## Background

Trinidadian guppies are a small species of fish native to the rivers of Trinidad (an island located in the Caribbean, near the coast of Venezuela). The mountains of Northern Trinidad are a source of multiple different rivers that flow across the island and eventually drain into the sea. Along the way, these rivers are punctuated by numerous waterfalls, which serve as barriers to larger species of fish. This means that guppies living upstream of waterfalls have fewer predators than the guppies living downstream, and this has lad to multiple different adaptations along the different river gradients.

Maracas-Bay-in-Trinidad-and-Tobago-beach-1200x825.jpgimage_5257e-Trinidadian-guppies.jpgaripo-waterfall.jpeg

For this exercise, you will be working with data from 3 sets of guppies, each of which has been collected from a different river: the Aripo, Guanapo, and Tacarigua.

In each dataset, there are samples from High Predation (HP) populations collected downstream of the major waterfalls, and Low Predation (LP) populations collected from the upstream river sections. Each sample name has a 4 letter code that denotes the river it was collected from, and whether it came from a low or high predation population.

All samples that start with "AP" are from the Aripo river, "GU" indicates samples from the Guanipo river, and "TR" indicates samples from the Tacaringua river. "HP" refers to high predation, and "LP" refers to Low Predation. After the 4 letter code, each sample is given a unique number to denote different individuals. For example, "APLP10" would be fish #10 from the Low Predation zone in the Aripo river.

map.jpg

We will ultimately be looking for variants that differentiate the High and Low predation populations, but first, we need to filter our data files and get them ready for analysis.

## Questions

Start by downloading the **3 VCF files** that correspond to the 3 different river datasets:

Even though these files have the ".vcf" extension, like many files we use in bioinformatics, they are just text files, so it is possible to open them (or partially open them) to see the information inside, or to explore them with basic unix commands.

#### Question 1: How many SNPs (variants) are in each dataset? How many samples are in each dataset? [*3 pts total*]

#### 1A. Aripo? *(1 pt)*

There are 83777 SNPS and 38 samples.

#### 1B. Guanapo? *(1 pt)*

There are 57931 SNPS and 37 samples.

#### 1C. Tacarigua? *(1 pt)*

There are 26874 SNPS and 26 samples.

Now that we have some basic information for each of the datasets, go ahead and upload your VCF files to the cluster, so that we can use a program called VCFtools to do some basic calculations. This is a very useful set of programs for working with VCF files, and you can see the full manual and all of the options [here](https://vcftools.sourceforge.net/man_latest.html)

VCFtools is already installed on the cluster, so to use it we only need to load the module:

module load vcftools

We are first going to use VCFtools to calculate Hardy-Weinberg expectations for each SNP, and to determine which sites are significantly deviating from Hardy-Weinberg Equilibrium. The command to do this will look like this:

vcftools --vcf in.vcf --out outName --hardy

In this command, in.vcf is the name of the input vcf file that we want to do the calculation on; outName will be a prefix for the output file that VCFtools will make. This can be any name you want; Vcftools will take this name and add ".hwe" to the file that it makes with all of the calculations. Finally, the --hardy is the option that calls up the Hardy-Weinberg calculation tool.

To run this on your datasets, enter the correct names for each input and output file. You can either run this command in a slurm script, or on the head node (these are small files, so none of our vcftools commands will create memory or time intensive jobs today).

The results produced by VCFtools will be in a file that ends in ".hwe." This file will be a tab-delimited text file with 8 columns (use "less" or "head" to get a quick glimpse of the file).

#### Question 2: How many SNPs significantly deviate from Hardy-Weinberg Equilibrium in each dataset [*3 pts total*]

R Code:

#### a <- read.table("./aripo.hwe",header=TRUE,sep="\t") g <- read.table("./guanapo.hwe",header=TRUE,sep="\t") t <- read.table("./tacarigua.hwe",header=TRUE,sep="\t") nrow(a[a$P\_HWE<0.05,]) nrow(g[g$P\_HWE<0.05,]) nrow(t[t$P\_HWE<0.05,])

#### 2A. Aripo? *(1 pt)*

12751 SNPS significantly deviated from HWE.

#### 2B. Guanapo? *(1 pt)*

14522 SNPS significantly deviated from HWE.

#### 2C. Tacarigua? (*1 pt)*

8884 SNPS significantly deviated from HWE.

*There are many ways that you can go about answering this question. If you're comfortable with coding, you could use a simple python or R script to do this. Or, you could try a command (or series of commands) in the shell. If you want to go this route, I recommend looking into awk rather than plain bash. [Here is a potentially useful tutorial.](https://www.tim-dennis.com/data/tech/2016/08/09/using-awk-filter-rows.html)*

