The Influence of Geography on Diversification in the Genus *Panthera*

Mirza Ali Murtuza Ahmadi (1095821)

Github: https://github.com/mirzaahmadi/Assignment_5_Binf6210

2024-12-08

Introduction

Determining accurate phylogenies for different taxa is crucial to scientific and evolutionary research, as phylogenetic tools allow us to understand evolutionary history, guide taxonomy and classification, inform conservation biology, and enhance ecological understanding (Munjal et al., 2018). Additionally, phylogenies also play key roles in human-centered applications, such as tracking the origins and spread of viruses, identifying potential drug targets, and assessing the impact of invasive species on ecosystems, providing a framework for advancing medicine, public health, and agriculture (Munjal et al., 2018). With more sophisticated and collaborative ways of storing data online, phylogenetic data can be integrated with other datasets to explore meaningful questions. For instance, with the increasing availability of geographic data, integrating it with phylogenetic analyses offers an opportunity to explore how geography influences evolutionary processes such as speciation, migration, and adaptation (Losos & Glor, 2003).

Despite the significance of phylogenetic data, accurately assessing evolutionary relationships within taxa remains challenging (Philippe et al., 2011). These challenges arise due to factors such as incomplete or inaccurate data, sampling bias, hybridization events, and ambiguity in morphological traits (Philippe et al., 2011). In some cases, even with advancements in computational tools and access to large databases, there is simply a lack of comprehensive studies combining phylogenetic and geographic data to address evolutionary questions (Philippe et al., 2011). This gap limits our understanding of how geographic factors shape diversification in various taxa.

This report, therefore, focuses on integrating geographic, species, and phylogenetic data to study diversification patterns within a phylogenetically debated group: the genus *Panthera* (King & Wallace, 2014). Specifically, this study aims to answer the question: **Do sister species within the genus *Panthera* inhabit the same geographic regions or different geographic regions?** By investigating this question, we aim to understand how geography influences diversification within this group and contribute to broader discussions on the relationship between geography and evolutionary processes.

Description of Dataset

To analyze my question, I integrated datasets from two public databases: The Barcode of Life Data System (BOLD) and the National Center for Biotechnology Information (NCBI) (Ratnasingham & Hebert, 2007; Sayers et al., 2022). Geographic information for the genus

Panthera was sourced from BOLD, which contains extensive taxonomic, specimen-specific, and geographic data. This dataset encompasses 496 entries across 42 variables, but for the purposes of this study, I focused on the following three variables: species names (with each entry including one of *P*. tigris (tiger), *P*. onca (jaguar), *P*. pardus (leopard), *P*. uncia (snow leopard), and *P*. leo (lion)), their associated countries, as well as their corresponding latitude and longitude coordinates. This data was obtained on December 3, 2024, using the R library 'BOLD', which allows the importing of (*Panthera*) data for analysis. Additionally, COI sequence data for each of the five *Panthera* species was obtained from NCBI. The number of sequence entries varied for each species dataset: jaguar (16 sequences), leopard (21 sequences), lion (12 sequences), snow leopard (5 sequences), and tiger (67 sequences). This data was also obtained on December 3, 2024, via the R library 'rentrez', which provides an interface for accessing and importing data from NCBI databases.

```
Load All Libraries ... (Full code in R Script)
```

Data Acquisition

```
# 1. DATA ACQUISITION ----
# Acquire geographic, species, and sequence data from BOLD and NCBI

#panthera_df <- as_tibble(bold_specimens(taxon='panthera'))
BOLD_data <- read_tsv("../data/panthera_df.tsv")


#The commented-out code below details how COI sequence data was obtained from
NCBI

... (Full code in R Script)

# Loop through each FASTA file, create data frames with standardized naming
conventions for efficient filtering, manipulation, and analysis of COI
sequences
for (file in list.files("../data", full.names=TRUE)) {
  extension <- file_ext(file)
  if(extension != "fasta") { #Since we are only dealing with FASTA files, all
other files can be ignored
    next
  } else {
    #create stringset variables with correct naming conventions to then be
converted to dataframes
    base_name <- file_path_sans_ext(basename(file))
    variable_name_stringset <- paste0(base_name, "_coi_stringset")
    stringset_data <- readDNAStringSet(file)
    assign(variable_name_stringset, stringset_data)

    variable_name_df <- paste0("df_", base_name)
    df_data <- data.frame(COI_title = names(stringset_data), COI_sequence =
paste(stringset_data))
    assign(variable_name_df, df_data)
```

```
    rm(list = variable_name_stringset, stringset_data, df_data) # Remove
unneeded stringset variables from the environment for clarity
  }
}
```

