# BINF6210 - Assignment 5

Braden Judson

December 10th, 2019

# 1. Introduction

Many patterns of biological diversity can be explained with the hypothesis of isolation-by-distance (IBD) (Wright 1942; Weber et al. 2016). IBD posits that more geographically distant populations are more distinct from one another than geographically nearby populations, as distance acts as an isolating mechanism that prevents gene flow and thus promotes differentiation (Wright 1942). However, some populations undergo sympatric differentiation and thus their divergence is not a product of geographic isolation (Bolnick and Fitzpatrick 2007). As a result, the diversity of many species' is the result of both allopatric and sympatric divergence due to a suite of ecological/evolutionary mechanisms (Bradbury et al. 2010).

Here, I assess the relative contribution of IBD on genetic differentiation using the three-spined stickleback (*Gasteroesteus aculeatus*) as a model organism. The three-spined stickleback is a small and ubiquitous fish found throughout the Northern hemisphere. As this fish is abundant, easy to sample, and diverse, it has served as a model organism for the field of population and evolutionary genetics (Bell and Foster 1994). Consequently, the barcode of life database (BOLD (http://www.boldsystems.org/index.php)) has over 200 wild stickleback samples catalogued that include gene sequence data, sampling location and more. Therefore I have chosen to use the BOLD system to investigate the relationship between genetic and geographic distance of fish, using the three-spined stickleback as a model to address the following hypothesis:

I hypothesize that geographic distance between stickleback populations is a significant contributor to genetic differentiation. Thus, I predict that samples from distant countries will exhibit a greater degree of genetic differentiation from one another than samples from geographically nearby countries. However, if geographic distance is an insignificant predictor of genetic distance, I expect to observe discordance between pairwise measures of genetic and geographic distance. This is primarily a biological question that explicitly test hypotheses, with elements of data exploration throughout.

# Libraries

```
library(tidyverse)
library(ggplot2)
library(adegenet)
library(apex)
library(ape)
library(Biostrings)
library(ggpmisc)
library(ape)
library(RSQLite)
library(DECIPHER)
library(geodist)
library(gdata)
library(graphics)
library(seqinr)
library(ade4)
library(qgraph)
library(ggspatial)
library(rnaturalearth)
library(rnaturalearthdata)
library(mmod)
library(outliers)
library(scales)
```

# Theme

Making a custom theme for future ggplot work.

```
Custom_Theme <- theme_bw() + theme(panel.background = element_rect(fill =  "white", colour = "black", line
type = "solid"), panel.grid.minor = element_line(colour = "white"), plot.caption = element_text(hjust = 0,
vjust = 1, size = 12))
```

# 2. Description of Data Set

```
#SticklebackBOLD <- read_tsv("http://www.boldsystems.org/index.php/API_Public/combined?taxon=Gasterosteus&
format=tsv") #Read data in from BOLD.
#write_tsv(SticklebackBOLD, "Stickleback.tsv") #Write to harddrive.
```

The steps above are my initial data acquisition steps to write the information to my local directory. The BOLD database was accessed November 25th, 2019.

BOLD is a public data repository that can be used to download data subsets as a .tsv format, which are friendly to manipulation through R.

```
Stickleback <- read_tsv("Stickleback.tsv") #Read .tsv into R from harddrive.
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   recordID = col_double(),
##   collection_code = col_logical(),
##   phylum_taxID = col_double(),
##   class_taxID = col_double(),
##   order_taxID = col_double(),
##   family_taxID = col_double(),
##   subfamily_taxID = col_logical(),
##   subfamily_name = col_logical(),
##   genus_taxID = col_double(),
##   species_taxID = col_double(),
##   subspecies_taxID = col_double(),
##   tax_note = col_logical(),
##   collection_event_id = col_logical(),
##   collectiondate_start = col_logical(),
##   collectiondate_end = col_logical(),
##   collectiontime = col_logical(),
##   sex = col_logical(),
##   habitat = col_logical(),
##   associated_specimens = col_logical(),
##   associated_taxa = col_logical()
##   # ... with 8 more columns
## )
```

```
## See spec(...) for full column specifications.
```

```
class(Stickleback) #Close, but we want a dataframe.
```

```
## [1] "spec_tbl_df" "tbl_df"      "tbl"         "data.frame"
```

```
Stickleback <- as.data.frame(Stickleback) #Convert to dataframe.
Stickleback[1:5, 1:10] #View a subsample of the dataframe to make sure formatting is consitent enough to w
ork with.
```

```
##       processid   sampleid recordID catalognum   fieldnum      institution_storing
## 1 ANGBF42583-19   KR477041 10140941       <NA>       <NA> Mined from GenBank, NCBI
## 2 ANGBF42584-19   KU867885 10140942       <NA>       <NA> Mined from GenBank, NCBI
## 3 ANGBF54082-19   MH205729 10852167       <NA>       <NA> Mined from GenBank, NCBI
## 4 ANGBF57310-19  NC_041244 10855395       <NA>       <NA> Mined from GenBank, NCBI
## 5    BCFB065-06 BCF-0134-1   235323  ROM-80264 BCF-0134-1     Royal Ontario Museum
##   collection_code      bin_uri phylum_taxID phylum_name
## 1              NA         <NA>           18    Chordata
## 2              NA BOLD:AAA8488           18    Chordata
## 3              NA         <NA>           18    Chordata
## 4              NA         <NA>           18    Chordata
## 5              NA BOLD:AAA8488           18    Chordata
```

```
unique(colnames(Stickleback)) #View variables in dataframe.
```

```
##  [1] "processid"                  "sampleid"
##  [3] "recordID"                   "catalognum"
##  [5] "fieldnum"                   "institution_storing"
##  [7] "collection_code"            "bin_uri"
##  [9] "phylum_taxID"               "phylum_name"
## [11] "class_taxID"                "class_name"
## [13] "order_taxID"                "order_name"
## [15] "family_taxID"               "family_name"
## [17] "subfamily_taxID"            "subfamily_name"
## [19] "genus_taxID"                "genus_name"
## [21] "species_taxID"              "species_name"
## [23] "subspecies_taxID"           "subspecies_name"
## [25] "identification_provided_by" "identification_method"
## [27] "identification_reference"   "tax_note"
## [29] "voucher_status"             "tissue_type"
## [31] "collection_event_id"        "collectors"
## [33] "collectiondate_start"       "collectiondate_end"
## [35] "collectiontime"             "collection_note"
## [37] "site_code"                  "sampling_protocol"
## [39] "lifestage"                  "sex"
## [41] "reproduction"               "habitat"
## [43] "associated_specimens"       "associated_taxa"
## [45] "extrainfo"                  "notes"
## [47] "lat"                        "lon"
## [49] "coord_source"               "coord_accuracy"
## [51] "elev"                       "depth"
## [53] "elev_accuracy"              "depth_accuracy"
## [55] "country"                    "province_state"
## [57] "region"                     "sector"
## [59] "exactsite"                  "image_ids"
## [61] "image_urls"                 "media_descriptors"
## [63] "captions"                   "copyright_holders"
## [65] "copyright_years"            "copyright_licenses"
## [67] "copyright_institutions"     "photographers"
## [69] "sequenceID"                 "markercode"
## [71] "genbank_accession"          "nucleotides"
## [73] "trace_ids"                  "trace_names"
## [75] "trace_links"                "run_dates"
## [77] "sequencing_centers"         "directions"
## [79] "seq_primers"                "marker_codes"
```

The dataset I have imported into R consists of 398 samples, of 80 variables each. However, many of the variables do not contain any data for the samples I have retrieved. See section 3 (below) for a more in-depth description of the raw data set.

# 3. Data Exploration and Quality Control

This section undertakes preliminary data filtering and various aspects of quality control. However, more technical analyses (i.e. population genetics) require additional data manipulation and filtering steps.

