Assessment 4

Nicole Hinojosa, Saqeena Thapa, Vanshika

2024-10-03

Part 1: Importing files, data wrangling, mathematical operations, plots and saving code on GitHub.

Introduction

This report analyzes RNA-seq count data for gene expression and tree circumference measurements at two different sites over a 20-year period.

Task 1: RNA-seq count data for gene expression, high and low expression of 3 genes.

1.1 Read in the file "gene_expression.tsv", making the gene identifiers the row names. Show a table of values for the first six genes.

1) Load libraries

```
#RNA-seq Count Data Analysis
#Load necessary libraries
library(R.utils)
```

```
## Loading required package: R.oo

## Loading required package: R.methodsS3

## R.methodsS3 v1.8.2 (2022-06-13 22:00:14 UTC) successfully loaded. See ?R.methodsS3 for help.

## R.oo v1.26.0 (2024-01-24 05:12:50 UTC) successfully loaded. See ?R.oo for help.

##
## Attaching package: 'R.oo'

## The following object is masked from 'package:R.methodsS3':
##
##     throw

## The following objects are masked from 'package:methods':
##
##     getClasses, getMethods

## The following objects are masked from 'package:base':
##
##     attach, detach, load, save

## R.utils v2.12.3 (2023-11-18 01:00:02 UTC) successfully loaded. See ?R.utils for help.

##
## Attaching package: 'R.utils'
```

```
## The following object is masked from 'package:utils':
##
##     timestamp

## The following objects are masked from 'package:base':
##
##     cat, commandArgs, getOption, isOpen, nullfile, parse, warnings
```

```r
#use BiocManager::install("Biostrings") if it is not already installed in your Rstudio
library(Biostrings)
```

```
## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min

## Loading required package: S4Vectors

## Loading required package: stats4

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

## The following object is masked from 'package:R.oo':
##
##     trim

## Loading required package: XVector

## Loading required package: GenomeInfoDb

##
## Attaching package: 'Biostrings'

## The following object is masked from 'package:base':
##
##     strsplit
```

```r
library(seqinr)
```

```
##
## Attaching package: 'seqinr'

## The following object is masked from 'package:Biostrings':
##
##     translate

## The following object is masked from 'package:R.oo':
##
##     getName
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:seqinr':
##
##     count

## The following objects are masked from 'package:Biostrings':
##
##     collapse, intersect, setdiff, setequal, union

## The following object is masked from 'package:GenomeInfoDb':
##
##     intersect

## The following object is masked from 'package:XVector':
##
##     slice

## The following objects are masked from 'package:IRanges':
##
##     collapse, desc, intersect, setdiff, slice, union

## The following objects are masked from 'package:S4Vectors':
##
##     first, intersect, rename, setdiff, setequal, union

## The following objects are masked from 'package:BiocGenerics':
##
##     combine, intersect, setdiff, union

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)
library(readr)
library(tidyr)
```

```
##
## Attaching package: 'tidyr'

## The following object is masked from 'package:S4Vectors':
##
```

```
##     expand
```

```
## The following object is masked from 'package:R.utils':
##
##     extract
```

2) Read in the gene expression data

```r
#Download the data from the github link provided
URL = "https://raw.githubusercontent.com/ghazkha/Assessment4/refs/heads/main/gene_expression.tsv"
download.file(URL, destfile = "gene_expression.tsv")

# Read the downloaded TSV file into R
gene_expression <- read.table("gene_expression.tsv", header = TRUE, sep = "\t", row.names = 1)
```

3) 1st First 6 rows of the gene_expression data

```r
# View the first few rows of the data
head(n=6, gene_expression)
```

```
##                              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
```

```r
head( head (n=6, gene_expression))
```

```
##                              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
```