*Or, you can do this in Excel, since these files are small enough to open and manipulate the way you would any spreadsheet.*

*The column that you want to use to determine significance is P\_HWE, and I would use a cutoff of p<0.05.*

#### Question 3: Of the sites that are not in HWE, how many of them show Excess Heterozygosity (versus a heterozygosity deficit)? Get a list of all of the non-HWE, HET\_EXCESS positions for each dataset. [6 *points total].*

R Code:

#### aexcess <- a[(a$P\_HWE<0.05) & (a$P\_HET\_EXCESS<0.05),] a3 <- paste(aexcess$CHR,aexcess$POS,sep="\_") gexcess <- g[(g$P\_HWE<0.05) & (g$P\_HET\_EXCESS<0.05),]

#### g3 <- paste(gexcess$CHR,gexcess$POS,sep="\_")

#### texcess <- t[(t$P\_HWE<0.05) & (t$P\_HET\_EXCESS<0.05),]

#### t3 <- paste(texcess$CHR,texcess$POS,sep="\_") nrow(aexcess)

#### nrow(gexcess)

#### nrow(texcess)

#### 3A. Aripo? *(2 pts*)

882 sites show excess heterozygosity.

#### 3B. Guanapo? *(2 pts)*

740 sites show excess heterozygosity.

#### 3C. Tacarigua *(2 pts*)

1179 sites show excess heterozygosity.

More R Code:

write.table(aexcess,"aexcess")

write.table(gexcess,"gexcess")

write.table(texcess,"texcess")

*You can approach this question similarly to however you approached question 2, but this time* ***be sure to save a list of the excess het positions to a separate file!***  *You also need to get this file into a specific format, where you have the chromosome and position separated by a "\_" for each position. For example:*

*"chr20\_21567"*

*Your file should have 1 line per position, with each position formatted with the underscore ("\_"). If you look in the VCF file, you will notice this corresponds to the "ID" column. You are going to need this list for each dataset in order to do the next steps.*

Once you have your list of SNPs that show a significant deviation from HWE, and specifically a significant excess of heterozygosity, you can use VCFtools to filter out these sites from your datasets.

vcftools --vcf in.vcf --out outName --recode --exclude list\_from\_question\_3

In the above command, you start off similarly to how we calculated HWE (with the input and output names), but then we have a --recode option, which tells VCFtools we want it to create a new VCF file after it applies our filter. Then we use the --exclude option to specify a list of SNPs that we want to exclude from our new file, and this option is followed by the file with the list of SNPs that we got in Question 3. VCFtools will then make a new file called "outName.recode.vcf" which should contain our filtered dataset that no longer has the excess heterozygosity SNPs.

Once you have your 3 filtered datasets, figure out which SNP IDs are present in **all 3 datasets**, and make a new list (in the same format as the lists from question 3). This question again has many ways that you could go about it, but I would probably recommend the "cut" command in bash as something that might be very useful to get just the ID column from each file.

#### Question 4: How many SNPs are present in all 3 datasets? [*3 pts*]

Once you have your list of overlapping SNPs, make sure you save this in a separate text file (and save your filtered datasets), as you will need these next week!

R Code:

aripo<-readLines("./aripo.recode.vcf")

aripo\_data<-read.table("./aripo.recode.vcf", stringsAsFactors = FALSE)

# filter for the columns names

aripo<-aripo[-(grep("#CHROM",aripo)+1):-(length(aripo))]

aripo\_names<-unlist(strsplit(aripo[length(aripo)],"\t"))

names(aripo\_data)<-aripo\_names

guanapo<-readLines("./guanapo.recode.vcf")

guanapo\_data<-read.table("./guanapo.recode.vcf", stringsAsFactors = FALSE)

# filter for the columns names

guanapo<-guanapo[-(grep("#CHROM",guanapo)+1):-(length(guanapo))]

guanapo\_names<-unlist(strsplit(guanapo[length(guanapo)],"\t"))

names(guanapo\_data)<-guanapo\_names

tacarigua<-readLines("./tacarigua.recode.vcf")

tacarigua\_data<-read.table("./tacarigua.recode.vcf", stringsAsFactors = FALSE)

# filter for the columns names

tacarigua<-tacarigua[-(grep("#CHROM",tacarigua)+1):-(length(tacarigua))]

tacarigua\_names<-unlist(strsplit(tacarigua[length(tacarigua)],"\t"))

names(tacarigua\_data)<-tacarigua\_names

all <- intersect(aripo\_data$ID, guanapo\_data$ID)

all2 <- intersect(all,tacarigua\_data$ID)

all2

There are 17172 SNPs present in all 3 datasets.

