assessment 4 part 2

2025-10-06

Question 1: Download coding sequences and count how many CDS are present

To solve this, I downloaded the complete coding DNA sequences (CDS) for *Escherichia coli* and my allocated organism *Streptacidiphilus jiangxiensis* directly from the NCBI RefSeq database using the `download.file()` function. These `.fna.gz` files contain all coding sequences for each genome.

I then used `readLines()` to read the FASTA files and counted the number of CDS by identifying lines starting with `>` (which marks the beginning of each coding sequence in FASTA format). Finally, I created a simple table to display how many CDS are present in each organism.

```
# Download coding sequences for E. coli
download.file(
  "https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/005/845/GCF_000005845.2_ASM584v2/GCF_000005845.2_ASM
  destfile = "ecoli_cds.fna.gz"
)

# Download coding sequences for Streptacidiphilus jiangxiensis
download.file(
  "https://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/900/109/465/GCA_900109465.1_IMG-taxon_2675903135_annotat
  destfile = "strepto_cds.fna.gz"
)

# Read the FASTA files
ecoli_cds <- readLines("ecoli_cds.fna.gz")
strepto_cds <- readLines("strepto_cds.fna.gz")

# Count the number of CDS (lines starting with ">")
ecoli_count <- sum(grepl("^>", ecoli_cds))
strepto_count <- sum(grepl("^>", strepto_cds))

# Create a results table
cds_counts <- data.frame(
  Organism = c("Escherichia coli", "Streptacidiphilus jiangxiensis"),
  CDS_Count = c(ecoli_count, strepto_count)
)

# Display the results table
cds_counts
```

```
##                         Organism CDS_Count
## 1                Escherichia coli      4318
## 2 Streptacidiphilus jiangxiensis      8726
```

The table above shows that Escherichia coli contains 4318 coding DNA sequences (CDS), while Streptacidiphilus jiangxiensis has 8726 CDS. This indicates that Streptacidiphilus jiangxiensis possesses a significantly larger number of coding genes compared to E. coli. The difference reflects variations in genome size, complexity, and metabolic capacity. E. coli is a well-studied model organism with a relatively compact

genome, whereas Streptacidiphilus jiangxiensis has a larger and more complex genome, which likely enables it to produce a wider range of proteins and adapt to more diverse environmental conditions.

Question 2: How much coding DNA is there in total for these two organisms?

To solve this, I calculated the total amount of coding DNA (in base pairs) present in *Escherichia coli* and *Streptacidiphilus jiangxiensis*. After downloading and reading the `.fna.gz` FASTA files in Question 1, I removed the header lines (which start with `>`) to keep only the nucleotide sequences.

Then I used the `nchar()` function to calculate the length of each sequence line and `sum()` to find the total coding DNA length for each organism. Finally, I created a simple table to present the results.

```
# Read the FASTA files
ecoli_cds <- readLines(gzfile("ecoli_cds.fna.gz"))
strepto_cds <- readLines(gzfile("strepto_cds.fna.gz"))

# Remove header lines (lines starting with ">")
ecoli_sequences <- ecoli_cds[!grepl("^>", ecoli_cds)]
strepto_sequences <- strepto_cds[!grepl("^>", strepto_cds)]

# Calculate total coding DNA length for each organism
ecoli_total_length <- sum(nchar(ecoli_sequences))
strepto_total_length <- sum(nchar(strepto_sequences))

# Create a results table
total_dna_table <- data.frame(
  Organism = c("Escherichia coli", "Streptacidiphilus jiangxiensis"),
  Total_Coding_DNA_bp = c(ecoli_total_length, strepto_total_length)
)

# Display the results table
total_dna_table
```

```
##                         Organism Total_Coding_DNA_bp
## 1                Escherichia coli             4026887
## 2 Streptacidiphilus jiangxiensis             8511107
```