1.2 Make a new column which is the mean of the other columns. Show a table of values for the first six genes.

Calculate Mean Expression

```r
# Calculate the mean across the samples and add as a new column
gene_expression <- gene_expression %>%
  mutate(mean_expression = rowMeans(select(., everything())))
# Show a table of values for the first six genes including the mean
head(n=6, gene_expression)
```

```
##                               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_expression
## ENSG00000223972.5_DDX11L1                            0       0.0000000
## ENSG00000227232.5_WASH7P                           143     146.3333333
## ENSG00000278267.1_MIR6859-1                          1       0.3333333
## ENSG00000243485.5_MIR1302-2HG                        0       0.3333333
## ENSG00000237613.2_FAM138A                            0       0.0000000
## ENSG00000268020.3_OR4G4P                             0       0.3333333
```

1.3 List the 10 genes with the highest mean expression.

Identify Top 10 Genes

```r
# List the 10 genes with the highest mean expression
top_genes <- gene_expression %>%
  arrange(desc(mean_expression)) %>%
  head(10)
# Print the top genes
print(top_genes)
```

```
##                           GTEX.1117F.0226.SM.5GZZ7 GTEX.1117F.0426.SM.5EGHI
## ENSG00000198804.2_MT-CO1                    267250                  1101779
## ENSG00000198886.2_MT-ND4                    273188                   991891
## ENSG00000198938.2_MT-CO3                    250277                  1041376
## ENSG00000198888.2_MT-ND1                    243853                   772966
## ENSG00000198899.2_MT-ATP6                   141374                   696715
## ENSG00000198727.2_MT-CYB                    127194                   638209
## ENSG00000198763.3_MT-ND2                    159303                   543786
## ENSG00000211445.11_GPX3                     464959                    39396
## ENSG00000198712.1_MT-CO2                    128858                   545360
## ENSG00000156508.17_EEF1A1                   317642                    39573
##                           GTEX.1117F.0526.SM.5EGHJ mean_expression
## ENSG00000198804.2_MT-CO1                    218923        529317.3
## ENSG00000198886.2_MT-ND4                    277628        514235.7
## ENSG00000198938.2_MT-CO3                    223178        504943.7
## ENSG00000198888.2_MT-ND1                    194032        403617.0
## ENSG00000198899.2_MT-ATP6                   151166        329751.7
## ENSG00000198727.2_MT-CYB                    141359        302254.0
## ENSG00000198763.3_MT-ND2                    149564        284217.7
## ENSG00000211445.11_GPX3                     306070        270141.7
## ENSG00000198712.1_MT-CO2                    122816        265678.0
## ENSG00000156508.17_EEF1A1                   339347        232187.3
```

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

Count Genes with Low Expression (Mean < 10)

