Comparative Codon Usage: Human vs Worm

Alice Jeon

2025-11-15

1 Introduction

This analysis compares codon usage bias (RSCU values) between **Human** and **Worm** genomes. It computes codon frequencies, performs normality tests, and visualizes results through histograms, QQ plots, and heatmaps.

2 Load Libraries

```
library(Biostrings)
library(dplyr)
library(ggplot2)
library(plotly)
library(tidyr)
library(nortest)
library(htmltools)
```

3 Define Genome Files

```
genomes <- list(
  Human = "Homo_sapiens.GRCh38.cdna.all.fa.gz",
  Worm = "Worm Caenorhabditis_elegans.WBcel235.cdna.all.fa.gz"
)
```

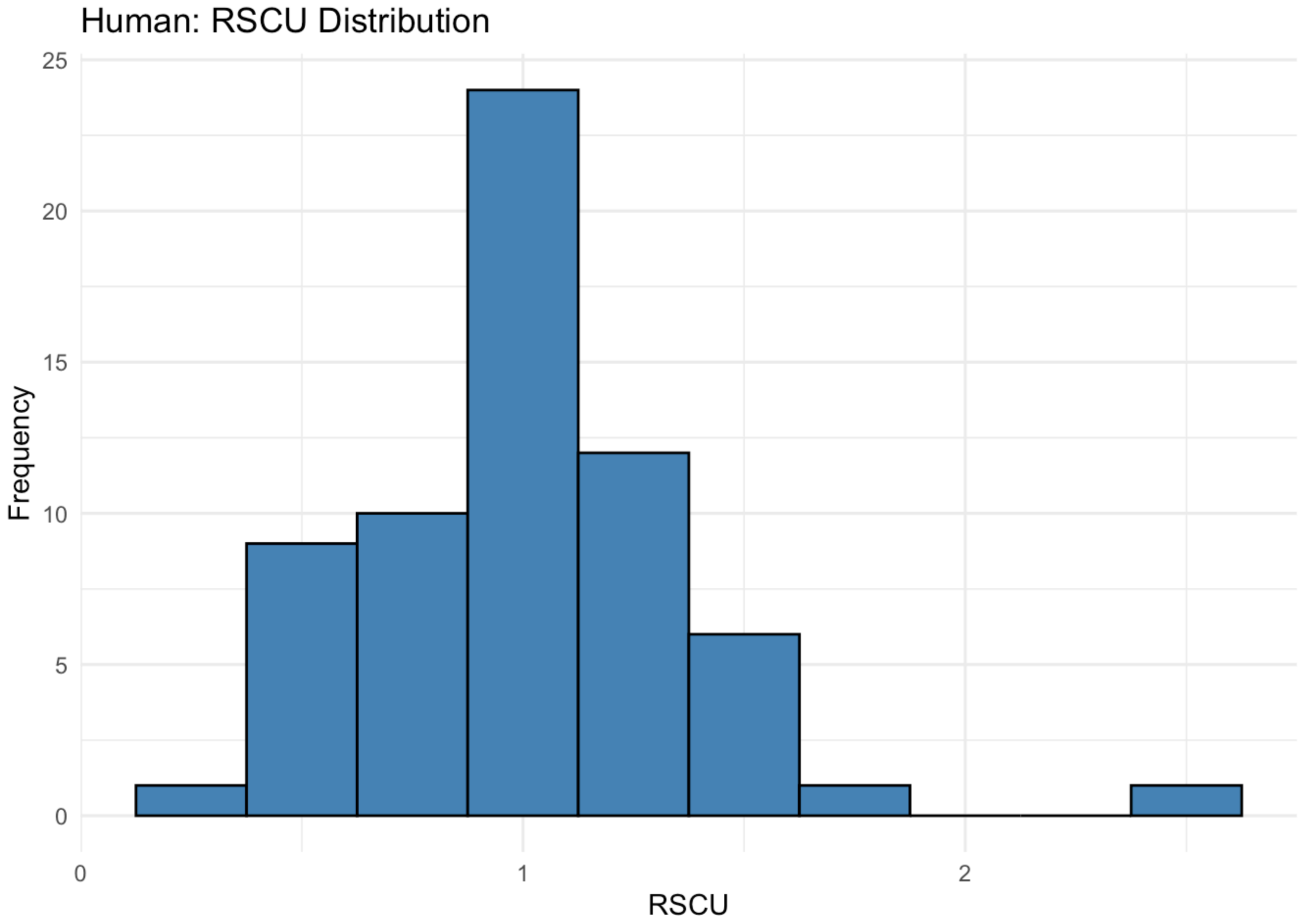
4 Function to Extract Codons

```r
extract_cds_codons <- function(seq) {
  s <- as.character(seq)
  start_pos <- regexpr("ATG", s)[1]
  if (start_pos == -1) return(character(0))
  coding_seq <- substring(s, start_pos)
  codons <- substring(coding_seq,
                      seq(1, nchar(coding_seq), 3),
                      seq(3, nchar(coding_seq), 3))
  codons <- codons[nchar(codons) == 3]
  stops <- c("TAA", "TAG", "TGA")
  stop_index <- which(codons %in% stops)
  if (length(stop_index) > 0) codons <- codons[1:min(stop_index)]
  return(codons)
}

codon_table <- GENETIC_CODE
aa_df <- data.frame(codon = names(codon_table), aa = as.vector(codon_table))
```

