Basic Evaluations

The input file “**NCBI_df**” is an example file, with the structure as shown:

seqid	taxid	kingdom	phylum	class	order	family	genus	species	sequence
LC795399	1316011	Eukaryota	Chordata	Actinopteri	Acanthuriformes	Acanthuridae	Acanthurus	Acanthurus auranticavus	AGTAGGAACGGCCTTAAGCCTCCTAATCCGAGCAGAAT...
PP735710	58324	Eukaryota	Chordata	Actinopteri	Cypriniformes	Leuciscidae	Phoxinus	Phoxinus phoxinus	CCTTTATCTTGTATTTGGTGCCTGAGCCGGAATGGTAG...
PP735709	58324	Eukaryota	Chordata	Actinopteri	Cypriniformes	Leuciscidae	Phoxinus	Phoxinus phoxinus	CCTTTATCTTGTATTTGGTGCCTGAGCCGGAATGGTAG...
PP735690	58324	Eukaryota	Chordata	Actinopteri	Cypriniformes	Leuciscidae	Phoxinus	Phoxinus phoxinus	CCTTTATCTTGTATTTGGTGCCTGAGCCGGAATGGTAG...
PP735687	58324	Eukaryota	Chordata	Actinopteri	Cypriniformes	Leuciscidae	Phoxinus	Phoxinus phoxinus	CCTTTATCTTGTATTTGGTGCCTGAGCCGGAATGGTAG...
PP735670	58324	Eukaryota	Chordata	Actinopteri	Cypriniformes	Leuciscidae	Phoxinus	Phoxinus phoxinus	CCTTTATCTTGTATTTGGTGCCTGAGCCGGAATGGTAG...

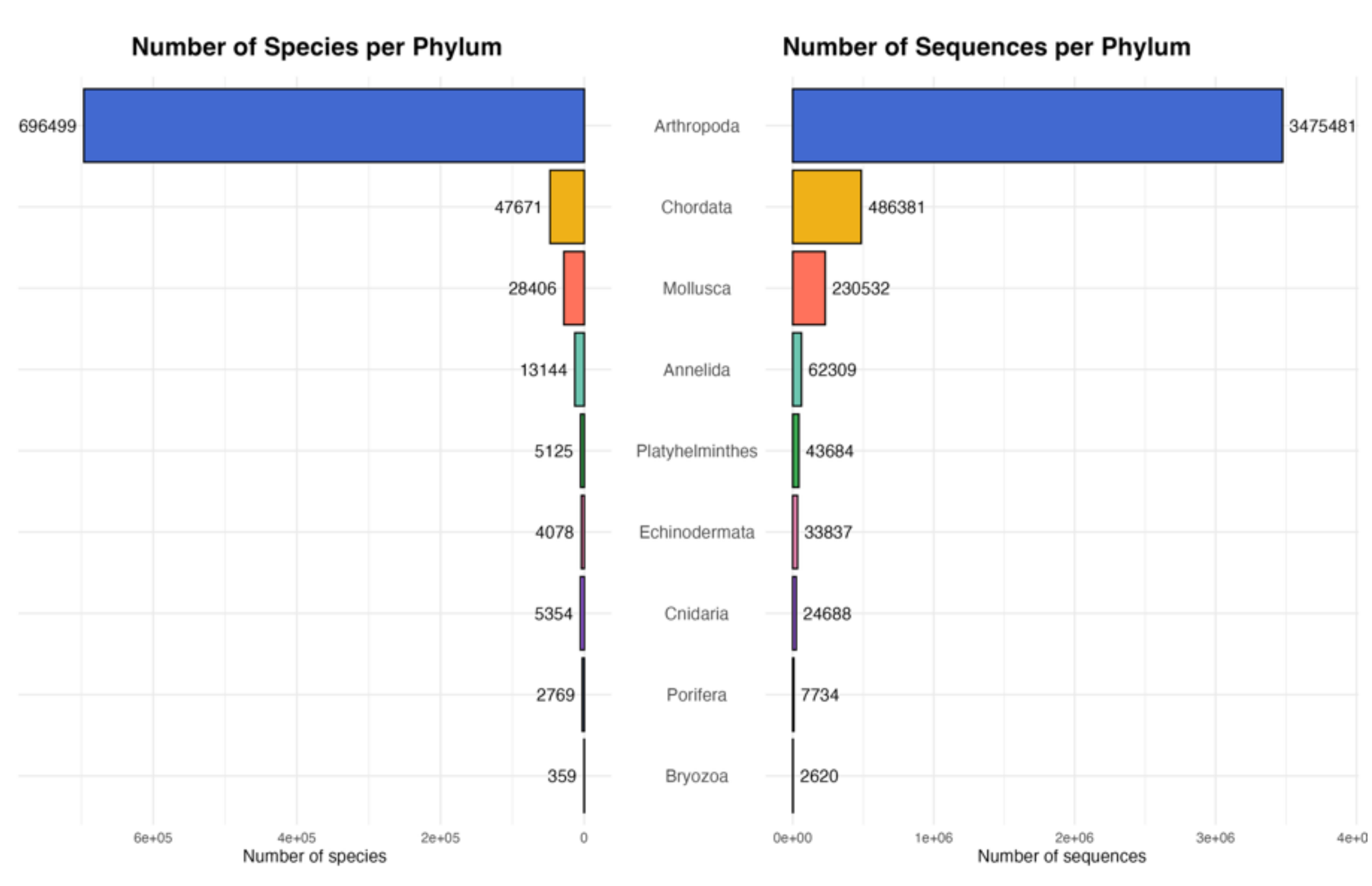
Number of sequence records and species per phylum

```
taxonomy_counts <- NCBI_df %>%
    group_by(phylum) %>%
    summarise(Frequency = n(),
              Species_Count = n_distinct(Species))
```

```
sequence_count <- ggplot(taxonomy_counts, aes(y = phylum, x = Frequency, fill=phylum)) +
    geom_bar(stat = "identity", color="black") +
    labs(y = "", x = "Number of sequences") +
    theme_minimal()+
    theme(legend.position = "none",
          plot.background = element_rect(fill = "white", color = NA),
