Population Genetic Analysis III: Biallelic SNP Data
BZ/MIP 577 Fall 2015

Kelly Pierce

Exercises

```
library(qvalue)
```

Download and load the data

Allele frequency table containing the same information as in the Bayescan input file:

```
# load the raw allele frequency data
setwd('~/Desktop/CSU_PopGen_Labs/Data/')
allele.freqs<-read.table('wide_format_Bayescan.txt', header=T)

# inspect the data format
head(allele.freqs)
```

```
##   Locus_1 A_1 B_1        p_1       q_1 A_2 B_2       p_2       q_2
## 1 locus_1 126  30 0.80769231 0.1923077  26   6 0.8125000 0.1875000
## 2 locus_2  30 126 0.19230769 0.8076923   6  26 0.1875000 0.8125000
## 3 locus_3  22 138 0.13750000 0.8625000   8  22 0.2666667 0.7333333
## 4 locus_4  13 145 0.08227848 0.9177215   2  30 0.0625000 0.9375000
## 5 locus_5 145  15 0.90625000 0.0937500  28   4 0.8750000 0.1250000
## 6 locus_6  39 115 0.25324675 0.7467532   8  24 0.2500000 0.7500000
```

Bayescan output file:

```
# load in the Bayescan output data
setwd('~/Desktop/CSU_PopGen_Labs/Data/')
bayescan.out<-read.table('Final_Merged_Both_Sites_Bayescan_Input_fst.txt')

# inspect the data format
head(bayescan.out)
```

```
##        prob log10.PO.    qval      alpha       fst
## 1 0.0938188 -0.984925 0.89356 -0.0129100 0.0049603
## 2 0.0908180 -1.000500 0.90035 -0.0054275 0.0050024
## 3 0.0902180 -1.003600 0.90144  0.0138280 0.0052013
## 4 0.0938190 -0.984930 0.89356 -0.0117640 0.0049650
## 5 0.0808160 -1.055900 0.90774 -0.0056248 0.0049612
## 6 0.0864170 -1.024100 0.90559 -0.0095594 0.0049918
```

Preliminary analysis

Before launching into the F_{ST} outlier analysis, it is often useful to visualze the raw data. Looking at the allele frequencies may provide clues as to which loci (if any) are under selection. Loci that have different allele frequences in the two subpopulations are possible contenders for being under selection and having outlier F_{ST} values.

