Assignment 5

Qingyuan Pei (Github)

2023-12-10

Introduction

The evolution of nuclear genes is generally slower than that of mitochondrial genes which can differ their phylogenetic trees(Lin & Danforth, 2004). Because of this, researchers often estimate phylogenetic relationship based on both of these genes for more accurate result. However, research about the discordance between different mitochondrial-gene trees are relatively limited.

COI (Cytochrome Oxidase I) and CytB (Cytochrome b) are both mitochondrial genes that are commonly used for phylogenetic studies, species identification, and evolutionary biology in birds. Whereas, the evolutionary rates between these two genes are different across more than 300 species in avian orders: COI evolves 14% slower than CytB on average (Lavinia et al., 2016). This means choosing different marker mitochondrial genes can also make the estimated phylogenetic trees vary.

To further explore the discordance of COI-gene trees and CytB-gene trees in a specific genera of bird, I use COI and CytB sequences data sets of *Acipenser* which are downloaded from NCBI and apply functions from packages such as "DECIPHER", "ape", and "dendextend" to make a tanglegram for examining topological congruence between the two phylogenetic trees draw from COI and CytB data sets.

Description of Data Set

The data set I choose for this project is the DNA sequences of COI gene and CytB gene in *Acipenser* from NCBI, and this data set is obtained on December 3, 2023. Here I use rentrez package and functions in Entrez_Functions.R, a course material, to obtain the data in this R script. I also limited the sequence lengths of the two genes to exclude whole genome sequences: the range of sequence length is 500-700 for COI gene and is 600-1200 for CytB gene.

Code Section 1 - – Data Acquisition, Exploration, Filtering, and Quality Control

Loading packages in this project.

```
library(tidyverse)
library(rentrez)
library(Biostrings)
library(ggtree)
library(DECIPHER)
library(ape)
library(dendextend)
library(phytools)
```

Search for the hits of COI and CytB gene of *Acipenser*.

```
NCBI_search <- entrez_search(
  db = "nuccore",
  term = "Acipenser[ORGN] AND COI[Gene] AND 500:700[SLEN]",
  use_history = T
  )
NCBI_search
```

```
## Entrez search result with 303 hits (object contains 20 IDs and a web_history object)
##  Search term (as translated):  "Acipenser"[Organism] AND COI[Gene] AND 0000000050 ...
```

```
NCBI_search1 <- entrez_search(
  db = "nuccore",
  term = "Acipenser[ORGN] AND CytB[Gene] AND 600:1200[SLEN]",
  use_history = T
  )
NCBI_search1
```

```
## Entrez search result with 319 hits (object contains 20 IDs and a web_history object)
##  Search term (as translated):  "Acipenser"[Organism] AND CytB[Gene] AND 000000006 ...
```

Using functions in Entrez_Functions.R to obtain data set from NCBI.

```
# Save the functions wrote in the R script to the current environment
source("Entrez_Functions.R")
# Fetch fasta files from NCBI
FetchFastaFiles(searchTerm = "Acipenser[ORGN] AND COI[Gene] AND 500:700[SLEN]",
                seqsPerFile = 100, fastaFileName = "Acipenser_fetch_1.fasta")
FetchFastaFiles(searchTerm = "Acipenser[ORGN] AND CytB[Gene] AND 600:1200[SLEN]",
                seqsPerFile = 100, fastaFileName = "Acipenser_fetch_2.fasta")
# Merge all fasta files and store it as a data frame
dfNCBI_COI <- MergeFastaFiles(filePattern = "Acipenser_fetch_1.fasta.*")
dfNCBI_CytB <- MergeFastaFiles(filePattern = "Acipenser_fetch_2.fasta.*")

# Add two new columns to include the species name and sequence length of each row,
# and filter out whole genome sequences and sequences with more than 5% of Ns.
Filter1 <- function(df) {
  filtered1 <- df %>%
    mutate(Species_Name = word(Title, 2L, 3L)) %>%
    mutate(Sequence_Length = nchar(Sequence)) %>%
    mutate(Gene_Type = word(str_extract(Title, "(?i)partial|complete"))) %>%
    filter(Gene_Type == "partial") %>%
    filter(str_count(Sequence, "N") <= (0.05 * Sequence_Length))
  return(filtered1)
}
dfCOI <- Filter1(dfNCBI_COI)
dfCytB <- Filter1(dfNCBI_CytB)
```

Draw a histogram to see the distribution of COI sequence lengths to further decide the following data filtering steps.

```
hist((dfCOI$Sequence_Length),
     xlab = "Sequence Length",
     ylab =  "Frequency",
     main = "Distribution of COI Sequence Lengths")
```

