Assignment 4 Part-2

Teja Sri

2025-09-16

```
## ============================================================
## Part 2: Examining biological sequence diversity

# We will follow the Week09 notes workflow to download, process, and analyse the coding DNA sequences (

## ============================================================
```

```
suppressPackageStartupMessages({
  library(seqinr)   # sequence IO + GC, count, translate, uco
  library(R.utils)  # gunzip
})
```

```
## ------------------------------------------------------------
## 1) Download CDS FASTA files from Ensembl
# Here I download the complete set of coding DNA sequences (CDS) for E. coli K-12 MG1655 and Campylobac
## ------------------------------------------------------------

## E. coli K-12 MG1655 (Ensembl Bacteria Release-62)
url_ecoli <- "https://ftp.ensemblgenomes.ebi.ac.uk/pub/bacteria/release-62/fasta/bacteria_0_collection/

## Campylobacter coli (GCA_003780985) (your assigned genome)
url_cci  <- "https://ftp.ensemblgenomes.ebi.ac.uk/pub/bacteria/release-62/fasta/bacteria_46_collection/

## filenames
ec_gz <- "ecoli_cds.fa.gz"; ec_fa <- "ecoli_cds.fa"
cc_gz <- "ccoli_cds.fa.gz"; cc_fa <- "ccoli_cds.fa"

## download + unzip (exact pattern used in Week09)

#download.file(url_ecoli, destfile = ec_gz, mode = "wb")
#gunzip(ec_gz)                  # makes 'ecoli_cds.fa'

#download.file(url_cci, destfile = cc_gz, mode = "wb")
#gunzip(cc_gz)                  # makes 'ccoli_cds.fa'

## check
print(list.files())
```

```
##  [1] "224886843 assignment-4 part-1.Rmd" "224886843 assignment-4 part-2.Rmd"
##  [3] "224886843-assignment-4-part-2.Rmd" "ccoli_cds.fa"
##  [5] "condition_treated_results.csv"     "data"
##  [7] "ecoli_cds.fa"                      "ecoli_cds.fa.gz"
##  [9] "gene_expression.tsv"               "growth_data.csv"
## [11] "LICENSE"                           "myrepo.Rproj"
```

```
## [13] "README.md"
```

```
## ------------------------------------------------------------
## 2) Read CDS FASTA into R (seqinr)
#The downloaded FASTA files are read into R using read.fasta() from the seqinr package. Each entry corr
## ------------------------------------------------------------
ec_cds <- read.fasta(ec_fa)    # list of DNA sequences (A,T,G,C)
cc_cds <- read.fasta(cc_fa)

## sanity peek
str(head(ec_cds))
```

```
## List of 6
##  $ AAC76904: 'SeqFastadna' chr [1:300] "a" "t" "g" "a" ...
##   ..- attr(*, "name")= chr "AAC76904"
##   ..- attr(*, "Annot")= chr ">AAC76904 cds chromosome:ASM584v2:Chromosome:4112967:4113266:1 gene:b391
##  $ AAC75356: 'SeqFastadna' chr [1:1203] "a" "t" "g" "t" ...
##   ..- attr(*, "name")= chr "AAC75356"
##   ..- attr(*, "Annot")= chr ">AAC75356 cds chromosome:ASM584v2:Chromosome:2413470:2414672:1 gene:b229
##  $ AAC74435: 'SeqFastadna' chr [1:489] "a" "t" "g" "t" ...
##   ..- attr(*, "name")= chr "AAC74435"
##   ..- attr(*, "Annot")= chr ">AAC74435 cds chromosome:ASM584v2:Chromosome:1418671:1419159:1 gene:b135
##  $ AAC76238: 'SeqFastadna' chr [1:273] "a" "t" "g" "a" ...
##   ..- attr(*, "name")= chr "AAC76238"
##   ..- attr(*, "Annot")= chr ">AAC76238 cds chromosome:ASM584v2:Chromosome:3347966:3348238:1 gene:b320
##  $ AAC74021: 'SeqFastadna' chr [1:1146] "a" "t" "g" "a" ...
##   ..- attr(*, "name")= chr "AAC74021"
##   ..- attr(*, "Annot")= chr ">AAC74021 cds chromosome:ASM584v2:Chromosome:994843:995988:-1 gene:b0935
##  $ AAC75128: 'SeqFastadna' chr [1:3318] "a" "t" "g" "a" ...
##   ..- attr(*, "name")= chr "AAC75128"
##   ..- attr(*, "Annot")= chr ">AAC75128 cds chromosome:ASM584v2:Chromosome:2143266:2146583:1 gene:b200
```

```
str(head(cc_cds))
```