Only include the species of interest (three-spined stickleback), and remove samples without geographic coordinates. As my biological question has a large geographic component, samples from unknown areas are not relevant to this project and are thus removed.

```
Stickleback <- Stickleback %>%
  filter(species_name == 'Gasterosteus aculeatus',
         !is.na(lat | lon))
#Unedited, raw dataframe is 398 rows and 80 volumns (from tibble header).

unique(Stickleback$species_name) #Ensure that you only have samples from species of interest.
```

```
## [1] "Gasterosteus aculeatus"
```

```
dim(Stickleback) #New, filtered dataframe is 126 entries of 80 variables each.
```

```
## [1] 126  80
```

```
sum((is.na(Stickleback$lat)) + (is.na(Stickleback$lon))) #Sum = 0. No missing geographic data.
```

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

Determine genetic marker to be used. Here, the only marker in the dataset is the gene COI-5P. This is a mitochondrial gene that codes for the protein cytochrome c oxidase I subunit, which is associated with the electron transport chain (i.e. metabolism and ATP production).

```
unique(Stickleback$markercode) #Turns out we only have one marker: COI-5P.
```

```
## [1] "COI-5P"
```

```
Stickleback$nucleotides <- as.character(Stickleback$nucleotides) #Convert to character.
ggplot(data = Stickleback, aes((nchar(Stickleback$nucleotides)))) +
  Custom_Theme +
  geom_histogram(col="black", fill="gray") + #Add a black outline and gray filler to plot.
  labs(x="Gene Length (Nucleotides)", y="Frequency") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
  labs(caption = "Figure 1: COI-5P gene length frequencies among 126 samples of three-spined\nstickleback
 (pre-filtering).") #Histogram reveals most sequences are between 600 and 700 nucleotides, but there is a
 single outlier ~1100bp that I will remove.
```

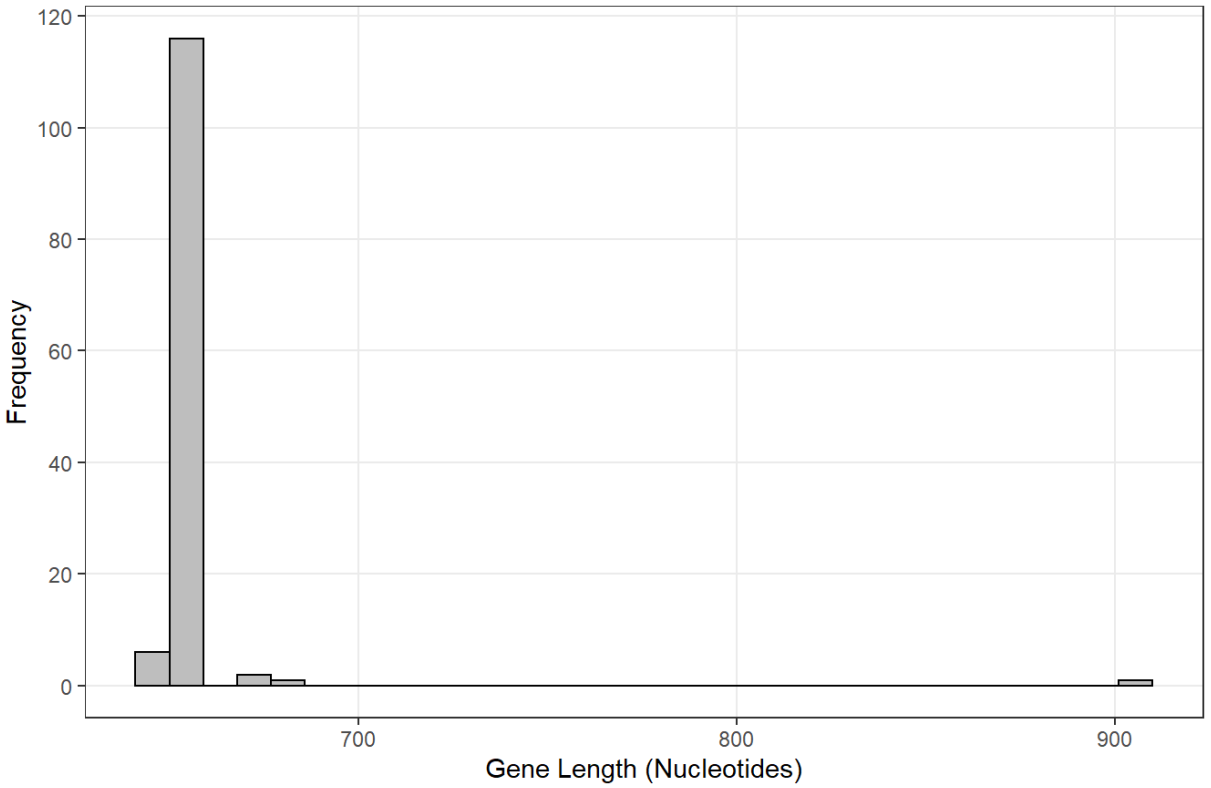


Figure 1: COI-5P gene length frequencies among 126 samples of three-spined stickleback (pre-filtering).

I suspect that the 905 nucleotide gene sample may be an outlier.

```
chisq.out.test(x = nchar(Stickleback$nucleotides), variance = var(nchar(Stickleback$nucleotides)))
```

```
##
##  chi-squared test for outlier
##
## data:  nchar(Stickleback$nucleotides)
## X-squared = 118.82, p-value < 2.2e-16
## alternative hypothesis: highest value 905 is an outlier
```

The above Chi-squared test suggests that the 905 nucleotide gene sample is an outlier ($p < 2.2 \times 10^{-16}$). Therefore, I remove the outlier and plot the new, filtered data set. The filtered data set has a mean COI-5P length of 654 +/- 4.7 nucleotides (+/- 1 SD).

```
Stickleback <- Stickleback %>%
  filter((nchar(Stickleback$nucleotides)) < 800) #Remove outlier.
dim(Stickleback) #Can see that we have one less row (now 125)- which is expected.
```

```
## [1] 125  80
```

```
rownames(Stickleback) <- Stickleback$processid

ggplot(data = Stickleback, aes((nchar(Stickleback$nucleotides)))) +
  Custom_Theme +
  geom_histogram(col="black", fill="gray") + #Add a black outline and gray filler to plot.
  labs(x="Gene Length (Nucleotides)", y="Frequency") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 5)) +
  labs(caption = "Figure 2: COI-5P gene length frequencies among 125 samples of three-spined\nstickleback
 (post-filtering).")
```

