

```
#!/usr/bin/python
# twoloc.py

# File twolocinc.py contains incomplete code that we gave to students
# for a lab exercise. This file contains the completed code.
from pgen import mnldev, Tabulator
from math import sqrt

nreps = 1000
twoN = 5000
s = 0.02 # selective advantage of allele A
c = 0.001 # recombination rate
w = [1.0+s, 1.0+s, 1.0, 1.0] # fitnesses of AB, Ab, aB, & ab

# We have not taught the students how to use Tabulator yet, so the lab
# just asks them to print lines of output and make their own
# tabulation.
tab = Tabulator(0,1,10)

print "2N=%d s=%f c=%f" % (twoN, s, c),
print "Fitnesses:", w
#print "%5s %6s %6s %6s" % ("rep", "pA", "pB", "rsq")
for rep in xrange(0, nreps):
    x = [1.0/twoN, 0, 0.5-1.0/twoN, 0.5] # freqs of AB, Ab, aB, & ab

    while True:
        # adjust x for recombination
        if c > 0:
            pA = x[0] + x[1]
            pB = x[0] + x[2]
            x = [x[0]*(1-c) + c*pA*pB,
                  x[1]*(1-c) + c*pA*(1-pB),
                  x[2]*(1-c) + c*(1-pA)*pB,
                  x[3]*(1-c) + c*(1-pA)*(1-pB)]
            # adjust x for gametic selection. There is no need to normalize
            # the new x so that it sums to 1. mnldev does that automatically.
            for i in range(len(x)):
                x[i] *= w[i]

            n = mnldev(twoN, x) # sample from multinomial
            x = [float(z)/twoN for z in n]
            pA = x[0]+x[1]
            pB = x[0]+x[2]
            if pA==0 or pA>=0.5 or pB==0 or pB==1:
                break

        if (pA >= 0.5) and (0 < pB < 1):
            D = x[0]*x[3] - x[1]*x[2]
            rsq = D*D/(pA*(1-pA)*pB*(1-pB))
            # The next print statement is as requested in the lab manual.
        # print "%5d %6.3f %6.3f %6.3f" % (rep, pA, pB, rsq)
        tab += rsq

# This prints the tabulation in a readable format. But it
# is not part of the assignment, because the students don't know about
# Tabulator.
print tab
```