```
## List of 6
##  $ ENSB:6o1kUYbIUjq3iNP: 'SeqFastadna' chr [1:768] "a" "t" "g" "t" ...
##   ..- attr(*, "name")= chr "ENSB:6o1kUYbIUjq3iNP"
##   ..- attr(*, "Annot")= chr ">ENSB:6o1kUYbIUjq3iNP cds primary_assembly:PDT000395653.1:SAMN10282307-
##  $ ENSB:6zyx89wbXmnEC6G: 'SeqFastadna' chr [1:318] "a" "t" "g" "a" ...
##   ..- attr(*, "name")= chr "ENSB:6zyx89wbXmnEC6G"
##   ..- attr(*, "Annot")= chr ">ENSB:6zyx89wbXmnEC6G cds primary_assembly:PDT000395653.1:SAMN10282307-
##  $ ENSB:EzzayXvjb4TDLQ0: 'SeqFastadna' chr [1:450] "a" "t" "g" "a" ...
##   ..- attr(*, "name")= chr "ENSB:EzzayXvjb4TDLQ0"
##   ..- attr(*, "Annot")= chr ">ENSB:EzzayXvjb4TDLQ0 cds primary_assembly:PDT000395653.1:SAMN10282307-
##  $ ENSB:Qj7AKKM3QJI6gJf: 'SeqFastadna' chr [1:891] "a" "t" "g" "a" ...
##   ..- attr(*, "name")= chr "ENSB:Qj7AKKM3QJI6gJf"
##   ..- attr(*, "Annot")= chr ">ENSB:Qj7AKKM3QJI6gJf cds primary_assembly:PDT000395653.1:SAMN10282307-
##  $ ENSB:cVI02kcQINYATR6: 'SeqFastadna' chr [1:1323] "a" "t" "g" "a" ...
##   ..- attr(*, "name")= chr "ENSB:cVI02kcQINYATR6"
##   ..- attr(*, "Annot")= chr ">ENSB:cVI02kcQINYATR6 cds primary_assembly:PDT000395653.1:SAMN10282307-
##  $ ENSB:Jvag1SFo15f3vTZ: 'SeqFastadna' chr [1:1827] "a" "t" "g" "g" ...
##   ..- attr(*, "name")= chr "ENSB:Jvag1SFo15f3vTZ"
##   ..- attr(*, "Annot")= chr ">ENSB:Jvag1SFo15f3vTZ cds primary_assembly:PDT000395653.1:SAMN10282307-
```