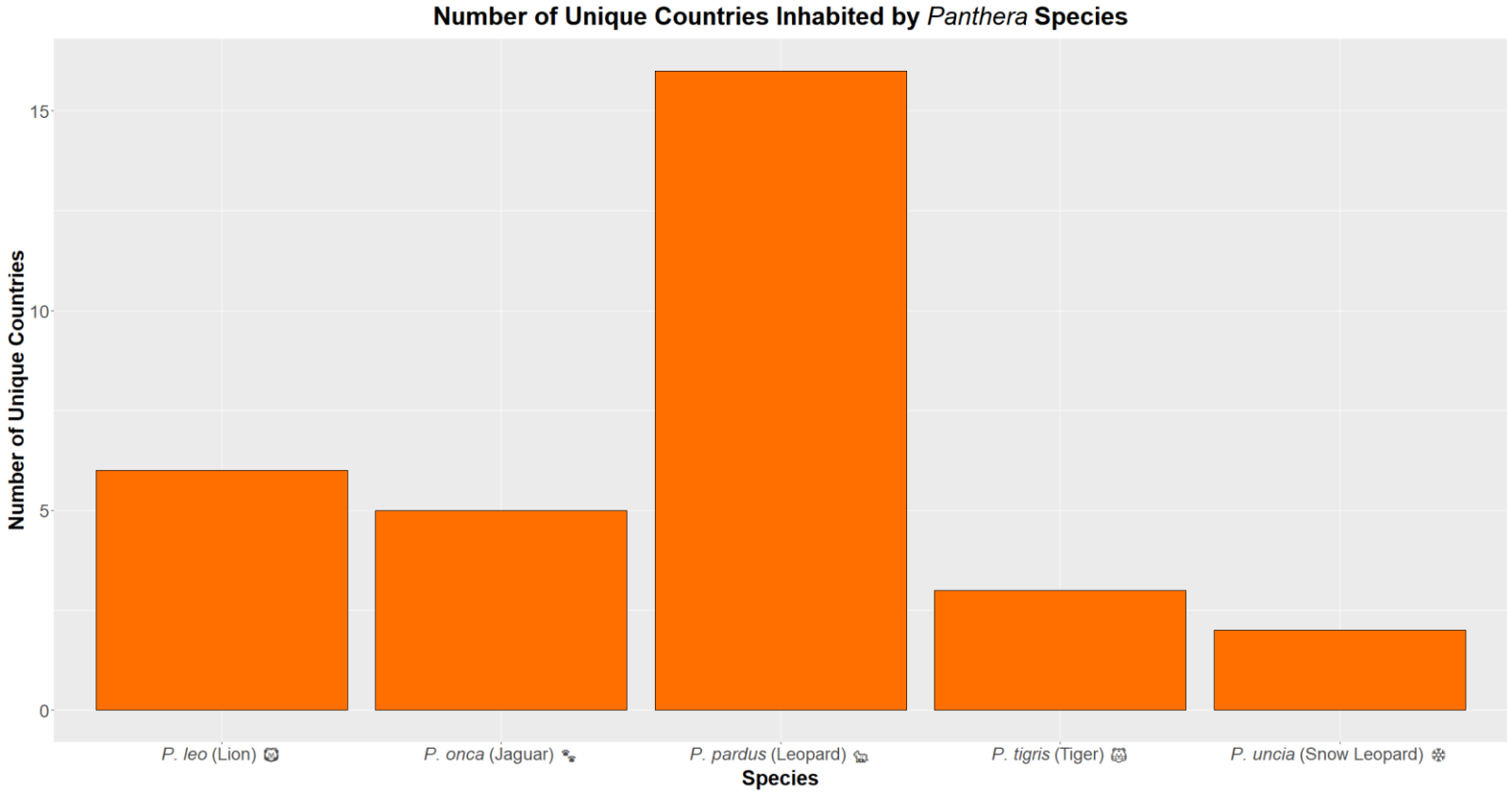
Data Exploration, Filtering, and Quality Control

```
# FILTER and EXPLORE sequence data, geographic data, and sequence data from
BOLD and NCBI

# Create a data frame containing only species names and  corresponding
countries, as these columns are relevant for downstream analysis
panthera_only_countries_df <- BOLD_data %>%
  select(species_name, country)

# The 'skim' function offers a comprehensive data frame summary (i.e. NA
counts, completeness, unique values) - It will be used for exploratory data
analysis frequently
skim(panthera_only_countries_df)

# Filter rows where animals were recorded in countries outside their wild
range, as these are likely non-wild individuals. This loop identifies unique
countries for each species, allowing subsequent exclusion of non-native
locations in the data sets.
for (species in unique(panthera_only_countries_df$species_name)) {
    cat("Species:", species, "\n")
    cat("Countries:",
unique(panthera_only_countries_df$country[panthera_only_countries_df$species_
name == species]), "\n\n")
}

# After researching species' ranges, update data sets to exclude the
following countries that are outside of species' wild range:
pardus_countries_to_exclude <- c("Germany", "Brazil")
tigris_countries_to_exclude <- c("South Africa", "Canada", "Brazil")
leo_countries_to_exclude <- c("Thailand", "Russia", "Canada")

#Creating Data frames for MAIN visualization #1

# update 'panthera_only_countries_df' to exclude non-native countries (and
locations that are not countries at all + NA values)
panthera_only_countries_df <- panthera_only_countries_df %>%
  filter(!(species_name == "Panthera pardus" & country %in%
pardus_countries_to_exclude)) %>%
  filter(!(species_name == "Panthera tigris" & country %in%
tigris_countries_to_exclude)) %>%
  filter(!(species_name == "Panthera leo" & country %in%
leo_countries_to_exclude)) %>%
  filter(country != "Unrecoverable", country != "Exception - Zoological
```

```
Park") %>%
  na.omit()

skim(panthera_only_countries_df)

# This density and subsequent world_map data frames will be be used
downstream for creation of chloropleth map
map_density_df <- panthera_only_countries_df %>%
  group_by(country) %>%
  summarise(unique_species_count = n_distinct(species_name)) %>%
  arrange(country) # Sort the dataframe by species in alphabetical order for
convenient manual checking of dataframe

world_map <- map_data("world")
merged_map_density_df <- world_map %>% # This dataframe will ultimately be
used to map density data
  left_join(map_density_df, by = c("region" = "country")) %>%
  select(-subregion)

# This latitude/longitude data frames will be used downstream for creation of
point map
panthera_LatLon_df <- BOLD_data %>%
  select(species_name, lat, lon, country) %>%
  filter(!(species_name == "Panthera pardus" & country %in%
pardus_countries_to_exclude)) %>%
  filter(!(species_name == "Panthera tigris" & country %in%
tigris_countries_to_exclude)) %>%
  filter(!(species_name == "Panthera leo" & country %in%
leo_countries_to_exclude)) %>%
  filter(country != "Unrecoverable", country != "Exception - Zoological
Park") %>%
  select(species_name, lat, lon, country) %>%
  na.omit()

skim(panthera_LatLon_df)

world_map <- map_data("world")
merged_map_coordinates_df <- world_map %>% #This dataframe will ultimately be
used to map coordinates
  left_join(panthera_LatLon_df, by = c("region" = "country")) %>%
  select(-subregion)

# Creating Data frames for MAIN visualization #3

# For downstream similarity analysis, a similarity matrix is created from
species and country data
# Create a list of unique countries for each species
species_countries <- panthera_only_countries_df %>%
  group_by(species_name) %>%
  summarise(countries = list(unique(country)))
```

```r
# Initialize an empty matrix to store the similarities
species_names <- unique(panthera_only_countries_df$species_name)
similarity_matrix <- matrix(0, nrow = length(species_names), ncol =
length(species_names))
rownames(similarity_matrix) <- species_names
colnames(similarity_matrix) <- species_names

# Calculate the similarity (shared countries) between each pair of species
for (i in 1:length(species_names)) {
  for (j in i:length(species_names)) {
    # Get the countries for species i and species j
    countries_i <- species_countries$countries[species_countries$species_name
== species_names[i]][[1]]
    countries_j <- species_countries$countries[species_countries$species_name
== species_names[j]][[1]]

    # Calculate the intersection (shared countries) between the two species
    shared_countries <- length(intersect(countries_i, countries_j))

    # Find the maximum number of countries for the species pair to normalize
    max_countries <- max(length(countries_i), length(countries_j))

    # Fill the matrix with the normalized shared country count as a decimal
to be used in a heatmap
    similarity_matrix[i, j] <- shared_countries / max_countries
    similarity_matrix[j, i] <- shared_countries / max_countries  # The matrix
is symmetric
  }
}

# Common names for the species - used for naming conventions
common_names <- c("Snow Leopard", "Tiger", "Leopard", "Jaguar", "Lion")

# Create new row and column names by combining scientific names and common
names
new_names <- paste(rownames(similarity_matrix), " (", common_names, ")", sep
= "")

rownames(similarity_matrix) <- new_names
colnames(similarity_matrix) <- new_names

# EXPLORATORY FIG 1: Create a figure to visually display the number of unique
countries each big cat species inhabits within its native range
summary_country_df <- panthera_only_countries_df %>%
  group_by(species_name) %>%
  summarise(number_of_unique_countries = n_distinct(country))

ggplot(summary_country_df, aes(x = species_name, y =
```

```
number_of_unique_countries, fill = species_name)) +
  geom_bar(stat = "identity", fill = "#FF6F00", color = "black") +
  labs(
    title = expression(bold("Number of Unique Countries Inhabited by") ~
bold(italic("Panthera")) ~ bold("Species")),
    x = "Species",
    y = "Number of Unique Countries"
  ) +
  scale_x_discrete(labels = c(
    "Panthera leo" = expression(italic("P. leo") ~ "(Lion) 🦁"),
    "Panthera onca" = expression(italic("P. onca") ~ "(Jaguar) 🐾"),
    "Panthera pardus" = expression(italic("P. pardus") ~ "(Leopard) 🐆"),
    "Panthera tigris" = expression(italic("P. tigris") ~ "(Tiger) 🐯"),
    "Panthera uncia" = expression(italic("P. uncia") ~ "(Snow Leopard) ❄")
  )) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 30, face = "bold"),
    axis.title.x = element_text(size = 25, face = "bold"),
    axis.title.y = element_text(size = 25, face = "bold"),
    axis.text.x = element_text(size = 21),
    axis.text.y = element_text(size = 21),
    legend.position = "none"
  )
```

