Data mining 3rd Report  
(Student Number: 1651046, Name: Daiki Kubo)

1. Save the following data in a text file called “DataGE1.txt” in the working directory of your R programming and calculate Euclidean distance and correlation matrix. You can use bioDist package.   
     
   **My Answer:**

・Source code and the Result

# read the data

data<-read.table("DataGE1.txt", header=T)

# split the data

split\_data<-data[,2:4]

# calculate the Euclidean distance

data.d<-dist(t(split\_data),method="euclidean")

# print the result

data.d  
 EX1 EX2

EX2 20.388232

EX3 6.872409 19.789644

# calculate the correlation matrix

corm<-cor(split\_data)

# print the result

corm

EX1 EX2 EX3

EX1 1.0000000 -0.6154635 0.7862770

EX2 -0.6154635 1.0000000 -0.2995611

EX3 0.7862770 -0.2995611 1.0000000

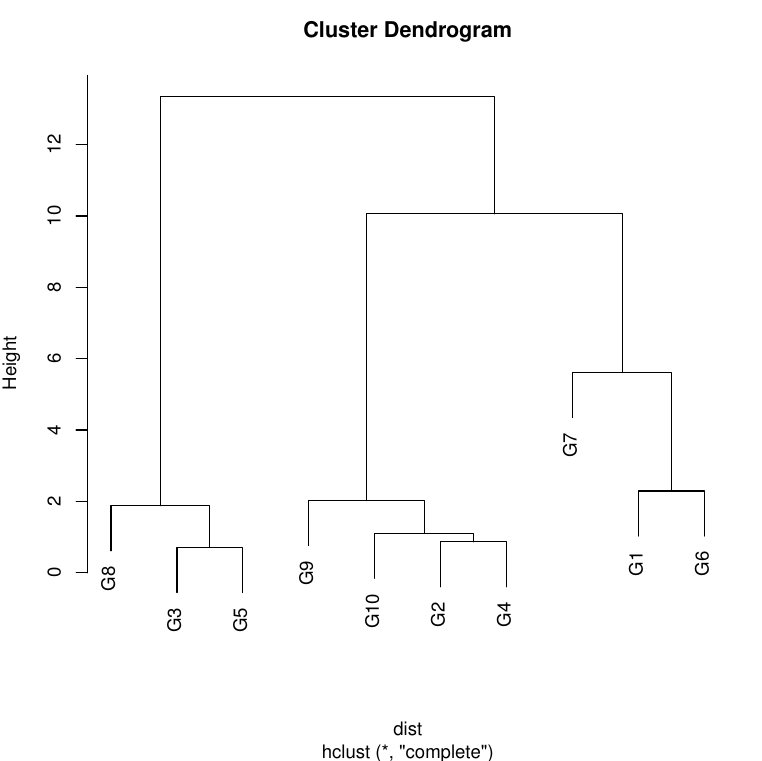
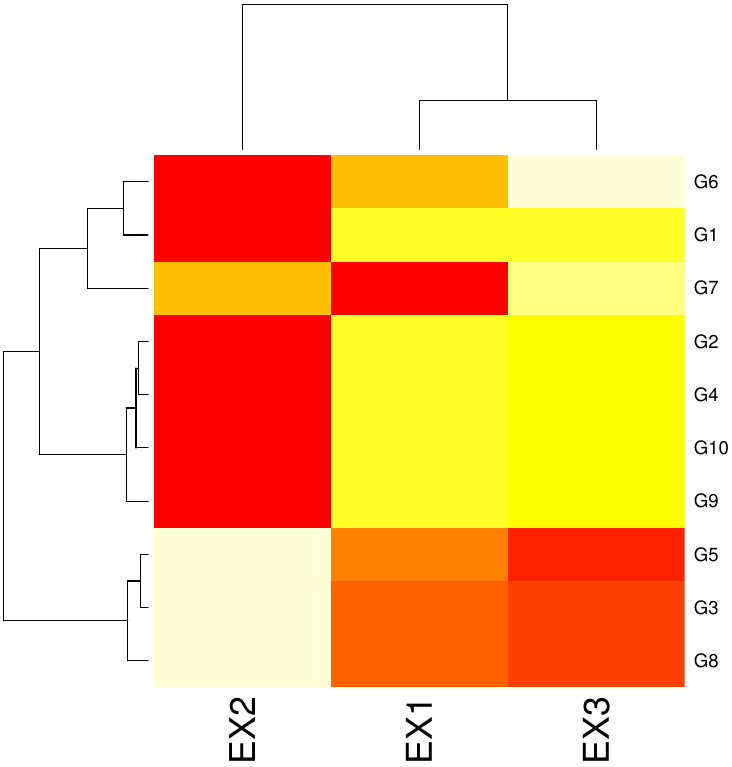
1. Run the following R commands and submit the results.  
   data <- read.table("DataGE1.txt", header=T, row.names="name")

dist <- dist(data, method = "euclidian")

plot(hclust(dist, method = "complete"))

heatmap(as.matrix(data))

