SelectionOnaSingleLocuswithTwoAlleles

August 27, 2015

1 Selection On a Single Locus with Two Alleles

In this assignment we are looking at the change in allele frequency at a single site over 200 generations due to selection in an infinitely large population. We restrict the number of alleles at our locus to two, represented by p and $(1-p)$. With two alleles there are three possible genotypes. The frequencies are p^2, $2p(1-p)$, and $(1-p)^2$. Since selection is occuring in our system, each genotype has a fitness value associated with it. We define the fitness values as W_{11}, W_{12} and W_{22}, with the subscripts 1 and 2 representing the alleles p and $(1-p)$, respectfully.

We can multiply the frequency of each genotype by its fitness value, which can be interpreted as the sum of the genotypic frequencies weighted by their genotypes, giving us the mean fitness of the population:

$$\overline{W} = p^2W_{11} + 2p(1-p)W_{12} + (1-p)^2W_{22}$$

By dividing the expected frequency of allele p by the mean fitness of all genotypes, we get the expected frequency of p in the next generation.

$$p_{t+1} = \frac{p^2W_{11} + p(1-2)W_{12}}{\overline{W}}$$

Using the recurrence equation above, we are able to write a simple for-loop using Python to examine the effect of selection on the frequency of p over 200 generations. We set the initial frequency of p to 0.01, say that $W_{11} = W_{12} = 1$, and that $W_{22} = 0.9$

```
In [6]: %matplotlib inline
        from __future__ import division
        import numpy as np
        import matplotlib.pyplot as plt
```

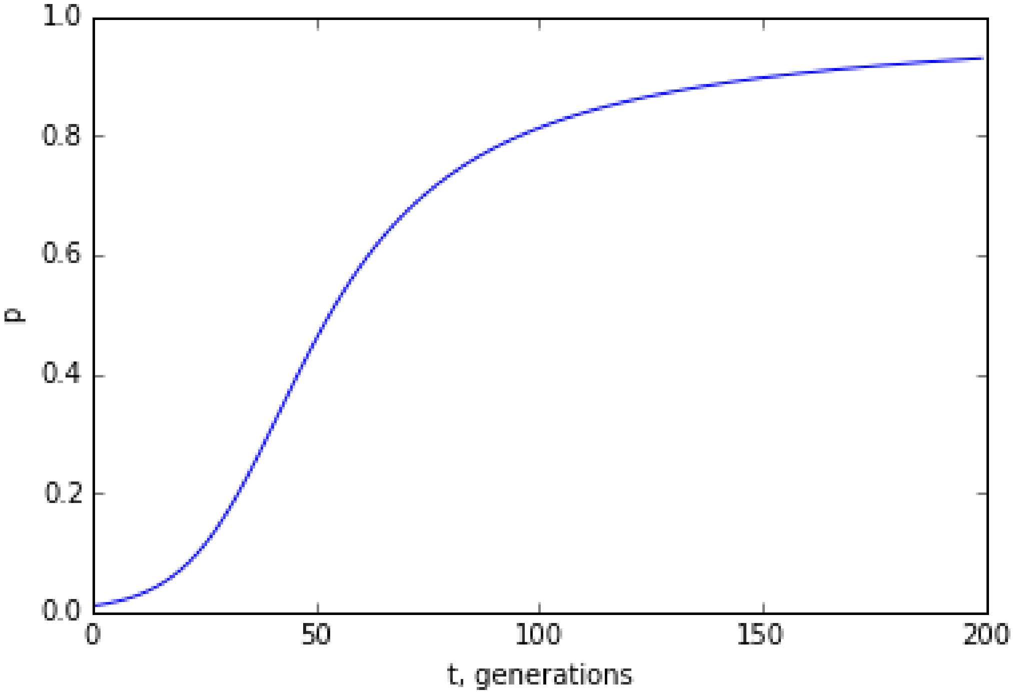
```
In [7]: W11 = W12 = 1
        W22 = 0.9
        p = 0.01
        generations = 200
```

```
In [8]: pFreq = []
```

```
In [9]: for x in range(0,generations):
            if x == 0:
                pFreq.insert(x, p)
            else:
                p = pFreq[x-1]
                W_mean = ((p**2)*W11) + (2*(1-p)*p*W12) + (((1-p)**2)* W22)
                pNew = (p*((p*W11) + ((1-p)*W12))) / W_mean
                pFreq.insert(x, pNew)
```

Now we plot the frequencies of p over 200 generations

```
In [11]: plt.plot(pFreq)
         plt.xlabel('t, generations')
         plt.ylabel('p')
         plt.show()
```


p
t, generations