Problem Set 2

Gitanshu Munjal

Saturday, April 25, 2015

Problem 1

Using the Pairwise FST data construct a neighbor joining tree including distances for the Igbo, Yoruba, Kongo, Bamoun, Fulani populations (branches don't have to be perfectly to scale). (15 pts)

```
################################################
# Neighbor Joining Tree
################################################

# Clear workspace, set working directory, read in Fst matrix (reduced from paper)
rm(list=ls())
setwd("C:\\Users\\uglysweaters\\Desktop\\GGG201D\\ProblemSet2\\")

fstmatrix <- read.table("data\\tab1.txt",header=T,row.names=1,sep="\t")
fstmatrix <- as.matrix(fstmatrix)
fstmatrix
```

```
##         Igbo Yoruba Kongo Bamoun Fulani
## Igbo      NA  0.084 0.282  0.293  3.905
## Yoruba 0.084     NA 0.291  0.318  4.034
## Kongo  0.282  0.291    NA  0.175  3.770
## Bamoun 0.293  0.318 0.175     NA  3.996
## Fulani 3.905  4.034 3.770  3.996     NA
```

```
# Find pair with minimum distance (ignore diagnoal NAs)
which(fstmatrix == min(fstmatrix,na.rm=T),T)
```

```
##        row col
## Yoruba   2   1
## Igbo     1   2
```

```
# record distance between Igbo and Yoruba
y1 <-c("I-Y",min(fstmatrix,na.rm=T))
```

```
##########
# Step 2
##########

# Make new matrix with one less row and one less column, set up col and row names
fstmatrix2 <- matrix(NA,nrow(fstmatrix)-1,ncol(fstmatrix)-1)
rownames(fstmatrix2) <- c("I-Y","K","B","F")
colnames(fstmatrix2) <- c("I-Y","K","B","F")

# Calculate rlevant pairwise distances using first matrix as described in class notes
fstmatrix2[2:4,1] <- (fstmatrix[3:5,"Igbo"]+fstmatrix[3:5,"Yoruba"])/2
fstmatrix2[2:4,2] <- fstmatrix[3:5,3] #Distance same as first matrix
fstmatrix2[2:4,3] <- fstmatrix[3:5,4] #Distance same as first matrix
fstmatrix2[2:4,4] <- fstmatrix[3:5,5] #Distance same as first matrix
fstmatrix2
```

```
##         I-Y     K     B     F
## I-Y      NA    NA    NA    NA
## K    0.2865    NA 0.175 3.770
## B    0.3055 0.175    NA 3.996
## F    3.9695 3.770 3.996    NA
```

```
# Find pair with minimum distance (ignore diagnoal NAs)
which(fstmatrix2 == min(fstmatrix2,na.rm=T),T)
```

```
##   row col
## B   3   2
## K   2   3
```

```
# record distance between Bamoun and Kongo
y2 <-c("B-K",min(fstmatrix2,na.rm=T))

##########
# Step 3
##########

# Make new matrix with one less row and one less column, set up col and row names
fstmatrix3 <- matrix(NA,nrow(fstmatrix2)-1,ncol(fstmatrix2)-1)
rownames(fstmatrix3) <- c("I-Y","B-K","Fulani")
colnames(fstmatrix3) <- c("I-Y","B-K","Fulani")

# Calculate rlevant pairwise distances using first matrix as described in class notes
fstmatrix3["I-Y",2] <- (fstmatrix["Kongo","Igbo"] + fstmatrix["Kongo","Yoruba"] +
                          fstmatrix["Bamoun","Igbo"] +fstmatrix["Bamoun","Yoruba"])/4
fstmatrix3["B-K",1] <- (fstmatrix["Kongo","Igbo"] + fstmatrix["Kongo","Yoruba"] +
                          fstmatrix["Bamoun","Igbo"] +fstmatrix["Bamoun","Yoruba"])/4
fstmatrix3["Fulani",1] <- (fstmatrix["Fulani","Igbo"] + fstmatrix["Fulani","Yoruba"])/2
fstmatrix3["Fulani",2] <- (fstmatrix["Fulani","Bamoun"]+ fstmatrix["Fulani","Kongo"])/2
```

```r
fstmatrix3
```

```
##           I-Y   B-K Fulani
## I-Y        NA 0.296     NA
## B-K    0.2960    NA     NA
## Fulani 3.9695 3.883     NA
```

```r
# Find pair with minimum distance (ignore diagnoal NAs)
which(fstmatrix3 == min(fstmatrix3,na.rm=T),T)
```

```
##     row col
## B-K   2   1
## I-Y   1   2
```

```r
# record distance between Igbo-Yoruba and Bamoun-Kongo
y3 <-c("I-Y-B-K",min(fstmatrix3,na.rm=T))

##########
# Step 4
##########

# Make new matrix with one less row and one less column, set up col and row names
fstmatrix4 <- matrix(NA,nrow(fstmatrix3)-1,ncol(fstmatrix3)-1)
rownames(fstmatrix4) <- c("I-Y-B-K","Fulani")
colnames(fstmatrix4) <- c("I-Y-B-K","Fulani")

# Calculate rlevant pairwise distances using first matrix as described in class notes
fstmatrix4["Fulani",1] <- (fstmatrix["Fulani","Igbo"] + fstmatrix["Fulani","Yoruba"]
                           + fstmatrix["Fulani","Bamoun"] + fstmatrix["Fulani","Kongo"])/4
fstmatrix4["I-Y-B-K",2] <- (fstmatrix["Fulani","Igbo"] + fstmatrix["Fulani","Yoruba"]
                           + fstmatrix["Fulani","Bamoun"] + fstmatrix["Fulani","Kongo"])/4
fstmatrix4
```

```
##         I-Y-B-K  Fulani
## I-Y-B-K      NA 3.92625
## Fulani  3.92625      NA
```

```r
# Find pair with minimum distance (ignore diagnoal NAs)
which(fstmatrix4 == min(fstmatrix4,na.rm=T),T)
```

```
##         row col
## Fulani    2   1
## I-Y-B-K   1   2
```

```r
# record distance between Igbo-Yoruba-Bamoun-Kongo and Fulani
y4 <-c("I-Y-B-K-F",min(fstmatrix4,na.rm=T))

y <- rbind(y1,y2,y3,y4)
colnames(y) <- c("NeighborsJoined","TiptoMRCADistance")
```

```
data.frame(y)
```

```
##    NeighborsJoined TiptoMRCADistance
## y1             I-Y             0.084
## y2             B-K             0.175
## y3         I-Y-B-K             0.296
## y4       I-Y-B-K-F           3.92625
```

