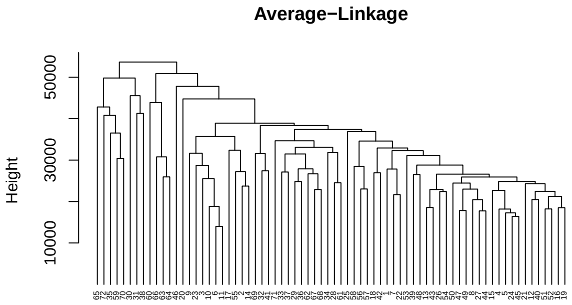


Figure 12: Average Linkage
Dendrogram with original data

The left dendrogram is the normalised data and the right is the dendrogram for the original data. It is difficult to form distinct clear clusters for both the normalised and original dendrograms.The clustering of the data has not been improved when using the normalised data. In the normalised dendrogram, there are no clear clumps of observations forming clusters. For the majority of the normalised data dendrogram, there tends to be a very large cluster with one or two observations merging with the larger cluster.

Additionally, the normalised dendrogram has the majority of observations merging in a small height range meaning a lot of the observations are located together. The occurrence of this is fine in the context of sub-clusters and small heights but in our case, this rapid clustering is occurring for the majority of observations and is causing one large cluster to be formed with many outliers merging last which can be seen to the very left of the dendrogram. Overall, this is not improving the clustering of the data which looking from the unsupervised point of view.

We can plot the dendrogram for the normalised data to make a comparison again with the Ward's method:

```r
Ward.normal <- hclust(euclid.dist.normalized, method = "ward.D2")
plot(Ward.normal, main = "Ward's", xlab = "", ylab = "", sub = "Figure 13: Wards Dendrogram
     with normalised data", hang = -1, cex = 0.53)
```

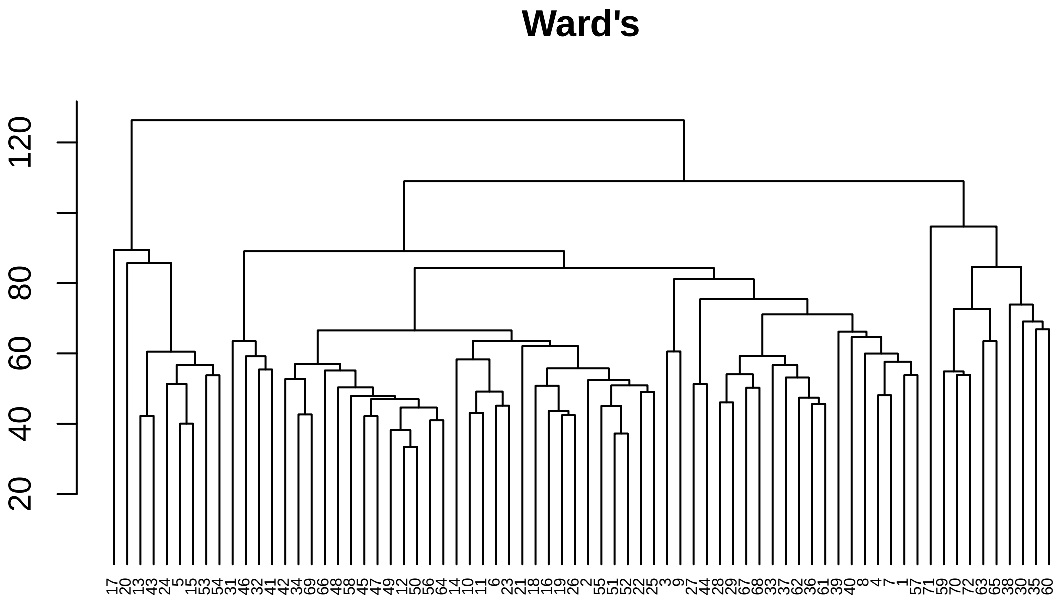


Figure 13: Wards Dendrogram
with normalised data

Then the normalised data dendrogram can be plotted side-by-side with the original data dendrogram for the Ward's method:

```r
#plotting normalised data Ward dendrogram
plot(Ward.normal, main = "Ward Method Normalised", xlab = "",
     sub = "Figure 13: Wards Dendrogram with normalised data", hang = -1, cex = 0.53)

#plotting the non-scaled data Ward dendrogram
plot(Ward, main = "Ward Method Linkage", xlab = "",
     sub = "Figure 13: Wards Dendrogram with normalised data", hang = -1, cex = 0.53)
```

