MEHNAZ_MEEM_A4_part_2

Mehnaz_meem

2025-10-06

##Question 1 ##Downloading the gemone

```
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 v1.27.1 (2025-05-02 21:00:05 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 v2.13.0 (2025-02-24 21:20:02 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
```

```
#Download and Process E. coli CDS
#E. coli K-12 MG1655 (GCA_000005845) CDS file
URL="http://ftp.ensemblgenomes.org/pub/bacteria/release-53/fasta/bacteria_0_collection/escherichia_coli_
download.file(URL,destfile="ecoli_cds.fa.gz")

if (!file.exists("ecoli_cds.fa")) {
  R.utils::gunzip("ecoli_cds.fa.gz")
}
```

```r
list.files()
```

```
##  [1] "ccoli_cds.fa"                "ccoli_cds.fa.gz"
##  [3] "ecoli_cds.fa"                "ecoli_cds.fa.gz"
##  [5] "gene_expression.tsv"         "growth_data.csv"
##  [7] "LICENSE"                     "MEHNAZ_MEEM_A4_part_1.pdf"
##  [9] "MEHNAZ_MEEM_A4_part_1.Rmd"   "MEHNAZ_MEEM_A4_part_2_files"
## [11] "MEHNAZ_MEEM_A4_part_2.pdf"   "MEHNAZ_MEEM_A4_part_2.Rmd"
## [13] "MEHNAZ_RSTUDIO.Rproj"        "README.html"
## [15] "README.md"                   "week 8 test file.Rmd"
## [17] "week 8 test.Rmd"             "WEEK 8.Rmd"
## [19] "week-8-test-file.html"       "week-8-test.html"
## [21] "WEEK-8.html"                 "Week-9.pdf"
## [23] "Week10.Rmd"
```

```r
#Download and Process C. coli CDS
#Campylobacter coli (GCA_003780985) CDS file
URL="https://ftp.ensemblgenomes.ebi.ac.uk/pub/bacteria/release-62/fasta/bacteria_46_collection/campyloba
download.file(URL,destfile="ccoli_cds.fa.gz")

if (!file.exists("ccoli_cds.fa")) {
  R.utils::gunzip("ccoli_cds.fa.gz")
}

list.files()
```

```
##  [1] "ccoli_cds.fa"                "ccoli_cds.fa.gz"
##  [3] "ecoli_cds.fa"                "ecoli_cds.fa.gz"
##  [5] "gene_expression.tsv"         "growth_data.csv"
##  [7] "LICENSE"                     "MEHNAZ_MEEM_A4_part_1.pdf"
##  [9] "MEHNAZ_MEEM_A4_part_1.Rmd"   "MEHNAZ_MEEM_A4_part_2_files"
## [11] "MEHNAZ_MEEM_A4_part_2.pdf"   "MEHNAZ_MEEM_A4_part_2.Rmd"
## [13] "MEHNAZ_RSTUDIO.Rproj"        "README.html"
## [15] "README.md"                   "week 8 test file.Rmd"
## [17] "week 8 test.Rmd"             "WEEK 8.Rmd"
## [19] "week-8-test-file.html"       "week-8-test.html"
## [21] "WEEK-8.html"                 "Week-9.pdf"
## [23] "Week10.Rmd"
```

##Calculates the total number of Coding DNA Sequences (CDS) for both organisms

```r
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
```

Load the CDS files

```
ecoli_cds_dna <- readDNAStringSet("ecoli_cds.fa")
ccoli_cds_dna <- readDNAStringSet("ccoli_cds.fa")
```

Calculate the number of coding sequences

```
num_ecoli <- length(ecoli_cds_dna)
num_ccoli <- length(ccoli_cds_dna)
```

Create the results table

```
cds_count_table <- data.frame(
  Organism = c("E. coli (K-12 MG1655)", "C. coli (GCA_003780985)"),
  `Number of Coding Sequences` = c(num_ecoli, num_ccoli)
)
```

Print the table

```r
print(cds_count_table)
```

```
##                   Organism Number.of.Coding.Sequences
## 1   E. coli (K-12 MG1655)                       4239
## 2 C. coli (GCA_003780985)                       1976
```

##The main difference lies in their gene set size, which reflects their lifestyles: E.coli (with >2.7× the genes) is a versatile generalist, while C.coli is a specialized pathogen with a streamlined, reduced genome optimized for a stable host environment.

##Question 2 ##Calculation of the total base pair length of all coding sequences

```r
library(Biostrings)
```

Load the CDS files

```r
ecoli_cds_dna <- readDNAStringSet("ecoli_cds.fa")
ccoli_cds_dna <- readDNAStringSet("ccoli_cds.fa")
```

Calculate the total length of coding DNA (sum of all sequence widths)

```r
total_length_ecoli <- sum(width(ecoli_cds_dna))
total_length_ccoli <- sum(width(ccoli_cds_dna))
```

Create the results table using base R data.frame()

```r
total_length_table <- data.frame(
  Organism = c("E. coli (K-12 MG1655)", "C. coli (GCA_003780985)"),
  `Total_Coding_DNA_Length_(base_pairs)` = c(total_length_ecoli, total_length_ccoli),
  row.names = NULL
)
```

Print the table

```r
print(total_length_table)
```

```
##                   Organism Total_Coding_DNA_Length_.base_pairs.
## 1   E. coli (K-12 MG1655)                              3978528
## 2 C. coli (GCA_003780985)                              1726818
```

##E. coli possesses approx 2.75 times more total coding DNA, supporting its versatile generalist role with a higher gene count. Conversely, C. coli's reduced coding length reflects its streamlined, specialized pathogenic existence.