The tree based on the above presented data looks as follows:

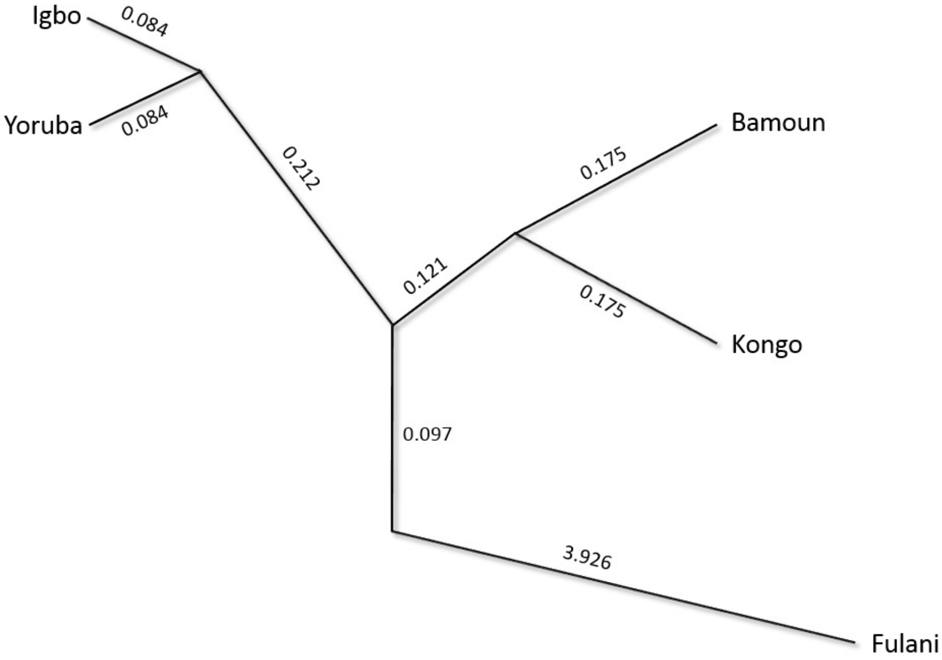


Figure 1: Neighbor Joining Tree based on data from Bryc et. al (2009).