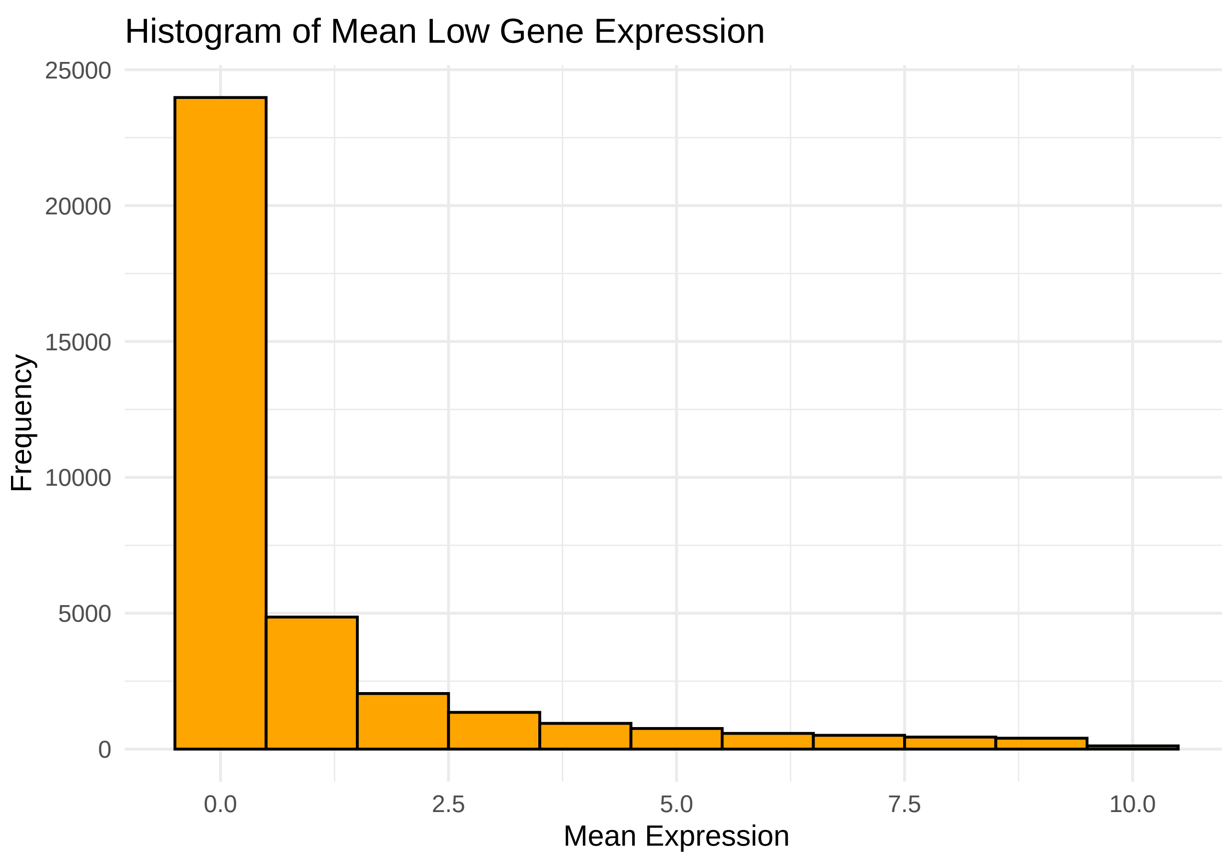
```
# Determine the number of genes with a mean < 10
num_genes_below_10 <- sum(gene_expression$mean_expression < 10)

# Print the number of genes
print(num_genes_below_10)
```

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

1.5 Make a histogram plot of the mean values and include it into your report.

Histogram of <10, Mean Values

```
# Make a histogram plot of the mean values
filtered_data <- gene_expression[gene_expression$mean_expression < 10, ]
ggplot(filtered_data, aes(x = mean_expression)) +
  geom_histogram( binwidth = 1, fill = "orange", color = "black") +
  labs(title = "Histogram of Mean Low Gene Expression", x = "Mean Expression", y = "Frequency") +
  theme_minimal()
```




```
# Save the plot to your report
ggsave("histogram_mean_low_gene_expression.png")
```

```
## Saving 6.5 x 4.5 in image
```

Task 2: Tree circumference measurements over 20 years.

2.1 Import "growth_data.csv" file into an R object. What are the column names?

Read Data to perform a Tree Circumference Data Analysis

```
# Read in the growth data
#Download the data from the github link provided
URL = "https://raw.githubusercontent.com/ghazkha/Assessment4/refs/heads/main/growth_data.csv"
download.file(URL, destfile = "growth_data.csv")

# Read the downloaded TSV file into R
growth_data <- read.csv("growth_data.csv")

head(growth_data)
```

```
##        Site TreeID Circumf_2005_cm Circumf_2010_cm Circumf_2015_cm
## 1 northeast   A012             5.2            10.1            19.9
## 2 southwest   A039             4.9             9.6            18.9
## 3 southwest   A010             3.7             7.3            14.3
## 4 northeast   A087             3.8             6.5            10.9
## 5 southwest   A074             3.8             6.4            10.9
## 6 northeast   A008             5.9            10.0            16.8
##   Circumf_2020_cm
## 1            38.9
## 2            37.0
## 3            28.1
## 4            18.5
## 5            18.4
## 6            28.4
```

```
# Show column names
cat("The column names are:", colnames(growth_data))
```

```
## The column names are: Site TreeID Circumf_2005_cm Circumf_2010_cm Circumf_2015_cm Circumf_2020_cm
```

