An Exploratory Analysis of Scleractinia

Akaash Sidhu

0850326

BINF*6210

December 12th, 2018

1. INTRODUCTION

Reef-building corals around the world are facing increased risks of extinction due to anthropogenic impacts and climate change (Kitahara *et al*., 2010). Among these corals are stony corals or hard corals, from the order Scleractinia ((Kitahara *et al*., 2010)). These animals (referred to individually as polyps) build themselves a hard skeleton which makes up most of the framework of coral reefs ((Kitahara *et al*., 2010). They are very important for the ecosystem and for this reason this taxon was chosen for this exploratory analysis.

This exploratory analysis of Scleractinia is primarily biological. The question that we will be exploring for this analysis is: What is the phylogenetic diversity of species within Scleractinia and how are they geographically distributed? This analysis will involve reconstructing phylogenetic trees to see if there are key differences between the two main ecological groups (azooxanthellate, zooxanthellate) and plotting to see if there are trends between the two groups on how they are geographically distributed.

2. DATASET

The dataset that I chose to address my topic came from the BOLD database. I obtained the data on Monday December 3rd, 2018 and I chose the TSV format. The raw dataset has 2461 observations and 80 variables. They key variables that I am interested in include: nucleotides, bin_uri, lat, lon, species_name, country, and markercode. The dataset gets filtered twice: once for the clustering and producing a dendrogram and another time to produce the phylogenetic tree on the world map.

```
Scleractinia <-
read_tsv("http://www.boldsystems.org/index.php/API_Public/combined?taxon=Scle
ractinia&format=tsv")
#From BOLD, we will obtain all the available data on Scleractinia. As only
one dataset is being used, this exploratory analysis is only as good as the
data available from BOLD.

#write_tsv(Scleractinia, "Scleractinia_BOLD_data.tsv")
#Scleractinia <- read_tsv("Scleractinia_BOLD_data.tsv")
#If needed, there is the option of saving the database into the working
directory and reading from the saved file directly.
```

3. DATA EXPLORATION AND QUALITY CONTROL

The R packages that were used in this exploratory analysis include: readr, Biostrings, stringr, plyr, tidyverse, ape, seqRFLP, dplyr, phangorn, adegenet, ade4, stats, phytools, mapdata, msa, RSQlite, DECIPHER, rworldmap.

The first step in creating clusters and producing an exploratory dendrogram was to analyze the dataset and start the filtering process for the first portion of this analysis.

```
names(Scleractinia)
#Generates all the variable names in the dataset that can be used for further
exploration.

unique(Scleractinia$markercode)
#Generates all the unique marker codes available in the dataset. We will now
find the most frequent marker code and that will be used to filter the
dataset.

table(Scleractinia$markercode)
#From these results, COI-5P is the most frequent marker code.

Scleractinia.COI5P <- Scleractinia %>%
  filter(markercode == "COI-5P") %>%
  filter(str_detect(nucleotides, "[ACGT]"))
#This is a subset of the main dataset that includes only records with the
marker code COI-5P and nucleotide sequences.

unique(Scleractinia.COI5P$species_name)
#There are 479 unique species present in this dataset. This is too many to
work with so we are going to filter out species that don't have more than 5
records.

SpeciesCount <- table(Scleractinia.COI5P$species_name)
#This creates a table of species and the number of records available in the
dataset for this species.

SpeciesRecord <- Scleractinia.COI5P %>%
  group_by(species_name) %>% filter(n() >=5 ) %>%
  drop_na(species_name, markercode, nucleotides)
#Filters the data such that species with 5 or more records are kept in the
dataset and NA values for species_name, markercode, and nucleotides are
dropped.
```

After the data frames were filtered to include the variables of interest, I randomly sampled to get 2 sequences per species. After the random sampling was completed, a new data frame was created to select 4 variables of interest.

```
length(SpeciesRecord$nucleotides)
#There are 1026 sequences available and that is a lot to work with. Thus we
are going to randomly sample 5 sequences per species to conduct a multiple
sequence alignment.

RandomSample <- SpeciesRecord %>%
  group_by(species_name) %>%
  sample_n(2)
#A new dataframe with random sampling is created so that there are fewer
sequences to work with such that alignment is easier and visualizations are
not so cluttered.

RandomSampleDf <- RandomSample %>%
  select(species_name, nucleotides, markercode, bin_uri) %>%
  drop_na(bin_uri)
```

```
#In this dataframe, I subsetted the random sample to includes 4 variables of
interest. We are going to use BIN to annotate the dendrogram and tree so we
need drop NA in bin_uri which leaves us with 132 sequences.
```