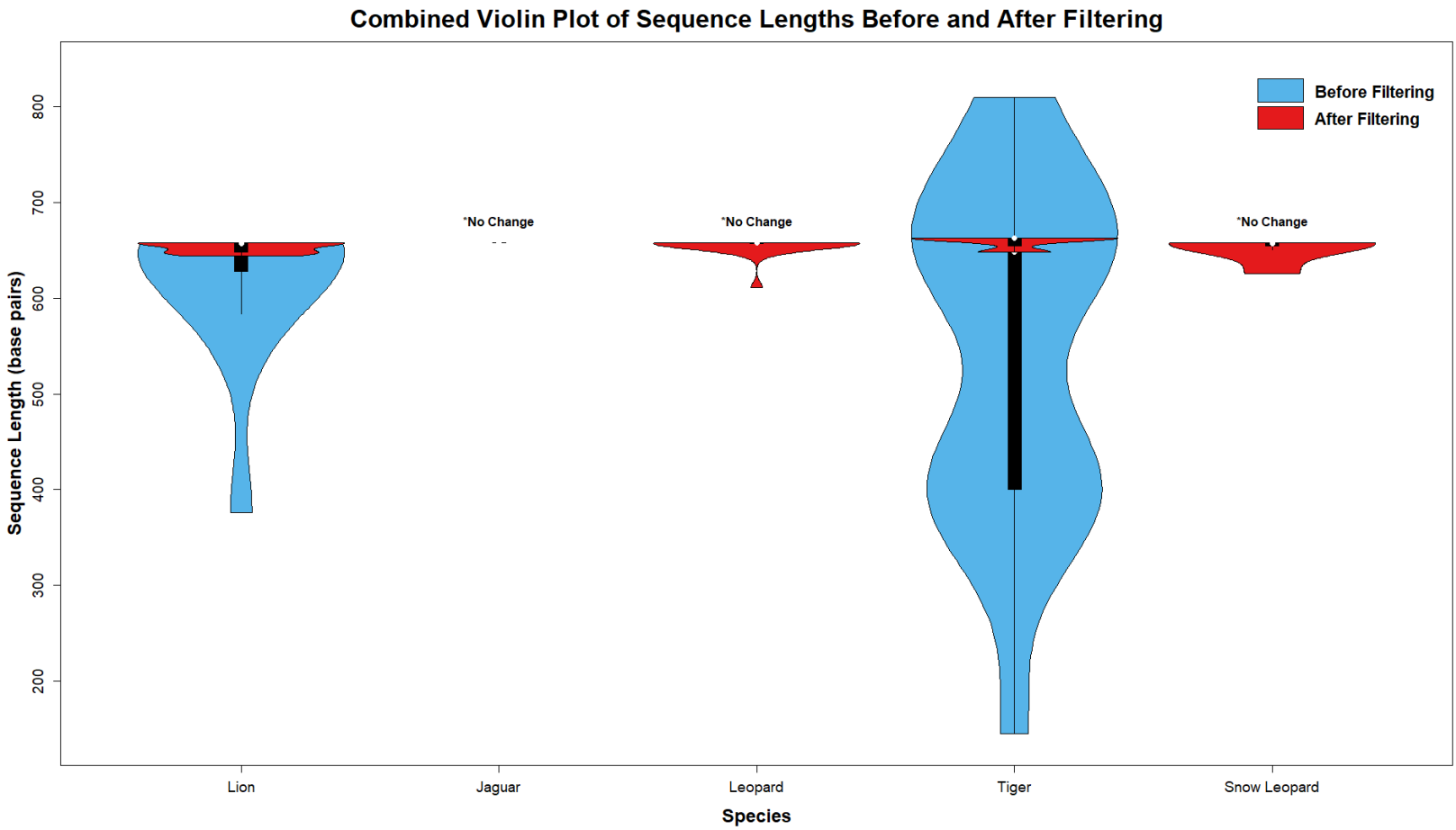


```r
# Filtering Sequence Data
# This function explores all data frame sequences, generating an info table, in order to explore the sequences before and after filtering
present_sequence_info <- function(list_of_dataframes) {
  # Create list of data frame names for generating sequence summaries and establishing naming conventions
  names <- c("Panthera leo (Lion)", "Panthera onca (Jaguar)", "Panthera pardus (Leopard)", "Panthera tigris (Tiger)", "Panthera Uncia (Snow Leopard)")
  # Initialize an empty summary table
  summary_table <- data.frame(Species = character(),
                              Num_Sequences = numeric(),
                              Min_Length = numeric(),
                              Max_Length = numeric(),
                              Mean_Length = numeric(),
                              Median_Length = numeric(),
                              stringsAsFactors = FALSE)
  # Loop through all data frames to fill in the summary table for the sequences
  for (i in seq_along(list_of_dataframes)) {
    df <- list_of_dataframes[[i]]
    sequence_name <- names[i]
    # Create DNAStringSet object from COI_sequence column so that dnastringset summary functions can be used
    sequences <- DNAStringSet(df$COI_sequence)

    # Create a DNAStringSet object from the COI_sequence column to enable the use of DNAStringSet summary functions
    sequence_lengths <- width(sequences)
    num_sequences <- length(sequences)
    min_length <- min(sequence_lengths)
    max_length <- max(sequence_lengths)
    mean_length <- mean(sequence_lengths)
    median_length <- median(sequence_lengths)

    # Append the results to the summary_table to visualize summary information in table format
    summary_table <- rbind(summary_table, data.frame(
      Species = sequence_name,
      Num_Sequences = num_sequences,
      Min_Length = min_length,
      Max_Length = max_length,
      Mean_Length = mean_length,
      Median_Length = median_length
    ))
  }
  print(kable(summary_table))
}
# Define constants for sequence filtering
missing_data <- 0.01 # A missing_data value of 0.01 was chosen because a 1%
```

```r
threshold is commonly used in sequence data processing
length_var <- 50 # A length_var of 50 was chosen as a reasonable range around
the median to account for natural variation in sequence lengths

# Assign names to the dataframes in the following list so we can access each
data frame by name directly in upcoming loop
named_dataframes <- list(df_lion = df_lion, df_jaguar = df_jaguar, df_leopard
= df_leopard,
                         df_tiger = df_tiger, df_snow = df_snow)
# Iterate over the list and apply sequence filtering to each one (created 5
new filtered data frames)
for (df_name in names(named_dataframes)) {
  df <- named_dataframes[[df_name]]  # Get the dataframe from the list

  # Perform filtering operations
  filtered_df <- df %>%
    filter(str_length(COI_sequence) <= 1000) %>% #Filter sequences over 1000
sequences to ensure no genomes are kept in filtered data set
    mutate(COI_sequence = str_remove_all(COI_sequence, "^N+|N+$|-")) %>%
#Remove all
    filter(str_count(COI_sequence, "N") <= (missing_data *
str_count(COI_sequence))) %>% # To ensure high quality data, remove any
sequences with over 1% of ambiguous nucleotides
    filter(str_count(COI_sequence) >= median(str_count(COI_sequence)) -
length_var &
             str_count(COI_sequence) <= median(str_count(COI_sequence)) +
length_var) %>% # Remove sequences that are not close to the median to
homogenise data
    select(COI_title, COI_sequence)

  # Dynamically name the new dataframe and assign it the filtered data
  assign(paste0("filtered_", df_name), filtered_df)

  rm(df, filtered_df)
}
# Compare sequence information before and after filtering for all five
panthera dataframes