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

Statistics

```
# Calculate mean and standard deviation for tree circumference
summary_stats <- growth_data %>%
  summarise(mean_start_2005_southwest = mean(Circumf_2005_cm[Site == "southwest"]),
    sd_start_2005_southwest = sd(Circumf_2005_cm[Site == "southwest"]),
    mean_start_2005_northeast = mean(Circumf_2005_cm[Site == "northeast"]),
    sd_start_2005_northeast = sd(Circumf_2005_cm[Site == "northeast"]),
    mean_end_2020_southwest = mean(Circumf_2020_cm[Site == "southwest"]),
    sd_end_2020_southwest = sd(Circumf_2020_cm[Site == "southwest"]),
    mean_end_2020_northeast = mean(Circumf_2020_cm[Site == "northeast"]),
    sd_end_2020_northeast = sd(Circumf_2020_cm[Site == "northeast"])
  )

# Print summary statistics
print(summary_stats)
```

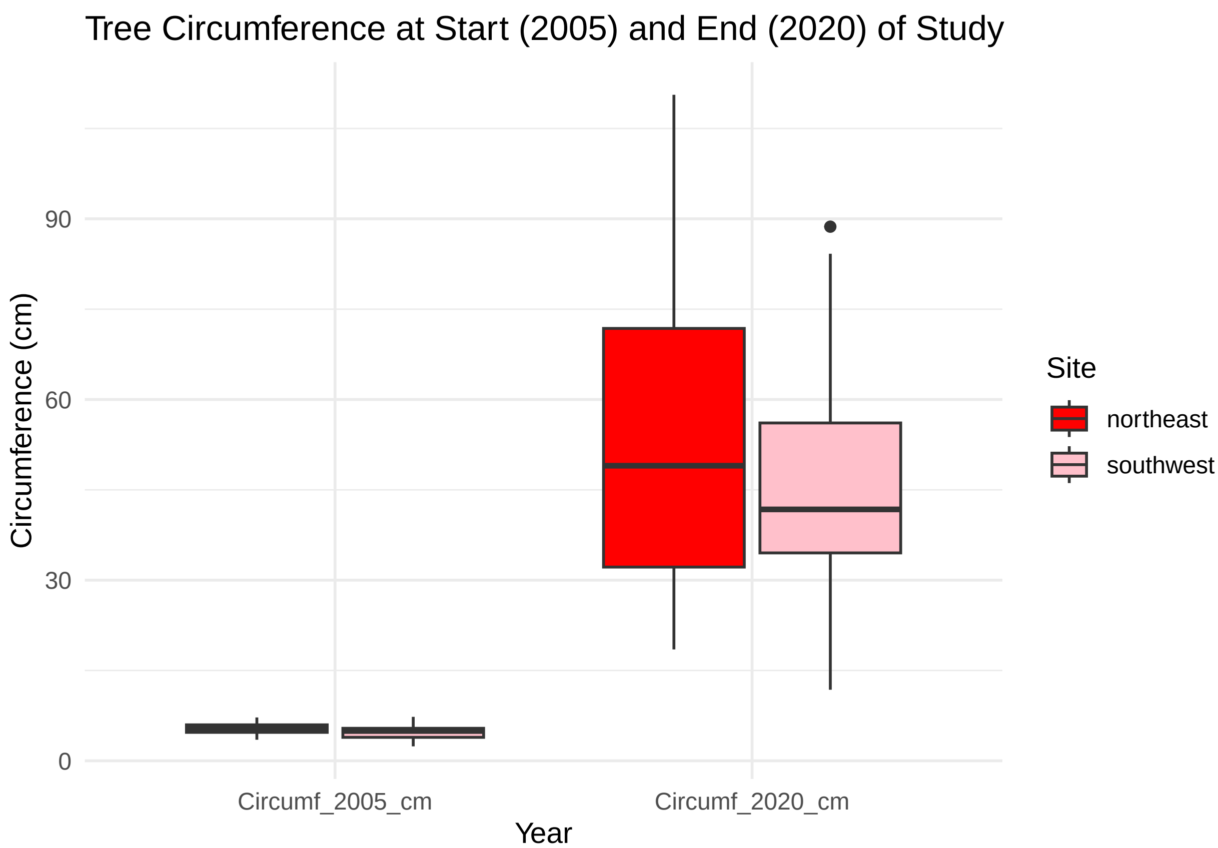
```
##   mean_start_2005_southwest sd_start_2005_southwest mean_start_2005_northeast
## 1                     4.862                1.147471                     5.292
```

```
##   sd_start_2005_northeast mean_end_2020_southwest sd_end_2020_southwest
## 1               0.9140267                  45.596              17.87345
##   mean_end_2020_northeast sd_end_2020_northeast
## 1                  54.228              25.22795
```