## Part Two Questions

Before starting any analysis of population structure, you need to create a final filtered and merged VCF file based on the work you did in Part One. Ultimately, you want a single VCF file that contains only the sites present in all 3 datasets (so that we don't have to worry about missing data in our later calculations). To do this, you will start by filtering each dataset to keep only the sites in your overlapping SNPs list. Then you will merge the three filtered datasets into one.

The command you will use to filter each dataset in order to keep only the SNPs in a list looks a lot like the command you used earlier when we wanted to exclude a certain list of SNPs. The only difference is now we will use a "--snps" option instead of a "--exclude" option:

vcftools --vcf in.vcf --out outName --recode --snps list\_from\_question\_4

Once you have the filtered VCF file for each dataset, it is time to merge them. We will use bcftools to do this, but in order for it to work we first have to zip and index each VCF file.

Start by loading the samtools module on the cluster:

module load samtools

Then, zip each file using the bgzip command:

bgzip yourfile.vcf

These files are not large, so this will run very quickly, and does not need to be submitted through a slurm script. Once you have the zipped files, index each of them with the tabix command:

tabix yourfile.vcf.gz

You should now have a file ending in .vcf.gz and another file ending in .tbi for each of your datasets. To get the merged file, run:

bcftools merge -Ov -o outName.vcf yourfile1.vcf.gz yourfile2.vcf.gz yourfile3.vcf.gz

Be sure to change outName to something you'll be able to remember is the name of your final VCF file!

### Question 5: What are the dimensions (number rows/number columns) of your final VCF file, and do they match up with what you were expecting based on your earlier calculations? (2 pts)

If anything DOESN'T match, then you may need to go back and re-check either your calculations or your merging step!

There are 17172 SNPs and 101 individuals. Yes, they did match up to Part 1’s summated individuals with their shared SNPs.

### Question 6: How much variation in the data can be explained by Principal Components? *[3 pts total]*

Before answering any of the following questions, you'll need to run the Principal Components Analysis. To do this, you can use a program called plink, which can run many different population genetic and quantitative genetics analyses on SNP data. This package is already installed on the cluster, so all you need to do is load the module and run one command on your VCF file:

module load plink/1.90

plink --vcf yourfile.vcf --pca NUM\_SAMPLES #Change Num\_SAMPLES to 6

Change NUM\_SAMPLES to be the total number of individuals in your merged file. This is the maximum number of PCs that can be calculated.

You may notice the plink was originally written for human data, as it will refer to your samples as "people" and will give you a message about there not being any sex information in the dataset. You don't need to worry about this, as it does not affect the pca calculation at all.

When it is finished (which will be very fast), it will make several new files, all starting with the generic prefix "plink" (unless you specified something else). It will also print a message to your screen telling you what files it is making and where it saved information. You should download these files and open them to look at what kind of information is in each.

#### Question 6A: How much variation is explained by the first 2 Principal Components? (1 pt)

R Code:

eigenval<-read.table("./plink.eigenval",header=FALSE)

eigenval[1,1]/sum(eigenval) #0.2607301

eigenval[2,1]/sum(eigenval) #0.2303938

26.1% of the variation is explained by the first principal component and 23% is explained by the second component for a total of 49.1% variation explained from both.

#### Question 6B: How many Principal Components do you need to keep in order to explain 90% of the variance? (1 pt)

R Code:

sum = 0

count = 0

for (i in eigenval[,1]) {

#print(i/sum(eigenval))

if (sum < 0.9) {

sum = sum + i/sum(eigenval)

count = count + 1

}

}

print(count)

print(sum)

