Submission File

mcgrathjes

2025-10-06

Part 1

Part 1 - Gene expression mean analysis

The file "gene_expression.tsv" contains RNA-seq count data for three samples of interest. The following report will read in files from a public github repository https://github.com/ghazkha/Assessment4.git, configure rows and columns and analyse the data for several parameters.

1. Read in the file and make gene identifiers the row names

Using raw data from https://github.com/ghazkha/Assessment4.git, we will download the file "gene_expression.tsv" and set the row names as the gene identifiers which are listed in the first column `row.names=1` under the heading "Name_Description", eg. ENSG00000223972.5_DDX11L1.

The following table shows values for the first six genes.

```
##                               GTEX.1117F.0226.SM.5GZZ7 GTEX.1117F.0426.SM.5EGHI
## ENSG00000223972.5_DDX11L1                            0                        0
## ENSG00000227232.5_WASH7P                           187                      109
## ENSG00000278267.1_MIR6859-1                          0                        0
## ENSG00000243485.5_MIR1302-2HG                        1                        0
## ENSG00000237613.2_FAM138A                            0                        0
## ENSG00000268020.3_OR4G4P                             0                        1
##                               GTEX.1117F.0526.SM.5EGHJ
## ENSG00000223972.5_DDX11L1                            0
## ENSG00000227232.5_WASH7P                           143
## ENSG00000278267.1_MIR6859-1                          1
## ENSG00000243485.5_MIR1302-2HG                        0
## ENSG00000237613.2_FAM138A                            0
## ENSG00000268020.3_OR4G4P                             0
```

2. Make a new column which is the mean of the other columns

Column 4 "Mean_Count" is added to the table with rounding to 3 significant figures and saved as a new variable.

The following table shows values for the first six genes.

```
##                               GTEX.1117F.0226.SM.5GZZ7 GTEX.1117F.0426.SM.5EGHI
## ENSG00000223972.5_DDX11L1                            0                        0
## ENSG00000227232.5_WASH7P                           187                      109
## ENSG00000278267.1_MIR6859-1                          0                        0
## ENSG00000243485.5_MIR1302-2HG                        1                        0
## ENSG00000237613.2_FAM138A                            0                        0
## ENSG00000268020.3_OR4G4P                             0                        1
##                               GTEX.1117F.0526.SM.5EGHJ Mean_Count
## ENSG00000223972.5_DDX11L1                            0      0.000
## ENSG00000227232.5_WASH7P                           143    146.000
## ENSG00000278267.1_MIR6859-1                          1      0.333
## ENSG00000243485.5_MIR1302-2HG                        0      0.333
## ENSG00000237613.2_FAM138A                            0      0.000
## ENSG00000268020.3_OR4G4P                             0      0.333
```

3. List the 10 genes with the highest mean expression

Using the new Mean_Count variable, we create a new data frame subset called "mean_table" comprised of gene identifiers and their mean count values. We then order the data from largest to smallest mean value and display the top 10 genes with highest mean expression.

```
##                             Gene Mean_Count
## 56179  ENSG00000198804.2_MT-CO1     529000
## 56191  ENSG00000198886.2_MT-ND4     514000
## 56186  ENSG00000198938.2_MT-CO3     505000
## 56169  ENSG00000198888.2_MT-ND1     404000
## 56185 ENSG00000198899.2_MT-ATP6     330000
## 56198  ENSG00000198727.2_MT-CYB     302000
## 56173  ENSG00000198763.3_MT-ND2     284000
## 16618   ENSG00000211445.11_GPX3     270000
## 56182  ENSG00000198712.1_MT-CO2     266000
## 18688 ENSG00000156508.17_EEF1A1     232000
```

4. Determine the number of genes with a mean <10

We will create a subset called "low_mean" that contains all data with a Mean_count of less than 10. By counting the rows in this subset, we can see that there are 35,988 genes with a mean <10.

```
## [1] 35988
```

