Models for exploring genome-fitness mapping in an introductory course

David A. Thompson*

School of Natural Sciences, Fresno Pacific University
Fresno, CA
USA

*Corresponding author: David Thompson, thompson@fresno.edu
Abstract

Evolutionary theory presents a number of conceptual hurdles for the undergraduate student. In addition, the extended timespan over which many evolutionary processes occur, coupled with the time constraints of the typical undergraduate laboratory, restricts the range of relevant laboratory exercises available to augment instruction in an undergraduate course. Computer-based simulations have the potential to overcome this barrier as they allow interaction with models of natural processes which occur on long timescales. Such simulations allow the student to pursue what would be impossible at the bench: to investigate multiple scenarios and to collect, visualize, and analyze large amounts of data. Here, we describe a set of agent-based models, developed using the NetLogo platform, designed to facilitate developing a better understanding of the interplay between random processes, organismal fitness, encoding, and genomic sequence space.
1 Introduction

Developing an understanding of concepts associated with evolutionary theory can be a substantial challenge for students (rev. in Gregory, 2009). Preestablished misconceptions, a general lack of understanding of the nature of science, preestablished sociocultural and religious beliefs, and difficulties in understanding random processes may hamper the ability of the student to understand or accept various aspects of evolutionary theory (Garfield, 2002; Lombrozo et al., 2008; Klymkowsky and Garvin-Doxas, 2008; Garvin-Doxas and Klymkowsky, 2008; Hokayem and BouJaoude, 2008; Thagard and Findlay, 2010; Yates and Marek, 2014).

Given the significance of the theory – and its movement toward a central position in the undergraduate biology curriculum – there is substantial interest in developing methods, recommendations, and tools which can be employed to improve the undergraduate learning experience in the area (American Association for the Advancement of Science, 2011). Several authors have developed sets of pedagogical recommendations embedded in broader analyses of the field (e.g., Alters, 2005; Nelson, 2008; Rosengren et al., 2012; Frasier and Roderick, 2011; Werth, 2012). Another group has shared a description of the curriculum for a GE course centered on evolutionary theory (O’Brien et al., 2009). Others have offered individual pedagogical recommendations and descriptions of specific pedagogical tools, approaches, methods (e.g., McComas, 1991; Pennock, 2007; Nehm and Reilly, 2007; Werth, 2009; Smith et al., 2009; Andrews et al., 2011; Speth et al., 2009; Gillings, 2012).

Agent-based modelling (ABM) describes the behavior of a system by defining a set of agents, autonomous entities, each of which behaves according to a specific set of rules and which typically participates only in local interactions (Epstein and Axtell, 1996; Epstein, 1999; Bonabeau, 2002). ABM has been applied to a variety of problems in the physical sciences, life sciences, and social sciences and has also found application in education as a teaching tool.

Here, we describe a set of agent-based models designed to facilitate developing a better understanding of random processes, fitness, and encoding. As the models are intended to be used as tools within the context of a larger, well-designed learning experience, we offer suggestions about ways in which the models might be incorporated into an undergraduate course.
2 Descriptions of the models

The simulations described here are intended to be used to assist the student in developing an understanding of the relationship between genome, phenotype, and organismal fitness. Rather than attempting to closely emulate a biological system, we have developed four simple models similar to the introductory models described in De Jong’s “Evolutionary Computation” (De Jong, 2006). The source code for the models is freely available under the GNU GPL at https://gitlab.com/dtho/genfi-nl.

2.1 EV algorithm on fitness landscape L1 (EV-L1)

The EV-L1 model employs a slightly modified version of the EV algorithm, a simple algorithm for modeling changes in a population over time (De Jong, 2006). The simulation begins with a population of organisms, each with a genome composed of a single chromosome which holds a single mutable value, a real number between -100 and 100. Fitness is defined using the fitness function \( F = 50 - x^2 \) where \( x \) represents the value of the chromosome and \( F \) represents fitness (Fig. 1). The EV algorithm loops through the following sequence until some arbitrary criterion is fulfilled:

1. Select a member of the population (a 'parent') at random.
2. Create a single 'child' organism by making an identical copy of the parent.
3. Subject the genome of that child organism to mutation. At each 'birth' of a new population member, the mutation operator mutates the chromosome by, at random, either adding 1.0 to or subtracting 1.0 from the parent value.
4. Compare the fitness associated with the newly-formed child organism with that of the parent organism and remove the least fit organism from the population.