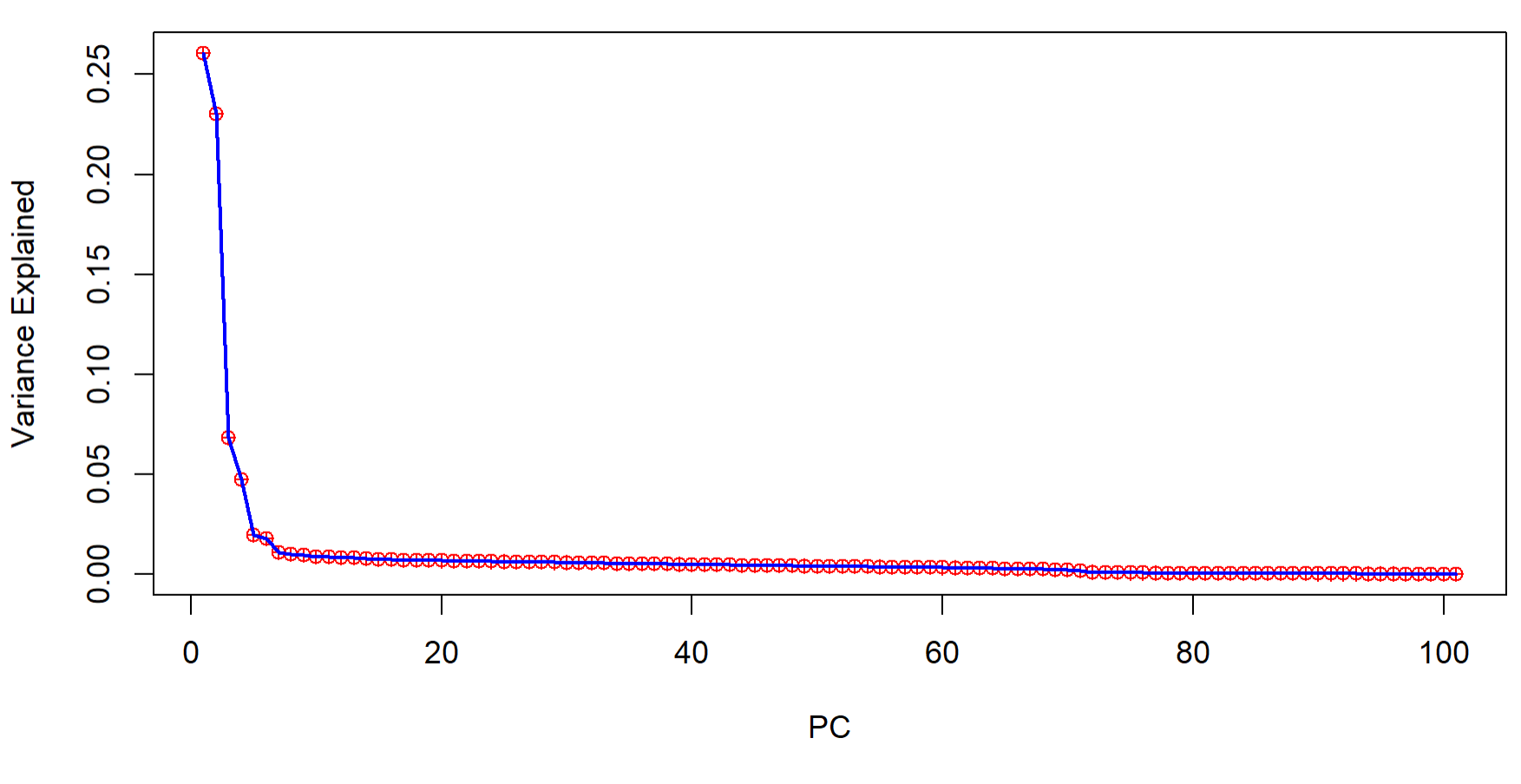
46 principal components are needed to explain 90% of the variance.

#### Question 6C: Make a scree plot. How many PCs would you keep based on this plot? (1 pt)

R Code:

plot(seq(1:101),y,xlab= "PC", ylab = "Variance Explained", pch = 10, col = "red", cex =1)

lines(seq(1:101), y, col="blue",lwd = 2)



I would keep up to the 6th PC, where the plot looks to level out at.

### Question 7: What does the PCA clustering look like? [3 pts total]

To create the various PCA plots, I am providing you with some R code. If you do not have R on your computer, you can download it [here](https://cran.r-project.org/)

Alternatively, you can also run R on the cluster (it is available as a module there), or you can create the plots using the program of your choice.

#### Question 7A: How many clusters do there appear to be based on your PCA plot (without any color coding)? (1 pt)

There appears to be 5 different clusters.

#### Question 7B: Do the clusters appear to correspond to the different rivers? Which river appears the most distinct? (1 pt)

Yes, they do correspond to the different rivers. The Tacarigua river appears to be the most distinct as it varies the most in the axis of PC1 and also varies in the PC2 component.

#### Question 7C: Do the High Predation (HP) and Low Predation (LP) fish form distinct clusters? Are they distinct in all of the rivers, none of them, or just some of them? (1 pt)

The HP and LP fish form distinct clusters in the Tacarigua and Guanapo rivers, but not the Aripo river (where it is all in one cluster/general area).

