Computer Exercise 3: RNA-seq data analysis

Introduction to Bioinformatics (MVE510), Autumn 2024

Group Member: Houshi He

Swedish Social Security Number: 20011114-4838

January 6, 2024

Contents

1 Questions and Answers

1.1 Question 1

Problem:
1. How many genes and how many samples are present in the count matrix?
2. How many male and female patients are there? How many have the disease?
3. How many genes are left after the low expression filtering?

Answer:
1. The dataset contains 58,037 genes and 80 samples.
2. There are 29 female patients and 51 male patients. 40 patients have the disease.
3. 38558 genes are left after the filtering.

1.2 Question 2

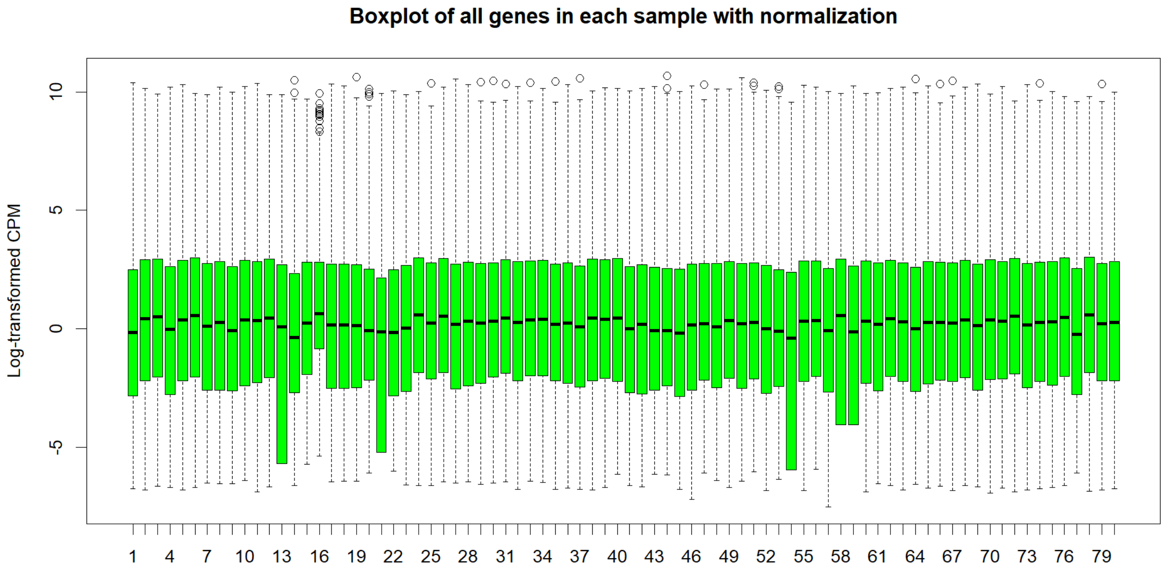
Problem:
1. Why is it important to normalize the data across samples?
2. What is the main reason to transform the counts using the logarithm?
3. Is there any additional advantage to log-transforming the data (hint: think about the distribution of the data and what kind of analysis we do later)?

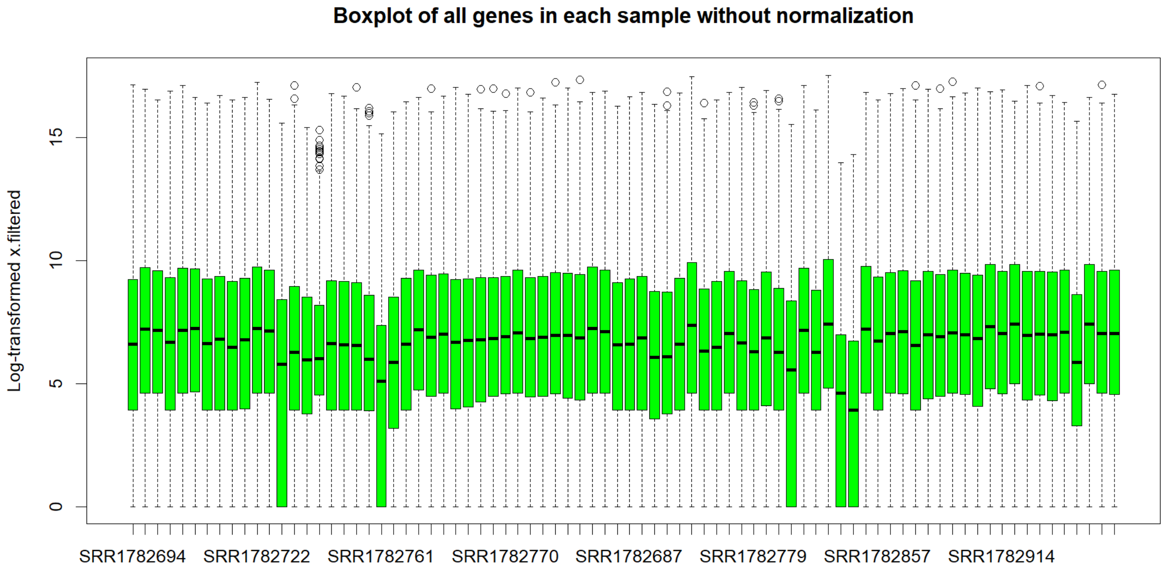
Answer:
1. Because it can help eliminate the effects of differences in sequencing depth or size.
2. We use log transformation to compress the range of data and reduce the influence of extreme values.
3. With log-transforming, the data can be adjusted to a normal distribution.