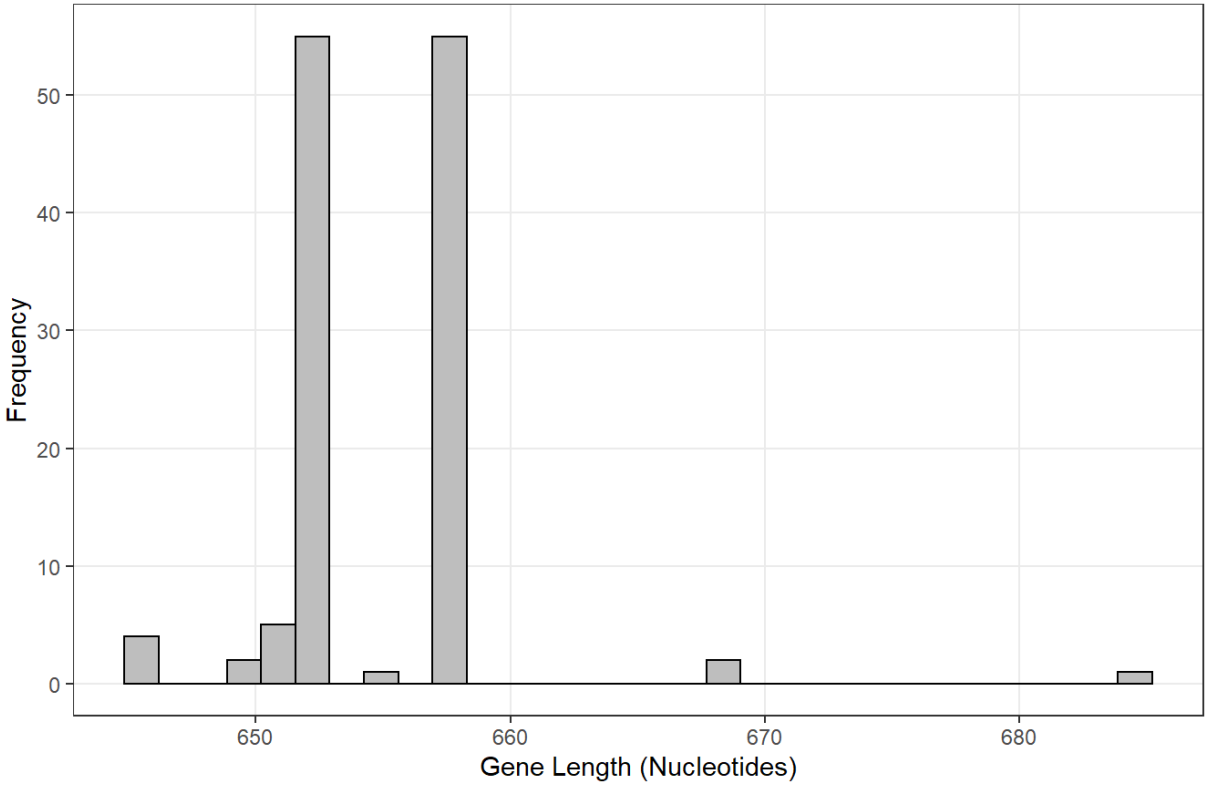


Figure 2: COI-5P gene length frequencies among 125 samples of three-spined stickleback (post-filtering).

```
#Add axis labels and a title.
#The histogram appears more normalized with the outlier removal (although it is right-skewed).

mean(nchar(Stickleback$nucleotides)) #654.8
```

```
## [1] 654.872
```

```
sd(nchar(Stickleback$nucleotides)) #4.7
```

```
## [1] 4.673149
```

```
max(nchar(Stickleback$nucleotides)) #684 - this will come in handy later.
```

```
## [1] 684
```

```
#Some preliminary statistics about COI-5P length distribution.
```

The above distribution of COI-5P length is expected. For instance, the nine-spined stickleback (*Pungitius pungitius*) has a mean COI-5P length of 627 nucleotides and the Whitefly (*Bemisia tabaci*) has a mean COI-5P gene length of 657 nucleotides (Denys et al. 2018; Dinsdale et al. 2010, respectively). As COI-5P exhibits significant between-species differentiation, these differences are expected and suggest that the mean gene length observed in this study is reasonable.

To make comparisons between sequences, they need to be aligned and the genetic distance between them needs to be estimated. Here, I use the TN93 genetic distance algorithm as it accounts for the differential rate of transverse and transitional mutations, and it does not assume equal base frequencies (Tamura and Nei 1993). These steps outlined below will be further "cleaned" and trimmed later using an external software, MEGA.

```
Stickleback$nucleotides <- DNAStringSet(Stickleback$nucleotides) %>% #Convert to DNAStringSet.
  na.omit() #Remove NAs.
class(Stickleback$nucleotides) #DNAStringSet - as expected.
```

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

```
SticklebackCOIAlignment <- DNAStringSet(muscle::muscle(Stickleback$nucleotides, quiet = TRUE), use.names =
T) #Perform a multiple sequence alignment using the package 'muscle'.
class(SticklebackCOIAlignment) #DNAStringSet - as expected.
```

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

```
BrowseSeqs(SticklebackCOIAlignment) #Visualize the initial alignment. Notably, there is a sequence (#1) th
at is longer than the rest by ~30bp (~15 on each end). Also, there are two sequences that are of poorer qu
ality on the 5' end. I will address these concerns through the software MEGA.
```

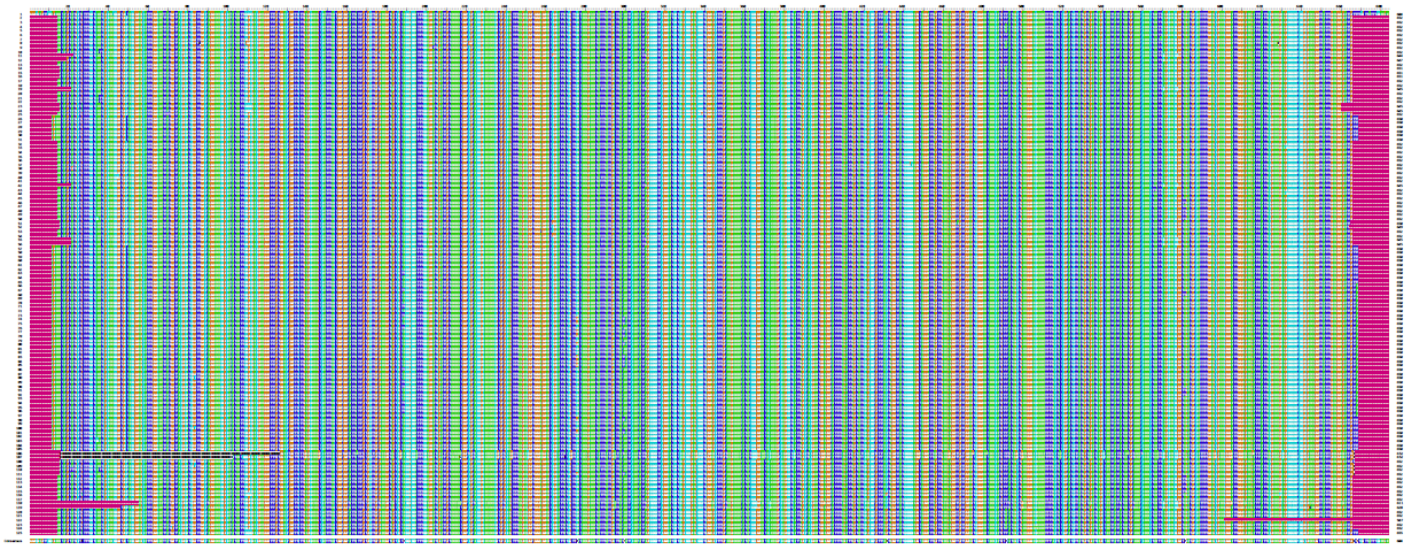

Figure 3: Alignment of raw (non-trimmed) COI-5P sequences among 125 samples of three-spined stickleback.

```
#To trim these sequences in MEGA, I need to convert them to .fas files.
Stickleback$nucleotides <- as.list(Stickleback$nucleotides)
seqinr::write.fasta(sequences = Stickleback$nucleotides, names = Stickleback$processid, open = "w", file.o
ut = "Fish.fas", nbchar = 700)
#Write COI-5P sequences to FASTA format (.fas) with headers as processIDs. The longest sequence in this da
ta set is ~685 nucleotides, so setting nbchar to 700 ensures that all data will be included in the .fas fi
le.
```

# 4. Main Software Tool Description

The main external (i.e. non-R) software I used for this project was MEGA (v 10.0.5, from: https://www.megasoftware.net/ (https://www.megasoftware.net/)) (Kumar et al. 2018). MEGA (Molecular Evolutionary Genetics Analysis) is a software that facilitates the manipulation and analysis of molecular data through a friendly, graphical user interface (GUI). MEGA can be used to build phylogenetic relationships, conduct population genetic analyses, and manipulate raw sequence data. For this project, MEGA offered an easy-to-use interface that allowed me to align and trim 123 sequences and work with the output easily in R. Steps are outlined below:

Through MEGA, I open 'Fish.fas' and perform a muscle alignment with default parameters (-400 gap open score, 16 iterations, UPGMA cluster method). As the COI-5P sequences I am working with have no internal gaps, modifying the gap score (tried both -200 and -600) did not alter the alignment at all.

