Analysis Report (Part4): Exploratory Analysis

(Last updated: 11/05/21)

In this section, I will explore the data (read counts) to get an idea of what the distribution of the data will look like.

1. Loading the Necessary Libraries and Data

```
library(AnnotationDbi)
library(org.Hs.eg.db)
library(dplyr)
par(pch = 19)
tropical = c("darkorange", "dodgerblue", "hotpink", "limegreen", "yellow")
palette(tropical)
# read a merged_counts file
merged_counts = read.table("../merged_counts-v2.tsv", quote = "", sep = '\t')
# Read phenotype sample data
pheno_data = read.csv("../phenotype_data.tsv", quote = "", sep = '\t')
```

2. Checking the Phenotype Data

```
table(pheno_data$sex)
```

```
##
## F M
## 4 2
```

```
sum(pheno_data$age_group == " ")
```

```
## [1] 0
```

```
table(pheno_data$age_group, useNA = "ifany")
```

```
##
## adult fetal
##     3     3
```

```
table(pheno_data$sex, pheno_data$age_group)
```

```
##
##     adult fetal
##   F     2     2
##   M     1     1
```

3. Normalization of Expression Data (Read Counts)

This is done here according to the Read Per Million (RPM) unit.

```
x = as.matrix(merged_counts)
counts_RPM = t(t(x) * 1e6 / colSums(x))
dim(counts_RPM)
```

```
## [1] 60671     6
```

```
head(counts_RPM, 2)
```

```
##                   SRX683795   SRX683796   SRX683799 SRX683793  SRX683794
## ENSG00000000003 16.34302175 21.95274849 13.63664102 4.3178724 6.86487631
## ENSG00000000005  0.02687997  0.04746146  0.06882847 0.1349335 0.07845573
##                  SRX683797
## ENSG00000000003 3.85272170
## ENSG00000000005 0.03409488
```

4. Checking the Distribution of the Expression Data (Read Counts)

Here, are some checks of the distribution of the expression data (output not shown)

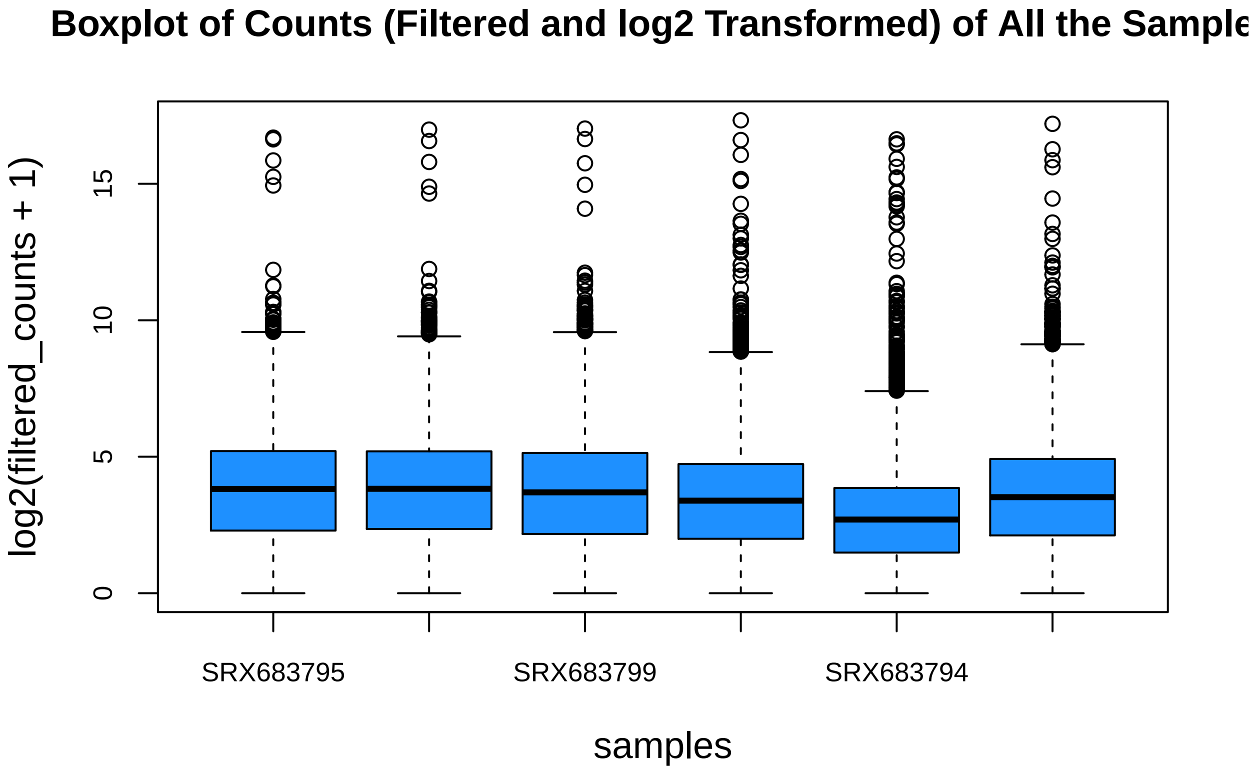
```
is.na(counts_RPM[1,])
sum(is.na(counts_RPM))

# Make the distribution of NA's by genes
gene_na = rowSums(is.na(counts_RPM))
gene_na[5]
gene_na[1:5]
# Make the distribution of NA's by samples
sample_na = colSums(is.na(counts_RPM))
sample_na[6]
table(sample_na)
```