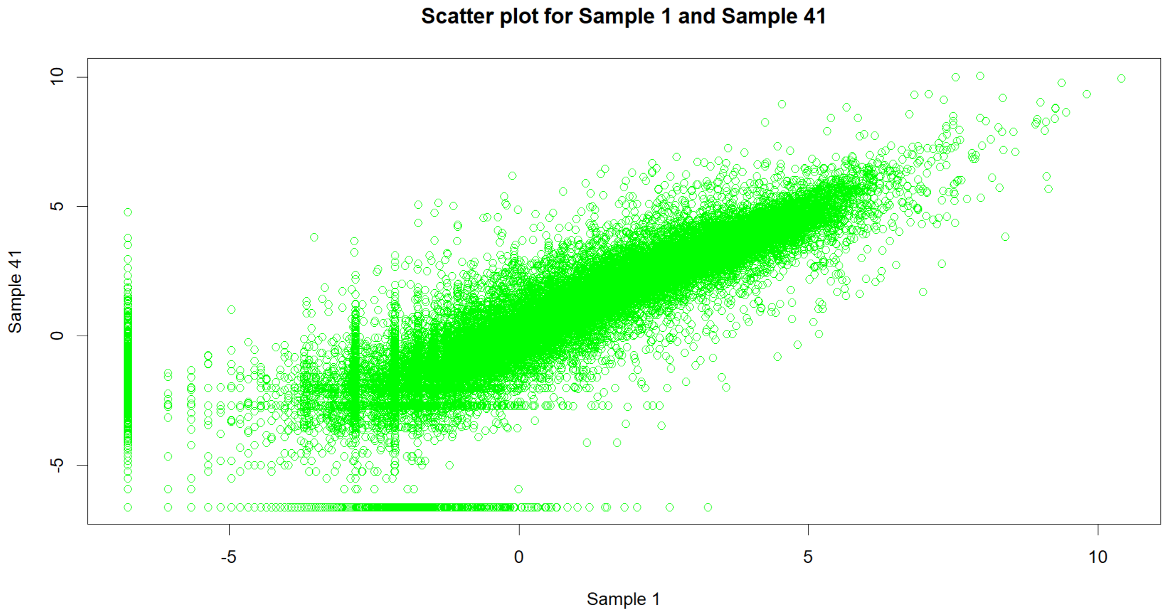
1.3 Question 3

Problem:
1. How does the distribution of counts look after normalization? Does it look as expected?
2. How do you explain the ‘stripes’ of genes that you can see at the bottom and left of the plot?

Answer:
1. After normalization, the variability reduces, and the distribution becomes more stable and closer to a normal distribution. We can see from the plot that after normalization, the medians are almost the same.
2. I think these ”stripes” are caused by genes with expression values of zero. Genes with zero counts in sample 41 and non-zero counts in sample 1 result in the bottom stripe. Genes with zero counts in sample 1 and non-zero counts in sample 41 result in the left stripe.


Boxplot of all genes in each sample with normalization
Log-transformed CPM
10
5
0
-5
1
4
7
10
13
16
19
22
25
28
31
34
37
40
43
46
49
52
55
58
61
64
67
70
73
76
79


Boxplot of all genes in each sample without normalization
Log-transformed x.filtered
15
10
5
0
SRR1782694
SRR1782722
SRR1782761
SRR1782770
SRR1782687
SRR1782779
SRR1782857
SRR1782914