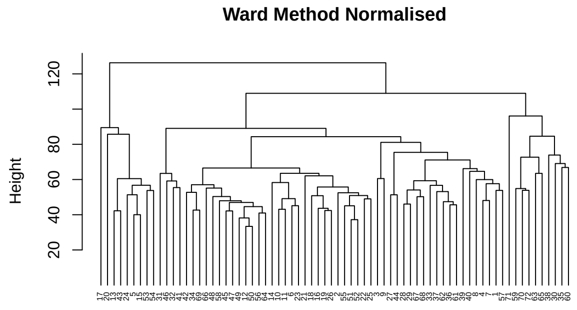


Figure 13: Wards Dendrogram with normalised data

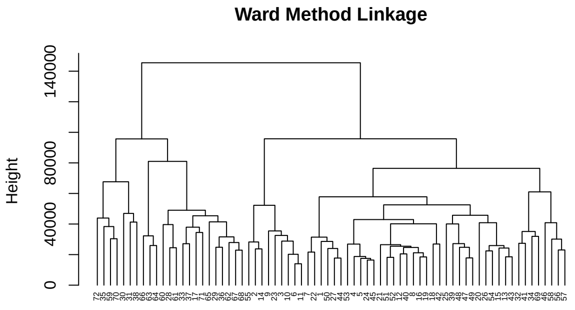


Figure 13: Wards Dendrogram with normalised data

The left dendrogram is the dendrogram for normalised data and the right is for the original data. The clustering of the data has not been improved when using the normalised data.

The density of the cluster merges is higher up the dendrogram for the normalised data. This indicates that the Ward's method is having a harder time identifying tissue samples with similar gene expression levels when the data is normalised. The length of the vertical lines in the normalised dendrogram when there are large clusters (e.g. k = 2) is smaller than those found in the un-normalised dendrogram. This may indicate the algorithm isn't noticing the significance of the distance between the main clusters and hence is merging them together quicker after they have formed.

It is clear that normalisation of the data has not improved the amount of clear partitions that can be seen from the dendrogram.

Question 9.b

The analysis of the dendrograms improvement was purely based on the visual interpretation of the dendrograms, in a completely unsupervised way. The class labels stored earlier can now be used to confirm whether or not normalisation has helped the clustering of the tissue samples. A partition with k = 2 clusters from each of the four normalised dendrograms produced in the section above can then be used to compute the contingency table of the true class labels by the assigned cluster label. These contingency tables can be then used to discuss the correspondence between classes and clusters resulting from each linkage algorithm.

Firstly, the cut at k = 2 can be done to the normalised, Single-Linkage clustering result and the corresponding contingency table can be generated as follows:

```
#cut the tree at k = 2 clusters
cut.tree.SLA <- cutree(SLA.normal, k=2)

#compute the dendrogram for k = 2 clusters
table(cutree = cut.tree.SLA, classlabels = Classes)
```

```
##       classlabels
## cutree  1  2
##      1 47 24
##      2  0  1
```

Looking at the cluster labeled '1' by the cut tree, it is made up of 47 observations of the sub-type leukemia labeled '1' and 24 observations of sub-type leukemia labeled '2'. Now looking at the cluster labeled '2' by the cut tree, is made up of 0 observations of the sub-type leukemia labeled '1' and 1 observation of the sub-type leukemia ‘2’. This suggests that the cluster labeled 1 is associated with the sub-type ‘1’ and the cluster labeled 2 is associated with the sub-type ‘2’. If this is the case, then there are 24 class type ‘2’ with incorrect resulting cluster with a label ‘1’. This is a high amount of observations with incorrect clustering results and suggests that their is a lack of correspondence between classes and clusters resulting from the Single-Linkage Algorithm. This confirms the findings from the unsupervised visual interpretation of the Single-Linkage Algorithm dendrogram that normalisation has not helped with clustering.

