# Notebook

August 10, 2017

## 1 Moran Processes

The evolutionary models discussed in the previous chapters assume an infinite population that can be divided in to infinitessimal parts. Finite populations can also be studied using a model called a Moran Process (first described in 1958).

---

### 1.1 Moran process with neutral drift

Consider a population of two types of fixed size $N$. This can be represented as a vector of the form: $(i, N-i)$ where $i \geq 0$ represents the number of individuals of the first type.

The term **neutral** drift refers to the fact that the two types reproduce at the same rate.

The Moran process is as follows:

- At a given time step: select a random individual for reproduction and a random individual for elimination
- The eliminated individual is replaced by a new individual of the same type as the individual chosen for reproduction.
- Proceed to the next time step.
- The process terminates when there is only one type of individual in the population.

---

Here is some simple Python code that simulates such a Process assuming an initial population of $(3,3)$:

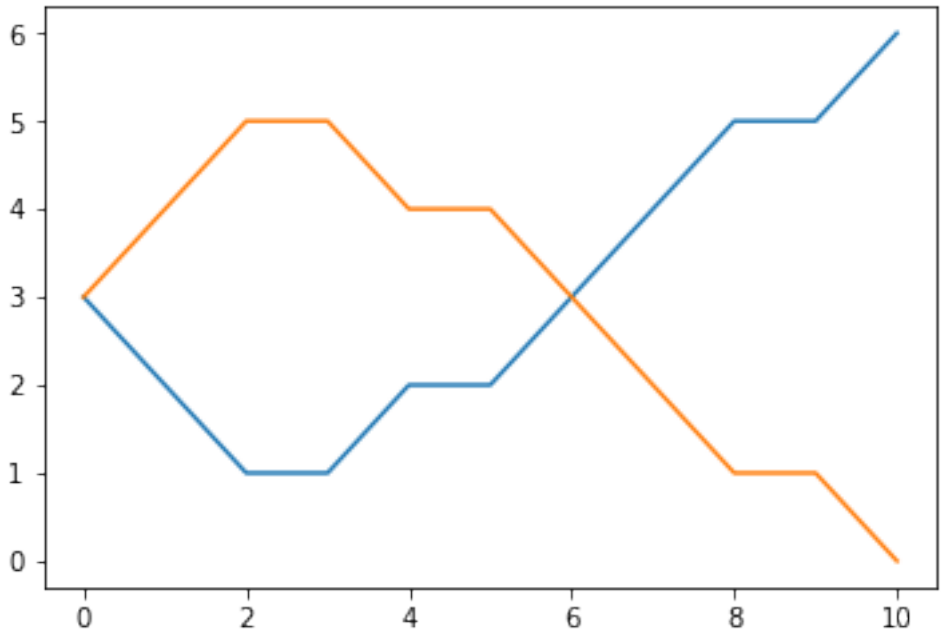
```
In [2]: import numpy as np
        import matplotlib.pyplot as plt
        import random
        %matplotlib inline

        def neutral_moran(N, i=1, seed=0):
            """
            Return the population counts for the Moran process with neutral drift.
            """

            population = [0 for _ in range(i)] + [1 for _ in range(N - i)]
```

```
    counts = [(population.count(0), population.count(1))]
    random.seed(seed)
    while len(set(population)) == 2:
        reproduce_index = random.randint(0, N - 1)
        eliminate_index = random.randint(0, N - 1)
        population[eliminate_index] = population[reproduce_index]
        counts.append((population.count(0), population.count(1)))
    return counts

N = 6
plt.plot(neutral_moran(N=N, i=3, seed=6));
```




For different seeds we see we obtain different results. What becomes of interest is not the path but the end result: which strategy overcomes the presence of the other?

```
In [3]: def neutral_fixation(N, i=None, repetitions=10):
            """
            Repeat the neutral Moran process and calculate the fixation probability
            """
            fixation_count = 0
            for seed in range(repetitions):
                final_counts = neutral_moran(N=N, i=i, seed=seed)
                if final_counts[-1][0] > 0:
                    fixation_count += 1

            return  fixation_count / repetitions
```