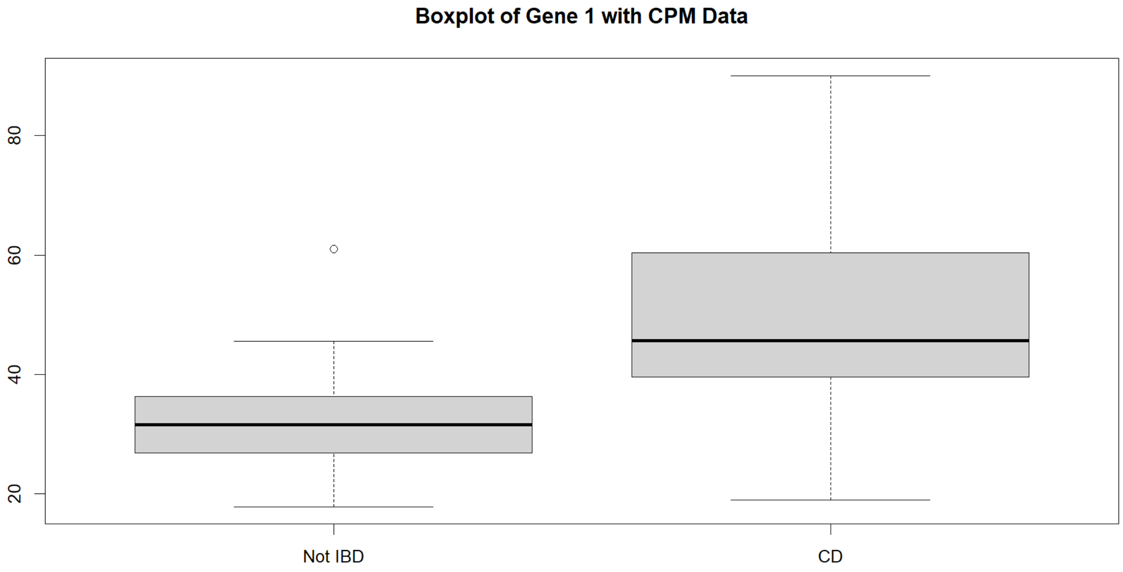
1.4 Question 4

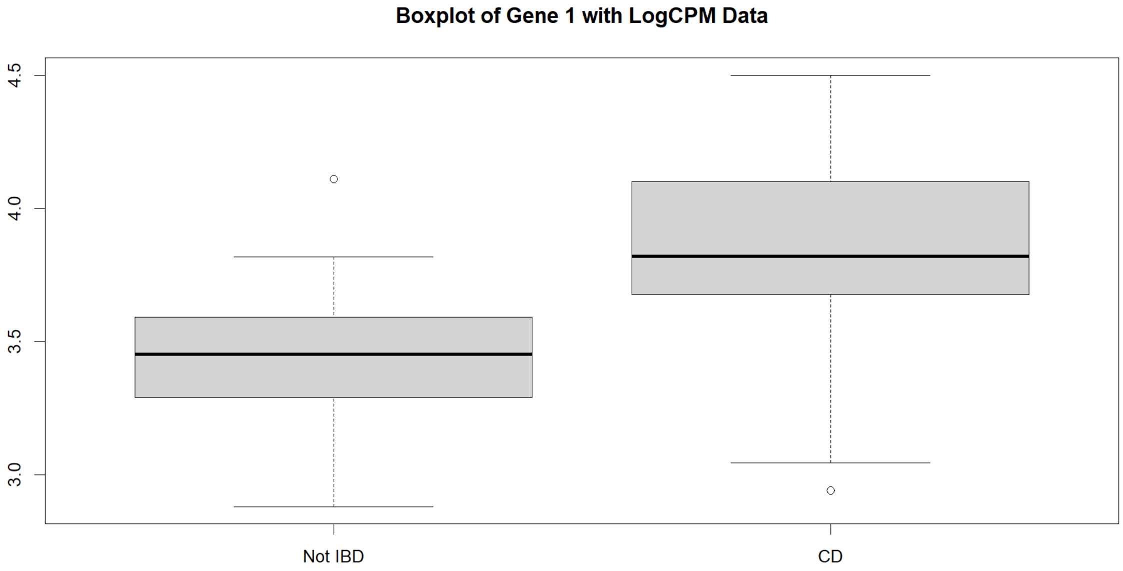
Problem:

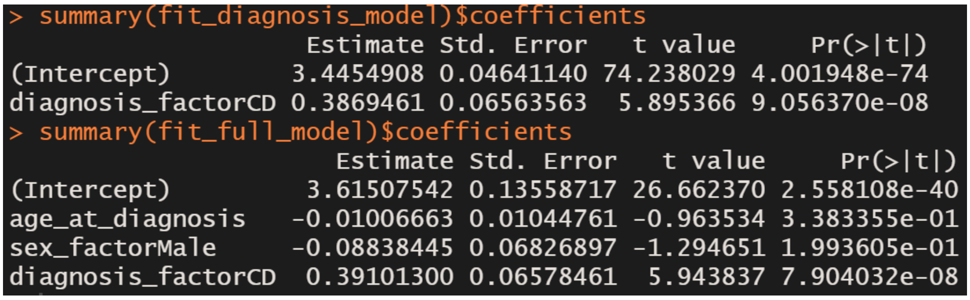
1. Can you see from the plot if there is a difference in expression between the two patient groups for the gene ENSG00000000003?
2. Is the gene ENSG00000000003 differentially expressed when comparing the two diagnosis groups, when using the first linear model (fit1)? What is the p-value? Is the gene significantly differentially expressed using the second linear model (fit2)?
3. Is the gene up-regulated or down-regulated in the disease group compared to the controls? What is the effect size, i.e. the value of the parameter associated with the independent variable specifying if the patient is sick or healthy? Is the expression of the gene influenced by age or gender? What is the difference in using the model in fit2 compared to fit1?

Answer:

1. Yes, there is a difference, gene 1 is expressed more in the diseased group.
2. The p-value for fit1 is 9.056370e-08 and for fit2 is 7.904032e-08. Therefore, the gene is differentially expressed when using both fit1 and fit2.
3. The value in the Estimate column is positive, so the gene is up-regulated. The estimated value of effect size is 0.3869461. The p-value for age is 0.3383 and the p-value for gender is 0.1994, both of them are larger than 0.05, so the expression of the gene is not influenced by age or gender. Fit1 only considers the diagnosis group, while fit2 also considers the two factors: age and gender.







```
> summary(fit_diagnosis_model)$coefficients
                   Estimate Std. Error   t value     Pr(>|t|)
(Intercept)       3.4454908 0.04641140 74.238029 4.001948e-74
diagnosis_factorCD 0.3869461 0.06563563  5.895366 9.056370e-08
> summary(fit_full_model)$coefficients
                     Estimate Std. Error   t value     Pr(>|t|)
(Intercept)        3.61507542 0.13558717 26.662370 2.558108e-40
age_at_diagnosis  -0.01006663 0.01044761 -0.963534 3.383355e-01
sex_factorMale    -0.08838445 0.06826897 -1.294651 1.993605e-01
diagnosis_factorCD 0.39101300 0.06578461  5.943837 7.904032e-08
```

1.5 Question 5

Problem:
1. Why is it, in this case, important to adjust the p-values for multiple testing?
2. How many genes are significantly differentially expressed when using the first model (only one factor)? What false discovery rate cutoff did you use? How many genes are differentially expressed (comparing disease to control) when using the second model (three factors)? Out of the significant genes, how many are up-regulated and down-regulated respectively, when comparing disease to control? Explain also the reason for calculating the FDR in this case and how it can be used to ensure that the results do not contain a large number of false positives.
3. Which gene is the most significant when comparing disease to control? Is it up or downregulated in the disease group? What is the effect size (log fold-change)? Explain the role of this gene and try to explain why it is differentially expressed in patients with Chron's disease.
4. Have a look at the other top 5 most significant genes. Are they up- or down-regulated? What are they doing? Can you explain their role based on the biological question in this study?
5. How many genes are significantly associated with age and gender respectively? Which gene is the most significant for the gender factor? Why do you think it is differentially expressed?
6. Why do we expect to get false positives?