Describe how you could use bootstrapping to assess the relationships presented in your tree. (5 pts)

We could pull the genotype data used for the study and resample n (where n = number of individuals used in the study from a given sub-population) individuals from each of the subpopulations with replacement multiple (say 1000) time. These data could then be used to generate trees as discussed in part a. We could then compare our tree from part a to these bootstrapped trees and assess the relationships in our tree by counting how many times a given relationship from our tree in part a occurs equivalently in the bootstrapped set of trees. Considering extreme findings for the sake of simplicity, if a relation is found to be the same 999 times one could be pretty confident in it compared to one that is found to be the same say only 30 out of 1000 times.

Problem 2

Do these results indicate that the individual is exactly 1/3 Scandinavian? In your answer, describe how maximum likelihood is used to estimate admixture across multiple loci. (8 pts)

The admixture model underlying the presented figure uses a maximum likelihood framework to make inferences. Before we interpret the presented data, lets take a look at the underlying analyses.

In a simple case, consider an individual originating from a single population. Genetic assignment may then be realized by genotyping that individual and determining the likelihood of the genotype under different populations based on allele frequencies at the genotyped loci in each of the populations under consideration. The population that presents the maximum-likelihood following such a calculation can naively be considered the population of origin. Under our simple case of origin from one population, we expect this signal to be rather strong (as shown in figure 1 for example).

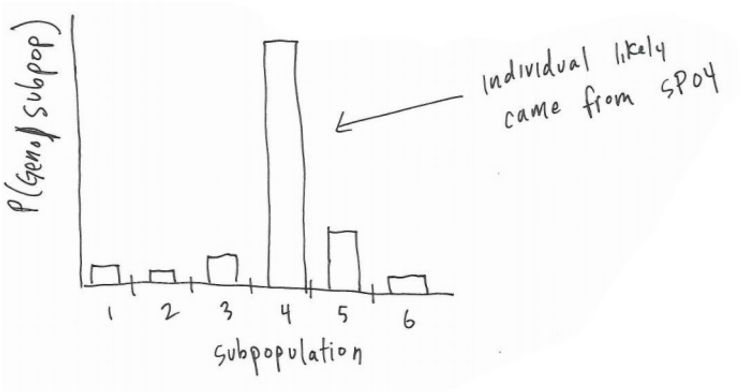


Figure 2: Likelihood Versus Subpopulation. Looks like the genotype/individual very likely came from subpopulation 4.

Now consider an individual with hybrid origins. The maximum-likelihood signal, as discussed above, from such an individual might not be as defining as presented in figure 1 because of the underlying hybrid origins (figure 2 for example). Under such a scenario, one expands the maximum-likelihood framework to account for admixture.

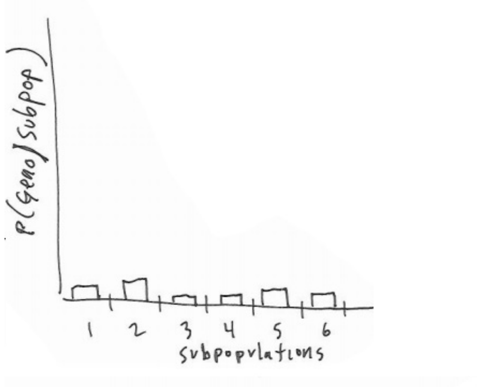


Figure 3: Likelihood Versus Subpopulation. Single underlying subpopulation of origin unlikely, possibly admixed genotype/individual