5. Make a histogram plot of the mean values

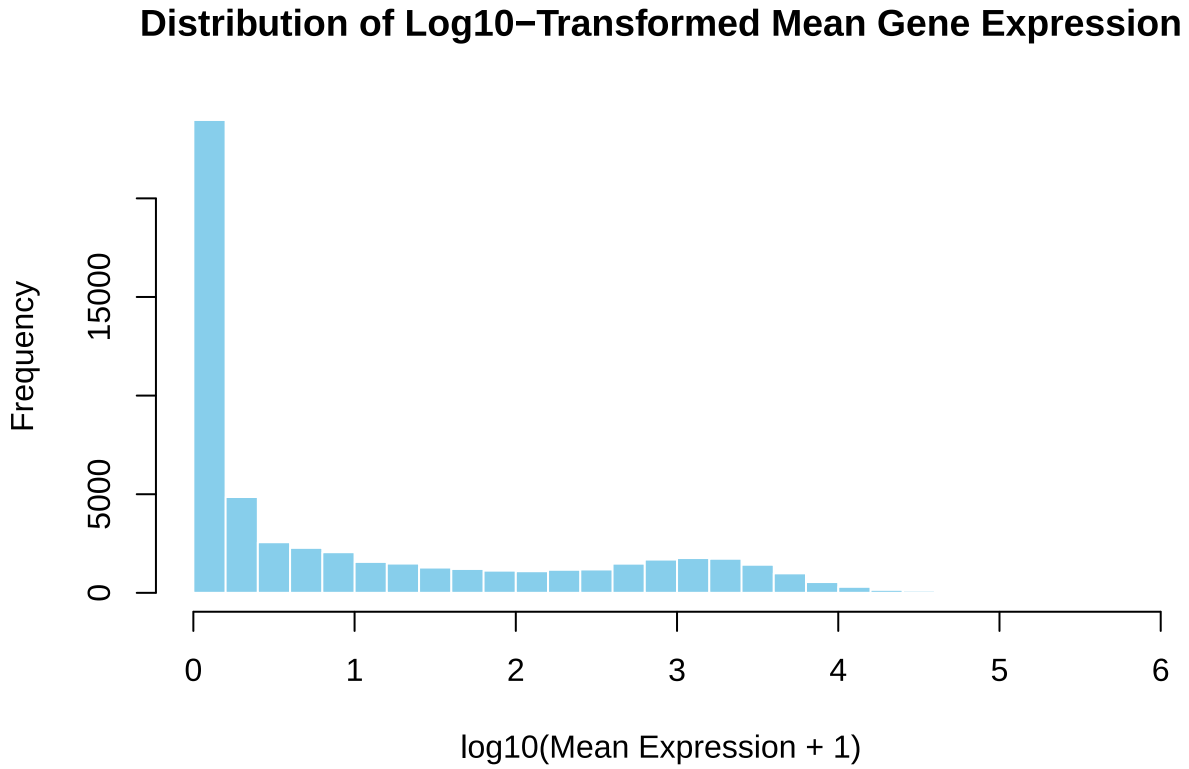
Considering the large number of genes with a mean <10, we evaluate the summary statistics of numeric vector "Mean_Count" to assess appropriate histogram parameters.

```
##  num [1:56200] 0 146 0.333 0.333 0 0.333 0.667 1 1.67 1 ...
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##      0.00      0.00      1.33    827.17     89.00 529000.00
```

We can see that there is a large range of values from 0.00 to 529,000 in this data set. There is a significant right skew demonstrated by the lower half of the genes (Min-Median) having a range of 0.00-1.33 and the upper half of the genes (Median-Max) extending from 1.33 to 529,000.

Due to these characteristics, it is appropriate to log10 transform the data to visually display a histogram of this data as described by Love *et al.* (2022). As we have known values of 0 in our data set, we add "+1" to the log transformation.



Part 1 - Tree growth calculations

The file "growth_data.csv" contains tree circumference measurements for two growth sites, "control site" and "treatment site" as measured over 5 year intervals.

6. Import this csv file into an R object. What are the column names?

Using raw data file from public repository previously listed, the "growth_data" csv file is imported to the project for analysis.

The column names are listed as follows:

```
## Site
## TreeID
## Circumf_2005_cm
## Circumf_2010_cm
## Circumf_2015_cm
## Circumf_2020_cm
```

7. Calculate the mean and standard deviation of tree circumference at the start and end of the study at both sites.

Using column 3 “Circumf_2005_cm” as the **start of the study**, and column 6 “Circumf_2020_cm” as the **end of the study**, we calculate the mean and standard deviation grouped by the two study sites: “northeast” and “southwest” using **tapply**.

Table 1: Mean and SD of tree circumference at start and end of study

	Site	Mean_2005	SD_2005	Mean_2020	SD_2020
northeast	northeast	5.292	0.9140267	54.228	25.22795
southwest	southwest	4.862	1.1474710	45.596	17.87345

Start of the study - 2005 The mean tree circumference at the start of the study was 5.29cm with a standard deviation of 0.91 at the northeast site.

