Alireza Amanatchi (STID:1210976)

Project C: Geography and Evolutionary Diversification

Introduction

The Danio genus, a group of freshwater fish that originate from South and Southeast Asia, like India has captivated scientists, fish enthusiasts and geneticists alike due, to its diversity and ability to thrive in different aquatic environments(1). These small charming fish, including the known zebrafish (Danio rerio) have not only become popular additions to our aquariums but have also played a significant role in scientific research. One of the tools used to unravel the evolutionary mysteries surrounding Danio species is the Cytochrome c Oxidase subunit I (COI) gene(2). COI, which is found in mitochondria is highly regarded for its function as both a powerhouse and a molecular barcode(3). This makes it an invaluable resource for studying phylogenetics and evolutionary biology(3). With this gene, as our compass we embark on a journey to comprehend the geographical factors that have shaped the evolutionary fate of Danio species. Through this exploration we hope to shed light on their tapestry of life.
Our main objective, for this project is to investigate how genetic data and geographic factors interact in the diversification of Danio species. Specifically, we plan to use the COI gene as a tool to create trees that reveal the evolutionary relationships among different Danio species. These phylogenetic trees will provide insights into the history of Danio. By combining data with information, we aim to answer important questions about how geography influences diversification and whether closely related species tend to inhabit the same geographic areas or if their evolution has been driven by large scale separation. Through this study we hope to uncover the connections, within the Danio genus and contribute to our understanding of speciation and evolutionary diversification on a scale.

```
#Install the rentrez package
install.packages("rentrez")
#Install the seqinr package
install.packages("seqinr")
#Install the Biostrings package
install.packages("Biostrings")
# Install the tidyverse package (which includes multiple packages)
install.packages("tidyverse")
#loading all acquired packages via library()
library("rentrez")
library("seqinr")
library("Biostrings")
library("tidyverse")
#performing the search in pubmed by searching the phylum
search.res.danio <- entrez_search(db = "pubmed", term = "Danio")
#now let's see how the classification of teh result
class(search.res.danio)
```

```
#see the resulted hits
search.res.danio
#see the sample id relating to our search
search.res.danio$ids
#classifying the variables which is PMID from pubmed
class(search.res.danio$ids)
#let see how many results we got from our search
length(search.res.danio$ids)
#as there are just 20 which is because of default retmax rate so we need to change the retmax since it limit the number of records returned by by the search
#using to see what are the availbale contents to be searched under "nuccore" to identify the abbreviation using for search.
entrez_db_searchable("nuccore")
#performing the search in pubmed based on nucleotide data base and using phoronis in terms of genus searching
danio_search <- entrez_search(db="nuccore", term = "Danio[ORGN]")
#see the quantity of hits
danio_search

#as COI gene sequences are mostly between 400 to 700, the search needs to be narrow down by the sequence length as well. Also, I put retmax number 100 as I need more than 20 data
danio_search.COI <- entrez_search(db="nuccore", term = "Danio[ORGN] AND COI[gene] AND 400:700[SLEN]", retmax=100)
#see the quantity of the results
length(danio_search.COI$ids)
#review the summary of the result which is assigned to COI_summary
COI_summary <- entrez_summary(db = "nuccore", id = danio_search.COI$ids)
COI_summary
#see the classification
class(COI_summary)

#Helper function called extract_from_esummary takes specific elements from each of your list elements.(take a quick look at organisms file)
extract_from_esummary(COI_summary,"organism")

#Now we have to use entrez_fetch() function from "rentrez" package to solicit the modified data from NCBI
COI_fetch <- entrez_fetch(db = "nuccore", id = danio_search.COI$ids, rettype = "fasta")
#see what is the classification of acquired file from NCBI
class(COI_fetch)
#quick data inspection to get a sense of what the data looks like without displaying the entire dataset.
head(COI_fetch)
#set up specific directory in porder to save my data into.
```