Here is the R code for reading your data into R and getting a table set up so that you can plot with various color-coding schemes:

### Set up your working directory to be where you have your plink results files:

setwd("/YOUR/FILE/PATH/GOES/HERE")

### Read in your Plink Eignevector results file

my.plink = read.table("plink.eigenvec")

### Extract just PC1 and PC2 into their own table

my.pcs = my.plink[,3:4]

colnames(my.pcs) = c("PC1","PC2")

### Add a column that contains the 2 letter code for each river (the first 2 letters of each sample name)

my.pcs$River = substr(my.plink$V1, 1, 2)

### Add another column that contains the 2 letter code for high or low predation (next 2 letters of each name)

my.pcs$Predator = substr(my.plink$V1, 3, 4)

### Check that table looks correct

head(my.pcs)

PC1 PC2 River Predator

1 0.0236388 -0.1194410 AP HP

2 0.0191676 -0.0965316 AP HP

3 0.0196426 -0.1311700 AP HP

4 0.0247948 -0.0949192 AP HP

5 0.0249765 -0.0920698 AP HP

6 0.0261028 -0.1289290 AP HP

Once you have your table set up, here is the code to create the basic PCA plot with no colors:

### Create a basic PCA plot (no color coding)

plot(my.pcs$PC1, my.pcs$PC2, xlab="PC 1", ylab="PC 2", pch=20)

Here is the R code to color-code by River:

### Install and load ggplot2

install.packages("ggplot2")

library(ggplot2)

### Create a plot color-coded by river (using custom colors that should be color-blind safe)

myplot = ggplot(data = my.pcs, aes(x=PC1, y=PC2, col=River)) + geom\_point() + scale\_color\_manual(values=c("#56B4E9", "#D55E00", "#009E73"))

myplot + theme\_bw()

And here is the code to create a plot color-coded by River, with different symbols for high and low predation:

### Create a plot color-coded by river, with circles and triangles to denote HP and LP

myplot = ggplot(data = my.pcs, aes(x=PC1, y=PC2, col=River, shape=Predator)) + geom\_point() + scale\_color\_manual(values=c("#56B4E9", "#D55E00", "#009E73"))

myplot + theme\_bw()

### Question 8: Calculate Fst between High and Low Predation Populations [*3 pts total*]

You will be using VCFtools to calculate Fst between the high and low predation fish. The VCFtools function needs 2 lists that define which individuals make up the 2 populations that you want to compare. So, list #1 will be a file that has a 1 column list of all of the sample names for fish in high predation populations. List #2 will be a separate file that has 1 column of all of the sample names for fish in the low predation populations. Start by creating these 2 files.

Then, calculate Fst in 10Kb sliding windows using the following command:

vcftools --vcf YOURFILE.vcf --out outNAME --weir-fst-pop YOUR-LIST-HP-INDIVIDUALS --weir-fst-pop YOUR-LIST-LP-INDIVIDUALS --fst-window-size 10000 --fst-window-step 10000

Be sure to change the file names to match your own!

#### Question 8A: What is the overall mean weighted Fst score for the whole dataset? Does this score suggest a high, low, or intermediate level of differentiation? *(1 pt)*

R code:

my.pcs2 = my.plink[,3:4]

colnames(my.pcs2) = c("PC1","PC2")

my.pcs2$River = my.plink$V1

my.pcs2$Predator = substr(my.plink$V1, 3, 4)

hp = my.pcs2$River[my.pcs2$Predator == "HP"]

lp = my.pcs2$River[my.pcs2$Predator == "LP"]

write.table(lp,"lp", col.names = FALSE, row.names = FALSE)

write.table(hp,"hp", col.names = FALSE, row.names = FALSE)

#Then I removed the quotations from the text files.