```
## ------------------------------------------------------------
## 3) How many coding sequences (CDS) in each organism? (table)
```

```r
# calculated the number of CDS entries for each organism and present the results in a summary table.
## ------------------------------------------------------------
n_ec <- length(ec_cds)
n_cc <- length(cc_cds)

cds_count_tbl <- data.frame(
  Organism = c("E. coli K-12 MG1655", "Campylobacter coli (GCA_003780985)"),
  CDS_count = c(n_ec, n_cc)
)
print(cds_count_tbl)
```

```
##                             Organism CDS_count
## 1                E. coli K-12 MG1655      4239
## 2 Campylobacter coli (GCA_003780985)      1976
```

```r
## ------------------------------------------------------------
## 4) How much coding DNA in total? (table)
# By summing the lengths of all CDS, we obtain the total number of coding base pairs in each organism a
## ------------------------------------------------------------
ec_len <- as.numeric(summary(ec_cds)[,1])
cc_len <- as.numeric(summary(cc_cds)[,1])

total_bp_tbl <- data.frame(
  Organism = c("E. coli K-12 MG1655", "Campylobacter coli (GCA_003780985)"),
  Total_coding_bp = c(sum(ec_len), sum(cc_len))
)
print(total_bp_tbl)
```

```
##                             Organism Total_coding_bp
## 1                E. coli K-12 MG1655         3978528
## 2 Campylobacter coli (GCA_003780985)         1726818
```