dataframes <- list(df_lion, df_jaguar, df_leopard, df_tiger, df_snow)
present_sequence_info(dataframes)

# EXPLORATORY FIG 2: Create a 'vioplot' exploratory figure to showcase
distribution of sequence lengths before and after filtering

#This function counts the nucleotides in each sequence for subsequent violin
plots
count_nucleotides <- function(df) {
  df %>%
    filter(str_length(COI_sequence) <= 1000) %>% # Exclude whole genome
sequences (This is relevant for jaguar dataframe)
```

```r
    mutate(Sequence_length = width(DNAStringSet(as.character(COI_sequence))))
}
# Apply the function to the original datasets (before filtering)
df_lion_w_counts <- count_nucleotides(df_lion)
df_jaguar_w_counts <- count_nucleotides(df_jaguar)
df_leopard_w_counts <- count_nucleotides(df_leopard)
df_tiger_w_counts <- count_nucleotides(df_tiger)
df_snow_w_counts <- count_nucleotides(df_snow)

sequence_lengths_original <- list(
  Lion = df_lion_w_counts$Sequence_length,
  Jaguar = df_jaguar_w_counts$Sequence_length,
  Leopard = df_leopard_w_counts$Sequence_length,
  Tiger = df_tiger_w_counts$Sequence_length,
  Snow_Leopard = df_snow_w_counts$Sequence_length
)
# Apply the function to the filtered datasets
filtered_df_lion_w_counts <- count_nucleotides(filtered_df_lion)
filtered_df_jaguar_w_counts <- count_nucleotides(filtered_df_jaguar)
filtered_df_leopard_w_counts <- count_nucleotides(filtered_df_leopard)
filtered_df_tiger_w_counts <- count_nucleotides(filtered_df_tiger)
filtered_df_snow_w_counts <- count_nucleotides(filtered_df_snow)

sequence_lengths_filtered <- list(
  Lion = filtered_df_lion_w_counts$Sequence_length,
  Jaguar = filtered_df_jaguar_w_counts$Sequence_length,
  Leopard = filtered_df_leopard_w_counts$Sequence_length,
  Tiger = filtered_df_tiger_w_counts$Sequence_length,
  Snow_Leopard = filtered_df_snow_w_counts$Sequence_length
)
# Plot the combined violin plots
par(mfrow = c(1, 1), cex.axis = 1.2, cex.main = 2) #Adjust the plot and axes
titles for optimal size

vioplot(
  sequence_lengths_original$Lion, sequence_lengths_original$Jaguar,
  sequence_lengths_original$Leopard, sequence_lengths_original$Tiger,
  sequence_lengths_original$Snow_Leopard,
  names = c("Lion", "Jaguar", "Leopard", "Tiger", "Snow Leopard"),
  col = "#56B4E9", border = "black", ylim = c(140, 840),
  main = "Combined Violin Plot of Sequence Lengths Before and After
Filtering"
)
# Add the second violin plot to the same axis to showcase comparison
vioplot(
  sequence_lengths_filtered$Lion, sequence_lengths_filtered$Jaguar,
  sequence_lengths_filtered$Leopard, sequence_lengths_filtered$Tiger,
  sequence_lengths_filtered$Snow_Leopard,
  names = c("Lion", "Jaguar", "Leopard", "Tiger", "Snow Leopard"),
```

```r
  col = "#E41A1C", border = "black", add = TRUE
)
text(2, 680, "*No Change", cex = 1, col = "black", font = 2)
text(3, 680, "*No Change", cex = 1, col = "black", font = 2)
text(5, 680, "*No Change", cex = 1, col = "black", font = 2)

# Manually adjust the size and position of the x and y axes to improve
clarity and interpretability of the plot
mtext("Species", side = 1, line = 3, cex = 1.5, font = 2)
mtext("Sequence Length (base pairs)", side = 2, line = 2.5, cex = 1.5, las =
0, font = 2)
# Add a legend to distinguish between the violin visualizations before
filtering vs after (font size and positioning are adjusted as required)
legend(
  x = 4.72, y = 855,
  legend = c("Before Filtering", "After Filtering"),
  fill = c("#56B4E9", "#E41A1C"),
  bty = "n",              # No border around the legend
  cex = 1.37,              # Increase font size
  x.intersp = 0.2,        # Reduce space between text and symbols
  y.intersp = 0.6,
  pt.cex = 1.5,           # Controls the size of the symbols, impacts the box
size
  text.font = 2,
  xpd = TRUE
)
```



