# Project 3: Geography and Evolutionary Diversification for genus Ursus

Alisha Soares

## Introduction

Ursus is a genus of the Ursidae family of bears. There are four main species within the Ursus genus: *Ursus americanus* (American black bear), *Ursus arctos* (Brown bear), *Ursus maritimus* (Polar bear), and *Ursus thibetanus* (Asian black bear). Many bear species are considered endangered, and many are labeled vulnerable species by IUCN, including *Ursus maritimus* and *Ursus thibetanus* ( Garshelis & Steinmetz, 2016; Wiig et al., 2015). Conservationists are especially interested in the relationships between Ursus species, as populations have decreased due to human interference, habitat loss, and climate change (Su et al. 2018). The phylogeny of bears both at the genus and family level is complex as evolutionary radiation is rapid (Kumar et al. 2017). The recent growth in computational biology has galvanized scientific research, especially in the fields of ecology and evolution. Software tools and algorithms can use genomic data to discover differences between species within a genus.

As bear populations are separated geographically, it is interesting to study the geophylogeny of the species and how they have diversified over time. The discovery of hybrid specimens in some regions (Cahill et al., 2015) has raised the question about diversification and geographic location in the evolution of the species. For this study, the COI marker gene was used to analyze the variation in Ursus species, to see if conclusions may be made about the diversification of the species and their geographic distributions.

## Description of Data Set

The data was obtained from BOLD Systems, for the genus Ursus, which the is Latin word for bear. The data used for the study had 80 variables of which the process id, species name, marker code (COI-5P), nucleotide sequences, country, longitude and latitude were kept for further analysis. The process id was kept for recording purposes, while the other variables kept were used for generation of the dendrogram and geophylogeny. The initial sample contained 666 records for Ursus, which was reduced to 31 samples after filtering steps were completed. These filtering steps ensured that only data which included each of the variables mentioned above were part of the final data set. Of this smaller sample only sequences which met the missing data allowance of 1% were kept, and Ns and gaps removed from the sequences. The filtered data set dfCOI was then used in the multiple sequence alignment and further analysis.

## Code - Part1

### Packages and Libraries

```
#install.packages("tidyverse")
library(tidyverse)
```

```
#install.packages("seqinr")
library(seqinr)
#install.packages("stringi")
library(stringi)
#install.packages("ape")
library(ape)
#install.packages("RSQLite")
library(RSQLite)
#install.packages("phytools")
library(phytools)
#install.packages("maps")
library(maps)
#install.packages("mapdata")
library(mapdata)
#install.packages("ggdendro")
library(ggdendro)
#install.packages("BiocManager")
library(BiocManager)
#BiocManager::install(c("Biostrings", "muscle", "DECIPHER"))
library(Biostrings)
library(muscle)
library(DECIPHER)
```

## Data Acquisition

### Obtain Ursus data from BOLD Systems.

```
data <- read_tsv(
  file = "http://www.boldsystems.org/index.php/API_Public/combined?taxon=Ursus&format=tsv")

# Write the data to disk.
write_tsv(data, "Ursus_BOLD_data.tsv")
```

### Set Variables

```
# Read in the data file and assign the data to a new data frame.
data <- read_tsv(file = "Ursus_BOLD_data.tsv")

# Define the amount of missing data and sequence length variability.
missing.data <- 0.01
length.var <- 50

# Specify the model of molecular evolution to be used for estimation of distance matrix.
# Also specify the clustering threshold and clustering method.
chosen.model <- "K80"
clustering.threshold <- 0.01
clustering.method <- "single"
```

## Data Filtering

```
# Create a new data frame, select the columns to be included in the dataframe.
# Filter the nucleotide sequences present for COI-5P marker code.
# Remove Ns and gaps from the sequences.
# Remove records that lack species name, sequences or geographical data.
dfCOI <- data %>%
  select(processid, genus_name, species_name, markercode, nucleotides, country, lat, lon)%>%
  filter(!is.na(species_name)) %>%
  filter(!is.na(lat)) %>%
  filter(!is.na(lon)) %>%
  filter(!is.na(nucleotides)) %>%
  filter(markercode == "COI-5P") %>%
  mutate(nucleotides2 = str_remove_all(nucleotides, "^N+|N+$|-")) %>%
  filter(str_count(nucleotides2, "N") <= (missing.data * str_count(nucleotides))) %>%
  filter(str_count(nucleotides2) >= median(str_count(nucleotides2)) - length.var &
           str_count(nucleotides2) <= median(str_count(nucleotides2)) + length.var)

# Check that all gaps have been removed.
sum(str_count(dfCOI$nucleotides2, "-"))
```

```
## [1] 0
```

```
# remove original data frame
rm(data)
```

```
# Add new column with numerical identifier to show sequence number.
dfCOI$ID <- 1:nrow(dfCOI)

# Create a unique identifier for each sequence.
# Use the numerical ID of the sequence and the species name of the sample.
dfCOI$Unique_ID <- paste(dfCOI$species_name, dfCOI$ID, sep = "_")
```