##Question 3 ##Calculation of the mean and median lengths of all coding sequences ##Generate a boxplot for visual comparison.

```
library(Biostrings)
library(ggplot2)
```

##Load the CDS files

```
ecoli_cds_dna <- readDNAStringSet("ecoli_cds.fa")
ccoli_cds_dna <- readDNAStringSet("ccoli_cds.fa")
```

##Calculate lengths for all sequences

```
ecoli_lengths <- width(ecoli_cds_dna)
ccoli_lengths <- width(ccoli_cds_dna)
```

##Calculate Mean and Median

```
ecoli_stats <- c(Mean = mean(ecoli_lengths), Median = median(ecoli_lengths))
ccoli_stats <- c(Mean = mean(ccoli_lengths), Median = median(ccoli_lengths))
```

##Create the summary table

```
stats_df <- data.frame(
  Organism = c("E. coli (K-12 MG1655)", "C. coli (GCA_003780985)"),
  Mean_Length_bp = c(ecoli_stats["Mean"], ccoli_stats["Mean"]),
  Median_Length_bp = c(ecoli_stats["Median"], ccoli_stats["Median"]),
  row.names = NULL
)
print(stats_df)
```

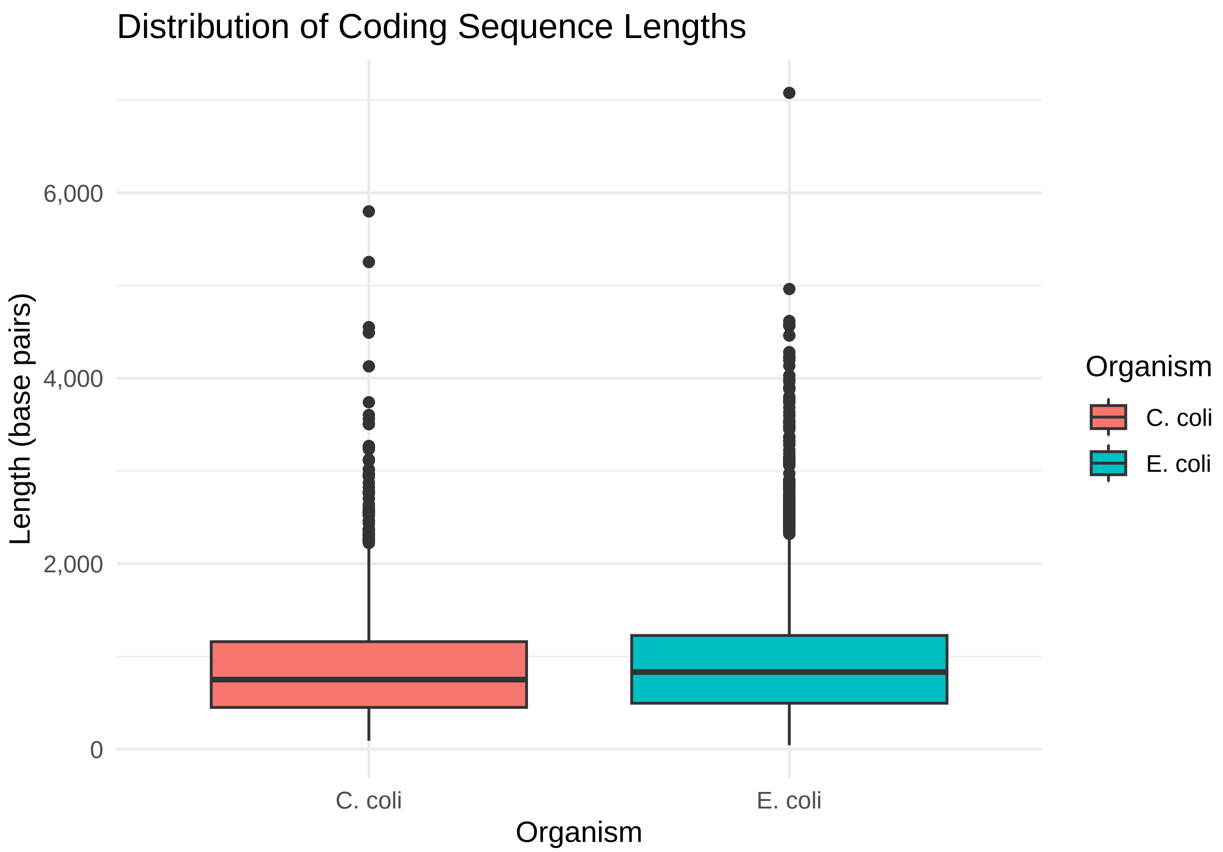
```
##                  Organism Mean_Length_bp Median_Length_bp
## 1   E. coli (K-12 MG1655)       938.5534              831
## 2 C. coli (GCA_003780985)       873.8957              750
```