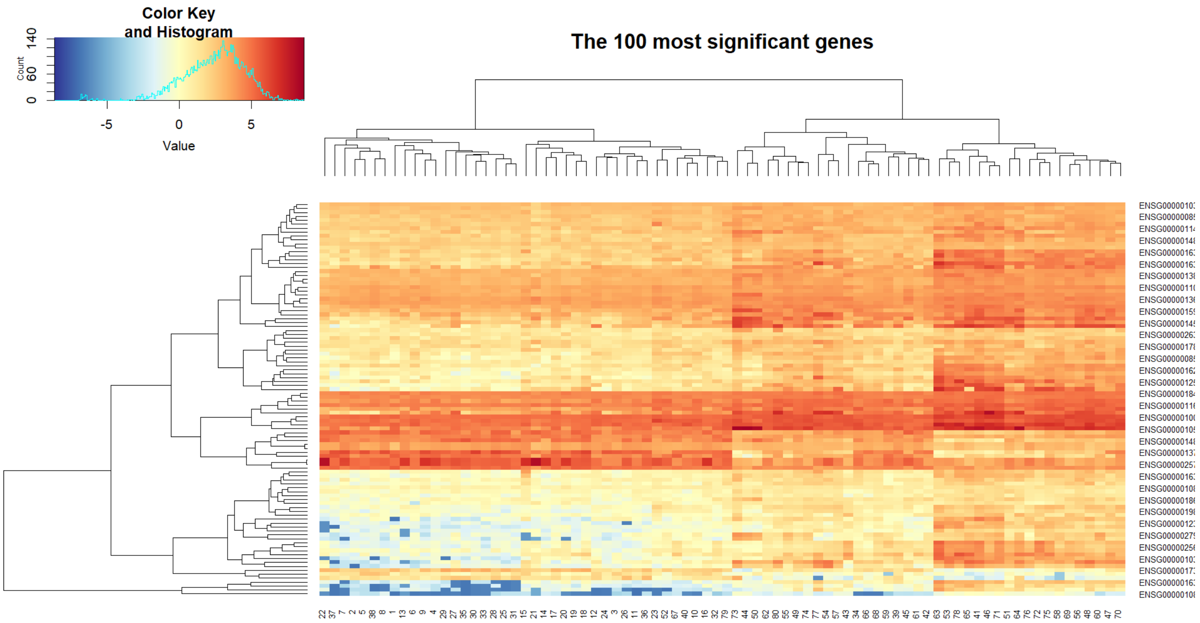
Answer:
1. Because adjusting the p-value is crucial to controlling the false positive rate for multiple testing.
2. 5702 genes are significantly differentially expressed when using the first model. I use 0.05 as the cutoff. 5464 genes are significantly differentially expressed when using the second model. For the first model, 2499 are up-regulated and 3203 are down-regulated. For the second model, 2338 are up-regulated and 3126 are down-regulated. By calculating FDR, we can ensure that the proportion of false positives among the significant genes found is as low as possible.
3. The most significant gene is ENSG00000185499. It is down-regulated. The effect size is 0.7295113. This gene(MUC1) encodes a membrane-bound protein that is a member of the mucin family. Mucins are O-glycosylated proteins that play an essential role in forming protective mucous barriers on epithelial surfaces. This protein is expressed on the apical surface of epithelial cells that line the mucosal surfaces of many different tissues including lung, breast stomach, and pancreas. MUC1 may play a crucial protective role in Chron's disease. Patients with Chron's disease often have defects in intestinal barrier function, resulting in decreased or altered expression levels of MUC1.
4. "ENSG00000203747", "ENSG00000183010", "ENSG00000150337", "ENSG00000162747", and "ENSG00000182240" are the other top 5 most significant genes. They are all up-regulated. They are Fc gamma receptor IIIa, Pyrroline-5-carboxylate reductase 1, Fc gamma receptor Ia, Fc gamma receptor IIIb, and Beta-secretase 2. For Beta-secretase 2, this gene encodes an integral membrane glycoprotein that functions as an aspartic protease. The encoded protein cleaves amyloid precursor protein into amyloid beta peptide, which is a critical step in the etiology of Alzheimer's disease and Down syndrome. For Pyrroline-5-carboxylate reductase 1, this gene encodes an enzyme that catalyzes the

NAD(P)H-dependent conversion of pyrroline-5-carboxylate to proline. This enzyme may also play a physiologic role in the generation of NADP(+) in some cell types. For the other 3 Fc gamma receptors, they are involved in the immune response. Changes in their expression may contribute to Crohn's disease by affecting immune function and inflammation.
5. 0 gene is significantly associated with age, and 49 genes are significantly associated with gender. ENSG00000067048(DDX3Y (DBY)) is the most significant for the gender factor. The protein encoded by this gene is a member of the DEAD-box RNA helicase family, characterized by nine conserved motifs, included the conserved Asp-Glu-Ala-Asp (DEAD) motif. These motifs are thought to be involved in ATP binding, hydrolysis, RNA binding, and in the formation of intramolecular interactions. This protein shares high similarity to DDX3X, on the X chromosome, but a deletion of this gene is not complemented by DDX3X. Mutations in this gene result in male infertility, a reduction in germ cell numbers, and can result in Sertoli-cell-only syndrome.
6. Because we have a large number of tests and random variation in data. When performing multiple statistical tests, each test has a certain probability of showing a significant result even if the null hypothesis is true.

1.6 Question 6

Problem:
1. Did the samples cluster as expected? Describe the clustering of the genes, how do the genes group in the clustering?
2. How do we expect the samples to cluster? What determines how the genes and samples cluster?

Answer:
1. I think the samples cluster very well as expected. From the top of the plot, we can clearly and easily see how the genes are clustered. The expression level of each gene can be determined from the color depth.
2. We expect the samples to cluster based on their disease status (IBD vs. non-IBD). Samples with similar gene expression are likely to group together.