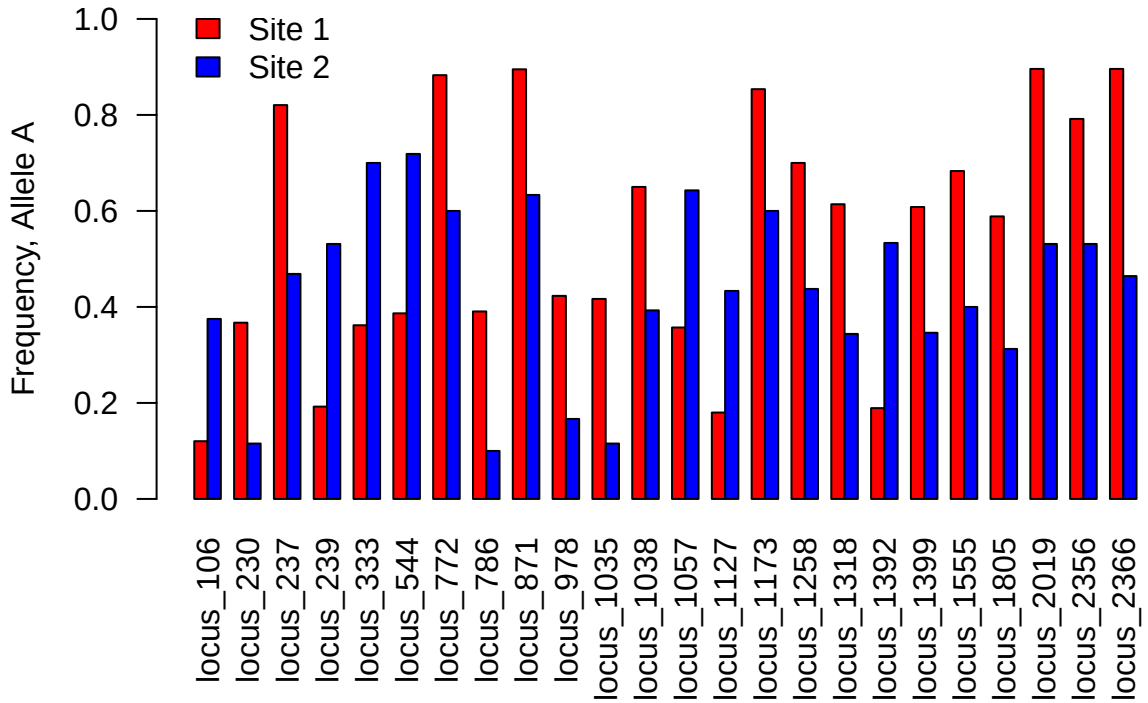
We will start by looking for loci with an allele frequencies with large differences between the site 1 and site 2 subpopulations. We will use the *abs()* function to calculate the absolute value of the difference in allele frequencies between the two populations, and search for alleles where that difference is $> 25\%$. We will also plot the results as a barplot.

```
# extract the loci that have more than 25% difference
# in frequency between the two populations
disparate.freq <- allele.freqs[which(abs(allele.freqs$p_1-allele.freqs$p_2)>0.25),]

# how many were there?
length(disparate.freq[,1])
```

```
## [1] 24
```

```
par(mar=c(7, 4, 3, 2))
barplot(rbind(disparate.freq$p_1, disparate.freq$p_2),
        beside=T, col=c('red', 'blue'),
        names=disparate.freq$Locus_1,
        ylab='Frequency, Allele A',
        las=2, ylim=c(0,1))
legend(x=-1, y=1.06, legend=c('Site 1', 'Site 2'), fill=c('red', 'blue'),
       bty='n') # put the legend exactly where you want it by specifying x and y
```



Question 1. How many SNPs with disparate allele frequencies in the two subpopulations did you find?

From executing the code above, there are 24 alleles that have disparate frequencies in the two populations.

Question 2. Do you expect any of these will have high and/or statistically significant F_{ST} values? How high should F_{ST} be before you believe that locus might be under selection?

A variety of responses are possible here. There are no set guidelines for how high F_{ST} should be before you infer significant structure, and no set guidelines for how high F_{ST} should be for a given locus before you infer that locus should be under selection. What F_{ST} value constitutes an outlier will depend on the underlying distribution of F_{ST} values in the data set at hand. Therefore the answer will vary from study to study. In our case, students should point out generally that these loci with allele frequency differences of more than 25% are the top contenders for being significant outliers and should have higher F_{ST} values than the other loci.

Multiple Comparisions in Genetic Inference

NGS studies, including reduced-representation genome sequencing studies, can identify 1,000s of SNPs. When we calculate F_{ST} or any other descriptive statistic for each SNP, we also want to determine if those values are *statistically significant*; that is, do they differ from our null hypothesis?

Recall from basic statistics that we define a *p-value* as the probability of observing our data if the null hypothesis is true. It is possible to obtain a data sample that is consistent with our null hypothesis but highly unlikely to have been observed. When we reject our null hypothesis even when our data are consistent with it, we are making a Type I error. To control for this possiblity of making the wrong conclusion about our data's consistency with the null hypothesis, we infer statistical significance when our p-values are lower than our threshold for making a Type I error. This threshold is called the α-level (not to be confused with the α_{ij} parameter in the Bayescan model!). The α level is typically 0.05.

Multiple comparisons increase the probability of a Type I error

Every time we make a comparison of a sample to a given null hypothesis with α=0.05, we have a 5% chance of rejecting our null hypothesis even when it is true. Over numerous comparisons to the same null hypothesis, that 5% chance of error compounds. If you make enough comparisons it is almost guaranteed that you will make a Type I error.

This is particularly problematic for analysis of individual loci in NGS studies where 1,000s of loci have been identified. We will make 2387 comparisons in conducting the F_{ST} outlier analysis.