##Create a combined data frame for the boxplot

```
length_df <- data.frame(
  Length = c(ecoli_lengths, ccoli_lengths),
  Organism = factor(c(rep("E. coli", length(ecoli_lengths)), rep("C. coli", length(ccoli_lengths))))
)
```

##Generate the Boxplot

```
p_boxplot <- ggplot(length_df, aes(x = Organism, y = Length, fill = Organism)) +
  geom_boxplot() +
  scale_y_continuous(labels = scales::comma) +
  labs(
    title = "Distribution of Coding Sequence Lengths",
    y = "Length (base pairs)",
    x = "Organism"
  ) +
  theme_minimal()
print(p_boxplot)
```



##The key difference lies not in average length (mean/median are similar), but in gene count and complexity. Both distributions are right-skewed, yet E. coli exhibits slightly more long gene outliers, reflecting its broader functional complexity and capacity for multi-domain proteins compared to the streamlined C. coli genome.

##Question 4 ##Calculations and plots the frequency of DNA bases and amino acids for the total coding and protein sequences of both organisms.

##Nucleotide Frequency

##Calculate total length

```
total_length_ecoli <- sum(width(ecoli_cds_dna))
total_length_ccoli <- sum(width(ccoli_cds_dna))
```

##Collapse all CDS into a single string for frequency calculation

```
ecoli_nuc_freq <- alphabetFrequency(DNAStringSet(paste(ecoli_cds_dna, collapse = "")))[1, 1:4] / total_l
ccoli_nuc_freq <- alphabetFrequency(DNAStringSet(paste(ccoli_cds_dna, collapse = "")))[1, 1:4] / total_l
```

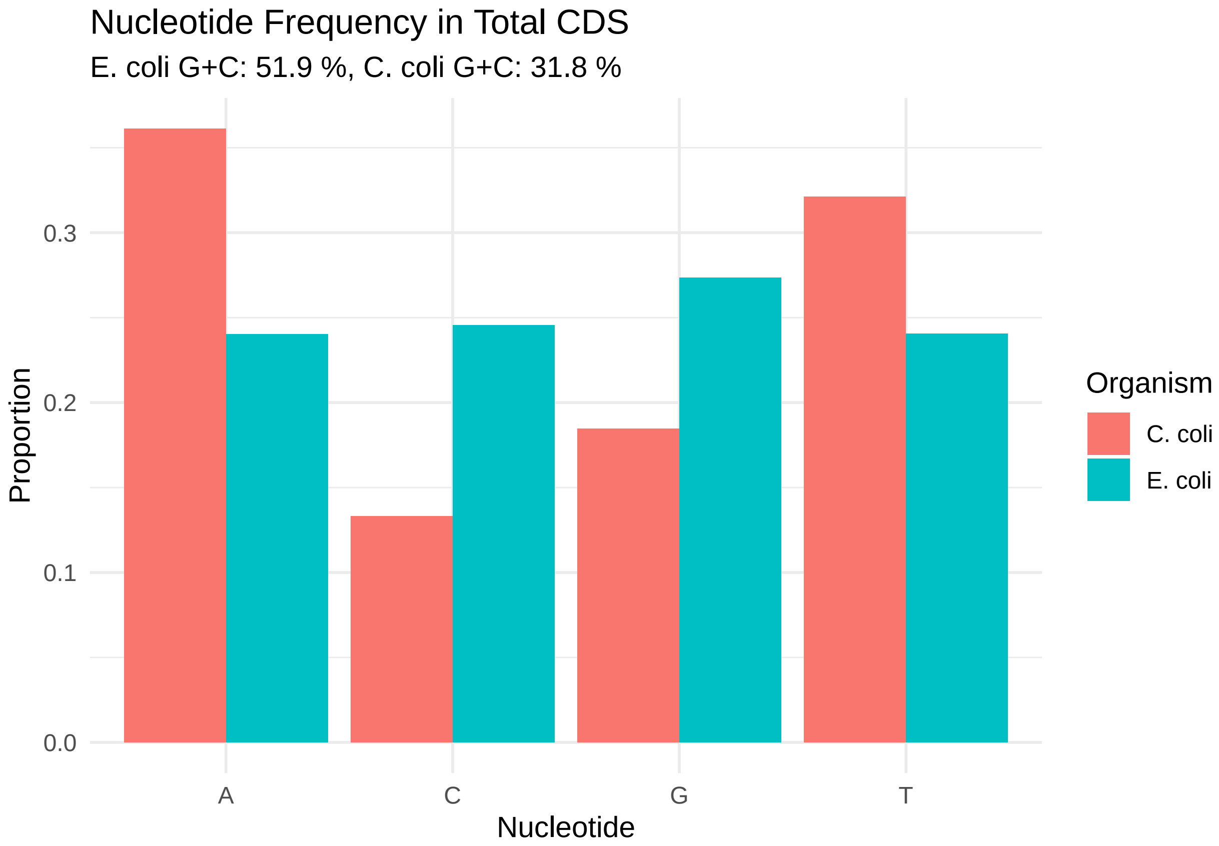
##Create data frame for plotting

```
nuc_df <- data.frame(
  Nucleotide = names(ecoli_nuc_freq),
  Ecoli = ecoli_nuc_freq,
  Ccoli = ccoli_nuc_freq
)

nuc_df_long <- data.frame(
  Nucleotide = rep(nuc_df$Nucleotide, 2),
  Organism = c(rep("E. coli", 4), rep("C. coli", 4)),
  Frequency = c(nuc_df$Ecoli, nuc_df$Ccoli)
)
```