5 Process Genomes

```
results_list <- list()
top_bottom_list <- list()

for (genome_name in names(genomes)) {
  cat("\nProcessing genome:", genome_name, "\n")
  fasta_file <- genomes[[genome_name]]
  cdna_seqs <- readDNAStringSet(fasta_file)

  all_codons <- unlist(lapply(cdna_seqs, extract_cds_codons), use.names = FALSE)

  codon_df <- data.frame(codon = all_codons) %>%
    count(codon, name = "obs_count") %>%
    inner_join(aa_df, by = "codon")

  rscu_df <- codon_df %>%
    group_by(aa) %>%
    mutate(expected = sum(obs_count) / n(),
           RSCU = obs_count / expected) %>%
    ungroup() %>%
    mutate(aa2 = ifelse(codon %in% c("TAA","TAG","TGA"), "*", aa))

  results_list[[genome_name]] <- rscu_df

  # Histogram
  print(ggplot(rscu_df, aes(x = RSCU)) +
    geom_histogram(binwidth = 0.25, fill = "steelblue", color = "black") +
    labs(title = paste0(genome_name, ": RSCU Distribution"), x = "RSCU", y = "Frequ
ency") +
    theme_minimal())

  ad_test <- ad.test(rscu_df$RSCU)
  print(ad_test)

  if(ad_test$p.value > 0.05){
    print("RSCU Distribution is normal")
  } else {
    print("RSCU Distribution is not normal")
  }

  print(ggplot(rscu_df, aes(sample = RSCU)) +
    stat_qq(color = "darkred") +
    stat_qq_line(color = "black") +
    labs(title = "Q-Q Plot of RSCU Values", x = "Theoretical Quantiles", y = "Sampl
e Quantiles") +
    theme_minimal())

  rscu_df <- rscu_df %>% mutate(RSCU_one = RSCU == 1)

  print(ggplot(rscu_df, aes(x = reorder(codon, RSCU), y = RSCU, fill = aa)) +
    geom_bar(stat = "identity") +
    coord_flip() +
    labs(title = paste0(genome_name, ": Codon Usage (RSCU)"), x = "Codon", y = "RSC
U") +
    theme_minimal())

  heat_plot <- ggplot(rscu_df, aes(x = aa2, y = codon, fill = RSCU,
                                   text = paste0("Codon: ", codon,
                                                 "<br>Amino Acid: ", aa2,
                                                 "<br>RSCU: ", round(RSCU, 3)))) +
    geom_tile(color = "white") +
    geom_tile(data = subset(rscu_df, RSCU_one == TRUE),
              aes(x = aa2, y = codon),
              fill = NA, color = "grey", size = 0.6, inherit.aes = FALSE) +
    scale_fill_gradient2(low = "blue", mid = "white", high = "red", midpoint = 1) +
    labs(title = paste0(genome_name, ": RSCU Heatmap"),
         x = "Amino Acid", y = "Codon", fill = "RSCU") +
    theme_minimal()

  #print(heat_plot)
  #print(ggplotly(heat_plot, tooltip = "text"))
  p <- ggplotly(heat_plot, tooltip = "text")
  p


  codon_table_df <- rscu_df %>%
    group_by(aa2) %>%
    summarize(
      most_used = codon[which.max(RSCU)],
      most_used_RSCU = RSCU[which.max(RSCU)],
      least_used = codon[which.min(RSCU)],
      least_used_RSCU = RSCU[which.min(RSCU)]
    ) %>%
    ungroup() %>%
    mutate(Genome = genome_name)

  top_bottom_list[[genome_name]] <- codon_table_df
}
```