Question 3. If we perform 2387 statistical tests – one for each SNP – and expect that 5% of our p-values would be statistically significant by chance alone, roughly how many Type I errors do you predict we would make?

5% of 2387 is 119.35.

Question 4. What might be the consequence of incorrectly inferring that some American dog tick SNPs have significantly high F_{ST} values when they really do not?

We identified evidence for tick subpopulations at two collection sites. This could be a result of drift or selection. If we infer incorrectly that selection drives this pattern we would be saying there are different selective pressures on ticks at the two collection sites. This would prompt follow-up studies to determine exactly the nature of that selection and the areas of the genome upon which selection was acting. A false positive result would mean all that follow-up work was predicated on an incorrect conclusion – a lot of time (and also money!) to waste purusing an unfruitful aveneue.

Simulated data show the consequence of multiple comparisons

We can draw 2387 samples from a Student's T distribution to represent possible test statistics obtained in statistical analysis. If all 2387 test statistics come from the same distribution they will all be consistent with the same null hypothesis and none should be statistically significant. But this is the equivalent of doing 2387 t-tests – will some yield *p-values* less than α? (Note: we do not do t-tests to analyze SNP data, but this simple example will highlight the dangers of multiple comparisons in even the simplest statistical context.)

```
# Draw sample data points (test statistics) from a simulated Student's T distribution.
# Degrees of freedom (df) is arbitrarily set.
simulated.values<-rt(length(bayescan.out[,1]), df=98)

# Determine the probability of making each observation randomly from
# the given distribution using the function dt().
# dt() gives the area under the curve for values
# equal to or more extreme than the observed value,
# otherwise known as the p-value.

simulated.p<-dt(simulated.values, df=1)
```

Question 5. In the case of the Student's T distribution we will state our null hypothesis as mean = 0. All of our simpulated p-values were calculated from data drawn from the same Student's T distribution with mean = 0 – all of our data should be consistent with the null hypothesis and none should be statistically significant. How many observations in the simulated data were made with less than 5% probability from our original distribution?

Everyone will have a slightly different answer here because the values are drawn randomly. But the point is that there are some number of observations with p-values less than 0.05, even when all the data come from the same distribution.

```
length(simulated.p[simulated.p<0.05])
```

```
## [1] 64
```

You can also graph the distributions as demonstrated in the lab guide:

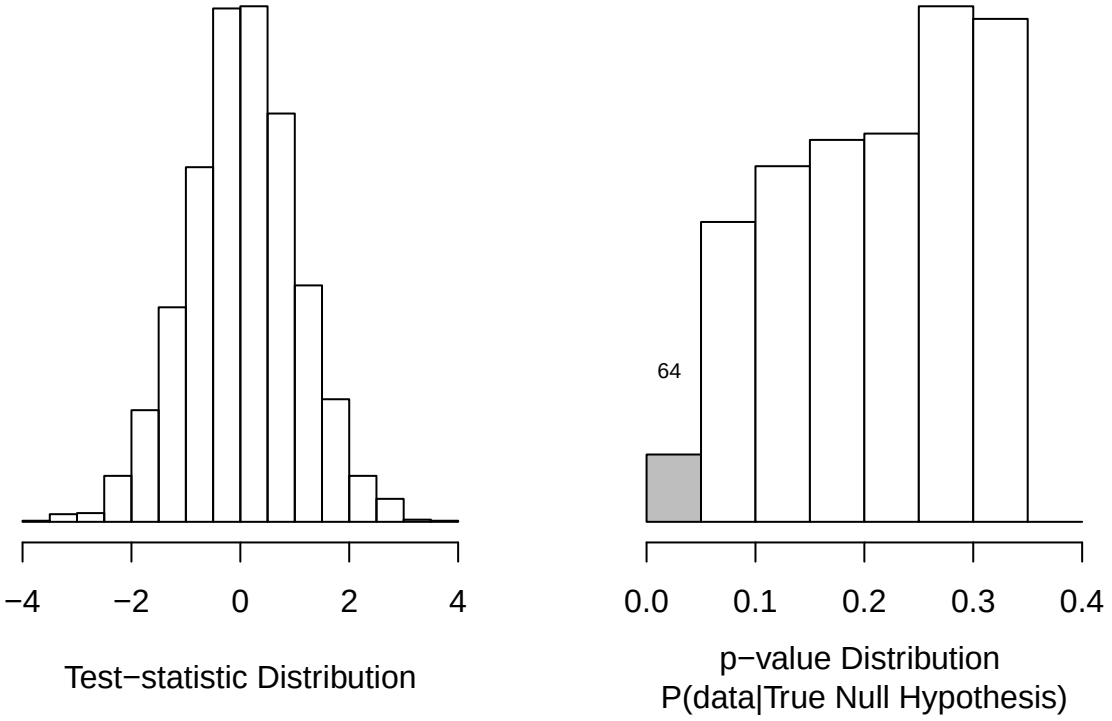
```
par(mfrow=c(1,2), mar=c(6,2,2,2))

# panel 1 shows the original values from the Student's T distribution
hist(simulated.values, main='', xlab='Test-statistic Distribution',
```

```
     las=1, freq=F, axes=F)
axis(side=1)

# panel 2 shows the distribution of p-values
# the gray area of the histogram represents the
# false-positive p-values < 0.05, which we would like to remove.
hist(simulated.p, main='', xlab='', ylab='', las=1,
     breaks=seq(0,0.4, 0.05), freq=F, col=c('gray', rep('white', 8)),
     axes=F)
axis(side=1)
mtext(side=1, line=3.5, text='p-value Distribution \nP(data|True Null Hypothesis)')

# length(simulated.p[which(simulated.p<0.05)]) counts
# the number of p-values less than alpha = 0.05
text(x=0.021, y=1.2, labels=length(simulated.p[which(simulated.p<0.05)]), cex=0.7)
```



Correcting multiple comparisons with *qvalues*.

We can detect and remove Type I errors in many ways, but the standard for genetic data is to perform a False Discovery Rate (FDR) correction by calculating a *q-value*. The *q-value* for a SNP will always be higher than the p-value, but if the allele frequencies for the SNP are truly significantly different from our null model of drift then the *q-value* should still be less than α.

We can illustrate the process of calculating *q-values* with our simulated data:

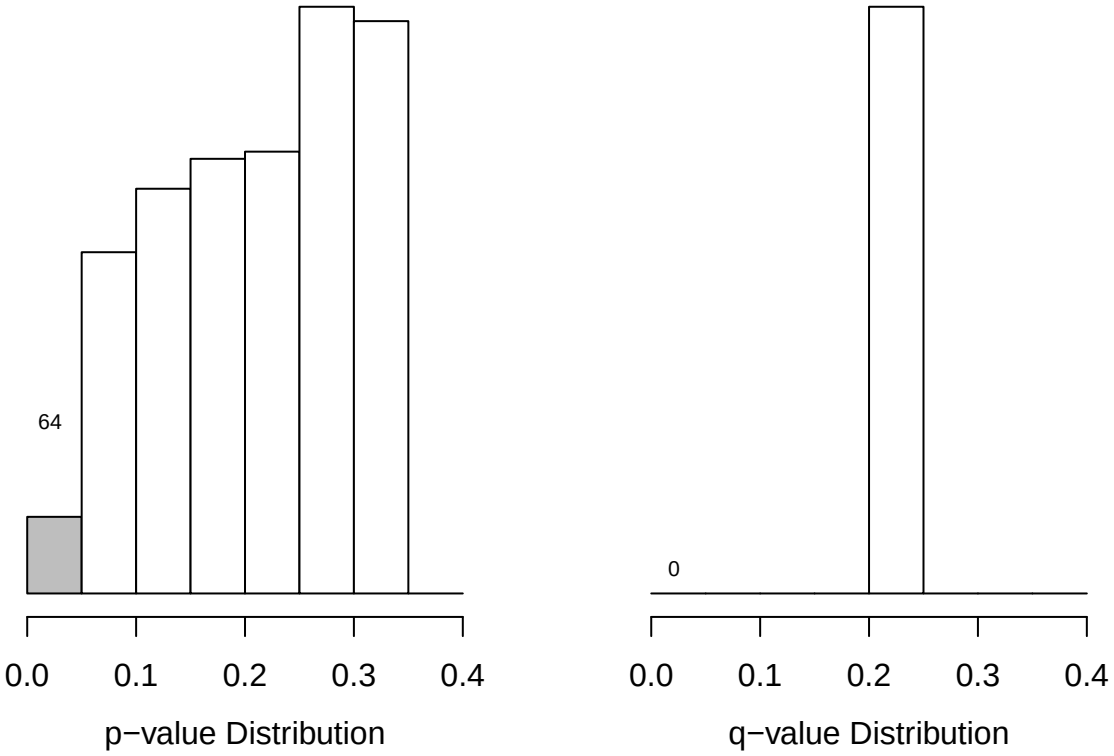
```
# perform the FDR correction with the qvalue() function:
simulated.q<-qvalue(simulated.p, lambda=0.2)
```

```
# plot the comparison
par(mfrow=c(1,2), mar=c(4,2,2,2))

# panel 1 shows the distribution of p-values
# the gray area of the histogram represents the
# false-positive p-values < 0.05, which we would like to remove.
hist(simulated.p, main='', xlab='', ylab='', las=1,
     breaks=seq(0,0.4,0.05), freq=F, col=c('gray', rep('white', 8)),
     axes=F)
axis(side=1)
mtext(side=1, line=2.5, text='p-value Distribution')
text(x=0.021, y=1.2, labels=length(simulated.p[which(simulated.p<0.05)]), cex=0.7)

# panel 2 shows the q-values (corrected p-values)
hist(simulated.q$qvalues, main='', ylab='', xlab='', las=1,
     breaks=seq(0,0.4,0.05), axes=F)
axis(side=1)
mtext(side=1, line=2.5, text='q-value Distribution')

# length(simulated.q$qvalues[which(simulated.q$qvalues<0.05)]) counts
# the number of q-values less than alpha = 0.05
text(x=0.021, y=100, labels=length(simulated.q$qvalues[which(simulated.q$qvalues<0.05)]), cex=0.7)
```



Question 6. How many simulated q-values are less than 0.05?

None of the q-values are statistically significant:

```
simulated.q$qvalues[simulated.q$qvalues<0.05]
```

```
## numeric(0)
```