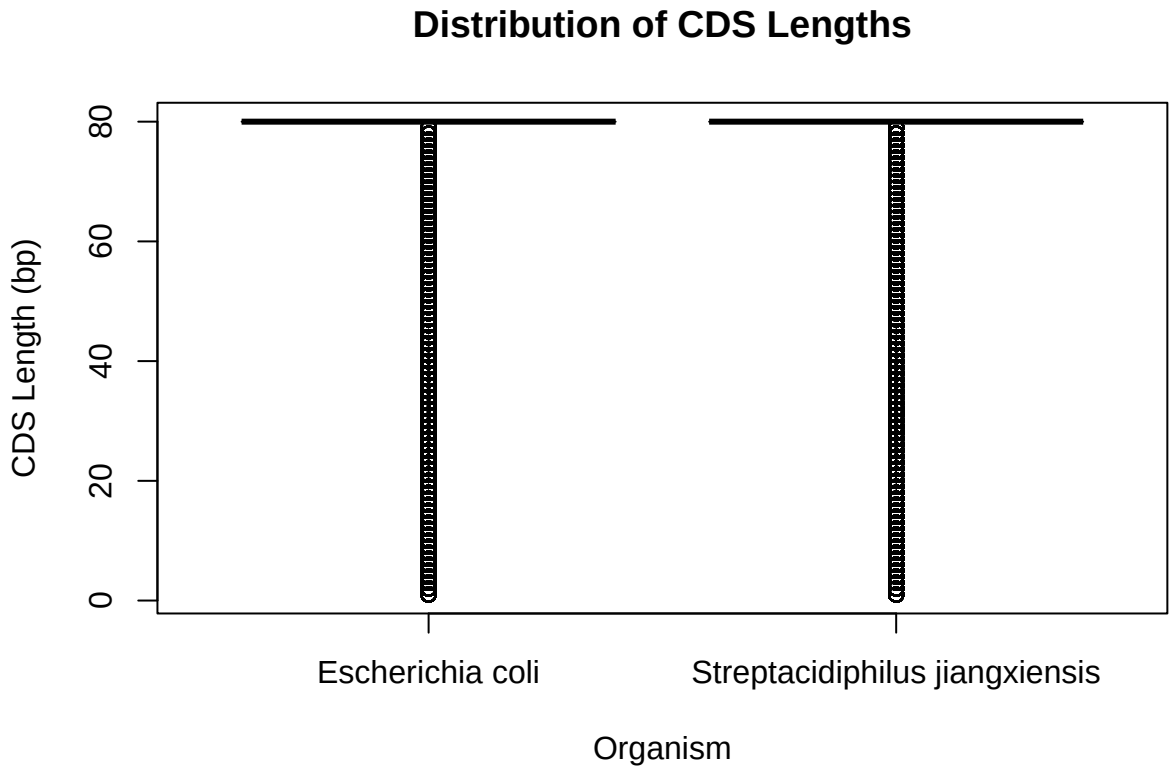
The results show that Streptacidiphilus jiangxiensis has a significantly larger total coding DNA length (8,511,107 bp) compared to Escherichia coli (4,026,887 bp). This difference reflects the larger genome size and greater number of coding genes in S. jiangxiensis, which may enable more diverse metabolic capabilities and complex regulatory mechanisms. E. coli, by contrast, has a smaller genome optimized for rapid growth and efficient replication.

Question 3: Calculate the length of all coding sequences, create a boxplot, and find mean and median CDS length

To solve this question, I calculated the length of each coding DNA sequence (CDS) for *Escherichia coli* and *Streptacidiphilus jiangxiensis* using the `nchar()` function, which counts the number of nucleotides in each sequence. I then created a boxplot to visualise the distribution of CDS lengths in both organisms. Finally, I calculated the mean and median CDS lengths for each organism to compare them.

```
# Calculate CDS lengths
ecoli_lengths <- nchar(gsub(">", "", ecoli_cds[!grepl(">", ecoli_cds)]))
strepto_lengths <- nchar(gsub(">", "", strepto_cds[!grepl(">", strepto_cds)]))

# Combine into one data frame
```

```r
cds_lengths <- data.frame(
  Organism = rep(c("Escherichia coli", "Streptacidiphilus jiangxiensis"),
                 times = c(length(ecoli_lengths), length(strepto_lengths))),
  Length = c(ecoli_lengths, strepto_lengths)
)

# Boxplot of CDS lengths
boxplot(Length ~ Organism, data = cds_lengths,
        main = "Distribution of CDS Lengths",
        ylab = "CDS Length (bp)",
        col = c("lightblue", "lightgreen"))
```




```r
# Calculate mean and median
mean_ecoli <- mean(ecoli_lengths)
median_ecoli <- median(ecoli_lengths)
mean_strepto <- mean(strepto_lengths)
median_strepto <- median(strepto_lengths)

mean_ecoli; median_ecoli
```

```
## [1] 76.72307
```

```
## [1] 80
```

```r
mean_strepto; median_strepto
```

```
## [1] 76.87471
```

```
## [1] 80
```