The next step in the exploratory analysis is to convert the RandomSampleDF into a DNAStringSet to be used for a multiple sequence alignment (MSA) with MUSCLE.

```
class(RandomSampleDf$nucleotides)
#In order to perform a Multiple Sequence Alignment(MSA), we need to convert
the class into a DNAStringSet.

RandomSampleDf$nucleotides <- DNAStringSet(RandomSampleDf$nucleotides)
Scler.Sequence <- RandomSampleDf$nucleotides
class(Scler.Sequence)
#A DNAStringSet is created to be used for MSA.

Scler.MUSCLE <- DNAStringSet(muscle::muscle(Scler.Sequence, maxiters = 2,
diags = TRUE))
#Using the MUSCLE algorithm, an MSA is conducted. The maxiters is set to 2
because my laptop cannot handle more than 2 iterations as it results in the
program crashing. Therefore, in order to account for computational power only
2 iterations are used. As this is an exploratory analysis, MUSCLE was chosen
over other algorithms such as DECIPHER because of speed. MUSCLE is not only
efficient with speed but it is decently accurate therefore I picked MUSCLE.
If this was a hypothesis testing analysis, I would have gone for DECIPHER for
optimal accuracy.

mean(unlist(lapply(Scler.MUSCLE, str_count, ("-"))))
#The average number of gaps per sequence is 216.0152. There are a lot of gaps
in the sequences which could imply that it is difficult to align the
different species due to large differences at the molecular level.

names(Scler.MUSCLE) <- RandomSampleDf$bin_uri
Scler.MUSCLE
#Changes the names to BIN identifiers.

dnaBin.Scler <- as.DNAbin(Scler.MUSCLE)
#Converting the sequences for clustering into dnaBIN object.

dist.Scler <- dist.dna(dnaBin.Scler, model = "TN93")
#Create a distance matrix using the alignment from MUSCLE. The model TN93 is
chosen because it is a model of sequence evolution. The TN93 model takes into
account differences between transitions and transversions. Also, it is the
best option since the function IDClusters() from DECIPHER will be used.

dist.Scler.DF <- as.data.frame(as.matrix(dist.Scler))
#Created a dataframe of the distance matrix to better view any patterns in
distance.

table.paint(dist.Scler.DF, cleg=0, clabel.row=0.5, clabel.col=0.5)
#A visualization of the distance matrix, where darker shades of gray
represent greater distances.
```

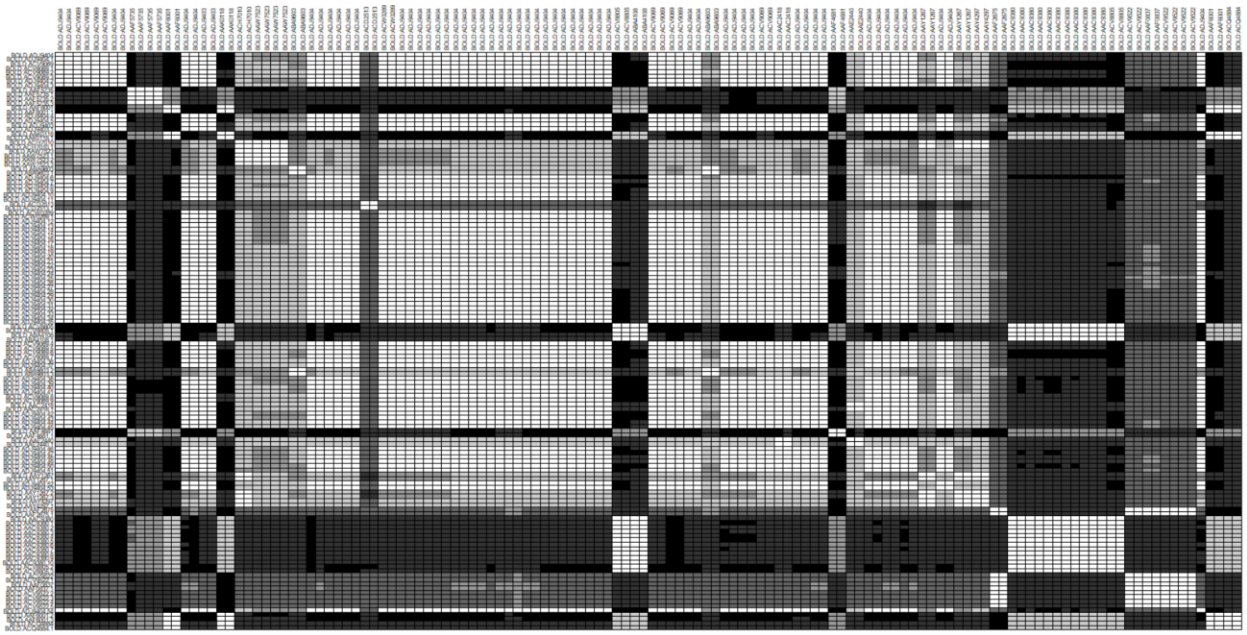

Figure 1: Distance Matrix with Corresponding BIN Identifiers. The darker shades of gray represent greater distances and from this it can imply that the azooxanthellate group of Scleractinia species may be the darker shades as they will have greater distances from the zooxanthellate, which are greater in abundance thus must be the white shades.