          axis.text.y =element_text(hjust = 1,size = 11),
          plot.margin = unit(c(1, 0, 0, 0), "cm"),
          plot.title = element_text(hjust = 0.5, face = "bold", size = 16, margin = margin(t = 10, b = 10)),
          plot.title.position = "plot") +
    geom_text(data =taxonomy_counts, aes(y = phylum, x = Frequency+2000, hjust=0, label = Frequency), color = "black",size=3) +
    xlim(0, max(taxonomy_counts$Frequency) * 1.1)
```

```
species_count <- ggplot(taxonomy_counts, aes(y = phylum, x = -Species_Count, fill=phylum)) +
    geom_bar(stat = "identity", color="black") +
    labs(y = "Phylum", x = "Number of species",title = "Number of Species") +
    theme_minimal() +
    theme(legend.position = "none",
          plot.background = element_rect(fill = "white", color = NA),
          axis.text.y = element_blank(),
          axis.title.y = element_blank(),
          plot.margin = unit(c(1, -0.5, 0, 0), "cm"),
          plot.title = element_text(hjust = 0.5, face = "bold", size = 16, margin = margin(t = 10, b = 10)),
          plot.title.position = "plot") +
    geom_text(data =taxonomy_counts, aes(y = phylum, x = -Species_Count-100,hjust=1, label = Species_Count), color = "black",size=3) +
    scale_y_discrete(position = "right")+
    scale_x_continuous(labels = abs, limits = c(-7000,0))
```

```
seq_sp_number <- species_count + sequence_count + plot_layout(ncol = 2)
```



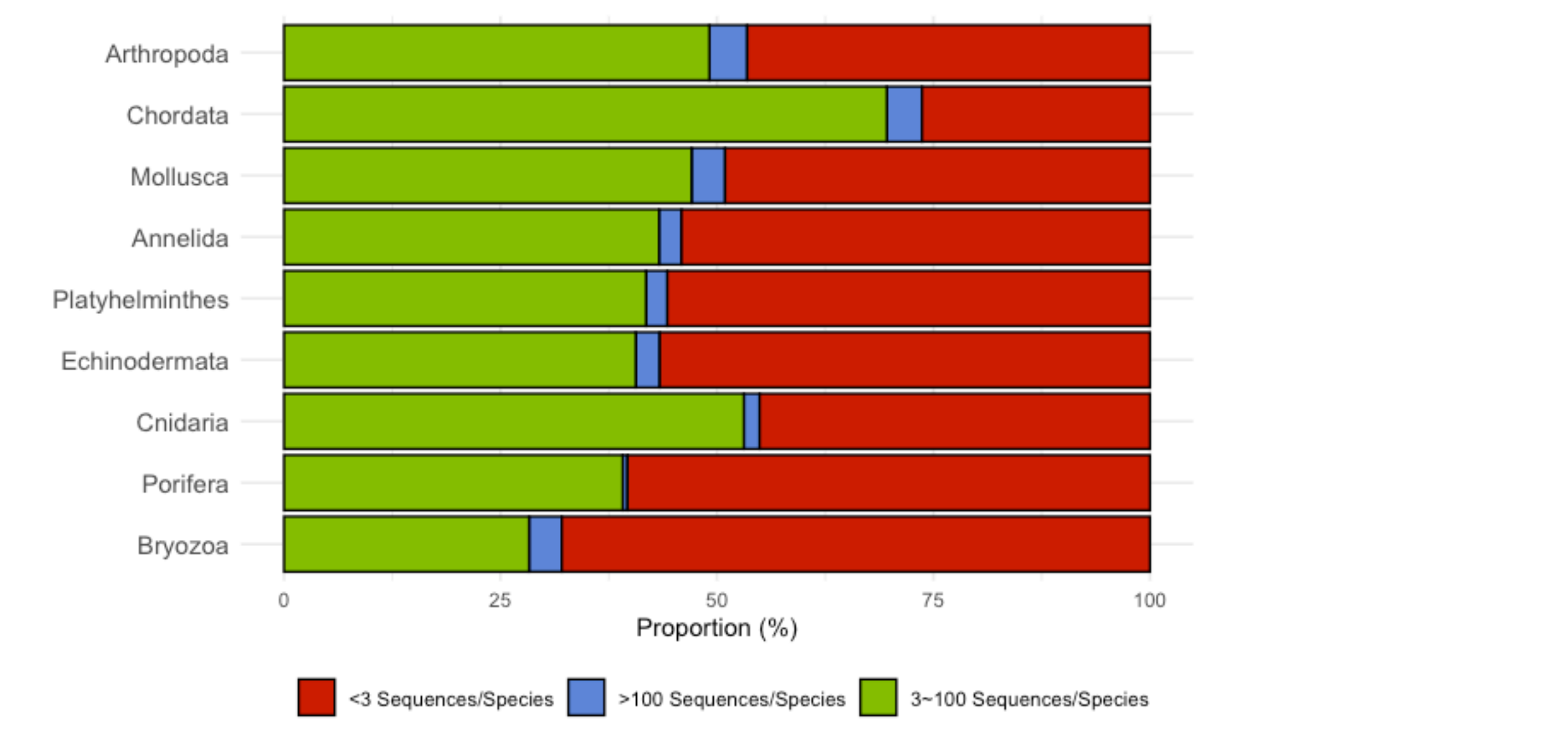
Species representation check

```
species_representation <- NCBI_df %>%
    group_by(phylum, species) %>%
