Hierarchical clustering (upgma and linkage matrix). The goal of the lab is to get experience using hierarchical clustering, better understand how trees are constructed, and how R’s hclust object works. We will also illustrate WGCNA clustering and multidimensional scaling. Include plots in your Word doc report and use a color to indicate your answers.

**A.** Background on the inner workings of hclust. hclust is an R function that takes a distance matrix as input and performs the UPGMA merging algorithm when you specify method="average" (average linkage); hclust does other linkages too. It returns the tree encoded as a 2-column linkage matrix, described more below. The hclust object can then be plotted with the plot function, which knows how to plot a tree given an hclust object.

Recall the distance matrix below from lecture. We manually constructed its UPGMA tree during class: (((A-B)-C)-(D-E))-F.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F |
| A | x | 2 | 4 | 6 | 6 | 8 |
| B |  | x | 4 | 6 | 6 | 8 |
| C |  |  | x | 6 | 6 | 8 |
| D |  |  |  | x | 3 | 8 |
| E |  |  |  |  | x | 8 |
| F |  |  |  |  |  | x |

**1.** Based on what we found in lecture, write a 2-column linkage matrix in Word using the following hclust rules.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | a (-1) | b (-2) | c(-3) | d(-4) | e(-5) | f(-6) |
| a |  | 2 | 4 | 6 | 6 | 8 |
| b |  |  | 4 | 6 | 6 | 8 |
| c |  |  |  | 6 | 6 | 8 |
| d |  |  |  |  | 3 | 8 |
| e |  |  |  |  |  | 8 |
| f |  |  |  |  |  |  |

a. Label the object A, B, C, D, E, F as negative numbers -1, -2, -3, -4, -5, -6.

b. Put the closest pair of objects in the first row of the matrix (this merged node will be referenced as 1 later in the merge matrix). Since the closest pair are A and B, the first row is

-1 -2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ab (1) | c(-3) | d(-4) | e(-5) | f(-6) |
| ab | x | 4 | 6 | 6 | 8 |
| c |  | x | 6 | 6 | 8 |
| d |  |  | x | 3 | 8 |
| e |  |  |  | x | 8 |
| f |  |  |  |  | x |

c. Next we will merge AB (1) with C (-3), and this merger will be referenced as +2 later in the merge matrix).

-1 -2

1 -3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | abc (3) | d(-4) | e(-5) | f(-6) |
| abc | x | 6 | 6 | 8 |
| d |  | x | 3 | 8 |
| e |  |  | x | 8 |
| f |  |  |  | x |

d. Repeat until you finish the tree.

|  |  |  |  |
| --- | --- | --- | --- |
|  | abc (3) | de(4) | f(-6) |
| abc | x | 6 | 8 |
| de |  | x | 8 |
| f |  |  | x |

|  |  |  |
| --- | --- | --- |
|  | abcde (5) | f(-6) |
| abcde | x | 8 |
| f |  | x |

**2.** Create a script with the following *incomplete* code. Based on your UPGMA merging calculation and your linkage matrix above, fill in the blanks in the code below to plot the tree.

# make a tree by hand with R hclust

a <- list() # initialize empty object

# define merging pattern:

# negative numbers are leaves,

# positive are merged clusters (defined by row number in $merge)

a$merge <- matrix(c(-1, -2, # 1 A-B

1, -3, # 2 (A-B)-C

-4, -5, # 3 D-E

2, 3, # 4 root ((A-B)-C)-(D-E)

4, -6 ), ncol = 2, byrow=TRUE )

a$height <- c(1, 2, 2, 3, 4) # define merge heights

a$order <- 1:6 # order of leaves(trivial if hand-entered)