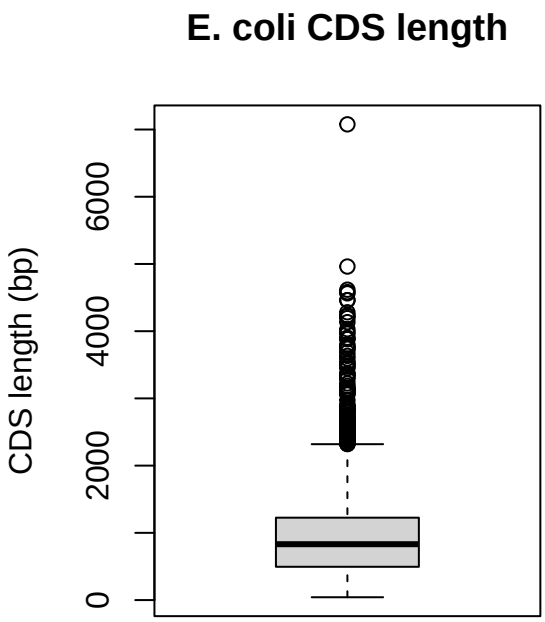
```r
## ------------------------------------------------------------
## 5) CDS length distributions + mean/median (boxplot + stats)
# Examine the distribution of coding sequence lengths, and compute mean and median lengths for both org
## ------------------------------------------------------------
cat("E. coli mean len = ", mean(ec_len), " median = ", median(ec_len), "\n")
```

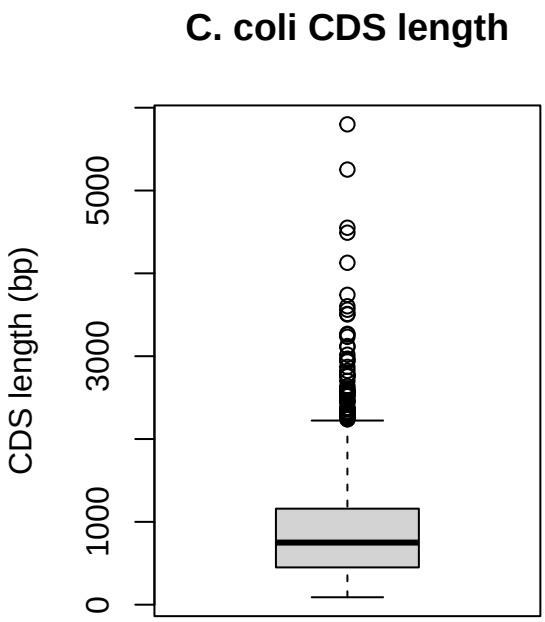
```
## E. coli mean len =  938.5534  median =  831
```

```r
cat("C. coli mean len = ", mean(cc_len), " median = ", median(cc_len), "\n")
```

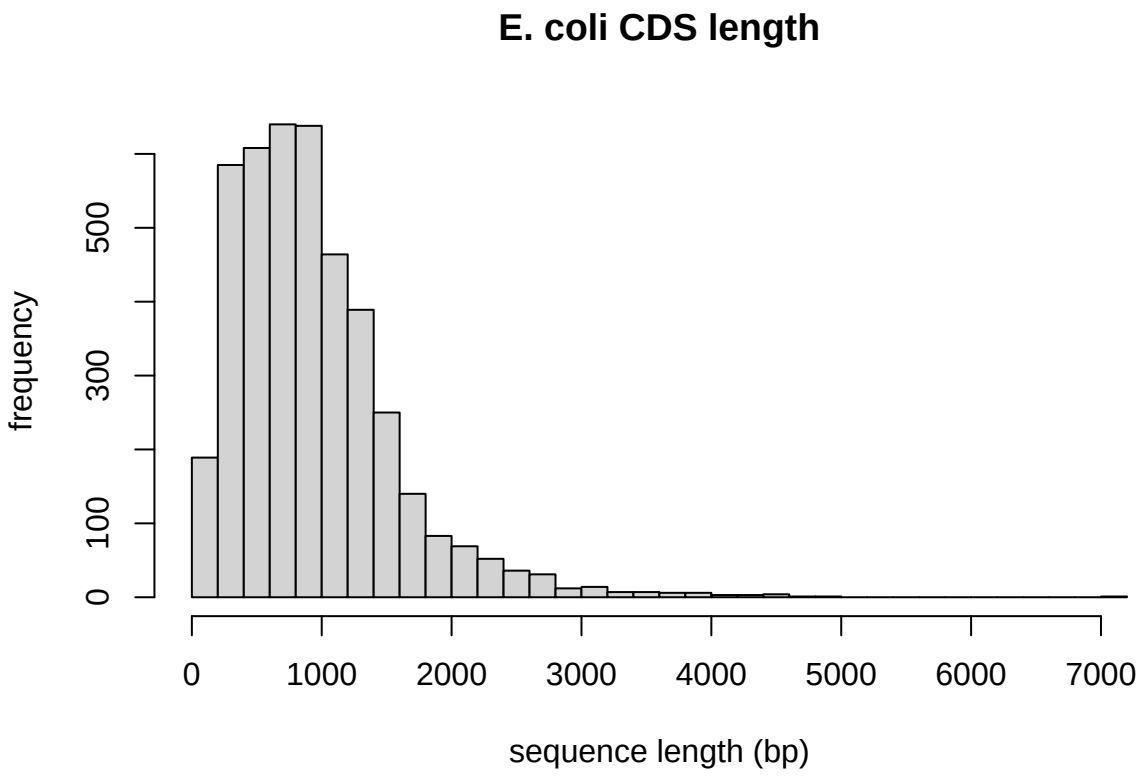
```
## C. coli mean len =  873.8957  median =  750
```

```r
par(mfrow = c(1,2))
boxplot(ec_len, ylab = "CDS length (bp)", main = "E. coli CDS length")
boxplot(cc_len, ylab = "CDS length (bp)", main = "C. coli CDS length")
```

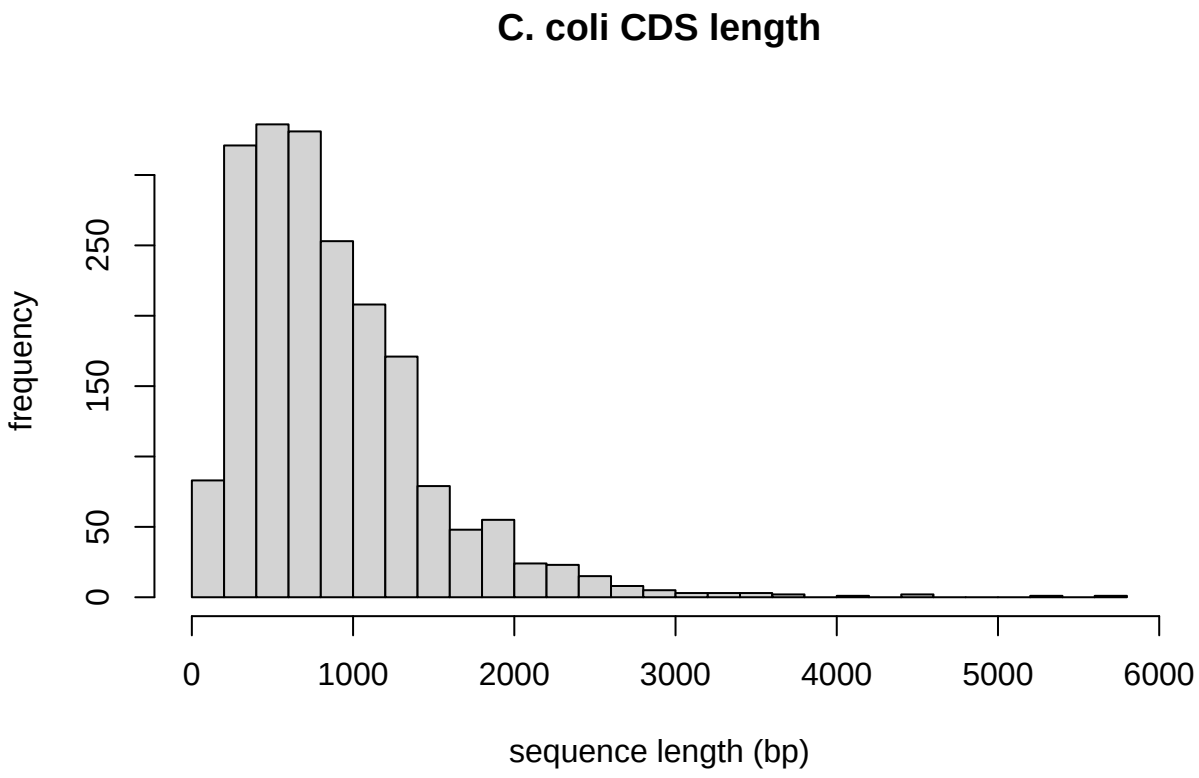




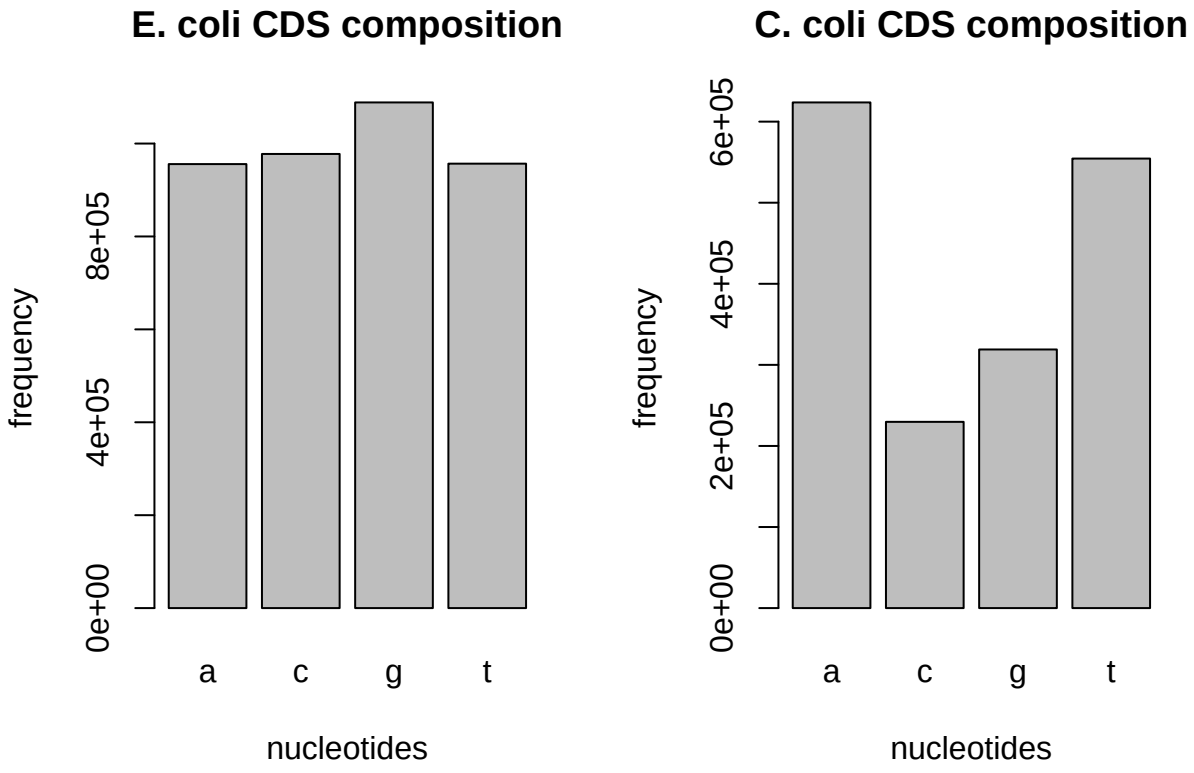

```
par(mfrow = c(1,1))

## (optional) histogram like in notes
hist(ec_len, xlab="sequence length (bp)", ylab="frequency",
     main="E. coli CDS length", breaks = 40)
```




```
hist(cc_len, xlab="sequence length (bp)", ylab="frequency",
     main="C. coli CDS length", breaks = 40)
```



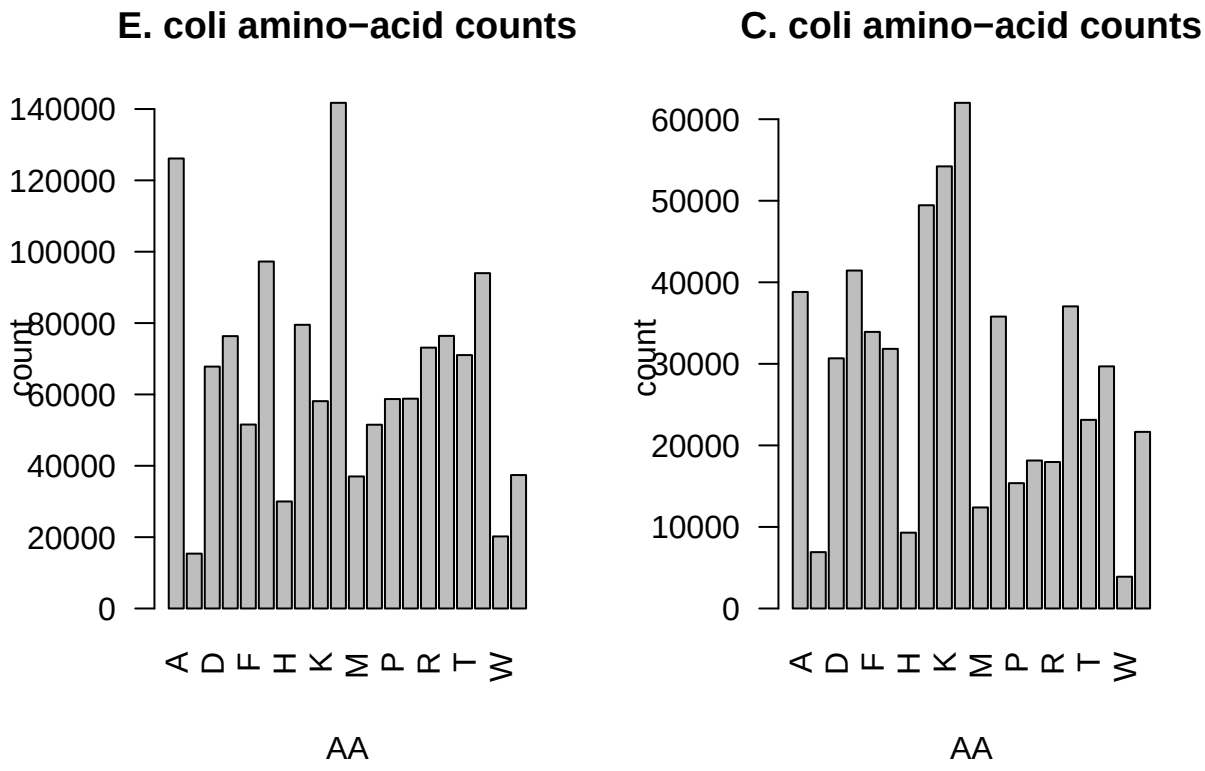