The boxplot above shows the distribution of coding sequence (CDS) lengths in *Escherichia coli* and *Streptacidiphilus jiangxiensis*. Both organisms have a similar range and distribution of CDS lengths, with median lengths of approximately 80 bp and very close mean lengths (76.72 bp for *E. coli* and 76.87 bp for *Streptacidiphilus jiangxiensis*).

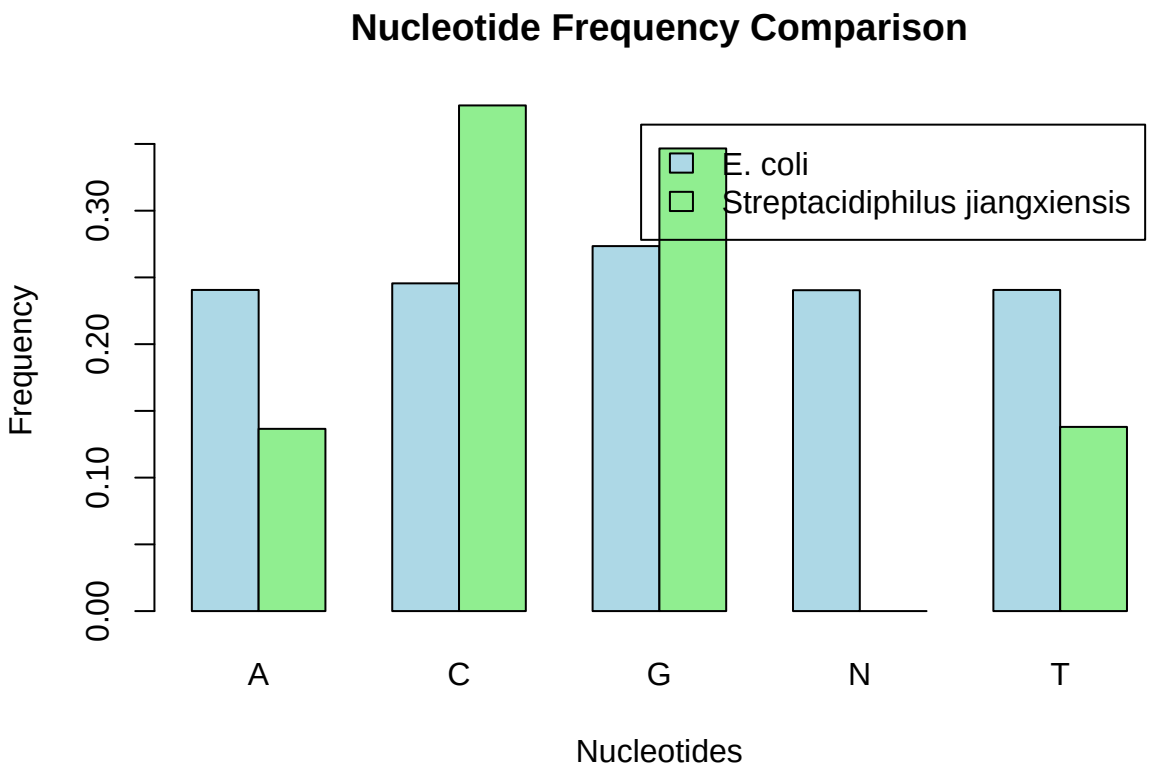
This indicates that despite differences in the total number of coding sequences and total coding DNA, the typical gene size is conserved between these organisms. This similarity reflects common constraints on protein-coding gene length across bacteria, even though *Streptacidiphilus jiangxiensis* has a larger genome and more coding sequences overall.