## Data Exploration

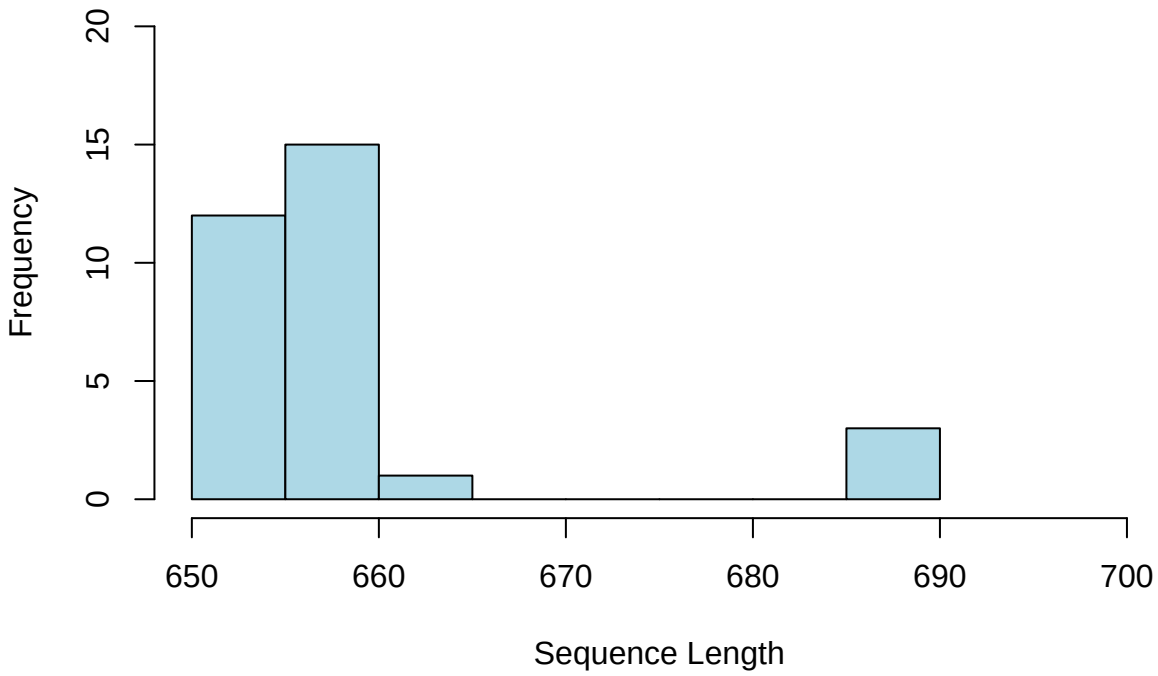
```
# Create a summary of sequence lengths for data extrapolation
summary(nchar(dfCOI$nucleotides2))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   652.0   652.0   657.0   658.4   658.0   687.0
```

Table 1: Summary of nucleotide sequence lengths

```
# Create a histogram to display the distribution of sequence lengths.
hist(nchar(dfCOI$nucleotides2),xlab = "Sequence Length", ylab = "Frequency",
     main = "Figure 1: Frequency Histogram of COI Sequence Lengths",
     col = "lightblue",
     ylim = c(0, 20), xlim = c(650, 700),
     breaks = 5)
```

**Figure 1: Frequency Histogram of COI Sequence Lengths**

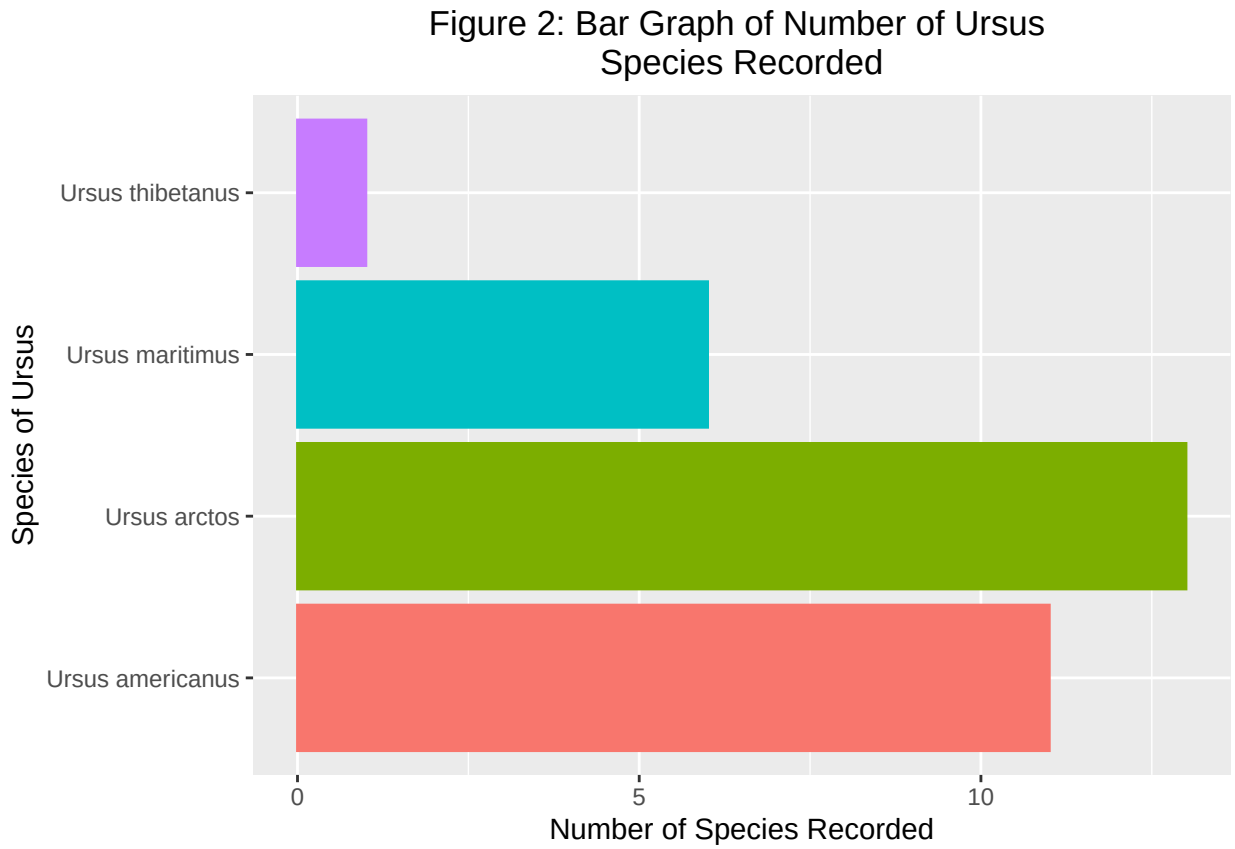



```
sort(table(dfCOI$species_name), decreasing = TRUE)
```

```
##
##     Ursus arctos Ursus americanus  Ursus maritimus Ursus thibetanus
##               13               11                6                1
```

