Assignment 1

Adham Khaled, Juan Manuel Medina, Antonio Ortega, Isabella Skandorff & Andreas Vincent

May 8, 2017

```
library("ggplot2")
library("cowplot")
library("reshape2")
theme_set(theme_bw())
```

Load the dataset:

- Gene name
- mRNA molecule length (base pairs)
- Genome length
- Exon count

```
df <- read.table("gene_lengths_v2.txt", header = T)
```

```
##       name mrna_length genome_length exon_count
## 1   PP8961        2596          2596          1
## 2 FLJ00038         794          2615          6
## 3    OR4F5         918           918          1
## 4    OR4F3         937           937          1
## 5   OR4F16         937           937          1
## 6   SAMD11        2555         18842         14
```

Question 1

```
#### Full distribution
upper_x <- 150
 p <- ggplot() +
  geom_polygon(data = data.frame(x = c(0, 0, 20, 20),
                                 y = c(0, 1500, 1500, 0)),
               mapping = aes(x = x, y = y), fill = "grey", alpha = 0.5) +

  geom_histogram(data = df,
                 mapping = aes(x = exon_count, fill = -..count..),
                 breaks = seq(from = 0, to = upper_x, by = 1)) +
  scale_x_continuous(name = "Exon count per gene",
                     breaks = seq(0, upper_x, 30),
                     limits = c(0, upper_x))+
  scale_y_continuous(name = "Number of genes",
                     breaks = seq(0, 1500, 100),
                     limits = c(0, 1500)) +
  guides(fill = FALSE)

#### First 20 bins
upper_x <- 20
q <- ggplot(data = subset(df, exon_count <= upper_x),
```

```
      mapping = aes(x = exon_count)) +
  geom_histogram(binwidth = 1, mapping = aes(
                 fill = -..count..)) +

  scale_x_continuous(breaks = seq(0, 20, 1)) +
  scale_y_continuous(limits = c(0, 1500),
                     breaks = seq(0, 1500, 100)) +

  labs(x = "Exon count", y = "") +
  guides(fill = FALSE) +

  geom_hline(yintercept = max(table(df$exon_count)),
             linetype = "dashed") +
  geom_text(mapping = aes(x = which.max(table(df$exon_count)), y = max(table(df$exon_count))),
            label =  max(table(df$exon_count)), vjust = -.3)

plot_grid(p, q, nrow = 1, ncol = 2, rel_widths = c(1, 2), labels = "AUTO")
```