a$labels <- LETTERS[1:6] # labels of leaves

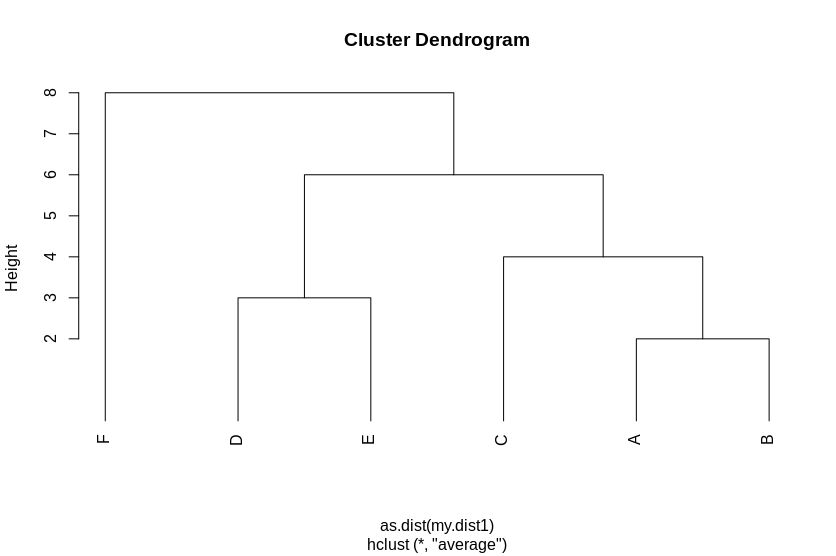
class(a) <- "hclust" # make it an hclust object

plot(a) # look at the result

#convert to a dendrogram object if needed

ad <- as.dendrogram(a)

**B.** UPGMA from distance matrix computed by built-in **hclust** and the **WGCNA** (Weighted Gene Correlation Network Analysis) tool. Given a distance matrix and a linkage tree, WGCNA attempts to find the best clusters. Clusters are objects that are more closely related to each other than with objects outside the cluster (like a clade). Add the code below to your script to read in the distance matrix distance\_matrix.txt (from Harvey) and plot the dendrogram. Show the dendrogram.



**3.** View the distance matrix in Excel and paste it in the Word doc. Look up help on read.table and look at my.dist1 to tell what header=T does.

A B C D E F

A 999 2 4 6 6 8

B 2 999 4 6 6 8

C 4 4 999 6 6 8

D 6 6 6 999 3 8

E 6 6 6 3 999 8

F 8 8 8 8 8 999

**4.** What does header=T do in this situation (see what my.dist1 looks like with header=T and header=F)?

it reads the csv file as if it has a header like any standard file reading method in any programming language. if header=F then it assumes the file just starts with the data. if header=T then it assumes each column is begun with some meta data like a title and will name the columns in the r program accordingly.

**# Set the Working Directory First**

my.dist1 <- read.table("distance\_matrix.txt",sep="\t", header=T)

rownames(my.dist1) = colnames(my.dist1)

my.dist1[lower.tri(my.dist1)] = t(my.dist1)[lower.tri(my.dist1)]

**hc1** <- **hclust**(as.dist(my.dist1), method="average")

plot(hc1,hang=-1)

Add the following code to your script to install and run WGCNA.

**5.** Looking at dynamicMods, what leaves are clustered together?

(A,B,C) and (D,E) and (F)

**Installing WGCNA.** For Mac, first install XCode from the App Store (if not already). This might also require you to upgrade your Mac OS. Then use one of the following methods for Windows or Mac.

# method 1

install.packages("WGCNA", dependencies=T)

source("http://bioconductor.org/biocLite.R")

biocLite(c("AnnotationDbi", "impute", "GO.db", "preprocessCore"))

biocLite("WGCNA")

install.package("WGCNA")

# method 2

Library(BiocManager)

BiocManager::install("WGCNA")

# load WGCNA and run on the hc1 dendrogram

cutree(hc1,k=3) # hclust cluster labels

library(WGCNA)

dynamicMods = cutreeDynamic(dendro = **hc1**, distM = my.dist1,

deepSplit = 2, pamRespectsDendro = FALSE,

minClusterSize = 2)

names(dynamicMods) <- colnames(my.dist1)

Add the following code to show the clustered data.

**6.** What do the colors at the bottom represent?

Each color is a different cluster.

dynamicColors = labels2colors(dynamicMods) # colors might not work

table(dynamicColors)