Table 2: The number of sequences available for each species.

```
# Create a bar graph to see the number of species of Ursus found.
dfSpecies.count <- dfCOI %>%
  dplyr::group_by(species_name) %>%
  base::subset(!species_name == "NA")

ggplot(data = dfSpecies.count) +
  geom_bar(mapping = aes(x = species_name, colour = species_name, fill = species_name)) +
  labs(title = "Figure 2: Bar Graph of Number of Ursus \n Species Recorded ",
       x = "Species of Ursus", y = "Number of Species Recorded") +
  theme(legend.position="none", plot.title = element_text(hjust = 0.5)) +
  coord_flip()
```




```
# Change nucleotide data formart for use in muscle package from Bioconductor.
dfCOI <- as.data.frame(dfCOI)
dfCOI$nucleotides2 <- DNAStringSet(dfCOI$nucleotides)

# Confirm the changes
class(dfCOI)
```

```
## [1] "data.frame"
```

```
class(dfCOI$nucleotides2)
```

```
## [1] "DNAStringSet"
## attr(,"package")
## [1] "Biostrings"
```

## Description of Main Software Tools

The muscle algorithm and DECIPHER method were used for the multiple sequence alignment needed for the analysis. After consideration, the chosen model of molecular evolution was the Kimura-2-parameter genetic distance (K80) model. The model is used for estimating the distance matrix and takes into account the differences in transversion and transition rates between nucleotides, and considers an equal distribution for all nucleotides bases. This model was the best option for this study as it accounts for the underlying biological phenomenon causing differences in species. The clustering threshold was set to 0.01, with clusters being at most 1% different from each other.

There were two options considered for the clustering method for the analysis: neighbor-joining and single linkage clustering. The Neighbor-joining method assumes that lineages evolve at different rates and can be more accurate than unweighted pair group methods (Kim et al. 1993). The analysis was first conducted with Neighbor-joining, and the resulting dendrogram within R provided more accurate branch lengths. However, as the branch lengths were very short, the dendrogram in R was trickier to analyze. Single linkage clustering improved the readability of the dendrogram and provided the same tree topology as the Neighbor-joining clustering method. The main difference between the two clustering methods was in the estimation of the branch lengths. This issue was solved by also looking at the dendrograms generated in MEGAX using Neighbor-joining.

# Code - Part2

## Sequence Alignment

```
# Set numerical ID as unique identifiers for the sequence names.
# This is for the file to be used in MEGAX.
names(dfCOI$nucleotides2) <- dfCOI$ID

# Align the sequences using default settings of muscle package.
# Here gap opening penalty = 400 and maxiters = 8.
dfCOI.alignment <- DNAStringSet(muscle::muscle(dfCOI$nucleotides2))

# Write to file.
writeXStringSet(dfCOI.alignment, "dfUrsus.fasta")

# Use BrowseSeqs function from DECIPHER package to view the sequences in the browser
#BrowseSeqs(dfCOI.alignment)
```

