

Honors Thesis

Ben Roos

February 24, 2015

Table of Contents

1. Introduction
2. Model
3. Implementation
4. Lesson Plan
5. Discussion

1 Introduction

2 Model

2.1 Population

The simulation uses a Moran Model to describe a finite population of 1024 individuals. The simulation tracks a single allele for each individual. This allele is selected independently for each individual, and alleles are represented using numbers that begin at 0 and may be infinitely high. Each individual is geographically isolated from others in a grid that is 32 cells long and 32 cells wide. Each individual takes up only one cell, and each cell contains only one individual. Thus, the size of the population will never vary from 1024 at any step of simulation.

2.2 Initializing the Population

When the simulation initializes, a population is randomly generated. The allele number for each individual in the population is selected sequentially, beginning with the upper left corner of the grid, proceeding through each row from left to right, and ending with lower right corner of the grid. The first individual is immediately assigned the allele number 0. With probability $\frac{i}{i + \theta}$, where i is the number of individuals in the population generated so far, and θ is equal to $2N\mu$, where N is the number of individuals in the population and μ is the mutation rate, subsequent individuals in the population will be assigned an allele number equal to that of a randomly selected member of the population. With probability $1 - \frac{i}{i + \theta}$, subsequent individuals are assigned to new allele numbers, momentarily becoming the only member of the population with the given allele. Thus, individuals that are assigned alleles earlier in the initialization process are more likely to be assigned new alleles, while individuals that are assigned later in the process are more likely to be assigned an allele from the population.

2.3 Simulation

At each step of the simulation, a randomly selected individual dies, leaving its corresponding cell momentarily empty. A parent allele is then randomly selected from the immediate neighbors of the empty cell. With probability $1 - \mu$, the new individual will have the same allele as its parent, and with probability μ a mutation will occur, and a unique allele number will be assigned to the new individual. 2000 steps occur during each generation, and the population is redrawn in the visualization once for each generation.

3 Implementation

3.1 Design

3.1.1 Layout

The application contains three components: a control panel, with which users can control the simulation and its parameters; a grid, on which the simulation is displayed; and graphs, on which statistics regarding the simulation are shown.

3.1.2 Input Parameters

The control panel includes a text field in which users can enter the mutation rate that will be used during the simulation. The default mutation rate is 0.00001, and is displayed in the text field when the page loads.

3.1.3 Running the Simulation

To start and pause the simulation, users click on a single button, which will display either "Start" or "Pause" depending on whether the simulation is running. Users can reset the simulation by clicking "Reset." This control generates a new initial population, reinitializes the grid, and clears the graphs.

3.1.4 Export

If users would like to export a JSON representation of the population at a given point in the simulation, they can click "Export" while the simulation is paused. The format of the JSON object is `{ cells: [{color: hex_val, allele: [-2, ∞), mutationNumber: [0, ∞], {...}, ...] }`. Each cell object includes the CSS hex value for the color associated with the allele, the allele number, and the mutation number, and the JSON object that is created is a list of these cell objects.

3.1.5 Barriers

Users also have the ability to create a barrier in a cell by clicking on the desired cell. When a barrier is created, the allele number is set to -1 and color associated with the cell is #000000 (black). Barriers cannot act as parent cells, so they are never replicated, and barriers cannot die during the simulation and subsequently be replaced. Thus, using barriers, users can construct physical constraints around which the movement of alleles of subpopulations is restricted. Users can create multiple barriers by clicking on multiple cells or by clicking and dragging. If users want to remove a barrier, they can click on the chosen cell a second time. This action will set its allele number to -2 and its color to #FFFFFF (white). In this state, the cell can be considered empty; neighboring cells can replicate their alleles into an empty cell, but it cannot be chosen as a parent. Barriers can be cleared either by clicking or by clacking and dragging.

3.1.6 Barrier Templates

If users are running a local copy of the application, they can also create templates of barriers so that barrier patterns can be applied with a single click. In order to create a template, users must first create the barrier pattern that they would like to save by clicking, or clicking and dragging, over the cells that should be barriers. Users can then export the grid to a JSON object by clicking "Export." They can then locate BarrierTemplate.js in their local copy of the application. They should replace the JSON object that is assigned to the barrierTemplate variable with the JSON object that they just downloaded. Now, in their

local copy, the "Template" button on the control panel will refer to the barrier pattern they defined.

3.1.7 Forced Mutations

Users can force a mutation to occur in a similar manner to creating barriers. By holding the SHIFT button while clicking, or clicking and dragging, they can force grid cells to contain individuals with mutant alleles. When a user performs this action, the allele number is changed to the mutant allele number and a new color is assigned to the cell.

3.1.8 Graphs

Two graphs are displayed to the right of the grid. The upper graph displays the number of unique alleles in the population over time and the lower graph displayed the allele frequencies of each unique allele in the population over time. In the lower graph, the colors associated with each allele match the colors of the simulation.

3.2 Architecture

3.2.1 Control Panel

3.2.2 Grid

3.2.3 Graphs

Number of Alleles

Allele Frequency

4 Lesson Plan

5 Discussion