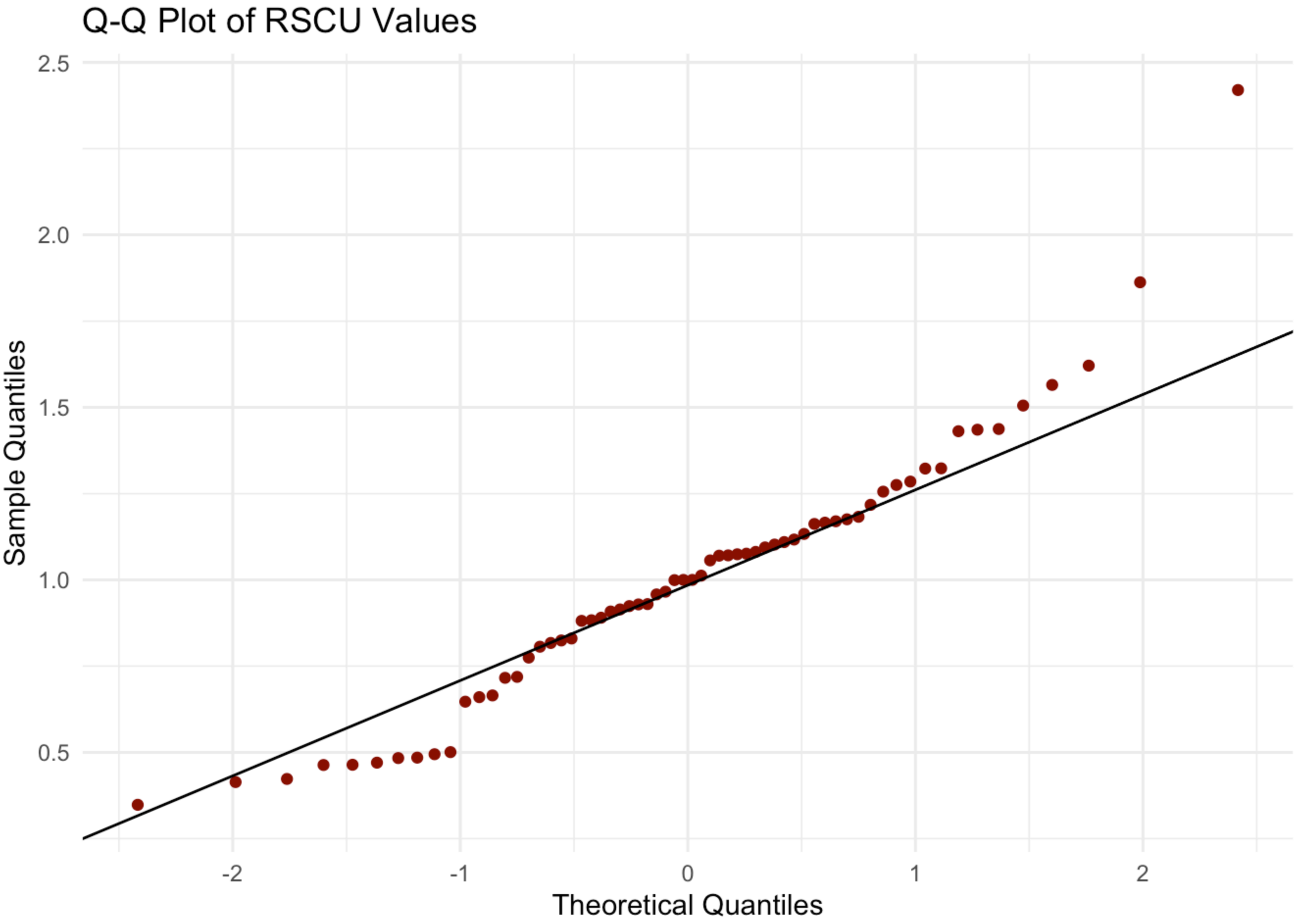
Processing genome: Human

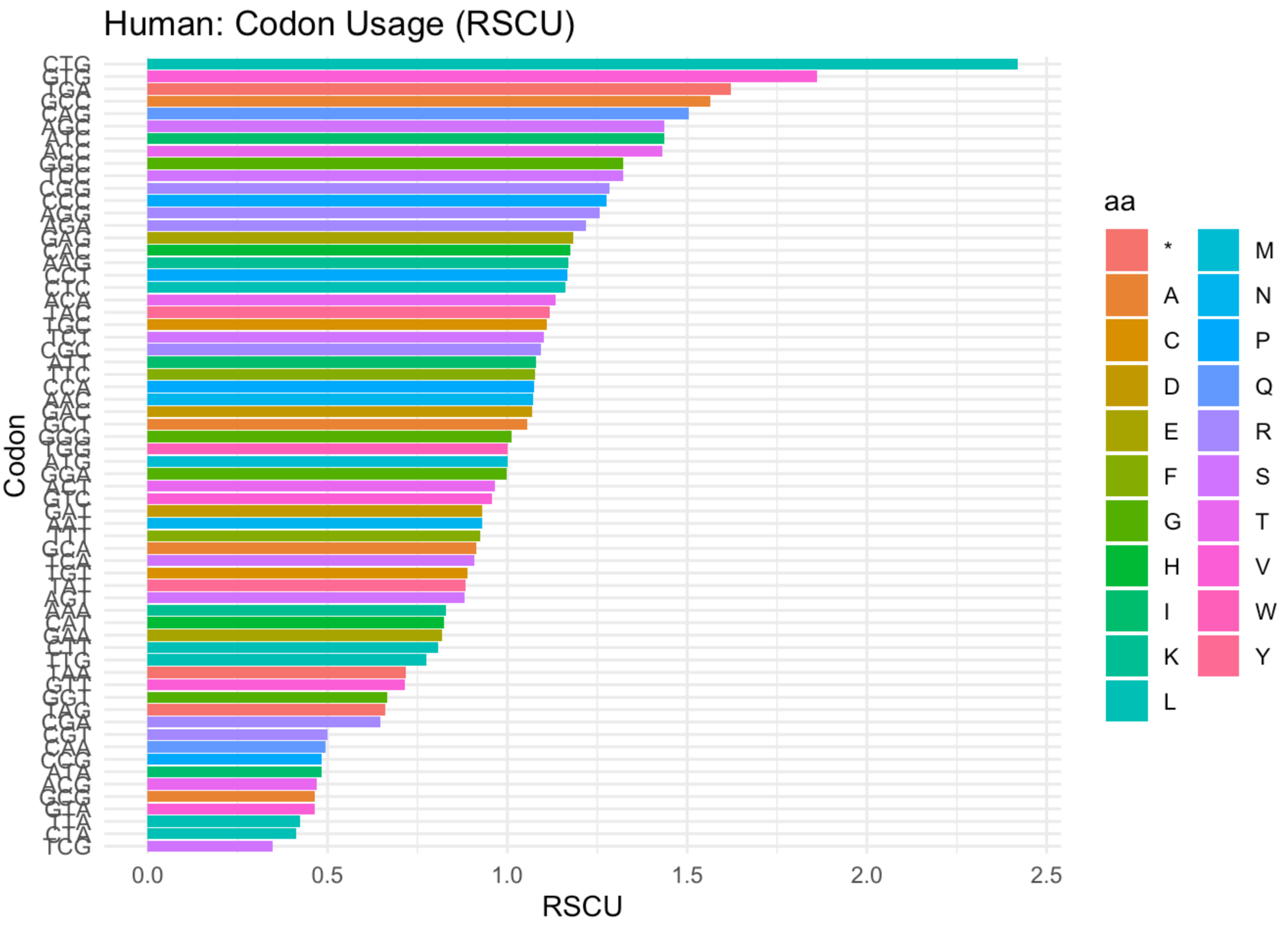