Main Software Tools Description

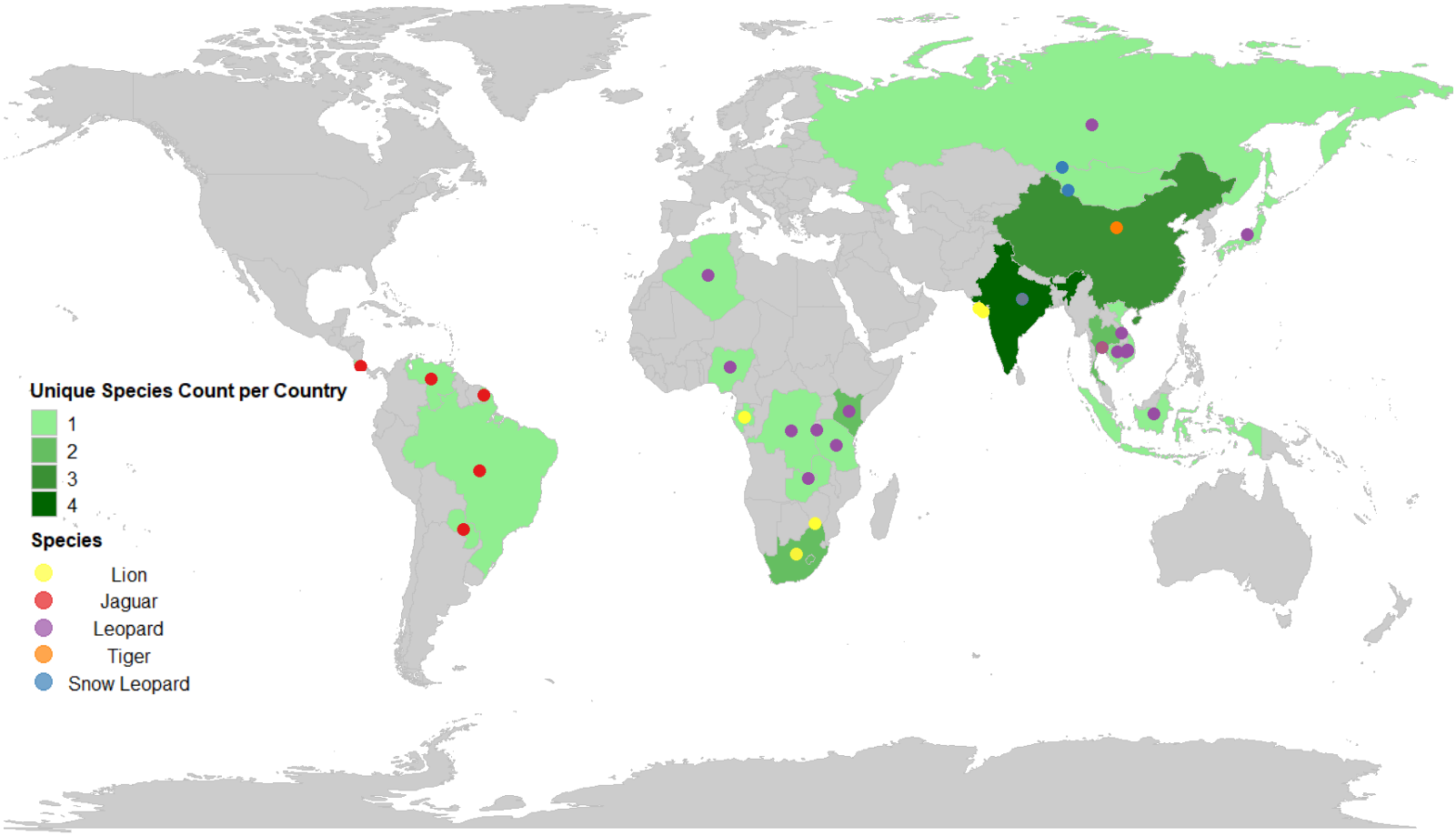
For this assignment, the three primary software tools I used were ‘Biostrings’ (Pagès et al., 2024), ‘MSA’ (Multiple Sequence Alignment) (Bodenhofer et al., 2015), and ‘APE’ (Analysis of Phylogenetics and Evolution) (Paradis & Schliep, 2019). I chose ‘Biostrings’ because it is widely used in R for handling biological sequences, offering strong capabilities for sequence filtering, such as nucleotide counting and removal of ambiguous bases. However, A limitation of this library is its reliance on additional libraries for multi-sequence alignment, which led me to use ‘MSA’ for alignment. ‘MSA’ is commonly used in R for multi-sequence alignment and incorporates various algorithms, such as *ClustalW*, which I applied to align five COI sequences from different species. Lastly, I used ‘APE’ for phylogenetic analysis due to its comprehensive functionality for tree manipulation and plotting, and my prior familiarity with the package. A potential weakness of ‘APE’ and ‘MSA’ is that they are not intuitive to use initially, and troubleshooting errors took some time. Although I considered alternatives like ‘Phangorn’ for phylogenetic analysis, I opted for ‘APE’ due to my previous experience with it.