    summarize(Sequence_Count = n()) %>%
    ungroup()
```

```
species_representation_summary <- species_representation %>%
    group_by(phylum) %>%
    summarize(
        total_species = n(),
        over_representative = sum(Sequence_Count > 100),
        under_representative = sum(Sequence_Count < 3),
        normal_representative= sum(Sequence_Count >= 3 & Sequence_Count <= 100),
        proportion_over_representative = over_representative / total_species * 100,
        proportion_under_representative = under_representative / total_species * 100,
        proportion_normal_representative = normal_representative / total_species * 100
    )
```

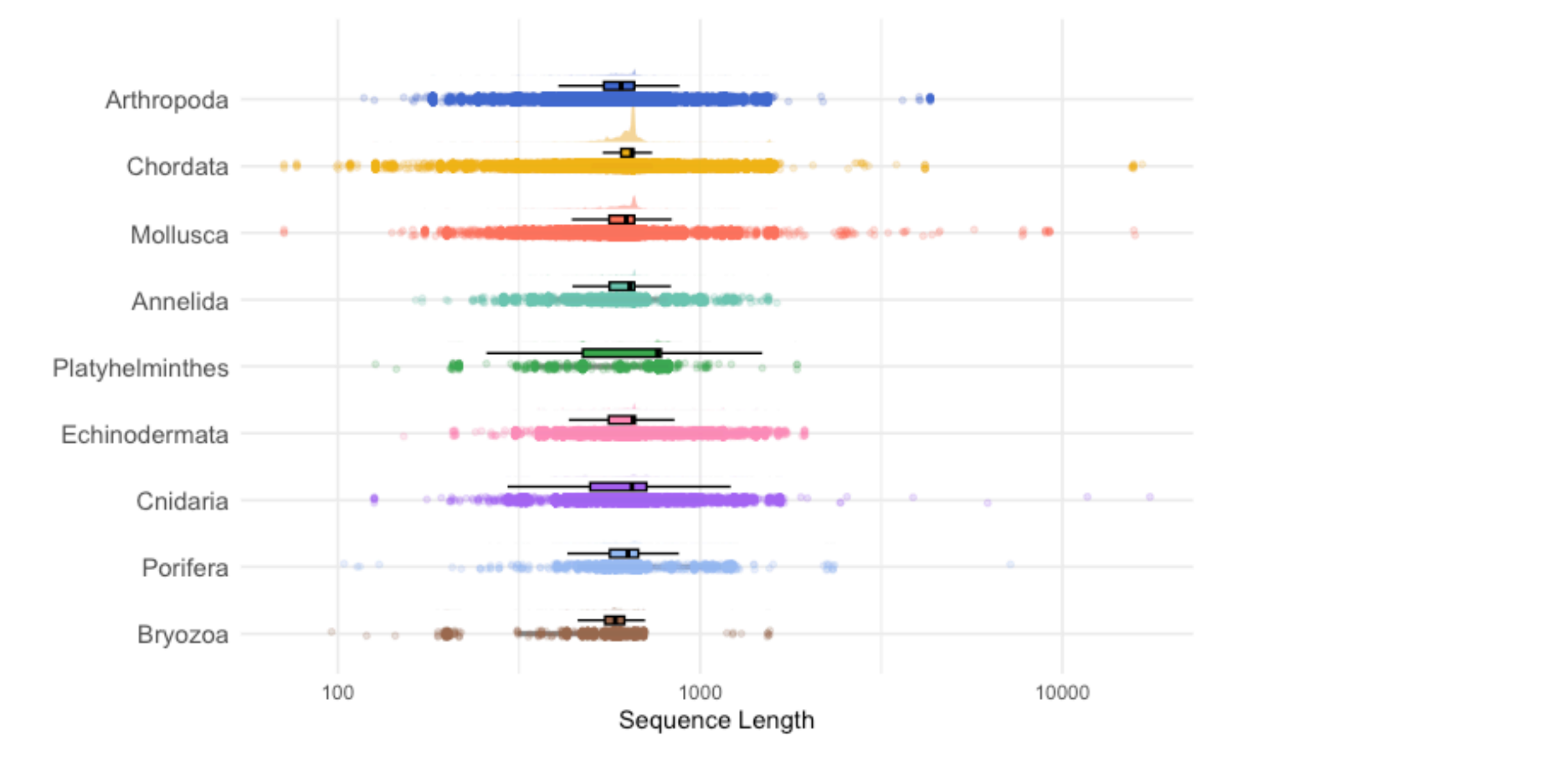
```
representation_long <- species_representation_summary %>%
    pivot_longer(cols = c(proportion_over_representative, proportion_under_representative, proportion_normal_representative),
                 names_to = "Representation_Type",
                 values_to = "Proportion")
```

```
representation <- ggplot(representation_long, aes(x = phylum, y = Proportion, fill = Representation_Type)) +
    geom_bar(stat = "identity",color="black") +
    labs(x = "",
         y = "Proportion (%)",
         fill="") +
    theme_minimal() +
    scale_fill_manual(values=c("#CC0C00FF","#5C88DAFF","#84BD00FF"),labels = c("proportion_under_representative" = "<3 Sequences/Species", "proportion_over_representative" = ">100 Sequences/Species", "proportion_normal_representative" = "3~100 Sequences/Species"))+
    coord_flip() +
    theme(legend.position = "bottom",plot.background = element_rect(fill = "white", color = NA),axis.text.y =element_text(hjust = 1,size = 11))
