EEB459 group assignment 4

Meng

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(5, popSize, EGFvec)
    return(prob_vector*popSize)
}

# a
popSize = 100
EGFvec = c(0.5, 0, 0.5)
DoDrift(popSize, EGFvec)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] 4400 4900 5100 4900 5000
## [2,]    0    0    0    0    0
## [3,] 5600 5100 4900 5100 5000
```

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

```
##             [,1]       [,2]  [,3]        [,4]       [,5]
## [1,] 5.0078e+11 5.0012e+11 5e+11 5.00654e+11 5.0033e+11
## [2,] 0.0000e+00 0.0000e+00 0e+00 0.00000e+00 0.0000e+00
## [3,] 4.9922e+11 4.9988e+11 5e+11 4.99346e+11 4.9967e+11
```

```
# c
popSize = 100
EGFvec = MakeHWfreq(0.5)
DoDrift(popSize, EGFvec)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] 3500 2400 2000 2100 2200
## [2,] 4200 5400 4600 4800 5300
## [3,] 2300 2200 3400 3100 2500
```

```
# d
popSize = 10^6
EGFvec = MakeHWfreq(0.5)
DoDrift(popSize, EGFvec)
```

```
##              [,1]        [,2]        [,3]        [,4]        [,5]
## [1,] 2.50758e+11 2.50069e+11 2.50559e+11 2.49352e+11 2.50264e+11
## [2,] 4.99040e+11 4.99965e+11 4.99422e+11 4.99879e+11 4.99410e+11
## [3,] 2.50202e+11 2.49966e+11 2.50019e+11 2.50769e+11 2.50326e+11
```

Results

The differences of the result make sense. The larger the population size is, the less the numbers of each genotype change or fluctuate, the closer they are to the expected values. This pattern is consistent when using different expected genotype frequencies.

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)
```

```r
    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
```

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

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

```r
# GNvec as a 2d matrix
DoSelection2 <- function(h, s, GNvec){
    # genotype frequencies
    Geno_freq <- GNvec/colSums(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 = colSums(W_vector * Geno_freq)
    s_Geno_freq <- W_vector *  Geno_freq / W_mean
    # allele frequency
    p = s_Geno_freq[1,] + 0.5 * s_Geno_freq[2,]
    return(p)
}

# test example, to be deleted
# GNvec = DoDrift(popSize = 100, EGFvec = MakeHWfreq(0.5))
# Geno_freq<-GNvec/colSums(GNvec)
# colSums(Geno_freq)
# W_vector<-c(1.1, 1.1, 1)
# W_mean = colSums(W_vector * Geno_freq)
# s_Geno_freq <- W_vector *  Geno_freq / W_mean
# colSums(s_Geno_freq)
# p = s_Geno_freq[1,] + 0.5 * s_Geno_freq[2,]
# c
h=1
s=0.1
GNvec=c(333,666,333)
DoSelection(h, s, GNvec)
```

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

```r
# d
h = 1
s = 0.1
GNvec = DoDrift(popSize = 100, EGFvec = MakeHWfreq(0.5))
GNvec/colSums(GNvec)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] 0.29 0.24 0.31 0.24 0.27
## [2,] 0.51 0.52 0.48 0.51 0.51
## [3,] 0.20 0.24 0.21 0.25 0.22
```

```r
DoSelection2(h, s, GNvec)
```

```
## [1] 0.5560581 0.5108875 0.5615855 0.5046202 0.5361851
```

```r
# e
h = 1
s = 0.1
GNvec = DoDrift(popSize = 10^6, EGFvec = MakeHWfreq(0.5))
DoSelection2(h, s, GNvec)
```

```
## [1] 0.5116724 0.5117351 0.5113152 0.5115085 0.5121041
```

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, the fitness of Aa and aa is the same, after selection the genotype frequencies don't change, thus allele frequency p doesn't change (p=0.25). In b), h=0.5,s=0.1, the fitness of Aa genotype is larger than aa by hs=0.05, after selection, the frequency of Aa increases, allele frequency p increases (p=0.256).

c vs d vs e: in all three cases, the fitness of genotypes are W_AA=W_Aa=1.1 > W_aa=1, we expect allele frequency p to increase after selection. In d) and e), genetic drift diviate allele and genotype frequencies from the expected values, the larger the population size is (e), the less drift will make an influence, the closer the allele and genotype frequencies are to the expected value before and after selection. If the population size is small, the allele and genotype frequencies can be smaller or larger than expected (before and after selection) due to drift.