# Plot the dendrogram and colors underneath

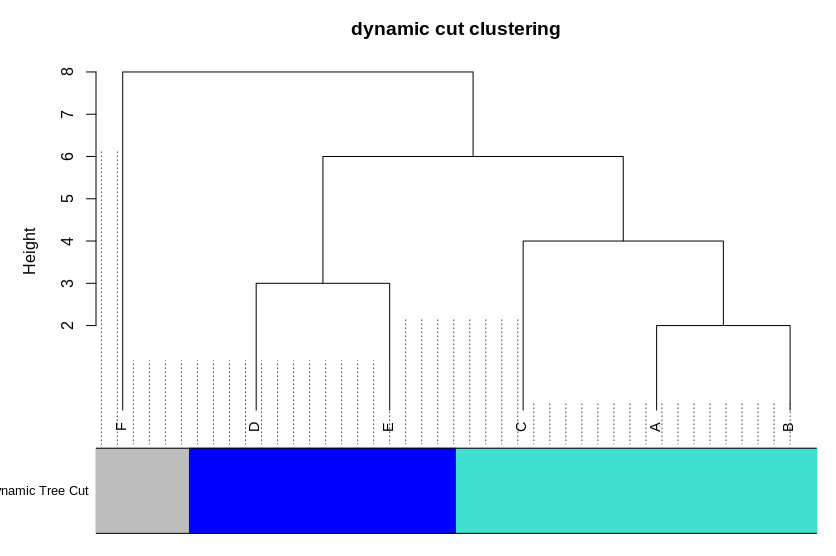
sizeGrWindow(8,6)

plotDendroAndColors(**hc1**, dynamicColors, "Dynamic Tree Cut",

dendroLabels = NULL, hang = -1,

addGuide = TRUE, #guideHang = 0.05,

main = "dynamic cut clustering")



**C.** Consider the following small distance matrix for four objects A-D. The objects might be taxa (sequences) or genes from an expression experiment. **7.** Do the UPGMA by hand, showing the new UPGMA distance matrices as you merge clusters (include your work on the Word document or upload a photo of handwritten work). **8.** Create the linkage matrix as you did in A.1, and list the height of each branch. For example, the first row of the linkage matrix is -1, -2 with height .14.

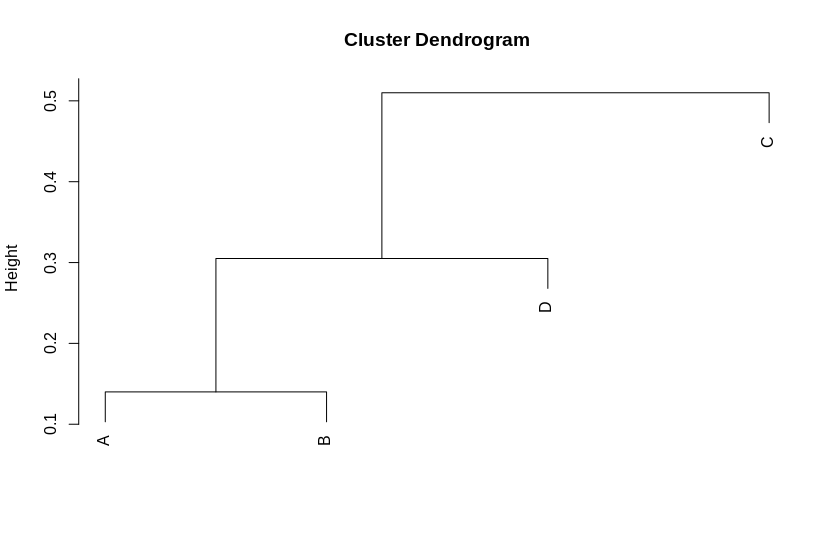
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| A | x | 0.14 | 0.48 | 0.33 |
| B |  | x | 0.5 | 0.28 |
| C |  |  | x | 0.55 |
| D |  |  |  | x |

|  |  |  |  |
| --- | --- | --- | --- |
|  | AB | C | D |
| AB | x | 0.49 | 0.305 |
| C |  | x | .55 |
| D |  |  | x |

|  |  |  |
| --- | --- | --- |
|  | ABD | C |
| ABD | x | 0.51 |
| C |  | x |

**9.** Mirror the code you used in A.2 to plot the dendrogram for this new matrix.

The results are the same just flipped because of the way in which the graph is formatted



Use code below to verify your results by applying hclust to the distance matrix.