Anderson-Darling normality test

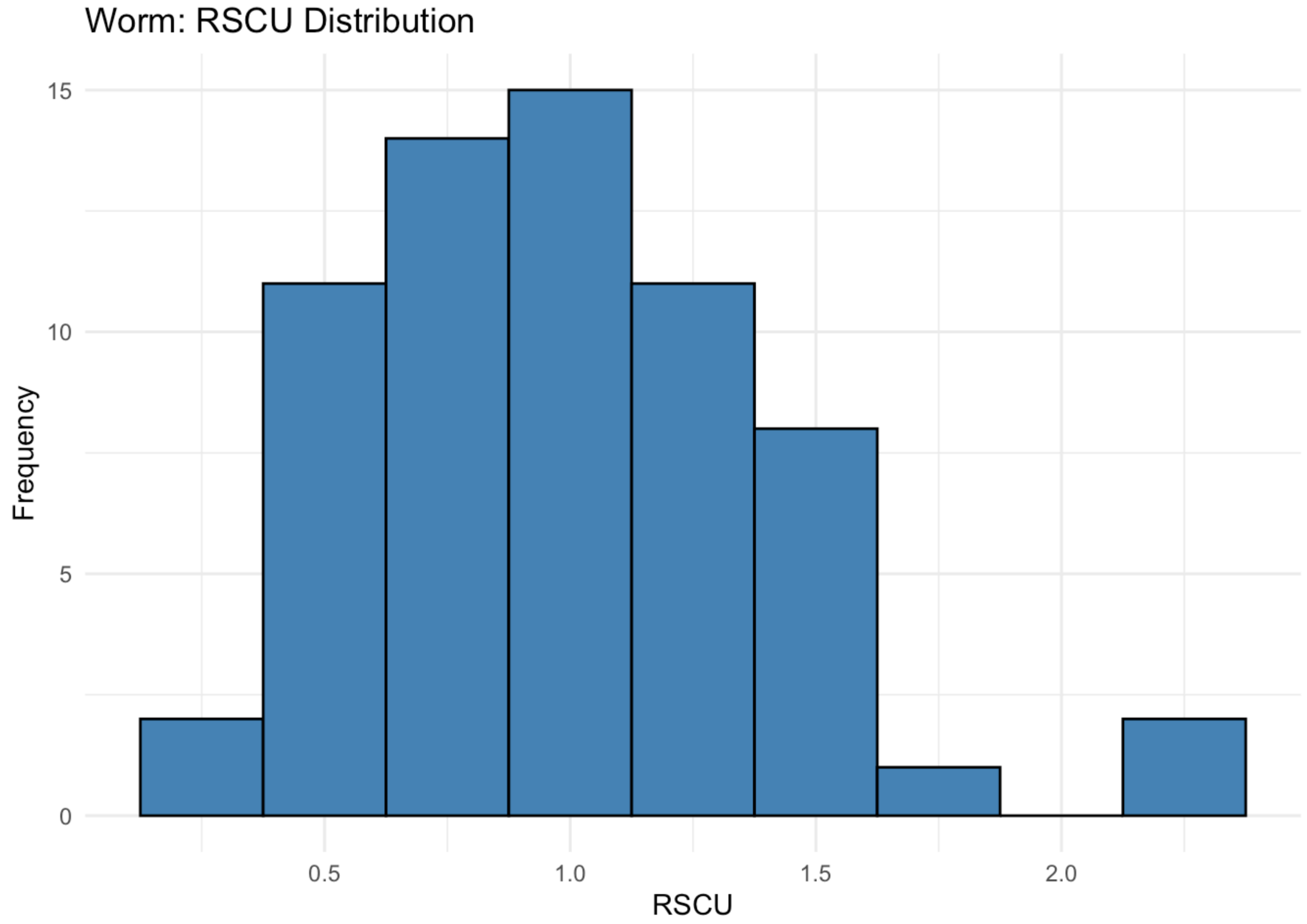
data: rscu_df$RSCU A = 0.62371, p-value = 0.09993

```
[1] “RSCU Distribution is normal”
```