First, I zoom out and view the alignment while scanning for poor quality sequences. I notice that GERFW112-13 and GERF113-13 have a large number of N's (>2% of total sequence length), thus indicating poor sequence quality (especially on the 5' end). In contrast, all other sequences have very few N's (1 at most). Thus, I manually remove GERFW112-13 and GERF113-13 from the dataset. Then, I trim the first (most 5') 55 nucleotides and the most terminal (most 3') 82 nucleotides. This step ensures that all sequences are as long as possible, without incorporating blank sections in the alignment. The resulting trimmed alignment is 546 nucleotides in length.

Consequently, I need to correct my original "Stickleback" dataframe to remove the two samples I eliminated using MEGA.

```
RemoveRowNames <- c("GERFW112-13", "GERFW113-13")
Stickleback <- Stickleback[!(row.names(Stickleback) %in% RemoveRowNames), ]
dim(Stickleback) #123 rows, as expected (125 - 2 = 123).
```

```
## [1] 123  80
```

In the above case, I performed a default muscle alignmnet on the stickleback COI-5P sequence data. I performed the muscle alignment as it globally aligns the sequences against each other while accounting for indels and the relative likelihood of all nucleotide mutation combinations. As the muscle alignment methodology uses k-mer frequencies, the algorithm is more accurate and faster than most other sequence alignment tools (Edgar 2004).

```
SticklebackCOI5P <- apex::read.multiFASTA(c("SticklebackMEGA_trimmed.fas")) #Read in file as "multidna" ob
ject.
class(SticklebackCOI5P) #Confirm formatting
```

```
## [1] "multidna"
## attr(,"package")
## [1] "apex"
```

```
plot(SticklebackCOI5P, cex = 0.2, main = NULL, sub = "Figure 4. COI-5P nucleotide proportions among 123 sa
mples of three-spined stickleback.", mar = c(5, 1, 1, 1))
```

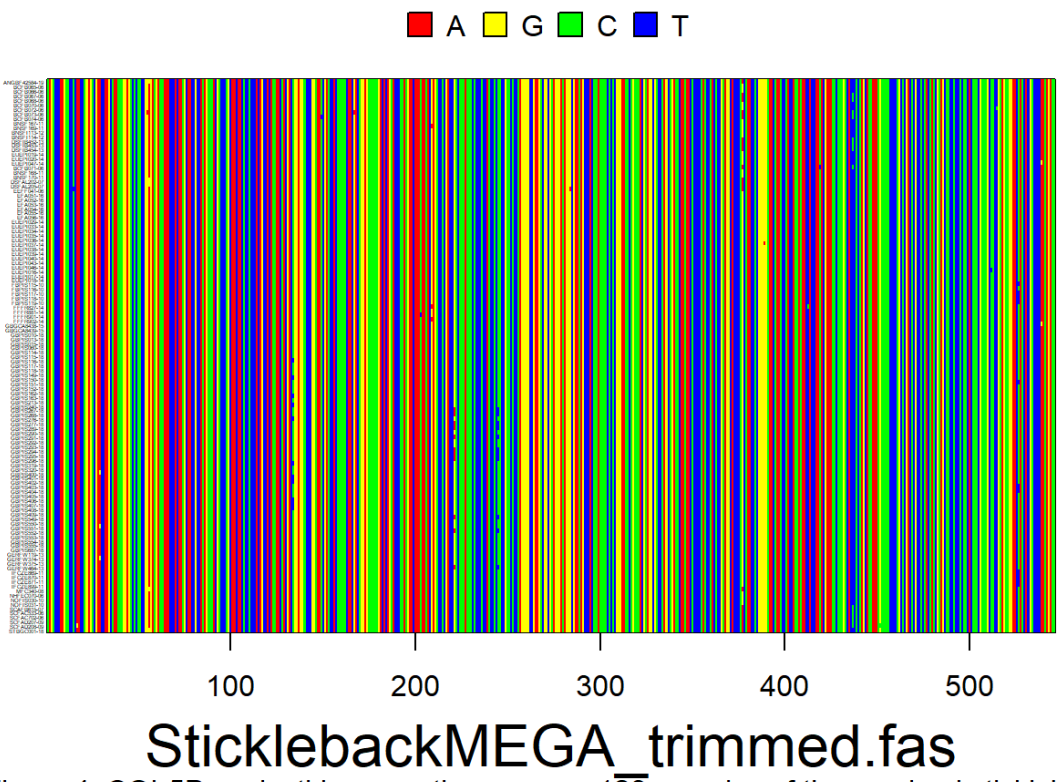

Figure 4. COI-5P nucleotide proportions among 123 samples of three-spined stickleback.

```
#Visualize the trimmed and filtered alignment.
```

The above alignment (Figure 3) has a conserved sequence composition among samples. Additionally, base frequencies appear somewhat uniform in their distribution among COI-5P.

```
getLocusNames(SticklebackCOI5P)
```

```
## [1] "SticklebackMEGA_trimmed.fas"
```

```
(setLocusNames(SticklebackCOI5P) <- gsub(".fas", "", getLocusNames(SticklebackCOI5P))) #Gets rid of the .f
as portion of the object name to make it easier to work with.
```

```
## [1] "SticklebackMEGA_trimmed"
```

Read in the trimmed and filtered .fasta file as as a DNAbin to assess the relative and pairwise genetic distance between samples.

```
SticklebackDNABin <- read.FASTA("SticklebackMEGA_trimmed.fas")
class(SticklebackDNABin) #"DNAbin"
```

```
## [1] "DNAbin"
```

Here, I calculate the pairwise genetic distance between samples using the TN93 (Tamura and Nei 1993) algorithm.

```
SticklebackGenDist <- ape::dist.dna(SticklebackDNABin, pairwise.deletion = TRUE, as.matrix = TRUE, model =
"TN93")
class(SticklebackGenDist) #Matrix - as expected. Need to label rows/columns as unique sample processIDs.
```

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

```
SticklebackID <- as.character(Stickleback$processid) #To give column/row names meaning, I want to assign t
hem to the processIDs of each sample.
rownames(SticklebackGenDist) <- SticklebackID
colnames(SticklebackGenDist) <- SticklebackID
SticklebackGenDist <- as.data.frame(SticklebackGenDist) #More friendly for downstream analysis.
```

# 5a. Main Analyses I: Geographic Data Analysis

Among my 123 samples, I will calculate the linear geographic distance between each of them. I will then create a pairwise distance matrix between all samples (123x123 = 15129 data points).