For simplicity, say we have a good suspicion (because sadly for this hypotheical species, there are only two populations! other reasons could be based in biology, geography, etc.) that the individual is a hybrid between two populations. Under such a scenario, we expect a fraction of this individual's genome (θ) to originate from population 1 and the remainder $(1 - \theta)$ from population 2. Based on this and a knowledge of allele frequencies at genotyped loci in the original two populations, we can develop probabilistic expectations for each genotyped loci as shown follows (for a single locus and two populations s1 and s2).:

$$\begin{aligned} P(GenotyeObserved \mid \theta) \; = \; & P[both\ alleles\ from\ s1] \; + \; P[allele1\ from\ s1\ \&\ allele2\ from\ s2] \; + \\ & P[allele1\ from\ s2\ \&\ allele2\ from\ s1] \; + \; [both\ alleles\ from\ s1] \end{aligned}$$

$$\begin{aligned} = \; & (\theta * f_{s1}) \quad (\theta * f_{s1}) \; + \; (\theta * f_{s1}) \quad ((1-\theta) * f_{s2})) \; + \\ & ((1-\theta) * f_{s2}) \;\; (\theta * f_{s1}) \; + \; ((1-\theta) * f_{s2}) \quad ((1-\theta) * f_{s2}) \end{aligned}$$

where,
θ is the fraction of the genome originating from subpopulation 1 and f_{s1} the frequency of the observed allele in subpopulation 1.

For a multi-locus genotype, the probability of the total genotype observed can then be calculated simply by multiplying the probabilities at each locus and can be represented by the following equation:

$$\prod_{i=1}^{n} P(G_i \mid \theta)$$

where,
G is the genotype at the i^{th} locus.

Next, we vary the value of our parameter of interest (θ; range:0-1) to maximize the above likelihood. The value of θ that results in the maximum likelihood is then our best candidate for the fraction of the genome originating from population 1.**These values are what is reported in the data shown**. Given the large number of loci used for determining the presented data, we can be sure that **it is very likely that this individual has 1/3 Scandinavian ancestry.**

What is one method you could use to assess confidence in these admixture estimates? (6 pts)

For thin slices of θ (say 0 to 1 in steps of 0.001 for example), I would determine the total genotype probability as described above, determine the variance of the determined probability distribution, and build a confidence interval around it. I would also be interested in visualizing and maybe reinforcing the findings by doing a PCA with the SNP set of the individual.

Why is accounting for genetic ancestry useful for studying the genetic basis of complex traits (such as GWAS)? (6 pts)

Accounting for genetic ancestry is useful in dissecting the genetic basis of complex traits as association studies based on diversity panels with uncharacterized population structure are extremely vulnerable to spurious associations leading to an inflation of false positives.

Diversity panels composed of mixed and/or admixed individuals with unknown proportions of multiple genetic origins could give rise to linkage disequilibrium between unlinked loci resulting in an inflation of false positive associtations that are not actually involved in phenotypic variation for the trait being investigated [1].

In case-control type disease association studies, spurious associations are likely if the frequency of disease varies across populations under consideration [2]. Consider an extreme scenario where 75% of the diseased individuals cases come from one population and two populations are under consideration for the disease association study. Under such a situation, one would expect to find lots of false positive alleles that associate with the phenotype purely because of their enriched frequency in one population.

References

[1] Mezmouk, S., Dubreuil, P., Bosio, M., Décousset, L., Charcosset, A., Praud, S., and Mangin, B. Effect of population structure corrections on the results of association mapping tests in complex maize diversity panels. *Theoretical and applied genetics 122*, 6 (2011), 1149–1160.

[2] Pritchard, J. K., Stephens, M., Rosenberg, N. A., and Donnelly, P. Association mapping in structured populations. *The American Journal of Human Genetics 67*, 1 (2000), 170–181.

Problem 3

How is genetic drift is affected by inital allele frequency and population size?