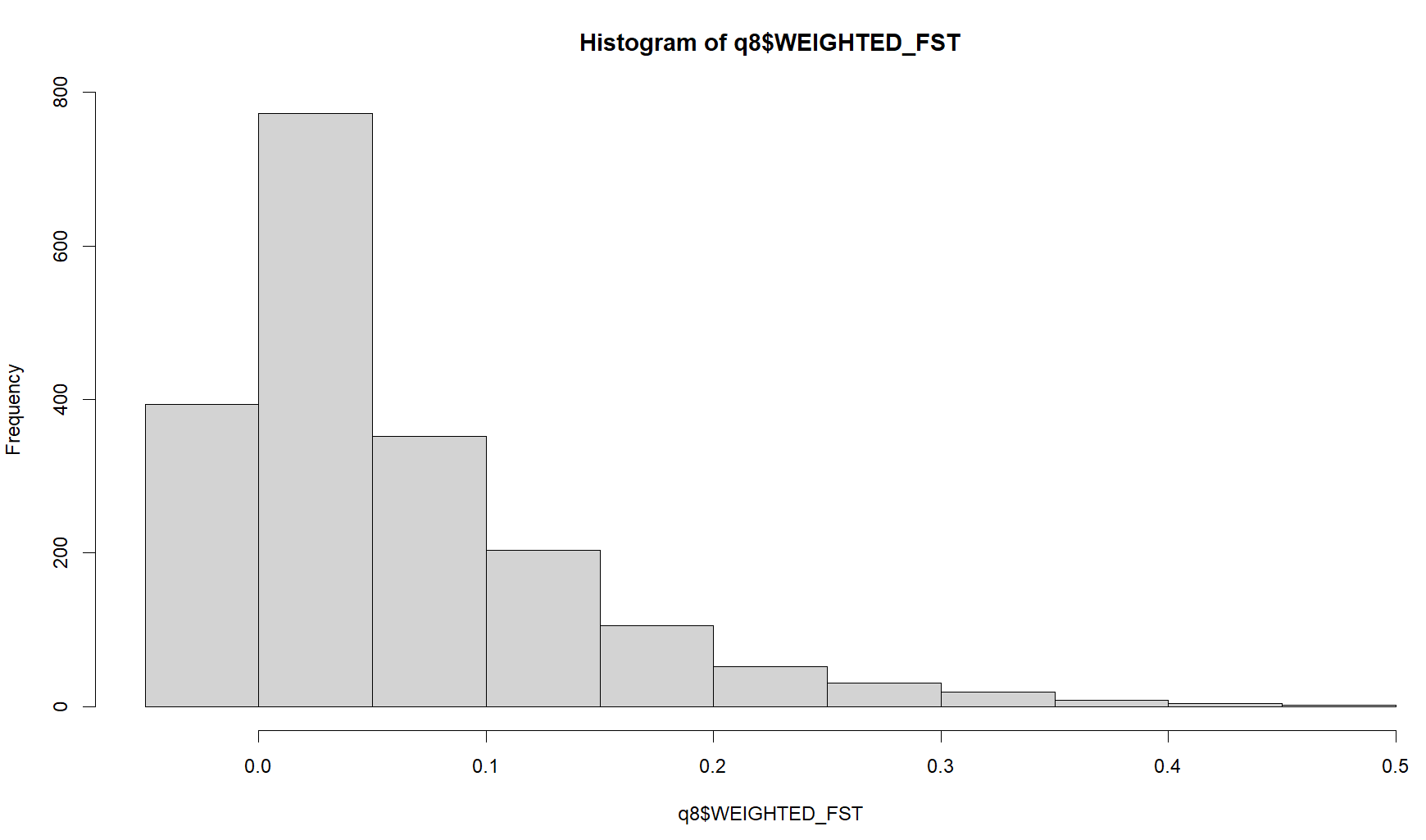
q8 <- read.table("./q8.windowed.weir.fst", sep="\t", header = TRUE)

mean(q8$WEIGHTED\_FST)

The overall mean weighted Fst score for the whole dataset is 0.05822074. This suggests that individuals vary just as much within a group as they do between groups (i.e., the groups are identical)/low level of differentiation.

#### Question 8B: What does the distribution of Fst scores look like? *(1 pt)*

*Hint: if you don't know how to do this in another program, then look up how to use the hist(x) function in R.*



The distribution of data looks very right/positively skewed.

#### Question 8C: On average, which value is higher, mean or weighted Fst? *(1 pt)*

R Code:

sum(q8$WEIGHTED\_FST > q8$MEAN\_FST) # 784

sum(q8$WEIGHTED\_FST < q8$MEAN\_FST) # 868

sum(q8$WEIGHTED\_FST == q8$MEAN\_FST) # 291

Mean Fst was generally higher in value.