##Nucleotide Bar Plot

```
p_nuc_bar <- ggplot(nuc_df_long, aes(x = Nucleotide, y = Frequency, fill = Organism)) +
  geom_bar(stat = "identity", position = position_dodge()) +
  labs(
    title = "Nucleotide Frequency in Total CDS",
    y = "Proportion",
    subtitle = paste("E. coli G+C:", round(sum(ecoli_nuc_freq[c("G", "C")])*100, 1),
                     "%, C. coli G+C:", round(sum(ccoli_nuc_freq[c("G", "C")])*100, 1), "%")
    ) +
  theme_minimal()
print(p_nuc_bar)
```



##Amino Acid Frequency

##Translate DNA to Protein Sequences

```
ecoli_protein <- Biostrings::translate(ecoli_cds_dna)
ccoli_protein <- Biostrings::translate(ccoli_cds_dna)
```

##Collapse protein sequences and calculate frequency (using AA_ALPHABET to exclude symbols like '*')

```
ecoli_aa_freq <- alphabetFrequency(AAStringSet(paste(ecoli_protein, collapse = "")))[1, AA_ALPHABET]
ccoli_aa_freq <- alphabetFrequency(AAStringSet(paste(ccoli_protein, collapse = "")))[1, AA_ALPHABET]
```

Normalize

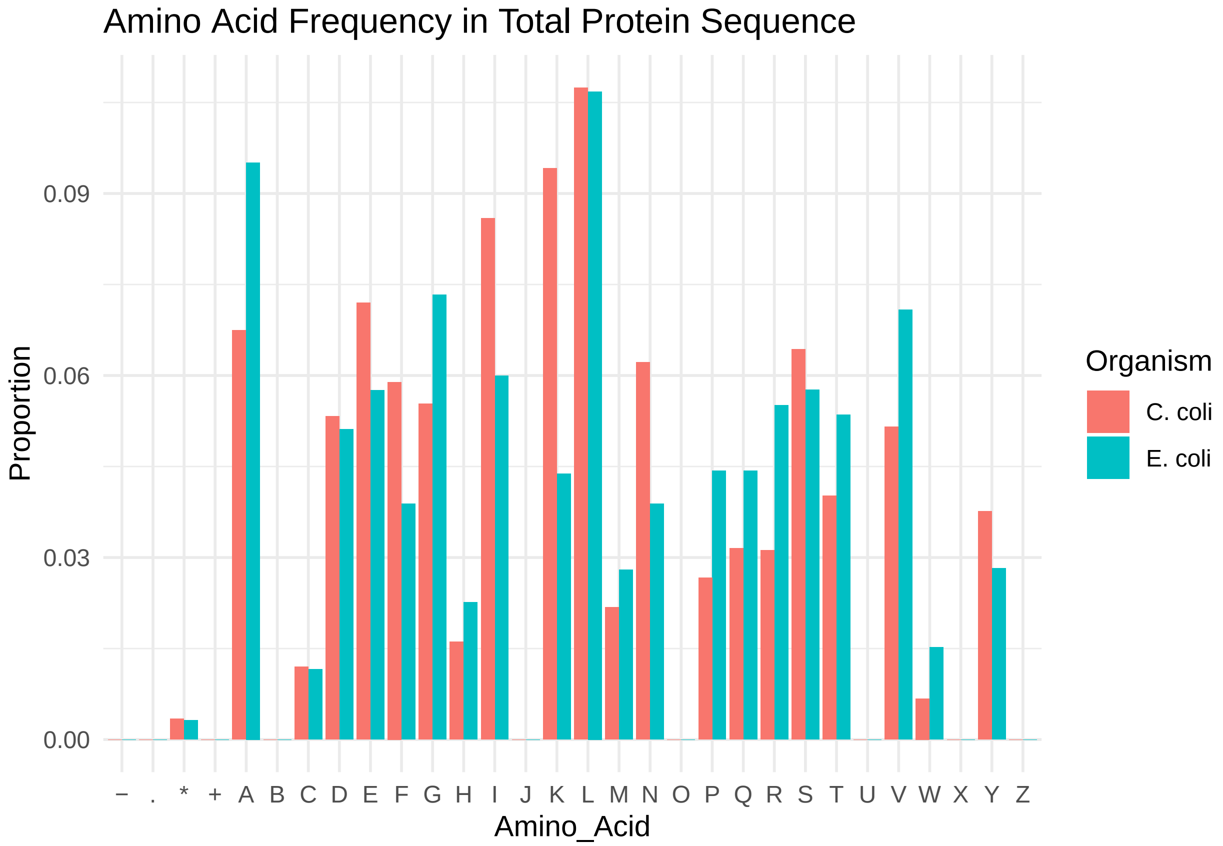
```
ecoli_aa_freq_norm <- ecoli_aa_freq / sum(ecoli_aa_freq)
ccoli_aa_freq_norm <- ccoli_aa_freq / sum(ccoli_aa_freq)
```