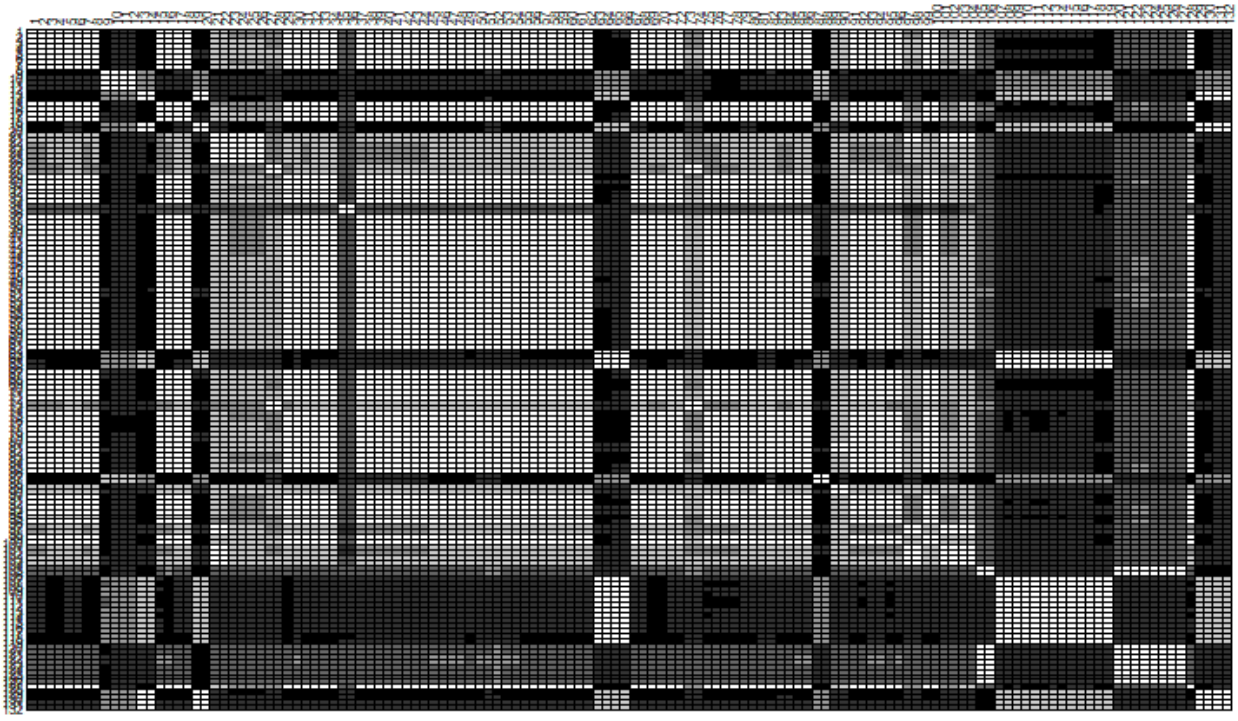

Figure 2: Distance Matrix with Corresponding Sequence #s

```
dend.Scler <- IdClusters(dist.Scler, method = "NJ", cutoff= 0.02, showPlot =
TRUE, type = "both", verbose = TRUE)
#Creates both a dendogram and clusters the sequences using the NJ method.
```

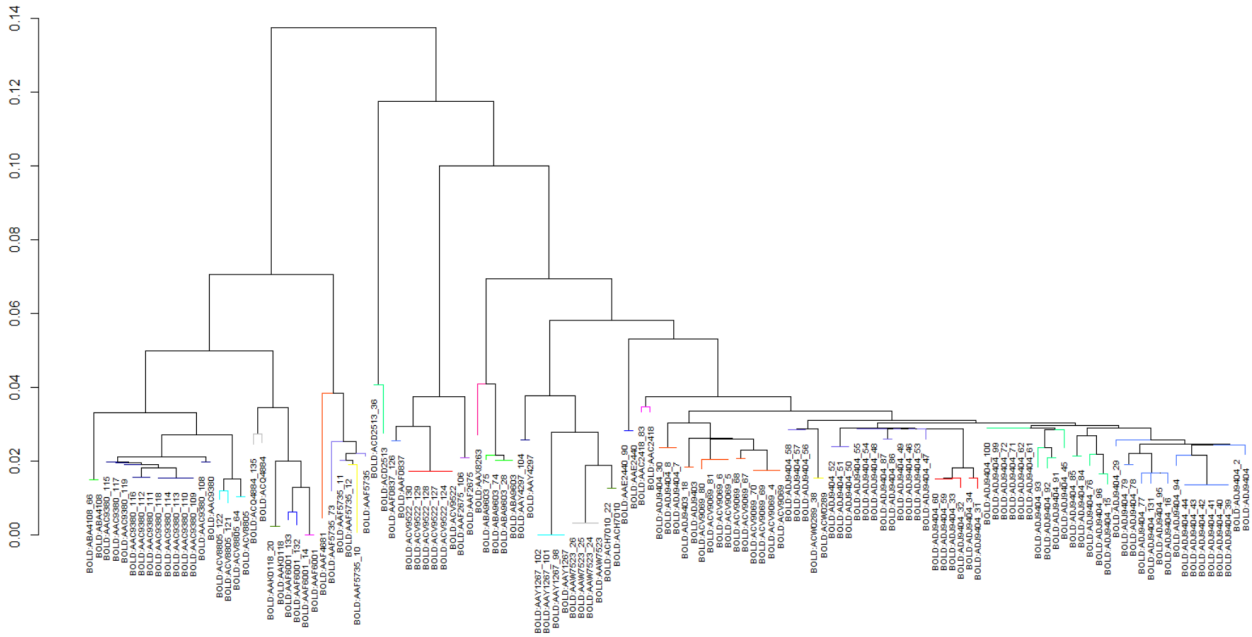

Figure 3*:* This is an unrooted dendrogram of species of Scleractinia that have been clustered by BINs. There are 2 main clades that can be observed and there are some sequences that are on their own that can be explored further with BLAST. After a BLAST analysis of these sequences that are clustered on their own, there are no outliers or mislabeled data. There are certain species of corals that are not available in abundance and as such there is limited data available which can be the reason as to why certain BINs are on their own. The first clade is likely azooxanthellate species, as there is limited data on them and they lack symbiotic dinoflagellates and the second clade is likely zooxanthellate species, which contain symbiotic dinoflagellates.