Similarly, a cut at k = 2 clusters can be done to the Complete-Linkage Algorithm dendrogram and then the corresponding contingency table can be generated as follows;

```
#cut the tree at k = 2 clusters
cut.tree.CLA <- cutree(CLA.normal, k=2)

#Form the contingency table for k = 2 clusters
table(cutree = cut.tree.CLA, classlabels = Classes)
```

```
##       classlabels
## cutree  1  2
##      1 45 25
##      2  2  0
```

Looking at the cluster labeled ‘1’ by the cut tree, it is made up of 45 observations of the sub-type leukemia labeled ‘1’ and 25 observations of sub-type leukemia labeled ‘2’. Now looking at the cluster labeled ‘2’ by the cut tree, is made up of 2 observation of the sub-type leukemia labeled ‘1’ and 0 observations of the sub-type leukemia ‘2’. There are two situations which both result in a lack of correspondence between classes and clusters resulting from the Complete-Linkage Algorithm. Firstly, class ‘1’ could be associated with cluster ‘1’ which means there are 45 observations correctly clustered and a total of 27 with the wrong clustering result. It would also mean all 25 observations of class label ‘2’ would be incorrectly clustered. On the other hand, class ‘1’ could be associated with cluster label ‘2’ meaning that 2 observations of class label ‘1’ are correctly clustered and the remaining 45 have an incorrect cluster result and all 25 observations of class label ‘2’ are correctly clustered. Either way, there is a high amount of incorrect clustering results which indicates in both situations, that there is a lack of correspondence between classes and clusters. This confirms the findings from the unsupervised visual interpretation of the Complete-Linkage Algorithm dendrogram that normalisation has not helped with clustering. Note that the Complete-Linkage Algorithm was deemed to generate prominent clusters that corresponded approximately to the clusters without the normalisation which consolidates that normalisation did not help the clustering of the tissue samples.

Similarly, a cut at k = 2 clusters can be done to the Average-Linkage Algorithm dendrogram and then the corresponding contingency table can be generated as follows;

```
#Cut the dendrogram at k = 2 clusters
cut.tree.ALA <- cutree(ALA.normal, k=2)

#Form the contingency table for k = 2 clusters
table(cutree = cut.tree.ALA, classlabels = Classes)
```

```
##       classlabels
## cutree  1  2
##      1 46 25
##      2  1  0
```

Looking at the cluster labeled ‘1’ by the cut tree, it is made up of 46 observations of the sub-type leukemia labeled ‘1’ and 25 observations of sub-type leukemia labeled ‘2’. Now looking at the cluster labeled ‘2’ by the cut tree, is made up of 1 observation of the sub-type leukemia labeled ‘1’ and 0 observations of the sub-type leukemia ‘2’. There are two situations which both result in a lack of correspondence between classes and clusters resulting from the Average-Linkage Algorithm. Firstly, class ‘1’ could be associated with cluster ‘1’ which means there are 46 observations correctly clustered and a total of 26 with the wrong clustering result. It would also mean all 25 observations of class label ‘2’ would be incorrectly clustered. On the other hand, class ‘1’ could be associated with cluster label ‘2’ meaning that 1 observation of class label ‘1’ was correctly clustered and the remaining 46 have an incorrect cluster result and all 25 observations of class label ‘2’ are correctly clustered. Either way, there is a high amount of incorrect clustering results which indicates in both situations, that there is a lack of correspondence between classes and clusters. This confirms the findings from the unsupervised visual interpretation of the Average-Linkage Algorithm dendrogram that normalisation has not helped with clustering.

Similarly, a cut at k = 2 clusters can be done to the Ward’s method dendrogram and then the corresponding contingency table can be generated as follows;

```
#Cut the dendrogram at k = 2 clusters
cut.tree.Ward <- cutree(Ward.normal, k=2)

#Create the contingency table for the
table(cutree = cut.tree.Ward, classlabels = Classes)
```

```
##       classlabels
## cutree  1  2
##      1 38 25
##      2  9  0
```