5. Boxplot of Counts (Filtered and log2 Transformed) of All the Samples

```
counts_RPM = as.data.frame(counts_RPM)
fil_counts_RPM = filter(counts_RPM, rowMeans(counts_RPM) > 1)
dim(fil_counts_RPM)
```

```
## [1] 18863     6
```

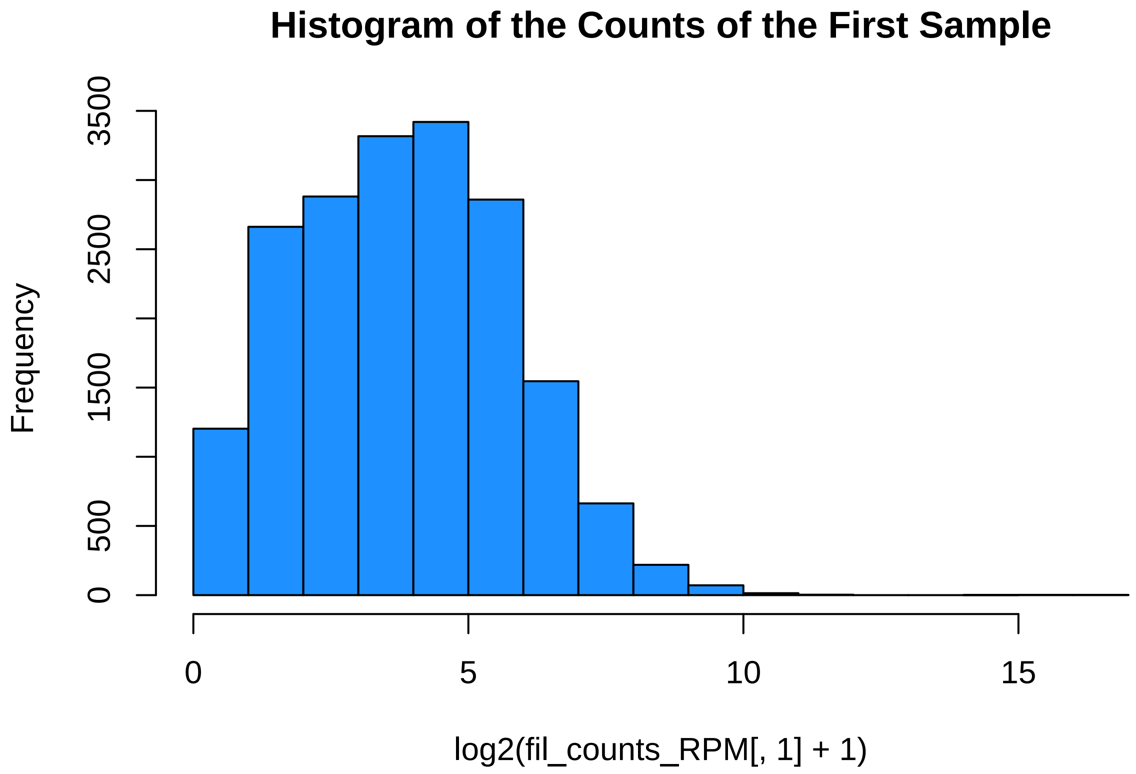
```
boxplot(as.matrix(log2(fil_counts_RPM+1)), col=2, cex.axis = 0.8, cex.lab = 1.2,
        xlab = "samples", ylab = "log2(filtered_counts + 1)",
        main = "Boxplot of Counts (Filtered and log2 Transformed) of All the Samples")
```



6. Histogram of the Counts of the First Sample

Plot a histogram to show the probability/frequency distribution of the filtered counts data of the first sample.