**My Answer:**

1. Try to run the following commands on R and print and explain the results.

x <- rbind(matrix(rnorm(50, mean = 1, sd = 0.4), ncol = 2),

matrix(rnorm(50, mean = 5, sd = 0.3), ncol = 2),

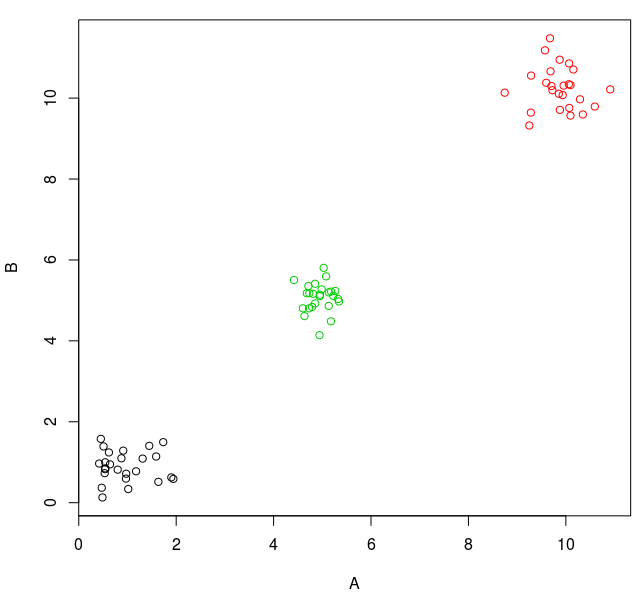
matrix(rnorm(50, mean =10, sd = 0.5), ncol = 2))

colnames(x) <- c("A", "B")

cl <- kmeans(x, 3)

plot(x, col = cl$cluster)

points (cl$centers, col = 1:3, pch = 10, cex=2)  
  
**My Answer:**



1. Calculate mahalanobis distances between genes/entities using the file "PCATutorial.txt" and plot the histogram of the distance values.  
     
   **My Answer:**

・Source code and the Result

# read the data

data<-read.table("PCATutorial.txt")

# calculate the mean vector

data.m<-colMeans(data[,2:11])

# calculate the variance vector

data.v<-var(data[,2:11])

# calculate the mahalanobis distance

data.d<-mahalanobis(data[,2:11],data.m,data.v)

# print the result

data.d

[1] 3.408220 9.994395 4.378390 6.927457 12.170131 5.774105 12.963329

[8] 5.093791 9.364135 4.349806 5.361963 5.616462 9.529145 3.715516

[15] 19.599393 8.875223 9.595899 10.907811 3.310497 8.090750 24.449418

[22] 9.491348 65.703331 8.865881 8.088876 13.308141 8.697294 2.053555

[29] 3.668418 2.547446 6.034446 5.085934 22.224254 8.996320 3.291643

[36] 4.305001 5.994517 2.246300 9.004675 2.623629 8.235770 5.838948

[43] 7.575015 8.710352 11.601862 5.937850 17.417198 5.729940 10.334906

[50] 10.127014 7.805269 26.930544 16.232711 5.729235 4.099976 11.010301

[57] 14.265910 9.725121 6.359884 16.307032 3.376125 4.349111 17.636961

[64] 15.024597 14.290483 8.686555 12.206035 3.365235 2.786247 3.985177

[71] 3.511172 7.474670 13.542272 8.394010 14.925676 5.053864 3.633595

[78] 3.903057 2.425281 5.689392 12.392946 1.159194 2.020871 38.751063

[85] 5.671907 9.286835 11.864345 19.833825 36.219045 5.486088 2.155258

[92] 18.587618 12.143284 10.218926 9.627749 8.786299 10.458571 9.003855

[99] 4.275137 22.115983

# plot the result

hist(data.d)