Note: Some fish are sampled from the same lake/estuary and thus have a pairwise geographic distance of 0.00 between them. Also, output distances are in meters.

```
#Calculate the distance between all samples.
SticklebackCoord <- Stickleback[ , 47:48] #Isolating geographic coordinates.
SticklebackCoord <- as.data.frame(SticklebackCoord) #Keep geographic coordinates as a dataframe.
head(SticklebackCoord)
```

```
##                  lat      lon
## ANGBF42584-19 61.031 -152.135
## BCFB065-06    48.117  -70.267
## BCFB066-06    48.117  -70.267
## BCFB067-06    48.117  -70.267
## BCFB068-06    48.117  -70.267
## BCFB070-06    46.773  -71.355
```

The function “geodist” uses differences in latitude and longitude to calculate the linear geographic distance (i.e. “geodesic distance”) between points (in meters). Here, I use this function to calculate the pairwise geographic distance between all stickleback samples.

```
Stickleback_Dist <- geodist::geodist(x = SticklebackCoord, sequential = FALSE, measure = "geodesic")
class(Stickleback_Dist) #matrix - as expected.
```

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

```
dim(Stickleback_Dist) #123x123 - as expected for pairwise comparisons of 123 samples.
```

```
## [1] 123 123
```

```
Stickleback_Dist[c(1:6), c(1:6)] #Visualize a 6x6 subsample.
```

```
##         [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,]       0 5109333.1 5109333.1 5109333.1 5109333.1 5166310.4
## [2,] 5109333       0.0       0.0       0.0       0.0  170470.3
## [3,] 5109333       0.0       0.0       0.0       0.0  170470.3
## [4,] 5109333       0.0       0.0       0.0       0.0  170470.3
## [5,] 5109333       0.0       0.0       0.0       0.0  170470.3
## [6,] 5166310  170470.3  170470.3  170470.3  170470.3       0.0
```

The 6x6 subsample dataframe (above) yields expected values. First, the diagonal row of 0's is a good sign, as comparing samples to themselves should return a geographic distance of 0. At first the subsample appears to be inflated with 0's. However, I manually checked the initial "Stickleback" dataframe and these samples are all from the same lake, and thus 0 geographic distance between samples is expected. Additionally, I manually checked (using Google Earth Pro™) the linear, geographic distance between samples at different sites and correspond closely with the values in produced dataframe.

```
SticklebackID <- as.character(Stickleback$processid) #To give column/row names meaning, I want to assign t
hem to the processIDs of each sample.
SticklebackID <- as.vector(SticklebackID) #Turn processIDs into a character vector.
class(SticklebackID) #Character.
```

```
## [1] "character"
```

```
SticklebackSampleDist <- base::as.data.frame(Stickleback_Dist) #Convert the matrix into a dataframe.
rownames(x = SticklebackSampleDist) <- SticklebackID
colnames(x = SticklebackSampleDist) <- SticklebackID
#Change rows and columns to processIDs.
SticklebackSampleDist[1:5,1:5] #Visualize that data looks how it should - it does.
```

```
##               ANGBF42584-19 BCFB065-06 BCFB066-06 BCFB067-06 BCFB068-06
## ANGBF42584-19             0    5109333    5109333    5109333    5109333
## BCFB065-06          5109333          0          0          0          0
## BCFB066-06          5109333          0          0          0          0
## BCFB067-06          5109333          0          0          0          0
## BCFB068-06          5109333          0          0          0          0
```

# 5b. Main Analyses II: Population Genetics Analysis

Populations are determined a priori by assigning their country of origin (as denoted by BOLD - note: some "countries" are not truly countries, e.g. Arctic Ocean) as their population. However, countries almost definitely contain multiple populations of stickleback and thus these estimates of differentiation between populations are likely conservative.

```
length(unique(Stickleback$country)) #Samples from 16 countries.
```

```
## [1] 16
```

```
unique(Stickleback$country) #View countries. Should see the relative contribution of each:
```

```
##  [1] "United States"  "Canada"         "North Sea"      "Arctic Ocean"
##  [5] "Netherlands"    "Denmark"        "Sweden"         "Russia"
##  [9] "Poland"         "Latvia"         "Lithuania"      "Germany"
## [13] "France"         "Czech Republic" "Japan"          "Norway"
```

```
ggplot(data.frame(Stickleback), aes(x = country)) +
  geom_bar(colour = "black", fill = "gray") +
  xlab("Sample Country") +
  ylab("Number of Samples") +
  Custom_Theme +
  theme(axis.text.x = element_text(angle=90, hjust=1, vjust=0), text = element_text(size=12)) +
  labs(caption = "Figure 4. Number of three-spined stickleback samples from each of 16 countries.")
```

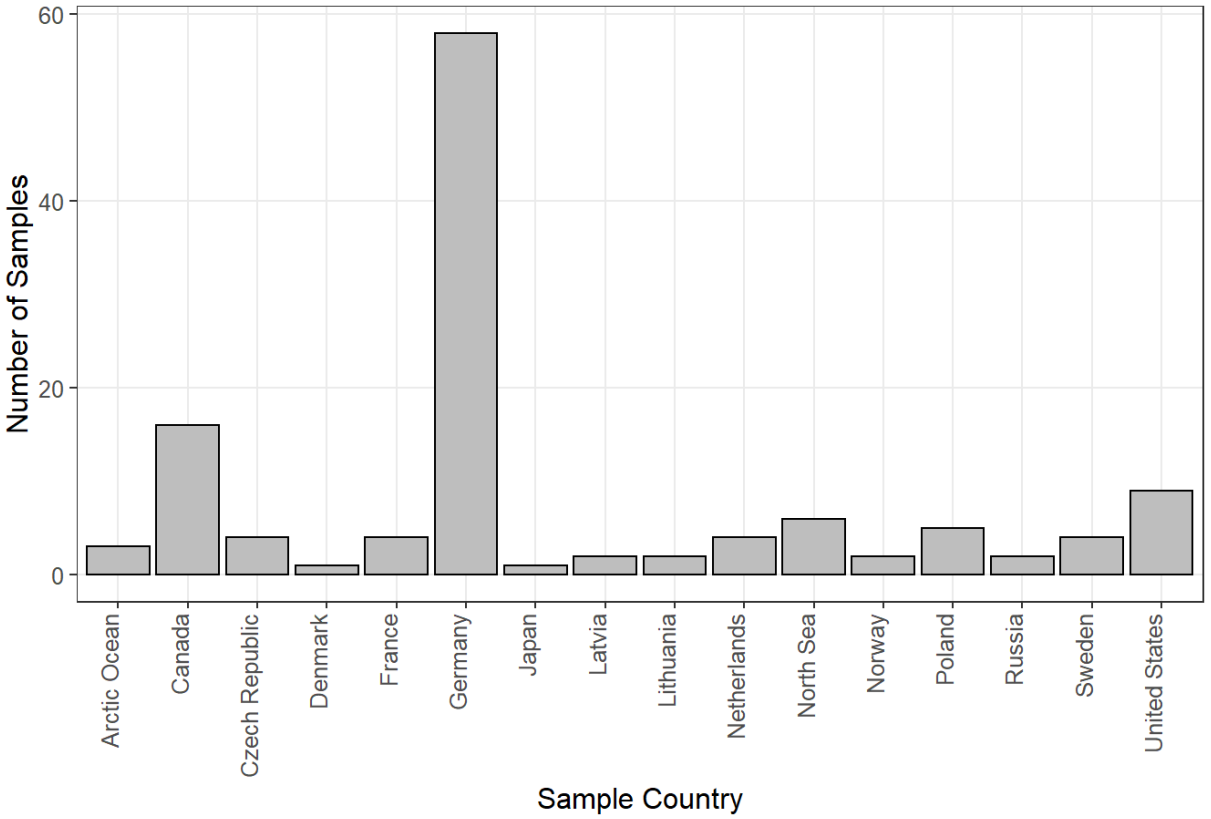


Figure 4. Number of three-spined stickleback samples from each of 16 countries.