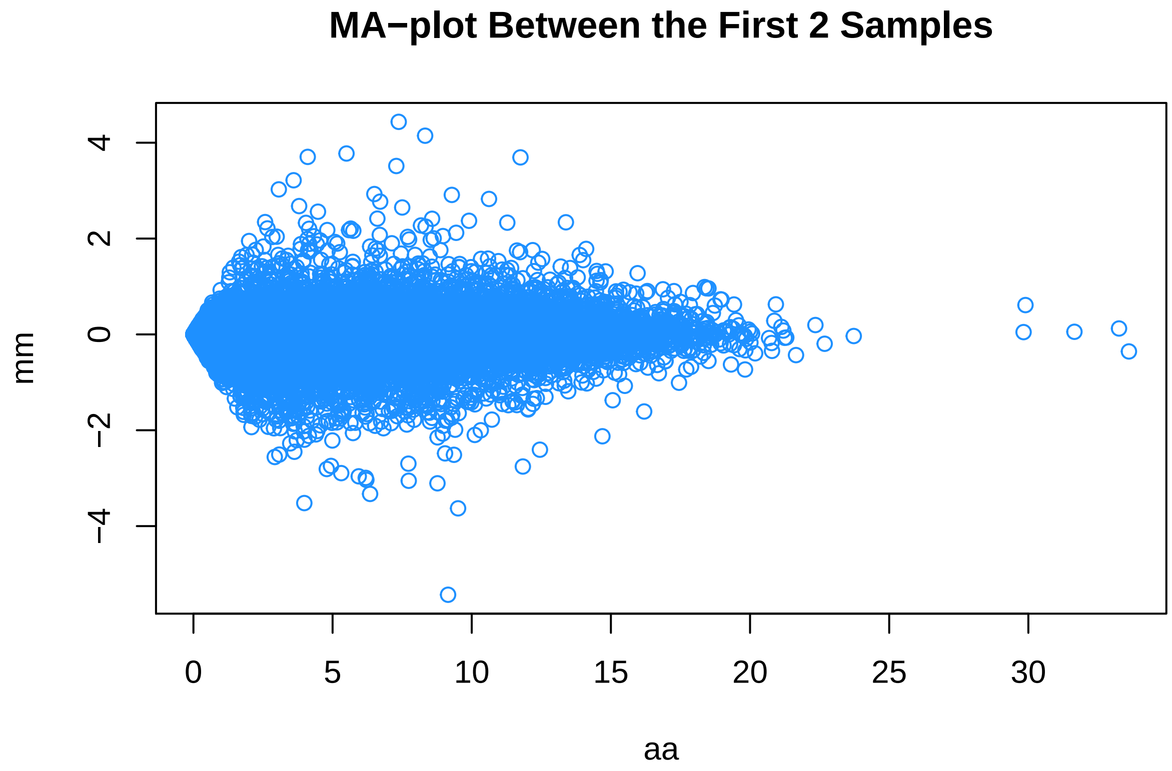
```
hist(log2(fil_counts_RPM[,1]+1), col = 2, Xlab = "log2(filtered_counts[,1] + 1)",
     main = "Histogram of the Counts of the First Sample")
```



7. MA-plot Between the First 2 Samples

The MA-plot between the first 2 samples is used to visualize the differences between measurements in those samples.

```
aa = log2(counts_RPM[,1]+1) + log2(counts_RPM[,2]+1)
mm = log2(counts_RPM[,1]+1) - log2(counts_RPM[,2]+1)
plot(aa, mm, col=2, main = "MA-plot Between the First 2 Samples")
```



8. Counts of Chromosome Y Genes in Male and Female Samples

Since cells of female samples lack Y chromosomes, we expect that male samples have more genes on this chromosome than female samples.

```
par(pch = 19)
rownames(counts_RPM) = sub("\\.\\d+$", "", rownames(counts_RPM))
chr = AnnotationDbi::select(org.Hs.eg.db, keys = rownames(counts_RPM),
                            keytype = "ENSEMBL", columns = "CHR")
```