Looking at the cluster labeled ‘1’ by the cut tree, it is made up of 38 observations of the sub-type leukemia labeled ‘1’ and 25 observations of sub-type leukemia labeled ‘2’. Now looking at the cluster labeled ‘2’ by the cut tree, is made up of 9 observations of the sub-type leukemia labeled ‘1’ and 0 observations of the sub-type leukemia ‘2’. There are two situations with both result in a lack of correspondence between classes and clusters. Firstly, class ‘1’ could be associated with cluster ‘1’ which means there are 38 observations correctly clustered and a total of 34 with the wrong clustering result. It would also mean all 25 observations of class label ‘2’ would be incorrectly clustered. On the other hand, class ‘1’ could be associated with cluster label ‘2’ meaning that 9 observations of class label ‘1’ are correctly clustered and the remaining 38 have an incorrect cluster result and all 25 observations of class label ‘2’ are correctly clustered. Either way, there is a high amount of incorrect clustering results which indicates in both situations, that there is a lack of correspondence between classes and clusters. This confirms the findings from the unsupervised visual interpretation of the Ward’s method dendrogram that normalisation has not helped with clustering. Note that the Wards Algorithm was deemed to generate prominent clusters that corresponded approximately to the clusters without the normalisation which consolidates that normalisation did not help the clustering of the tissue samples.

Activity 2

Introduction

This activity involves clustering genes according to their gene-expression levels across different conditions in a controlled experiment. The main goal of the activity is to identify genes that show similar expression patterns over a wide range of experimental conditions. We have a dataset consistent of gene expression levels as a subset of 205 selected genes of yeast from 20 different measurements (experimental conditions). The data contains 205 observations (rows) and 29 variables (columns).

Question 10

Firstly reading the data into a data frame in R using the function read.arff() found in the library 'foreign' can be done as follows;

```
library(foreign)
Yeast_Data <- read.arff("yeast.arff")
Yeast_Data <- data.frame(Yeast_Data)
dim.data.frame(Yeast_Data)
```

```
## [1] 205  21
```

Then the rightmost column of the data can be set aside, storing it separately from the remaining data frame. The resulting data frame has 21 columns rather than the required 20 columns. This rightmost column is the labels (cluster1, cluster2, cluster3 and cluster4) indicate genes whose expression patterns reflect four functional categories. This last column isn't used in the clustering and will only be used for external assessment of the results post clustering. So in order to remove it for the clustering, it must be removed and stored separately from the remaining data frame as follows:

```
#the rightmost column is assigned to the variable Classes
yeast.classes <- as.matrix(Yeast_Data[,21])

#summary(Classes)

#The rightmost column is then removed from the Yeast_Data
Yeast_Data <- Yeast_Data[,-21]
```

Question 11

Instead of using euclidean distance to define how close observations are, the similarity between genes in terms of their gene expression profiles for different measurements is better captured by a correlation measure. The correlation used in our case is the Pearson standard correlation. We can use the cor() function in R which computes the Pearson correlation, but note that we want to compute the correlation between the rows rather than columns which is the default for the cor() function. So firstly, we need to transpose the data matrix to turn all the rows into columns and vice-versa. The output will be a 205x205 correlation matrix. Afterwards the correlation matrix can be calculated using cor()

```
#dim(Yeast_Data)

#transpose the matrix
Yeast_Data_T <- t(Yeast_Data)

correlation <- cor(Yeast_Data_T, method = "pearson")
```

Correlation is a similarity measure that ranges from -1 (lowest similarity) to 1 (highest similarity). But we need the dissimilarity matrix whose values are between 0 and 1. We can scale the data to be between 0 and 1 by taking (1-correlation)/2. Note that (1-correlation) will scale the data between 0 and 2 and then dividing by 2 will scale the data between 0 and 1.

```
#scaling the data: transforming from similarity to dissimilarity
scaled.correlation <- as.dist((1-correlation)/2)
```

Question 12

We can call the Complete-Linkage Algorithm and then plot the resulting dendrogram using the following code;

```
#run complete-linkage Algorithm with the correlation-based dissimilarities and method = "complete"
CLA.clust.corr <- hclust(scaled.correlation, method = "complete")

#plot the dendrogram
plot(CLA.clust.corr, main = "Complete-Linkage", xlab = "",
  sub = "Figure 15: Complete Linkage Dendrogram using correlation matrix with class labels",
  hang = -1, labels = yeast.classes, cex = 0.2)
```