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

Boxplot of circumferences

```r
# Reshape data from wide to long format for Circumf_2005_cm and Circumf_2020_cm
long_data <- growth_data %>%
  select(Site, TreeID, Circumf_2005_cm, Circumf_2020_cm) %>%
  pivot_longer(cols = starts_with("Circumf"),
               names_to = "Year",
               values_to = "Circumference")

# Filter for only the start and end years
long_data <- long_data %>%
  filter(Year %in% c("Circumf_2005_cm", "Circumf_2020_cm"))

# Create a box plot
ggplot(long_data, aes(x = Year, y = Circumference, fill = Site)) +
  geom_boxplot() +
  labs(title = "Tree Circumference at Start (2005) and End (2020) of Study",
       x = "Year",
       y = "Circumference (cm)") +
  scale_fill_manual(values = c("northeast" = "red", "southwest" = "pink")) +
  theme_minimal()
```




```
# Save the box plot to your report
ggsave("boxplot_tree_circumference.png")
```

```
## Saving 6.5 x 4.5 in image
```

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

Mean Growth Calculation

```
# Calculate growth over the last 10 years for each tree
growth_data <- growth_data %>%
  mutate(Growth_10_years = Circumf_2020_cm - Circumf_2010_cm)

# Calculate mean growth at each site
mean_growth <- growth_data %>%
  group_by(Site) %>%
  summarise(mean_growth = mean(Growth_10_years, na.rm = TRUE), .groups = 'drop')

# Print the mean growth
print(mean_growth)
```

```
## # A tibble: 2 x 2
##   Site      mean_growth
##   <chr>           <dbl>
## 1 northeast        42.9
## 2 southwest        35.5
```

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

T-Test for Growth Difference

```
# Perform t-test to compare growth between sites
t_test_result <- t.test(Growth_10_years ~ Site, data = growth_data)

# Print t-test results
print(t_test_result)
```

```
##
##  Welch Two Sample t-test
##
## data:  Growth_10_years by Site
## t = 1.8882, df = 87.978, p-value = 0.06229
## alternative hypothesis: true difference in means between group northeast and group southwest is not
## 95 percent confidence interval:
##  -0.3909251 15.2909251
## sample estimates:
## mean in group northeast mean in group southwest
##                   42.94                   35.49
```

Interpretation: p-value: The p-value of 0.06229 suggests that the difference in mean growth between the two sites is not statistically significant at the conventional alpha level of 0.05. However, it is close to this threshold, indicating a potential trend toward significance.

Mean Comparison: The mean growth in the northeast (42.94 cm) is higher than that in the southwest (35.49 cm). This suggests that trees in the northeast experienced greater growth compared to those in the southwest over the last 10 years.

Confidence Interval: The confidence interval includes zero, which means we cannot conclusively say that there is a true difference in growth between the two sites. The upper limit (15.29 cm) indicates that, while the northeast shows higher growth, it is possible that the actual difference might be minimal or even negative.