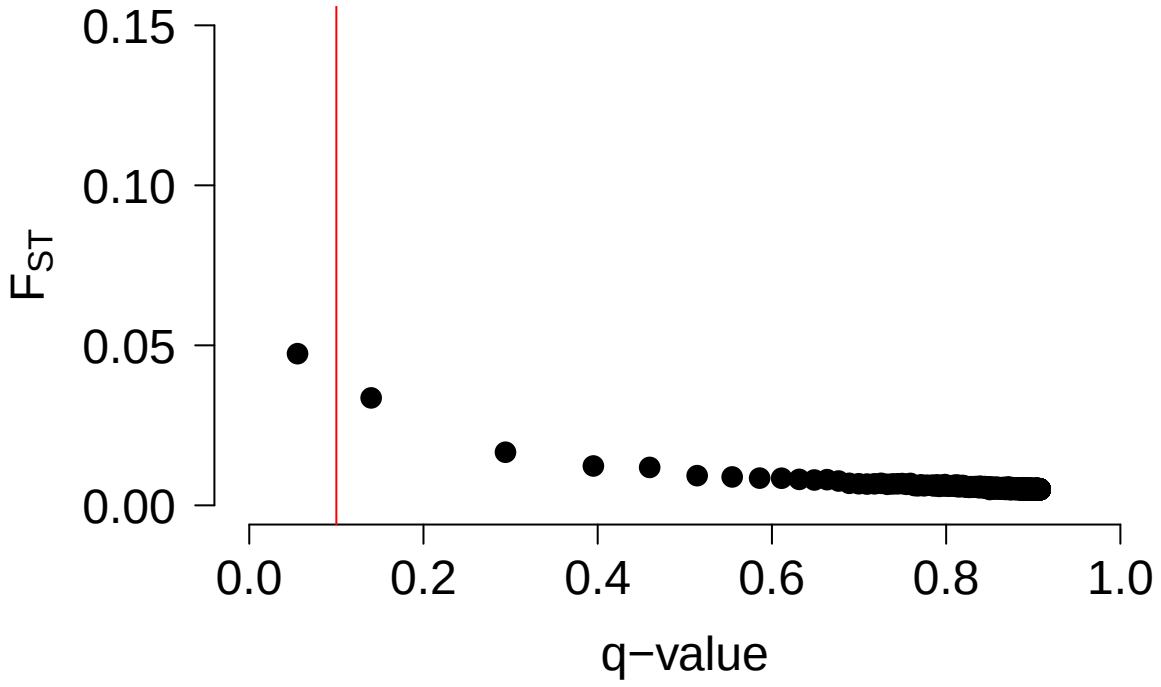
Bayescan Analysis & Results

Now that we understand how to properly control for the multiple comparisons that Bayescan will perform, it is time to look at the Bayescan results. Performing the F_{ST} calculation on the allele counts using Bayescan requires about one hour of supercomputer time. Instead of having all of you simultaneously access a supercomputer to run the same job, you are provided with the output of the Bayescan analysis on these data.

Bayescan also calculates the *q-value* for each F_{ST} estimate, so you will not need to calculate it manually (though now you know how should the need arise in the future).

Question 7. Plot the F_{ST} values from the Bayescan output against their corresponding *q-values* using the code below.

```
par(mar=c(5,6,4,2))
plot(bayescan.out$qval, bayescan.out$fst, pch=16, cex=1.5, cex.axis=1.5, xlim=c(0,1),
     ylim=c(0,0.15), bty='n', las=1, cex.lab=1.5, xlab='q-value', ylab='')
mtext(side=2, line=4, text=expression('F'['ST']), cex=1.5)
abline(v=0.1, col='red')
```



Question 8. How many SNPs have significant F_{ST} at the $\alpha = 0.1$ level? What are the corresponding F_{ST} values?

```
# Count the number significant at alpha = 0.1
sig<-bayescan.out$qval[bayescan.out$qval < 0.1]
length(sig)
```

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

```
# What is the corresponding FST value? (round to 3 digits)
fst<-bayescan.out$fst[bayescan.out$qval < 0.1]
round(fst, 3)
```

```
## [1] 0.047
```

Question 9. Modify the code from Question 8 to count the number of SNPs with significant F_{ST} values at the $\alpha = 0.05$ level.

```
# Count the number significant at alpha = 0.1
sig<-bayescan.out$qval[bayescan.out$qval < 0.05]
length(sig)
```

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

```
# What is the corresponding FST value? (round to 3 digits)
fst<-bayescan.out$fst[bayescan.out$qval < 0.05]
round(fst, 3)
```