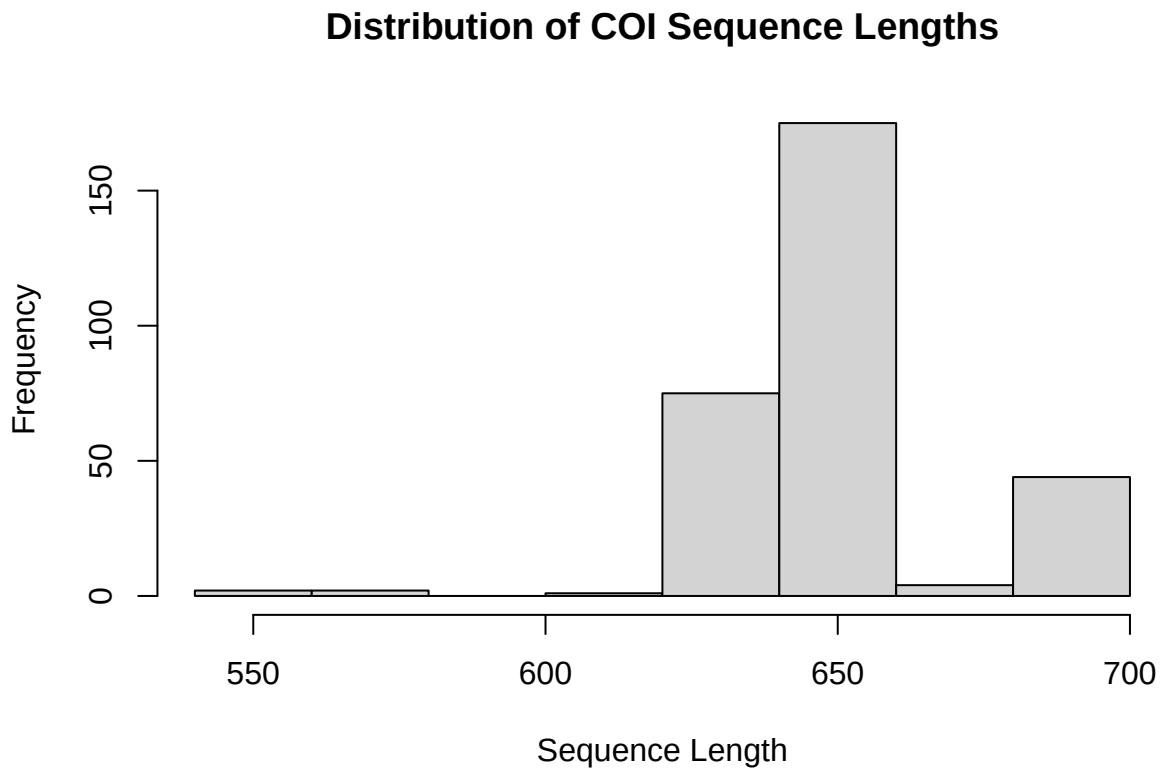

Figure 1: The majority of COI sequences distributes around nucleotides in length.

Draw a histogram to see the distribution of CytB sequence lengths to further decide the following data filtering steps.

```
hist((dfCytB$Sequence_Length),
     xlab = "Sequence Length",
     ylab =  "Frequency",
     main = "Distribution of CytB Sequence Lengths")
```

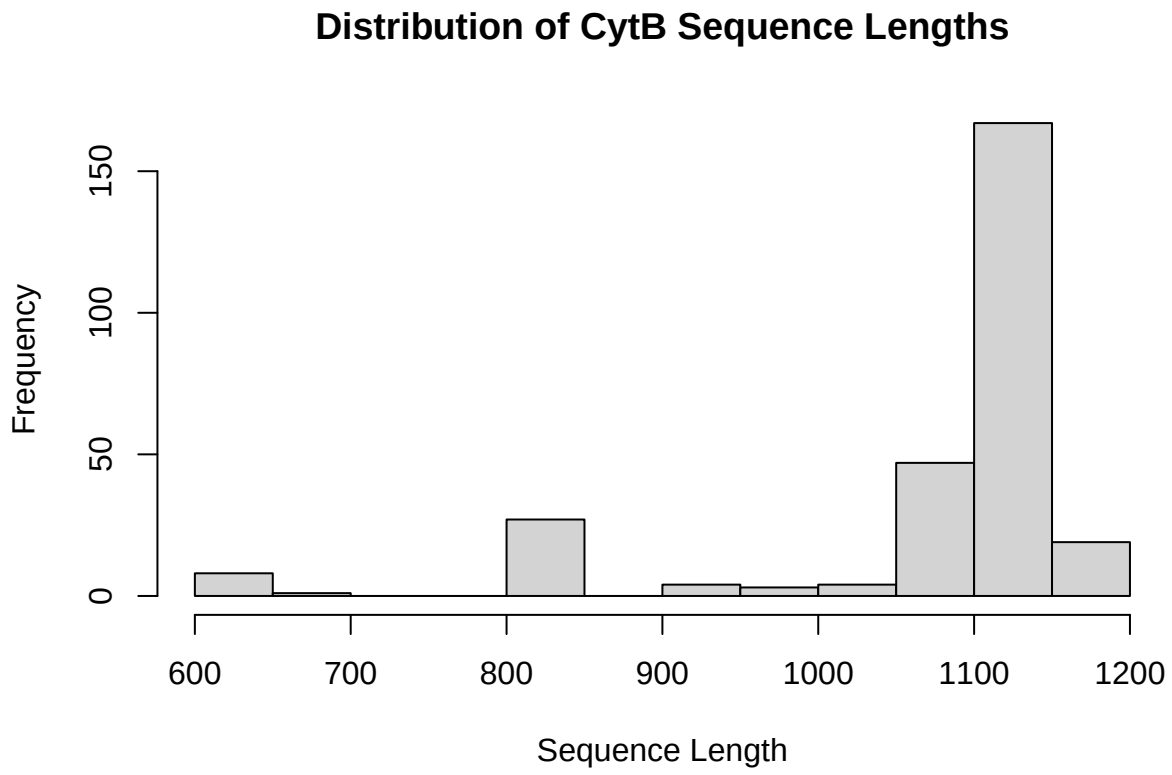

Figure 2: The majority of CytB sequences distributes around the median of nucleotides in length.