Part 2: Examining biological sequence diversity

Introduction

This report compares the sequence features of *Streptacidiphilus jiangxiensis* (GCA_900109465) with *Escherichia coli*. *Escherichia coli* is a Gram-negative, rod-shaped bacterium commonly found in the intestines of warm-blooded organisms, playing a vital role in gut health. While some strains of *Escherichia coli* can cause illnesses, the bacterium is extensively used as a model organism in molecular biology due to its relatively simple genome and well-studied genetics. In contrast, *Streptacidiphilus jiangxiensis* is a Gram-positive bacterium isolated from acidic environments (Huang et al., 2004). This species is characterized by its unique metabolic pathways and adaptations to specific ecological niches, which may hold potential for applications in bioremediation or antibiotic development.

Questions:

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.

Sequences

```
# URLs for the coding DNA sequences
URL_Ecoli <- "https://ftp.ensemblgenomes.ebi.ac.uk/pub/bacteria/release-59/fasta/bacteria_117_collectio
```

```r
URL_Streptacidiphilus <- "https://ftp.ensemblgenomes.ebi.ac.uk/pub/bacteria/release-59/fasta/bacteria_5

# Downloading the sequences
download.file(URL_Ecoli, destfile = "e_coli_cds.fa.gz")
download.file(URL_Streptacidiphilus, destfile = "streptacidiphilus_cds.fa.gz")

#Decompress the files

gunzip("e_coli_cds.fa.gz")
gunzip("streptacidiphilus_cds.fa.gz")

# Reading the sequences
ecoli_seqs <- seqinr::read.fasta ("e_coli_cds.fa")
streptacidiphilus_seqs <- seqinr::read.fasta ("streptacidiphilus_cds.fa")
```

CDS count

```r
# Count coding sequences
ecoli_count <- length (ecoli_seqs)
streptacidiphilus_count <- length (streptacidiphilus_seqs)

# Creating a summary table
coding_counts <- data.frame(
  Organism = c("Escherichia coli", "Streptacidiphilus jiangxiensis"),
  Coding_Sequences = c(ecoli_count, streptacidiphilus_count)
)

coding_counts
```

```
##                         Organism Coding_Sequences
## 1               Escherichia coli             4931
## 2 Streptacidiphilus jiangxiensis             8650
```

Answer: *Escherichia coli* contains 4,931 coding sequences while *Streptacidiphilus jiangxiensis* has a substantially higher count (8,650) of coding sequences, points to a significant disparity in genetic diversity between the two bacterial species. This greater number of coding sequences in *Streptacidiphilus jiangxiensis* is indicative of a more complex genetic framework, which may translate into enhanced functional capabilities and a broader range of physiological adaptations (Wright, 1990, Malik et al., 2020).

The expanded repertoire of genes in *Streptacidiphilus jiangxiensis* likely contributes to its metabolic versatility, enabling it to thrive in diverse environments. This versatility allows *Streptacidiphilus jiangxiensis* to exploit various substrates, potentially including organic compounds found in soil or plant matter, that *Escherichia coli* might not utilize as efficiently. For example, *Streptacidiphilus jiangxiensis* may possess unique enzymes or metabolic pathways that allow it to break down complex carbohydrates, synthesize essential nutrients, or produce secondary metabolites, such as antimicrobial compounds, which can offer competitive advantages in its ecological niche (Wright, 1990).

Moreover, the increased number of coding sequences may also reflect evolutionary adaptations to environmental pressures. In soil ecosystems, where nutrient availability can fluctuate, the ability to produce a wide range of enzymes and metabolites could enhance survival and reproduction (Harman and Uphoff, 2019). *Streptacidiphilus jiangxiensis* may engage in symbiotic relationships with plants or other soil microorganisms, leveraging its genetic diversity to facilitate nutrient exchange or improve soil health (Cong et al., 2021). In contrast, *Escherichia coli*, while highly adaptable and successful in its own right, is often more specialized for life in nutrient-rich environments, such as the gastrointestinal tract of mammals. Thus, the greater coding sequence count in *Streptacidiphilus jiangxiensis* underscores its potential for metabolic innovation and ecological resilience, reflecting a sophisticated evolutionary response to its environment (Wright, 1990; Malik et al., 2020).