Create data frame for plotting

```
aa_df_long <- data.frame(
  Amino_Acid = rep(names(ecoli_aa_freq_norm), 2),
  Organism = c(rep("E. coli", length(AA_ALPHABET)), rep("C. coli", length(AA_ALPHABET))),
  Frequency = c(ecoli_aa_freq_norm, ccoli_aa_freq_norm)
)
```

Amino Acid Bar Plot

```
p_aa_bar <- ggplot(aa_df_long, aes(x = Amino_Acid, y = Frequency, fill = Organism)) +
  geom_bar(stat = "identity", position = position_dodge()) +
  labs(title = "Amino Acid Frequency in Total Protein Sequence", y = "Proportion") +
  theme_minimal()
print(p_aa_bar)
```



##Nucleotide:C. coli is highly A/T-rich G+C approx 30%, imposing a major compositional bias on its coding sequences, whereas E. coli maintains a balanced G+C content near 50%.

##Amino Acid: This bias translates directly to protein: C. coli favors A/T -rich amino acids (e.g., Lysine, Isoleucine), demonstrating a genomic constraint where the nucleotide pool limits the available protein components; E. coli favors G/C-rich amino acids (e.g., Alanine, Glycine).

##Question 5 ##Read coding DNA Sequences

```
library(seqinr)
```

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

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

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

```
cds_ecoli <- read.fasta("ecoli_cds.fa")
cds_ccoli <- read.fasta("ccoli_cds.fa")
```

##Calculate RSCU

```
codon_ecoli <- uco(unlist(cds_ecoli), index = "rscu", as.data.frame = TRUE)
codon_ccoli <- uco(unlist(cds_ccoli), index = "rscu", as.data.frame = TRUE)
```

##Sort codons by RSCU values

##E.coli

```
sorted_ecoli <- codon_ecoli[order(codon_ecoli$RSCU), ]
top10_codons_ecoli_under <- head(sorted_ecoli, 10)
top10_codons_ecoli_over <- head(sorted_ecoli[order(-sorted_ecoli$RSCU), ], 10)
```

##C.coli

```
sorted_ccoli <- codon_ccoli[order(codon_ccoli$RSCU), ]
top10_codons_ccoli_under <- head(sorted_ccoli, 10)
top10_codons_ccoli_over <- head(sorted_ccoli[order(-sorted_ccoli$RSCU), ], 10)
```

##Print top codons

```
print(top10_codons_ecoli_under)
```

```
##      AA codon   eff        freq      RSCU
## agg Arg   agg  1420 0.001070748 0.1165351
## ata Ile   ata  5486 0.004136706 0.2069902
## tag Stp   tag   294 0.000221690 0.2080679
## aga Arg   aga  2573 0.001940165 0.2111584
## cta Leu   cta  5149 0.003882592 0.2179763
## cga Arg   cga  4619 0.003482946 0.3790674
## gga Gly   gga 10350 0.007804394 0.4257245
## aag Lys   aag 13521 0.010195479 0.4653348
## ccc Pro   ccc  7238 0.005457797 0.4932198
## aca Thr   aca  9116 0.006873899 0.5133967
```

```
print(top10_codons_ecoli_over)
```

```
##      AA codon   eff        freq     RSCU
## ctg Leu   ctg 70714 0.053321731 2.993586
## cgc Arg   cgc 29441 0.022199919 2.416134
## cgt Arg   cgt 27979 0.021097501 2.296152
## ccg Pro   ccg 31074 0.023431279 2.117479
## taa Stp   taa  2726 0.002055534 1.929229
## acc Thr   acc 31139 0.023480292 1.753692
## agc Ser   agc 21291 0.016054430 1.671805
## ggc Gly   ggc 39536 0.029812031 1.626226
## aaa Lys   aaa 44592 0.033624496 1.534665
## att Ile   att 40501 0.030539687 1.528128
```

```
print(top10_codons_ccoli_under)
```

```
##      AA codon  eff        freq       RSCU
## cgg Arg   cgg  117 0.000203264 0.03908468
## ctg Leu   ctg  863 0.001499289 0.08350670
## ttc Phe   ttc 2815 0.004890498 0.16598856
## gac Asp   gac 2832 0.004920032 0.18464548
## ctc Leu   ctc 1989 0.003455489 0.19246214
## tcg Ser   tcg 1204 0.002091709 0.19500081
## cag Gln   cag 1800 0.003127139 0.19840176
## ccg Pro   ccg  810 0.001407213 0.21080026
## tcc Ser   tcc 1348 0.002341880 0.21832317
## cga Arg   cga  719 0.001249118 0.24018707
```

```
print(top10_codons_ccoli_over)
```

```
##      AA codon   eff        freq     RSCU
## aga Arg   aga  9589 0.016658965 3.203274
## tta Leu   tta 27068 0.047025222 2.619188
## cct Pro   cct  8875 0.015418533 2.309694
## taa Stp   taa  1263 0.002194209 1.917510
## agt Ser   agt 11826 0.020545304 1.915348
## gct Ala   gct 18392 0.031952412 1.895350
## ttt Phe   ttt 31103 0.054035225 1.834011
## gat Asp   gat 27843 0.048371629 1.815355
## caa Gln   caa 16345 0.028396160 1.801598
## ctt Leu   ctt 18003 0.031276602 1.742029
```