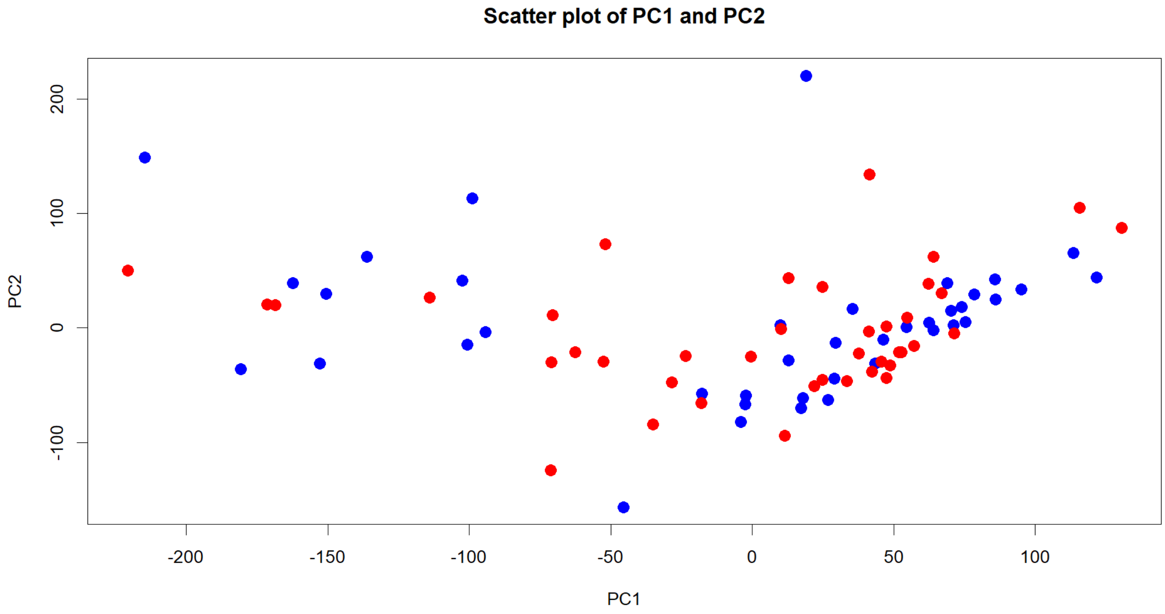
1.7 Question 7

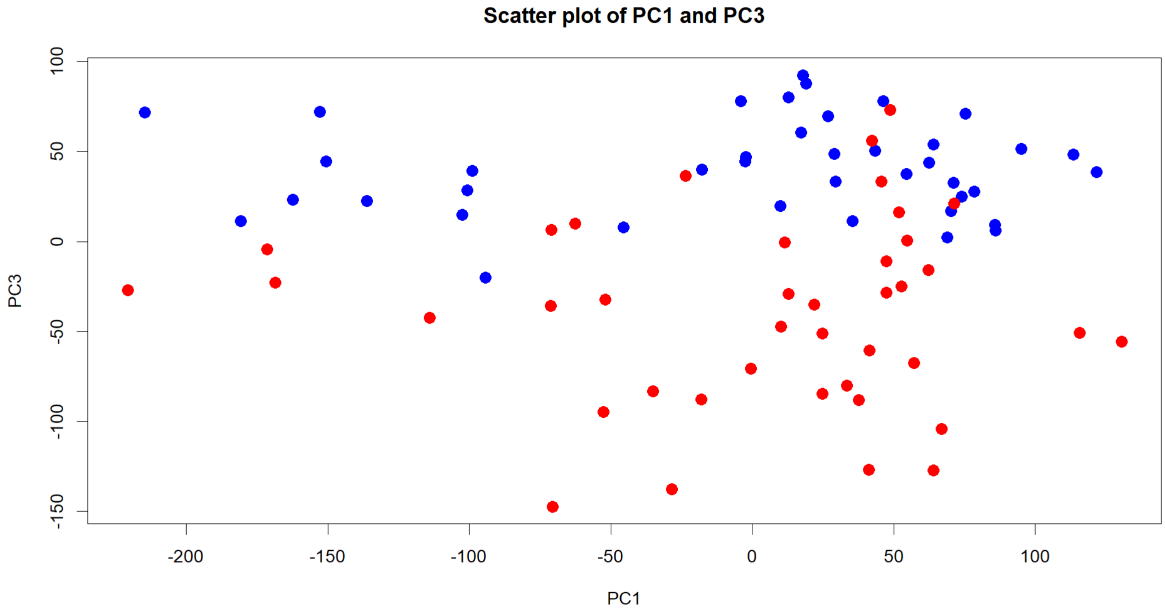
Problem:

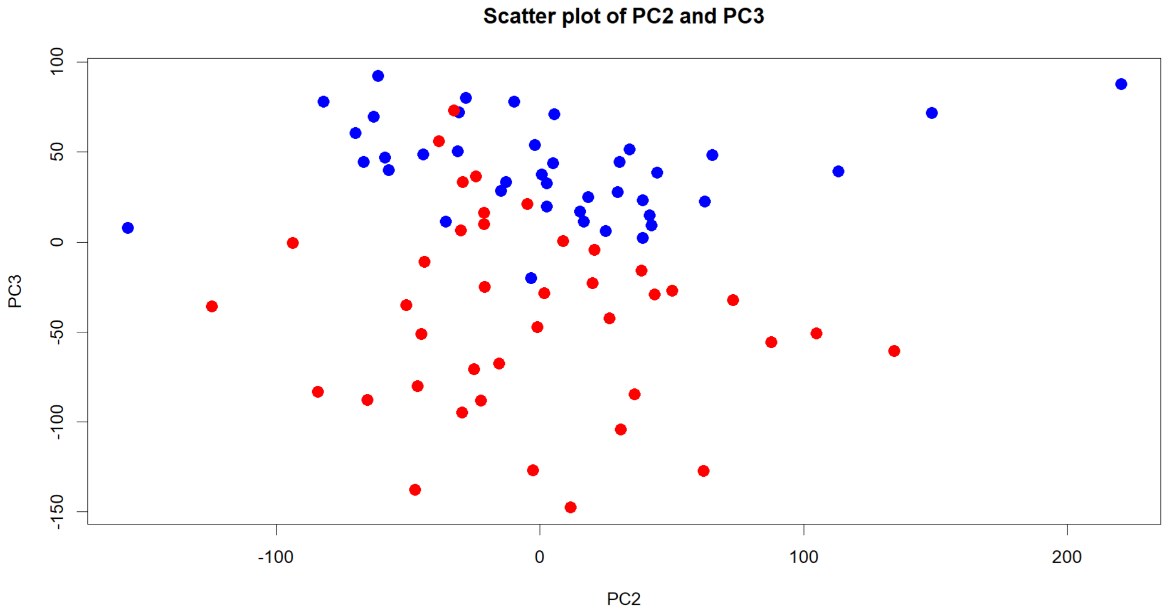
1. How much of the variability in the data is explained by the first two principal components (PC1 and PC2)?
2. Are the first two components sufficient to separate the Chron's disease samples from the controls?
3. Try to include the third principal component as well, for example by plotting PC1 against PC3 and PC2 against PC3. Do you see any separation in any of the groups? Are any of the samples problematic? Why could that be?
4. Color the samples based on gender as well. Do the groups separate in the plot?

Answer:

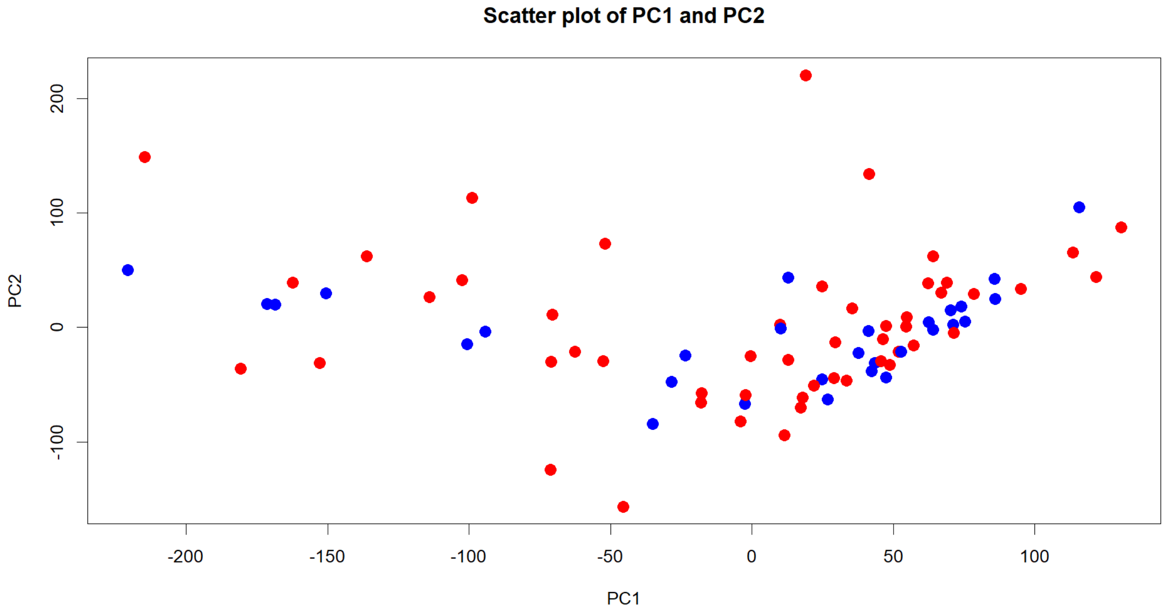
1. From Proportion of Variance, we can see the first two principal components (PC1 and PC2) together explain 16.43%(10.86% + 5.57%) of the variability in the data.
2. I don't think so. First, 16.43% is not big enough, and from the first plot, we can see that the samples are not separated.
3. By plotting PC1 against PC3 and PC2 against PC3, I can see the separation. Yes, there are problematic samples. I think it's mainly because we don't consider enough principal components, especially we didn't consider the most important one or the second most important one.
4. The last 3 plots are based on gender, we can see that the samples are not separated.