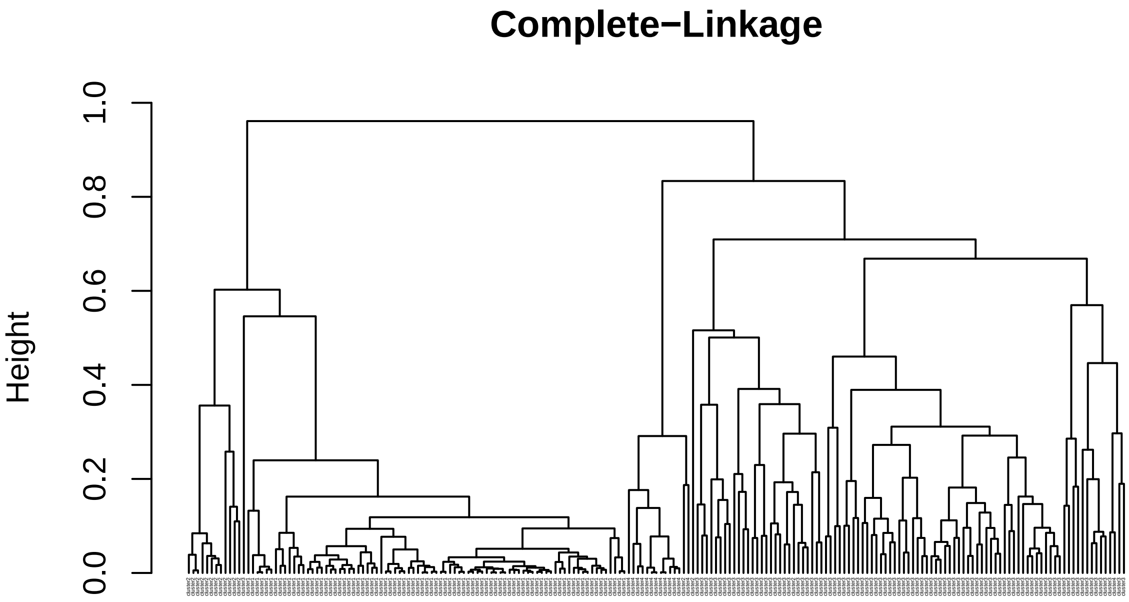


Figure 15: Complete Linkage Dendrogram using correlation matrix with class labels

This dendrogram indicates that there are three main clusters that are formed by the Complete-Linkage Algorithm. Making a partition at k = 3 clusters we can see three distinct clusters, with the left containing the most observations followed secondly by the rightmost cluster. The middle cluster has much less observations than the other two.

The left cluster is clearly associated with the gene expression level labeled 'cluster 1' with the majority of observations having that gene expression level label. Also remarking that this cluster contains a sub-cluster of 13 observations, 12 being of cluster label 'cluster2' and 1 being of cluster label 'cluster3'. These are located to the very left within the main left cluster in their own sub-grouping. This tells us that the Complete-Linkage Algorithm has joined two clusters corresponding to labels 'cluster1' and 'cluster2' and could not accurately detect that they are unique clusters.

The middle cluster is the smallest partitioned cluster consisting of 14 observations 13 of those are of label 'cluster4' and one of 'cluster2'. The Complete-Linkage Algorithm has done well to partition the 'cluster4' observations into their own main cluster with the exception of one outlier 'cluster2' observation.

The right cluster is much larger than the middle cluster with the majority of the observations being of label ‘cluster3’ with 1 of ‘cluster2’ and 1 of ‘cluster4’. This main cluster is clearly associated with the gene expression level labeled ‘cluster3’. The Complete-Linkage Algorithm has done well to partition the ‘cluster3’ gene expression level into its own cluster.

Overall there does seem some prominent clusters in the dendrogram corresponding approximately to the classes (‘cluster1’, ‘cluster2’, ‘cluster3’, ‘cluster4’). Overall, the left cluster is associated with ‘cluster1’, the middle cluster is ‘cluster4’ and the right cluster has prominent class label ‘cluster3’. But also note, there are less clusters found to be prominent than classes, with 3 prominent clusters and 4 classes. The Complete-Linkage Algorithm hasn’t been able to indicate ‘cluster2’ as a prominent singular cluster and has rather chosen to merge the sub-cluster containing the ‘cluster2’ gene expression level with the larger ‘cluster1’ cluster.