Can see Germany has (by quite a bit) contributed the most samples to this analysis, whereas Japan has contributed the fewest.

```
world <- ne_countries(scale = "medium", returnclass = "sf")

ggplot(data = world) +
  geom_sf(fill = "gray90") +
  xlab("Longitude") + ylab("Latitude") +
  theme(panel.background = element_rect(fill = "white", colour = "black", linetype = "solid", size = 2)) +
  annotation_scale(location = "bl", width_hint = 0.1, pad_y = unit(0.6, "cm")) +
  annotation_north_arrow(location = "bl", which_north = "true", pad_y = unit(1.2, "cm"), pad_x = unit(0.5,
"cm"), width = unit(0.8, "cm"), height = unit(0.8, "cm")) +
  coord_sf(ylim = c(0, 85), xlim = c(-180, 180)) +
  geom_point(data = Stickleback, aes(x = lon, y = lat), colour = "black", size = 2) +
  geom_point(data = Stickleback, aes(x = lon, y = lat), colour = "darkgoldenrod", size = 1) +
  labs(caption = "Figure 5. Stickleback sample locations (gold) (n = 123).") +
  theme(plot.caption = element_text(hjust = 0, size = 12)) +
  theme(plot.margin = unit(c(0,1,0,1), "cm"))
```

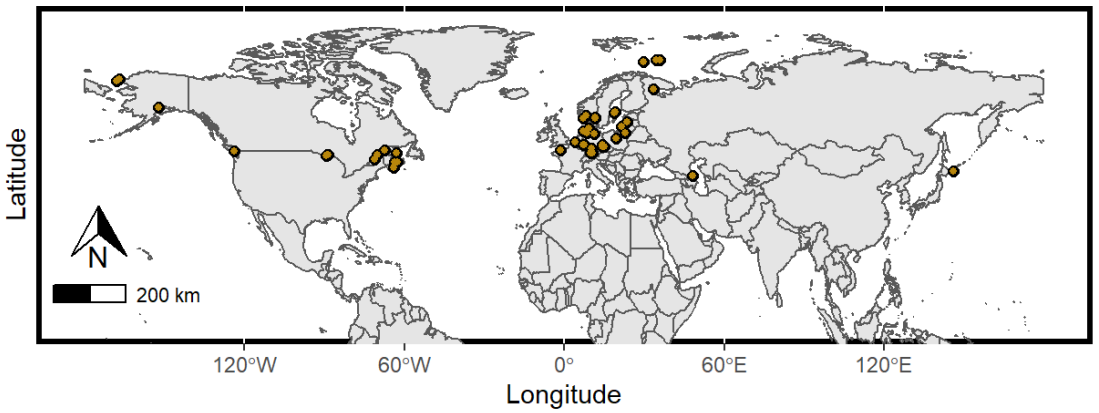

Figure 5. Stickleback sample locations (gold) (n = 123).

```
#I need to sort the Stickleback dataframe by country. I return an error because the column 'nucleotides' i
s still formatted as a DNAStringSet from previous analyses. I will convert it back into characters.
class(Stickleback$nucleotides) #Biostrings.
```

```
## [1] "list"
```

```
Stickleback$nucleotides <- as.character(Stickleback$nucleotides)
```

Here, I convert the sequence alignment into a “genind” object, where each sequence variant is assigned as an allele and individuals are assigned to their respective population. The following analyses were adopted from the page “Population Genetics in R” (https://popgen.nescent.org/PopDiffSequenceData.html).

```
Stickleback.gid <- apex::multidna2genind(SticklebackCOI5P, mlst = TRUE) #mlst = TRUE treats all sequences
 as if they are from the same locus (which they are in this case all from COI-5P).
class(Stickleback.gid) #"genind"
```

```
## [1] "genind"
## attr(,"package")
## [1] "adegenet"
```

```
print(Stickleback.gid) #123 individuals, 1 locus, 19 alleles
```

```
## /// GENIND OBJECT /////////
##
##  // 123 individuals; 1 locus; 19 alleles; size: 35.9 Kb
##
##  // Basic content
##    @tab:  123 x 19 matrix of allele counts
##    @loc.n.all: number of alleles per locus (range: 19-19)
##    @loc.fac: locus factor for the 19 columns of @tab
##    @all.names: list of allele names for each locus
##    @ploidy: ploidy of each individual  (range: 1-1)
##    @type:  codom
##    @call: df2genind(X = xdfnum, ind.names = x@labels, ploidy = 1)
##
##  // Optional content
##    - empty -
```

```
Strata_Country <- data.frame(Stickleback$country)
colnames(Strata_Country) <- "Country" #Change column name from "SticklebackCorrected.country" to "Country"
- much easier to work with.

strata(Stickleback.gid) <- Strata_Country
Stickleback.gid
```

```
## /// GENIND OBJECT /////////
##
##  // 123 individuals; 1 locus; 19 alleles; size: 38.4 Kb
##
##  // Basic content
##    @tab:  123 x 19 matrix of allele counts
##    @loc.n.all: number of alleles per locus (range: 19-19)
##    @loc.fac: locus factor for the 19 columns of @tab
##    @all.names: list of allele names for each locus
##    @ploidy: ploidy of each individual  (range: 1-1)
##    @type:  codom
##    @call: df2genind(X = xdfnum, ind.names = x@labels, ploidy = 1)
##
##  // Optional content
##    @strata: a data frame with 1 columns ( Country )
```

```
setPop(Stickleback.gid) <- ~Country #Use countries as a proxy for "populations".
summary(Stickleback.gid) #123 individuals, Pops range from 1 to 58 individuals, number of alleles per grou
p range from 1 to 8, 0% missing data.
```

```
##
## // Number of individuals: 123
## // Group sizes: 9 16 6 3 4 1 4 2 5 2 2 58 4 4 1 2
## // Number of alleles per locus: 19
## // Number of alleles per group: 2 8 2 1 2 1 2 2 2 1 1 5 4 1 1 1
## // Percentage of missing data: 0 %
## // Observed heterozygosity: 0
```

I calculate some basic statistics about population structure and genetic diversity. Then, I generate a pairwise matrix of genetic distance between all populations. As some populations (i.e. Countries) have small sample sizes, I return several negative values that are converted into zeroes.

```
diff_stats(Stickleback.gid) #Summary stats. Hs = 0.356, Ht = 0.664, Gst = 0.464, Jost's D = 0.510.
```

```
## $per.locus
##                                Hs        Ht       Gst Gprime_st         D
## SticklebackMEGA_trimmed 0.3562375 0.6640239 0.4635171 0.7449923 0.5099792
##
## $global
##        Hs        Ht   Gst_est Gprime_st     D_het    D_mean
## 0.3562375 0.6640239 0.4635171 0.7449923 0.5099792 0.5099792
```

As observed heterozygosity (Hs) is lower than expected heterozygosity (Ht), we can infer that the global population of sticklebacks does not randomly breed and there must be a hierarchical structure to the population (Allendorf et al. 2013).

```
PairwiseGst <- pairwise_Gst_Nei(Stickleback.gid, linearized = FALSE)
```

Nei's measure of pairwise Gst is an estimate of single locus genetic differentiation. This value is calculated using the differences of allele frequencies and distributions within and between populations (determined a priori) (Nei 1973).

Knaus and Grunwald (https://knausb.github.io/vcfR_documentation/genetic_differentiation.html) suggest that negative Gst (or Fst) should be interpretted as 0.00.

```
PairwiseGst[PairwiseGst < 0] <- 0.000
sum(PairwiseGst < 0) #zero, as expected - all negative numbers are removed.
```

```
## [1] NA
```

As I want to create a visual genetic similarity plot between the Stickleback of each country, I need pairwise measurements of *similarity*, whereas Gst is a measurement of dissimilarity. This analysis was adopted from Stackoverflow (https://stackoverflow.com/questions/3081066/what-techniques-exists-in-r-to-visualize-a-distance-matrix) by the user jmb (https://stackoverflow.com/users/4099425/jmb).

```
PairwiseGst <- 1-PairwiseGst #Basically calculating the opposite of each Gst value (i.e. 1 - (proportion o
f dissimilarity) = (proportion of similarity)).
```

I calculate the genetic similarity (1-Gst) between all samples and plot their relationship.

```
qgraph(PairwiseGst, layout='spring', vsize=5, edge.color = "ivory4", node.width=1, shape="ellipse", labels
=rownames(PairwiseGst), label.cex=1, edge.width=0.5, fade = TRUE, trans = FALSE, esize = 11, repulsion = 2
, title = "Figure 6: Pairwise genetic similarities (1-Gst) of stickleback populations among 16 countries.
 \nShorter and thicker lines suggest a closer genetic relationship.")
```

Figure 6: Pairwise genetic similarities (1-Gst) of stickleback populations among 16 countries. Shorter and thicker lines suggest a closer genetic relationship.

