EEB459 group assignment 4

Group A (Meng, Rachel, Julia, Bianca)

Note that allele frequency p is assumed to be the frequency of A allele.

1) MakeHWfreq()

```
MakeHWfreq <- function(p){
    p_AA = p * p
    p_Aa = 2 * p * (1-p)
    p_aa = (1-p)^2
    return(c(p_AA, p_Aa, p_aa))
}

MakeHWfreq(0.1)
```

```
## [1] 0.01 0.18 0.81
```

```
MakeHWfreq(0.5)
```

```
## [1] 0.25 0.50 0.25
```

```
MakeHWfreq(0.9)
```

```
## [1] 0.81 0.18 0.01
```

2) DoDrift()

```
DoDrift <- function(popSize, EGFvec){
    # sample 5 times at this generation
    prob_vector <- rmultinom(1, popSize, EGFvec)
    return(prob_vector)
}

# a
popSize = 100
EGFvec = c(0.5, 0, 0.5)
replicate(5, DoDrift(popSize, EGFvec), simplify = T)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   40   52   48   56   53
## [2,]    0    0    0    0    0
## [3,]   60   48   52   44   47
```

```
# b
popSize = 10^6
EGFvec = c(0.5, 0, 0.5)
replicate(5, DoDrift(popSize, EGFvec), simplify = T)
```

```
##        [,1]   [,2]   [,3]   [,4]   [,5]
## [1,] 500268 499619 499352 500634 500120
## [2,]      0      0      0      0      0
## [3,] 499732 500381 500648 499366 499880
```

```
# c
popSize = 100
EGFvec = MakeHWfreq(0.5)
replicate(5, DoDrift(popSize, EGFvec), simplify = T)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   19   17   28   20   20
## [2,]   62   55   54   55   50
## [3,]   19   28   18   25   30
```

```
# d
popSize = 10^6
EGFvec = MakeHWfreq(0.5)
replicate(5, DoDrift(popSize, EGFvec), simplify = T)
```

```
##        [,1]   [,2]   [,3]   [,4]   [,5]
## [1,] 250391 250553 249910 249795 250215
## [2,] 499431 499622 500198 500145 499957
## [3,] 250178 249825 249892 250060 249828
```

Results

The differences of the result make sense. We expected that drift should be stronger in small populations, and that smaller populations experienced more variance in allele frequency change due to drift, because of the equation V(p')=(p(1-p))/2N. Since 2N is in the denominator, a bigger population would decrease the change in variance. We observed the larger the population size is, the less the numbers of each genotype change or fluctuate (i.e. the closer they are to the expected values) and this is consistent with what we have learned about drift. This pattern is consistent when using different expected genotype frequencies. A and B have the same expected genotype frequencies, but B has a significantly larger population, so it makes sense that B's observed genotype frequencies are much closer to the expected frequencies. C and D have the same p frequency, meaning their expected genotype frequencies are also equivalent, but again D has a larger population than C. As such, D's observed genotype frequencies are much closer to the expected than C's.

3) DoSelection()

```
DoSelection <- function(h, s, GNvec){
    # genotype frequencies
    Geno_freq <- GNvec/sum(GNvec)
    # selection model
    W_AA = 1 + s
    W_Aa = 1 + h * s
    W_aa = 1
    W_vector <- c(W_AA, W_Aa, W_aa)
    W_mean = sum(W_vector * Geno_freq)
    s_Geno_freq <- Geno_freq * W_vector / W_mean
    # allele frequency
    p = s_Geno_freq[1] + 0.5 * s_Geno_freq[2]
    return(p)
}

# a
h=0
s=0.1
GNvec=c(0,333,333)
DoSelection(h, s, GNvec)
```

```
## [1] 0.25
```

```
# b
h=0.5
s=0.1
GNvec=c(0,333,333)
DoSelection(h, s, GNvec)
```

```
## [1] 0.2560976
```

```
# c
h=1
s=0.1
GNvec=c(333,666,333)
DoSelection(h, s, GNvec)
```

```
## [1] 0.5116279
```

```
# d
h = 1
s = 0.1
GNvec = DoDrift(popSize = 100, EGFvec = MakeHWfreq(0.5))
replicate(5, DoSelection(h, s, GNvec) , simplify = T)
```

```
## [1] 0.5167442 0.5167442 0.5167442 0.5167442 0.5167442
```

```
# e
h = 1
```

```
s = 0.1
GNvec = DoDrift(popSize = 10^6, EGFvec = MakeHWfreq(0.5))
replicate(5, DoSelection(h, s, GNvec) , simplify = T)
```

```
## [1] 0.511762 0.511762 0.511762 0.511762 0.511762
```