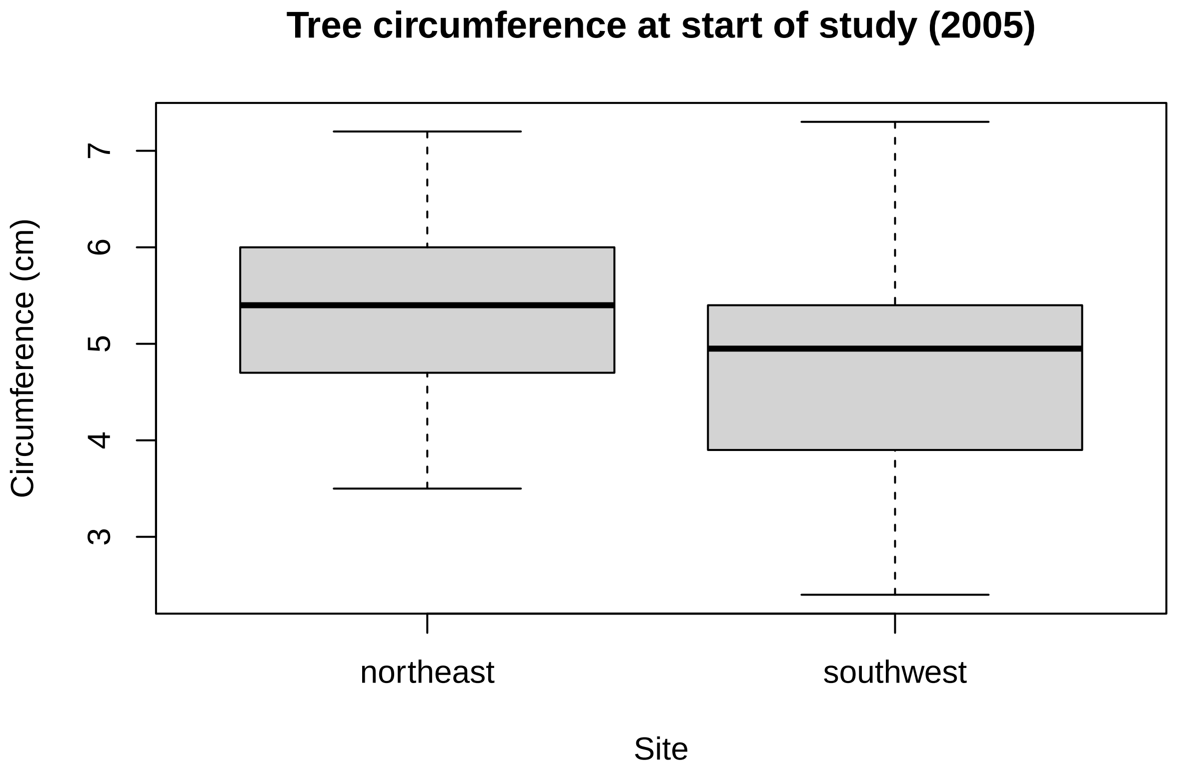
The mean circumference at the start of the study was 4.86cm with a standard deviation of 1.14 at the southwest site.

End of the study - 2020 The mean tree circumference at the end of the study was 54.23cm with a standard deviation of 25.23 at the northeast site.

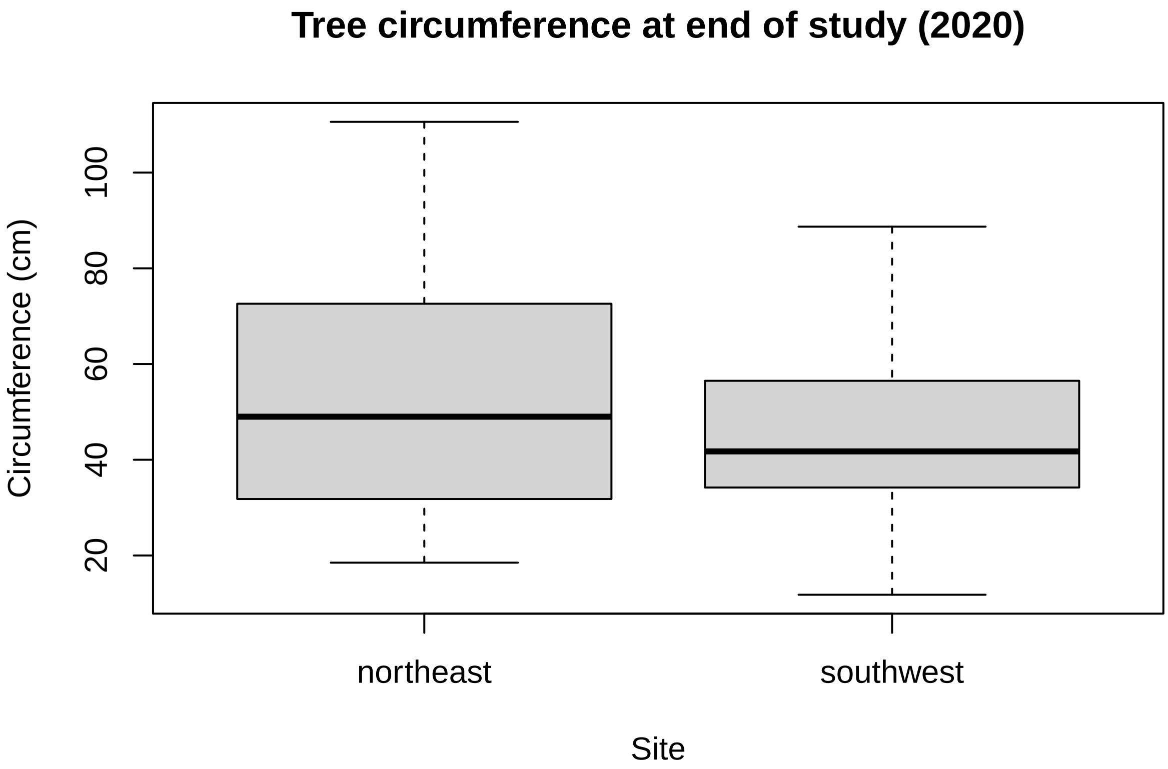
The mean circumference at the end of the study was 45.60cm with a standard deviation of 17.87 at the southwest site.