Main Analysis

```
# Remove temporary dataframes used for exploratory plots to clean up the
environment
rm(df_jaguar_w_counts, df_leopard_w_counts, df_lion_w_counts,
df_snow_w_counts, df_tiger_w_counts, filtered_df_jaguar_w_counts,
filtered_df_leopard_w_counts, filtered_df_lion_w_counts,
filtered_df_snow_w_counts, filtered_df_tiger_w_counts,
sequence_lengths_filtered, sequence_lengths_original, summary_country_df)

# 3. MAIN ANALYSIS ----
# MAIN VISUALIZATION 1: Create a map to visualize the geographic distribution
of Panthera species

# Create a choropleth map showing species density by country and plot
coordinates to visualize range overlap
map <- ggplot() +
  geom_polygon(data = merged_map_density_df, aes(x = long, y = lat, group =
group, fill = unique_species_count), color = "grey") +
  scale_fill_gradient(low = "lightgreen", high = "darkgreen", na.value =
"grey80") +

  # Add species points using 'lat.y' and 'lon' (geom_point) using
merged_map_coordinates_df
  geom_point(data = merged_map_coordinates_df, aes(x = lon, y = lat.y, color
= species_name), size = 5, alpha = 0.7) +

  scale_color_manual(
    values = c(
      "Panthera onca" = "#E41A1C",
      "Panthera pardus" = "#984EA3",
      "Panthera tigris" = "#FF7F00",
```

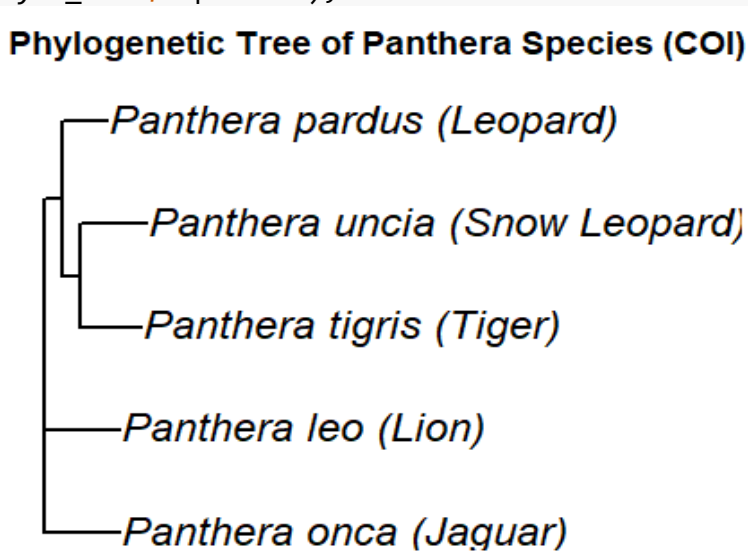
```
      "Panthera leo" = "#FFFF33",
      "Panthera uncia" = "#377EB8"
    ),
    labels = c(
      "Panthera onca" = "Jaguar",
      "Panthera pardus" = "Leopard",
      "Panthera tigris" = "Tiger",
      "Panthera leo" = "Lion",
      "Panthera uncia" = "Snow Leopard"
    )
  ) +

  # Add the legend for the colored species points
  guides(color = guide_legend(title = "Species", title.theme =
element_text(face = "bold")),
         fill = guide_legend(title = "Unique Species Count per Country",
title.theme = element_text(face = "bold"))) +

  labs(
    title = expression("Geographic Distribution and Unique Species Count by
Country for" ~ italic("Panthera") ~ "Genus"),
    fill = "Unique Species Count per Country",
    color = "Species"
  ) +

  theme_void() +
  theme( #The following adjustments are done to improve aesthetics and
clarity of visualization
    plot.title = element_text(hjust = 0.5, size = 22, face = "bold", family =
"Helvetica"),
    legend.position = c(0.17, 0.4),
    legend.title = element_text(size = 14, face = "bold", hjust = 0, margin =
margin(r = 10)),
    legend.text = element_text(size = 12, hjust = 0.5),
    legend.key.width = unit(0.6, "cm"),
    legend.key.height = unit(0.6, "cm"),
    legend.box.margin = margin(10, 10, 10, 10),
    legend.box.spacing = unit(1.5, "cm")
  )
print(map) #NOTE: This might around a minute to properly load
```

