A Framework for Teaching Biology Using StarLogo TNG:

From DNA to Evolution

by

Yaa-Lirng Tu

Submitted to the Department of Electrical Engineering and Computer Science
In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
February 2009

Author ______________________________
Department of Electrical Engineering and Computer Science
January 29, 2008

Certified by ______________________________
Eric Klopfer
Associate Professor of Education
Thesis Supervisor

Accepted by ______________________________
Arthur C. Smith
Professor of Electrical Engineering
Chairman, Department Committee on Graduate Theses

A Framework for Teaching Biology Using StarLogo TNG:

From DNA to Evolution

by

Yaa-Lirng Tu

Submitted to the Department of Electrical Engineering and Computer Science
In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

This thesis outlines a 10-unit biology curriculum implemented in StarLogo TNG. The curriculum moves through units on ecology, the DNA-protein relationship, and evolution. By combining the three topics, it aims to highlight the similarities among different scales and the relationships between them. In particular, through the curriculum, students can see how small-scale changes in molecular processes can create large-scale changes in entire populations. In addition, the curriculum encourages students to engage in problem-based learning, by which they are trained to approach questions creatively and independently.

Thesis Supervisor: Eric Klopfer
Title: Associate Professor

Acknowledgements

I would first and foremost like to thank my advisor, Eric Klopfer, who was always patient and supportive, and who always inspired me with new ideas when I had none. His enthusiasm and humor keep the StarLogo team a warm and creative environment to work in. I would like to thank Daniel Wendel, who was always the one to lend a helping hand, giving me advice and encouragement and great conversation. Without him StarLogo might still be crashing on my computer! Thanks to Ricarose Roque and Wendy Huang for showing me different teaching techniques and for always providing their smiling faces and good cheer. To Susan Yoon, for entrusting me with a NetLogo project that inspired the unit on cancer. A large portion of my project would not be possible without Mike D'Ambrosio, whose new level features made so many different new possibilities available. Liz Kim is responsible for a lot of great data collection from the various workshops. Finally, a big thanks goes to the rest of the team, including Peter Nga, Lily Liu, and Bianca Farrell, for all of the improvements they've made on the software, as well as fun times eating the delicious Dominican food after the workshops!

I would also like to thank the many teachers and students who have contributed to the development of the curriculum. The biggest thanks goes to Hal Scheintaub, whose many years of experience with StarLogo made him an invaluable resource and whose excitement was always contagious. Thanks to Rebecca Veilleux at Lawrence High School and to the Lawrence Family Development Charter School for letting us experiment on their students. And finally, thanks to the teachers present at the summer Imagination Toolbox workshop for their open minds and eagerness to help usher forth a new age in teaching.

Thesis aside, I must thank my family and friends for their love and support. To my parents, for raising me, pushing me, and always having faith in me. To my friends and housemates for keeping me sane, and to my students for driving me nuts. To my dance coaches, partners, and the entire MIT ballroom dance team, who provided me with ample distraction but a real family away from home. And to Andriy, who not only answered many biology questions for me, but also put up with me and kept me company when I most needed it.

Table of Contents

List of Figures

List of Tables

1. Introduction

In an age where computers have transformed the sharing and manipulation of an increasing wealth of biological data, high school biology teaching is falling behind. Previously, biology was a science that dealt with single molecules, single organisms, and single concepts. Now, bioinformatics – placed in the spotlight with advances such as the sequencing of the human genome – utilizes computers to store and parse large datasets, from which complex conclusions about entire biological systems can be drawn. In order to reflect this paradigm shift, high school biology courses should train students to understand systems, instead of just individuals, and to feel comfortable with computers as tools, instead of just pen and paper.