8. Make a box plot of tree circumference at the start and end of the study at both sites.

The following two boxplots show the tree circumference at the start of the study in 2005 at the northeast and southwest sites.



The following two boxplots show the tree circumference at the end of the study in 2020 at the northeast and



southwest sites.

9. Calculate the mean growth over the last 10 years at each site.

We will first calculate the total growth between 2010 and 2020 (last 10 years).

We can then calculate the mean for each site using **tapply**.

```
## northeast southwest
##     42.94     35.49
```

The mean growth over the last 10 years at the northeast site was 42.94cm which was greater than the southwest site which had a mean growth of 35.49cm.

10. Use the t.test to estimate the p-value that the 10 year growth is different at the two sites.

Using a two-sample t-test, we evaluate the 10 year growth difference to get a p-value of:

```
## [1] 0.06229256
```

As p = 0.06 which is greater than 0.05, the growth difference is **not** considered to be significantly different between the two sites.

Part 2 - Examining biological sequence diversity

This report will analyse the biological sequence features of *Salmonella enterica subsp. enterica serovar Weltevreden (GCA_005518735)* and compare it against *Escherichia coli str. K-12 substr. MG1655 str. K12 (GCA_000005845)* in order to examine the biological sequence diversity of the two organisms.

1. Download the whole set of coding DNA sequences for E. coli and your organism of interest. How many coding sequences are present in these organisms? Present this in the form of a table. Describe any differences between the two organisms.

We will be downloading *S. Weltevreden* and *E. coli* genomic sequence data from one of the biggest databases of genomic data - Ensembl https://bacteria.ensembl.org/index.html. There are many E. coli strains available in Ensembl, but as previously described, we will focus today on “Escherichia coli str. K-12 substr. MG1655 str. K12 (GCA_000005845)”.

We will download the sequences into R with the **seqinr** package and decompress them with the **R.utils** package.

Load *E. coli* and *S. Weltevreden* genomes, unzip and list the files to confirm successful:

```
##  [1] "applied-bioinformatics-assessment4.Rproj"
##  [2] "ecoli_cds.fa"
##  [3] "ecoli_cds.fa.gz"
##  [4] "LICENSE"
##  [5] "Part_1"
##  [6] "Part_2"
##  [7] "raw-data"
##  [8] "README.md"
##  [9] "rsconnect"
## [10] "Submission_file.html"
## [11] "Submission_file.Rmd"
## [12] "Submission_file_files"
## [13] "sweltevreden_cds.fa"
## [14] "sweltevreden_cds.fa.gz"
```

We will now read in the fasta files using **seqinr::read.fast()** to create a “cds” list, assess how many coding sequences are present using **length()** for each organism, and finally create a table using the **knitr::kable** function of R to compare the two files.

Table 2: Number of coding sequences in each organism

File	Num_CDS
ecoli_cds.fa	4239
sweltevreden_cds.fa	4585

We can see that there are **4239** coding sequences in *E. coli* and **4585** coding sequences in *S. weltevreden* demonstrating that S. weltevreden has more coding sequences than E. coli.

This difference may be indicative of genome size variation, strain-specific genes, differing regulatory systems or mutations (insertions, deletions, or substitutions of nucleotides), which lead to genetic variation over evolutionary time (Winfield and Groisman, 2004).

2. How much coding DNA is there in total for these two organisms? Present this in the form of a table. Describe any differences between the two organisms.

By extracting the gene lengths for each organism using the **summary()** function, we can convert the first column of the output into a numeric vector (e.g., `ecoli_len <- as.numeric(summary(ecoli_cds)[,1])`), before calculating the total length of all genes by summing them.

Table 3: Total coding DNA in sample organism genomes

Organism	Total_Coding_DNA_bp
E. coli	3978528
S. weltevreden	4294851

We can see that E. coli has **3.98Mbp** of coding DNA and S. weltevreden has **4.29Mbp**.

Whilst both organisms have similar total coding DNA lengths, there are more bp's present in the S. weltevreden genome by approximately 0.316Mbp which may reflect additional genes or a slightly larger genome.

3. Calculate the length of all coding sequences in these two organisms. Make a boxplot of coding sequence length in these organisms. What is the mean and median coding sequence length of these two organisms? Describe any differences between the two organisms.

By using the total length object of each respective organism calculated in the previous section, we can add these totals together to produce a new `combined_total` object.