```



Length of sequences

```
length_dis <- ggplot(NCBI_df, aes(x = phylum, y = Length, fill = phylum)) +
    stat_halfeye(
        adjust = 0.5,
        width = 0.8,
        .width = 0.9,
        justification = -0.5,
        point_color = NA,
        alpha = 0.5
    ) +
    geom_boxplot(
        width = 0.12,
        outlier.shape = NA,
        color = "black",
        position = position_nudge(x = 0.20)
    ) +
     geom_jitter(
        aes(color = phylum),
        width = 0.05,
        alpha = 0.2,
        size = 1
    ) +
    labs(x = "",
         y = "Sequence Length") +
    theme_minimal() +
  coord_flip() +
  theme(legend.position = "none",plot.background = element_rect(fill = "white", color = NA),axis.text.y =element_text(hjust = 1,size = 11)) +
  scale_y_log10()
```



Ambiguous nucleotides

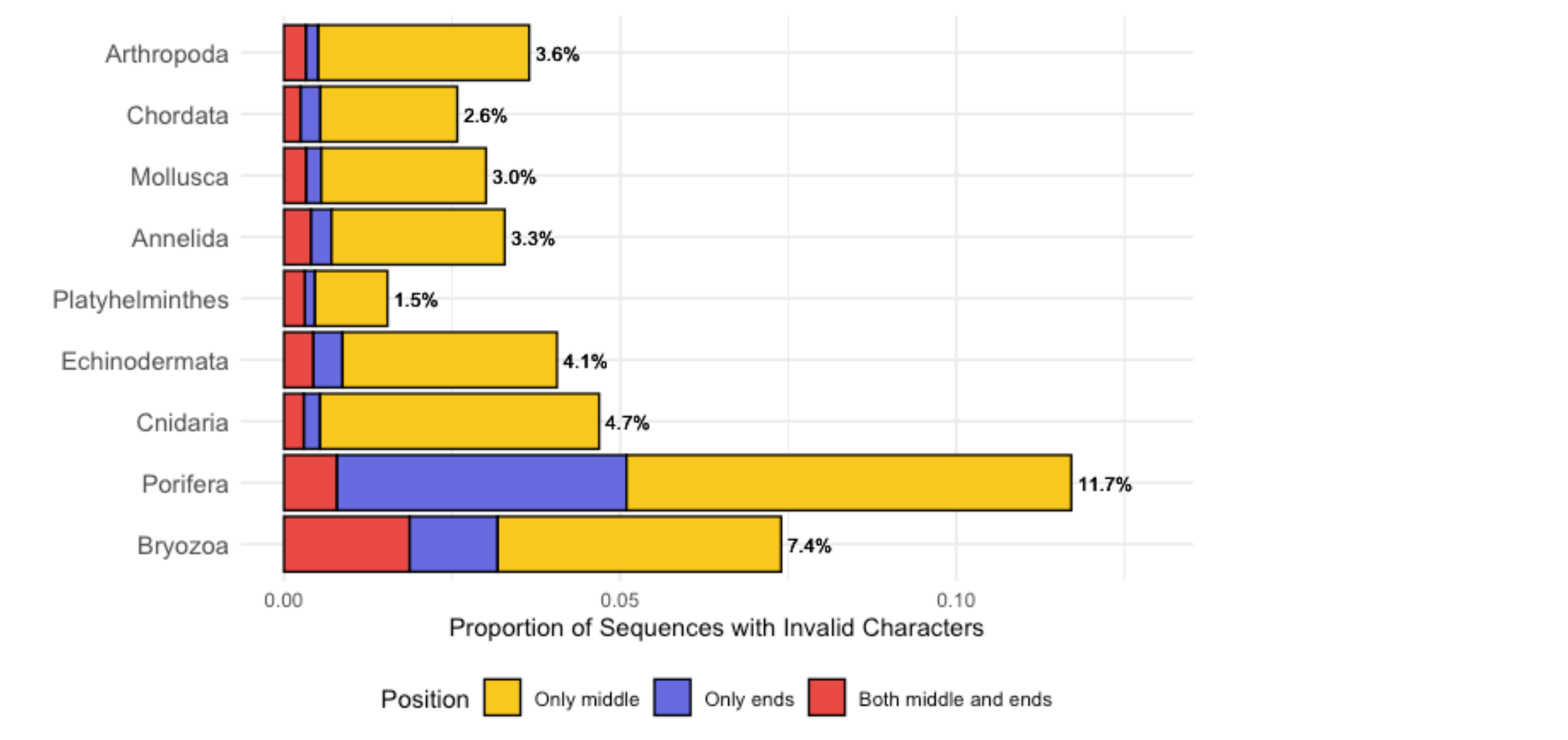
```
categorize_invalid_positions <- function(sequence) {
    five_prime <- substr(sequence, 1, 10)
    three_prime <- substr(sequence, nchar(sequence) - 9, nchar(sequence))
    middle <- substr(sequence, 11, nchar(sequence) - 10)

    invalid_in_five_prime <- any(grepl("[^ATCG]", unlist(strsplit(five_prime, "")), ignore.case = TRUE))
    invalid_in_three_prime <- any(grepl("[^ATCG]", unlist(strsplit(three_prime, "")), ignore.case = TRUE))
    invalid_in_middle <- any(grepl("[^ATCG]", unlist(strsplit(middle, "")), ignore.case = TRUE))

    if ((invalid_in_five_prime | invalid_in_three_prime) & invalid_in_middle) {
        return("Ends and Middle")