```
## ------------------------------------------------------------
## 6) Nucleotide frequency (total coding DNA) + barplots
# calculated the nucleotide composition (A, C, G, T) across all CDS, and display the absolute counts in
# Should print lower-case 'a','t','g','c' etc.
# check what function you're calling
## ------------------------------------------------------------
## 6) Nucleotide frequency (total coding DNA) + barplots
## ------------------------------------------------------------

# Flatten DNA into one long vector of lowercase bases
ec_dna <- unlist(ec_cds)
cc_dna <- unlist(cc_cds)

# Count nucleotides (must use lowercase alphabet here)
ec_nt_counts <- seqinr::count(ec_dna, wordsize = 1, alphabet = c("a","c","g","t"))
cc_nt_counts <- seqinr::count(cc_dna, wordsize = 1, alphabet = c("a","c","g","t"))

# Barplots
par(mfrow = c(1,2))
barplot(ec_nt_counts, main = "E. coli CDS composition", ylab = "frequency", xlab = "nucleotides")
barplot(cc_nt_counts, main = "C. coli CDS composition", ylab = "frequency", xlab = "nucleotides")
```




```
par(mfrow = c(1,1))
```

```
## ------------------------------------------------------------
## 7) Protein translation + amino acid frequency (+ barplots)
# The CDS are translated into protein sequences. We then compute amino acid frequencies and present the
## ------------------------------------------------------------
# Translate CDS to proteins
ec_prot <- unlist(lapply(ec_cds, seqinr::translate))
cc_prot <- unlist(lapply(cc_cds, seqinr::translate))

# Remove stop codons ("*")
ec_prot <- ec_prot[ec_prot != "*"]
cc_prot <- cc_prot[cc_prot != "*"]

# Standard uppercase amino acid alphabet
aa <- c("A","C","D","E","F","G","H","I","K","L",
        "M","N","P","Q","R","S","T","V","W","Y")

# Count amino acids
ec_aa_counts <- seqinr::count(ec_prot, wordsize = 1, alphabet = aa)
cc_aa_counts <- seqinr::count(cc_prot, wordsize = 1, alphabet = aa)

# Barplots
par(mfrow = c(1,2))
barplot(ec_aa_counts, main = "E. coli amino-acid counts",
        ylab = "count", xlab = "AA", las = 2)
```

```r
barplot(cc_aa_counts, main = "C. coli amino-acid counts",
        ylab = "count", xlab = "AA", las = 2)
```




```r
par(mfrow = c(1,1))
```

```r
## ------------------------------------------------------------
## 8) Codon usage & bias (RSCU) using uco() (as in notes)
##    We pass the whole coding DNA (unlisted).
# Codon usage tables are generated, and relative synonymous codon usage (RSCU) values are calculated. T
## ------------------------------------------------------------
ec_uco_raw  <- uco(ec_dna)                                    # counts
cc_uco_raw  <- uco(cc_dna)

ec_rscu_df  <- uco(ec_dna, index = "rscu", as.data.frame = TRUE)
cc_rscu_df  <- uco(cc_dna, index = "rscu", as.data.frame = TRUE)

## quick look at most over/under (by RSCU)
ec_top_over  <- head(ec_rscu_df[order(-ec_rscu_df$RSCU), ], 10)
ec_top_under <- head(ec_rscu_df[order(ec_rscu_df$RSCU), ], 10)

cc_top_over  <- head(cc_rscu_df[order(-cc_rscu_df$RSCU), ], 10)
cc_top_under <- head(cc_rscu_df[order(cc_rscu_df$RSCU), ], 10)

print(head(ec_rscu_df))
```

```
##      AA codon   eff        freq      RSCU
```

```
## aaa Lys   aaa 44592 0.033624496 1.5346652
## aac Asn   aac 28454 0.021455674 1.1049453
## aag Lys   aag 13521 0.010195479 0.4653348
## aat Asn   aat 23049 0.017380046 0.8950547
## aca Thr   aca  9116 0.006873899 0.5133967
## acc Thr   acc 31139 0.023480292 1.7536924
```

```
print(head(cc_rscu_df))
```

```
##      AA codon   eff        freq      RSCU
## aaa Lys   aaa 45376 0.078831701 1.6739896
## aac Asn   aac  5240 0.009103449 0.2927783
## aag Lys   aag  8837 0.015352515 0.3260104
## aat Asn   aat 30555 0.053083185 1.7072217
## aca Thr   aca  7410 0.012873389 1.2815081
## acc Thr   acc  4100 0.007122928 0.7090665
```

```
## simple side-by-side comparison table (mean RSCU by codon)
## merge by codon to see differences
rscu_cmp <- merge(
  ec_rscu_df[, c("codon", "RSCU")],
  cc_rscu_df[, c("codon", "RSCU")],
  by = "codon", suffixes = c("_Ecoli", "_Ccoli")
)
## largest absolute differences in RSCU
rscu_cmp$abs_diff <- abs(rscu_cmp$RSCU_Ecoli - rscu_cmp$RSCU_Ccoli)
rscu_cmp <- rscu_cmp[order(-rscu_cmp$abs_diff), ]
head(rscu_cmp, 12)
```

```
##    codon RSCU_Ecoli RSCU_Ccoli abs_diff
## 9    aga  0.2111584  3.2032738 2.992115
## 31   ctg  2.9935864  0.0835067 2.910080
## 23   ccg  2.1174787  0.2108003 1.906678
## 61   tta  0.7756807  2.6191882 1.843507
## 24   cct  0.6286201  2.3096942 1.681074
## 26   cgc  2.4161344  0.8618674 1.554267
## 40   gct  0.6401484  1.8953497 1.255201
## 28   cgt  2.2961524  1.0572908 1.238862
## 32   ctt  0.6174796  1.7420291 1.124550
## 19   cag  1.3060426  0.1984018 1.107641
## 17   caa  0.6939574  1.8015982 1.107641
## 8    act  0.6583034  1.7325436 1.074240
```