Results

a vs. b: In both a) and b), there are only two genotypes Aa and aa, the initial allele frequency of A is p=333/666*0.5=0.25. In a), h=0, s=0.1, which makes it so the fitness of Aa and aa is the same. After selection, the genotype frequencies don't change (because their relative fitnesses are equal), thus allele frequency p doesn't change (p=0.25). In b), h=0.5,s=0.1, so the fitness of Aa genotype is larger than aa by hs=0.05. After selection, the frequency of Aa increases and correspondingly, allele frequency p increases (p=0.256). It makes sense that the frequency of p increases, as the only genotype carrying the A allele has increased fitness over the aa genotype.

c vs. d vs. e: In all three cases, the fitness of genotypes are WAA=1.1, WAa=1.1, Waa=1, so, we expect allele frequency p to increase after selection. After selection, c) and e) have similar values which makes sense because e) has such a high population size that it converges more on true HWE during sampling despite the drift. In d), genetic drift deviates allele and genotype frequencies from the expected values; because the population size is smaller, there is more stochasticity in starting frequency so we see more variable outcomes after selection compared to populations with no drift or even larger populations with drift.

Overall we are seeing that drift makes selection less clear to see, and that drift has bigger implications in a smaller population, even if they are all experiencing the same selection.

4) DoOneFullGeneration()

```
DoOneFullGeneration <- function(h, s, popSize, p){
    EGFvec = MakeHWfreq(p)
    GNvec = DoDrift(popSize, EGFvec)
    return(DoSelection(h, s, GNvec))
}

# a
h=1
s=0.1
n=100
p=0.5
replicate(5, DoOneFullGeneration(h, s, n, p), simplify = T)
```

```
## [1] 0.4720149 0.5264870 0.5233209 0.5444033 0.5177074
```

```
# b
h=1
s=0.1
n=10^6
```

```
p=0.5
replicate(5, DoOneFullGeneration(h, s, n, p), simplify = T)
```

```
## [1] 0.5109142 0.5118574 0.5110973 0.5113414 0.5115927
```

5) DoManyGenerations()