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.

Total Coding DNA Length

```
# Calculate total coding DNA length
ecoli_length <- as.numeric(summary(ecoli_seqs)[,1])
streptacidiphilus_length <- as.numeric(summary(streptacidiphilus_seqs)[,1])

# Creating a summary table
total_lengths <- data.frame(
  Organism = c("Escherichia coli", "Streptacidiphilus jiangxiensis"),
  Total_Length = c(sum(ecoli_length), sum(streptacidiphilus_length))
)

total_lengths
```

```
##                         Organism Total_Length
## 1               Escherichia coli      4593474
## 2 Streptacidiphilus jiangxiensis      8422779
```

Answer: The genomic analysis reveals that *Escherichia coli* comprises approximately 4,593,474 base pairs of coding DNA, whereas *Streptacidiphilus jiangxiensis* possesses a significantly larger genomic footprint of about 8,422,779 base pairs. This substantial difference in coding DNA indicates a more complex genome in *Streptacidiphilus*, which is often associated with greater metabolic diversity. A larger genomic content can provide a broader array of genes, facilitating the synthesis of diverse proteins that are crucial for various metabolic pathways and biochemical processes (Bentley, 2009).

The expanded coding capacity of *Streptacidiphilus* likely equips it with the ability to thrive in complex and potentially harsh environmental conditions. For instance, the organism might possess genes that allow it to metabolize a wider range of substrates, adapt to changes in nutrient availability, or withstand environmental stresses, such as acidity or high salinity (Rasko et al., 2008). Such metabolic versatility could enable *Streptacidiphilus* to exploit ecological niches that are less accessible to simpler organisms like *E. coli*, which tends to thrive in nutrient-rich environments typical of the mammalian gut (Lozica et al., 2022).

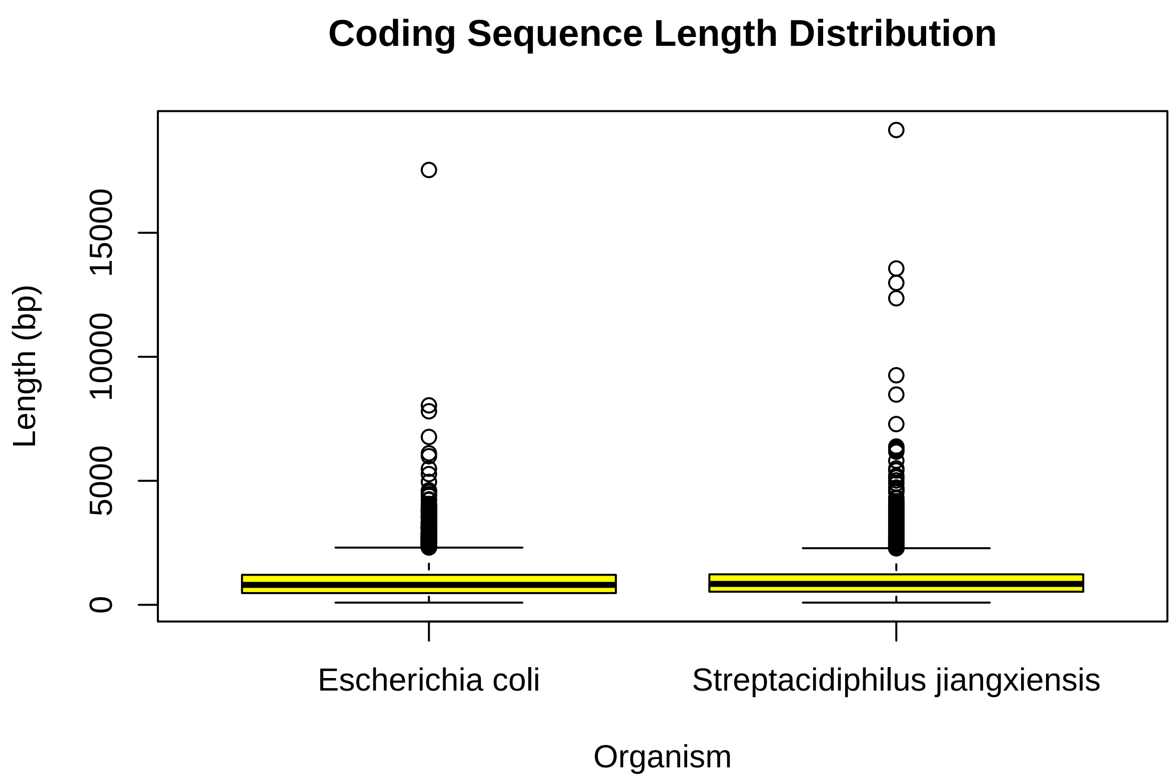
Furthermore, the increased genetic content in *Streptacidiphilus* may include genes responsible for specialized functions, such as biosynthesis of secondary metabolites, antibiotic resistance, or symbiotic interactions with other organisms (Jarocki et al., 2019). These attributes are particularly valuable for survival in competitive ecosystems where resource availability can be unpredictable. In essence, the larger coding DNA in *Streptacidiphilus jiangxiensis* reflects an evolutionary strategy that enhances its adaptability and functional repertoire, illustrating how genomic complexity can influence ecological success and resilience (Huang et al., 2004).