Filter out sequences that their length are not the majority.

```
dfCOI_filtered <- dfCOI %>%
  #remove sequences with length shorter than 600
  filter(Sequence_Length >= 600)

dfCytB_filtered <- dfCytB %>%
  #remove sequences that are longer and shorter than the median legnth +/- 100
  filter(Sequence_Length >= median((Sequence_Length) - 100) &
           Sequence_Length <= median((Sequence_Length) + 100))
```

Check if the number of species between 10 to 50.

```
length(unique(dfCOI$Species_Name))
```

```
## [1] 15
```

```
length(unique(dfCytB$Species_Name))
```

```
## [1] 18
```

Here I write a function for DNA sequence alignment and use it for COI gene sequences.

```
# Function for sequence alignment
Alignment <- function(filtered_df) {
  df <- as.data.frame(filtered_df)
  df$nucleotides <- DNAStringSet(df$Sequence)
  # Assign the second word of the species name and the number in the title as
  # the name of each nucleotides sequence to better see the alignment result.
  df$Alignment_id <- paste(word(df$Species_Name, 2L),
                           word(df$Title, 1L),
                           sep = " ")
  names(df$nucleotides) <- df$Alignment_id
  alignment <- AlignSeqs(df$nucleotides)
  return(alignment)
}

dfCOI.alignment.test <- Alignment(dfCOI_filtered)
BrowseSeqs(dfCOI.alignment.test)
```

After seeing the alignment result, I decided to delete the sequences that are highly possible to be outliers as follows: ON097839.1, MT455724.1, MK903722.1, MK903723.1, MK903724.1, MK903725.1, MK903726.1, HM902200.1, MF122058.1, MF122059.1, KC578836.1, KC578837.1, KF929572.1, KF558304.1, KC500133.1, KC578827.1, EU523886.1, EU523878.1, KM286430.1, JQ623904.1, JQ623905.1, JQ623906.1, KX145481.1, KX145032.1, KC500105.1

Here I filtered out the sequences and then write another function for aligning sequences by their amino acid translation, in order to filter outliers that might be missed in the last step.

```
dfCOI_filtered1 <- dfCOI_filtered %>%
  filter(!grepl('ON097839.1|MT455724.1|MK903722.1|MK903723.1|MK903724.1|MK903725.1|MK903726.1|HM902200.

Translation_Alignment <- function(filtered_df) {
  df <- as.data.frame(filtered_df)
  df$nucleotides <- DNAStringSet(df$Sequence)
  df$Alignment_id <- paste(word(df$Species_Name, 2L),
                                word(df$Title, 1L),
                                sep = " ")
  names(df$nucleotides) <- df$Alignment_id
  align.translation <- AlignTranslation(df$nucleotides,
                                        # AAStringSet: Amino Acid StringSet
                                        type="AAStringSet")
  return(align.translation)
```

```
}

dfCOI.translation <- Translation_Alignment(dfCOI_filtered1)
BrowseSeqs(dfCOI.translation)
```

After seeing the Amino Acid alignment, I decide to delete sequences that are clearly different comparing to the majority of the species sequences: ON097800.1, ON097762.1, ON097679.1, MT455123.1, KX145047.1, MG648396.1, MG648380.1, MG648371.1, MG648367.1, MG648366.1, KX145047.1, GQ328798.1, GQ328790.1, HQ960593.1, KC500103.1, FJ809729.1, FJ809728.1, FJ809727.1, FJ809726.1, FJ809725.1, FJ809722.1, HQ960593.1, HQ960592.1, HQ960591.1, HQ960590.1

Here I filter out the sequences then write another function for making a dendrogram to see the quality of data filtering.

```
dfCOI_filtered2 <- dfCOI_filtered1 %>%
    filter(!grepl("ON097800.1|ON097762.1|ON097679.1|MT455123.1|KX145047.1|MG648396.1|MG648380.1|MG64837
        Title))
```

Check the number of species for COI gene after all the data filtering steps.

```
unique(dfCOI_filtered2$Species_Name)
```

```
##  [1] "Acipenser oxyrinchus"      "Acipenser transmontanus"
##  [3] "Acipenseriformes sp."      "Acipenser ruthenus"
##  [5] "Acipenser dabryanus"       "Acipenser fulvescens"
##  [7] "Acipenser stellatus"       "Acipenser baerii"
##  [9] "Acipenser nudiventris"     "Acipenser persicus"
## [11] "Acipenser gueldenstaedtii" "Acipenser medirostris"
## [13] "Acipenser schrenckii"      "Acipenser sinensis"
## [15] "Acipenser brevirostrum"
```

DNA sequence alignment for CytB gene sequences.

```
dfCytB.alignment.test <- Alignment(dfCytB_filtered)
BrowseSeqs(dfCytB.alignment.test)
```

After seeing the alignment result, I decided to delete the sequences in CytB group that are highly possible to be outliers: FJ974042.1, AJ249693.1, AJ249694.1, FJ974041.