Defining a function

Based on the premise that expected change in allele frequency ($\Delta^{f}_{expected}$) for a given locus in a population of size n is equal to the summation from 0 to n of the product of the binomial probability of a given change and the absolute value of the magnitude of change in frequency over a generation, I define an R function that takes initial frequency and population size as input parameters to return the expected change in allele frequency as follows:

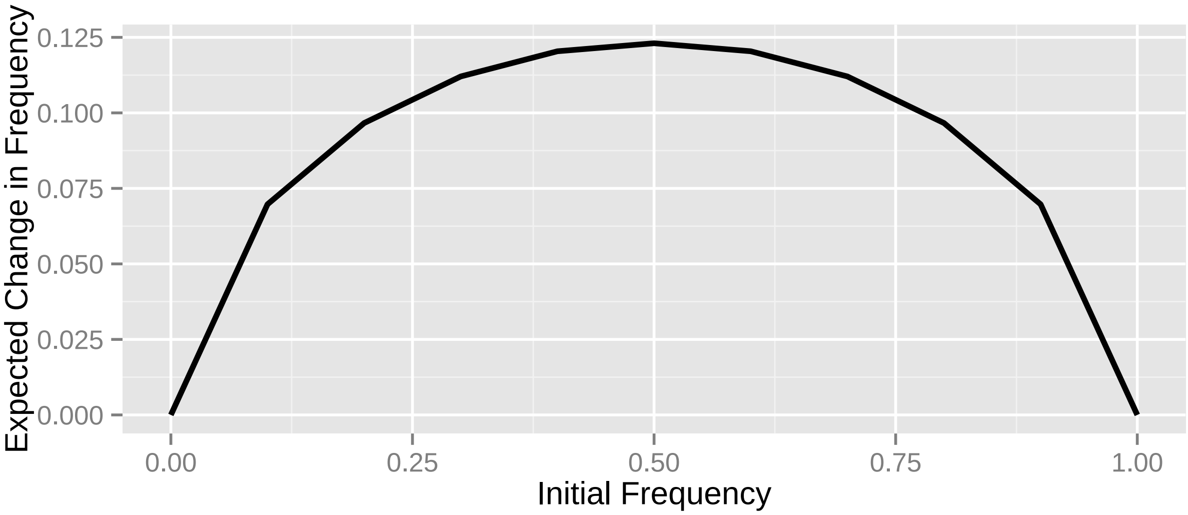
```r
efreqchange <- function(freqgen0,popsize){

  #empty matrix with 6 columns
  plotmatrix <- matrix(,,6)
  colnames(plotmatrix) <- c("nchosen","freqgen0","dbinom","freqgen1",
                            "deltaf","weighted")

  #column 1 = chosen n,
  #column 2 = initial frequency
  #column 3 = binomial probabilty of popsize choose n
  #column 4 = frequency in next generation = n/popsize
  for(nchosen in 0:popsize){

    roww <- c(nchosen, freqgen0, dbinom(nchosen,size=popsize,freqgen0),
              nchosen/popsize,NA,NA)
    plotmatrix <- rbind(plotmatrix,roww)
  }

  #column 5 = delta frequency = nextgenfrequency - initialfrequency
  #remove rows with NA by ordering and removing end of matrix
  plotmatrix <- plotmatrix[order(plotmatrix[,2]),]
  plotmatrix[,5] <- plotmatrix[,4] -plotmatrix[,2]
  plotmatrix <- plotmatrix[1:nrow(plotmatrix)-1,]


  #change delta to absolute delta
  for(i in 1:nrow(plotmatrix)){
    if(plotmatrix[i,5]<0){
      plotmatrix[i,5] <- -(plotmatrix[i,5])
    }
    else{plotmatrix[i,5] <- (plotmatrix[i,5])}
  }

  #column 6 = deltaf * binomprobability
  plotmatrix[,6] <- plotmatrix[,5]*plotmatrix[,3]

  #expected change in allele frequency based on premise
  e <- sum(plotmatrix[,6])
  return(e)

}
```

Using R or Excel, create a plot that shows how expected change in allele frequency for allele A changes depending on the frequency of A in generation 1 in populations of size of 2N=10 and 2N=100. (10 pts)

```
popsize <- 10
freqchange <- matrix(NA,11,2)
colnames(freqchange) <- c("InitialFrequency" , "ExpectedChange")

for(freqgen0 in seq(0,1,0.1)){

  y <- efreqchange(freqgen0,popsize)

  freqchange[(10*freqgen0)+1,1] <- freqgen0
  freqchange[(10*freqgen0)+1,2] <- y
}

(freqchange <- data.frame(freqchange))
```

```
##    InitialFrequency ExpectedChange
## 1               0.0     0.00000000
## 2               0.1     0.06973569
## 3               0.2     0.09663676
## 4               0.3     0.11206773
## 5               0.4     0.12039487
## 6               0.5     0.12304688
## 7               0.6     0.12039487
## 8               0.7     0.11206773
## 9               0.8     0.09663676
## 10              0.9     0.06973569
## 11              1.0     0.00000000
```