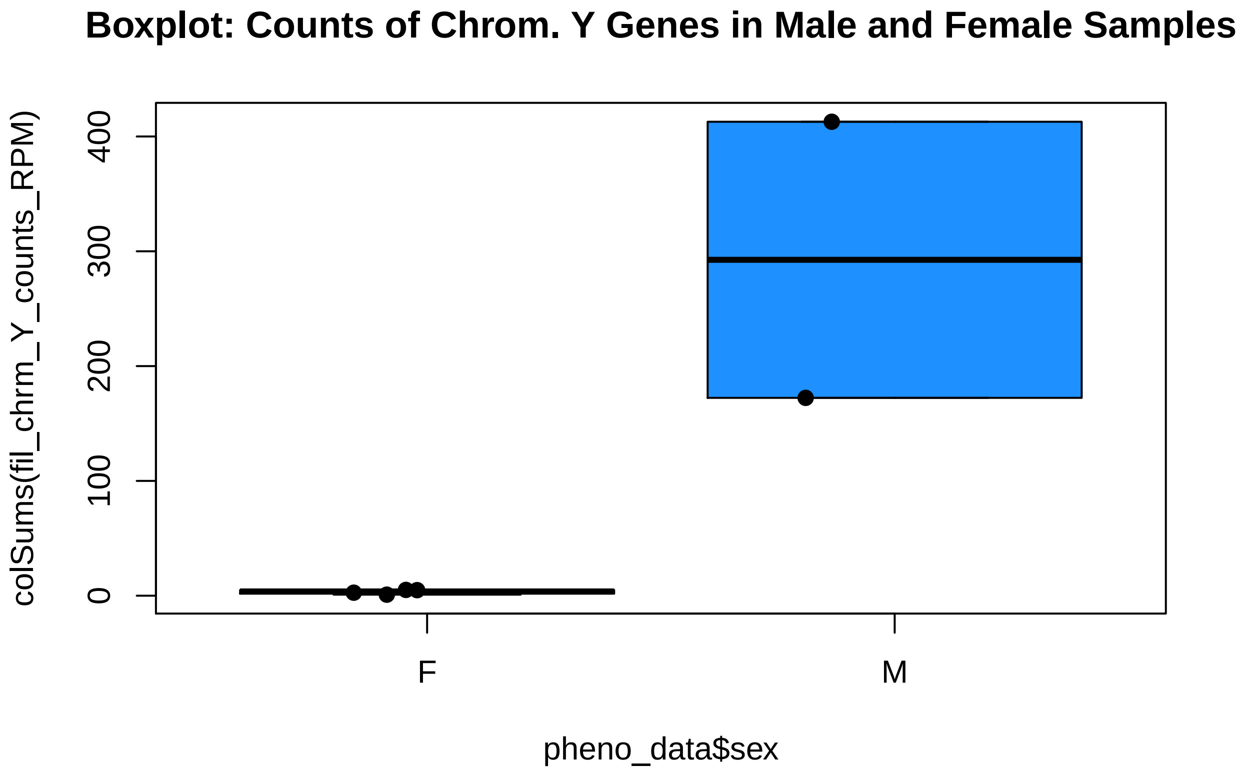
```
## 'select()' returned 1:many mapping between keys and columns
```

```
chr = chr[!duplicated(chr[,1]),]
# Confirm that the annotations still have the same sort as the counts
all(chr[,1] == rownames(counts_RPM))
```

```
## [1] TRUE
```

```
## [1] TRUE
# Select the chromosome Y samples
fil_chrm_Y_counts_RPM = filter(counts_RPM, chr$CHR == "Y")
# Male samples have more genes on chromosome Y than females
```

```
pheno_data$sex = as.factor(pheno_data$sex)
boxplot(colSums(fil_chrm_Y_counts_RPM) ~ pheno_data$sex, col=2,
        main = "Boxplot: Counts of Chrom. Y Genes in Male and Female Samples")
points(colSums(fil_chrm_Y_counts_RPM) ~ jitter(as.numeric(pheno_data$sex)))
```


Boxplot: Counts of Chrom. Y Genes in Male and Female Samples
colSums(fil_chrm_Y_counts_RPM)
F
M
pheno_data$sex