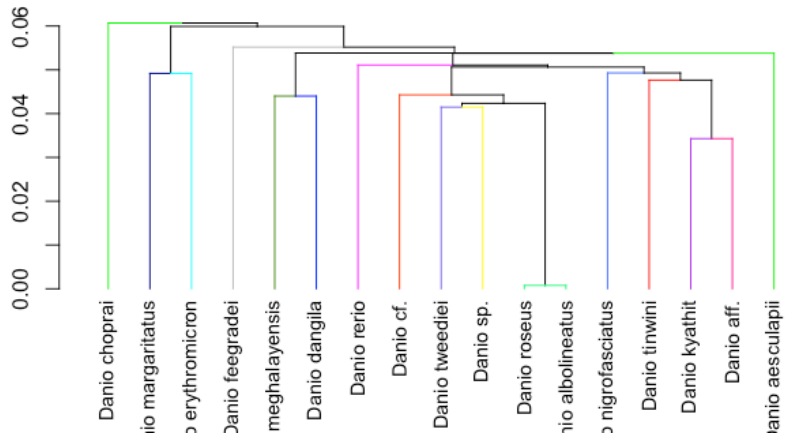
```
setwd("/Users/alireza/Desktop/Bioinformatic/Assignments /#2/Assignment 2")
#investigating current work directory path
getwd()
#write my file in specific working directory as set up earlier, and keep copy of that!
write(COI_fetch, "COI_fetch.fasta", sep = "\n")
#After checking the length of the data and making sure that we haven't downloaded WGS by mistake. we need to read data!
COI.stringset <- readDNAStringSet("COI_fetch.fasta")
#Now lets see what is our file classification and quick overview on
class(COI.stringset)
#head() to display few row from columns that is provided by command names()
head(names(COI.stringset))
#Now we need to make a data frame for further analysis!
dfCOI <- data.frame(COI_title = names(COI.stringset), COI_sequence = paste(COI.stringset))
#lets see how it looks like!
view(dfCOI)

#The first column from our data frame indicate that the names are not very clear and nice! So, I decided to add another column to my dfCOI, named Species_name
#to achieve this, we can use dplyr package to run pipe line
#loading dplyr package
library(dplyr)
#Use the pipe operator (%) to apply a sequence of operations
dfCOI <- dfCOI %>%
# First, use the mutate function to create a new column "Species_Name"
# This new column extracts words 2 to 3 from the "COI_Title" column
  mutate(species_name = word(COI_title, 2L, 3L)) %>%
# Finally, use the select function to rearrange the columns as specified
  select(COI_title, species_name, COI_sequence) %>%
# view the ultimate data frame
  view()
#See the unique species related to the column
unique(dfCOI$species_name)
#Afterward, the data needs to be cleared up regarding alignment; so some packages needs to be installed, followed by loading them.
#installing required packages
install.packages("ape")
#loading ape() package
library(ape)
#installing required packages for DECIPHER
install.packages("RSQLite")
#loading the package
library(RSQLite)
#Install Biocmanager to do tasks related to sequences
```

```
install.packages("BiocManager")
#library the package
library(BiocManager)

#Then, install any needed packages. for alignment and clustering
BiocManager::install(c("Biostrings", "muscle", "msa", "DECIPHER"))
#Load libraries
library(Biostrings)
library(muscle)
library(DECIPHER)
#lets take a looka at to the summmary of the sequence lengths
summary(nchar(dfCOI$COI_sequence))
#using histogram to show the frequency of their length
x <- nchar(dfCOI$COI_sequence)
hist(x)
#The frequency shows good result as most of them are more than 650bp
#we need to also put accession ID as a column as it is likely for species_name to be repeated
thorugh the data.
dfCOI <- dfCOI %>%
  ## This command extracts first words from the "COI_Title" column then name the column as
asccession.id
  mutate(accession.id = word(COI_title, 1L)) %>%
  #selecting the useful columns
  select(accession.id, COI_title, species_name, COI_sequence) %>%
  #view the data frame
  view()
#as there are more than one sequences for each species so we one random sample from them.
Indeed, this step will be helpful when I want to merge 2 data frames as there is no lon and lat
data from NCBI.
dfCOI <- dfCOI%>%
#grouping by the species_name column
  group_by(species_name)%>%
#a random sample for each
  sample_n(1)%>%
  view()

#see the classification
class(dfCOI)
#see the classification of sequences
class(dfCOI$COI_sequence)
#change the format to data.frame
dfCOI <- as.data.frame(dfCOI)
#we need to change it to the format to do DNA sequences task
dfCOI$COI_sequence <- DNAStringSet(dfCOI$COI_sequence)
```