Scatter plot of PC1 and PC2

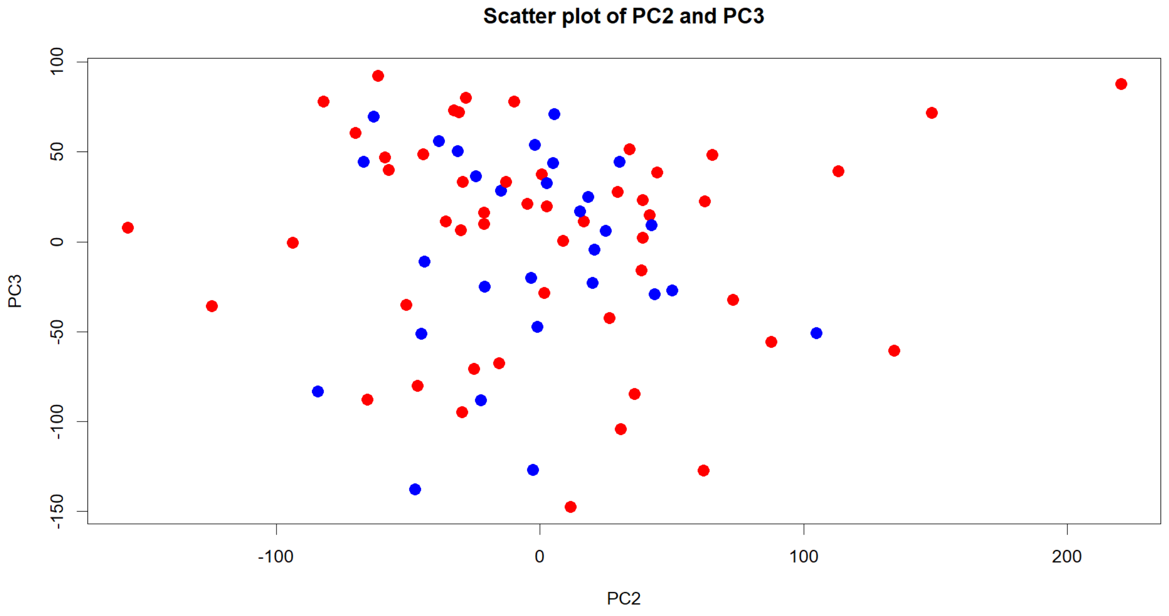

Scatter plot of PC1 and PC3

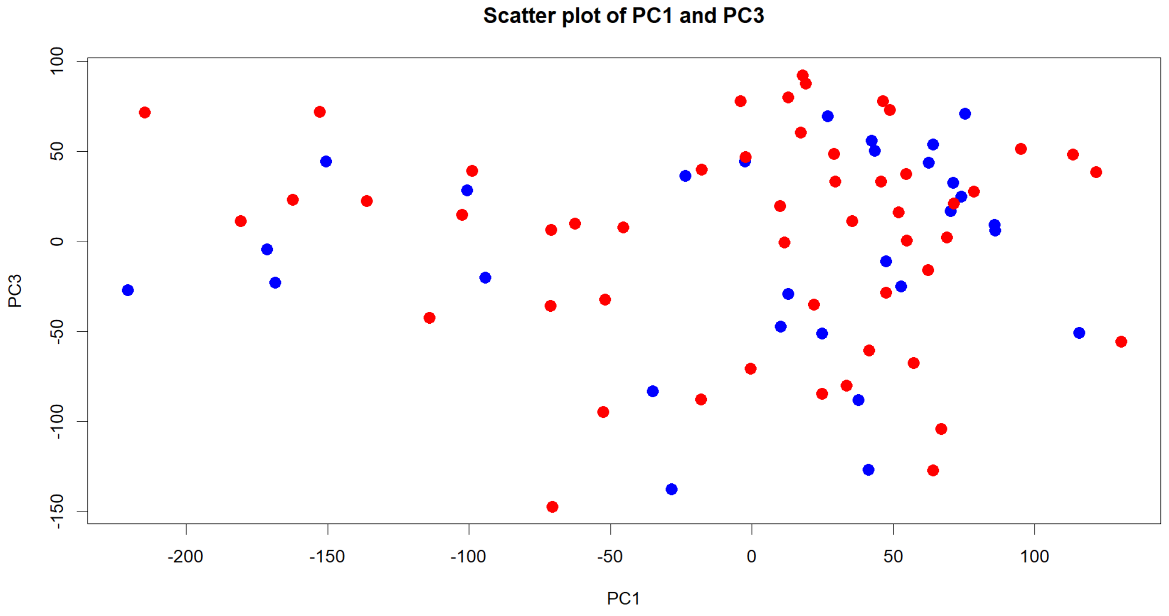


Scatter plot of PC2 and PC3
PC3
100
50
0
-50
-100
-150
-100
0
100
200
PC2



Scatter plot of PC1 and PC2
PC2
200
100
0
-100
-200
-150
-100
-50
0
50
100
PC1





2 Appendix

```
# Code for Question 1
x = read.table('counts_matrix.txt')
metadata = read.table('metadata.txt', sep='\t', header=TRUE)

countNonZero <- function(x) {
  counts <- numeric(nrow(x))
  for (i in seq_len(nrow(x))) {
    counts[i] <- sum(x[i, ] > 0)
  }
```

```r
  counts
}

x.filtered <- x[counts >= 20, ]
proportion_filtered <- nrow(x.filtered) / nrow(x) > 0.25
nrow(x.filtered)
# Code for Question 2
calculateCPM <- function(x) {
  x <- x + 1
  numCols <- ncol(x)
  numRows <- nrow(x)
  cpmMatrix <- matrix(0, numRows, numCols)

  for (col in seq_len(numCols)) {
    colSum <- sum(x[, col])
    for (row in seq_len(numRows)) {
      cpmMatrix[row, col] <- (x[row, col] / colSum) * 1e6
    }
  }

  return(cpmMatrix)
}

filteredCPM <- calculateCPM(x.filtered)
filteredLogCPM <- log(filteredCPM)
# Code for Question 3
x.filtered.log <- log(1 + x.filtered)

boxplot(filteredLogCPM, col = "green", ylab = "Log-transformed CPM", add = FALSE)
title("Boxplot of all genes in each sample with normalization")

boxplot(x.filtered.log, col = "green", ylab = "Log-transformed x.filtered", add = FALSE)
title("Boxplot of all genes in each sample without normalization")

plot(filteredLogCPM[, 1], filteredLogCPM[, 41], col = "green",
     main = "Scatter plot for Sample 1 and Sample 41",
     xlab = "Sample 1", ylab = "Sample 41")
# Code for Question 5
gene.length = nrow(filteredLogCPM)
fit1.pval = fit1.coeff = fit2.pval = fit2.coeff = vector(length = gene.length, mode = "double")

for (i in 1:gene.length) {
  fit1.i = lm(filteredLogCPM[i,] ~ diagnosis_factor)
  fit2.i = lm(filteredLogCPM[i,] ~ age_at_diagnosis + sex_factor + diagnosis_factor)

  fit1.pval[i] = summary(fit1.i)$coefficients["diagnosis_factorCD", "Pr(>|t|)"]
  fit1.coeff[i] = summary(fit1.i)$coefficients["diagnosis_factorCD", "Estimate"]

  fit2.pval[i] = summary(fit2.i)$coefficients["diagnosis_factorCD", "Pr(>|t|)"]
  fit2.coeff[i] = summary(fit2.i)$coefficients["diagnosis_factorCD", "Estimate"]
}

fit1.padjust = data.frame(p.adjust(fit1.pval, method = "fdr"), fit1.coeff)
fit2.padjust = data.frame(p.adjust(fit2.pval, method = "fdr"), fit2.coeff)

rownames(fit1.padjust) = rownames(fit2.padjust) = rownames(filteredLogCPM)
colnames(fit1.padjust) = colnames(fit2.padjust) = c("p-adjust", "coefficients")

cutoff = 0.05

fit1.diff.coeff = fit1.padjust[fit1.padjust[, 1] < cutoff, 2]
fit2.diff.coeff = fit2.padjust[fit2.padjust[, 1] < cutoff, 2]

num.fit1.diff = length(fit1.diff.coeff)
num.fit2.diff = length(fit2.diff.coeff)

fit1.coeff.up.num = sum(fit1.diff.coeff > 0)