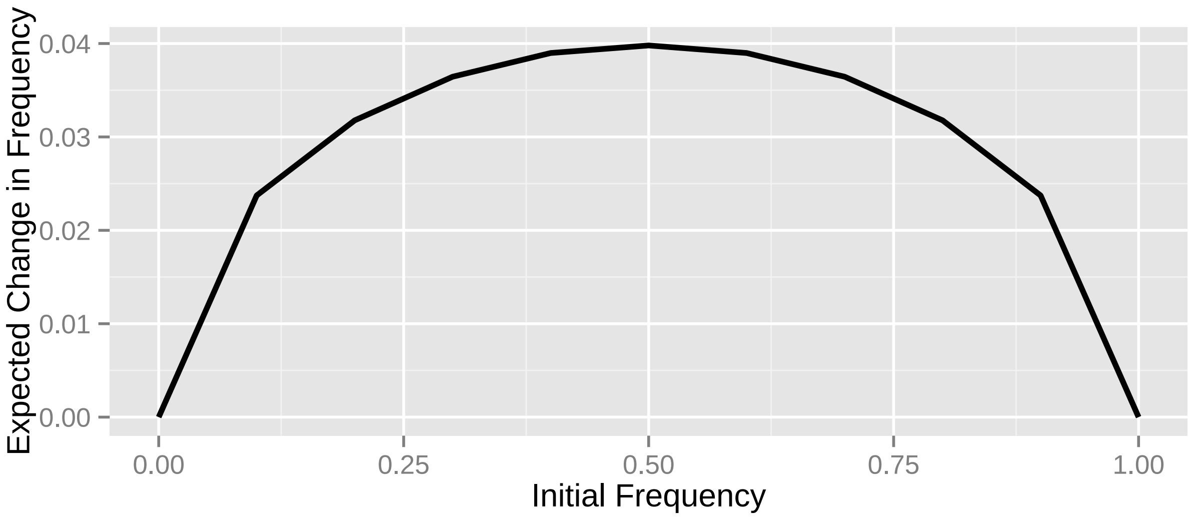
```
library(ggplot2)
ggplot(data=freqchange,aes(freqchange[,1],freqchange[,2])) +
        geom_line(size=1) +
        labs(y="Expected Change in Frequency",x="Initial Frequency")
```



```
popsize <- 100
freqchange <- matrix(NA,11,2)
colnames(freqchange) <- c("InitialFrequency" , "ExpectedChange")

for(freqgen0 in seq(0,1,0.1)){

  y <- efreqchange(freqgen0,popsize)

  freqchange[(10*freqgen0)+1,1] <- freqgen0
  freqchange[(10*freqgen0)+1,2] <- y
}


(freqchange <- data.frame(freqchange))
```

```
##    InitialFrequency ExpectedChange
## 1               0.0     0.00000000
## 2               0.1     0.02373576
## 3               0.2     0.03177607
## 4               0.3     0.03644922
## 5               0.4     0.03898519
## 6               0.5     0.03979462
## 7               0.6     0.03898519
## 8               0.7     0.03644922
## 9               0.8     0.03177607
## 10              0.9     0.02373576
## 11              1.0     0.00000000
```

```
library(ggplot2)
ggplot(data=freqchange,aes(freqchange[,1],freqchange[,2])) +
        geom_line(size=1) +
        labs(y="Expected Change in Frequency",x="Initial Frequency")
```