Now that the relationships between species could be seen, filtering data again was needed to include geographical parameters. A function was also created to search through a list to increase the amount of data available.

```
#Creating a search function to be used to gather additional data for
geographical distribution.
searchlist <- function(x,y){
  n <- NULL
  nn <- NULL
  for (i in x) {
    n <- grep(i,y)    #Returns the row numbers
    print(i)     #This will print the name of the search element from the list
    print(n)     #This will print which row contains the search element
    nn <- c(nn,n)  #Returns a vector of row number which corresponds to the
input search element
  }
  return(nn)
}
#The function "searchlist" searches through a list of names in a specific
variable within a dataframe (i.e. Scleractinia$bin_uri). The first arguement
is the name you want to search for and the second arguement is the specific
variable within a dataframe you want to search through.

GeoSeqData.Subset <- Scleractinia.COI5P %>%
  select(nucleotides, species_name, markercode, lat, lon, bin_uri) %>%
  drop_na(lat, lon, species_name, bin_uri)
```

```
#Creates a dataframe that includes longitudinal and latitutdinal data for
geographical distribution of species.

GeoSeqData.Subset <- GeoSeqData.Subset %>%
  arrange(species_name)
#Arrange the data by species name for the purposes of keeping the dataframe
tidy.

Unique.List <- unique(GeoSeqData.Subset$species_name)
#Create a list of unique species from filtered GeoSeqData.Subset that will be
searched through with the searchlist function.
searchlist(Unique.List, GeoSeqData.Subset$species_name)
#Using the searchlist function, the unique list of species is searched
through GeoSeqData.Subset$species_name and the returned values in the condole
are the row numbers for each species.

GeoSeqData.Subset <- GeoSeqData.Subset %>%
  arrange(species_name,lon) #
#Arranging the data again such that species with similar geographical
information are kept together in the dataframe. This helps with visualizing
the data better.

GeoSeqData.Subset.1 <- GeoSeqData.Subset
#Create a copy of GeoSeqData.Subset so that during our selection process we
still have the original to look at in case something goes wrong.

GeoSeqData.Subset.2 <- ddply(GeoSeqData.Subset.1,~lon) %>%
  group_by(species_name,lon) %>%
  sample_n(1)
#This subset will split the dataframe by longitude and randomly select
species_name and corresponding longitude numbers.

selectlist <- searchlist(GeoSeqData.Subset.2$bin_uri,
GeoSeqData.Subset$bin_uri)
#Using the searchlist function, we are going to look for more longitude data
using the BIN identifier.

Scler.Final.Subset <- GeoSeqData.Subset[selectlist,]
rownames(Scler.Final.Subset) <- c(1:741)
#Scler.Final.Subset is randomly selected data compiled according to
geographical location and species.

Scler.Final.Subset$nucleotides <-
DNAStringSet(Scler.Final.Subset$nucleotides)
#Convert to a DNAStringSet to be used for the MUSCLE analysis.

GeoScler.MSA <- DNAStringSet(muscle::muscle(Scler.Final.Subset$nucleotides,
maxiters = 2))
#Performed a MSA using MUSCLE from the muscle package. The maxiters is set to
2 because my laptop cannot handle more than 2 iterations.

names(GeoScler.MSA) <- Scler.Final.Subset$bin_uri
GeoScler.MSA
#Changed the names of the aligned sequences to BIN identifiers for
clustering.
```

4. MAIN SOFTWARE TOOL

The main software tool that was used is R Studio. The packages used to create the main visualizations include ape, rworldmap, and phytools.

Ape was chosen because it is a comprehensive package for reading, writing, plotting, and manipulating phylogenetic trees. It also has built in algorithms (i.e. NJ) that make it user friendly to create simple trees during exploration. An alternative to ape was not found since ape has the advantage to "ladderize" simple trees that I really found helpful during exploration.

Emmanuel Paradis, Simon Blomberg, Ben Bolker, Joseph Brown, Julien Claude, Hoa Sien Cuong, Richard Desper, Gilles Didier, Benoit Durand, Julien Dutheil, RJ Ewing, Olivier Gascuel, Thomas Guillerme, Christoph Heibl, Anthony Ives, Bradley Jones, Franz Krah, Daniel Lawson, Vincent Lefort, Pierre Legendre, Jim Lemon, Eric Marcon, Rosemary McCloskey, Johan Nylander, Rainer Opgen-Rhein, Andrei-Alin Popescu , Manuela Royer-Carenzi, Klaus Schliep, Korbinian Strimmer, Damien de Vienne. (2018). "Package 'ape'". https://cran.r-project.org/web/packages/ape/ape.pdf

Rworldmap was chosen because it allows for quick mapping of species and their sample sites (countries). I have used rworldmap in the past and I found it to be user friendly. The strengths of the package are that it is user friendly and allows for customization of the plot. The weakness is that every time I've used it only a certain number of species will get plotted. An alternative to this is ggplot but I have not been able to research it extensively enough to put it to use in the timeframe of this project.

Andy South. (2016) "Package 'rworldmap: Mapping Global Data'". Version 1.3-6. https://cran.r-project.org/web/packages/rworldmap/index.html

Phytools was chosen because it has a wide range of functions available for phylogenetic analysis. It was mainly used for visualizing and manipulating phylogenetic trees. This was the main tool used to project a phylogenetic tree onto a geographic map to visualize phylogenetic diversity and geographic distribution. The strengths of this package are that it allows for a lot of customizations and creates beautiful visualizations, however, it is very specific as to what it needs and can require a lot of object coercion.