Question 13

We can call the Complete-Linkage Algorithm and then plot the resulting dendrogram. This dendrogram can then be used to determine if some prominent clusters in the dendrogram correspond approximately to the classes

```
#run complete-linkage Algorithm with the correlation-based
#dissimilarities and method = "average"
ALA.clust.corr <- hclust(scaled.correlation, method = "average")

#plot the dendrogram
plot(ALA.clust.corr, main = "Average-Linkage", xlab = "",
  sub = "Figure 16: Average Linkage Dendrogram using correlation matrix with class
  labels", hang = -1, labels = yeast.classes, cex = 0.2)
```

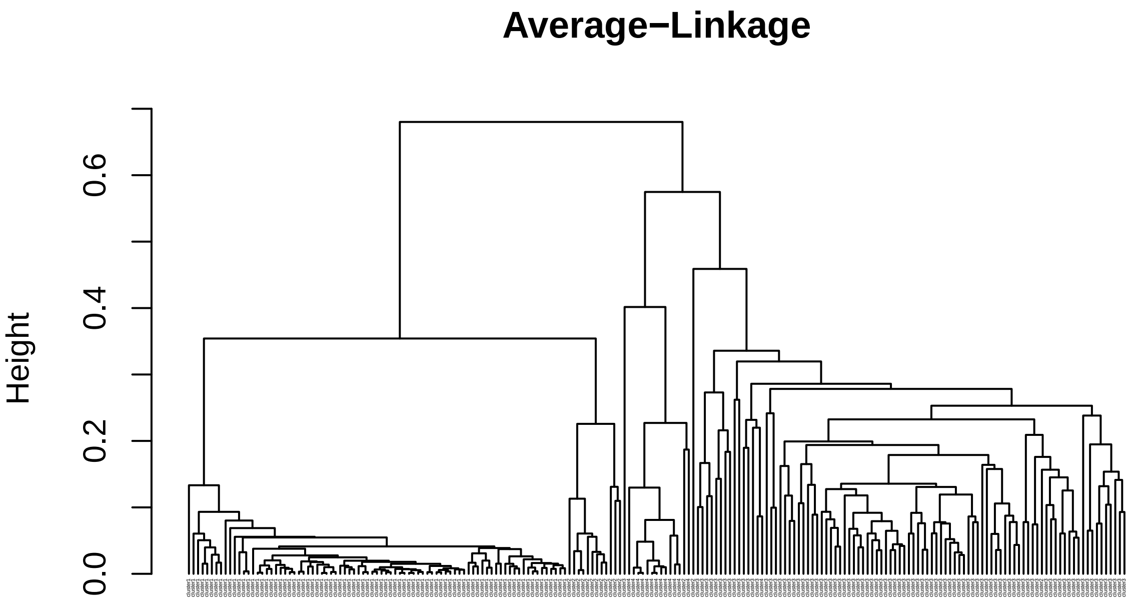


Figure 16: Average Linkage Dendrogram using correlation matrix with class labels

Similarly to Complete-Linkage Algorithm, this dendrogram indicates that there are three main prominent clusters. Making a partition at k = 3 clusters we can see three distinct clusters, with the left containing the most observations followed secondly by the rightmost cluster.

The middle cluster has much less observations than the other two. The left cluster is clearly associated with the gene expression level labeled ‘cluster 1’, this cluster contains a sub-cluster of 12 observations all being of cluster label ‘cluster2’. This cluster is the direct right child sub-cluster to the main prominent cluster. The presence of this sub-cluster consistent of class labels ‘cluster2’ within the main cluster of predominant class labels ‘cluster1’ means that the Average-Linkage Algorithm has joined the clusters of labels ‘cluster1’ and ‘cluster2’ and could not accurately detect that they are unique clusters and were merged to early. The algorithm was unable to distinguish between the gene expression levels of ‘cluster1’ and ‘cluster2’ which resulting in them merging.

The middle cluster is the smallest partitioned cluster consisting of 15 observations 13 of those are of label ‘cluster4’ and 1 of ‘cluster2’ and 1 of ‘cluster3’. The Complete-Linkage Algorithm has done well to partition the ‘cluster4’ observations into their own main cluster with the exception of the two outlier observations with other gene expression labels.

The right cluster is much larger than the middle cluster with the majority of the observations being of label ‘cluster3’ with 2 of ‘cluster2’ labels. This predominant cluster is clearly associated with the gene expression level labeled ‘cluster3’. The Average-Linkage Algorithm has done well to partition the ‘cluster3’ gene expression level into its own cluster.