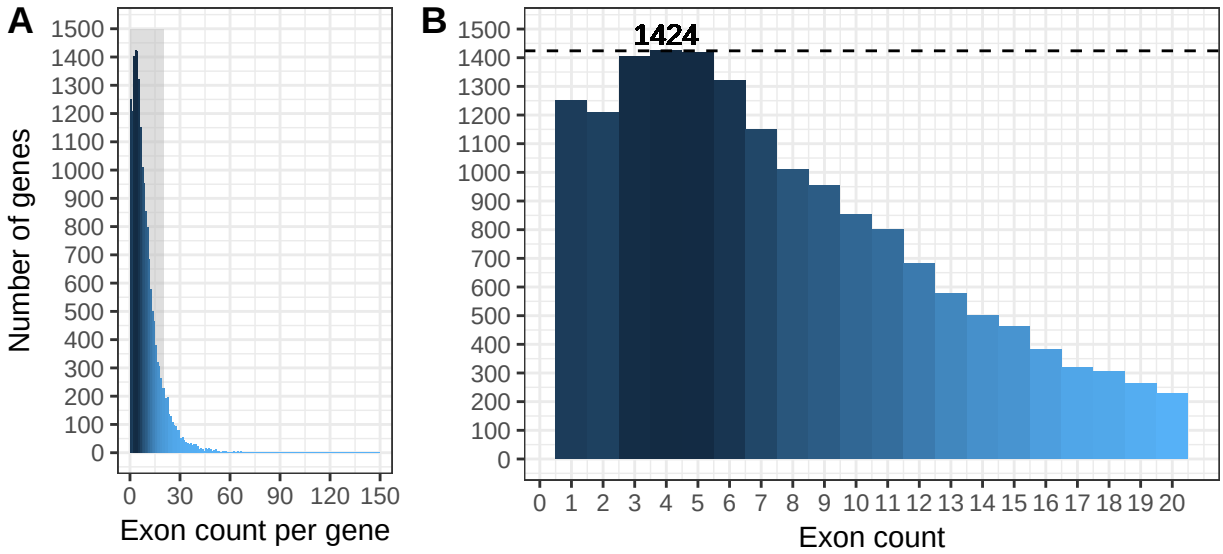

Figure 1. **A** *Histogram showing the distribution of the exon counts. Even though most of the genes contain less than 60 exons, as many as 150 may be found in some of them.* **B** *Detail for genes with max. 20 exons. The mode can be visualized at 3-5 exons per gene (max found at 4). The number of exons per gene decreases steadily beyond it.*

The majority of genes tend to be formed by a relatively low number of exons (Figure 1). 1424 are formed by 4 exons.

Question 2

```
df$intron_length <- df$genome_length - df$mrna_length
head(df)
```

```
##       name mrna_length genome_length exon_count intron_length
## 1   PP8961        2596          2596          1             0
## 2 FLJ00038         794          2615          6          1821
## 3    OR4F5         918           918          1             0
## 4    OR4F3         937           937          1             0
```

```
## 5    OR4F16          937           937          1            0
## 6    SAMD11         2555         18842         14        16287
```

Basically, the total length of introns for each gene is obtained by substracting the length of the mRNA (the exon length in this case) from the entire genome length.

Question 3

On the histograms, the number of bins should be exactly the same, and the x-axis should have the same scale. Comment the plot – are exons larger than introns or vice versa?

```
plot_grid(plot_grid(cols_list[[1]], cols_list[[2]], ncol = 2, labels = "AUTO"),
          mylegend, nrow = 2, rel_heights = c(1, 0.1))
```