##Create Barplots

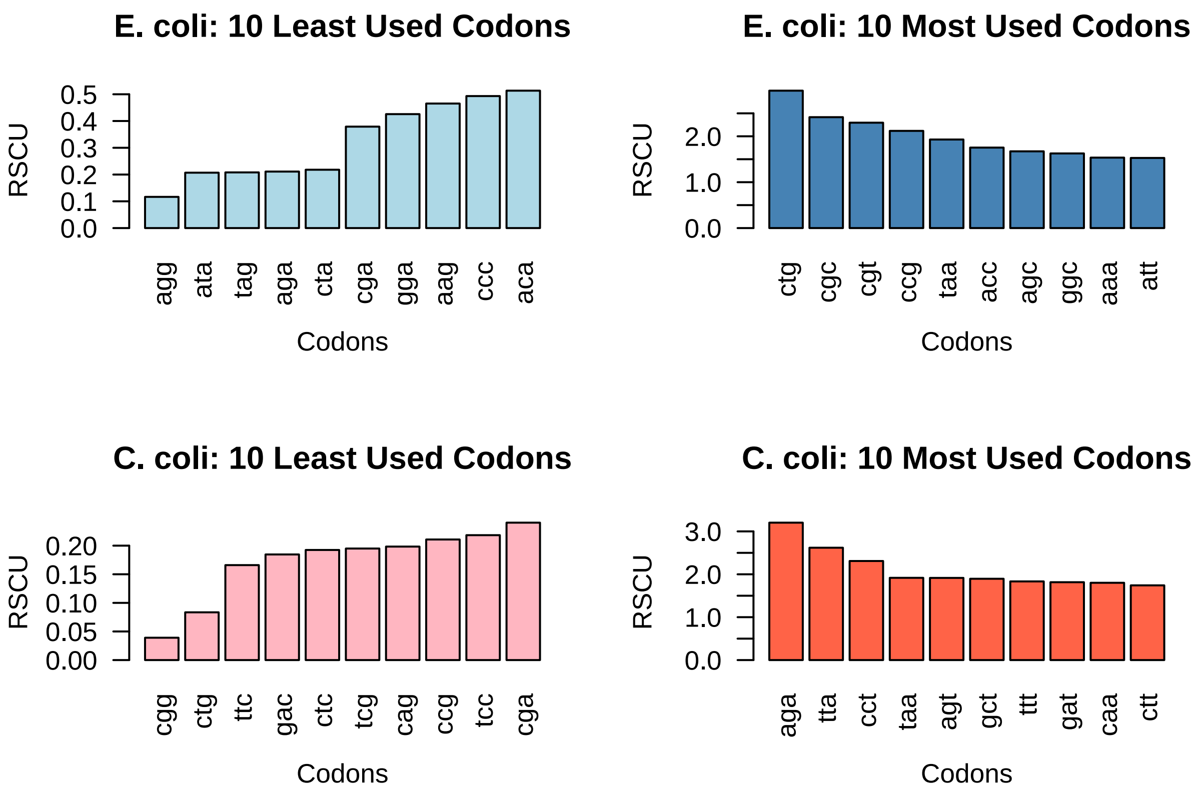
```
par(mfrow = c(2, 2))  # 2x2 layout for four plots

barplot(top10_codons_ecoli_under$RSCU, names.arg = top10_codons_ecoli_under$codon,
        xlab = "Codons", ylab = "RSCU", col = "lightblue",
        main = "E. coli: 10 Least Used Codons", las = 2)

barplot(top10_codons_ecoli_over$RSCU, names.arg = top10_codons_ecoli_over$codon,
        xlab = "Codons", ylab = "RSCU", col = "steelblue",
        main = "E. coli: 10 Most Used Codons", las = 2)

barplot(top10_codons_ccoli_under$RSCU, names.arg = top10_codons_ccoli_under$codon,
        xlab = "Codons", ylab = "RSCU", col = "lightpink",
        main = "C. coli: 10 Least Used Codons", las = 2)

barplot(top10_codons_ccoli_over$RSCU, names.arg = top10_codons_ccoli_over$codon,
        xlab = "Codons", ylab = "RSCU", col = "tomato",
        main = "C. coli: 10 Most Used Codons", las = 2)
```



##Figure:Codon usage bias (RSCU) in Escherichia coli and Campylobacter coli.
The plots show the top 10 over-represented and under-represented codons in each organism.

E. coli shows a stronger codon bias, frequently using codons ending with G or C, consistent with its higher GC content and efficient translational machinery.
Campylobacter coli exhibits weaker bias, with higher preference for A- or T-ending codons, reflecting its lower genomic GC content.

These differences in codon bias may relate to variations in genomic composition, tRNA abundance, and adaptation to different environmental or metabolic conditions.