4) DoOneFullGeneration()

```
DoOneFullGeneration <- function(h, s, popSize, p){
    EGFvec = MakeHWfreq(p)
    GNvec = DoDrift(popSize, EGFvec)
    return(DoSelection2(h, s, GNvec))
}

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

```
## [1] 0.5083260 0.4472618 0.4939598 0.4656437 0.4945017
```

```
# b
h=1
s=0.1
n=10^6
p=0.5
DoOneFullGeneration(h, s, n, p)
```

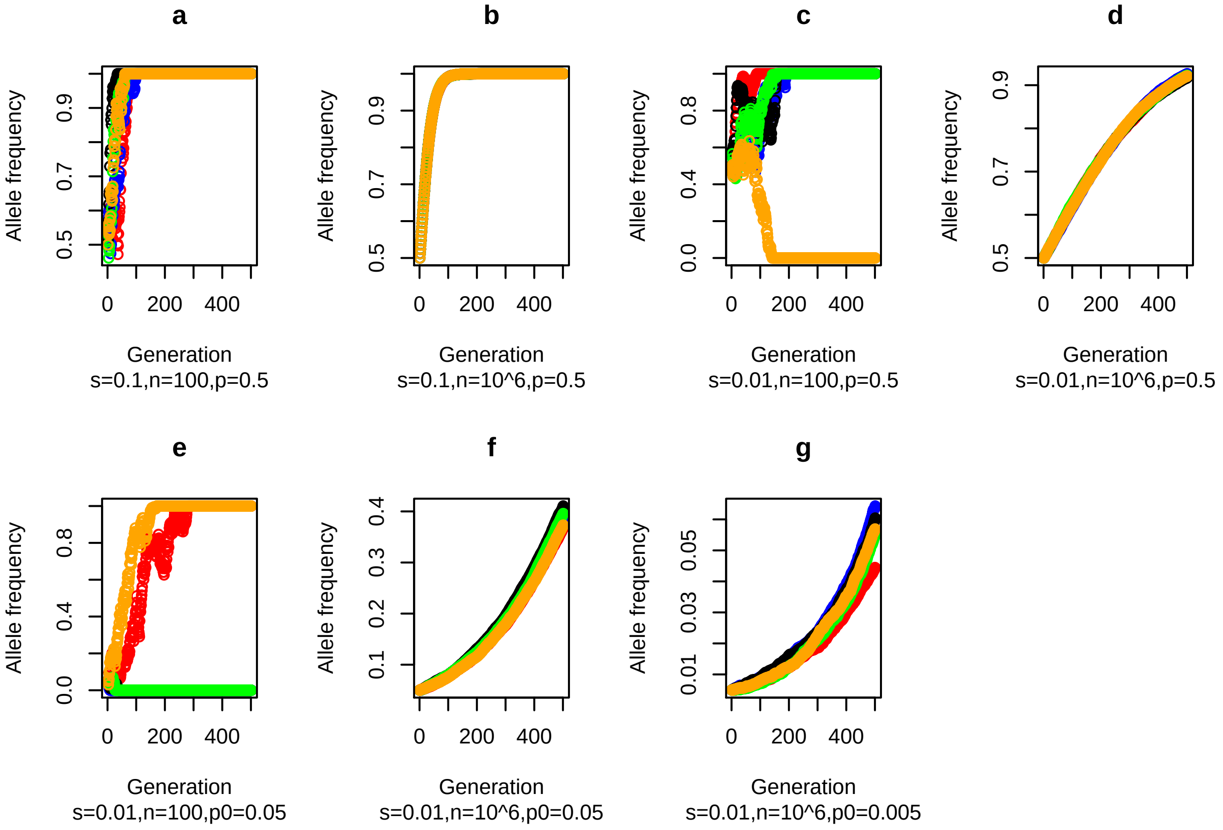
```
## [1] 0.5116176 0.5118478 0.5119193 0.5113473 0.5122598
```

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)
```

```r
}

# 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))}
```

```r
# 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 sizeis smaller, there are more fluctuations in allele frequencies untill fixation.

a vs. c as well as b vs d: Selection is weaker (hs is smaller) in c & d compared to a & b, it takes longer for allele A to go to fixation.

d 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), it means drift can overwhelm the selectio, A allele could be lost to fixed. f and g have larger population size, under weak positive selection, allele A slowly goes to fixation.

6) DoManyGenerationsV2()

```r
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
for (i in 1:5){
    print(DoManyGenerationsV2(h, s, n, p0, gmax))
}
```

```
## [1] 347   1
## [1] 101   1
## [1] 70  0
## [1] 274   1
## [1] 233   1
```

```r
# b
h=0.5
s=0.01
n=100
p0=1/200
gmax=500
for (i in 1:5){
    print(DoManyGenerationsV2(h, s, n, p0, gmax))
}
```

```
## [1] 1 0
## [1] 9 0
## [1] 500.000000   0.631273
## [1] 1 0
## [1] 1 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] 0.000496
```

```
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] 0.000578
```

```
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] 0.001232
```

```
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] 8e-06
```

```
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] 1.6e-05
```

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

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

Results

wait till the lecture on fixation probability