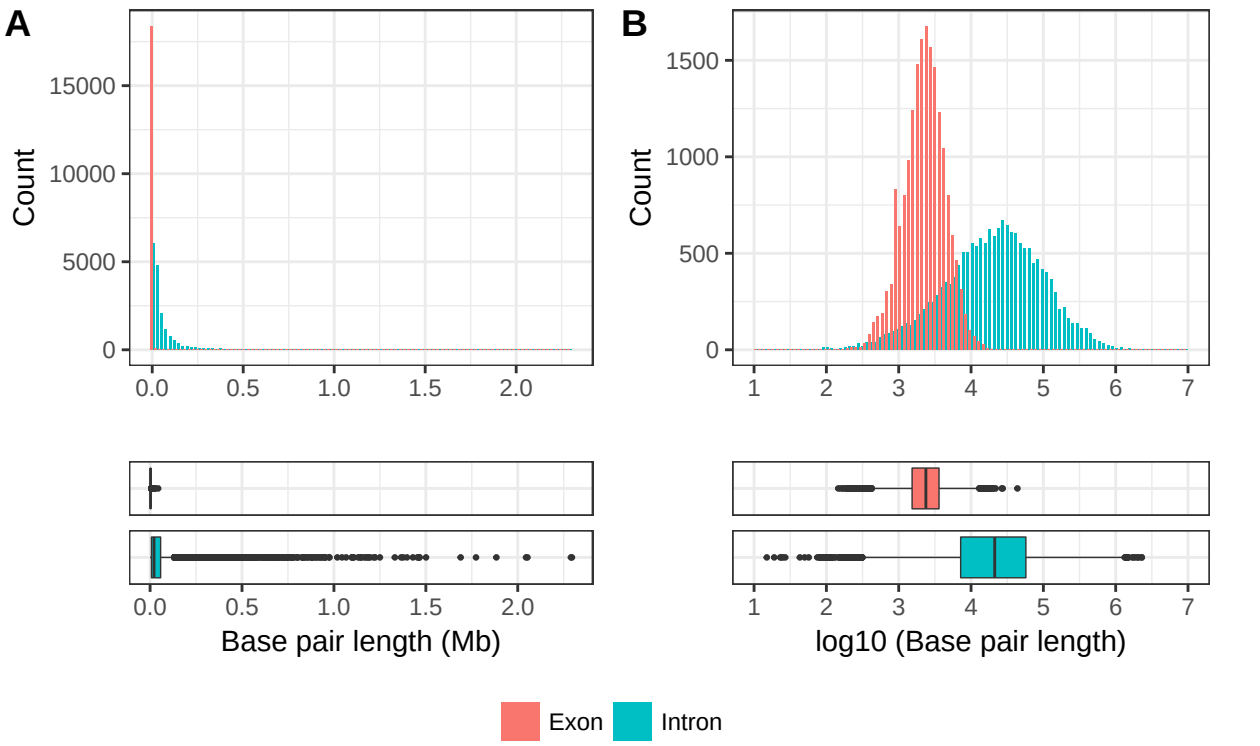

Figure 2. Distribution of intron and exons lengths in linear (A) and log10 (B) scale. *As can be seen in the histograms (top) and the boxplots (bottom), the median of the introns is one order of magnitude bigger than the median of the exons.*

The histograms and box-plots in Figure 2.B are presented with a logarithmic scale in x-axis in order to clearly appreciate the differences between the two subsets. They show that while most of exons (red) tend to have shorter lengths -with a peak around 2 kB-, the introns (blue) have a more right-tailed distribution, with generally longer lengths, covering an extremly wide span.

Question 4

We need to test the difference of the means of both distributions. The Student's T test may be used if a normal distribution can be assumed. Otherwise, only the corresponding non parametric test ought to be used (Wilcoxon test). In order to test normality, a Q-Q plot between the observed lengths and the normal distribution was drawn.

```
lengths.only <- data.frame(exon.length = df$mrna_length, intron.length = df$intron_length)
exon.mean <- mean(lengths.only$exon.length)
exon.sd <- sd(lengths.only$exon.length)
normal.quantiles <- qnorm(((1:nrow(df)) - 0.5) / nrow(df), mean = exon.mean, sd = exon.sd)
exon.length <- sort(lengths.only$exon.length)
new_set <- data.frame(normal.quantiles, exon.length)
slope <- diff(normal.quantiles) / diff(exon.length)

Exon.qqplot <- ggplot(data = new_set,
                      mapping = aes(x = normal.quantiles, y = exon.length / 1e3)) +
  geom_point() + labs(size = "Nitrogen",
                      x = "Normal Quantiles",
                      y = "Exon Lengths (Kb)",
                      title = "QQ-plot ~ Exon Lengths") +
  geom_abline(intercept = 0, slope = 1, color="red", linetype="dashed", size=1.0)

intron.mean <- mean(lengths.only$intron.length)
intron.sd <- sd(lengths.only$intron.length)
normal.quantiles <- qnorm(((1:nrow(df)) - 0.5) / nrow(df), mean = intron.mean, sd = intron.sd)
intron.length <- sort(lengths.only$intron.length)
new_set <- data.frame(normal.quantiles, intron.length)
slope <- diff(normal.quantiles) / diff(intron.length)

Intron.qqplot <- ggplot(data = new_set,
                        mapping = aes(x = normal.quantiles,y = intron.length / 1e6)) +
  geom_point() + labs(size = "Nitrogen",
                      x = "Normal Quantiles",
                      y = "Intron Lengths (Mb)",
                      title = "QQ-plot ~ Intron Lengths") +
  geom_abline(intercept = 0, slope = 1, color="red", linetype="dashed", size=1.0)

plot_grid(Exon.qqplot, Intron.qqplot)
```