### Question 9: Identify highly differentiated genomic regions using Fst [*4 pts total*]

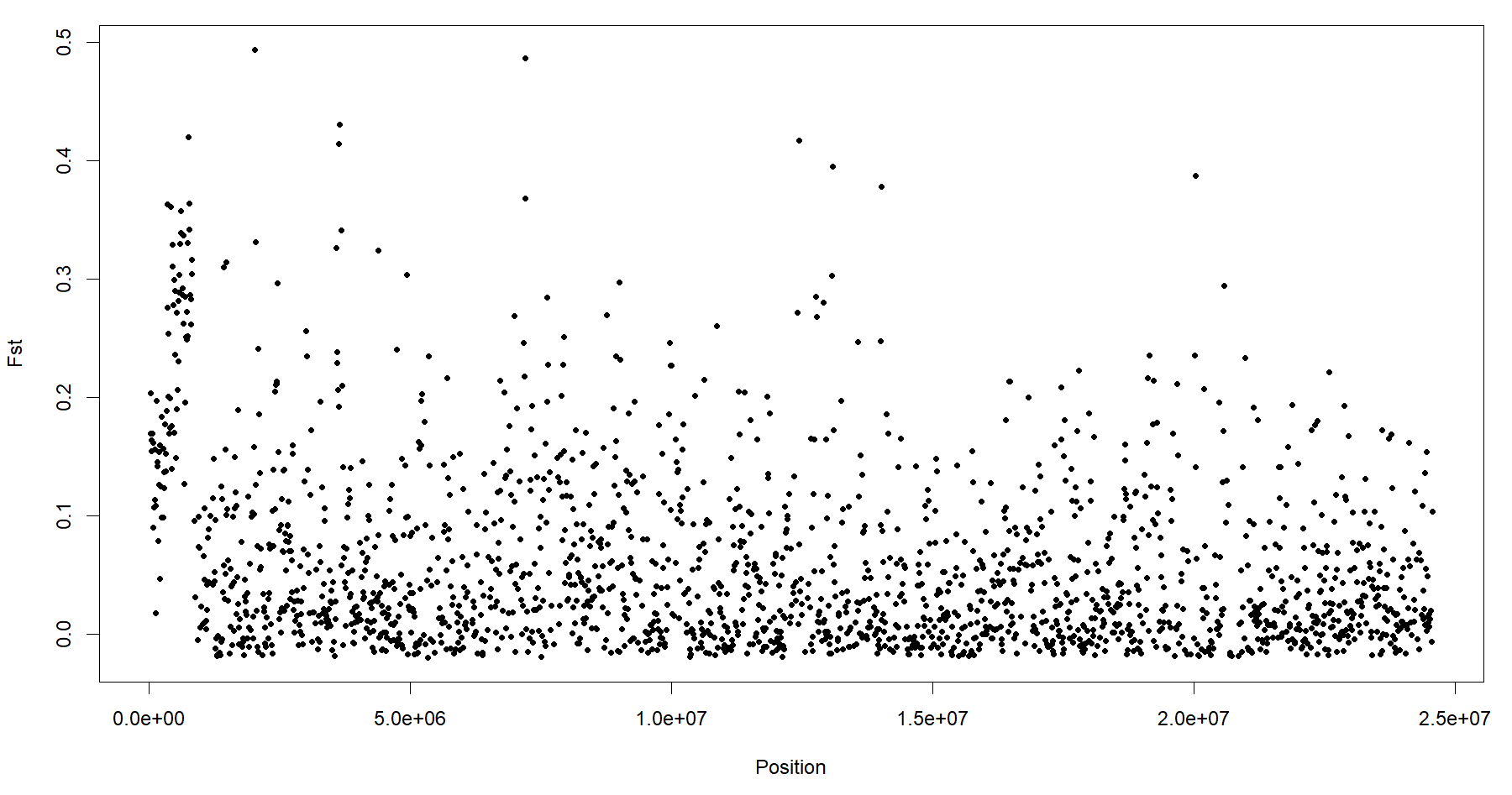
To do this last part of the exercise, you will be examining plots of Fst values along the chromosome. If you don't know how to do this in another program, I've provided you with some R code below.

#### Question 9A: Plot weighted Fst by position along the chromosome (you can just use the BIN START positions). Is there any region that stands out, or are there high Fst sites all along the chromosome? *(1 pt)*

*Here is some R code to read in the Fst results and do a simple plot along the chromosome:*

my.fst = read.table("YOUR-FST-RESULTS.fst", header=TRUE, stringsAsFactors = FALSE)

plot(my.fst$BIN\_START, my.fst$WEIGHTED\_FST, xlab="Position", ylab="Fst", pch=20)