Here I filtered out the sequences and then use the function "AlignTranslationTest" I write for aligning sequences by their amino acid translation, in order to filter outliers that might be missed in the last step.

```
dfCytB_filtered1 <- dfCytB_filtered %>%
  filter(!grepl('FJ974042.1|AJ249693.1|AJ249694.1|FJ974041.1', Title))

dfCytB.translation <- Translation_Alignment(dfCytB_filtered1)

BrowseSeqs(dfCytB.translation)
```

After seeing the result of amino acid alignment of CytB group, I decide to delete the sequences that are apparently outliers comparing to the majority of the species: FJ974040.1, AJ277594.1, FJ974043.1

Here I filter out the sequences and check the number of species for CytB gene

```r
dfCytB_filtered2 <- dfCytB_filtered1 %>%
  filter(!grepl('FJ974040.1|AJ277594.1|FJ974043.1', Title))
```

After all the data filtering steps I need to see the common species between COI and CytB groups and filter out other species that are not shared by these two marker genes.

```r
common_species <- intersect(dfCOI_filtered2$Species_Name,
                            dfCytB_filtered2$Species_Name)
length(common_species)
```

```
## [1] 10
```

Filter out other species that are not shared by both COI and CytB group.

```r
dfCOI_final <- dfCOI_filtered2 %>%
  filter(Species_Name %in% common_species)

dfCytB_final <- dfCytB_filtered2 %>%
  filter(Species_Name %in% common_species)
```

I write and use the function "Dendrogram" for making dendrograms to see the quality of data filtering for COI and CytB gene sequences.

```r
Dendrogram <- function(filtered_df, clustering_method) {
  df <- as.data.frame(filtered_df)
  df$nucleotides <- DNAStringSet(df$Sequence)
  # I only used the first to letters of the second word in the column Species_Name
  # and the number of each sequence which is the first word in the column Title
  df$Alignment_id <- paste(substr(word(df$Species_Name, 2L), 1, 2),
                           word(df$Title, 1L),
                           sep = " ")
  names(df$nucleotides) <- df$Alignment_id
  df <- AlignSeqs(df$nucleotides)
  dnaBin <- as.DNAbin(df)
  # I use TN93 here as it is reliable for both COI and CytB gene
  distanceMatrix <- dist.dna(dnaBin, model = "TN93",
                             as.matrix = TRUE, pairwise.deletion = TRUE)
  clusters <- DECIPHER::TreeLine(myDistMatrix = distanceMatrix,
                                 method = clustering_method,
                                 collapse = -1,
                                 showPlot = FALSE,
                                 type = "both",
                                 verbose = FALSE)
  return(clusters)
}
```

I use the package "ggtree" to make the phylogenetic tree of all the sequences from COI after data filtering and sequence alignment to check if there are still outliers. The number of sequences of my data is are too big to show a clear dendogram with all the labels seperate by using TreeLine function from DECIPHER. Therefore I decided to use ggtree to change the layout into circular for better showing the labels.

```
clusters.COI <- Dendrogram(dfCOI_final, "NJ")
COI.testtree <- as.phylo(clusters.COI[[2]])

ggtree(COI.testtree, layout="circular") +
  geom_tiplab(size=2, aes(angle=angle), color='blue')
```