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.

Coding Sequence Length Distribution

```
# Boxplot of coding sequence lengths

boxplot(list(`Escherichia coli` = ecoli_length,
             `Streptacidiphilus jiangxiensis` = streptacidiphilus_length),
        col = "yellow",
```

```r
        xlab = "Organism",
        ylab = "Length (bp)",
        main = "Coding Sequence Length Distribution")
```



Mean and Median Coding Sequence Length

```r
mean_median <- data.frame(
  Organism = c("Escherichia coli", "Streptacidiphilus jiangxiensis"),
  Mean_Length = c(mean(ecoli_length), mean(streptacidiphilus_length)),
  Median_Length = c(median(ecoli_length), median(streptacidiphilus_length))
)

mean_median
```

```
##                         Organism Mean_Length Median_Length
## 1               Escherichia coli    931.5502           804
## 2 Streptacidiphilus jiangxiensis    973.7317           843
```

Answer: The comparison of coding sequence lengths between *Escherichia coli* and *Streptacidiphilus jiangxiensis* reveals significant insights into their genomic architectures and evolutionary adaptations. *Escherichia coli*, with a mean coding sequence length of 931.55 bp and a median of 804 bp, is known for its streamlined genome, which is optimized for rapid growth and efficient protein production in the nutrient-rich environments of mammalian intestines (Koonin, 2009). This bacterium's genetic simplicity has made it a model organism for studying fundamental biological processes, allowing researchers to explore genetic functions and interactions in a relatively uncomplicated context (Lizana and Schwartz, 2024).

In contrast, *Streptacidiphilus jiangxiensis* presents a longer mean coding sequence of 973.73 bp and a median of 843 bp, suggesting a more intricate gene structure. This complexity may be indicative of specialized adaptations to its unique ecological niche, which includes survival in acidic environments (Malik et al., 2020). Longer coding sequences can allow for the encoding of more extensive and functionally diverse proteins, potentially enabling the bacterium to exploit a broader range of substrates or cope with environmental stresses (Maharjan and Ferenci, 2014). The differences in coding sequence length may reflect distinct evolutionary pressures faced by these organisms, with *Escherichia coli* evolving for rapid proliferation and *Streptacidiphilus jiangxiensis* potentially developing complex metabolic pathways or interactions to thrive in less hospitable conditions (Wan et al., 2022). This variation underscores the diverse functional requirements and ecological strategies that shape the genomes of bacteria, highlighting how environmental factors influence genetic structure and complexity (Chan et al., 2018).