Overall there does seem some prominent clusters in the dendrogram corresponding approximately to the classes (‘cluster1’, ‘cluster2’, ‘cluster3’, ‘cluster4’). Overall, the left cluster is associated with ‘cluster1’, the middle cluster is ‘cluster4’ and the right cluster has prominent class label ‘cluster3’. Note, there are less clusters found to be prominent than classes, with 3 prominent clusters and 4 classes. The Average-Linkage Algorithm hasn’t been able to indicate ‘cluster2’ as a prominent singular cluster and has rather chosen to merge the sub-cluster containing the ‘cluster2’ gene expression level with the larger ‘cluster1’ cluster.

Question 14

```r
#run complete-linkage Algorithm with the correlation-based dissimilarities and method = "average"
ward.clust.corr <- hclust(scaled.correlation, method = "ward.D2")

#plot the dendrogram
plot(ward.clust.corr, main = "Ward Method", xlab = "", sub = "Figure 16: Average Linkage Dendrogram
     using correlation matrix with class labels", hang = -1, labels = yeast.classes, cex = 0.2)
```

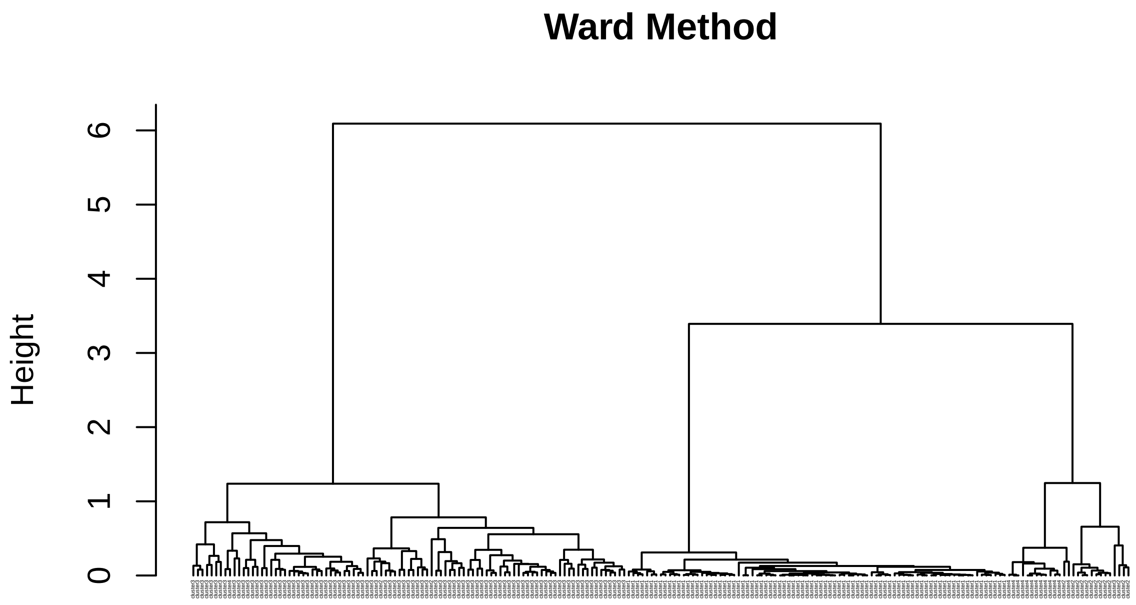


Figure 16: Average Linkage Dendrogram
using correlation matrix with class labels

The dendrogram produced using the Ward's method indicates that there are three main prominent clusters. Making a partition at k = 3 clusters we can see three distinct clusters, with the left containing the most observations followed secondly by the middle cluster. The rightmost cluster has much less observations than the other two.

The left cluster is clearly associated with the gene expression level labeled 'cluster3'. The majority of the observations being of label 'cluster3' with 2 of 'cluster2'. This predominant cluster is clearly associated with the gene expression level labeled 'cluster3'. The Ward's method has done well to partition the 'cluster3' gene expression level into its own cluster.