fit1.coeff.down.num = num.fit1.diff - fit1.coeff.up.num
fit2.coeff.up.num = sum(fit2.diff.coeff > 0)
fit2.coeff.down.num = num.fit2.diff - sum(fit2.diff.coeff > 0)

fit1.most.gene = rownames(fit1.padjust)[which.min(fit1.padjust[, 1])]
fit2.most.gene = rownames(fit2.padjust)[which.min(fit2.padjust[, 1])]

coeff.most.gene.fit1 = fit1.padjust["ENSG00000185499", 2]
coeff.most.gene.fit2 = fit2.padjust["ENSG00000185499", 2]

log.fold.change = log(sum(filteredLogCPM["ENSG00000185499", 41:80]) / sum(filteredLogCPM["ENSG00000185499", 1:40]))
log.fold.change.full = log(sum(filteredCPM["ENSG00000185499", 41:80]) / sum(filteredCPM["ENSG00000185499", 1:40]))

fit1.padjust.sorted = fit1.padjust[order(fit1.padjust[, 1]), ]
fit2.padjust.sorted = fit2.padjust[order(fit2.padjust[, 1]), ]

rownames(fit1.padjust.sorted)[1:6]
```

```
fit1.padjust.sorted[1:6, 2] > 0

rownames(fit2.padjust.sorted)[1:6]
fit2.padjust.sorted[1:6, 2] > 0

fit2.age.pval = fit2.sex.pval = vector(length = gene.length, mode = "double")

for (i in 1:gene.length) {
  fit2.i = lm(filteredLogCPM[i, ] ~ age_at_diagnosis + sex_factor + diagnosis_factor)
  fit2.age.pval[i] = summary(fit2.i)$coefficients["age_at_diagnosis", "Pr(>|t|)"]
  fit2.sex.pval[i] = summary(fit2.i)$coefficients["sex_factorMale", "Pr(>|t|)"]
}

fit2.age.padjust = p.adjust(fit2.age.pval)
fit2.sex.padjust = p.adjust(fit2.sex.pval)

rownames(filteredLogCPM)[fit2.sex.padjust == min(fit2.sex.padjust)]
# Code for Question 6
library(gplots)
library(RColorBrewer)

xsig = as.matrix(filteredLogCPM[rownames(fit1.padjust.sorted)[1:100],])
mycols = rev(colorRampPalette(brewer.pal(11, "RdYlBu"))(255))
column.cols = c("purple", "orange")[metadata$diagnosis]

heatmap.2(xsig, trace=’none’, col=mycols, main=’The 100 most significant genes’, ColSideColors=column.cols)
# Code for Question 7
pca = prcomp(t(filteredLogCPM))
summary(pca)

PC1 = pca$x[, 1]
PC2 = pca$x[, 2]
PC3 = pca$x[, 3]

colors1 = c("red", "blue")[factor(metadata$diagnosis, levels = c("CD", "Not IBD"))]

plot(PC1, PC2,
     main = "Scatter plot of PC1 and PC2",
     xlab = "PC1", ylab = "PC2",
     col = colors1, pch = 19, cex = 1.5)

plot(PC1, PC3,
     main = "Scatter plot of PC1 and PC3",
```

```r
     xlab = "PC1", ylab = "PC3",
     col = colors1, pch = 19, cex = 1.5)

plot(PC2, PC3,
     main = "Scatter plot of PC2 and PC3",
     xlab = "PC2", ylab = "PC3",
     col = colors1, pch = 19, cex = 1.5)

gender.cols = c("red", "blue")[factor(metadata$Sex, levels = c("Male", "Female"))]

plot(PC2, PC3,
     main = "Scatter plot of PC2 and PC3",
     xlab = "PC2", ylab = "PC3",
     col = gender.cols, pch = 19, cex = 1.5)

plot(PC1, PC3,
     main = "Scatter plot of PC1 and PC3",
     xlab = "PC1", ylab = "PC3",
     col = gender.cols, pch = 19, cex = 1.5)

plot(PC1, PC2,
     main = "Scatter plot of PC1 and PC2",
     xlab = "PC1", ylab = "PC2",
     col = gender.cols, pch = 19, cex = 1.5)
```