

Modelling changes of phenotypic distribution in natural populations due to selection by means of computer algorithm

Kári Hlynsson

July 25, 2021

Abstract

In this paper we discuss the uses of computer simulation in predicting the change of phenotypic distributions in natural populations as a result of natural selection. By means of statistical analysis and data visualization, we propose a method with which certain evolutionary patterns in populations can be identified. All the findings of this publication are not meant to reflect the exact mechanisms of natural selection and precisely how adaption occurs but aims to display generally how traits in populations develop over time and react to new environmental factors.

Introduction

Algorithm construction

The algorithm used to simulate conditions presented in this paper was programmed using Python 3.9.5. Live plotting was accomplished by utilization of Matplotlib. However, in case of data visualization post-simulation, R was used. The author of this paper would like to note that the intent of this simulatory algorithm is not to reflect precisely how adaptation in species occurs but rather to display generally how the mechanisms of natural selection operate.

We begin by considering a population P of n individuals. All individuals of P share similar traits such as lifespan and number of offspring produced. Table 1 below shows some of the definitions of various traits that will be used in this algorithm.

Symbol	Trait	Description
P	Population	Set of individuals of the same species.
E	Environmental difficulty	Factor present in environment which affects survival or fertility of individual in population.
ς	Selection ratio	Ratio of individuals selected for environmental checks and total population size.
i	Individual	Individual organism of population P .
w_i	Fitness	An individual's fitness towards the environment.
$w_{\bar{z}}$	Comparative fitness	The ratio of an individual's fitness and the modal fitness value \bar{z} of population P .
λ	Reproductive capability	Number of offspring produced per individual.
τ	Genetic volatility	The likelihood that the offspring of an individual i will express drastic changes in phenotype and thus greatly affect fitness.

Table 1: Abbreviations and symbols used for algorithm variables.

The algorithm is *turn-based* in the sense that individuals perform certain tasks and are exposed to environmental factors on a single-turn basis. By using this method, we can clearly visualize and

analyze data gathered from simulations to estimate various factors in adaptation of the population to the specific environment. Thus when we say that the time elapsed is 500 intervals, we are referring specifically to how many *time intervals* or *turns* have passed from the start of the simulation. For the sake of simplicity, henceforth we shall refer to these intervals as *time intervals*.

An algorithmic component that requires careful consideration is how to determine *whether an individual survives in the environment or not*. While there are many ways to do this, we shall utilize an *environmental factor*, E , to carry out the logical comparison

$$P(w_i) = w_i \geq E \quad (1)$$

If $P(w_i)$ for a given individual is true, that individual is said to have *survived the environmental check*. Else, it is said to have failed and will be removed from the population, effectively *dying*. By specifying a selection ratio, ς , upon initial configuration, the percentage of the population that is selected each turn can be modified.

At the onset of each turn, all individuals who have attained the max age of their species will die. All individuals who survive the environmental check in the following turn are prompted to reproduce. If their *reproductive cooldown* timer has not reached zero, they will not reproduce that turn. Those that do reproduce produce λ offspring. Calculation of offspring fitness is acquired by the formula

$$w_c = w_p \pm \rho \quad (2)$$

Where ρ is uniformly a trivial shift in fitness to represent shuffling of genetic material and such. However, acquired *genetic volatility* depicted by the variable τ presents the probability of a *mutation* occurring when an individual reproduces. A mutation has a more drastic effect on an individual's fitness and can be either deleterious or beneficial.

Data analysis

We shall start this section by defining some key parameters that shall be used in the calculations of this paper. One variable of interest is the *modal location of the fitness value* in a population, which we symbolize \bar{z} . By studying the displacement of this location, we can see how much adaptation has taken place as a result of exposure to an environmental factor.

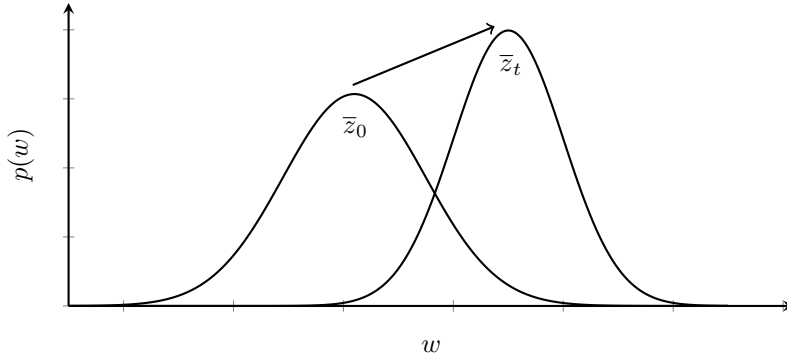


Figure 1: Determination of $\Delta\bar{z}$ can be used as indicator of adaptation.

Figure 1 depicts the vector displacement of \bar{z} at simulation start ($t = 0$) towards \bar{z}_t , that is when t time intervals have passed. We can represent the displacement with the vector notation

$$\vec{F}_S = \begin{pmatrix} \Delta\bar{z}_x \\ \Delta\bar{z}_y \end{pmatrix} = \begin{pmatrix} \bar{z}_{x(t)} - \bar{z}_{x(0)} \\ \bar{z}_{y(t)} - \bar{z}_{y(0)} \end{pmatrix} \quad (3)$$

and define it as the *force of selection* due to environmental factor E . This provides us with a variable that allows for quantitative observation of the mechanisms of selection within the algorithm. Note

that a high x value of a \vec{F}_S indicates that the *adaptive magnitude* was very high while a high y value indicates that the *adaptive saturation* is very dense.

An interesting facet of the algorithm that presented itself despite its unintended appearance in the simulation is the *relationship between birth/death rate and carrying capacity*. Imagine that n_d individuals die each turn while n_b are born into the population. Due to the algorithm's construction we can reasonably estimate the number of individuals born into the population each turn by usage of the reproductive capability constant λ and the number of *individuals who are capable of producing each round*.

$$n_b = \lambda n_r \tag{4}$$