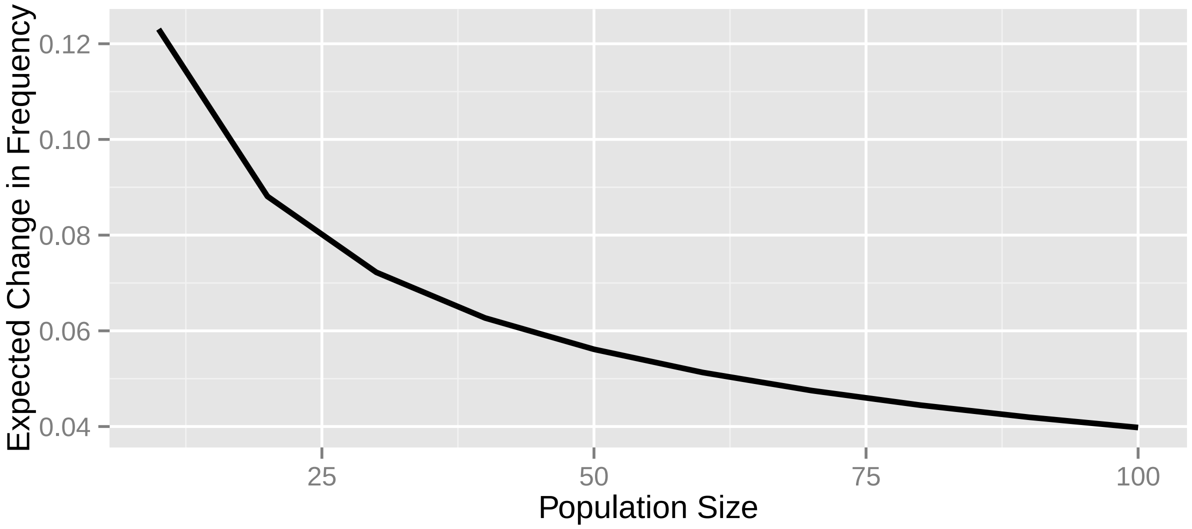


Using R or Excel, create a plot that shows how expected change in allele frequency for allele A changes depending on population size, starting with an allele frequency of f(A)=0.5 (10 pts)

```
freqgen0 <-0.5
freqchange <- matrix(NA,11,2)
colnames(freqchange) <- c("PopSize" , "ExpectedChange")
for(popsize in seq(10,100,10)){

  y <- efreqchange(freqgen0,popsize)
  freqchange[popsize/10,1] <- popsize
  freqchange[popsize/10,2] <- y
}

(freqchange <- data.frame(na.omit(freqchange)))
```

```
##    PopSize ExpectedChange
## 1       10     0.12304688
## 2       20     0.08809853
## 3       30     0.07223222
## 4       40     0.06268534
## 5       50     0.05613759
## 6       60     0.05128909
## 7       70     0.04751274
## 8       80     0.04446394
## 9       90     0.04193556
## 10     100     0.03979462
```

```
library(ggplot2)
ggplot(data=freqchange,aes(freqchange[,1],freqchange[,2])) +
        geom_line(size=1) +
        labs(y="Expected Change in Frequency",x="Population Size")
```



Problem 4

Sketch a neighbor joining tree based on the number of nucleotide differences for these three species. (5 pts)

```
###########
# Step 1
###########

#matrix with per site differences data
divmatrix <- matrix(NA,3,3)
colnames(divmatrix) <- c("A","B","C")
rownames(divmatrix) <- c("A","B","C")
divmatrix[,1] <- c(NA,6,14)/1000
divmatrix[,2] <- c(6,NA,17)/1000
divmatrix[,3] <- c(14,17,NA)/1000
divmatrix
```

```
##       A     B     C
## A    NA 0.006 0.014
## B 0.006    NA 0.017
## C 0.014 0.017    NA
```

```
#find pair with minimum per site differences
which(divmatrix == min(divmatrix,na.rm=T),T)
```

```
##   row col
## B   2   1
## A   1   2
```

```
#record number of per site differences
y1 <- c("A-B",min(divmatrix,na.rm=T))

###########
# Step 2
###########

#matrix reduced from step 1
divmatrix2 <- matrix(NA,2,2)
colnames(divmatrix2) <- c("A-B","C")
rownames(divmatrix2) <- c("A-B","C")
divmatrix2[,1] <- c(NA,sum(divmatrix[3,1:2])/2)
divmatrix2[,2] <- c(sum(divmatrix[3,1:2])/2,NA)
divmatrix2
```

```
##        A-B      C
## A-B     NA 0.0155
## C   0.0155     NA
```

```
#find pair with minimum per site differences
which(divmatrix2 == min(divmatrix2,na.rm=T),T)
```

```
##     row col
## C     2   1
## A-B   1   2
```

```
#record number of per site differences
y2 <- c("A-B-C",min(divmatrix2,na.rm=T))

y <- rbind(y1,y2)
colnames(y) <- c("NeighborsJoined","TiptoMRCADistance")
data.frame(y)
```

```
##    NeighborsJoined TiptoMRCADistance
## y1             A-B             0.006
## y2           A-B-C            0.0155
```