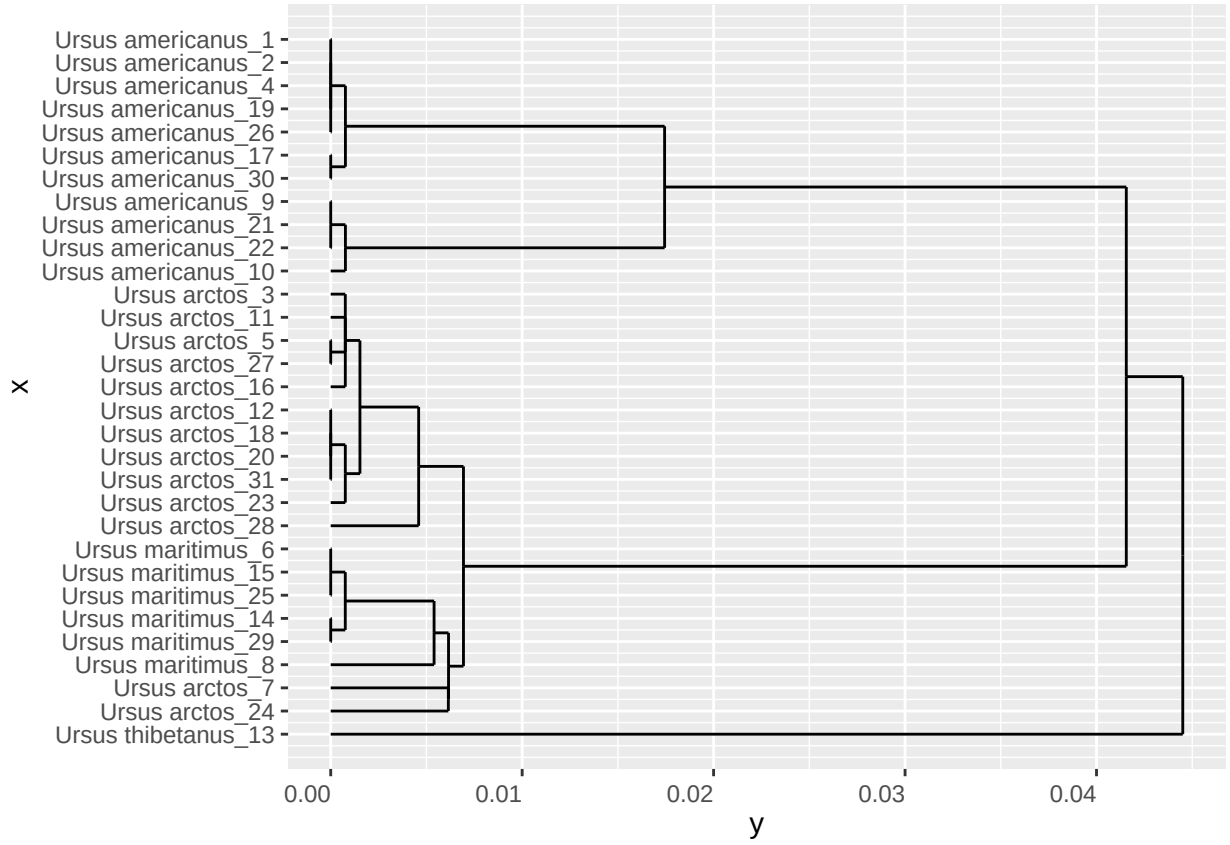
## Sequence Clustering

```
# Set unique identifiers for the sequence names.
# Include the species names as well as numerical ID
names(dfCOI$nucleotides2) <- dfCOI$Unique_ID

# Align the sequences using default settings of muscle package.
# Here gap opening penalty = 400 and maxiters = 8.
dfCOI.alignment <- DNAStringSet(muscle::muscle(dfCOI$nucleotides2))

# Convert data class to DNAbin (for clustering) using a function from ape package.
dnaBin.COI <- as.DNAbin(dfCOI.alignment)

# Create the distance matrix. then view some of the results to check the data.
distanceMatrix <- dist.dna(dnaBin.COI, model = chosen.model, as.matrix = TRUE,
                           pairwise.deletion = TRUE)
#head(distanceMatrix)
#boxplot(distanceMatrix)

# Cluster using variables identified in Code-Part1. This should give us a dendrogram.
#clusters.COI <- IdClusters(distanceMatrix,
```

```
                        #method = clustering.method,
                        #cutoff = clustering.threshold,
                        #showPlot = TRUE,
                        #type = "both",
                        #verbose = TRUE)
```

Here, ggdendro, a program to be used with ggplot2 was used to create the dendrogram. It has a wrapper code that allows ggplot2 to create a dendrogram using once line of code. As of now the program does not have a way to change the x and y axis labels using the once line of code. So the labels have been described in the caption below the figure.

```
ggdendrogram(clusters.COI[[2]], rotate = TRUE, size = 4, theme_dendro = FALSE,
             main = "Cluster Dendrogram for Ursus species",
             ylab = "Genetic Variability", xlab = "Specimen ID")
```




```
# Also see dendrogram generated in MEGAX on page 15 of the report.
```

**Figure 3: Dendrogram of Clustered Aligned Sequences for Ursus Species. x is the species names and y is the genetic variability.**