The educational theory of Bloom's Taxonomy splits the cognitive domain into six different levels: knowledge, comprehension, application, analysis, synthesis, and evaluation (Bloom *et al.* 1956). The first three skills are considered lower level skills and, though they provide a necessary basis of facts upon which learning should be built, are not sufficient to transform students into independent thinkers (Wood 2008). Teachers should therefore use various types of activities and assessments that give students opportunities to exercise their own problem solving abilities (Allen and Tanner 2002). This is what my thesis aims to do.

In this thesis, I will detail a 10-unit biology curriculum that I designed for the StarLogo TNG visual programming software. The curriculum begins with ecology, moves to DNA and proteins, and finally shifts to evolution. The evolution units are designed to tie together the two previous parts by highlighting the large-scale changes in ecosystems caused by small-scale changes in DNA.

Throughout the paper, I will focus on three student goals, around which I tried to shape each lesson:

1) *Increasing fluency in computer usage, particularly in modeling.* Students should gain familiarity with typical programming constructs, as well as be able to analyze benefits and pitfalls of various types of models.

2) *Encouraging self-exploration and critical thinking skills.* Students should approach the problems with an open mind and become comfortable presenting and testing their own ideas, rather than assume there is a right or wrong answer.
3) *Developing a "systems" mindset.* Students should understand the concept of emergent properties, the importance of randomness, and the interconnected relationships between the different pieces of the system.

Arguments for the value of these three goals will be presented in the following section, Achieving them should better prepare students to become active contributors in the real world of biology research.

2. Background and Motivation

2.1. StarLogo TNG

StarLogo: The Next Generation (TNG) is a graphical programming environment in which users program the actions of individual agent "breeds" by connecting together puzzle-shaped "code blocks" on a programming "canvas." Because the blocks can only be connected in certain ways, TNG lowers the entrance barrier for students learning to program by eliminating the frustration of syntax errors inherent in other text-based languages (Wang *et al.*, 2006). Students are engaged by the rich 3D world in which the agents live, called "Spaceland," with visuals reminiscent of many modern-day videogames (Klopfer *et al.*, in press).

The block interface design and 3D graphics of TNG improve upon its 2D text-based predecessor, StarLogo, while maintaining the same basic core: modeling decentralized systems. The original design of StarLogo was created such that students could program simple rules for individual agents to follow and, from the interactions of many such agents, be able to visualize the emergent properties. Some of the first models were built to understand traffic jams, termite mound-building, and predator-prey relationships (Resnick 1999). The latter two are still being used today and are incorporated into the biology curriculum for TNG.

```
Turtles
- Move
forward steps 1
right degs random 45
left degs random 45
```

Figure 1: A sample procedure in StarLogo TNG.

Many ecological models have already been detailed in the original StarLogo (Klopfer, Colella, and Resnick 2002), but the full potential of TNG has yet to be realized.

TNG still has all of the capabilities of the original StarLogo:

- Simple, intuitive commands for inexperienced programmers.
- Easy graphing tools for analysis of trends.
- Visualization of emergent properties of systems, particularly those that create order from randomness.

However, TNG can improve upon the original StarLogo in many ways:

- Students can exert their creativity by designing breeds and terrains.
- Students can use the game-like design of TNG to turn simulations into engaging 3D games. Games can also be used to test concepts learned in simulations (Klopfer *et al.*, in press).
- Students can visualize processes occurring over different scales and in multiple systems by utilizing the new level features.

The biology curriculum detailed in this thesis will try to take advantage of all of the above points in supplementing the teaching of two commonly misunderstood topics in biology: Evolution and the DNA-Protein Relationship.

2.2. Student Misconceptions in Biology

In this section I will detail the common misconceptions that are held amongst high school students (as well as college students and beyond) regarding evolution and the DNA-protein relationship. I will address all of these in the curriculum using StarLogo TNG.