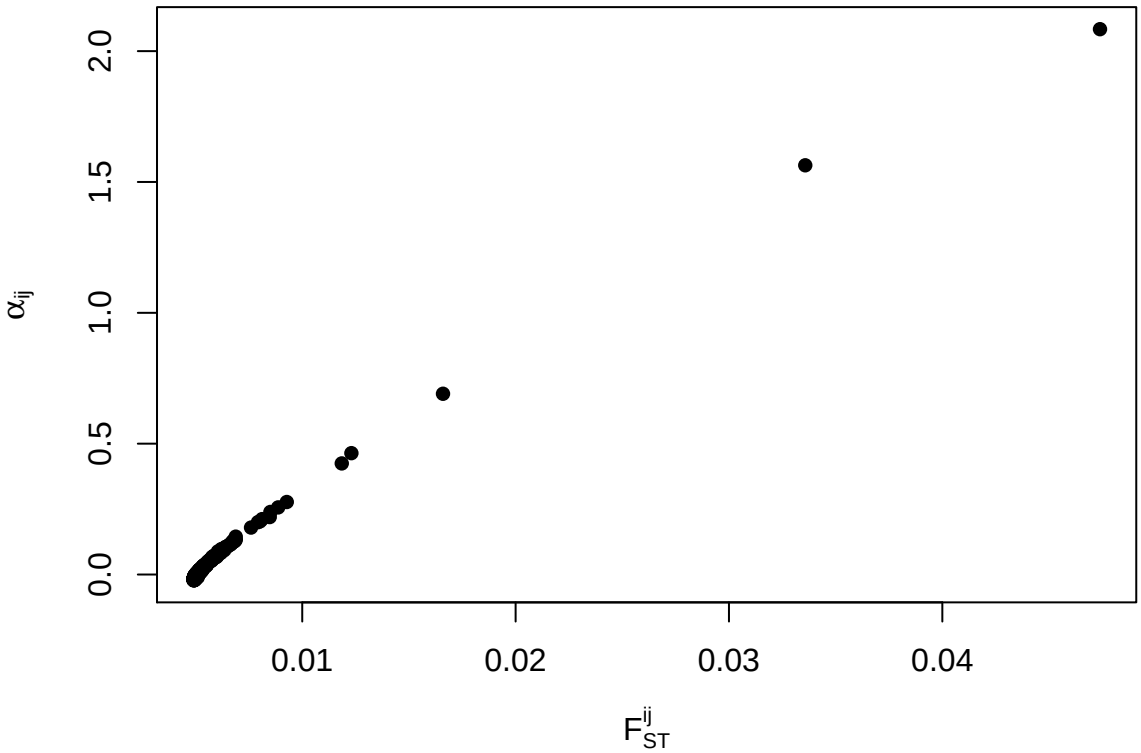
```
## numeric(0)
```

Question 10. Did your expectation about the number of SNPs with significant F_{ST} values based on the raw allele frequencies differ from the Bayescan results?

We observed some loci that had very different allele frequencies in our two subpopulations, so it would not have been unreasonable to expect some of those loci to have signficant outlier F_{ST} values. However, becaues Bayescan considers the underlying distribution of F_{ST} values across all loci, it is able to more rigorously evalute the data.

Question 11. In the introduction we described the Bayescan model as estimating parameters α_i and β_j to describe F_{ST}^{ij}. However, we have talked almost exclusively about the F_{ST}^{ij} estimates themselves. As you may have noticed, the Bayescan output also contains the α_{ij} estimate for each locus. Plot F_{ST}^{ij} vs α_{ij} and describe how these parameters are related. Is the relationship what you would expect?

```
par(mfrow=c(1,1), mar=c(5,5,2,2))
plot(bayescan.out$fst, bayescan.out$alpha, pch=16,
     xlab=expression('F'['ST']^'ij'), ylab=expression(alpha['ij']))
```



Higher values of F_{ST} correspond to higher locus-level α_{ij} parameters. Stare at the equations in the introduction of the lab guide to see how more extreme α_{ij} parameters will lead to higher F_{ST} estimates. In a more conceptual sense, α_{ij} captures variation that affects a given locus uniquely and differently from the rest of the population. As such, if α_{ij} is either positive or negative, something interesting is happening at locus *ij*. When α_{ij} is estimated to be 0, the locus-level effects are not important in the model and we infer no selection. (Note: positive values of α_{ij} indicate directional selection and negative values indicate balancing selection).

Question 12. Why do you think Bayescan does not report β_j?

The β_j parameter describes population-level effects, which are not directly relevant to assessing the drift vs. selection hypothesis. It is also possible to calculate β_j given the F_{ST} and α_{ij} values, so if you really wanted to know it you could figure it out.

Question 13. What do these results imply about the structuring of the two American dog tick subpopulations? Is it driven by genetic drift or natural selection?

It appears that at an $\alpha = 0.05$ level, none of the F_{ST} values are statistically significant outliers. While there was one F_{ST} value significant at $\alpha = 0.1$, that is not a stringent enough cut-off to infer selection. We can therefore conclude that the data are most consistent with the hypothesis that drift is the mechanism underlying the observed population structure.