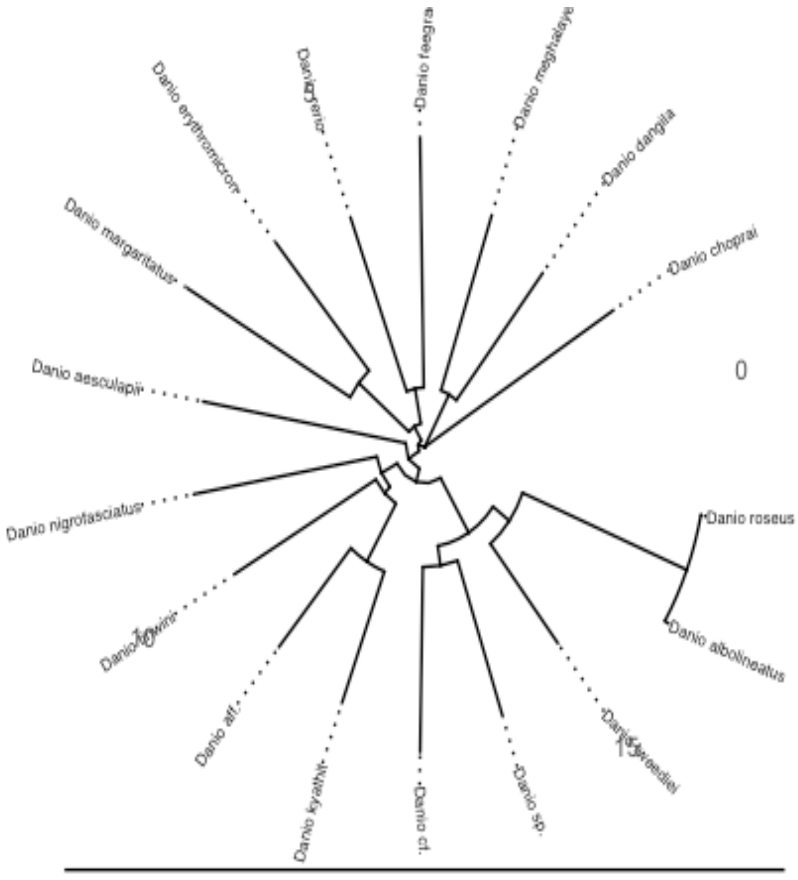
```
#see the classification again!
class(dfCOI$COI_sequence)
#lets put species_name as a individual column otherwise we get our result based on accession.id
names(dfCOI$COI_sequence) <- dfCOI$species_name
#see the columns
names(dfCOI$COI_sequence)
#take a look at sequences in R
dfCOI$COI_sequence
#using BrowsSeq to look at them in HTML format which is easier to read through
BrowseSeqs(dfCOI$COI_sequence)
#double-check whether ther are any N/A data
sum(is.na(dfCOI$COI_sequence))

#### ALIGNMENT
#alignment needs to be done by muscle and the gap open set up at -300 at first try
dfCOI.alignment <- DNAStringSet(muscle::muscle(dfCOI$COI_sequence, gapopen= -300), use.names=TRUE)
#lets see the results in HTML version
BrowseSeqs(dfCOI.alignment)
#now run the alignment again with default -600 as there are not many gaps
dfCOI.alignment <- DNAStringSet(muscle::muscle(dfCOI$COI_sequence, gapopen= -600), use.names=TRUE)
#lets see the results in HTML version
BrowseSeqs(dfCOI.alignment)
#lets take a look at first sequence length
length(dfCOI.alignment[[1]])

#let see the quantity of gaps exist in our alignment
mean(unlist(lapply(as.character(dfCOI.alignment), str_count, "-")))
#also getting a summary
summary(unlist(lapply(as.character(dfCOI.alignment), str_count, "-")))

#now it looks like ultimately we reached reasonable rates for the gaps by making comparison of various gapopen rates between -300to -600.
#now we look for the frequency of the gaps based on the sequence length as well to give us a insight how is our result.
gap.Freq <- unlist(lapply(as.character(dfCOI.alignment), str_count, "-"))
hist(gap.Freq)
#then we can save the alignment file as we might use it later in other softwares just in case.
writeXStringSet(dfCOI.alignment, file = "dfCOI.alignment.fasta")
```