##Question 6

```r
library(Biostrings)
library(kmer)
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
```

```
library(ggplot2)
```

##Load CDS files (using the Biostrings function, which is often cleaner)

```
cds_ecoli <- readDNAStringSet("ecoli_cds.fa")
cds_ccoli <- readDNAStringSet("ccoli_cds.fa")
```

##Create protein object

```
prot_ecoli <- Biostrings::translate(cds_ecoli)
prot_ccoli <- Biostrings::translate(cds_ccoli)
```

##Convert to character vectors

```
prot_ecoli_vec <- as.character(prot_ecoli)
prot_ccoli_vec <- as.character(prot_ccoli)
```

##Define amino acid alphabet (excluding stop codons)

```
aa <- sort(unique(unlist(strsplit(paste0(prot_ecoli_vec, collapse = ""), ""))))
aa <- aa[aa != "*"]
```

##Function to count protein k-mers

```
# --- Function to count protein k-mers ---
get_protein_kmer_freq <- function(protein_seq, k) {
  # Collapse all sequences into one string
  all_seq <- paste(as.character(protein_seq), collapse = "")

  # Extract k-mers
  kmers <- substring(all_seq, 1:(nchar(all_seq)-k+1), k:(nchar(all_seq)))

  # Count frequency
  freq_table <- table(kmers)

  # Convert to data frame
  df <- as.data.frame(freq_table, stringsAsFactors = FALSE)
  colnames(df) <- c("kmer", "frequency")

  # Sort by descending frequency
  df <- df %>% arrange(desc(frequency))
  return(df)
}
```

##Compute 3-mer frequencies

```
freq3_ecoli <- get_protein_kmer_freq(prot_ecoli, 3)
freq3_ccoli <- get_protein_kmer_freq(prot_ccoli, 3)
```

##Identify top 10 over- and under-represented k-mers

```
over_ecoli <- head(freq3_ecoli, 10)
under_ecoli <- tail(freq3_ecoli, 10)
```

```
over_ccoli <- head(freq3_ccoli, 10)
under_ccoli <- tail(freq3_ccoli, 10)
```

##Add organism column

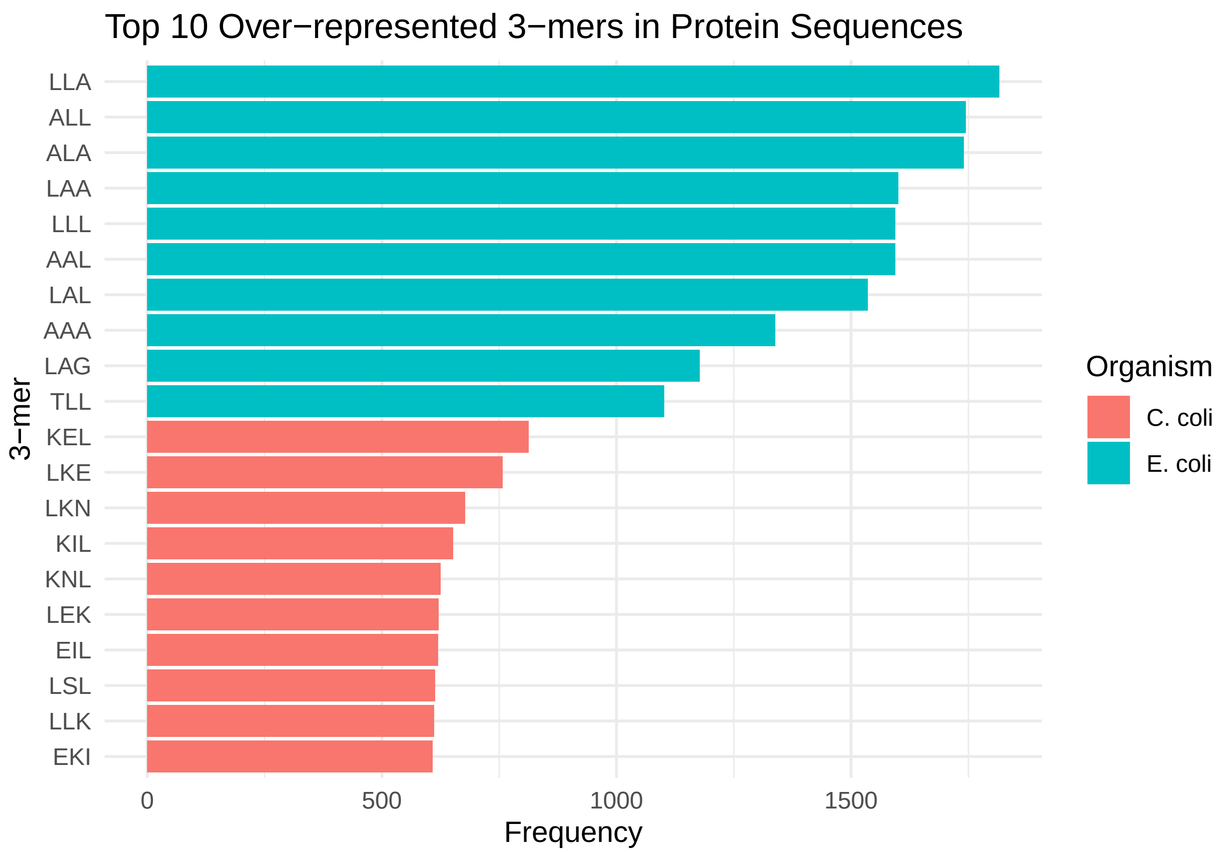
```
over_ecoli$Organism <- "E. coli"
under_ecoli$Organism <- "E. coli"
over_ccoli$Organism <- "C. coli"
under_ccoli$Organism <- "C. coli"
```