    } else if (invalid_in_middle & !invalid_in_five_prime & !invalid_in_three_prime) {
        return("Only Middle")
    } else if (invalid_in_five_prime | invalid_in_three_prime) {
        return("Ends")
    } else {
        return("No invalid characters")
    }
}
```

```
NCBI_df$InvalidCategory <- sapply(NCBI_df$sequence,categorize_invalid_positions)
```

```
N_number_sum <- NCBI_df %>%
    group_by(phylum) %>%
    summarise(
        Total_Sequences = n(),
        Sequences_With_N = sum(!InvalidCategory == "No invalid characters"),
        Sequences_With_N_ends = sum(InvalidCategory == "Ends"),
        Sequences_With_N_both = sum(InvalidCategory == "Ends and Middle"),
        Sequences_With_N_middle = sum(InvalidCategory == "Only Middle")
    )
```

```
N_long <- N_number_sum %>%
  pivot_longer(cols = c(Sequences_With_N_ends, Sequences_With_N_both, Sequences_With_N_middle),
               names_to = "N_position",
               values_to = "count") %>%
  mutate(proportion = count / Total_Sequences)
```

```
N_with_position <- ggplot(N_long, aes(x = proportion, y = phylum, fill = N_position)) +
  geom_bar(stat = "identity", position = "stack",color="black") +
  labs(x = "Proportion of Sequences with Invalid Characters",
       y = "",
       fill = "Position") +
  theme_minimal() +
  scale_fill_manual(values=c("#f9ca24","#686de0","#eb4d4b"),labels = c("Only middle","Only ends", "Both middle and ends")) +
  theme(legend.position = "bottom",plot.background = element_rect(fill = "white", color = NA),axis.text.y =element_text(hjust = 1,size = 11)) +