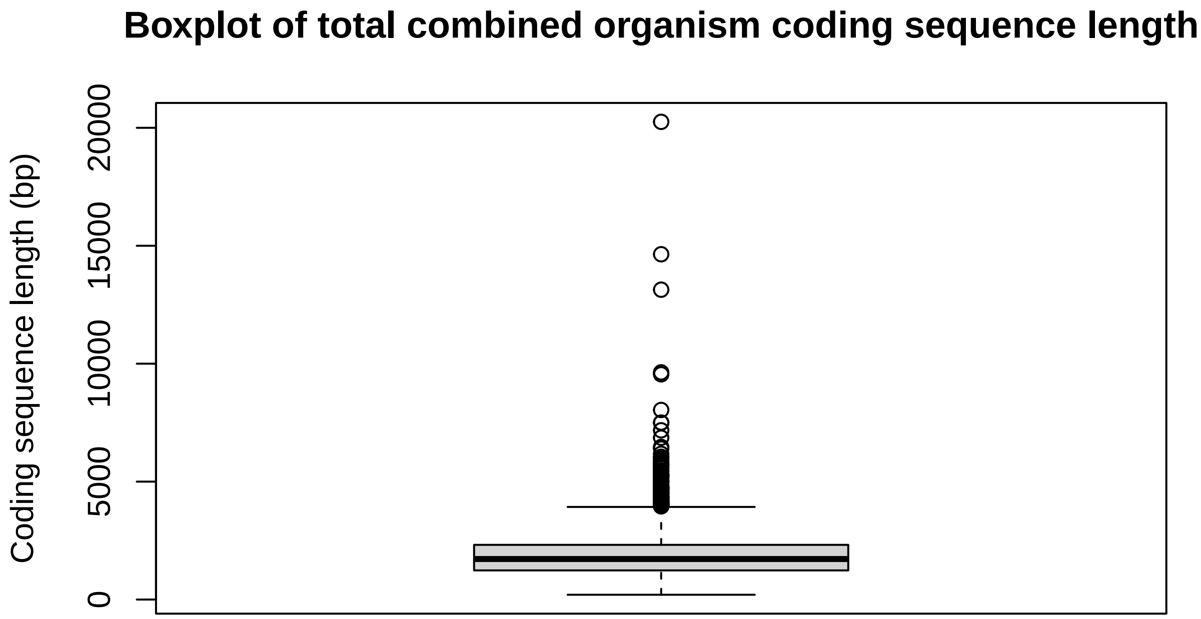
```
combined_total <- ecoli_total + sweltevreden_total
combined_total
```

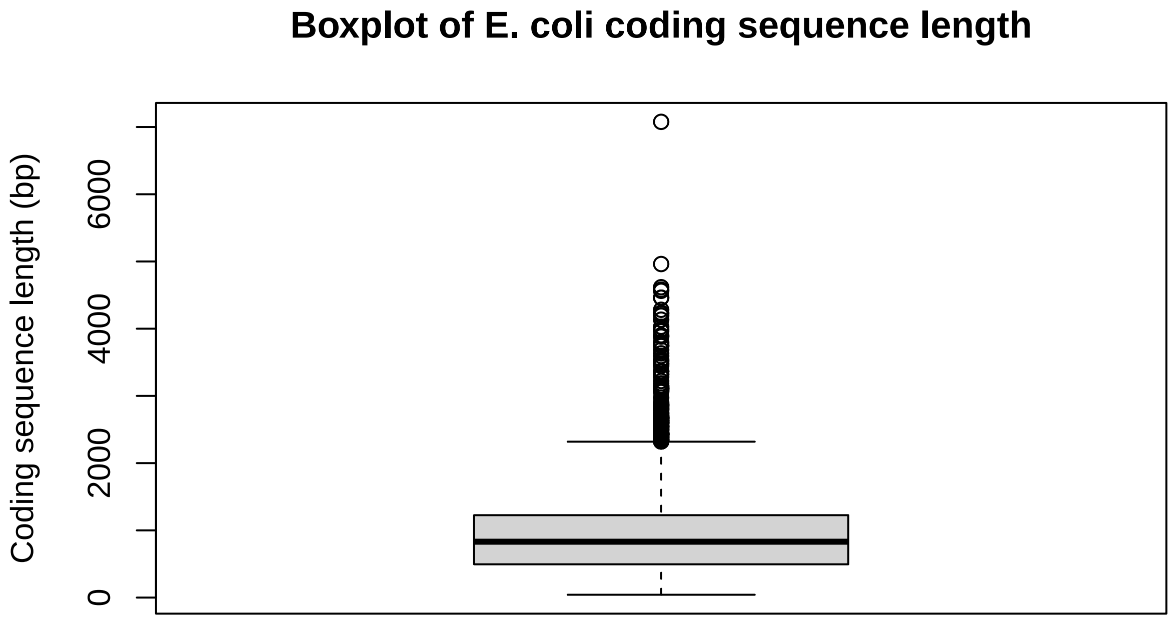
```
## [1] 8273379
```

This calculation demonstrates a combined total length of all coding sequences in our two organism genomes of **8.27Mbp**.