Liam J. Revell. (2018) "Package 'phytools'". Version 0.6-60. https://cran.r-project.org/web/packages/phytools/phytools.pdf

5. MAIN ANALYSES

```
GeoScler.BIN <- as.DNAbin(GeoScler.MSA)
#Convert into a DNAbin object to be used to create a distance matrix for
clustering.

GeoScler.Dist <- dist.dna(GeoScler.BIN, model = "TN93", as.matrix = T,
pairwise.deletion = F)
#Calculate the distance matrix by using TN93 model.

GeoScler.Cluster <- IdClusters(GeoScler.Dist, method = "NJ", cutoff = 0.04,
showPlot = T, type = "both", verbose = T)
#Clustered the sequences using the NJ method with a cutoff of 0.04. This
produces a dendrogram that can be used for visualizations. Neighbour joining
(NJ) was used because an unrooted dendrogram was the desired output.
```

```
GeoScler.Cluster[[1]]$bin_uri <- Scler.Final.Subset$bin_uri
GeoScler.Random <-
GeoScler.Cluster[[1]][sample(nrow(GeoScler.Cluster[[1]])),]
#In this subset, the clusters of the geographical-inclusive data were
randomized.

OTU.GeoScler <- merge(GeoScler.Ranom, Scler.Final.Subset, by ="bin_uri",
x.all=TRUE)
#In this dataframe, the two dataframes are merged so that all the data is
available to pull from.

OTU.GeoScler = OTU.GeoScler[!duplicated((OTU.GeoScler$cluster)),]
#As the sequences are in clusters, a single sequence is pulled from each
cluster.

Tree.Sequences <- data.frame(OTU.GeoScler$species_name,
OTU.GeoScler$nucleotides)
dataframe2fas(Tree.Sequences, file = "GeoScler_tree.fasta")
Final.Tree <- readDNAStringSet("GeoScler_tree.fasta", format = "fasta")
#A dataframe is created with species name and nucleotides and a FASTA file is
generated and read as a DNAStringSet. This is a necessary step to plot the
phylogenetic tree on the world map.

Final.Tree.MSA <- DNAStringSet(muscle::muscle(Final.Tree, maxiters = 2, diags
= TRUE))
Dist.Tree <- dist.dna(as.DNAbin(Final.Tree.MSA), model = "TN93", as.matrix =
T, pairwise.deletion = F)
#The sequences from the file were re-aligned again using 2 iterations for the
same reason as stated before. The DNAStringSet is converted to a DNAbin
object and the TN93 model is used to create a distance matrix.

GeoScler.Phylo <- bionj(Dist.Tree)
#Using the apepackage, the bionj is an improved NJ method that is used to
create this tree.

parsimony(GeoScler.Phylo,as.phyDat(as.DNAbin(Final.Tree.MSA)))
#In order to reconstruct the phylogenies, Maximum Parsimony is used because
we are creating a simple tree. With maximum parsimony, the shortest possible
tree is created that will explain the data.

tree <- optim.parsimony(GeoScler.Phylo, as.phyDat(as.DNAbin(Final.Tree.MSA)))
#This will return a maximum parsimony score.

plot(tree,cex =0.6)
# This is a simple visualization of the phylogenetic tree. There are 3
clusters that can be seen.
```

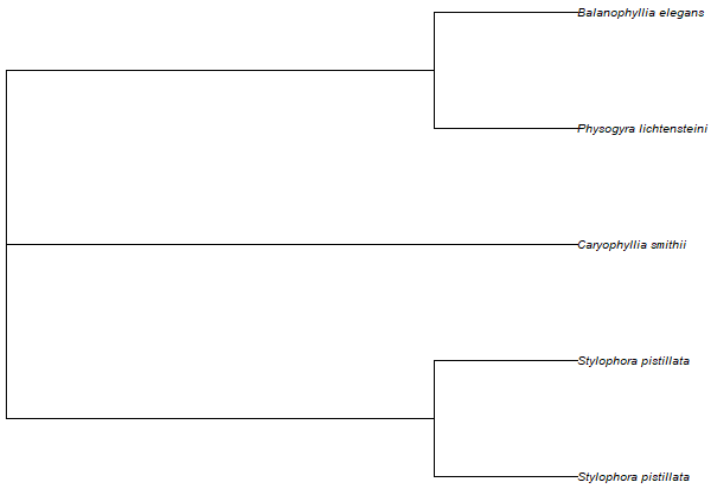

Figure 4*:* Simple tree of phylogenetic relationships derived from clustering and sampling one sequence per sample from data that includes geographical variables.

```
GeoTree <- untangle(upgma(Dist.Tree), "read.tree")
#Unfortunately, maximum parsimony would not work for the purposes of this map
so the upgma method from the ape package is used as upgma can provide a
rooted treat as a result.

GeoScler.Lon.Lat <- as.matrix(data.frame(OTU.GeoScler$lat, OTU.GeoScler$lon))
#The phylo.to.map function requires coordinates so a matrix is created that
has the latitude and longitude data.

row.names(GeoScler.Lon.Lat) <- GeoTree$tip.label
#The row names for the tree and the matrix need to be identical otherwise the
phylo.to.map function will not work.

par(oma=c(0,0,2,0))

mapplot <- phylo.to.map(GeoTree, GeoScler.Lon.Lat,plot=F)
plot(mapplot,fsize=0.02,asp=1.2, type = "phylogram", ftype="i")
title(main = "Phylogenetic Relationships of Scleractinia and their
Geographical Distribution")
#Visualization of phylogenetic relationships among Scleractinia species and
their geographical distribution on the world map.
```