## Geophylogeny

```
# Multiple sequence alignment was generated in MEGAX and saved in the Newick format.
Phylo <- "(((((((15:0.00000000,23:0.00000000):0.00000387,21:0.00152079):0.00001616,
(5:0.00148530,26:0.00003936):0.00151420):0.00274830,(8:0.00000000,16:0.00194764):
0.00031887):0.00005610,(6:0.00034629,24:0.00000000):0.00030346):0.00009279,7:
0.00024444):0.00306313,17:0.00603868):0.00224303,((18:0.00504330,28:0.00723928):
0.00283780,(19:0.01048269,(11:0.00000000,(30:0.00000000,(31:0.00000000,(10:0.00000000,
29:0.00000000):0.00154709):0.00001814):0.00001034):0.00042326):0.00358681):0.00458937,
(9:0.05362622,((((4:0.00000000,20:0.00000000):0.00000000,3:0.00000000):0.00002518,14:
0.00151119):0.01335415,(1:0.00000000,(12:0.00000000,(13:0.00000000,(25:0.00000000,
(27:0.00000000,(2:0.00000000,22:0.00000000):0.00165878):0.00006347):0.00003296):0.00001714):
0.00000893):0.02164138):0.02624070):0.03933877);"

# Use the read.tree function from ape to use the above data.
# Check the output to see how many tips and internal nodes in the tree.
tree <-read.tree(text = Phylo)
tree
```

```
##
## Phylogenetic tree with 31 tips and 29 internal nodes.
##
## Tip labels:
##   15, 23, 21, 5, 26, 8, ...
##
## Unrooted; includes branch lengths.
```

```
# Use function from phytools to ensure that all tips in tree are aligned
treeUltra <- phytools::force.ultrametric(tree)
ape::is.rooted(treeUltra)
```

```
## [1] FALSE
```

```
# Root tree so that the phylogenetic tree can be plotted onto the map
Tree <- root(treeUltra, 4, resolve.root = TRUE)
Tree
```

```
##
## Phylogenetic tree with 31 tips and 30 internal nodes.
##
## Tip labels:
##   15, 23, 21, 5, 26, 8, ...
##
## Rooted; includes branch lengths.
```

```
# Check that the tree is bifurcating.
ape::is.binary(Tree)
```

```
## [1] TRUE
```

```
# Make a separate data set for the geographic coordinates of the Ursus data.
location <- dfCOI[,c(10,7,8)]
```

```
# Make a matrix for use in mapping.
columnames <- column_to_rownames(location, var = "ID")
#class(columnames)

Geo <- as.matrix(columnames)
class(Geo)
```

```
## [1] "matrix" "array"
```

```
#Geo
```

```
# Assign colors to ID
cols<-setNames(sample(rainbow(n=Ntip(Tree))),
               Tree$tip.label)

#Assign an object to plot using the phylo.to.map function
#obj <- phytools::phylo.to.map(tree = Tree, coords = Geo, rotate = TRUE,
                               #type = "phylogram", fsize = 0.7)
```

```
phytools::plot.phylo.to.map(obj, rotate = TRUE, type = "phylogram", fsize = 0.7,
                            colors = cols)
```

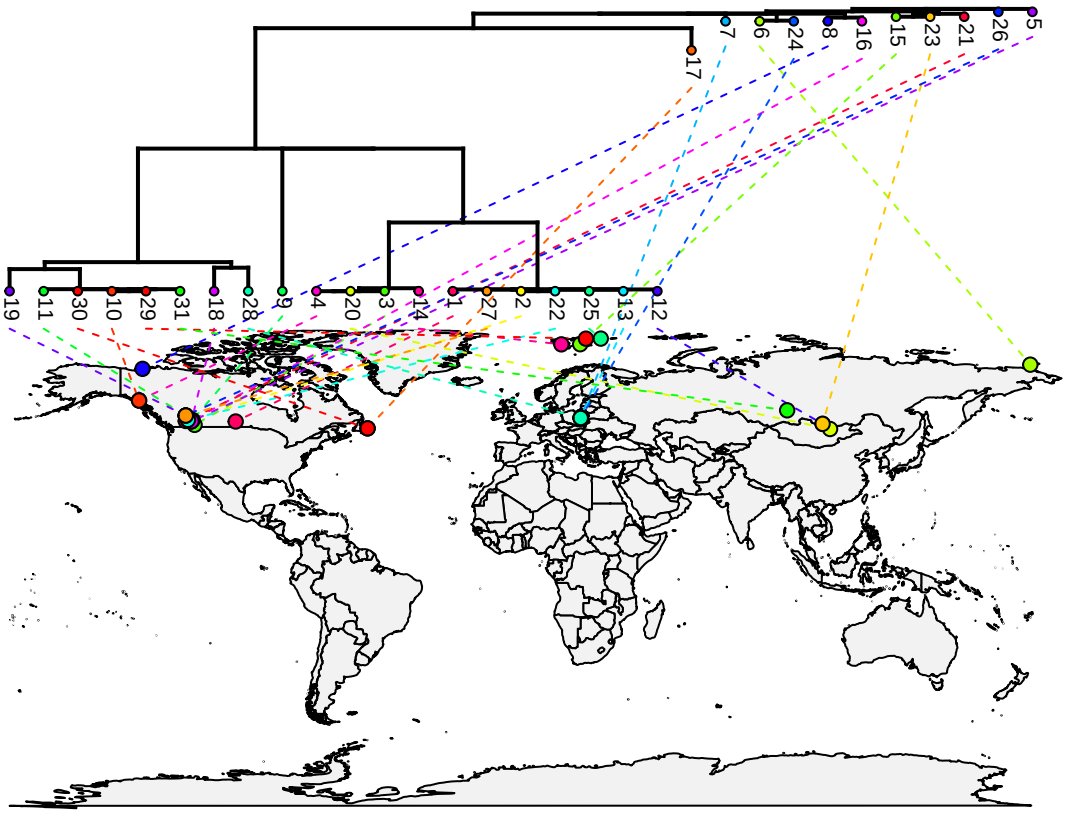



**Figure 4: Geophylogeny map for species of genus Ursus.**

```
# Clear up global environment
rm(clusters.COI, columnames, dfCOI, dfCOI.alignment, dfSpecies.count, dnaBin.COI,
   location, obj, tree, Tree, treeUltra, Phylo, Geo, distanceMatrix)
```

## Bonus Section

I initially chose the genus Sardinella for my project. The aim of the study using genus Sardinella was also to generate a dendrogram and geophylogeny for the Sardinella species. However, once the dendrogram was created, I realized some issues with using the genus for this topic. I have added the code used to generate the dendrogram for genus Sardinella below, to aid the discussion about why I changed to genus Ursus for my project.