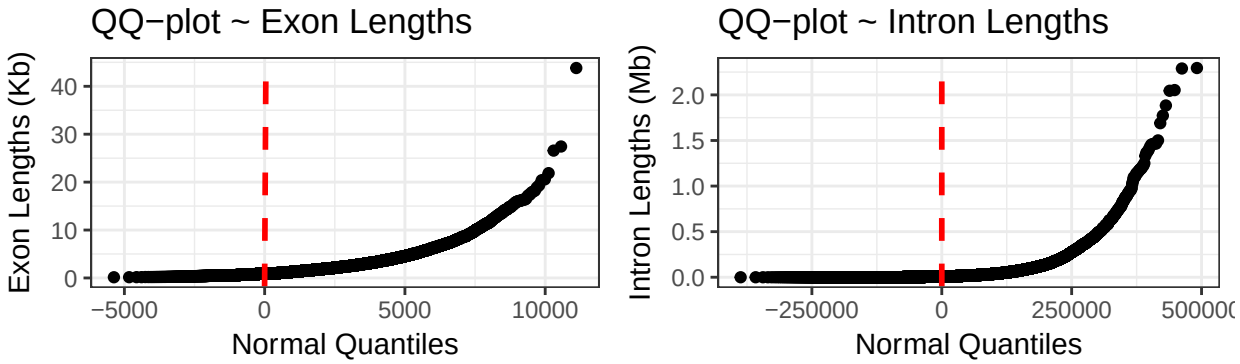

Figure 3 *QQplots of exon and intron length against normal distribution. Each point in the two plots above carries an observed component of intron or exon length (x-axis), and an expected component drawn from a normal distribution (y-axis) with a mean and standard deviation equal to the that of the intron or exon length sets. Both plots show significant deviations from the abline which maps evenly increasing values of a normal distribution, to itself.*

As we are comparing two data subsets that are not following a normal distribution (as seen in Figure 4), we have chosen to perform a Wilcoxon test to investigate if there is a significant difference in the lengths of the introns and exons of the genes of our data set. Our null hypothesis is that there is no significant difference in the U-statistic between the length of the exons and introns of the data set.

```r
wilcox.test(df$mrna_length, df$intron_length, alternative = "two.sided")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  df$mrna_length and df$intron_length
## W = 58458000, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

As our p-value is $< 2.2e\text{-}16$, it is below the significance threshold of 0.05, we can reject the null hypothesis and accept the alternative hypothesis, that is, there is a significant difference in the U-statistic between the length of the exons and introns. In addition to the previous observations of the lengths of introns and exons, we can conclude that the introns are significantly longer than the exons of the data set.

Question 5

In order to determine whether total exons length is more correlated to the total intron length than the number of exons, we have calculated the Pearsons' correlation coefficient for both cases.

```r
r1 <- round(cor(df$mrna_length, df$intron_length, method = "pearson"), digits = 2)
r2 <- round(cor(df$mrna_length, df$exon_count, method = "pearson"), digits = 2)

par(mfrow = c(1,2))
model_1 <- lm(df$intron_length / 1e6 ~ df$mrna_length)
plot(df$mrna_length, df$intron_length / 1e6, pch = 19, xlab = "mRNA length", ylab = "Introns Length (Mb)
abline(model_1, col = "blue")
text(x = 30000, y = 1.5, labels = paste("r2 = ", r1, sep = ""))