```
## ------------------------------------------------------------
## 9) K-mer profiling on proteins (k = 3..5) as in notes
##    "Identify 10 over- and under-represented k-mers in C. coli"
##    We'll do k=3,4,5 and show the top/bottom for each, then
##    check whether they are similarly ranked in E. coli.
## ------------------------------------------------------------

## Helper: return top/bottom 10 kmers that actually occur
top_bottom_kmers <- function(aa_vec, k, alphabet) {
  tab <- seqinr::count(aa_vec, wordsize = k, alphabet = alphabet, freq = TRUE)
  tab <- sort(tab, decreasing = TRUE)
  over  <- head(tab, 10)
  under <- head(tab[tab > 0], 10)  # keep only kmers with nonzero frequency
```

```
  list(over = over, under = under, allfreq = tab)
}

## Run for C. coli (your organism of interest)
km3_cc <- top_bottom_kmers(cc_prot, 3, aa)
km4_cc <- top_bottom_kmers(cc_prot, 4, aa)
km5_cc <- top_bottom_kmers(cc_prot, 5, aa)

cat("\nTop 10 C. coli tri-peptides (k=3):\n");  print(km3_cc$over)
```

```
##
## Top 10 C. coli tri-peptides (k=3):
```

```
##
##         KEL         LKE         LKN         KIL         KNL         EIL
## 0.001422525 0.001324900 0.001181951 0.001143598 0.001091300 0.001087813
##         LEK         LLK         LSL         EKI
## 0.001082583 0.001077353 0.001073867 0.001059920
```

```
cat("\nBottom 10 C. coli tri-peptides (k=3):\n"); print(km3_cc$under)
```

```
##
## Bottom 10 C. coli tri-peptides (k=3):
```

```
##
##         KEL         LKE         LKN         KIL         KNL         EIL
## 0.001422525 0.001324900 0.001181951 0.001143598 0.001091300 0.001087813
##         LEK         LLK         LSL         EKI
## 0.001082583 0.001077353 0.001073867 0.001059920
```

```
cat("\nTop 10 C. coli tetra-peptides (k=4):\n");  print(km4_cc$over)
```

```
##
## Top 10 C. coli tetra-peptides (k=4):
```

```
##
##         EILK         ELLK         LKEL         AKEL         KELK         LEKL
## 0.0001969921 0.0001847891 0.0001847891 0.0001830458 0.0001743293 0.0001743293
##         LENL         KELL         ENLK         EELK
## 0.0001638696 0.0001621263 0.0001603830 0.0001534098
```

```
cat("\nBottom 10 C. coli tetra-peptides (k=4):\n"); print(km4_cc$under)
```

```
##
## Bottom 10 C. coli tetra-peptides (k=4):
```

```
##
##         EILK         ELLK         LKEL         AKEL         KELK         LEKL
## 0.0001969921 0.0001847891 0.0001847891 0.0001830458 0.0001743293 0.0001743293
##         LENL         KELL         ENLK         EELK
## 0.0001638696 0.0001621263 0.0001603830 0.0001534098
```

```
cat("\nTop 10 C. coli penta-peptides (k=5):\n");  print(km5_cc$over)
```

```