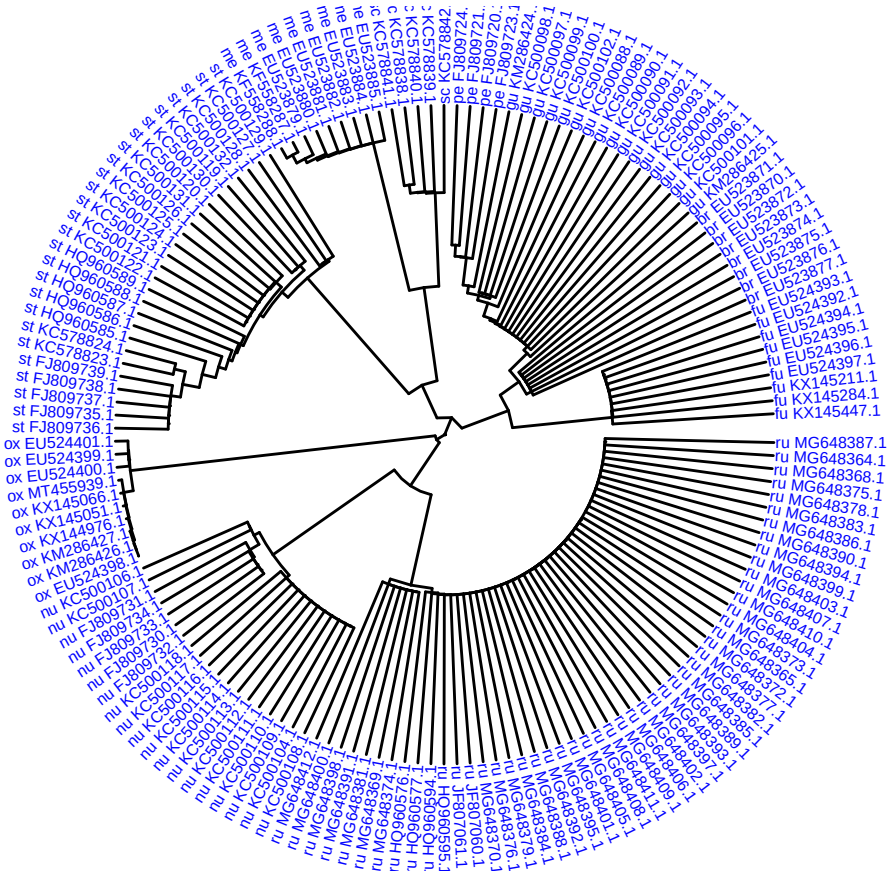

Figure 3: Phylogenetic tree of all sequences belongs to COI gene marker

Draw a phylogenetic tree of all the sequences from CytB after data filtering and sequence alignment to check if there are still outliers.

```
clusters.CytB <- Dendrogram(dfCytB_final, "NJ")
CytB.testtree <- as.phylo(clusters.CytB[[2]])
ggtree(CytB.testtree, layout="circular") +
  geom_tiplab(size=1.8, aes(angle=angle), color='purple')
```

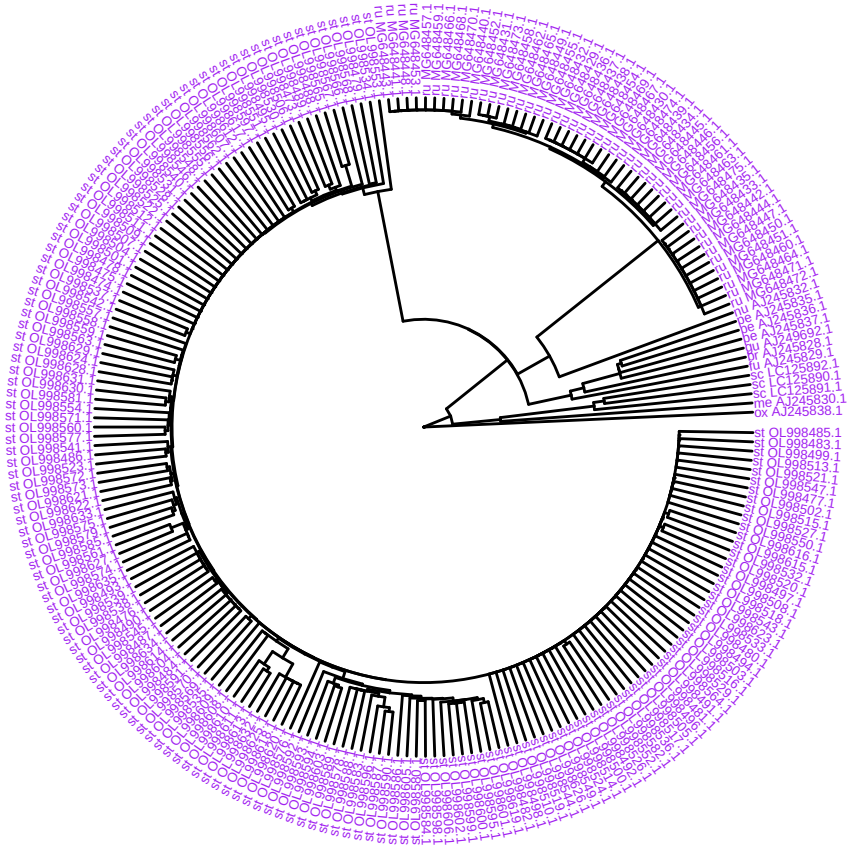

Figure 4: Phylogenetic tree of all sequences belongs to CytB gene marker

Main Software Tools Description

For the main analysis, I choose package "dendextend" (Galili, 2015) as the main software tool to make comparison between the two phylogenetic trees using sequences from COI and CytB markers. It is easy to understand how to use the package for making tanglegram and the code block can be tidy and short. I also used "phytools" to compare the ouputs between the two packages and I found "dendextend" can give the figure I want with less coding and better visualization. The tanglegram made by "phytools" are without scale bar if users do not use "scale.bar" when ploting it. Even if the scale bars are added, this is still not meaningful because the two trees are rescaled internally before plotting in the "cophylo" function. However, there are still some disadvantages in "dendextend": the limited functions to change the tip labels of Neighbour Joining trees. To be more specific, I cannot make the labels lining up neatly as a column. Besides, I can only change the font size to see all the labels otherwise the longest label can only be shown partly.