###clustering and phylogenetic analysis

#Our alignment needs to be set up as DNAbin to do clustering and phylogeny
```
dna.bin.COI.alignment <- as.DNAbin(dfCOI.alignment)
```
#see the class
```
class(dfCOI.alignment)
```
#for clustering 3 percent is assigned
```
threshhold <- 0.03
```
#TN93 model applied for analysing the distance between the sequences
```
taken.model <- "TN93"
```
#The given R code calculates a genetic distance matrix from a DNA sequence alignment. It uses a specified modelto determine genetic distances and handles missing data with pairwise deletion. The resulting distance matrix, stored in distanceMatrix, is useful for tasks such as phylogenetic tree construction and genetic diversity assessment in DNA sequence analysis.
```
distanceMatrix <- dist.dna(dna.bin.COI.alignment, model = taken.model, as.matrix = TRUE, pairwise.deletion = TRUE)
```
#view first rows and columns data about the distance rate
```
head(distanceMatrix)
```
#This R code uses the DECIPHER package to perform hierarchical clustering on a genetic distance matrix (distanceMatrix). It employs the "single" linkage method to group sequences into clusters based on their genetic distances. The threshold variable determines when to stop forming clusters. If showPlot is set to TRUE, it displays a dendrogram plot illustrating the clustering results. The outcome is stored in the dfCOI.cluster variable, providing insights into the relationships among the DNA sequences.
```
dfCOI.cluster <- DECIPHER::TreeLine(myDistMatrix = distanceMatrix,
                                    method = "single",
                                    cutoff = threshhold,
                                    showPlot = TRUE,
                                    type = "dendrogram",
                                   verbose = TRUE)
```



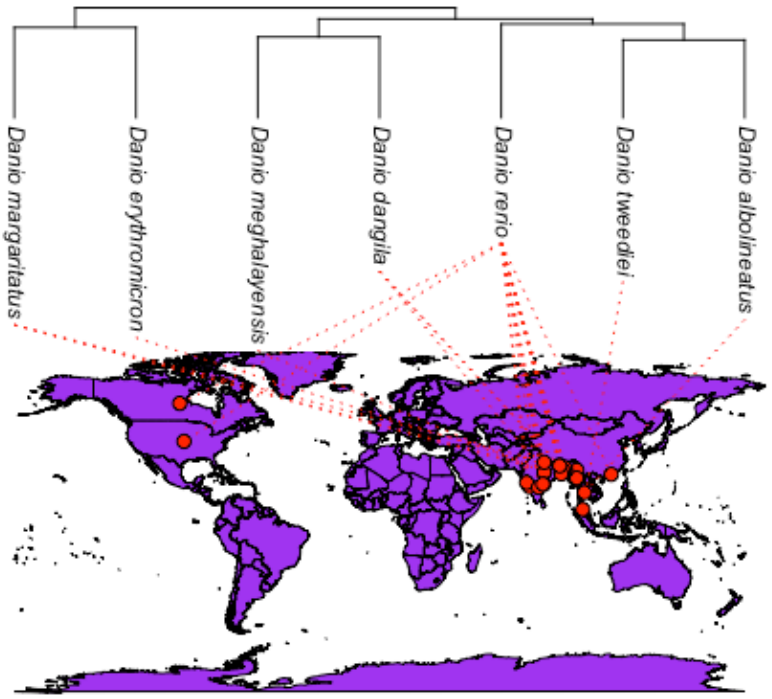
```
#figure 2: circular phylogram
#install required package
install.packages("phangorn")
#load the library
library(phangorn)
#install required package
install.packages("ggplot2")
#load the library
library("ggplot2")
#install required package
BiocManager::install("ggtree")
#load the package
library("ggtree")
#make a matrix and assign it matrix.COI.alignment
matrix.COI.alignment <- as.matrix(distanceMatrix)
#making a tree with neighbor joining method
tree.ape <- nj(matrix.COI.alignment)
#see the classification
class(dna.bin.COI.alignment)
#review the tip labels and internals nodes of the tree
tree.ape
#circular phylogram
library(ggtree)
#The provided R code using the ggtree package creates a circular phylogenetic tree plot from the tree.ape object. It suppresses the legend with theme_tree2(legend.position = "none"), and labels the tree tips with geom_tiplab(align = TRUE, size = 2). The resulting tree_plot variable stores the tree plot with these customizations for visualization and analysis of phylogenetic relationships in a circular layout.
tree_plot <- ggtree(tree.ape, layout = "circular") +
  theme_tree2(legend.position = "none") +
  geom_tiplab(align = TRUE, size = 2)
# view the tree
tree_plot
```