Organisms are visualized on a two-dimensional grid (Fig. 2). The position of the organism on the x axis corresponds to the value of that organism’s chromosome. The strip across the bottom of the main window is a simple one-dimensional heat map showing the relationship between chromosome value and fitness (where a lighter shade corresponds to greater fitness). The y coordinate of an organism has no significance; organisms are represented at different y coordinates simply to facilitate visualization.

2.2 GA on fitness landscape L2a using floating-point representation (GA-L2a-float)

With a simple genetic algorithm (GA), a population of chromosomes, each encoding a candidate solution, is subjected to evaluation by a fitness function. A chromosome which represents a better solution is more likely to be retained in the system. Genetic variability is introduced into the system by the action of genetic operations, such as mutation or crossover, on selected individuals (Holland, 1975; Goldberg and Holland, 1988; Foster, 2001).

The GA-L2a-float model employs a simple GA with a population composed of chromosomes, each composed of two "residues", \( res_1 \) and \( res_2 \), where both \( res_1 \) and \( res_2 \) are floating point numbers within the range \([-5.5, 4.5]\). Genomes experience mutation via a mutation operator which samples from a continuous Gaussian distribution, thus facilitating infrequent changes of larger magnitude, as well as more frequent changes of smaller magnitude.
In the NetLogo model, each chromosome resides in an organism (in NetLogo jargon, a ‘turtle’). At birth (when a child turtle is generated), each residue of the genome of the newly-hatched turtle is potentially acted on by the mutation operator. The probability, for each residue, of mutation can be varied via the user interface using the slider control associated with the RESIDUE-MUTATION-RATE value. A value of 100 corresponds to a probability of 1 (certainty) that a residue will be acted on by the mutation operator; a value of 0 corresponds to a probability of 0 that a residue will be acted on by the mutation operator.

The fitness landscape L2a (Fig. 3) is described by the fitness function \( F = ph_1^2 + ph_2^2 \) where \( ph_1 \) is equal to the value of \( res_1 \) and \( ph_2 \) is equal to the value of \( res_2 \).

The relationship between genome and fitness can be readily visualized using the NetLogo user interface. Organisms are visualized on a two-dimensional grid (Fig. 4). The x- and y-positions of the organism correspond to the values of \( res_1 \) and \( res_2 \), respectively. A two-dimensional heat map communicates the relationship between chromosome value and organism fitness (a lighter shade corresponds to greater fitness).

### 2.3 GA on fitness landscape L2a using bit encoding (GA-L2a-bit)

Apart from the composition of each genome and the mutation operator, the GA-L2a-bit model is essentially identical to the GA-L2a-float model.

Here, each genome is composed of a single chromosome containing 10 mutable residues, each assuming a value of either zero or one. The residues behave as two 5-bit words. Each 5-bit word represents a floating-point value where the first bit is treated as the sign bit (0 and 1 corresponding to negative and positive values, respectively), the last bit is treated as a fractional component (\( 2^{-1} \)), and the intervening bits are the binary representation of the corresponding integer value. Mutation occurs at random across the genome with the mutation operator acting as a simple bit flip function.

The fitness landscape L2a (Fig. 3) is defined by the fitness function \( F = ph_1^2 + ph_2^2 \) where \( ph_1 \) is the floating-point value corresponding to the first 5-bit word \( (res_1 \text{ through } res_5) \) of the genome and \( ph_2 \) is the floating-point value corresponding to the second 5-bit word \( (res_6 \text{ through } res_{10}) \) of the genome.

Organisms are visualized on a two-dimensional grid where the x- and y-positions of the organism correspond to the values of \( ph_1 \) and \( ph_2 \), respectively. A two-dimensional heat map communicates the relationship between the values of \( ph_1 \) and \( ph_2 \) and organism fitness (a lighter shade corresponds to greater fitness).

If, in the NetLogo user interface, the switch control associated with VISUALIZE-P is set to 'On', the visualization is altered in several ways. Turtles in the two “generations” under consideration are highlighted prior to deletion of turtles in the parent generation. Offspring are represented using the 'turtle' shape. A numeric fitness value is shown for each turtle and a ‘trace’ is generated between each turtle in the newest generation and its parent. Traces persist for several generations before being deleted. In conjunction with the ‘step’ button control, this provides the ability to visualize the consequence of each mutation event as it occurs, providing the student with a tool for step-wise analysis of the simulation.
2.4 GA on fitness landscape L4 using floating-point representation (GA-L4-float)

The 'GA-L4-float' model employs a simple GA with a population where each organism has a simple ‘genome’ composed of a single chromosome. Each chromosome is composed of two "residues", $res_1$ and $res_2$, each a floating point number. The genome experiences mutation via a mutation operator which samples from a continuous Gaussian distribution.