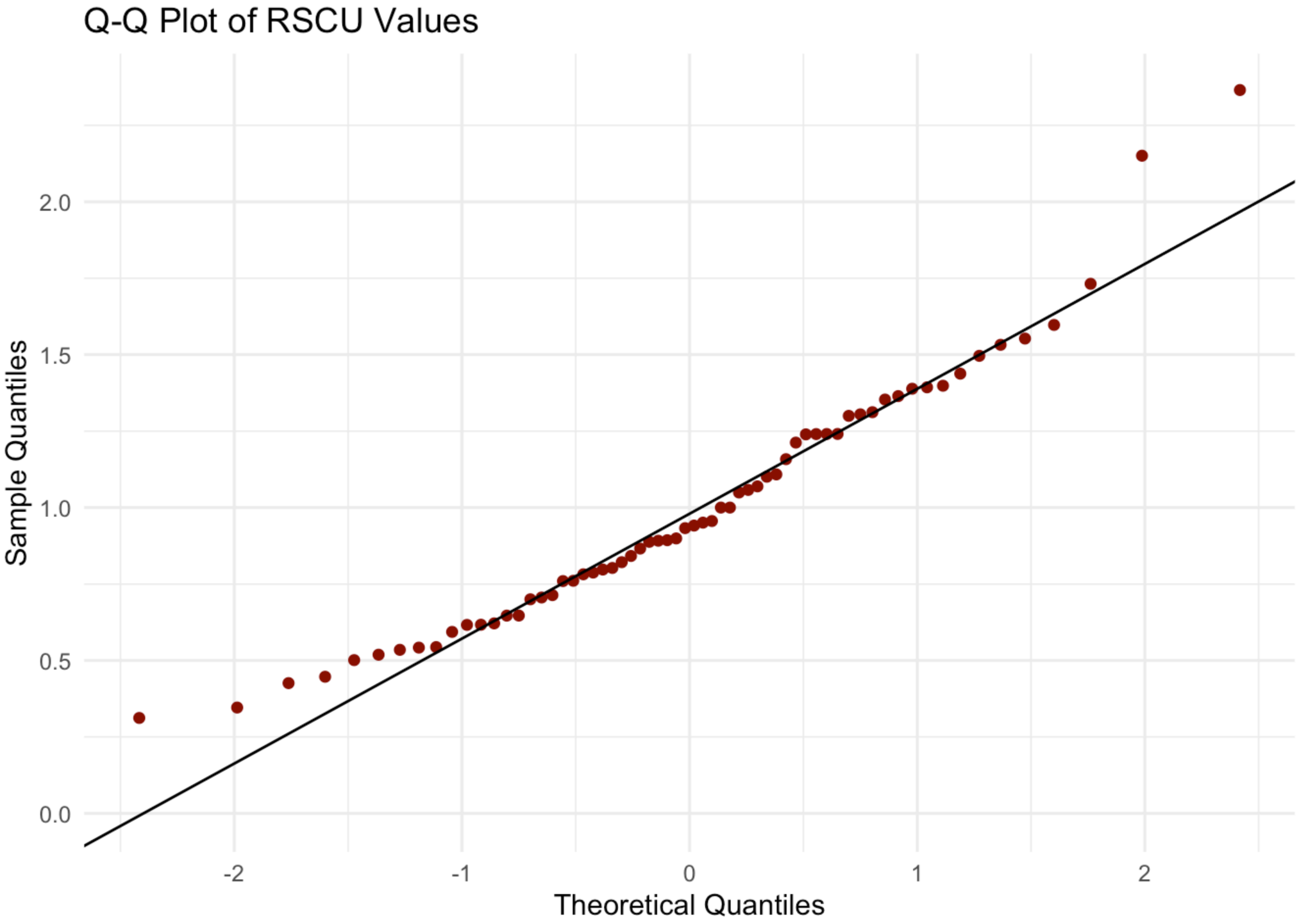


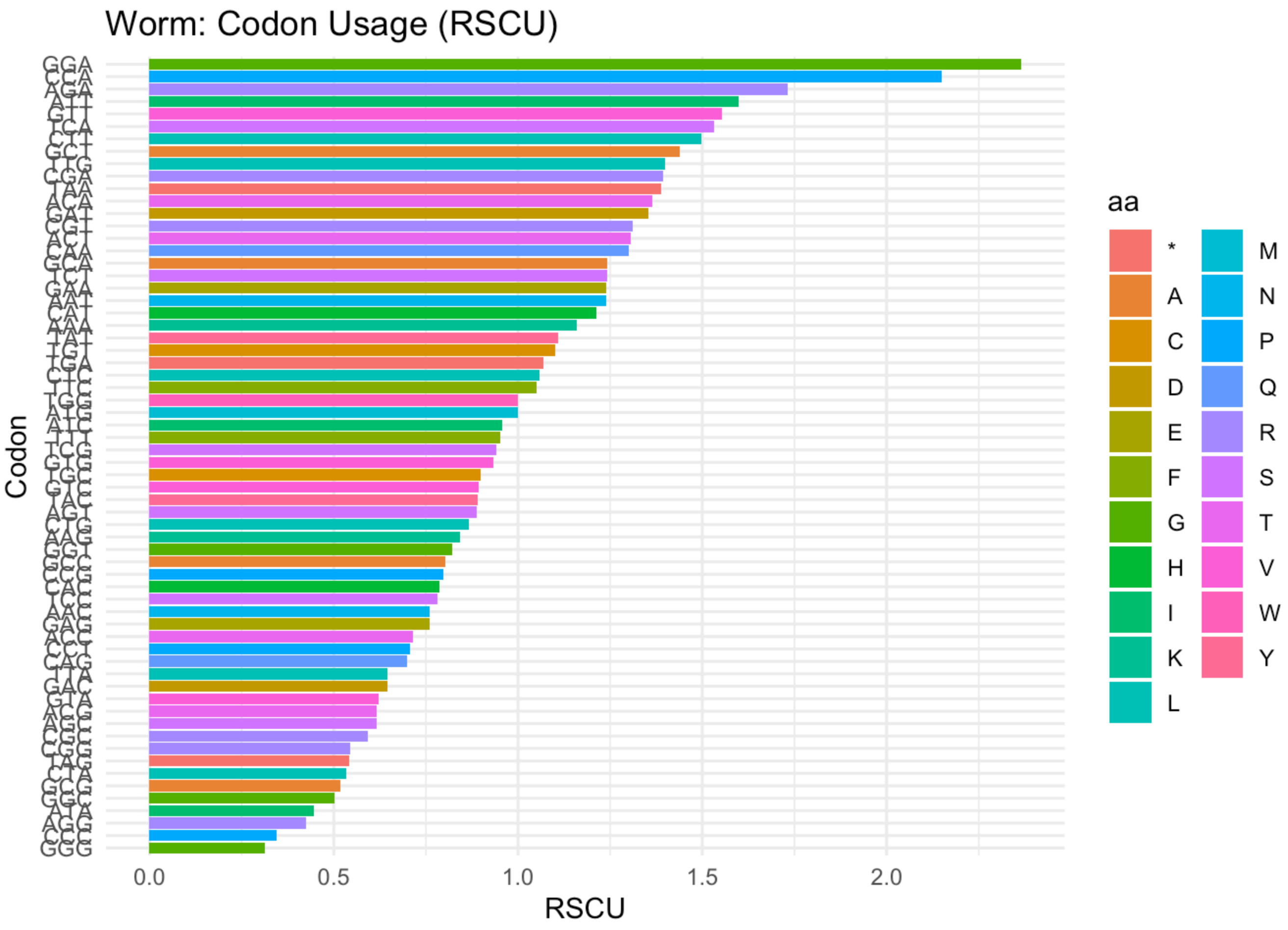
Processing genome: Worm



```
Anderson-Darling normality test

data: rscu_df$RSCU A = 0.59649, p-value = 0.1169
```

```
[1] “RSCU Distribution is normal”
```





6 Comparison Table

```
comparison_df <- bind_rows(top_bottom_list) %>%
  mutate(
    most_used = paste0(most_used, " (", round(most_used_RSCU, 3), ")"),
    least_used = paste0(least_used, " (", round(least_used_RSCU, 3), ")")
  ) %>%
  select(Genome, aa2, most_used, least_used) %>%
  pivot_longer(cols = c(most_used, least_used),
               names_to = "Usage",
               values_to = "Codon_RSCU") %>%
  unite("Genome_Usage", Genome, Usage, sep = "_") %>%
  pivot_wider(names_from = aa2, values_from = Codon_RSCU)

print(comparison_df)
```

```
## # A tibble: 4 × 22
##   Genome_Usage `*`   A     C     D     E     F     G     H     I     K     L
##   <chr>        <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 Human_most_… TGA … GCC … TGC … GAC … GAG … TTC … GGC … CAC … ATC … AAG … CTG …
## 2 Human_least… TAG … GCG … TGT … GAT … GAA … TTT … GGT … CAT … ATA … AAA … CTA …
## 3 Worm_most_u… TAA … GCT … TGT … GAT … GAA … TTC … GGA … CAT … ATT … AAA … CTT …
## 4 Worm_least_… TAG … GCG … TGC … GAC … GAG … TTT … GGG … CAC … ATA … AAG … CTA …
## # ℹ 10 more variables: M <chr>, N <chr>, P <chr>, Q <chr>, R <chr>, S <chr>,
## #   T <chr>, V <chr>, W <chr>, Y <chr>
```

```
write.csv(comparison_df, "codon_RSCU_comparison.csv", row.names = FALSE)
cat("\nSaved: codon_RSCU_comparison.csv\n")
```

```
##
## Saved: codon_RSCU_comparison.csv
```