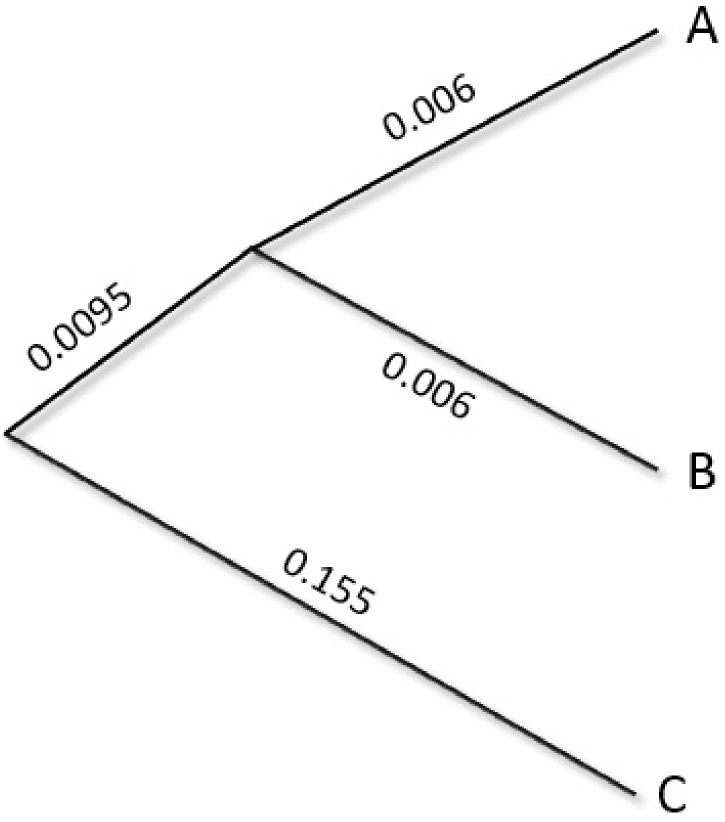


Figure 4: Neighbor Joining Tree based on data from Bryc et. al (2009).

Given the divergence time between A and B and the number of observed differences, what is the estimated mutation rate for these species? (5 pts)

```
abdivergetime <- 2000000     #divergence time between A and B in years
abdiff <- 6/1000             #per site differences between A and B
acdiff <- 14/1000            #per site differences between A and C
bcdiff <- 17/1000            #per site differences between B and C
gtime <- 3                   #generation time


abdivergegen <- abdivergetime/gtime     #divergence time between A and B in generations

mu = abdiff / (abdivergegen * 2)
mu
```

```
## [1] 4.5e-09
```

Thus, the mutation rate is 4.5 x 10^{-9} per site per generation

Assuming the same mutation rate for species C, what is the divergence time between A and C? What about between B and C? (5 pts)

```
acdivergegen = acdiff / (2*mu)
acdivergetime = acdivergegen * gtime
acdivergetime
```

```
## [1] 4666667
```

Thus, the divergence time between A and C is approximately 4.7 million years.

```
bcdivergegen = bcdiff / (2*mu)
bcdivergetime = bcdivergegen * gtime
bcdivergetime
```

```
## [1] 5666667
```

Thus, the divergence time between A and C is approximately 5.7 million years.

What is the estimated divergence time since the last common ancestor of A, B, and C? (5 pts)

```
abcdivergegen = 0.0155 / (2*mu)
abcdivergetime = abcdivergegen * gtime
abcdivergetime
```

```
## [1] 5166667
```

Thus, the divergence time since the last common ancestor between A, B, and C is approximately 5.2 million years.