GA-L4a-float is distinguished primarily by a fitness landscape with multiple fitness optima. Organismal fitness is defined by the fitness function $F = \sin(1.2 \times res_1) \times \sin(1.2 \times res_2)$ where both $res_1$ and $res_2$ are interpreted as radians (Fig. 5) and where both $res_1$ and $res_2$ are floating point numbers within the interval [-5.5, 4.5].

As with the other GA models, organisms are visualized on a two-dimensional grid with their position corresponding to the values of $res_1$ and $res_2$. A two-dimensional heat map communicates the relationship between chromosome value and organism fitness (a lighter shade corresponds to greater fitness).
3 Discussion

Computer-based simulations can function as effective learning tools (rev. in Rutten et al., 2012; Smetana and Bell, 2012). Such simulations offer the student the opportunity to interact with simplified models of complex systems, to visualize the dynamics of complex systems, to rapidly explore hypothetical scenarios, and to alter the time scale of a process, permitting conceptual interaction with natural processes which are intractable for analysis in the undergraduate laboratory course due to time constraints. Here, we used NetLogo as it is particularly well-suited for use as an educational tool (Wilensky, 1999; Sklar, 2007; Wilensky and Rand, 2015).

The simulations we describe are intended to be used as tools within the context of a larger, well-designed learning experience. In this context, they are intended to provoke investigation of abstract evolutionary scenarios, reinforcing and developing understanding of concepts such as sequence, genotype, and phenotype spaces and the fitness landscape (Wright, 1932; Gavrilets, 2004). The models are further intended to serve as tools to challenge the student to engage in “algorithmic thinking” and to develop the student’s ability to connect population-level behavior to underlying rules acting at the level of individual “actors”.

The simplest of the models, EV-L1, utilizes a chromosome composed of a single mutable value and a smooth fitness landscape with a single fitness maximum and two fitness minima (Fig. 1). Given the simple ±1 mutation operator and the smooth landscape, the behavior of the system is straightforward, with the average fitness increasing uniformly toward the maximum fitness as the simulation progresses. This simple simulation, clearly far-removed from complex biological systems, is intended to serve as an accessible and non-threatening point of engagement for the student.

Unlike the L1 landscape, the L2a landscape has three near-optimal solutions at [-5.5, 4.5], [4.5, -5.5], and [4.5, 4.5] and one optimal solution at [-5.5, -5.5]. The simulations on this landscape, GA-L2a-float and GA-L2a-bit, exhibit behaviors that are more complex than that of EV-L1. Although, with the default model values, average population fitness rises as the simulation progresses, it does not do so in a uniform manner, nor does it necessarily approach the fitness of the optimal solution.

Both the GA-L2a-float and the GA-L2a-bit models provide an opportunity to explore the possibility that, depending on the nature of the landscape, a system may become trapped at a suboptimal solution. Neither model guarantees convergence to the optimal solution and, depending on the model parameters, either model may be trapped at suboptimal solutions (Fig. 4). With the GA-L2a-float model, the NetLogo interface provides, via the slider control associated with the MUTATION-DELTA value, a straightforward way to alter the Gaussian mutation operator. One might extend a learning exercise incorporating these models by asking the student to characterize the relationship between the definition of the operator and the probability of “escaping” suboptimal maxima. Alternatively, one might ask the student to investigate whether mutation plays a similar role in biological systems.

Although the the GA-L2a-bit model uses the same GA and the same fitness function as does the GA-L2a-float model, the behavior of the system is markedly different due to the different encoding system and the necessarily different mutation operator. As the two values, phe₁ and phe₂, are encoded as two 5-bit words, the variation in the average population fitness is much more substantial than that observed with float encoding (Fig. 6).

Considered together, the distinct behaviors of GA-L2a-float and GA-L2a-bit provide an opportunity to explore the significance of encoding. The contrast in the behaviors of the two systems is intended to highlight, for the student, the significance of the nature of the encoding system in facilitating or limiting opportunities for marked phenotypic changes with only “small”
changes in the genomic sequence. An astute student familiar with the process of translation will likely note that position sensitivity is also a hallmark of the biological genetic code. One pedagogical extension would be to ask the student whether particular encoding systems might be more or less “desirable” in biological systems or in the application of GAs for problem-solving.