##Combine data frames for plotting

```
over_df <- rbind(over_ecoli, over_ccoli)
under_df <- rbind(under_ecoli, under_ccoli)
```

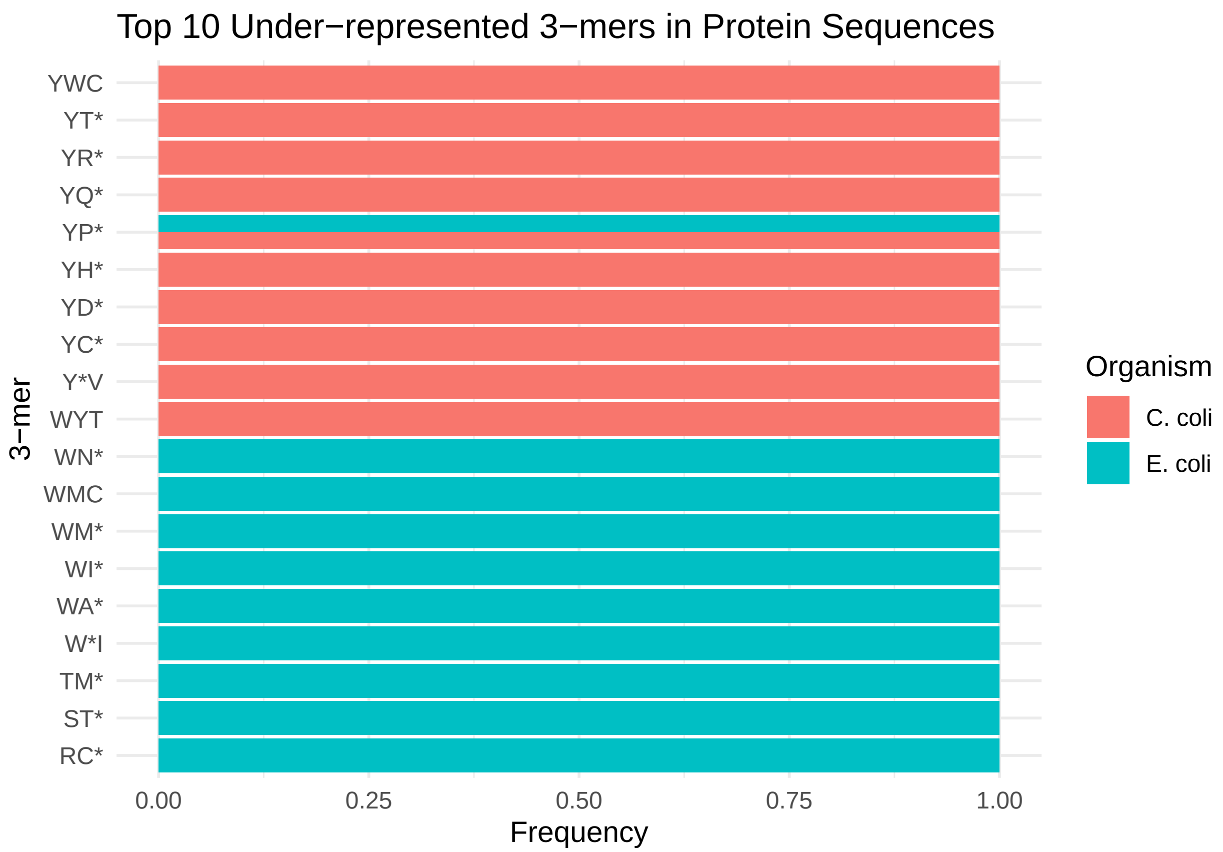
##Plot over-represented 3-mers

```
ggplot(over_df, aes(x = reorder(kmer, frequency), y = frequency, fill = Organism)) +
  geom_col(position = "dodge") +
  coord_flip() +
  labs(title = "Top 10 Over-represented 3-mers in Protein Sequences",
       x = "3-mer",
       y = "Frequency") +
  theme_minimal()
```



##Plot under-represented 3-mers

```
ggplot(under_df, aes(x = reorder(kmer, frequency), y = frequency, fill = Organism)) +
  geom_col(position = "dodge") +
  coord_flip() +
  labs(title = "Top 10 Under-represented 3-mers in Protein Sequences",
       x = "3-mer",
       y = "Frequency") +
  theme_minimal()
```


Top 10 Under−represented 3−mers in Protein Sequences
3−mer
YWC
YT*
YR*
YQ*
YP*
YH*
YD*
YC*
Y*V
WYT
WN*
WMC
WM*
WI*
WA*
W*I
TM*
ST*
RC*
0.00
0.25
0.50
0.75
1.00
Frequency
Organism
C. coli
E. coli