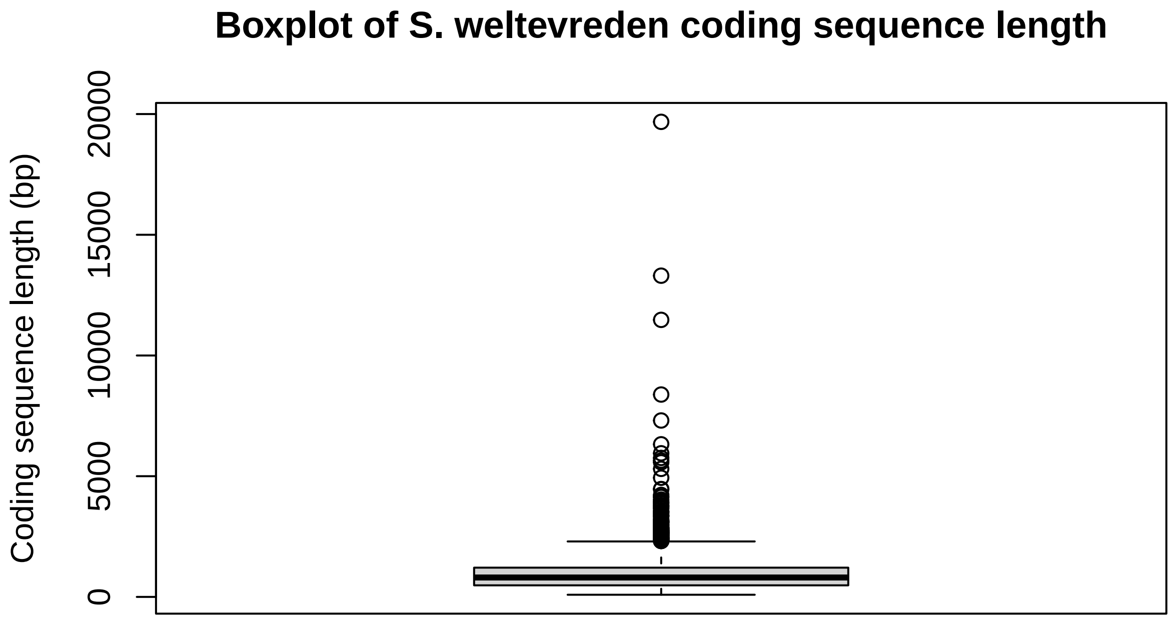
This combined total is represented by the following boxplot, with individual organism coding sequence length distributions represented below:

```
## Warning in ecoli_len + sweltevreden_len: longer object length is not a multiple
## of shorter object length
```


Boxplot of total combined organism coding sequence length
Coding sequence length (bp)
0
5000
10000
15000
20000



Boxplot of E. coli coding sequence length



The distribution of these boxplots show a right skew with several outliers that lengthen the whiskers towards the upper maximum limits.

These observations are supported by calculating the mean and median values of individual organisms, and a new combined vector which is created by:

```
# Combine the lengths into a single vector
combined_len <- c(ecoli_len, sweltevreden_len)
```

```
# Calculate mean combined
mean_len <- mean(combined_len)
mean_len
```

```
## [1] 937.5996
```

```
# Calculate median combined
median_len <- median(combined_len)
median_len
```

```
## [1] 816
```

The mean coding sequence length of these two organisms combined is **937.6bp** and the median coding sequence length is **816bp**.

```r
# E. coli
ecoli_len <- as.numeric(summary(ecoli_cds)[,1])
mean(ecoli_len)
```

```
## [1] 938.5534
```

```r
median(ecoli_len)
```

```
## [1] 831
```

```r
# S. weltevreden
sweltevreden_len <- as.numeric(summary(sweltevreden_cds)[,1])
mean(sweltevreden_len)
```

```
## [1] 936.7178
```

```r
median(sweltevreden_len)
```

```
## [1] 804
```

Individually, E.coli has a mean sequence length of **938.6bp** and median of **831bp**, whilst S. weltevreden mean is slightly lower at **936.7bp** and lower median of **804bp**.

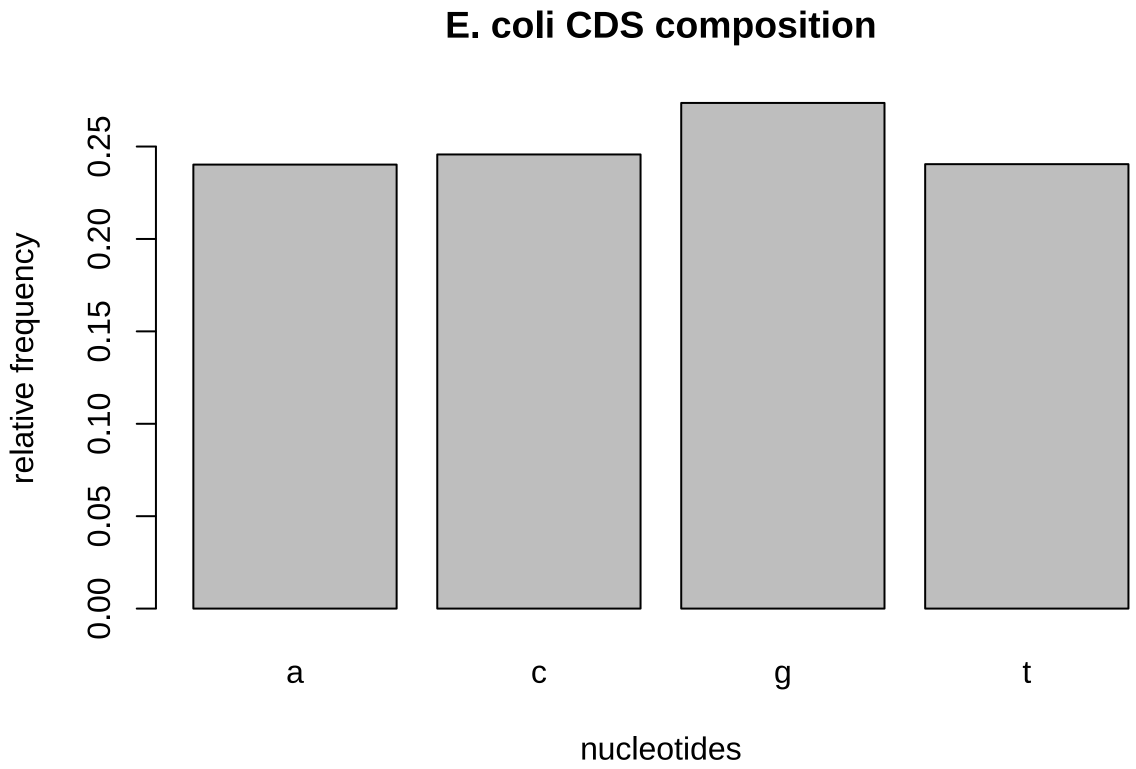
This indicates that although both organisms have similar average gene sizes, S. weltevreden may possess a greater number of shorter coding sequences. Such variation could reflect differences in genome organisation, gene density, or the presence of smaller accessory genes that contribute to environmental adaptation or virulence (McClelland et al., 2001).