Geographic Distribution and Unique Species Count by Country for *Panthera* Genus



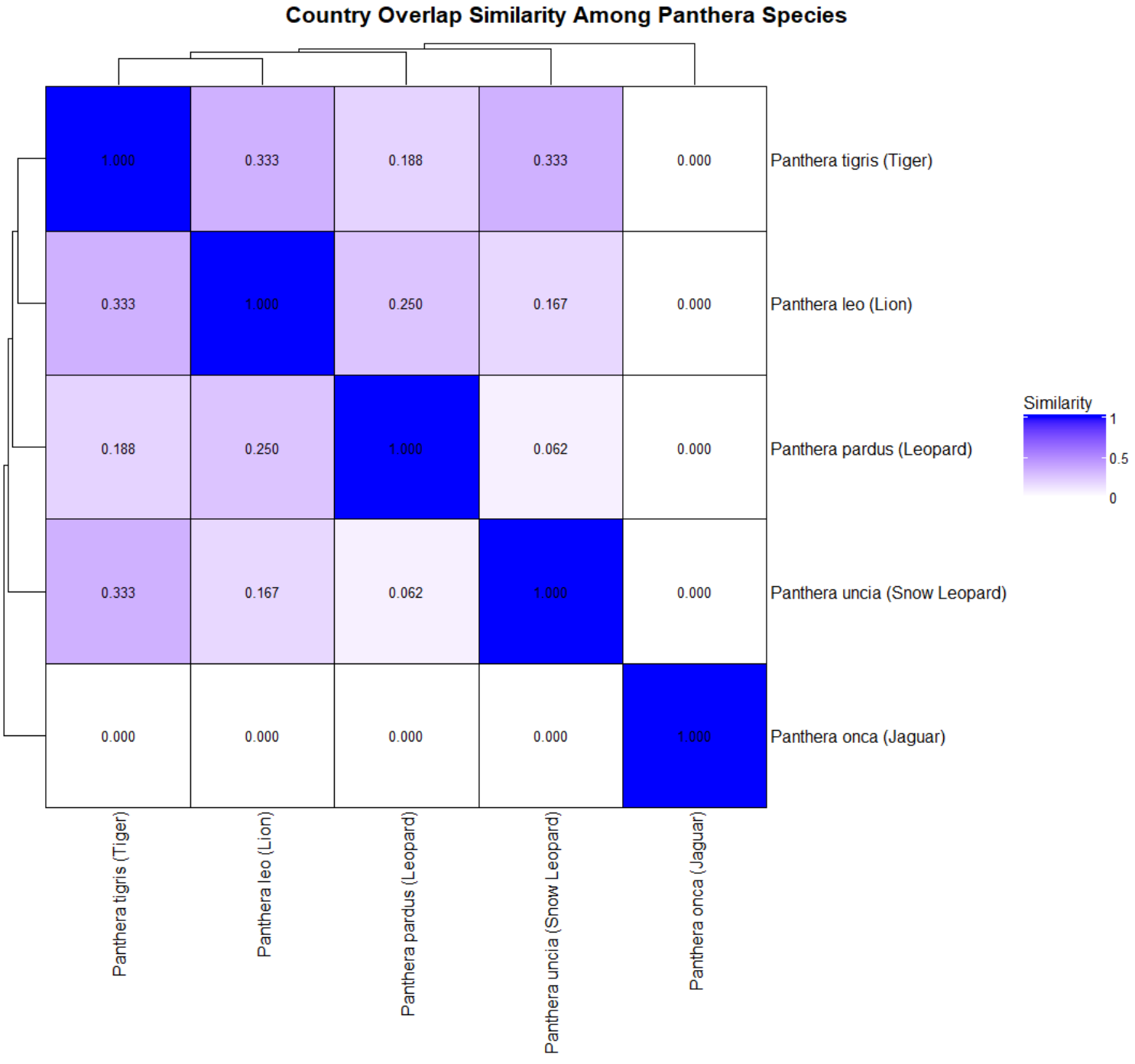

```
MAIN VISUALIZATION 2: Create a phylogeny to showcase evolutionary relatedness
between species in the Panthera genus

#First, create a phylogeny from sequence data

# Initialize empty vector for sequences
sequence_data <- character()
# Add one sequence from each filtered data frame to create the phylogeny.
# Since all sequences have been filtered, any sequence from each data frame
can be chosen.
set.seed(119)
for (filtered_df in filtered_dataframes) {
  random_row <- filtered_df[sample(nrow(filtered_df), 1), ]
  random_row_COI <- random_row$COI_sequence
  sequence_data <- c(sequence_data, random_row_COI)
}
sequence_names <- c("Panthera leo (Lion)",
                    "Panthera onca (Jaguar)",
                    "Panthera pardus (Leopard)",
                    "Panthera tigris (Tiger)",
                    "Panthera uncia (Snow Leopard)")

# Perform multiple sequence alignment using ClustalW method, as this method
is commonly used for multi-sequence alignment
alignment <- msa::msa(sequence_data, method = "ClustalW", type = "dna")
```

```r
aligned_sequences <- msa::msaConvert(alignment, type = "seq") # Convert the
alignment to a character matrix
aligned_sequences <- as.DNAbin(aligned_sequences)
# Create the phylogenetic tree using distance matrix (K80 model) as this
model is very commonly used in models for DNA sequence evolution
dist_matrix <- dist.dna(aligned_sequences, model = "K80")
# Create and plot the phylogenetic tree using the Neighbor Joining method
(this method was chosen because of it's efficiency)
phylo_tree <- ape::nj(dist_matrix)
phylo_tree$tip.label <- sequence_names

par(mar = c(2, 2, 4, 2))  # Adjust margins (bottom, left, top, right)
# Plot the phylogeny
plot(
  phylo_tree,
  main = "Phylogenetic Tree of Panthera Species (COI)",
  tip.label = gsub("_", " ", phylo_tree$tip.label),
  cex = 1.5,
  edge.width = 2,
  label.offset = 0.0005,
  no.margin = FALSE
)
```




```r
# MAIN VISUALIZATION 3:
Create a heatmap using the similarity matrix to showcase how similar
countries are that species live in

heatmap_legend_param <- list(
  title = "Similarity",
  title_gp = gpar(fontsize = 12),
  labels_gp = gpar(fontsize = 10),
  grid_width = unit(20, "mm"),
  grid_height = unit(8, "mm")
)
# customize the heatmap for aesthetics and clarity
heatmap <- Heatmap(similarity_matrix,
                   name = "Similarity",
                   col = colorRamp2(c(0, 1), c("white", "blue")),
                   show_row_names = TRUE,
                   show_column_names = TRUE,
```

```
                    width = unit(7, "in"),
                    height = unit(7, "in"),
                    rect_gp = gpar(col = "black", lwd = 0.5),
                    heatmap_legend_param = heatmap_legend_param,
                    cell_fun = function(j, i, x, y, width, height, fill) {
                      # Add text annotations for each cell
                      grid.text(sprintf("%.3f", similarity_matrix[i, j]), x,
y, gp = gpar(fontsize = 10, col = "black"))
                    })
draw(heatmap, heatmap_legend_side = "right")
grid.text("Country Overlap Similarity Among Panthera Species",
          x = unit(0.5, "npc"),    # Center the title horizontally
          y = unit(0.95, "npc"),   # Position the title slightly below the
top of the plotting area
          gp = gpar(fontsize = 16, fontface = "bold"))
```



Results and Discussion

My report examined the relationship between geography and diversification using the genus *Panthera*, specifically investigating whether sister species inhabit the same or different geographic regions. My analysis found some evidence supporting speciation within the same region, though this was limited. In the phylogenetic analysis, the tiger and snow leopard, the most closely related species, shared the highest number of countries (similarity score of 0.333), suggesting that speciation can occur within the same region. However, the leopard, the sister species to the tiger-snow leopard clade, showed lower geographic overlap with either species (0.188 with tiger and 0.062 with snow leopard), indicating that geographic proximity doesn't always correlate with genetic relatedness. Interestingly, the leopard shared a higher similarity score (0.250) with the lion, a more distantly related species, suggesting again that geographic overlap does not always align with genetic relatedness. Although, high similarity scores between distantly related species, like tigers and lions (0.333), suggest that allopatric speciation may play a role in contributing to evolutionary divergence. The jaguar, being somewhat of an outgroup to the rest of the tree, showed no significant geographic overlap with others, reinforcing the idea that its range doesn't follow the same speciation patterns. While the data somewhat supports the role of geographic proximity in speciation, it also highlights the complexity of diversification, making definitive conclusions difficult. Overall, although there is some overlap in range, many of these big cat species occupy distinct geographic ranges, and their roles as apex predators in these environments likely influence their distribution. This ecological niche differentiation may prevent their overlap in certain areas, as the same niche would otherwise be redundant. Such niche differentiation is supported in the literature, where it is often suggested that species within the same functional role tend to avoid direct geographic overlap to reduce competitive pressures and resource overlap (Purcell & Stigall, 2021; Wang et al., 2023).