# create matrix

my.dist2 <- matrix(data=rep(0,16),ncol=4)

my.dist2[1,2]<-.14; my.dist2[1,3]<-.48; my.dist2[1,4]<-.33;

my.dist2[2,3]<-.50; my.dist2[2,4]<-.28;

my.dist2[3,4]<-.55;

# make matrix symmetric

# make the lower triangle equal to the upper triangle

my.dist2[lower.tri(my.dist2)] = t(my.dist2)[lower.tri(my.dist2)]

diag(my.dist2)<-999 # big number to guarantee it’s not the min

# compare with hclust

test <- hclust(as.dist(my.dist2), method="average") # average linkage, UPGMA

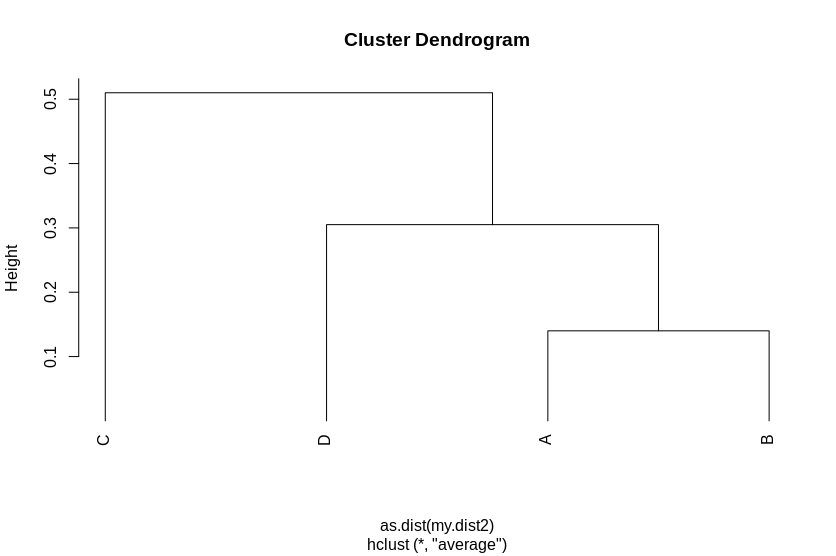
test$merge

test$height

test$order

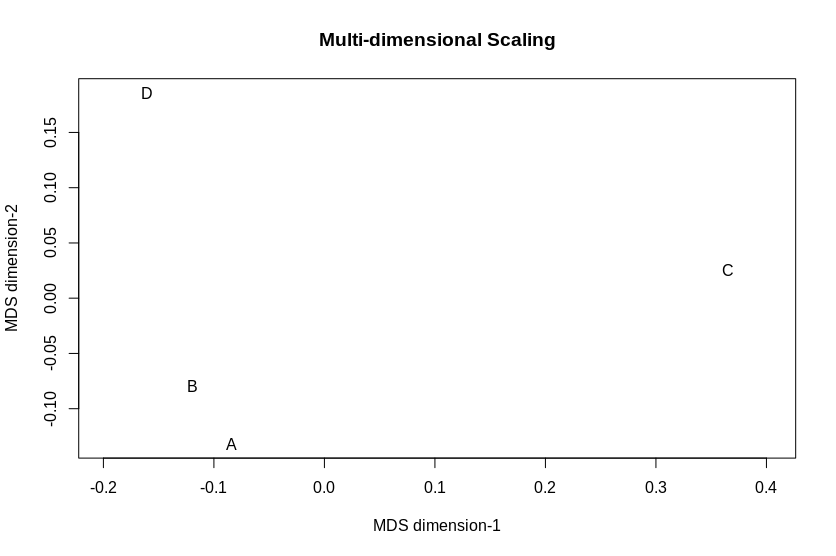
test$labels<-LETTERS[1:4]

plot(test,hang=-1)



**D.** Multidimensional Scaling: MDS plot of a distance matrix. You can visualize the relationships between the objects in an abstract 2d or 3d space using multidimensional scaling (MDS) given the pairwise distance matrix between the objects. MDS is an optimization and visualization technique that creates an artificial space such that the distance between the objects in the space is close to the distances in the original distance matrix.

**10.** Show the 2d MDS plot for my.dist2. How does this visualization change your perspective on how the objects are clustered?



## MDS, you might need to refresh (sweep) the plot window

locs<-cmdscale(as.dist(my.dist2))

x<-locs[,1]

y<-locs[,2]

**# pch=NA hides plot symbols:**

plot(x,y,main="Multi-dimensional Scaling",

xlab="MDS dimension-1", ylab="MDS dimension-2", pch=NA,asp=1)

# pch specifies plot symbol; we want null because we are using letter next

**# plot the text labels instead of symbols:**

text(x,y,test$labels,cex=1)

it is about how I expected since a and b are really close, d is a little further away and then c is really far. It is pretty intuitive given the data.