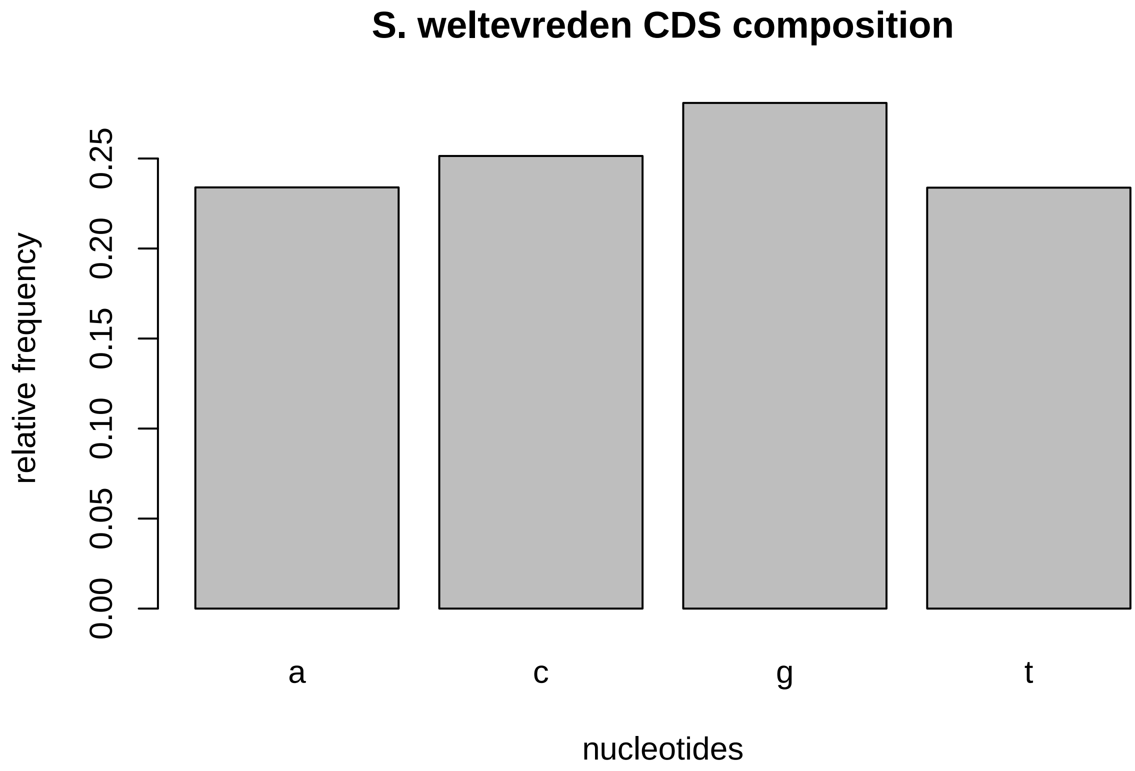
4. Calculate the frequency of DNA bases in the total coding sequences for both organisms. Perform the same calculation for the total protein sequence. Create bar plots for nucleotide and amino acid frequency. Describe any differences between the two organisms.

In order to compute the frequency of DNA bases in the organism coding sequence data sets, we first need to unlist both cds objects, and store the sequences in a "_dna" variable.

We then calculate the frequency of each nucleotide using `<organism>_composition <- count(<organism>_dna, 1)`, which stores these total counts in _composition. We can further examine these counts as relative frequency _proportion using `<organism>_proportion <- <organism>_composition/sum(<organism>_composition)`.

