CruzHack 2018

Joey Liao, Kyle Chang, Michael Chan

1/20/2018

1 Abstract

This program uses Java to simulate mutations in cells during reproduction and demonstrates a visual representation of evolution over time.

2 Single Nucleotide Polymorphism Simulator (SNPS)

2.1 Background

DNA sequence consists of the nucleotides Adenosine, Thymine, Guanosine, or Cytosine (A,T,G, and C). These “letters” act as biochemical “code” that creates a cell and tells the cell how to function. It is a “sequence of letters”, and you may think of it as a biological coding language.

When a cell reproduces, there is a small chance for each “letter” in this “string of code” to mutate (about 200 mutations in an entire human DNA sequence, or a 3.3 x 10^{-8} probability per letter). For example, by random chance, an A can turn into a T. This is called a single nucleotide polymorphism (SNP), which in basic English means “one letter gets changed”.

SNPs lead to changes in the DNA sequence , or its script. Fortunately, many sections of the sequence contain non-functional code (introns), so consequential mutations in offspring cells are noticed slowly. Imagine if we can speed up the process of cell reproduction. We could predict years of cell reproduction and see the future DNA sequences and mutations of all a single cell’s progeny. This simulation is the next best thing that can help researchers study the nature of cell mutation, and also predict volatility in carcinomas, super bacteria, and viruses for applications in farming and medicine.

An SNP can replace a letter in the DNA code, which can have negative effects such as creating a different protein structure, leading to harmful defects. These cells have a high probability of death before it can pass on its genes, thus ending the existence of the cell’s unlucky design by nature.

On the contrary, cells may rarely develop beneficial sequences, or in other words, a sequence that allows it to survive a certain environmental hazard. These cells survive, whereas the rest die out (perhaps it’s the perfect sequence which creates a chemical reaction that creates a high-base outer-coating which can neutralize acid and thus make the cell immune to acidic material). This “survivor cell” can pass on its genes and create a new generation which is immune to said environmental hazards.

For example, bacterial pests grow on crops, and farmers use pesticides to combat them (environmental factor introduced). Our model and simulation will predict how effective the pesticides will be before the bacteria evolves and develops immunity, becoming a super bacteria. We can simulate many trials of a would-be 10 year period to obtain approximate but useful data that helps farmers plan when to upgrade to a more effective pesticide. Farmers can also determine an optimal amount of pesticides to use so that the pests don’t evolve as fast as the model predicts. The applications extend to oncology, virulence, etc.

2.2 Our Program

Everything was created in Java.

We have a GUI which allows users to choose numerous custom parameters for the simulation.

Pretend some users consist of scientists researching some arbitrary bacteria. First they’ll need a starting cell’s sequence (a single cell to grow in petri dish). They know the sequence of this cell from DNA sequencing techniques, and should enter a long string of A, T, G, and Cs.

The scientists will allow the cell to grow in population. Cells thrive in the medium and are assumed to not die. They will set a total time for the simulation, and an interval rate which determines when the cells will reproduce. They will also set a probability that a letter in a new cell may mutate (realistically will be small number).

The scientists will input a time which an environmental factor (EF) will be introduced. This EF will define a new reproduction rate and a introduce a death rate per interval.

Cells with a certain sequence at a certain location in their DNA strings will be allowed to survive an environmental factor. This will be entered as a “Beneficial Sequence”. Cells that contain this lucky trait will have their own custom death and reproduction rate defined by the scientists (usually beneficial for survival). In theory, those with the sequence survive and have a high chance of passing the beneficial sequence to the next generation.

After all these parameters are set, the program will run, displaying an animated graph that updates in real time. This graph represents the cell population’s ratio of beneficial sequence cells to those without it. What the scientists should expect to see over time is that the beneficial sequence cells increase to approach 100% of the population and the others go down towards 0 (unless they add parameters to force other outcomes).

There are many examples that expectations can be manipulated by choosing the right parameters. One is that the scientists introduce the environmental factor too early, and so cells are not able to reproduce enough times to obtain a cell with the beneficial sequence. As a result, the scientists can expect to see that all their cells die out/go extinct.

At the end of the simulation, there will be a statistics page that describe the results of simulation, and the users may be able to rerun or reset the program.