Code Section 2 - Main Analysis

I write a function to find one representative sequence for each species of the two marker genes.

```
FindRepSeq <- function(dendrogram, df_final) {
  # Transfer the dendrograms into phylos
  tree <- as.phylo(dendrogram)
  # Create tibbles showing the information of the trees
  treetibble <- tidytree::as_tibble(tree)
  treetibble$species <- paste(word(treetibble$label, 1L))
  # Find the median of the parent node of each species
  parent.median <- treetibble %>%
    group_by(species) %>%
    summarize(Median = round(median(parent)))
  # Set seed for making the result reproducible
  set.seed(137)
  # Choose one sequence with the median parent node number to be the representative
  rep.sequences <- treetibble %>%
    group_by(species) %>%
    filter(species != "NA") %>%
    inner_join(parent.median, by = "species") %>%
    filter(parent == Median) %>%
    sample_n(1)
  # Save the column label in rep.sequences into a new vector
  rep_sequences <- rep.sequences$label
  represent <- df_final %>%
    # Create a new column in the df_final that match the pattern off rep.sequences
    mutate(id = paste(substr(word(Species_Name, 2L),1,2),
                      word(Title, 1L), sep = " ")) %>%
    # Filter out sequences that do not match with elements in rep_sequences
    filter(id %in% rep_sequences)
  return(represent)
}
```

Find the representative sequences of species both exist in COI and CytB group and save them into new data frames for the next step.

```
COI_represent <- FindRepSeq(clusters.COI[[2]],dfCOI_final)
CytB_represent <- FindRepSeq(clusters.CytB[[2]],dfCytB_final)
```

Making new dendrograms on species level for creating the tanglegram to compare these two next.

```
# COI representative sequences alignment and making dendrogram
dfCOI_represent <- as.data.frame(COI_represent)
dfCOI_represent$nucleotides <- DNAStringSet(dfCOI_represent$Sequence)
# I use the Species_Name here as the names for the alignment
names(dfCOI_represent$nucleotides) <- word(dfCOI_represent$Species_Name, 2L)
dfCOI_rep_alignment <- AlignSeqs(dfCOI_represent$nucleotides)
COI_dnaBin <- as.DNAbin(dfCOI_rep_alignment)
COI_distanceMatrix <- dist.dna(COI_dnaBin, model = "TN93",
                               as.matrix = TRUE, pairwise.deletion = TRUE)
COI_rep_clusters <- DECIPHER::TreeLine(myDistMatrix = COI_distanceMatrix,
                                       method = "NJ",
```

```
                                    collapse = -1,
                                    showPlot = FALSE,
                                    type = "both",
                                    verbose = FALSE)

# CytB representative sequences alignment and making dendrogram
dfCytB_represent <- as.data.frame(CytB_represent)
dfCytB_represent$nucleotides <- DNAStringSet(dfCytB_represent$Sequence)
names(dfCytB_represent$nucleotides) <- word(dfCytB_represent$Species_Name, 2L)
dfCytB_rep_alignment <- AlignSeqs(dfCytB_represent$nucleotides)
CytB_dnaBin <- as.DNAbin(dfCytB_rep_alignment)
CytB_distanceMatrix <- dist.dna(CytB_dnaBin, model = "TN93",
                            as.matrix = TRUE, pairwise.deletion = TRUE)
CytB_rep_clusters <- DECIPHER::TreeLine(myDistMatrix = CytB_distanceMatrix,
                                    method = "NJ",
                                    collapse = -1,
                                    showPlot = FALSE,
                                    type = "both",
                                    verbose = FALSE)
```

Making the tanglegram to compare the two trees using package "dendextend".

```
# Save the two dendrograms into two vectors
dend_COI <- COI_rep_clusters[[2]]
dend_CytB <- CytB_rep_clusters[[2]]
# Save the two vectors into a dendlist
dl <- dendlist(dend_COI, dend_CytB)
# Create the tanglegram
tanglegram(dl, cex_main = 1,
           # Name the two trees with the two markers
           main_left = "COI Clustering",
           main_right = "CytB Clustering",
           # Change label font
           lab.cex = 0.6,
           # Change columns width, first column(COI Clustering) as 4
           # and third column (CytB Clustering) as 6 to match the scale bars
           # for better visualization
           columns_width = c(4,3,6),
           highlight_branches_lwd = FALSE)
```