The NetLogo interfaces for both the GA-L2a-float and GA-L2a-bit models use nearly identical representations to visualize the relationship between $ph_1$, $ph_2$, and fitness. However, the representation used with the bit model is not a “fitness landscape” in the sense that adjacent positions correspond to a pair of sequences that vary by a single sequence element. Indeed, the higher dimensionality of the bit model provides an opportunity for the student to begin to consider the difficulties associated with dimensionality and, more generally, with modelling more complex genotype-fitness spaces. In an introductory course, one might, as an exercise, ask the student to consider whether a small movement in sequence space necessarily corresponds to a small movement in genotype or phenotype space. An alternative exercise would be to investigate how one might visualize or represent the actual “fitness landscape” (genome-fitness mapping) associated with GA-L2a-bit, or to speculate regarding the nature of the fitness landscape for the genomic sequence space of a biological system. In a more advanced class, these models might serve as a starting point for consideration of approaches such as Gavrilet’s ‘holey’ fitness landscapes or Kauffman’s NK model (Kauffman, 1993; Gavrilets, 1997; Gavrilets and Gravner, 1997).

The GA-L4 model introduces a fitness landscape with multiple fitness optima. As the model arrives at different solutions on different trials, it offers the opportunity to explore the concept of stochasticity. As with the GA-L2a-float and GA-L2a-bit models, this model allows the student, via the NetLogo interface, the ability to explore the relationship between the definition of the mutation operator and the capacity to move between solutions and, at the conceptual level, to return to exploring the notion of to what extent it is advantageous for such systems to have an ability to “escape” local optima.

None of the models described here incorporate any explicit notion of an environment. One conceptual pitfall, for the student, is to misinterpret the position of a turtle in the NetLogo world as representing a physical position in an environment. However, the absence of an explicit environment from these models also offers the opportunity for the student to compare and contrast these models with the “real world” of biology. A learning goal of the larger curricular structure should include the arrival of the undergraduate student at the recognition that fitness is, directly, a function of phenotype and environment and, only indirectly, a function of genomic sequence. In an advanced course, one might pursue further the discussion of the complexity of mapping from genotype to phenotype with readings from the literature or other activities.

Since natural evolutionary processes tend to occur over time scales which are intractable for analysis in the laboratory component of a typical undergraduate course, computer-based simulations offer potential utility as educational tools for students studying evolutionary theory. The simple models described here are student-accessible and potentially of some utility across disciplines. The models do not purport to simulate anything even closely resembling a evolution of a biological population – rather, they provide an accessible point of engagement for the student in an introductory course. Use of the NetLogo platform provides the student with a user-friendly interface, visualization tools, and an opportunity to transition to ‘tinkering’ with the code of the models themselves. These qualities suggest the potential utility of such models in introductory courses within the life sciences as well as in introductory computer science or interdisciplinary courses. The pedagogical utility of the models might be enhanced by augmenting them with tools to track and visualize the genetic history of individual organisms or by providing additional models which allow the student to explore other fitness landscapes as well as other encoding
strategies'.
Acknowledgements

J Balwit

References


4 Figures
Figure 1: The fitness function for EV-L1. The EV-L1 algorithm uses the fitness function $F = 50 - x^2$ where $x$ represents the value of the chromosome and $F$ represents fitness. $x$ is restricted to assuming values between -100 and 100.
Figure 2: The NetLogo world for EV-L1 after 15 ticks. In a single representative simulation, the EV-L1 model, using default parameters, was allowed to proceed for 15 ticks. The image is a view of the NetLogo world at that point in the simulation.
Figure 3: The fitness function for fitness landscape L2a. The GA-L2a-bit and GA-L2a-float algorithms use the fitness function $F = ph_1^2 + ph_2^2$ where $F$ represents fitness and $ph_1$ and $ph_2$ are each restricted to assuming values between -5.5 and 4.5. The $F$ axis is scaled by a factor of 0.2 relative to the scales of the other axes.
Figure 4: Views of the NetLogo world upon completion of two runs of the GA-L2a-float simulation. The GA-L2a-float simulation was repeatedly allowed to run to completion, using the default values of the simulation. Representative views of the outcomes of such simulations are shown, in one case yielding a population trapped at a suboptimal solution (right) and, in another case, yielding a population at the optimal solution (left).
Figure 5: The fitness function for GA-L4-float. The GA-L4-float algorithm uses the fitness function \( F = \sin(1.2 \times \text{res}_1) \times \sin(1.2 \times \text{res}_2) \) where both \( \text{res}_1 \) and \( \text{res}_2 \) are interpreted as radians and where both \( \text{res}_1 \) and \( \text{res}_2 \) are floating point numbers restricted to values between -5.5 and 4.5.
Figure 6: Fitness distributions associated with the GA-L2a-float simulation and of the GA-L2a-bit simulation. In a single representative simulation, each model was allowed to proceed through 1500 births using the default parameters for the model. For each model, the distribution of individual fitness values of 500 organisms (those from 1001 to 1500 births) is represented with a boxplot.