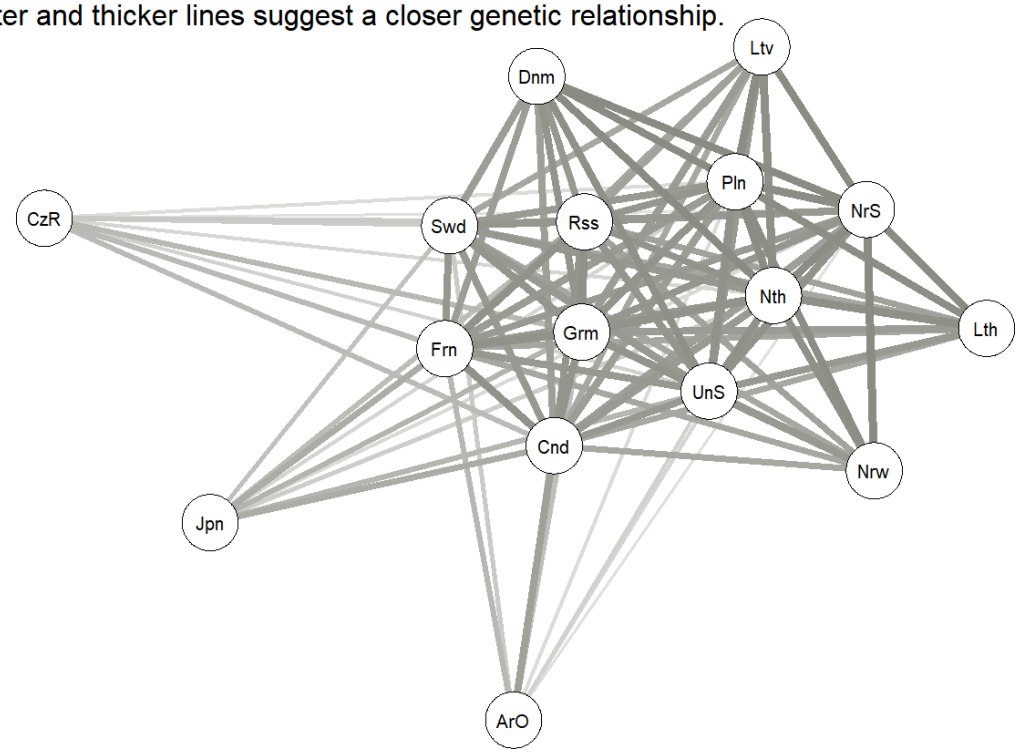


Figure 6 illustrates the pairwise, mean relatedness between 16 populations of three-spined stickleback.

# 5c. Main Analyses III: Genetic and Geographic Distance Comparisons

To compare pairwise genetic and geographic distances among the populations studied, I have to arrange the data similarily.

```
#We have a matrix for pairwise geographic distance and genetic distance for all 123 samples.
#Check that both matrices are configured in the same order - to make comparisons between them.
sum(colnames(SticklebackGenDist) == colnames(SticklebackSampleDist)) #True 123 times - same order, same va
lues.
```

```
## [1] 123
```

```
sum(rownames(SticklebackGenDist) == rownames(SticklebackSampleDist)) #Same as above.
```

```
## [1] 123
```

```
sum(rownames(SticklebackGenDist) == colnames(SticklebackGenDist)) #Same as above - rows and columns are sa
me across both dataframes.
```

```
## [1] 123
```

```
SticklebackGenDist_List <-as.data.frame(unmatrix(x = SticklebackGenDist, byrow = TRUE))
SticklebackSampleDist_List <- as.data.frame(unmatrix(x = SticklebackSampleDist, byrow = TRUE))
#For comparing the content of each matrix, I convert each one into a long dataframe that contains the valu
es of every pairwise comparison, and merge them into a single dataframe containing the genetic and geograp
hic distance between every pair of samples (15625 x 2 values).
Combine <- cbind(SticklebackGenDist_List, SticklebackSampleDist_List) #Data frame with a column for geneti
c distance and geographic distance for all pairwise combinations of samples.
colnames(Combine) <- c("GenDist", "GeoDist")
```

I then plot the relationship between genetic and geographic distance and perform a statistical test to assess the significance of their relationship.

```
ggplot(data = Combine, mapping = aes(x = GeoDist, y = GenDist)) +
  geom_point(position = "jitter", size = 1, alpha = 0.05, colour = "black") +
  ylab("TN93 Genetic Distance") +
  xlab("Linear Geographic Distance (m)") +
  Custom_Theme +
  labs(caption = "Figure 7: The relationship between linear geographic distance (m) and TN93 genetic\ndist
ance among the COI-5P gene across 123 three-spined sticklebacks. \nSolid blue line depicts the mean geneti
c distance +/- 95% confidence interval.") +
  geom_smooth(method = "lm", se = TRUE, colour = "cadetblue") +
  stat_poly_eq(parse = TRUE, formula = Combine$GenDist ~ Combine$GeoDist, aes(label = paste(..eq.label..,
 sep = "~~~")), label.x = 0.92, label.y = 0.9)
```

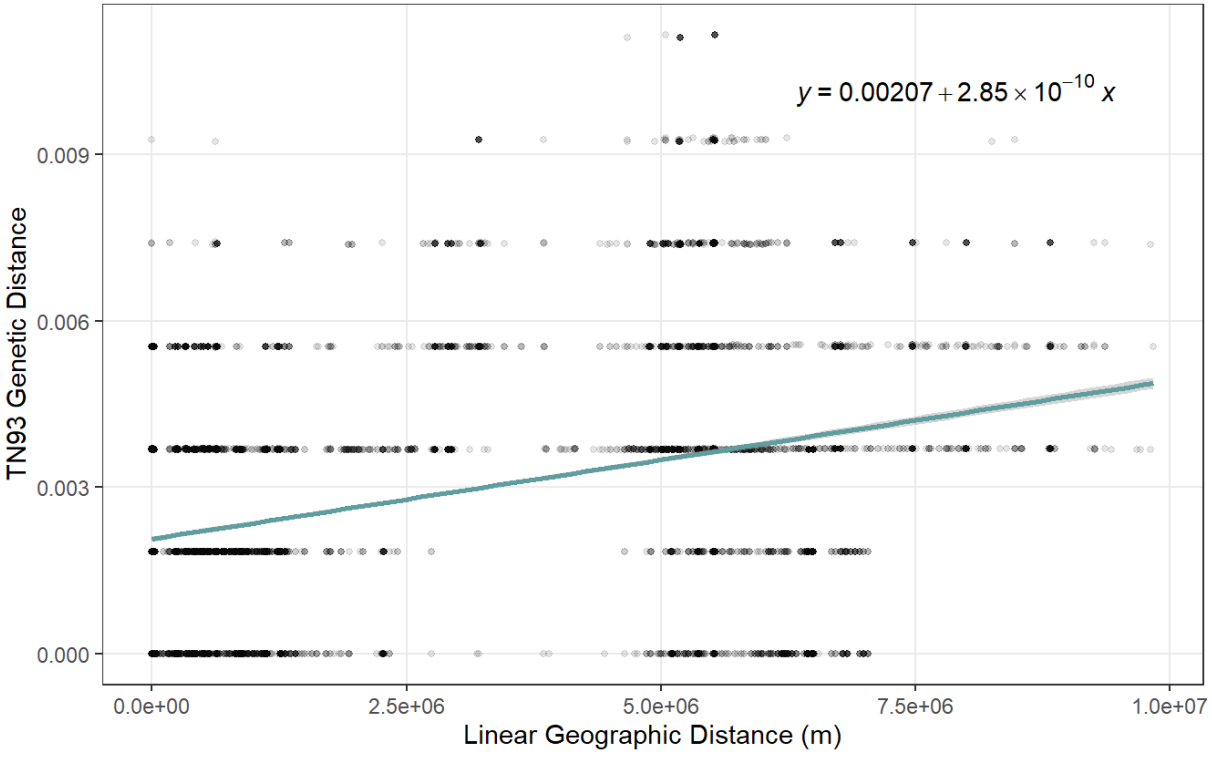


Figure 7: The relationship between linear geographic distance (m) and TN93 genetic distance among the COI-5P gene across 123 three-spined sticklebacks.
Solid blue line depicts the mean genetic distance +/- 95% confidence interval.