#Next step would be merging data frames from BOLD and NCBI as NCBI doesn't provide data related to GPS (longitude & Latitude)
#Organizing the data mined by Bold as it is including GPS data
#getting data from Bold via API
dfpho <-
read_tsv("http://www.boldsystems.org/index.php/API_Public/combined?taxon=Danio&format=tsv")
#saving the file
write_tsv(dfpho,"Phoro.Bold")
#See the classification
class(dfpho)
#viewing the data
view(dfpho)
#See the including columns
names(dfpho)

```
#Using pipe line to organizing data from bold
dfpho.COI.geo <- dfpho %>%
  # Filter rows with marker code "COI-5P"
  filter(markercode == "COI-5P") %>%
  # Filter out rows with missing lat or lon values
  filter(!is.na(lat) & !is.na(lon)) %>%
  # Filter out rows with missing species_name
  filter(!is.na(species_name)) %>%
  #selecting columns contains of data related to species_name, latitude, and longitude
  select(species_name, lat, lon) %>%
  #review the resulted data frame
  view()
#lets see the unique species in their column
unique(dfpho.COI.geo$species_name)
#see the sequences classification
class(dfCOI$COI_sequence)
#as it is biostring it is needed to be character for further manipulation
dfCOI$COI_sequence <- as.character(dfCOI$COI_sequence)

#preparing data from NCBI regarding merging to acquired data from BOLD
  dfCOI.V2 <- dfCOI %>%
  # First, use the mutate function to create a new column "Species_Name"
  # This command extracts second words   from the "COI_Title" column
  mutate(species_name = word(COI_title, 2L, 3L)) %>%
  # Finally, use the select function to rearrange the columns as specified
  select(species_name, COI_sequence) %>%
#removing N/A from species_name
  filter(!is.na(species_name))%>%
  # view the ultimate data frame
  view()

#lets merge these 2 data frame together.
dfmerge <- merge(dfCOI.V2, dfpho.COI.geo, by="species_name", all=F)
#view the data frame
view(dfmerge)
#lets distinct combination of these data frames in species_name, lat, lon
dfmerge2 <- dfmerge %>%
  distinct(species_name, lat, lon) %>%
  view()
#lets reconstruct the data by removing a column and convert it into a matrix to make sure there
is no duplicate
row_names <- dfmerge2$species_name
dfmerge2$species_name <- NULL
dfmerge2 <- as.matrix(dfmerge2)
```

```
row.names(dfmerge2) <- row_names
# lets convert our tree to hclus which is proper choice for using clusters in hierarchical shape
dfCOI.cluster <- as.hclust(dfCOI.cluster)
#to use our tree in phytools we need them in phytool format
dfCOI.cluster.phylo <- as.phylo(dfCOI.cluster)
#view the file and tips
dfCOI.cluster.phylo

#Figure 3 Phylogeny tree on mapping plot
## now we have to merge the data from NCBI and BOLD as there are no data related to GPS or geographical analysis
#install packages
install.packages("phytools")
#load the package
library(phytools)
#install the package
install.packages("maps")
#load the package
library(maps)
#loading by library
library(mapdata)
#by using keep tip we just want to use tip labels in our figure to match to the spots
tree <- keep.tip(dfCOI.cluster.phylo, unique(rownames(dfmerge2)))
#This R code employs the phylo.to.map function to create a map visualization with a phylogenetic tree (phylogram) overlaid on it. The tree variable represents the phylogenetic tree, and dfmerge2 contains geographic data. It sets the type to "phylogram," avoids tree rotation (rotate = FALSE), and doesn't display the map immediately (plot = FALSE). The resulting map visualization is stored in the objective variable for further examination or plotting if needed.
objective<- phylo.to.map(tree, dfmerge2, type = "phylogram", rotate = F, plot = F)
#This R code uses the plot function to create a plot of the map visualization stored in the objective variable. It customizes various aspects of the plot, including the panel split, font size, font type, aspect ratio, line style, background color.
plot(objective, split = c(0.5, 0.5), fsize = 0.75, ftype = "i", asp = 1, from.tip = F, lty = "dotted", map.bg = "purple", map.fill = "lightblue", lwd = 1, pts = F, cex.points = 1, delimit_map = T)
```