##
## Top 10 C. coli penta-peptides (k=5):
```

```
##
##        GKSTL        KELLE        DEILK        KEELK        KELKK        LKEKL
```

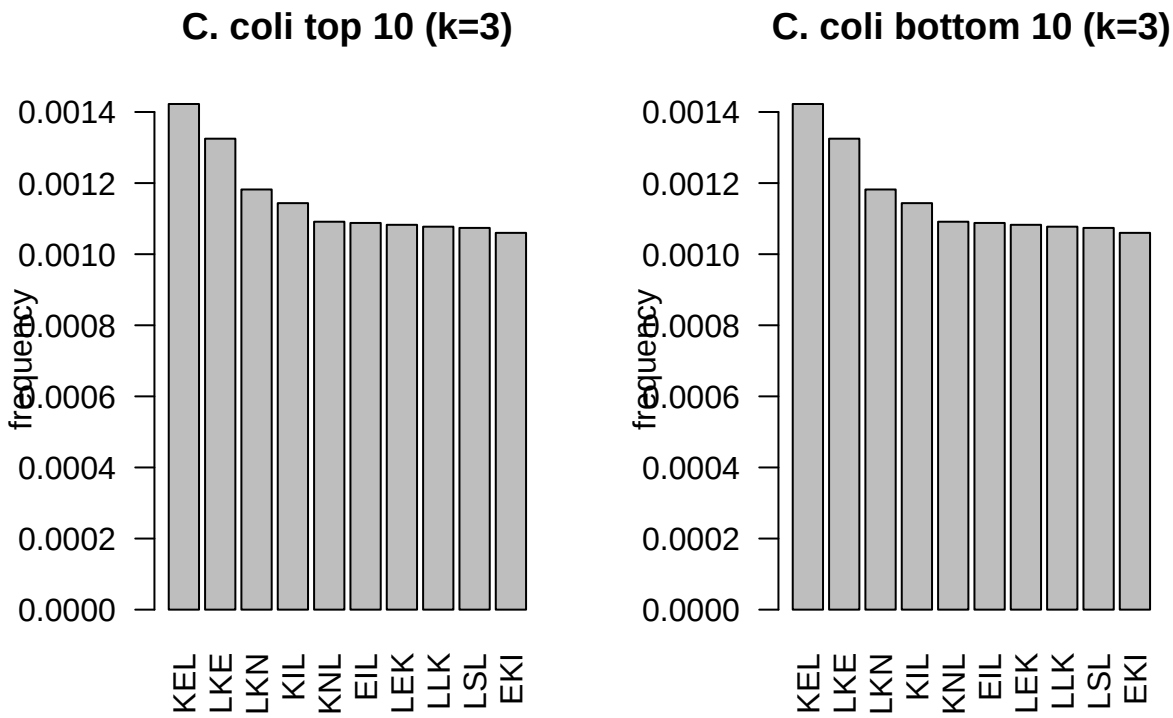
```
## 4.532570e-05 4.183911e-05 3.486592e-05 3.137933e-05 3.137933e-05 3.137933e-05
##        SSSSS        ELEKL        ELLEK        KELAK
## 3.137933e-05 2.963603e-05 2.963603e-05 2.963603e-05
```

```
cat("\nBottom 10 C. coli penta-peptides (k=5):\n"); print(km5_cc$under)
```

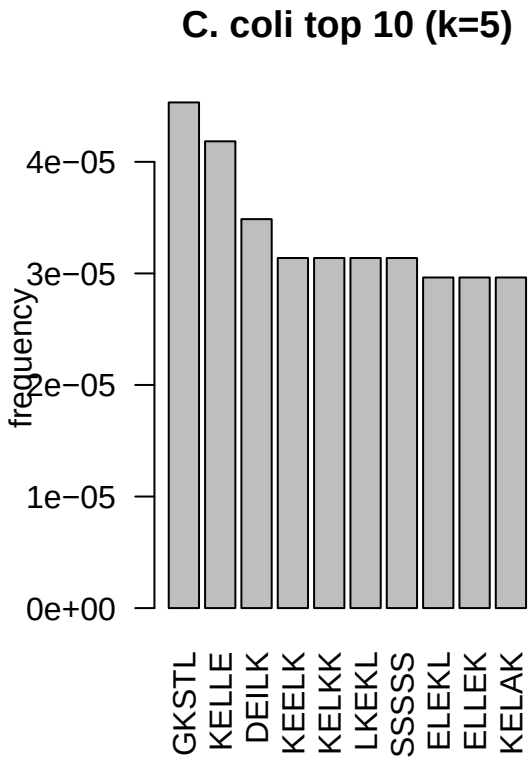
```
##
## Bottom 10 C. coli penta-peptides (k=5):
```

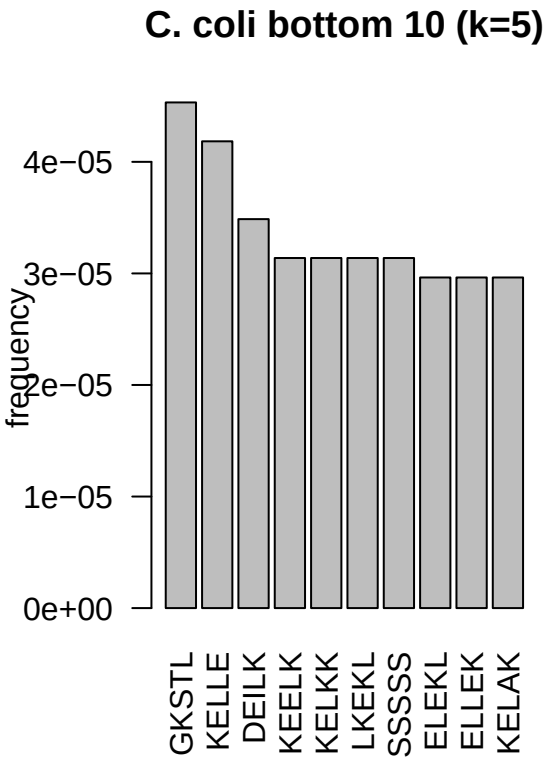
```
##
##        GKSTL        KELLE        DEILK        KEELK        KELKK        LKEKL
## 4.532570e-05 4.183911e-05 3.486592e-05 3.137933e-05 3.137933e-05 3.137933e-05
##        SSSSS        ELEKL        ELLEK        KELAK
## 3.137933e-05 2.963603e-05 2.963603e-05 2.963603e-05
```

```
## Quick barplots for k=3 and k=5
par(mfrow = c(1,2))
barplot(km3_cc$over,  las = 2, main = "C. coli top 10 (k=3)", ylab = "frequency")
barplot(km3_cc$under, las = 2, main = "C. coli bottom 10 (k=3)", ylab = "frequency")
```




```
par(mfrow = c(1,2))
barplot(km5_cc$over,  las = 2, main = "C. coli top 10 (k=5)", ylab = "frequency")
barplot(km5_cc$under, las = 2, main = "C. coli bottom 10 (k=5)", ylab = "frequency")
```






```
par(mfrow = c(1,1))

## ------------------------------------------------------------
## Compare C. coli tri-peptide rankings vs E. coli
## ------------------------------------------------------------

## E. coli tri-peptide frequencies (same alphabet)
ec_tri_freq <- seqinr::count(ec_prot, wordsize = 3, alphabet = aa, freq = TRUE)

## build comparison tables for the C. coli top/bottom 10 (k=3)
compare_kmers <- function(named_freq_vec, other_freq_table) {
  kmers <- names(named_freq_vec)
  data.frame(
    kmer        = kmers,
    freq_Ccoli  = as.numeric(named_freq_vec),
    freq_Ecoli  = as.numeric(other_freq_table[kmers]),
    row.names   = NULL
  )
}

cmp_over_3  <- compare_kmers(km3_cc$over,  ec_tri_freq)
cmp_under_3 <- compare_kmers(km3_cc$under, ec_tri_freq)

cat("\nC. coli top 10 tri-peptides vs E. coli frequency:\n")
```

```
##
```

```
## C. coli top 10 tri-peptides vs E. coli frequency:
```

```
print(cmp_over_3)
```

```
##    kmer  freq_Ccoli   freq_Ecoli
## 1   KEL 0.001422525 0.0003162031
## 2   LKE 0.001324900 0.0003585653
## 3   LKN 0.001181951 0.0002609811
## 4   KIL 0.001143598 0.0002095413
## 5   KNL 0.001091300 0.0001868473
## 6   EIL 0.001087813 0.0003252807
## 7   LEK 0.001082583 0.0004243779
## 8   LLK 0.001077353 0.0005174233
## 9   LSL 0.001073867 0.0007239388
## 10  EKI 0.001059920 0.0002329918
```