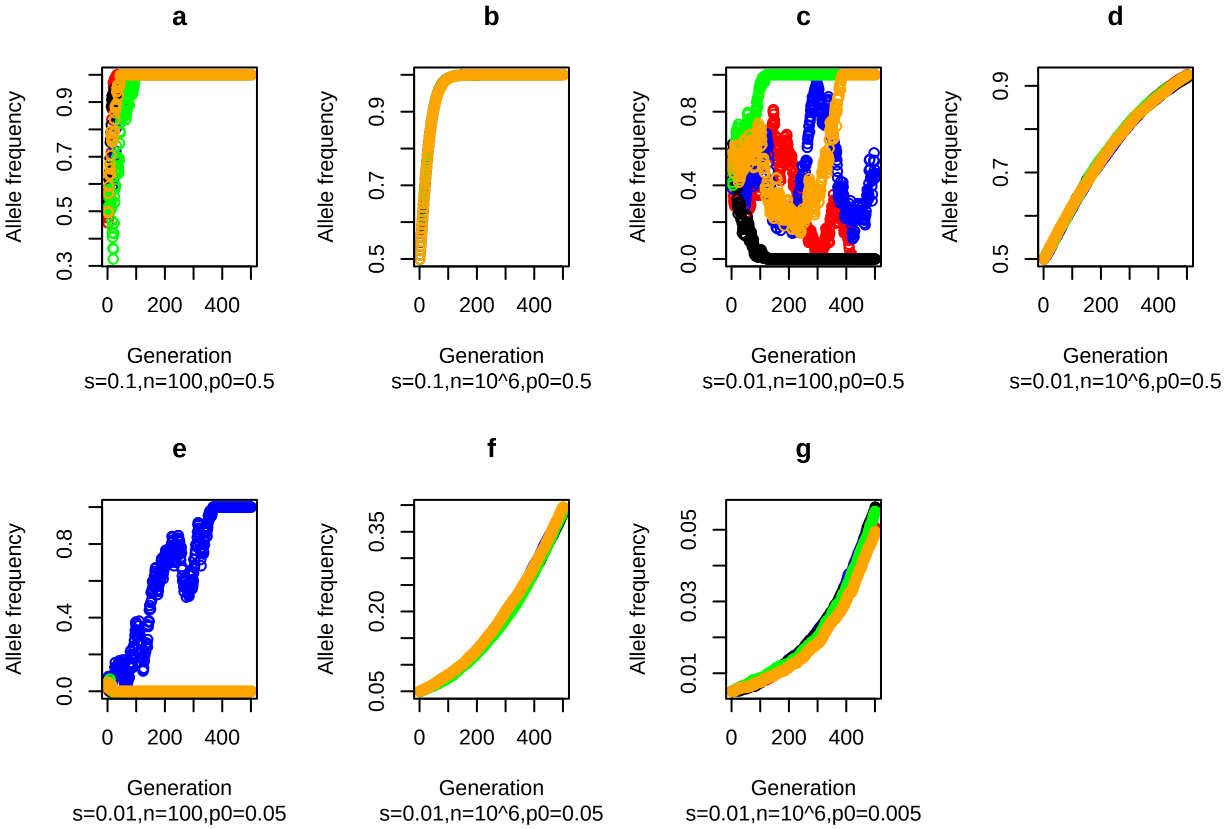
```
DoDrift <- function(popSize, EGFvec){
    # sample once at each generation
    prob_vector <- rmultinom(1, popSize, EGFvec)
    return(prob_vector*popSize)
}

DoOneFullGeneration <- function(h, s, popSize, p){
    EGFvec = MakeHWfreq(p)
    GNvec = DoDrift(popSize, EGFvec)
    return(DoSelection(h, s, GNvec))
}

DoManyGenerations <- function(h, s, popSize, p0, g){
    p_list <- c(p0)
    for(i in 1:g){
        p<-DoOneFullGeneration(h, s, n, p0)
        p_list<-append(p_list, p, after = length(p_list))
        p0<-p
    }
    return(p_list)
}

# a
h=0.5
s=0.1
n=100
p0=0.5
g=500
outputa<-c()
for (i in 1:5){
outputa<-append(outputa, DoManyGenerations(h, s, n, p0, g), after = length(outputa))}
# b
h=0.5
s=0.1
n=10^6
p0=0.5
g=500
outputb<-c()
for (i in 1:5){
```

```
outputb<-append(outputb, DoManyGenerations(h, s, n, p0, g), after = length(outputb))}
# c
h=0.5
s=0.01
n=100
p0=0.5
g=500
outputc<-c()
for (i in 1:5){
outputc<-append(outputc, DoManyGenerations(h, s, n, p0, g), after = length(outputc))}
# d
h=0.5
s=0.01
n=10^6
p0=0.5
g=500
outputd<-c()
for (i in 1:5){
outputd<-append(outputd, DoManyGenerations(h, s, n, p0, g), after = length(outputd))}
# e
h=0.5
s=0.01
n=100
p0=0.05
g=500
outpute<-c()
for (i in 1:5){
outpute<-append(outpute, DoManyGenerations(h, s, n, p0, g), after = length(outpute))}
# f
h=0.5
s=0.01
n=10^6
p0=0.05
outputf<-c()
for (i in 1:5){
outputf<-append(outputf, DoManyGenerations(h, s, n, p0, g), after = length(outputf))}
# g
h=0.5
s=0.01
n=10^6
p0=0.005
g=500
outputg<-c()
for (i in 1:5){
outputg<-append(outputg, DoManyGenerations(h, s, n, p0, g), after = length(outputg))}
```



Results

a vs. b: Fitness of AA and Aa are higher than aa, allele A will go to fixation eventually, if population size is smaller, there are more fluctuations in allele frequencies until fixation.

a vs. c as well as b vs. d: Selection is weaker (hs is smaller) in c & d compared to a & b. With lower selection: it takes longer for allele A to go to fixation in the large population, and in the small population you see many more different outcomes (sometimes it goes to fixation at p=0, sometimes at p=1, one simulation showed after 500 generations p was being maintained between 0.1-0.8 fluctuating).

e vs. f vs. g: e has a smaller population size, there are more fluctuations in allele frequencies. The selective advantage is also small (hs=0.005), meaning drift can overwhelm the selection, and the A allele could be lost or fixed. f and g have larger population size, and are under weak positive selection so the A allele slowly goes to fixation.

6) DoManyGenerationsV2()

```
DoManyGenerationsV2 <- function(h, s, popSize, p0, gmax){
    i=0
    while(i < gmax && p0 != 1 && p0 != 0){
        p0<-DoOneFullGeneration(h, s, n, p0)
        i = i + 1
    }
    return(c(i, p0))
}

# a
h=0.5
s=0.01
n=100
p0=0.5
gmax=500
replicate(5, DoManyGenerationsV2(h, s, n, p0, gmax), simplify = 5)
```

```
##      [,1]        [,2] [,3] [,4] [,5]
## [1,]  263 500.0000000  142  270  359
## [2,]    1   0.9153496    1    1    1
```

```
# b
h=0.5
s=0.01
n=100
p0=1/200
gmax=500
replicate(5, DoManyGenerationsV2(h, s, n, p0, gmax), simplify = 5)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    2    2    1    1
## [2,]    0    0    0    0    0
```