### Obtain Sardinella data from BOLD Systems.

```
data <- read_tsv(
  file = "http://www.boldsystems.org/index.php/API_Public/combined?taxon=Sardinella&format=tsv")

# Write the data to disk.
write_tsv(data, "Sardinella_BOLD_data.tsv")

# Read in the data file and assign the data to a new data frame.
data <- read_tsv(file = "Sardinella_BOLD_data.tsv")

# Keep the missing data, sequence length variability the same as that for Ursus
# Keep the chosen model of molecular evolution, and the clustering threshold and method the same.
```

### Data Filtering

```
# Create a new data frame, select the columns to be included in the dataframe.
# Filter the nucleotide sequences present for COI-5P marker code.
# Remove Ns and gaps from the sequences.
# Remove records that lack species name, sequences or geographical data.
dfCOI <- data %>%
  select(processid, genus_name, species_name, markercode, nucleotides, country, lat, lon)%>%
  filter(!is.na(species_name)) %>%
  filter(!is.na(lat)) %>%
  filter(!is.na(lon)) %>%
  filter(!is.na(nucleotides)) %>%
  filter(markercode == "COI-5P") %>%
  mutate(nucleotides2 = str_remove_all(nucleotides, "^N+|N+$|-")) %>%
  filter(str_count(nucleotides2, "N") <= (missing.data * str_count(nucleotides))) %>%
  filter(str_count(nucleotides2) >= median(str_count(nucleotides2)) - length.var &
           str_count(nucleotides2) <= median(str_count(nucleotides2)) + length.var)

# Remove original Sardinella data
rm(data)

# Add new column with numerical identifier to show sequence number.
```

```r
dfCOI$ID <- 1:nrow(dfCOI)

# Create a unique identifier for each sequence
# Use the numerical ID of the sequence and the species name of the sample.
dfCOI$Unique_ID <- paste(dfCOI$species_name, dfCOI$ID, sep = "_")

# Change nucleotide data format for analysis with muscle package from Bioconductor.
dfCOI <- as.data.frame(dfCOI)
dfCOI$nucleotides2 <- DNAStringSet(dfCOI$nucleotides)
```

### Sequence Alignment and Clustering

```r
# Set unique ID as unique identifiers for the sequence names.
names(dfCOI$nucleotides2) <- dfCOI$Unique_ID

# Align the sequences using default settings of muscle package.
# Here gap opening penalty = 400 and maxiters = 8.
dfCOI.alignment <- DNAStringSet(muscle::muscle(dfCOI$nucleotides2))

# Write to file.
writeXStringSet(dfCOI.alignment, "dfSardinella.fasta")

# Use BrowseSeqs function from DECIPHER package to view the sequences in the browser
#BrowseSeqs(dfCOI.alignment)

# Convert data class to DNAbin (for clustering) using a function from ape package.
dnaBin.COI <- as.DNAbin(dfCOI.alignment)

# Create the distance matrix. then view some of the results to check the data.
distanceMatrix <- dist.dna(dnaBin.COI, model = chosen.model, as.matrix = TRUE,
                           pairwise.deletion = TRUE)

# Cluster using variables identified in Code-Part1. This should give us a dendrogram.
#clusters.COI <- IdClusters(distanceMatrix,
                           #method = clustering.method,
                           #cutoff = clustering.threshold,
                           #showPlot = TRUE,
                           #type = "both",
                           #verbose = TRUE)
```

```r
ggdendrogram(clusters.COI[[2]], rotate = TRUE, size = 4, theme_dendro = FALSE,
             main = "Cluster Dendrogram for Sardinella species",
             ylab = "Genetic Variability", xlab = "Specimen ID")
```

