Introduction to Population Genetics

International Statical Genetics Workshop

This practical runs on the workshop RStudio server.

First login to RStudio

```
https://workshop.colorado.edu/rstudio/
```

Copy the Practical document into your home directory using the following R commands.

```
setwd("~/.")
system("mkdir Day2")
system("cp /home/loic/2023/Practical_PopGen.html ~/Day2/.")
setwd("Day2")
```

Part 1: Visualize changes in allele frequencies

The R commands below define two functions: (1: generateNextGeneration()) generates genotypes for the next generation from a set of genotypes in the current population, and (2: sim), which uses (1) to simulate the evolution over g generations of a population with a constant sample size (N), a certain number m of SNPs segregating in the founding (or ancestral) population with a frequency (p). The sim function then plot the frequency of each of the m alleles over the course of g generations (each curve corresponds to a SNP).

```
generateNextGeneration <- function(x,n){
  m   <- ncol(x)
  N   <- nrow(x)
  iM  <- sample(1:N,n,replace = TRUE)
  iF  <- sample(1:N,n,replace = TRUE)
  xm  <- x[iM,]
  xf  <- x[iF,]
  xa  <- t(sapply(1:n,function(k) rbinom(m,1,prob=0.5*xm[k,])))
  xb  <- t(sapply(1:n,function(k) rbinom(m,1,prob=0.5*xf[k,])))
  xn  <- xa + xb
  return(xn)
}

sim <- function(N,  # sample size
                p,  # allele frequency in the founding population
                m,  # number of marker
                g){ # number of generations
  ## Simulate genotypes in the founding population
  x              <- do.call("cbind",lapply(1:m,function(j) rbinom(N,size=2,prob=p)))
  alleleFreq     <- matrix(NA,nrow=g+1,ncol=m)
  alleleFreq[1,] <- colMeans(x)/2
  for(gen in 1:g){
    # cat(paste0("\tGen #",gen,"\n"))
    x <- generateNextGeneration(x,N)
    alleleFreq[gen+1,] <- colMeans(x)/2
  }
  par(mar=c(5,5,3,2))
  matplot(1:(1+g),alleleFreq,type="l",lty=1,axes=FALSE,
          main=paste0("Size of the population: N = ",N),
          xlab="Number of generations",ylab="Allele frequencies",
          col=sample(colors(),m),log="x",cex.lab=1.2)
  axis(1);axis(2)
  abline(h=p,col="grey")
  return(alleleFreq)
}
```

Execute these two functions in your `R` terminal then run the following command.

```
t1 <- system.time( m50    <- sim(N=50  ,p=0.05,m=20,g=1000) )
```

Question 1a. What is the allele frequency after 1000 generations?

Question 1b. Do you see the same patterns if you run this command multiple times?

Question 1c. Try different sizes for the ancestral population (e.g., $N = 100$, or $N = 200$). How does it affect the frequency trajectories?

Question 1d. Try different allele frequency in the ancestral population (e.g., $p = 0.1$, or $p = 0.01$). How does it affect the frequency trajectories?

Question 1e. Run the following command. What can you say about the effect of the ancestral population size on the probability of fixation?

```
t1 <- system.time( m50    <- sim(N=50  ,p=0.05,m=20,g=1000) )
t2 <- system.time( m100   <- sim(N=100 ,p=0.5,m=20,g=1000) )
t3 <- system.time( m200   <- sim(N=200 ,p=0.5,m=20,g=1000) )
```

The objects `m50`, `m100` and `m200` contains the frequency trajectories of three simulated populations with sizes 50, 100 and 200 respectively. The following `R` commands count and visualize the proportion of fixed alleles over 1000 generations.

Run the following command. What can you can

```
f50  <- apply(m50*(1-m50)==0  ,1,mean)
f100 <- apply(m100*(1-m100)==0,1,mean)
f200 <- apply(m200*(1-m200)==0,1,mean)

par(mar=c(5,5,3,2))
matplot(cbind(f50,f100,f200),type="l",lwd=2,lty=1,
        xlab="Number of generations",axes=FALSE,cex.lab=2,
        ylab="Fraction of fixed allele frequencies")
axis(1);axis(2)
legend(800,0.25,legend=c("N=50","N=100","N=200"),
       box.lty=0,fill=1:3,horiz=FALSE,border=0,cex=1)
```

Part 2: Genetic drift creates more differentiation

The `R` commands below define a function called `FstSim`, which tracks $F_{ST}(t)$ between the ancestral population and the popuelation at time t. This functions reuses the `generateNextGeneration` function defined above. Input parameters for that function are size of the ancestral population (N), the allele frequency (p) in the ancestral population, the number m of SNPs segregating in the ancestral population, the number g of generations, the grow rate r (default $r = 0$, i.e. the population does not grow over time), and k an index for the replicate.

Run the following command to define the function.

```
FstSim <- function(N=100,  # sample size
                   p=0.5,  # allele frequency in the ancestral population
                   m=100,  # number of markers
                   g=500,  # number of generations
                   r=0.0,  # growth rate (r=0, no growth; r=0.01: 1% growth per generation)
                   k=0,    # replicate index
                   plotIt=TRUE){
  ## Simulate base population
  x              <- do.call("cbind",lapply(1:m,function(j) rbinom(N,size=2,prob=p)))
  alleleFreq     <- matrix(NA,nrow=g+1,ncol=m)
  alleleFreq[1,] <- colMeans(x)/2
  n <- N
  for(gen in 1:g){
    n <- round(n*(1+r))
    x <- generateNextGeneration(x,n)
    alleleFreq[gen+1,] <- colMeans(x)/2
  }
  cat(paste0("\tReplicate #",k,"\n"))
  fst <- apply(alleleFreq,1,function(x) mean((x-p)^2) / (p*(1-p)) )
  return(fst)
}
```

Question 2a. Under the Wright-Fisher model, we can predict that $F_{ST}(t) \approx 1 - e^{-t/(2N)}$. Run the following commands to verify how well this prediction works. If $N = 100$, then how many years do we have to wait for $F_{ST}(t)$ to reach 0.1 ? What if $N = 10,000$? (We assume that 1 generation ≈ 25 years).

```
r  <- 0.0    # growth rate of population size
m  <- 100    # number of SNPs
p0 <- 0.5    # allele frequency in the ancestral population
nG <- 500    # number of Generations
N  <- 100    # size of the ancestral population
dt <- 0:nG   # Time (in number of generations)

# Run simulation
Fst <- FstSim(N=N,p=p0,m=m,g=nG,r=r,k=0)
plot(dt,Fst,pch=19,cex=0.5,ylim=c(0,1),cex.lab=1.0,
     xlab="Number of generations",ylab="Fst between Current and Ancestral population",
     axes=FALSE);axis(1);axis(2)
ExpectedFst <- sapply(dt,function(t) 1-exp(-t/(2*N)))
lines(dt,ExpectedFst,col=2,lwd=2)
legend(0.5*nG,0.2,legend=c("Theoretical Expectation"),col=c(2,4),box.lty=0,lwd=2)
```

Question 2b. Change the growth rate of the population to $r = 0.01$ (1%). What do you see? [**WARNING: Setting growth rate beyond 1% will lead to extremely slow runs**].

Question 2c. Change the growth rate of the population to $r = -0.05$ *(-5%). What do you see?*