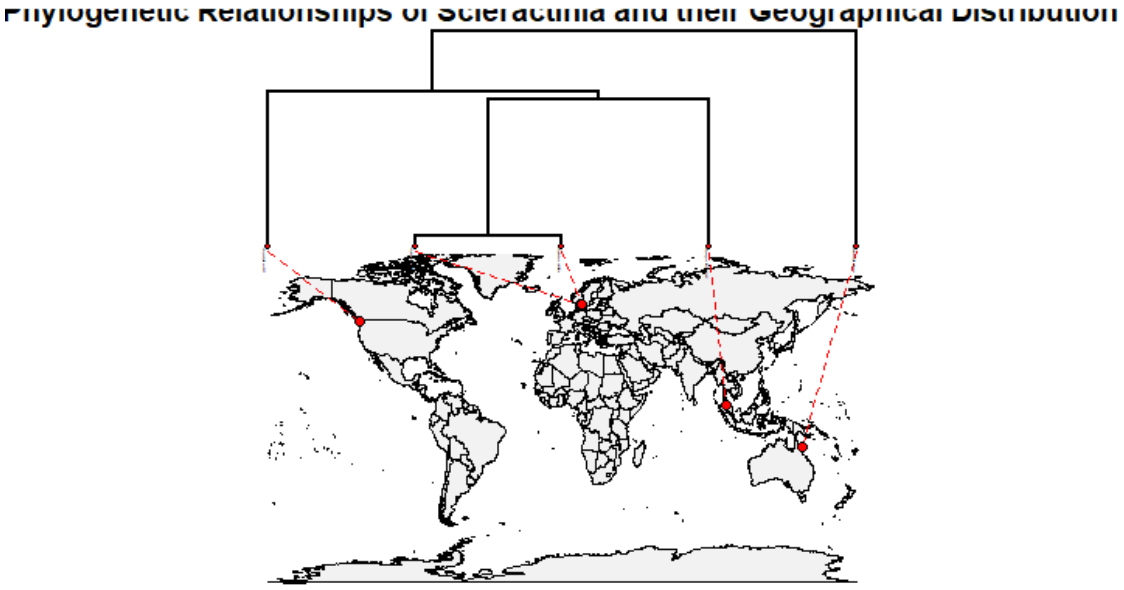

Figure 5: Phylogenetic relationships between Scleractinia species that take geographical distribution into account

```
MapScler <- Scleractinia.COI5P %>%
  group_by(species_name) %>% filter(n() >=1 ) %>%
  select(nucleotides, species_name, country, lat, lon) %>%
  drop_na(lat, lon, species_name, country) %>%
  group_by(species_name, country) %>%
  mutate(`No. of Records` = ifelse(row_number() == 1, n(), NA_integer_)) %>%
  ungroup()
#Created a dataframe where Number of Records for species data collected per
country is made available without repeats.

Na.Omit <- na.omit(MapScler, cols=c("No. of Records"))
#Omits the rows with NA so that there are only one species that is
representative per country.

match <- joinCountryData2Map(Na.Omit, joinCode = "NAME", nameCountryColumn =
"country", nameJoinColumn = "country")
#This will match the countries that there are species data available for.

testmap <- mapBubbles(dF=match, nameX = "lon", nameY = "lat", nameZSize =
"No. of Records", nameZColour = "species_name", catMethod = "categorical",
colourPalette = "rainbow", mapRegion = "world", oceanCol = "lightblue",
landCol = "wheat", addLegend = FALSE, addColourLegend = FALSE)
#Created a map that shows geographical dispersal of species. The map
exclusively shows data that includes "Species Name", "Country", "Latitude",
"Longitude" and the size of the bubbles represents the number of records
available for a specific species collected from that country.

title(main = "Geographical Dispersals of Species in Scleractinia")
#This will add a title.

do.call( addMapLegendBoxes, c(testmap,x='bottom',title="Species Name",
catMethod = "categorical", cex=0.7))
```

```
#Adds a legend to the worldmap to identify where the species have been
plotted.
```

