selection

June 5, 2024

1 Selection

Let’s consider the effect of selection on allele frequencies.

1.1 (1) Change in allele frequency

[ ]:

```
# changes in allele frequency
t <- 1:100000
f0 <- 0.01
s <- 0.00001
plot(f0/(f0+(1-s)^t*(1-f0)), ylab="frequency", xlab="generations")

# exponential distribution
lines(f0/(f0+exp(-s*t)*(1-f0)), type="l", col="red", lwd=2)
```

1.2 (2) Special cases

[ ]:

```
## directional selection
s <- 0.1 # selection coefficient

## additive
f <- rep(0,1000)
f[1] <- 0.01
for (t in 2:1000) f[t] <- f[t-1] + s*f[t-1]*(1-f[t-1])
plot(f, type="l", col="red")
legend("bottomright", col=c("red","black","blue"), legend=c("additive",␣
 ↪"dominant", "recessive"), lty=1, lwd=2)

## dominant
f <- rep(0,1000)
f[1] <- 0.01
for (t in 2:1000) f[t] <- f[t-1] + s*f[t-1]*(1-f[t-1])^2 / (1 - s*(1-f[t-1]^2))
lines(f, type="l", col="black", lwd=2)

## recessive
f <- rep(0,1000)
f[1] <- 0.01
```

```
for (t in 2:1000) f[t] = f[t-1] + (s*(f[t-1])^2*(1-f[t-1])) / (1 - s*(2*f[t-1]*(1-f[t-1]) + (1-f[t-1])^2))
lines(f, type="l", col="blue", lwd=2)
```

1.3 (3) Selection and drift

[ ]:

```
simulateTrajectory <- function(s, N, t=500, nrepl=100) {

        cat("2Ns =",2*N*s,"\n")

        # initialise frequencies
        fA <- matrix(NA, nrow=nrepl, ncol=t)
        fA[,1] <- 1/(2*N)

        # viability
        vAA <- 1
        vAa <- 1 - s
        vaa <- 1 - (2*s)

        for (r in 1:nrepl) {

                for (i in 2:t) {

                        # selection
                        fpA <- fA[r,i-1] * (2*vAA*fA[r,i-1] + (vAa*(1-fA[r,i-1]))) / (vAA*fA[r,i-1]^2 + 2*vAa*fA[r,i-1]*(1-fA[r,i-1]) + vaa*(1-fA[r,i-1])^2)

                        if (fpA <= 0) { fA[r,i:t] <- 0; break} # lost
                        if (fpA >= 1) { fA[r,i:t] <- 1; break} # fixed

                        # drift
                        fA[r,i] <- sum(sample(x=c(0,1), size=(2*N), replace=T, prob=c((1-fpA),fpA))) / (2*N)

                }

        }

        u <- 0
        if ((2*N*s) > -1) u <- 1/(2*N)
        if ((2*N*s) > 1) u <- 2*s

        cat("Lost = ", length(which(fA[,t]==0)), "\n")
        cat("Fixed = ", length(which(fA[,t]==1)), "\t (expected = ", (u*nrepl), ")\n")
        return(invisible(fA));

}
```

[ ]:

```
plotTrajectory <- function(fA, ylim=c(0,1), tlim=c(1,NA)) {
        cols <- colors()
        if (is.na(tlim[2])) tlim <- c(1,ncol(fA))
       ␣
 ↪plot(fA[1,],ylim=ylim,ty="l",xlim=tlim,col=cols[2],xlab="generations",ylab="frequency",lwd=
        for (i in 2:nrow(fA)) lines(fA[i,],type="l",col=cols[i+1],lwd=2)
}
```

[ ]:

```
plotTrajectory(simulateTrajectory(s=0.001, N=100, t=100, nrepl=100))
```

[ ]:

```
# what happens if we change "s" and "N"?
```