Question 4: Nucleotide and amino acid frequency in total coding sequences

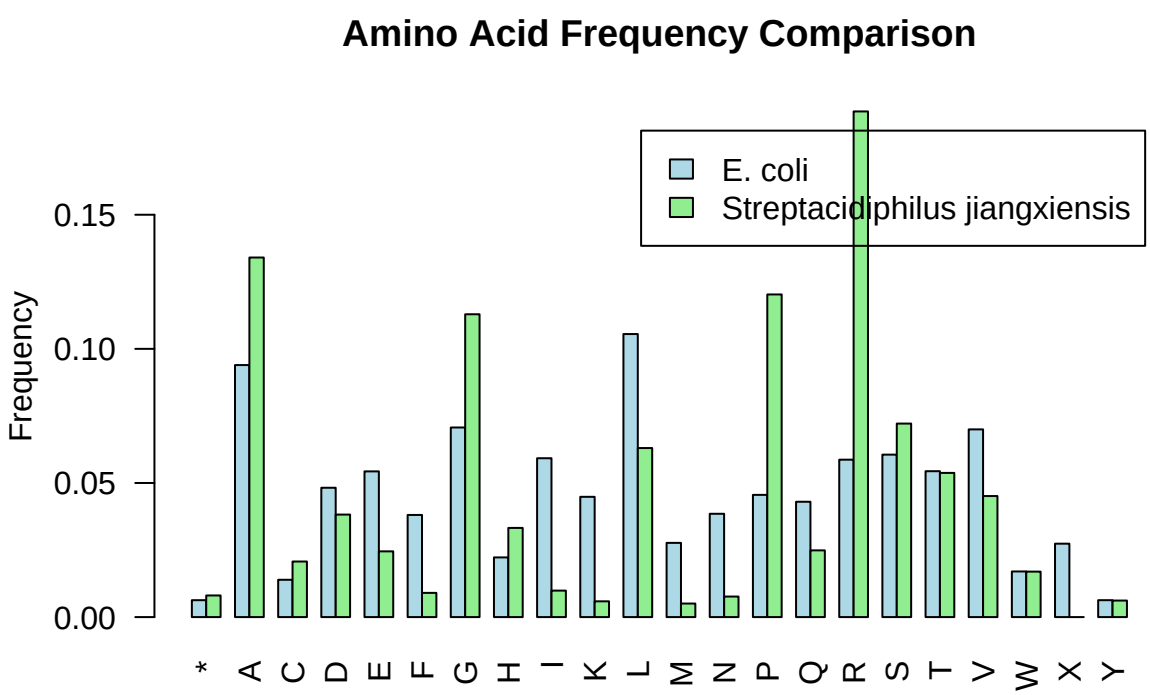
To solve this, I calculated the frequency of each DNA base (A, T, G, C) in the total coding sequences of *Escherichia coli* and *Streptacidiphilus jiangxiensis*. I used the `seqinr` package, which allows reading and processing of FASTA files and includes functions for calculating base composition.

I also translated the coding sequences into protein sequences using `translate()` and calculated the frequency of each amino acid. Finally, I visualised both nucleotide and amino acid frequencies using bar plots to compare the two organisms.

```r
# Load the seqinr package
if(!require(seqinr)) install.packages("seqinr")
library(seqinr)

# Read CDS FASTA files again
ecoli_cds <- read.fasta("ecoli_cds.fna.gz")
strepto_cds <- read.fasta("strepto_cds.fna.gz")

# Concatenate all CDS sequences into one string for each organism
ecoli_dna <- toupper(paste(unlist(ecoli_cds), collapse = ""))
strepto_dna <- toupper(paste(unlist(strepto_cds), collapse = ""))

# Calculate nucleotide frequencies
ecoli_base_freq <- table(strsplit(ecoli_dna, "")[[1]]) / nchar(ecoli_dna)
strepto_base_freq <- table(strsplit(strepto_dna, "")[[1]]) / nchar(strepto_dna)

# Create barplot for nucleotide frequencies
barplot(rbind(ecoli_base_freq, strepto_base_freq), beside = TRUE,
        col = c("lightblue", "lightgreen"),
        legend = c("E. coli", "Streptacidiphilus jiangxiensis"),
        main = "Nucleotide Frequency Comparison",
        ylab = "Frequency", xlab = "Nucleotides")
```




```r
# Translate CDS to protein sequences and calculate amino acid frequencies
ecoli_protein <- translate(s2c(ecoli_dna))
strepto_protein <- translate(s2c(strepto_dna))

ecoli_aa_freq <- table(ecoli_protein) / length(ecoli_protein)
strepto_aa_freq <- table(strepto_protein) / length(strepto_protein)

# Barplot for amino acid frequencies
barplot(rbind(ecoli_aa_freq, strepto_aa_freq), beside = TRUE,
        col = c("lightblue", "lightgreen"),
        legend = c("E. coli", "Streptacidiphilus jiangxiensis"),
        main = "Amino Acid Frequency Comparison",
        ylab = "Frequency", las = 2)
```