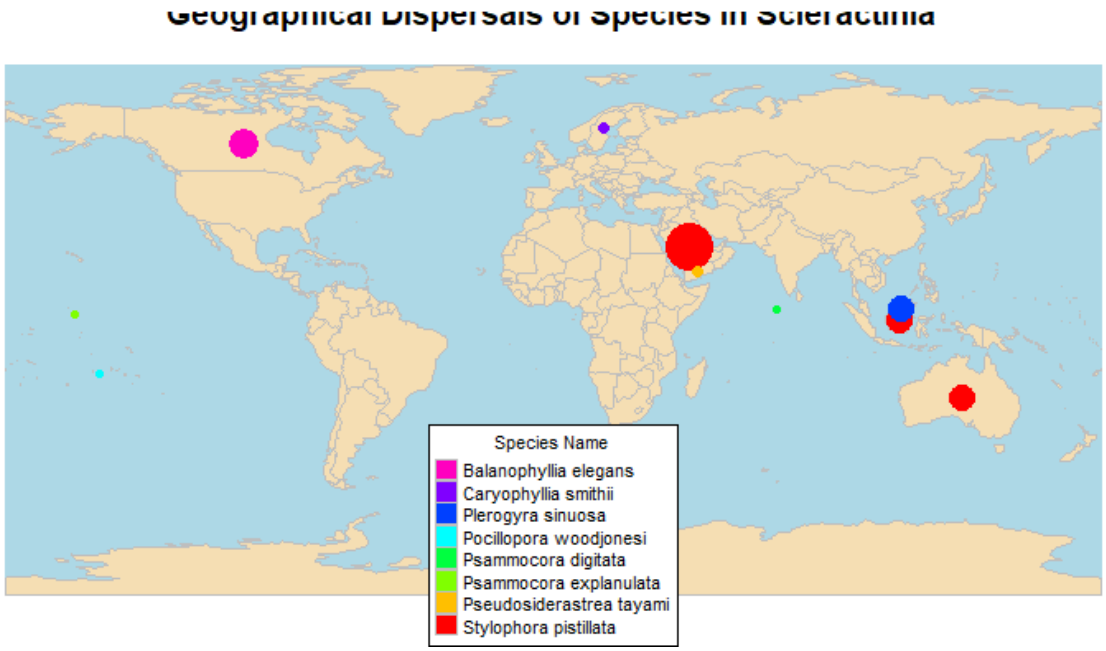

Figure 6: Geographical Dispersal of Species in Scleractinia using a Bubble Map where the larger the bubble, the greater the amount of records for that species.

6. BONUS SECTION: OPTIONAL

```
Sequence.List <- as.list(Scler.Sequence)
LargeDNAStringSet <- DNAStringSet(Sequence.List)
writeXStringSet(LargeDNAStringSet, 'Sequence.List.fasta')
#Created a FASTA file of the sequences that include numbers to indicate
species name.
writeXStringSet(Scler.MUSCLE, 'Scler.fasta')
#Created a FASTA file of the aligned sequences.

#Calculating and Visualizing Nucleotide Frequencies for Raw Sequences

Scler.Raw <- readDNAStringSet("Sequence.List.fasta","fasta")
#Read the raw sequences FASTA file to determine nucleotide frequencies.
Freq.Percentage.1 <- alphabetFrequency(Scler.Raw, as.prob = T,baseOnly=T)
#Count the number of bases (ATGC and other) per each sequence, there are 132
in total The result is a matrix where the numbers are a percentage of the
bases per each sequence.

#Calculating and Visualizing Nucleotide Frequencies for Aligned Sequences
Scler.Align <- readDNAStringSet("Scler.fasta","fasta")
#Read the aligned MSA fasta file to determine nucleotide frequencies.
Freq.Percentage <- alphabetFrequency(Scler.Align, as.prob = T,baseOnly=T)
```

```
#Count the number of bases (ATGC and other) per each sequence, there are 132
in total the result is a matrix where the numbers are a percentage of the
bases per each sequence.

#Visualize the two plots side-by-side

par(mfrow=c(1,2))
#Split the plot view into two so that both plots can be visualized next to
each other.

matplot(Freq.Percentage.1,type='l', xlab = "Sequences (n = 132)", ylab =
"Nucleotide Frequencies", main = "Nucleotide Frequencies of Raw Scleractinia
Sequences")
#Plot a line graph to represent each sequence and the percentage of
nucleotides in each sequence.
legend(legend = colnames(Freq.Percentage.1),"topright",lty=1:5,col=1:5)
#Add a legend to the plot for better assessment.

matplot(Freq.Percentage,type='l', xlab = "Sequences (n = 132)", ylab =
"Nucleotide Frequencies", main = "Nucleotide Frequencies of Aligned
Scleractinia Sequences")
#Plot a line graph to represent each sequence and the percentage of
nucleotides in each sequence.
legend(legend = colnames(Freq.Percentage),"topright",lty=1:5,col=1:5)
#Add a legend to the plot for better assessment.

par(mfrow=c(1,1))
```