The results of this study were somewhat expected, given the complexity of drawing definitive conclusions on speciation patterns. We observed evidence supporting both within-region speciation and allopatric divergence, which aligns with the idea that multiple factors influence diversification. However, a significant limitation in this study was the small sample size, which restricts the ability to make definite conclusions. With a larger sample of species and more genetic sequences, this study could be expanded to provide a more comprehensive analysis. Additionally, biases in sample collection were present, as certain *Panthera* species, like tigers, are more widely studied and had a wealth of available sequences. In contrast, elusive species like the snow leopard, which inhabits remote regions, had very limited available data.

Given more time and resources, this research could be expanded significantly. Additional data from more geographic regions and animal samples would provide a more robust dataset for analysis. This would help refine our conclusions and potentially lead to a clearer understanding of the relationship between geography and diversification. Future research could also extend this study to other taxonomic groups, particularly those where there is debate about evolutionary origins, or where recent advancements in genetic sequencing have made new data available. With more complete phylogenies, it may be possible to gain more concrete insights into the factors driving speciation in these groups.

Reflection

This project taught me valuable lessons in data manipulation and integration, especially when working with diverse types of data. Combining sequence data with biological geographic data from different sources was challenging, requiring careful attention to detail to ensure the correct pairing of corresponding elements. A key takeaway from both this project and the course overall was the messiness of real-world data. Unlike curated datasets from previous courses, real-world data often includes missing values and inaccuracies, requiring effective filtering and careful analysis to derive meaningful conclusions. Additionally, I learned the importance of collaboration. Programming has an abundance of solutions, and discussing approaches with others often leads to new insights and technical skills. Collaboration not only improves code but also enhances learning, as multiple perspectives can open doors to more efficient solutions. Moving forward, I aim to improve my time management, especially when dealing with large assignments. In the past, I've struggled with perfectionism, trying to complete each step flawlessly before moving on, but I now realize that building a foundation first and refining later is a more effective approach. This strategy will help me manage my workload more efficiently.

Acknowledgements

I would like to express my gratitude to Thomas Tekle for his continuous support and constructive feedback throughout this project. His thoughtful challenges to my ideas, coding techniques, and analysis choices helped me refine my work. I also appreciate the advice from Isha Baxi, whose insights into improving the aesthetics and interpretability of my visualizations were invaluable.

References

Bodenhofer, U., Bonatesta, E., Horejš-Kainrath, C., & Hochreiter, S. (2015). msa: an R package for multiple sequence alignment. *Bioinformatics (Oxford, England)*, *31*(24), 3997–3999. https://doi.org/10.1093/bioinformatics/btv494

He, Q., Wang, S., Feng, K. *et al.* (2023). High speciation rate of niche specialists in hot springs. *ISME Journal*, 17, 1303–1314. https://doi.org/10.1038/s41396-023-01447-4

King, L. M., & Wallace, S. C. (2014). Phylogenetics of *Panthera*, including *Panthera atrox*, based on craniodental characters. *Historical Biology*, *26*(6), 827–833. https://doi.org/10.1080/08912963.2013.861462

Losos, J. B., & Glor R. E. (2003). Phylogenetic comparative methods and the geography of speciation. *Trends in Ecology and Evolution*, 18(5), 220-227. https://doi.org/10.1016/S0169-5347(03)00037-5

Munjal, G., Hanmandlu, M., & Srivastava, S. (2018). Phylogenetics Algorithms and Applications. *Ambient Communications and Computer Systems: RACCCS-2018*, *904*, 187–194. https://doi.org/10.1007/978-981-13-5934-7_17

Pagès H., Aboyoun P., Gentlemen, R., & DebRoy, S. (2024). Biostrings: Efficient manipulation of biological strings. *R package version 2.74.0*. https://bioconductor.org/packages/Biostrings

Paradis, E., & Schliep, K. (2019). ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R. *Bioinformatics*, 35**,** 526–528. doi:10.1093/bioinformatics/bty633.

Philippe, H., Brinkmann, H., Lavrov, D. V., Littlewood, D. T., Manuel, M., Wörheide, G., & Baurain, D. (2011). Resolving difficult phylogenetic questions: why more sequences are not enough. *PLoS biology*, *9*(3), e1000602. https://doi.org/10.1371/journal.pbio.1000602

Purcell, C. K. Q., & Stigall, A. L. (2021). Ecological niche evolution, speciation, and feedback loops: Investigating factors promoting niche evolution in Ordovician brachiopods of eastern Laurentia. *Paleogeography, Palaeoclimatology, Paleoecology*, 575, 110555. *https://doi.org/10.1016/j.palaeo.2021.110555*

Ratnasingham, S., & Hebert, P. D. (2007). bold: The Barcode of Life Data System (http://www.barcodinglife.org). *Molecular ecology notes*, *7*(3), 355–364. https://doi.org/10.1111/j.1471-8286.2007.01678.x

Sayers, E. W., Bolton, E. E., Brister, J. R., Canese, K., Chan, J., Comeau, D. C., Connor, R., Funk, K., Kelly, C., Kim, S., Madej, T., Marchler-Bauer, A., Lanczycki, C., Lathrop, S., Lu, Z., Thibaud-Nissen, F., Murphy, T., Phan, L., Skripchenko, Y., Tse, T., … Sherry, S. T. (2022). Database resources of the national center for biotechnology information. *Nucleic acids research*, *50*(D1), D20–D26. https://doi.org/10.1093/nar/gkab1112