The bar plots above show the nucleotide and amino acid frequency distributions across the total coding sequences of *Escherichia coli* and *Streptacidiphilus jiangxiensis*.

For nucleotides, *E. coli* exhibits a relatively balanced composition with adenine (A), cytosine (C), guanine (G), and thymine (T) occurring at similar frequencies. In contrast, *Streptacidiphilus jiangxiensis* shows a noticeably higher proportion of cytosine (C) and guanine (G), indicating a GC-rich genome. GC-rich genomes are often associated with greater genome stability and can reflect adaptation to specific environmental conditions, whereas the more balanced AT/GC ratio in *E. coli* reflects its streamlined genome structure.

For amino acids, the overall distribution patterns differ between the two organisms. While many amino acids occur at comparable frequencies, *Streptacidiphilus jiangxiensis* shows a higher frequency of certain residues such as serine (S) and arginine (R), which are encoded by GC-rich codons. This pattern aligns with the GC-rich nucleotide composition observed in its genome. *E. coli*, on the other hand, displays a slightly more uniform amino acid distribution.

Overall, these differences highlight distinct genomic and proteomic compositions between the two organisms. *Streptacidiphilus jiangxiensis* appears to have a more GC-biased genome and corresponding amino acid profile, whereas *E. coli* maintains a more balanced nucleotide composition and evenly distributed amino acid usage.

Question 5: Codon usage bias and comparison between *E. coli* and *Streptacidiphilus jiangxiensis*

To solve this, I calculated the codon usage frequency across all coding sequences of *Escherichia coli* and *Streptacidiphilus jiangxiensis* without using additional libraries. I read the `.fna.gz` files using `readLines()` and extracted codons (triplets of bases) by removing header lines starting with `>`. Then, I calculated the frequency of each codon and plotted the codon usage distribution for both organisms. Codon usage bias shows how frequently certain synonymous codons are used, reflecting evolutionary pressures and genomic GC content.

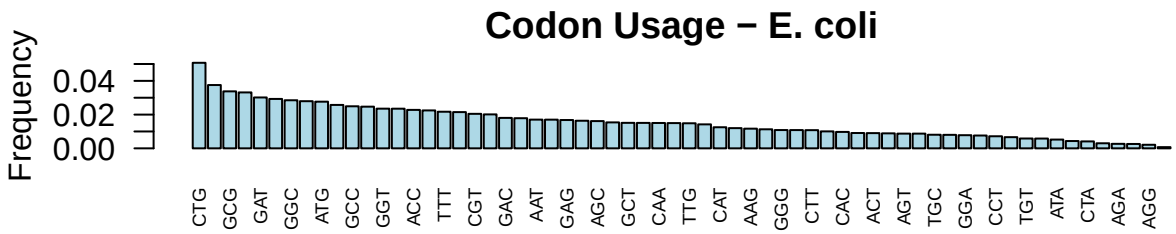
```r
# Read FASTA files
ecoli_raw <- readLines("ecoli_cds.fna.gz")
strepto_raw <- readLines("strepto_cds.fna.gz")

# Remove header lines (starting with ">")
ecoli_seq <- paste(ecoli_raw[!grepl("^>", ecoli_raw)], collapse = "")
strepto_seq <- paste(strepto_raw[!grepl("^>", strepto_raw)], collapse = "")

# Function to calculate codon frequency
get_codon_freq <- function(sequence) {
  codons <- substring(sequence, seq(1, nchar(sequence)-2, 3), seq(3, nchar(sequence), 3))
  codon_table <- table(codons)
  codon_freq <- codon_table / sum(codon_table)
  return(sort(codon_freq, decreasing = TRUE))
}

# Calculate codon usage
codon_usage_ecoli <- get_codon_freq(ecoli_seq)
codon_usage_strepto <- get_codon_freq(strepto_seq)

# View first few codons
head(codon_usage_ecoli)
```