I want to statistically quantify the relationship between genetic distance and geographic distance, and thus I will use a Mantel test between the two matrices. First, they need to be in the format “dist”.

```
set.seed(147)
SBSampleDist <- as.dist(SticklebackSampleDist)
SBGenDist <- as.dist(SticklebackGenDist)
mantel.rtest(m1 = SBSampleDist, m2 = SBGenDist, nrepet = 9999)
```

```
## Monte-Carlo test
## Call: mantelnoneuclid(m1 = m1, m2 = m2, nrepet = nrepet)
##
## Observation: 0.3196272
##
## Based on 9999 replicates
## Simulated p-value: 1e-04
## Alternative hypothesis: greater
##
##      Std.Obs  Expectation     Variance
## 4.8884780005 0.0004380617 0.0042633292
```

r value = 0.3756, p-value = 1e-04. Suggests matrices have a significant, positive relationship with each other. Output from Mantel test, using the Monte-Carlo method of permutation. This is expected given the visual plot obtained above.

# 7. Interpretation and Discussion

Despite the significant and positive relationship between genetic distance and geographic distance, the intensity of this relationship is weak ($y=0.00207 + 2.85x10^{-10}x$) (Figure 7). This weakness of this relationship can likely be attributed to the specific conservation of the COI-5P gene as it produces a functional protein associated with critical metabolic functions and thus significant mutations between populations are likely to be deleterious. To achieve a more accurate estimate of pairwise genetic differentiation, this research would benefit from analyzing multilocus allele frequencies using neutral and more polymorphic genetic markers (e.g. microsatellites).

Secondly, there are numerous cases of samples having zero (or low) geographic distance between them, yet are highly genetically differentiated. Stickleback have independently undergone sympatric divergence into pelagic and benthic morphs, and thus patterns of sympatric differentiation likely weaken the relationship between genetic and geographic distance (Marques et al. 2016). Additionally, the COI-5P gene sequence exhibits minimal variation (<3%) among within-species comparisons, and thus a more polymorphic locus would better illustrate between-population differentiation (however, this data is not accessible through public databases such as BOLD or NCBI) (Hebert et al. 2003). For instance, some stickleback samples are thousands of kilometers apart, yet exhibit minimal (even zero!) genetic differentiation at the COI-5P locus, which may be due to chance genetic similarities or ancestral connectivity between distant populations.

Interpopulation genetic similarities (see Figure 6) are (roughly) inversely proportional to geographic distance. For instance, certain pairwise measurements, such as between Japan and Latvia, show the least genetic similarity, which is expected as they are geographically disconnected from one another by continental Europe and Asia. Additionally, we expect that stickleback populations from the Czech Republic are highly differentiated from others as this populations is completely landlocked, whereas the majority of populations used in this study are not. However, most countries are grouped together with a relatively close genetic relationship. This may be due to the chance conservation of COI-5P haplotypes as a result of small samples sizes among many of the populations in this study. Alternatively, this may be due to shared ancestry or gene flow (i.e. genetic admixture) between populations. For some pairwise comparisons, such as between the neighboring countries of Finland and Sweden, the likelihood of gene flow between populations/countries is not unresonable.

In conclusion, geographic distance plays a modest, but significant role in the genetic differentiation of the three-spined stickleback (particularly at the mitochondrial COI-5P locus). However, we also observe significant genetic differentiation in the absence of geographic isolation, thus suggesting that local ecological conditions may also contribute to genetic differentiation in sympatry.

Reflection: I found this project satisfying very worthwhile. My thesis project will utilize similar analyses as used in this project, and thus the time I invested into learning the analytical methods was worthwile and interesting. I learned a lot about data visualization, particularly with respect to mapping (I have even helped lab mates do some mapping since this!). Overall, I'm

very happy with how this project turned out.

# 8. Reference list

References are formatted as per the style of the journal *Evolution*.


Allendorf, F. W., G. Luikart, and S. N. Aitken. 2013. Conservation and the genetics of populations. 2nd ed. John Wiley & Sons, Ltd, West Sussex.

Bell, M. A., & Foster, S. A. (1994). Introduction to the evolutionary biology of the threespine stickleback. In M. A. Bell, & S. A. Foster (Eds.), The evolutionary biology of the threespine stickleback (vol. 1, pp. 27). Oxford, UK: Oxford University Press.

Bolnick, D. I., and B. M. Fitzpatrick. 2007. Sympatric speciation: Models and empirical evidence. Annu. Rev. Ecol. Evol. Syst. 38:459–487.

Bradbury, I. R., M. W. Coulson, A. M. Cook, and P. Bentzen. 2010. Evidence for divergence and adaptive isolation in post-glacially derived bimodal allopatric and sympatric rainbow smelt populations. Biol. J. Linn. Soc. 101:583–594.

Denys, G. P. J., H. Persat, A. Dettai, M. F. Geiger, J. Freyhof, J. Fesquet, and P. Keith. 2018. Genetic and morphological discrimination of three species of ninespined stickleback *Pungitius spp.* (Teleostei, Gasterosteidae) in France with the revalidation of *Pungitius vulgaris* (Mauduyt, 1848). J. Zool. Syst. Evol. Res. 56:77–101.

Dinsdale, A., L. Cook, C. Riginos, Y. M. Buckley, and P. De Barro. 2010. Refined global analysis of *Bemisia tabaci* (Hemiptera: Sternorrhyncha: Aleyrodoidea: Aleyrodidae) mitochondrial cytochrome oxidase 1 to identify species level genetic boundaries. Ann. Entomol. Soc. Am. 103:196–208.

Edgar, R. C. 2004. MUSCLE: Multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res. 32:1792–1797.

Hebert, P. D. N., S. Ratnasingham, and J. R. DeWaard. 2003. Barcoding animal life: Cytochrome c oxidase subunit 1 divergences among closely related species. Proc. R. Soc. B Biol. Sci. 270:13–17.

Kumar, S., G. Stecher, M. Li, C. Knyaz, and K. Tamura. 2018. MEGA X: Molecular evolutionary genetics analysis across computing platforms. Mol. Biol. Evol. 35:1547–1549.

Marques, D. A., K. Lucek, J. I. Meier, S. Mwaiko, C. E. Wagner, L. Excoffier, and O. Seehausen. 2016. Genomics of rapid incipient speciation in sympatric threespine stickleback. PLoS Genet. 12:1–35.

Nei, M. 1973. Analysis of gene diversity in subdivided populations. Proc. Natl. Acad. Sci. U. S. A. 70:3321–3323.

Tamura, K., and M. Nei. 1993. Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. Mol. Biol. Evol. 10.

Weber, J. N., G. S. Bradburd, Y. E. Stuart, W. E. Stutz, and D. I. Bolnick. 2017. Partitioning the effects of isolation by distance, environment, and physical barriers on genomic divergence between parapatric threespine stickleback. Evolution. 71:342–356.

Wright, S. 1943. Isolation by distance. Genetics 332–335.


# Acknowledgements


Thanks to Dr. Sally Adamowicz for guiding this project and suggesting statistical analyses and providing package recommendations.