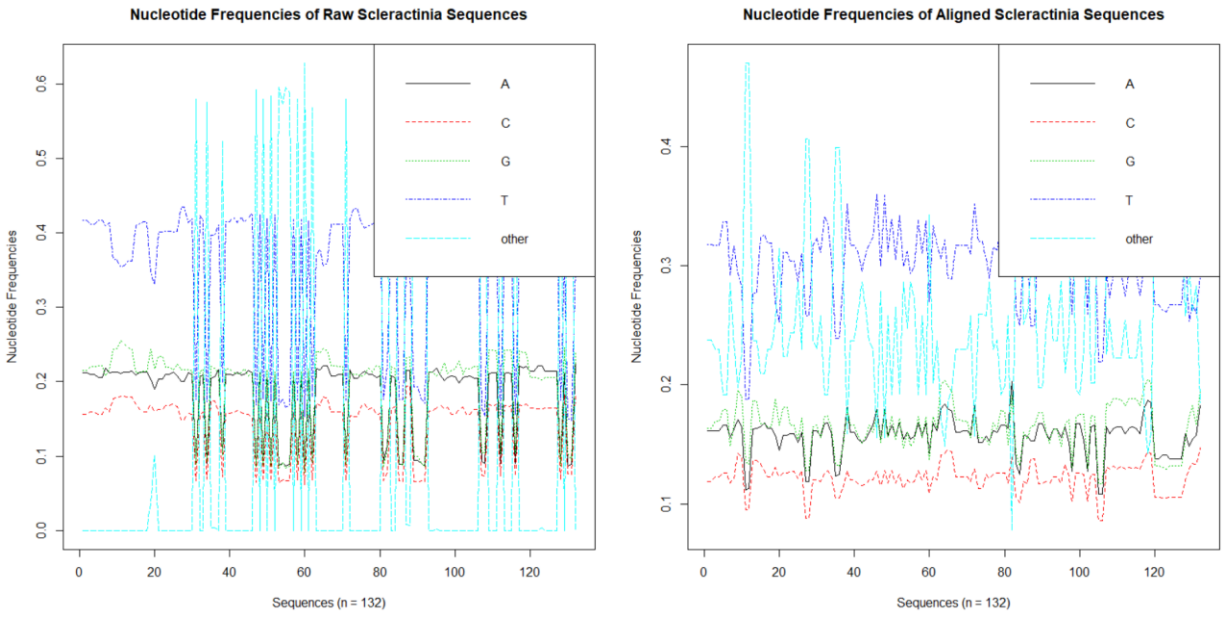

Figure 7: An additional visualization to view the raw Scleractinia sequences and the aligned Scleractinia sequences. The purpose of this figure was to view the effects of alignment on nucleotide frequencies.

7. INTERPRETATION AND DISCUSSION

In **Figure 4**, *Caryophylla smithii* does not have any other cluster sequences attached to it and this may be because it is a solitary stony coral that lives in aphotic zones. The cluster containing *Balanophyllia elegans* is an azooxanthellate species which means that it does not contain symbiotic dinoflagellates, which implies this cluster is not a reef-building cluster which is why it is clustered away from *Stylophora pistillata*. This can imply that despite being under the same order, species in Scleractinia that contain the symbiotic dinoflagellates have evolved differently to acclimate to their surroundings while those that do not contain symbiotic dinoflagellates remain hidden in aphotic zones as they cannot survive in photic zones.

In **Figure 5**, it is clear that a lot of the samples are collected from coastal regions around the world. *Balanophyllia elegans* was collected near Australia and it is also a solitary species. Since it is found on its own there is the implication of limited dispersal ability (Gerrodette, T., 1985). *Caryophylla smithii* was collected near Vancouver Canada, where it is likely it is found very deep as it inhabits aphotic zones. It is solitary and due to this may also have limited dispersal ability which may play a part in the limited evolution of this genus (Kitahara *et al*., 2010). *Physogyra lichtensteini* is found off the coast of Singapore and is an endangered species and it is unknown of whether it is reef-building or not (Kitahara *et al*., 2010). *Stylophora pistillata* is a widespread species of coral that is reef-building. Here, as only one sequence per species is taken it cannot be seen how widespread it is but, in the data sets it is clear that it is the most abundant geographically as it has been the most sampled. Overall, it seems like there is a lack of data available on geographical distribution of Scleractinia, which may be because many species are endangered and thus geographical data is being lost.

Further studies with geographical distribution and phylogenetic diversity should look into compiling data from multiple databases to increase the number of data available to visualize and analyze.

In **Figure 6**, it is clear that *Stylophora pistillata* is the most wide spread of the Scleractinia species due to the multiple bubbles present on the map and the size of those bubbles. As the size of the bubbles represent the amount of records it is also evident that there is a large variation in what species of Scleractinia are being studied.

Overall, phylogenetic diversity of Scleractinia species depends on a variety of factors, such as the presence of symbiotic dinoflagellates, and during exploration it is clear that there are differences at the molecular level due to the presence of symbiotic dinoflagellates since the 2 clades in **Figure 3** branch off into the two main ecological groups: zooxanthellate that live symbiotically with dinoflagellates in shallow tropical waters and azooxanthellate that are associated with aphotic zones. This can also imply that the presence of symbiotic dinoflagellates allows Scleractinia species to live in shallow waters as those that do not have symbiotic dinoflagellates live in colder and deeper waters (Kitahara *et al*., 2010).

The limitations of this study is the lack of information regarding deep-sea Scleractinia species which can have an effect on the results and geographical distribution (Kitahara *et al*., 2010). Primarily, work on Scleractinia focuses on reef-building corals which is important but that is biased. Therefore, the database used can introduce a form of bias and is a limiting factor for this exploratory analysis. If a hypothesis testing analysis was conducted, data from multiple sources should be compiled so that an equal number of

zooxanthellate and azooxanthellate are available to recreate phylogenetic relationships and assess their geographical distribution and its trends.

If I had more time to continue to research this topic, I would find a coral specific database that has a decent amount of information on azooxanthellate Scleractinia species along with geographical data to further look into the differences between the two groups and to see if there are geographic patterns that could related to the presence (or lack of) symbiotic dinoflagellates.

References

Gerrodette, T. (1981). Dispersal of the solitary coral Balanophyllia elegans by demersal planular larvae. *Ecology*, *62*(3), 611-619.

Kitahara, M. V., Cairns, S. D., Stolarski, J., Blair, D., & Miller, D. J. (2010). A comprehensive phylogenetic analysis of the Scleractinia (Cnidaria, Anthozoa) based on mitochondrial CO1 sequence data. *PloS one*, *5*(7), e11490.