model_2 <- lm(df$exon_count ~ df$mrna_length)
plot(df$mrna_length, df$exon_count, pch = 19, xlab = "mRNA length", ylab = "Exon count")
abline(model_2, col = "blue")
text(x = 30000, y = 150, labels = paste("r2 = ", r2, sep = ""))
```

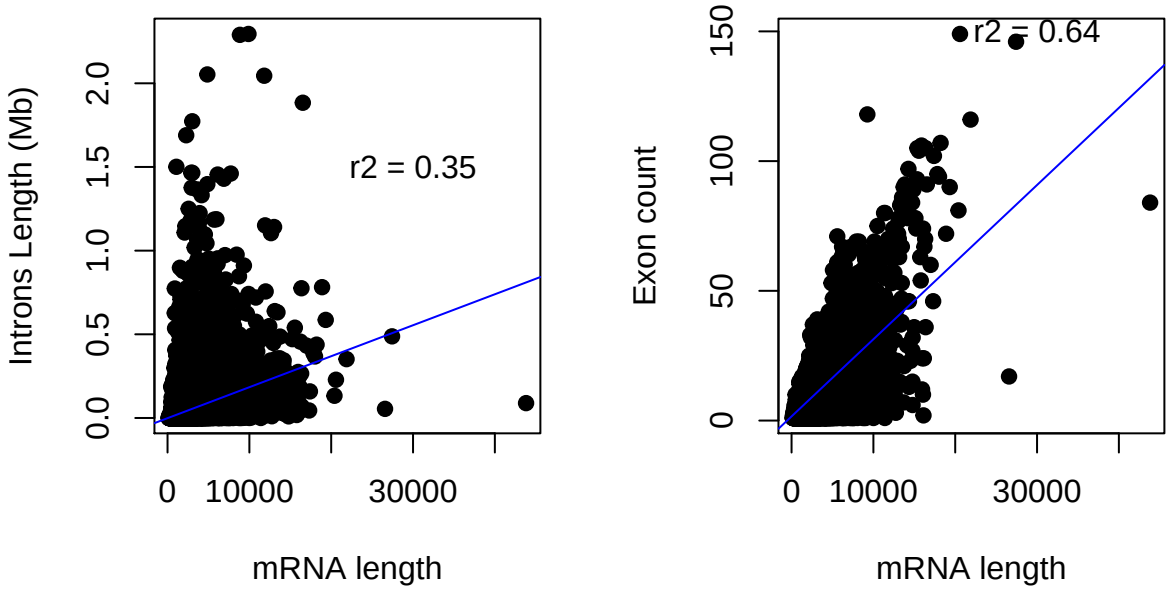

Figure 4 *Scatterplot of the studied variables showing regression lines and Pearson correlation coefficient. The cofficient sign (+/-) and range (from -1 to +1) indicates the direction of monotonicity and degree of*

linearity of the two variables under question, respectively.

Based on the correlation scores of 0.35 for exon VS intron length and 0.64 for exon length VS number of exons and the scatter plots, it can be concluded that whereas there is a positive correlation in both cases, they are not very strong. It seems that the exon´s length is more correlated with the number of exons than with the length of the introns of the genes belonging our data set.

Question 6

```
print(df[which.max(df$mrna_length), c(1,2,4)], row.names = FALSE)
```

```
##   name mrna_length exon_count
##  MUC16       43815         84
```

As can be seen above, the gene that has the longest total exon length is MUC16, with a length of 43815 base pairs and 84 exons.

Question 7

```
count_genes <- function( df, x1 = 0, x2 = max(df$mrna_length))
{
 total.mrna <- length(df$name)
 mrna.interval <- sum(df$mrna_length >= x1 & df$mrna_length <= x2)
 mrna.fraction = mrna.interval / total.mrna
 return ( mrna.fraction * 100)
}
```

Test this function with the mRNA lengths using the the five settings below:

- Using the default of x1 and x2;
- Using the default of x2 and set x1=10000;
- x1=1000 and x2=10000;
- x1=100 and x2=1000;
- x1=0 and x2=100.

Results:

```
x1 <- c(0, 1e4, 1e3, 100, 0)
x2 <- c(max(df$mrna_length), max(df$mrna_length), 1e4, 1e3, 100)
sapply(1:5, function(x) count_genes(df, x1[x], x2[x]))
```

```
## [1] 100.000000   1.130402  87.349235  11.541998   0.000000
```