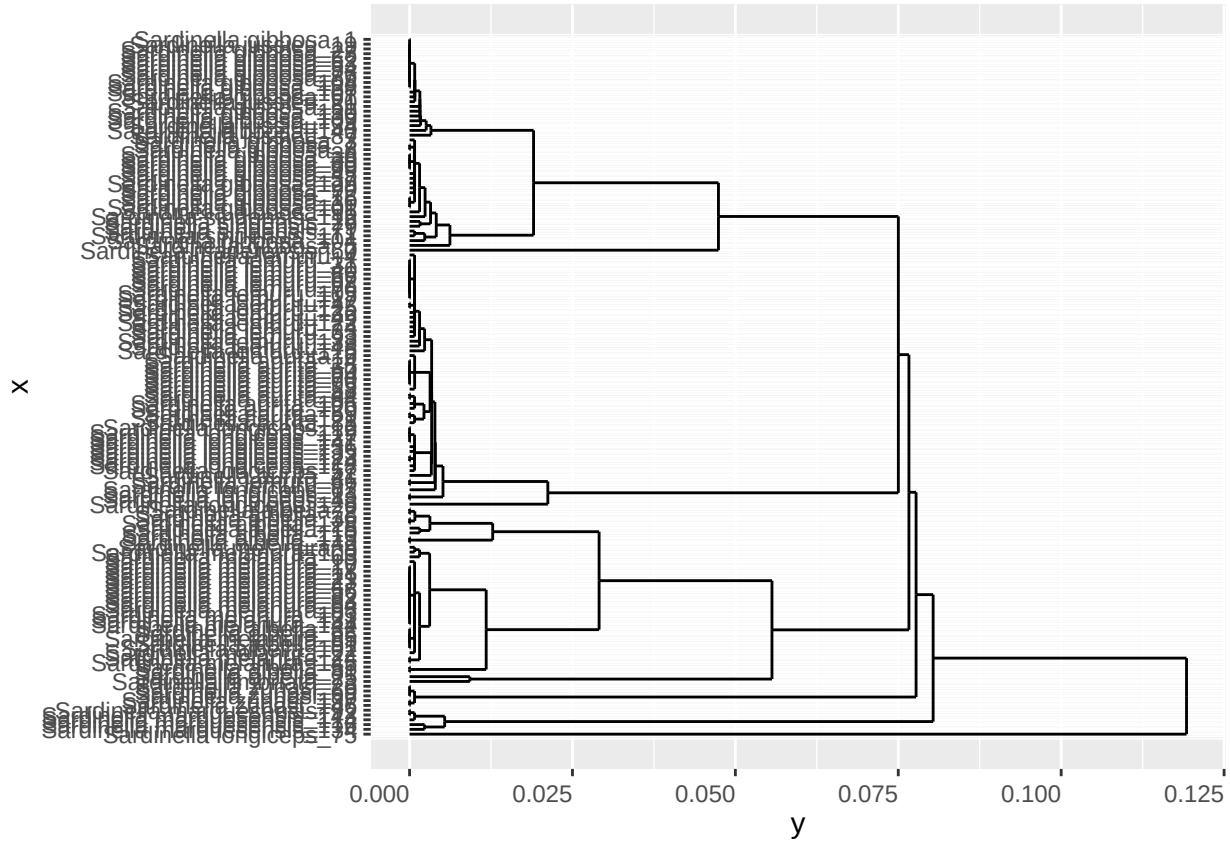



**Figure 5: Dendrogram of Clustered Aligned Sequences for Sardinella Species. x is the species names and y is the genetic variability.**

### So why not use Sardinella for geophylogeny?

As seen in the dendrogram for Sardninella species (Figure 5), there is a lot of data presented in a clumped fashion. When this phylogenetic information was used to generate a geophylogeny, there were too many data points in the map to make sense of. From the dendrogram generated in MEGAx (see page 16 of the report), it seems that many of the species share the same nodes. There are numerous factors that could cause this, and at this time I feel ill equipped to address these using R. Unlike Ursus, Sardinella are not landlocked and can intermingle in the ocean freely. There is also the case that species are difficult to properly identify as they are close in morphology (Bréchon et al.2013).After discussion with Jacqueline, I decided to study a landlocked genus that I could use to better understand phylogeny, and how to read dendrograms and geophylogenies.

## Discussion and Conclusion

This project analyzed the geophylogeny of the 4 species within the Ursus genus of the Ursidae bear family, with sequences from the COI gene of the mitochondrially encoded cytochrome c oxidase I (MT-COI) protein. Multiple sequence alignment and the resulting dendrogram were then used to generate a map illustrating the geographic distribution of Ursus species globally. Such studies are important as many species within the Ursidae family including *Ursus thibetanus* and *Ursus maritimus* are considered endangered species ( Garshelis & Steinmetz, 2016; Wiig et al., 2015). Even species such as *Ursus arctos* which are considered a least concerned species by IUCN (McLellan et al., 2017), are threatened due to habitat loss and interactions

with humans (Su et al., 2018). Many studies seek to understand phylogenetic relationships and species distribution, to tailor conservation methods and relocation schemes.

Before analysis, the data were filtered to remove records without species names, nucleotide sequences, longitudes, or latitudes. The nucleotide sequences were further filtered to remove Ns from the beginning of the sequences and to remove gaps. Sequence length was also varied by 50 nucleotides to remove the possibility of outliers. Of the 666 records obtained from BOLD, only 31 records were retained after the filtering step. This suggests that more information and collaboration need to be done to maintain complete records for this ecologically important group.

A histogram was generated (Figure 1) to display the frequency of different base pair lengths found in the COI sequence data, showing a good distribution. A bar graph was then generated with ggplot2 to show the number of each species in the analysis. *Ursus arctos*, the brown bear, had the largest number of sequences with 13 records. This could be because of the larger populations of brown bears compared to the other species found in the Northern hemisphere (McLellan et al., 2017). However, *Ursus thibetanus*, the Asian black bear had only 1 record even though it is a vulnerable species located in the forested mountains of Asia (Garshelis & Steinmetz, 2016).

Multiple sequence alignment was performed on the data and the phylogeny calculated using a distance matrix. As mentioned above, the Kimura-2-parameter genetic distance (K80) was used as the model of molecular evolution, with single-linkage clustering. The species were grouped using a 1% clustering threshold. Figure 3 shows the results of this analysis in the form of a dendrogram created with ggdendro. There is enough variation between the species that there are clear species clusters, so the COI gene appears to be a good marker for evolutionary diversification in this case. *Ursus maritimus* and *Ursus arctus* were closely related. There is an instance where 2 *Ursus arctus* specimens are in the close clustering to *Ursus maritimus* specimens. Recent research has been conducted to study hybridization in the evolutionary history of bears, including a cross between *Ursus arctus* (brown/grizzly bear) and *Ursus maritimus* (polar bear) called "grolars" (Kumar et al., 2017). Studies have indicated that as much as 8.8% of individual *Ursus arctus* genomes have *Ursus maritimus* origins (Cahill et al., 2015). It was also clear that *Ursus thibetanus* was the most distantly related to the other species having a different node.