Results and Discussion

Our dendrogram illuminates the intricate web of evolutionary relationships within the *Cyprinidae* group. The branching patterns depict the genetic distances between distinct *Danio asesculapii and Danio chorapi*, unveiling the diversity and relatedness among them. Figure 1. indicates that the most 2 relevant species are *Danio roseus* and *Danio albolineatus* of which share a common ancestor. However, further study is needed to make comparison with their evolutionary speed trend. Additionally, by comparing this genetic data to geographical information (Figure 3), we observe that species with shared geographic proximity often cluster together, highlighting the potential influence of geography on their genetic relatedness such as all species showed on map diversification. However, it is noteworthy that within our analysis, we have identified two instances of Danio rerio occurrences in distinct geographic locations. These findings might consider as meticulous investigation to elucidate the underlying ecological and environmental drivers responsible for such distributional patterns. Further research is imperative to discern the potential factors influencing the presence and adaptability of Danio rerio in these disparate habitats or their evolutionary.

Furthermore, according to figure 3 the geographical distribution of *Danio tweedei, Danio rerio*, and *Danio albolineatus* species are mirrored in the dendrogram. Therefore, they share geographic proximity often cluster together, indicating the potential influence of geography on the evolutionary processes that have shaped this taxon.

In addition to this, Figure 1 reveals the presence of monophyletic clades, where species cluster together based on their shared ancestry. These clades are indicative of shared evolutionary history and common descent. Such groupings provide insights into the evolutionary patterns and the degree of divergence among D. *nigrofasciatus, D. aff, D. tinwini, and D. kyathit* lineages. In

terms of conservation implication, by comparing Figure 3 and 1 it might claim that most of species may share similar ecological requirements and thus may benefit from similar conservation strategies. Ultimately, although the figure 2 is sharing mostly the same info same as figure 1, it compares the divergence among the species as the length of branches are measurable due to neighbor joining method for clustering.

In conclusion, this phylogenetic dendrogram serves as a valuable tool for unraveling the evolutionary history of *Danio* in species level. It offers a platform for further investigations into the genetic, ecological, and geographical factors that have contributed to the diversification and speciation of this taxon. A limitation of this study lies in the unavailability of extensive GPS data, which restricted our ability to precisely map the geographical distribution of *Danio* species. The lack of comprehensive geographic information hindered our capacity to unravel fine-scale biogeographical patterns and to discern the influence of specific environmental variables on the distribution of these species. In addition, the unrooted nature of the trees limited our capacity to infer the temporal aspects of speciation events. Future research should focus on expanding geographical datasets and incorporating more genetic markers to overcome these limitations and provide a more comprehensive understanding of the intricate interplay between genetics, geography, and the environment in the evolutionary history of the *Danio* genus.

References

1. McCluskey BM, Postlethwait JH. Phylogeny of Zebrafish, a “Model Species,” within Danio, a “Model Genus.” Mol Biol Evol [Internet]. 2015 Mar 1 [cited 2023 Oct 25];32(3):635. Available from: /pmc/articles/PMC4327152/
2. Bingpeng X, Heshan L, Zhilan Z, Chunguang W, Yanguo W, Jianjun W. DNA barcoding for identification of fish species in the Taiwan Strait. PLoS One [Internet]. 2018 Jun 1 [cited 2023 Oct 25];13(6). Available from: https://pubmed.ncbi.nlm.nih.gov/29856794/
3. Alberts B, Johnson A, Lewis J, Raff M, Roberts K, Walter P. The Mitochondrion. 2002 [cited 2023 Oct 25]; Available from: https://www.ncbi.nlm.nih.gov/books/NBK26894/