7) NewMutationManyTimes()

```
DoManyGenerations <- function(h, s, popSize, p0, g){
    p_list <- c(p0)
    for(i in 1:g){
        p<-DoOneFullGeneration(h, s, n, p0)
        p_list<-append(p_list, p, after = length(p_list))
        p0<-p
    }
    return(p_list)
}

NewMutationManyTimes <- function(h, s, popSize, m){
    p0 = 1/(2*popSize)
    p_list <- c()
    for (i in 1:m){
        p_final = DoManyGenerationsV2(h, s, n, p0, gmax=2000)[2]
        p_list <- append(p_list, p_final, after = length(p_list))
    }
    return(p_list)
}

# m=1000
m = 500000
# a
h = 0.5
s = 0
popSize = 1000
sim_a <- NewMutationManyTimes(h, s, popSize, m)
sum(sim_a != 0)/m
```

```
## [1] 0.000482
```

```
mean(subset(sim_a, sim_a!= 0))
```

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

```
# b
h = 0.5
s = 0.001
popSize = 1000
sim_b <- NewMutationManyTimes(h, s, popSize, m)
sum(sim_b != 0)/m
```

```
## [1] 0.000586
```

```
mean(subset(sim_b, sim_b!= 0))
```

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

```
# c
h = 0.5
s = 0.01
popSize = 1000
sim_c <- NewMutationManyTimes(h, s, popSize, m)
sum(sim_c != 0)/m
```

```
## [1] 0.001114
```

```
mean(subset(sim_c, sim_b!= 0))
```

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

```
# d
h = 0.5
s = 0.001
popSize = 10^5
sim_d <- NewMutationManyTimes(h, s, popSize, m)
sum(sim_d != 0)/m
```

```
## [1] 1e-05
```

```
mean(subset(sim_d, sim_d!= 0))
```

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

```
# e
h = 0.5
s = 0.01
popSize = 10^5
sim_e <- NewMutationManyTimes(h, s, popSize, m)
sum(sim_e != 0)/m
```

```
## [1] 1e-05
```

```
mean(subset(sim_e, sim_e!= 0))
```

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

Results

For a), in one simulation, we found the proportion of simulations in which the allele was not lost was 0.000484. The probability of fixation for a neutral allele is 1/2N, which in this case is 1/2000 or 0.0005. The number in this simulation is smaller than the expected value but extremely close, supporting neutrality.

For d), here the lower limit would be the fixation probability of a neutral mutation, which is equal to 1/2N. In d) the population size is 10^5, so the lower limit probability of fixation is 1/(2*10^5), which equals **0.000005**.

The upper limit could be calculated by 1-Pr(Lost), since we want the probability of any outcome other than Pr(Lost) (possible outcomes are fixation or p>0 after 2000 generations) and these are mutually exclusive. Now we need to define what Pr(Lost) equals. Pr(Lost)=1-Pr(est). Next, we will determine what Pr(est) is. The model assumes the allele is only present in heterozygotes (that is, rare).

$Ppoisson(K) = e^{-\mu}\frac{\mu^k}{k!}$, where μ is 1+s in the Haldane model.

In our example, μ = 1+hs. If we plug that into what's in the notes:

$1 - P_{est} = e^{-(1+hs)} e^{((1+hs)(1-P_{est}))}$

Assume s is very small, and take a 2nd order taylor series approximation:

$1 - P_{est} = 1 - P_{est} + \left(\frac{P_{est}^2}{2} - P_{est}sh\right) + O\left(\varepsilon^3\right)$

Solving for P_{est}, $\frac{P_{est}^2}{2} = P_{est}sh$, we have $P_{est} \approx 2hs$

Now that we know what Pr(est) is, we can substitute it into the model. Pr(Lost)=1-2hs

So the upper limit is: 1- (1-2hs) = 2hs. The upper limit fixation probability would be 2hs, in d) this would be 2 * 0.5 * 0.001= **0.001**.

The fixation probability predicted by Haldane is 2s, which would give us an upper limit of 0.002 and which is double the limit that we just calculated. This difference is due to the different fitness models we used: in Haldane's model he used a fitness model of Waa= 1, WAa= 1+s, WAA= 1+2s, while in our simulations we used a fitness model of Waa=1, WAa=1+(h*s), WAA=1+s. This suggests that by adding heritability, this may decrease the upper limit of fixation probability. By including a dominance coefficient of 0.5, it implies that the fitness of the heterozygote lies between the fitness of both homozygotes. This decreases the upper limit of fixation probability because h=0.5 implies a codominant or incomplete dominant environment, and an allele with this pattern of dominance is more likely to be lost than a fully dominant allele. In Haldane's model, only s is included in the result, but our result would be equivalent to Haldane's if h=1 (i.e. if one allele was dominant over the other). This shows that if there is a full dominant recessive relationship between alleles, the fixation probability of the dominant allele will be increased.