The right cluster is the smallest partitioned cluster consisting of 27 observations 13 of those are of label 'cluster4' and 14 of 'cluster2'. The Ward's method hasn't managed to partiton the gene expression levels with class label 'cluster4' into its own prominent cluster like Complete and the Average-Linkage Algorithms were able to do. Also note that Ward's Algorithm, like the other two algorithms, has also struggled to make four clear clusters with one associated with gene expression levels with class label 'cluster2'.

The middle cluster is a predominant cluster clearly associated with the gene expression level labeled 'cluster1'. All the gene expression levels within this cluster are of class type 'cluster1'. The Ward's method has done well to partition the ‘cluster1’ gene expression level into its own cluster.

Overall there does seem some prominent clusters in the dendrogram corresponding approximately to the classes (‘cluster1’, ‘cluster2’, ‘cluster3’, ‘cluster4’). Overall, the left cluster is associated with ‘cluster3’, the middle cluster is ‘cluster1’ and the right cluster has a mixture of both ‘cluster4’ and ‘cluster2’.

Note, there are less clusters found to be prominent than classes, with 3 prominent clusters and 4 classes. This is due to ‘cluster4’ and ‘cluster2’ being in one prominent cluster rather than two distinct cluster.

Question 15

We can perform a cut through the dendrogram of the Ward’s Algorithm to extract a partition with k = 4 clusters. The contingency table of the true class labels with the assigned cluster labels can be used to indicate the association (or lack of) between classes and extracted clusters revealed by the table. The following code executes the cut and then produces the contingency table.

```
#Cut the dendrogram at k = 4 clusters
cut.tree.ward <- cutree(ward.clust.corr, k=4)

#Compute the contingency table corresponding to k = 4 clusters
table(cutree = cut.tree.ward, classlabels = yeast.classes)
```

```
##       classlabels
## cutree cluster1 cluster2 cluster3 cluster4
##      1       83        0        0        0
##      2        0       12        1        0
##      3        0        2       92        1
##      4        0        1        0       13
```

Taking a partition at k = 4 means that the performance of Ward’s Algorithm can be assessed. Ward’s Algorithm has clustered the genes in a unsupervised way, meaning class labels can then be added to determine how well the algorithm has clustered the data.

Firstly note that the genes associated with label ‘cluster1’ is clearly corresponding to the cut cluster assigned the label of ‘1’. This can be indicated by the first column of the contingency table, with the only non-zero entry in the column being associated with the row associated to cluster label ‘1’

Now looking at the second column which is associated with class ‘cluster2’. This column of the contingency table has 12 observations in the cluster with label ‘2’, 2 observations in the cluster labeled ‘3’ and one observation in the cluster labeled ‘4’. This suggests that the genes associated with label ‘cluster2’ is clearly corresponding to the cluster assigned with the label ‘2’.

Similarly not looking at the third column associated with class ‘cluster3’, this has 92 observations in the cluster labeled ‘3’ and one observation in the cluster labeled ‘2’. This suggests that the genes with label ‘cluster2’ is clearly corresponding to the cluster assigned with the label ‘3’.

Now finally looking at the forth column associated with class ‘cluster4’, this column has 13 observations in the cluster labeled ‘4’ and 1 observation in the cluster labeled ‘3’. This indicates that the genes with label ‘cluster4’ is clearly corresponding to the cluster assigned with the label ‘4’.

Note that a partition at k = 4 was not clear from the dendrogram, with previously only 3 clusters being prominent. But now making a cut on the dendrogram with the known amount of clusters being k = 4, all of the observations found using the contingency table match up to what can be seen visually when making a cut through the dendrogram at k = 4 clusters. Partitioning the dendrogram into four distinct clusters, the left most cluster is consist of entirely genes labeled ‘cluster3’, the next cluster from the left is consistent of majority ‘cluster1’ labels, the next cluster consisting of majority ‘cluster4’ labels and finally the rightmost cluster consisting of labels with majority ‘cluster2’ labels. From left to right, the listing of the cluster labels associated to each cluster on the dendrogram can be given as follows: 3 (leftmost cluster of class label majority ‘cluster3’), 1, 2 and finally 4 (rightmost cluster of class label majority ‘cluster4’)

From the contingency table, there is a clear correspondence between classes (cluster1, cluster2, cluster3, cluster4) and the extracted clusters revealed by the table. There are clear associations between the class labels and the extracted clusters when partitioned at k = 4.