  geom_text(data =N_long, aes(y = phylum, x = Proportion_With_N+0.001,hjust=0, label = scales::percent(Proportion_With_N, accuracy = 0.1)), color = "black", size=3)+
  xlim(0, max(N_number_sum$Proportion_With_N) * 1.1)
```



Missing taxonomic information

```
missing_tax <- NCBI_df %>%
    group_by(phylum) %>%
    summarise(
        Total_Sequences = n(),
        Missing_Class = sum(class=="nan"|class=="")/Total_Sequences,
        Missing_Order = sum(order=="nan"|order=="")/Total_Sequences,
        Missing_Family = sum(family=="nan"|family=="")/Total_Sequences,
        Missing_Genus = sum(genus=="nan"|genus=="")/Total_Sequences,
        Missing_Species = sum(species=="nan"|species=="")/Total_Sequences
    )
```

```
missing_tax_long <- missing_tax %>%
    pivot_longer(cols = starts_with("Missing_"), names_to = "Rank", values_to = "Missing_Proportion") %>%
    mutate(Rank = gsub("Missing_", "", Rank))
```

```
missing_tax_rank <- ggplot(missing_tax_long, aes(x = Rank, y = phylum, fill = Missing_Proportion)) +
    geom_tile(color = "black")+
    scale_fill_gradient(low = "lightyellow", high = "violetred", name = "Proportion", labels = scales::percent)+
    labs(x = "Taxonomic Rank", y = "")+
    theme_minimal()+
    geom_text(aes(label = ifelse(Missing_Proportion > 0.001, scales::percent(Missing_Proportion, accuracy = 0.1), NA)),size = 3, color = "black")+
    theme(axis.text.y =element_text(hjust = 1,size = 11),legend.position = "bottom",legend.title = element_text(vjust = 1))+
    guides(fill = guide_colourbar(barwidth = 20, barheight = 1))
```