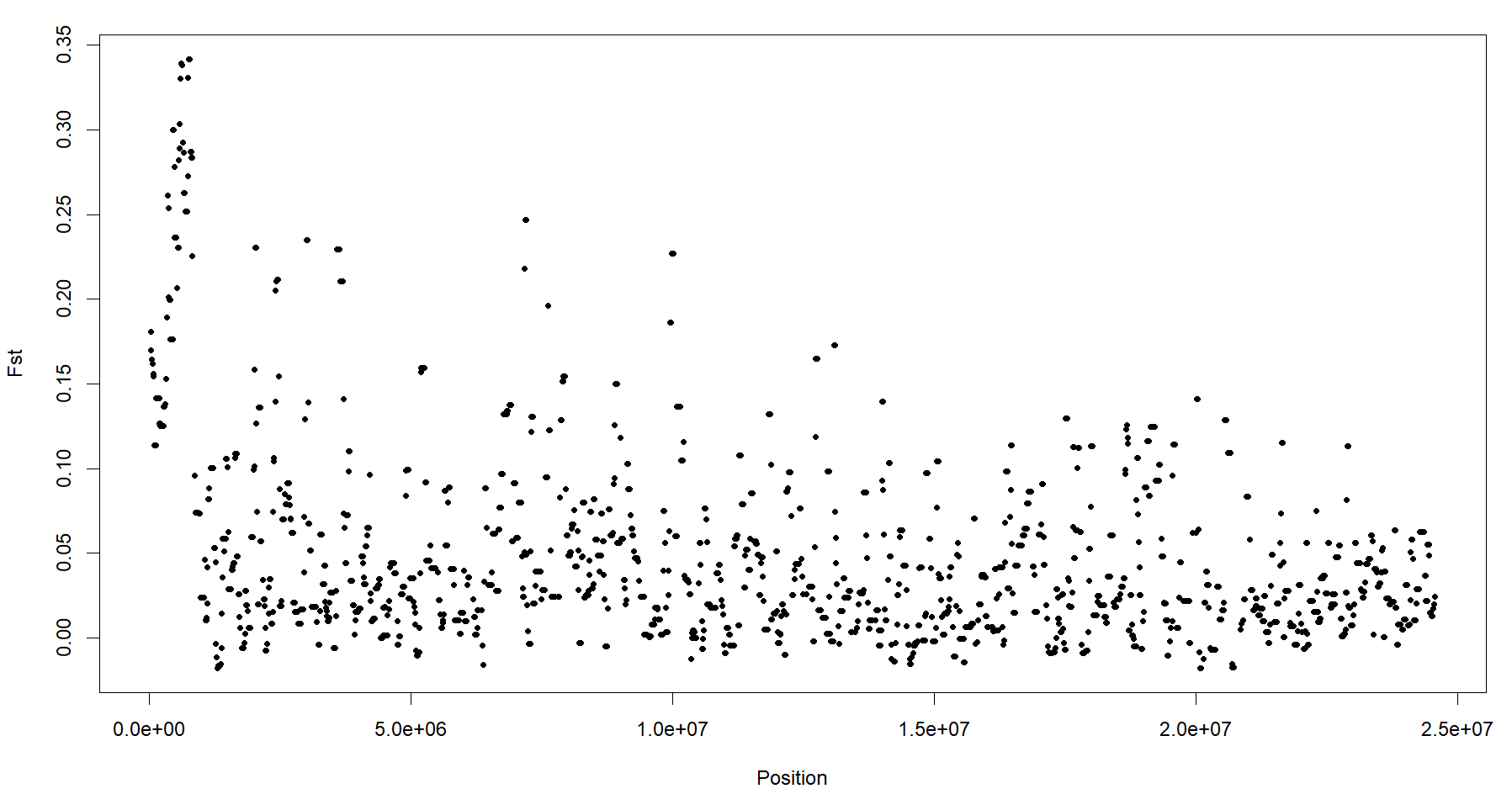


Only the beginning position seems to have generally higher Fst sites than everywhere else on the chromosome.

#### Question 9B: Use a smoothing function to try and remove some of the noise, then re-plot Fst along the chromosome. Does any region stand out now? Roughly where is it? *(1 pt*)

*Here is the R code for the smoothing function:*

smooth.fst = smooth(my.fst$WEIGHTED\_FST)



Now the beginning of the chromosome clearly stands out, roughly around the bin start positions of 580001 to 770001.

#### Question 9C: What are the beginning and end positions for the windows with the maximum smoothed Fst values? *(1 pt)*

*Hint, in R, running max(smooth.fst) will tell you what the highest value is; running which(smooth.fst==max(smooth.fst)) will give you the row numbers that correspond to those values (so then you only need to look up the chromosomal positions associated with the row numbers).*

R Code:

my.fst$BIN\_START[which(smooth.fst==max(smooth.fst))]

[1] 750001 760001 770001 are the beginning positions.

my.fst$BIN\_END[which(smooth.fst==max(smooth.fst))]

[1] 760000 770000 780000 are the end positions.

#### Question 9D: What gene(s) are located in this region, and what is the function of their human orthologs? *(1 pt).*

*Hint: the species name for guppy is Poecilia reticulata. Your SNP data set comes from chromosome 20, which is the same as LG20.*

<https://www.ncbi.nlm.nih.gov/nucleotide/NC_024350.1/?report=graph>

<https://www.ncbi.nlm.nih.gov/gene/103482221>

<https://www.ncbi.nlm.nih.gov/gene/1266>

CNN3/calponin 3 enables actin filament binding, cadherin binding involved in cell-cell adhesion, and calmodulin binding.