We then visualize the relative frequencies with `barplot(<organism>_proportion, xlab="nucleotides", ylab="frequency", main="<organism> CDS composition")`. This creates a bar plot where the x-axis represents the nucleotides (A, C, G, T), the y-axis shows their relative frequencies, with a plot title detailing which organism is represented.


E. coli CDS composition
relative frequency
0.00
0.05
0.10
0.15
0.20
0.25
a
c
g
t
nucleotides



We can see that both organisms have a very similar composition with the highest proportion of nucleotides being **Guanine (G)** of greater than 25%.

Dear markers: I have made an error somewhere so my objects are not containing the correct information from this point on. I have tried and tried to troubleshoot and have left off below trying to read the cds back in, translate and then unlist it again. I have included the commands to be used for the next sections but note there is incorrect output.

Thank you for your consideration in part marks.

Below we have read in the cds files for each organism and used **lapply**'s **translate** function to create a list of protein sequences. We attempt to unlist again to replicate the previous successful calculation step to count and quantify the frequency of amino acids.

```
ecoli_cds <- seqinr::read.fasta("ecoli_cds.fa")
ecoli_prot <- lapply(ecoli_cds, translate)
ecoli_prot_seq <- unlist(ecoli_prot)
ecoli_aa <- count(ecoli_prot,1)
ecoli_aa_prop <- ecoli_aa/sum(ecoli_aa)
ecoli_aa_prop
```

```
##
## a c g t
##
```

```
#barplot(ecoli_aa_prop,
        #xlab="amino acids",
        #ylab="relative frequency",
        #main="E. coli amino acid composition")


sweltevreden_cds <- seqinr::read.fasta("sweltevreden_cds.fa")
sweltevreden_prot <- lapply(sweltevreden_cds, translate)
sweltevreden_prot_seq <- unlist(sweltevreden_prot)
sweltevreden_aa <- count(sweltevreden_prot_seq,1)
sweltevreden_aa_prop <- sweltevreden_aa / sum(sweltevreden_aa)
sweltevreden_aa_prop
```

```
##
## a c g t
##
```

```
#barplot(sweltevreden_aa_prop,
        #xlab="amino acids",
        #ylab="relative frequency",
        #main="S. weltevreden amino acid composition")
```

5. Create a codon usage table and quantify the codon usage bias among all coding sequences. Describe any differences between the two organisms with respect to their codon usage bias. Provide charts to support your observations.

Using the **Seqinr** packages command **uco**, we can create a codon usage table for our organisms.

We can then quantify the codon usage bias by using `index="rscu"` which computes the Relative Synonymous Codon Usage (RSCU) values and expresses how preferred a codon of a particular amino acid is compared to the other codons which also encode it.

E. coli

```
uco(ecoli_cds)
```

```
##
## aaa aac aag aat aca acc acg act aga agc agg agt ata atc atg att caa cac cag cat
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## cca ccc ccg cct cga cgc cgg cgt cta ctc ctg ctt gaa gac gag gat gca gcc gcg gct
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## gga ggc ggg ggt gta gtc gtg gtt taa tac tag tat tca tcc tcg tct tga tgc tgg tgt
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## tta ttc ttg ttt
##   0   0   0   0
```

```
uco(ecoli_cds,index="rscu")
```

```
## aaa aac aag aat aca acc acg act aga agc agg agt ata atc atg att caa cac cag cat
##  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
## cca ccc ccg cct cga cgc cgg cgt cta ctc ctg ctt gaa gac gag gat gca gcc gcg gct
##  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
## gga ggc ggg ggt gta gtc gtg gtt taa tac tag tat tca tcc tcg tct tga tgc tgg tgt
##  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
## tta ttc ttg ttt
##  NA  NA  NA  NA
```

S. weltevreden

```r
uco(sweltevreden_cds)
```

```
##
## aaa aac aag aat aca acc acg act aga agc agg agt ata atc atg att caa cac cag cat
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## cca ccc ccg cct cga cgc cgg cgt cta ctc ctg ctt gaa gac gag gat gca gcc gcg gct
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## gga ggc ggg ggt gta gtc gtg gtt taa tac tag tat tca tcc tcg tct tga tgc tgg tgt
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## tta ttc ttg ttt
##   0   0   0   0
```

```r
uco(sweltevreden_cds,index="rscu")
```

```
## aaa aac aag aat aca acc acg act aga agc agg agt ata atc atg att caa cac cag cat
##  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
## cca ccc ccg cct cga cgc cgg cgt cta ctc ctg ctt gaa gac gag gat gca gcc gcg gct
##  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
## gga ggc ggg ggt gta gtc gtg gtt taa tac tag tat tca tcc tcg tct tga tgc tgg tgt
##  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
## tta ttc ttg ttt
##  NA  NA  NA  NA
```

6. In the organism of interest, identify 10 protein sequence k-mers of length 3-5 which are the most over- and under-represented k-mers in your organism of interest. Are these k-mers also over- and under-represented in E. coli to a similar extent? Provide plots to support your observations. Why do you think these sequences are present at different levels in the genomes of these organisms?