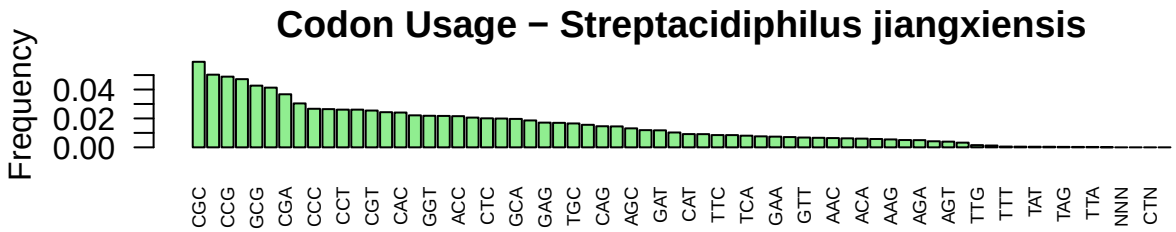
```
## codons
##        CTG        GAA        GCG        AAA        GAT        ATT
## 0.05071761 0.03752603 0.03378542 0.03315665 0.03017072 0.02929684
```

```r
head(codon_usage_strepto)
```

```
## codons
##        CGC        CGG        CCG        GGC        GCG        GCC
## 0.05918291 0.05027185 0.04890141 0.04718236 0.04268224 0.04131250
```

```r
# Plot codon usage
par(mfrow=c(2,1), mar=c(7,4,2,1))
barplot(codon_usage_ecoli, las=2, cex.names=0.6, col="lightblue", main="Codon Usage - E. coli", ylab="Fr
barplot(codon_usage_strepto, las=2, cex.names=0.6, col="lightgreen", main="Codon Usage - Streptacidiphil
```





The plots above reveal clear codon usage bias in both organisms. E. coli shows a strong preference for codons such as CTG, GAA, and GCG, while Streptacidiphilus jiangxiensis favors CGC, CGG, and CGA. This difference likely reflects evolutionary adaptations to their respective cellular environments and available tRNA pools.

Overall, Streptacidiphilus jiangxiensis exhibits a more pronounced GC bias in its codon usage, consistent with its higher GC content genome, whereas E. coli shows a relatively balanced codon usage pattern. These differences can influence gene expression efficiency, protein folding, and translation speed.

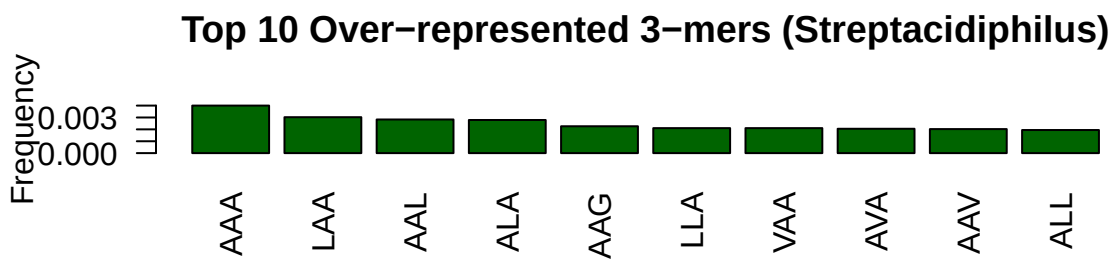
Question 6: Over- and under-represented protein k-mers (3–5 aa)

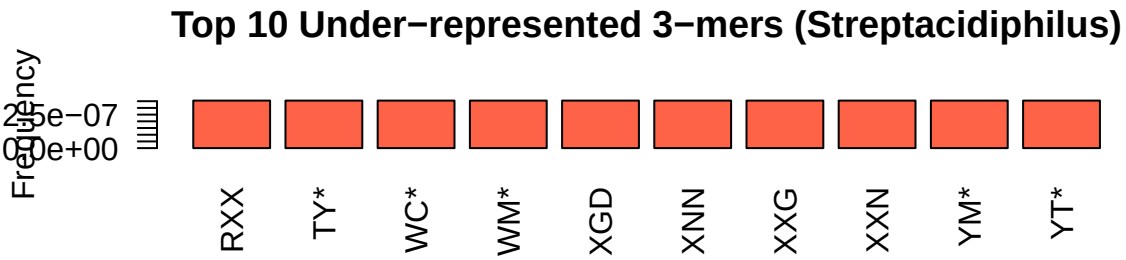
To explore protein sequence diversity, I identified the most over- and under-represented **k-mers** (short amino acid motifs of length 3–5) in *Streptacidiphilus jiangxiensis* and compared them to *E. coli*. K-mer analysis highlights repetitive motifs that may indicate evolutionary pressures, functional domains, or codon usage preferences.