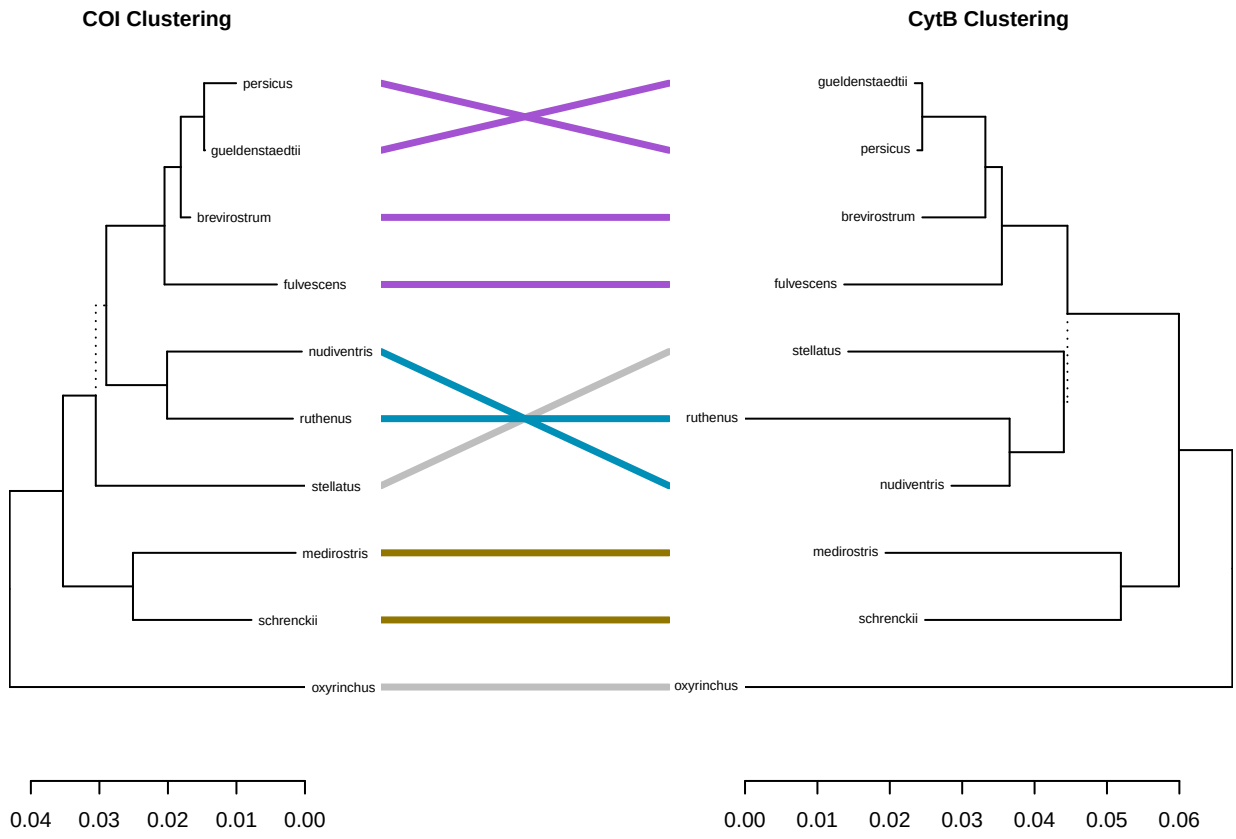

Figure 5: This is a Tanglergram Comparing the Differences between Two Phylogenetic Trees Made from COI and CytB Markers (by package "dendextend"). The connecting line between two trees are based on the common subtrees of both dendrograms. Distinct branches are marked with a dashed line.

Making the tanglegram to compare the two trees using package “phytools”.

```
# Convert the dendrograms of COI and CytB into phylo for using phytools
COI_phylo <- as.phylo(COI_rep_clusters[[2]])
CytB_phylo <- as.phylo(CytB_rep_clusters[[2]])
# Making cophylo tree
obj <- cophylo(COI_phylo, CytB_phylo, rotate = FALSE, print = TRUE)
# Plot the tree
plot(obj, link.type="curved", link.lwd=3, link.lty = "solid",
     link.col = make.transparent("blue",0.25), fsize = 0.8)
```