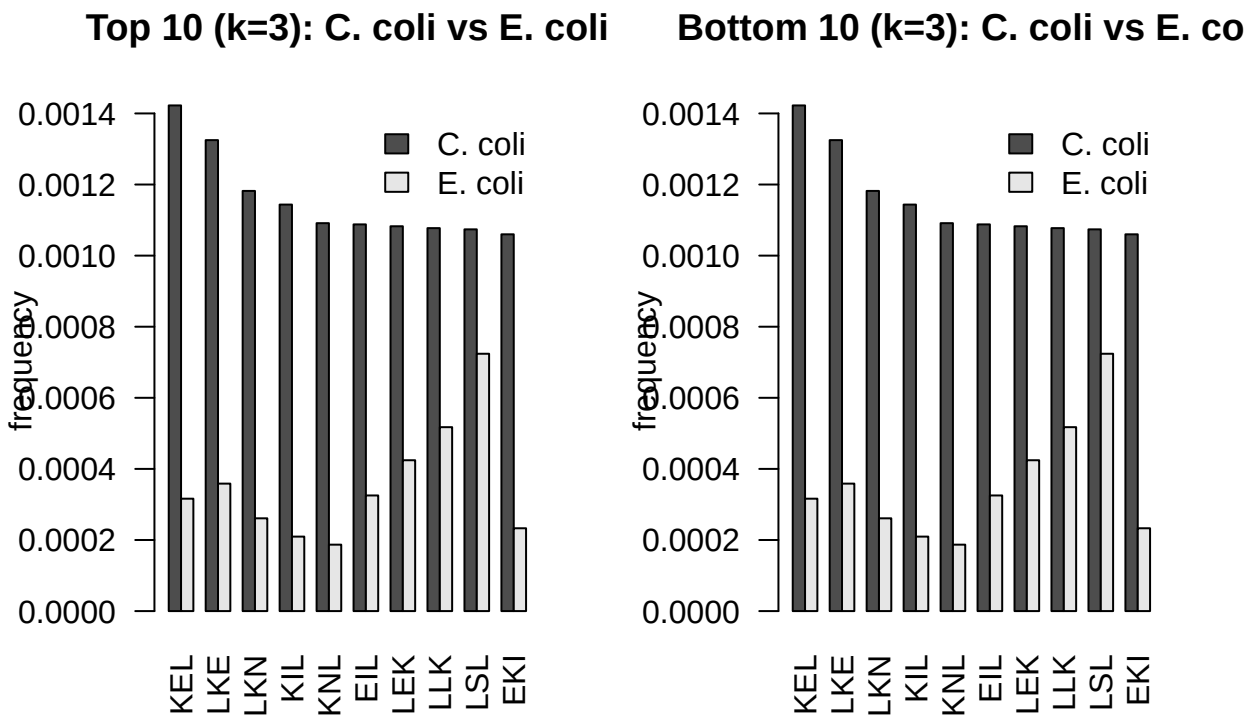
```
cat("\nC. coli bottom 10 tri-peptides vs E. coli frequency:\n")
```

```
##
## C. coli bottom 10 tri-peptides vs E. coli frequency:
```

```
print(cmp_under_3)
```

```
##    kmer  freq_Ccoli   freq_Ecoli
## 1   KEL 0.001422525 0.0003162031
## 2   LKE 0.001324900 0.0003585653
## 3   LKN 0.001181951 0.0002609811
## 4   KIL 0.001143598 0.0002095413
## 5   KNL 0.001091300 0.0001868473
## 6   EIL 0.001087813 0.0003252807
## 7   LEK 0.001082583 0.0004243779
## 8   LLK 0.001077353 0.0005174233
## 9   LSL 0.001073867 0.0007239388
## 10  EKI 0.001059920 0.0002329918
```

```
## Optional: side-by-side barplots to visualise the comparison (k=3)
par(mfrow = c(1,2))
barplot(rbind(cmp_over_3$freq_Ccoli, cmp_over_3$freq_Ecoli),
        beside = TRUE, names.arg = cmp_over_3$kmer, las = 2,
        main = "Top 10 (k=3): C. coli vs E. coli", ylab = "frequency")
legend("topright", bty = "n", legend = c("C. coli", "E. coli"), fill = gray.colors(2))
barplot(rbind(cmp_under_3$freq_Ccoli, cmp_under_3$freq_Ecoli),
        beside = TRUE, names.arg = cmp_under_3$kmer, las = 2,
        main = "Bottom 10 (k=3): C. coli vs E. coli", ylab = "frequency")
legend("topright", bty = "n", legend = c("C. coli", "E. coli"), fill = gray.colors(2))
```




```
par(mfrow = c(1,1))
```

```
## ------------------------------------------------------------
## 10) Small summary tables that I can use to paste into my report
# Finally, I have prepared the summary tables of CDS counts, total coding base pairs, and mean/median C
## ------------------------------------------------------------
cat("\n== Table: Number of coding sequences ==\n")
```

```
##
## == Table: Number of coding sequences ==
```

```
print(cds_count_tbl)
```

```
##                             Organism CDS_count
## 1                E. coli K-12 MG1655      4239
## 2 Campylobacter coli (GCA_003780985)      1976
```

```
cat("\n== Table: Total coding DNA (bp) ==\n")
```

```
##
## == Table: Total coding DNA (bp) ==
```

```
print(total_bp_tbl)
```

```
##                             Organism Total_coding_bp
## 1                E. coli K-12 MG1655         3978528
## 2 Campylobacter coli (GCA_003780985)         1726818
```

```r
cat("\n== Mean/Median CDS length (bp) ==\n")
```

```
##
## == Mean/Median CDS length (bp) ==
```

```r
mm_tbl <- data.frame(
  Organism   = c("E. coli K-12 MG1655", "Campylobacter coli"),
  Mean_bp    = c(mean(ec_len), mean(cc_len)),
  Median_bp  = c(median(ec_len), median(cc_len))
)
print(mm_tbl)
```

```
##              Organism  Mean_bp Median_bp
## 1 E. coli K-12 MG1655 938.5534       831
## 2  Campylobacter coli 873.8957       750
```