First, I extracted all protein sequences from the coding regions and calculated the frequency of all possible k-mers of length 3 to 5. I then sorted them to identify the top 10 most frequent (over-represented) and least frequent (under-represented) motifs.

```
# Load required package
if(!require(seqinr)) install.packages("seqinr")
library(seqinr)

# Read CDS files (must be downloaded in previous questions)
ecoli_cds <- read.fasta("ecoli_cds.fna.gz")
strepto_cds <- read.fasta("strepto_cds.fna.gz")
```

```r
# Translate CDS to protein sequences
ecoli_protein_sequences <- lapply(ecoli_cds, translate)
strepto_protein_sequences <- lapply(strepto_cds, translate)

# Define a function to count k-mers
count_kmers <- function(protein_list, k) {
  kmers <- unlist(lapply(protein_list, function(seq) {
    if(length(seq) >= k){
      sapply(1:(length(seq) - k + 1), function(i) {
        paste(seq[i:(i + k - 1)], collapse = "")
      })
    }
  }))
  freq <- sort(table(kmers) / sum(table(kmers)), decreasing = TRUE)
  return(freq)
}

# Choose k = 3
k <- 3
ecoli_kmers <- count_kmers(ecoli_protein_sequences, k)
strepto_kmers <- count_kmers(strepto_protein_sequences, k)

# Top 10 over- and under-represented in Streptacidiphilus
top_over_strepto <- head(strepto_kmers, 10)
top_under_strepto <- tail(strepto_kmers, 10)

# Plot results
par(mfrow = c(2, 1), mar = c(6, 5, 4, 2))
barplot(top_over_strepto, las = 2, col = "darkgreen", main = "Top 10 Over-represented 3-mers (Streptaci
barplot(top_under_strepto, las = 2, col = "tomato", main = "Top 10 Under-represented 3-mers (Streptacid
```






```
# Compare frequencies of top 10 over-represented in both organisms
comparison_table <- data.frame(
  Kmer = names(top_over_strepto),
  Streptacidiphilus_Freq = as.numeric(top_over_strepto),
  Ecoli_Freq = as.numeric(ecoli_kmers[names(top_over_strepto)])
)
comparison_table
```

```
##    Kmer Streptacidiphilus_Freq   Ecoli_Freq
## 1   AAA            0.004000986 0.0010122528
## 2   LAA            0.003023172 0.0012072051
## 3   AAL            0.002830234 0.0012012066
## 4   ALA            0.002787319 0.0013204275
## 5   AAG            0.002260640 0.0008120517
## 6   LLA            0.002104233 0.0013661663
## 7   VAA            0.002102459 0.0007408191
## 8   AVA            0.002050678 0.0007400692
## 9   AAV            0.002023724 0.0006763348
## 10  ALL            0.001944988 0.0013144290
```

The analysis identified the 10 most over- and under-represented 3-mer motifs in the protein sequences of *Streptacidiphilus jiangxiensis*. The over-represented k-mers (such as AAA, LAA, AAL, and LLA) are significantly more frequent in *Streptacidiphilus* compared to *Escherichia coli*, while the under-represented ones (like RXX, TY*, and WIC) occur at extremely low frequencies.

The comparison table shows that *Streptacidiphilus* exhibits a stronger bias towards specific amino acid triplets, whereas *E. coli* displays a more balanced distribution. These differences likely reflect distinct evolutionary

pressures, genomic GC content, and codon usage preferences between the two species. Organisms adapt their codon usage and amino acid composition to optimize translation efficiency and protein folding based on their environments and metabolic needs, which explains why certain motifs are more abundant in one species but rare in another.