The phylogeny, in the form of a rooted tree, was then used to map the geophylogeny of Ursus species globally. All of the specimens collected in this study were found, in the Northern hemisphere, in Canada, the USA, Poland, Norway, Mongolia, and Russia. Many of the specimens found in a cluster, were found in the same regions with a few exceptions, where specimens from a species were located in different countries. In further analyses, it would be important to note if the specimen location was that of its natural habitat or if it was collected from a specimen in captivity. Also int his study, only one marker was considered instead of using a multigene approach. Due to this there was good species coverage and enough data for the analysis. However,the study did not have a lot of genomic data and a broader view of the species tree. These differences would affect the geophylogeny and its analysis.Further research may also include analysis of a higher level of taxonomy such as Ursidae (family level), to better understand the phylogeny and evolution of bears. The research could also consider other marker genes or whole genome analyses. Such studies would help consrevation efforts of bear species worldwide.

## Acknowledgments

I would like to thank Sally and Jacqueline for the excellent instruction this semester. My skills in R have increased greatly over the past few months and will continue to do so as I review the topics covered in the course. For this project, Jacqueline was most helpful in helping me decide to change the genus of focus in order to better understand the concepts related to geophylogeny. She also helped with some troubleshooting fo the bar graph. As I was working on trouble shooting some errors with my mapping, Sally posted a helpful supplementary script to CourseLink, which made my solution search easier. Thank you very much for all the guidance this semester!

# References


1. McLellan, B.N., Proctor, M.F., Huber, D. & Michel, S. 2017. Ursus arctos (amended version of 2017 assessment). The IUCN Red List of Threatened Species 2017: e.T41688A121229971. https://dx.doi.org/10.2305/IUCN.UK.2017-3.RLTS.T41688A121229971.en.
2. Garshelis, D. & Steinmetz, R. 2016. Ursus thibetanus (errata version published in 2017). The IUCN Red List of Threatened Species 2016: e.T22824A114252336. https://dx.doi.org/10.2305/IUCN.UK.2016-3.RLTS.T22824A45034242.en.
3. Wiig, Ø., Amstrup, S., Atwood, T., Laidre, K., Lunn, N., Obbard, M., Regehr, E. & Thiemann, G. 2015. Ursus maritimus. The IUCN Red List of Threatened Species 2015: e.T22823A14871490. https://dx.doi.org/10.2305/IUCN.UK.2015-4.RLTS.T22823A14871490.en.
4. Cahill, J. A., Stirling, I., Kistler, L., Salamzade, R., Ersmark, E., Fulton, T. L., Stiller, M., Green, R. E., & Shapiro, B. (2015). Genomic evidence of geographically widespread effect of gene flow from polar bears into brown bears. Molecular ecology, 24(6), 1205–1217. https://doi.org/10.1111/mec.13038
5. Su, J., Aryal, A., Hegab, I. M., Shrestha, U. B., Coogan, S., Sathyakumar, S., Dalannast, M., Dou, Z., Suo, Y., Dabu, X., Fu, H., Wu, L., & Ji, W. (2018). Decreasing brown bear (Ursus arctos) habitat due to climate change in Central Asia and the Asian Highlands. Ecology and evolution, 8(23), 11887–11899. https://doi.org/10.1002/ece3.4645
6. Bréchon, A. L. , Coombs, S. H. Sims, D. W., Griffiths, A. M. (2013) Development of a rapid genetic technique for the identification of clupeid larvae in the Western English Channel and investigation of mislabelling in processed fish products. ICES Journal of Marine Science. Volume 70, Issue 2, Pages 399–407, https://doi.org/10.1093/icesjms/fss178
7. Kumar, V., Lammers, F., Bidon, T. et al. (2017). The evolutionary history of bears is characterized by gene flow across species. Sci Rep 7, 46487. https://doi.org/10.1038/srep46487
8. Edgar, R.C. MUSCLE: a multiple sequence alignment method with reduced time and space complexity. BMC Bioinformatics 5, 113 (2004). https://doi.org/10.1186/1471-2105-5-113
9. Kim, J., Rohlf, F.J., and Sokal, R. (1993). The Accuracy of Phylogenetic Estimation Using the Neighbor-Joining Method. Evolution 47(2), 471-486.
10. http://www.boldsystems.org/index.php/Taxbrowser_Taxonpage?taxon=URSUS&searchTax=Search+Taxonomy
11. https://www.rdocumentation.org/packages/ggdendro/versions/0.1.22
12. https://www.megasoftware.net/webhelp/walk_through_mega/constructing_trees_from_distance_data.htm
13. http://www.sthda.com/english/wiki/beautiful-dendrogram-visualizations-in-r-5-must-known-methods-unsupervised-machine-learning
14. https://rmarkdown.rstudio.com/
15. https://yihui.org/knitr/options/