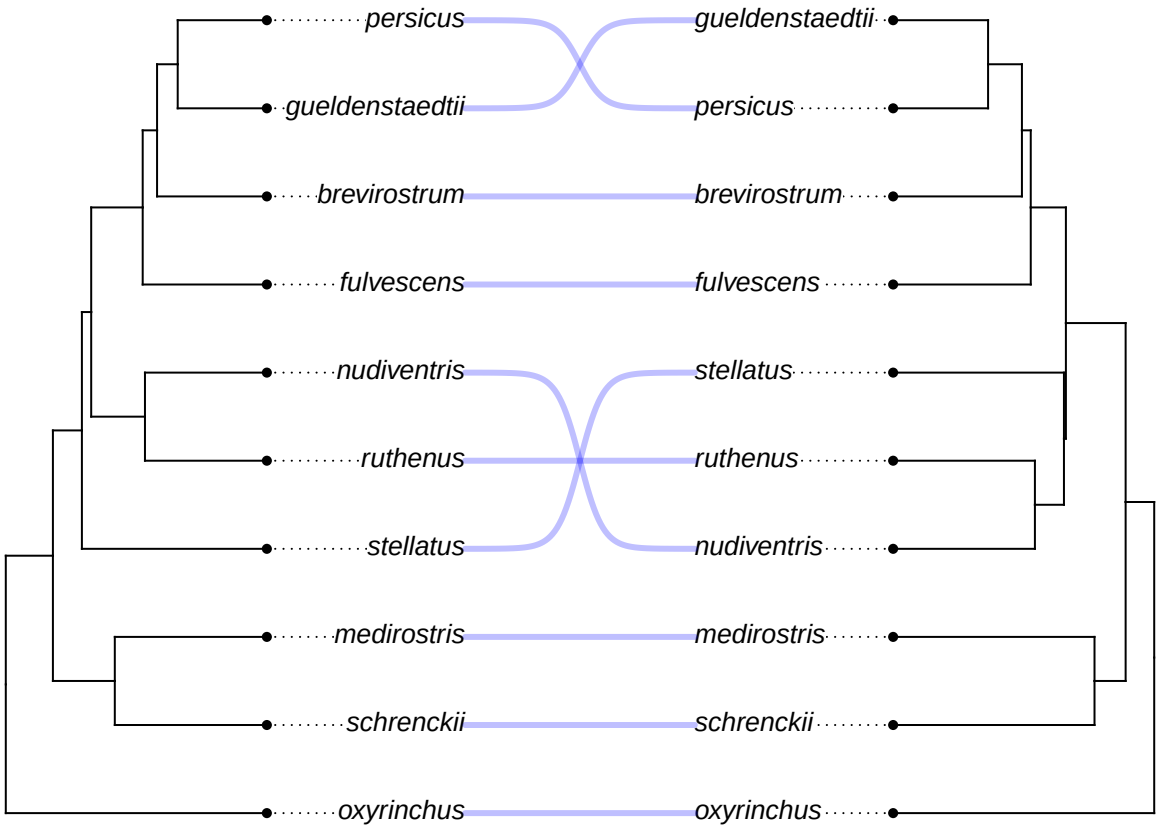

Figure 6: This is a Tanglergram Comparing the Differences between Two Phylogenetic Trees Made from COI and CytB Markers (by package "phytools"). The connecting line between two trees are based on the common subtrees of both dendrograms.

Results and Discussion

After visualizing the comparison between the two phylogenetic trees made from COI gene and CytB gene, the differences do exist in this project. This further strengthens the research finding that evolutionary rates between COI and CytB genes vary. However, this result is also possible to be induced by the the difference in the number of sequences between the same species of different gene markers (Dinh et al., 2019). For example, the relationships among *Acipenser stellatus*, *Acipenser ruthenus*, and *Acipenser nudiventris* differ in the trees from two marker genes. Comparing to the sample sizes of these three species from COI gene, those from CytB gene are significantly unbalanced. The number of sequences of *A. stellatus* is much bigger

than it of *A. ruthenus* and there is only one sequence of *A. nudiventris*. The similar situation also exists on the data sets of *Acipenser persicus* and *Acipenser gueldenstaedtii*.

Considering the limited time for finishing this research project, there are still many improvements can be made for more reliable results. The first step can be added is getting more DNA sequence data from other public data bases. Then conducting a model test to find the best model for the specific data set in this project by phangorn or other tools outside R environmennt, such as ModelTest-NG (Darriba et al., 2020). Checking if the clustering algorithm in this project is appropriate can also be added by applying correlation calculations (Güell et al., 2017).

References

Darriba, D., Posada, D., Kozlov, A. M., Stamatakis, A., Morel, B., & Flouri, T. (2020). ModelTest-NG: a new and scalable tool for the selection of DNA and protein evolutionary models. *Molecular biology and evolution*, 37(1), 291-294.

Dinh, T. D., Ngatia, J. N., Cui, L. Y., Ma, Y., Dhamer, T. D., & Xu, Y. C. (2019). Influence of pairwise genetic distance computation and reference sample size on the reliability of species identification using Cyt b and COI gene fragments in a group of native passerines. *Forensic Science International: Genetics*, 40, 85-95.

Galili, T. (2015). dendextend: an R package for visualizing, adjusting and comparing trees of hierarchical clustering. *Bioinformatics*, 31(22), 3718-3720.

Lavinia, P. D., Kerr, K. C., Tubaro, P. L., Hebert, P. D., & Lijtmaer, D. A. (2016). Calibrating the molecular clock beyond cytochrome b: assessing the evolutionary rate of COI in birds. *Journal of Avian Biology*, 47(1), 84-91.

Lin, C. P., & Danforth, B. N. (2004). How do insect nuclear and mitochondrial gene substitution patterns differ? Insights from Bayesian analyses of combined datasets. *Molecular phylogenetics and evolution*, 30(3), 686-702.

Güell, B, Antos, D, Muratore, I., & Schmitt, C. A. (2017). Module 24: An Intro to Phylogenetic Tree Construction in R. https://fuzzyatelin.github.io/bioanth-stats/module-24/module-24.html

Galil, T. (2023). Introduction to dendextend. https://cran.r-project.org/web/packages/dendextend/vignettes/dendextend.html

LEE. (2022). R Extract first two characters from a column in a dataframe. *Stackoverflow* https://stackoverflow.com/questions/72408598/r-extract-first-two-characters-from-a-column-in-a-dataframe

Revell, L. (2017). Plotting methods for phylogenies & comparative data in R. *Phylogenetic Tools for Comparative Biology* http://www.phytools.org/Cordoba2017/ex/15/Plotting-methods.html

Yu, G. (2022). Data Integration, Manipulation and Visualization of Phylogenetic Trees. *CRC Press*. https://yulab-smu.top/treedata-book/chapter2.html