Sources of confusion in both of the topics can be placed into three main categories:

1) Difficulty with the different *levels of organization* and being able to scale from the molecular level up to the ecosystem level.
2) Basic *ontological errors*, i.e. the miscategorization of concepts and the relationships between them. An example of an ontological error would be the belief held by many children that whales are fish because of their many obvious similarities (Ferrari and Chi 1998).
3) The prevalence of a *centralized mindset* over the idea of properties emerging from the combined efforts of many individual agents. Inherent in these ideas is the power of stochastic events, and not just particular *drivers*, that drive phenomena.

2.2.1. Evolution

The large time scale that evolution generally requires makes the concept inherently abstract. Despite tangible evidence in examples such as antibiotic resistance in bacteria, a large number of people still do not believe in evolution because it cannot be readily observed for larger species within a human lifespan. A lack of understanding about basic genetics further confuses students on the matter (Ferrari and Chi 1998), leading to many misconceptions about which traits can be inherited from parent to offspring. Though students may understand individual concepts required for evolution to occur, they do not have an intuitive understanding how those concepts fit together.

Levels of Organization

The scale of evolution begins at the *level* of genes, moving to individuals, then populations, and finally entire ecosystems (Ferrari and Chi 1998). A prevailing problem that students have when dealing with evolution is a distinct Lamarckian attitude, in which students think individual organisms adapt deliberately to an environment to become more evolutionarily fit. Where a population changes over time by varying the relative numbers of individuals with certain traits, students believe instead that all members of the population gradually change their individual traits. (Bishop and Anderson 1990, Stern and Ben-Akiva 2007). Students therefore confuse properties of the *individual* with properties of the *population.*

Ontological Errors

An *ontological error* associated with evolution is a belief that it is *event-driven* rather than a state of *equilibrium*. The following list of four attributes differentiating between events and equilibration is adapted from a paper by Ferrari and Chi (1998):

1) Distinct actions *vs*. uniform actions. Individuals within a population generally follow *uniform rules* (i.e. eat certain foods, mate with others of certain characteristics, live in a certain habitat).
2) Sequential *vs*. simultaneous. All individuals within the system are performing their own actions *simultaneously*.
3) Goal-directed *vs*. net effect. Nature does not have a particular "perfected" form in mind, rather, forms arise as the net effect of many selective pressures as well as random processes. In the study, not a single answer referred to evolution as "a *net effect* of the independent selection of many organisms competing for resources simultaneously."
4) Terminates *vs*. continuous. Evolution *continues* as long as mutations create variation within a species and hence never terminates.

The paper suggests that many student misconceptions about evolution stem from this error in categorization, with a significant correlation between students answers containing non-Darwinian ideas and student answers exhibiting an event mindset.

Presence of Drivers

Finally, many students consider natural selection to be synonymous with evolution, with selective pressures of the environment acting as *drivers* for change. Garvin-Doxas and Klymkowsky likened the phenomenon to diffusion, which is considered by most students to be taking place *only* in the presence of a concentration gradient rather than all the time (2008). They suggest that this is attributed to a prevailing attitude amongst students (and even many scientists as well) that biological systems are very efficient. Even advanced physics students, who understand the importance of stochastic processes in physics, do not believe that the "complex and often counterintuitive behaviors" of biology can rely on random events.

Because of this mindset, students have difficulty understanding that evolution can occur without evolutionary pressure. The authors found that students do not recognize that mutations are always occurring randomly and have difficultly linking the concepts of bottlenecks and genetic drift to evolution. Population bottleneck, a random reduction of the population size that causes an artificial sampling of the gene pool, and genetic drift, changes in the gene pool that arise from randomness in death and reproduction, are stochastic phenomena that are at least as important as natural selection in causing evolution.

2.2.2. The DNA-Protein Relationship

At the heart of evolution lies a basic concept of genetics, or how traits are passed on from parent to offspring by the transfer of the basic code of life: DNA. But without a firm understanding of the relationship between DNA, RNA, proteins, and physical traits, students never get the full picture of how simple mutations can give rise to such diversity, from the tiniest bacteria up to human beings. Even though the details of transcription of DNA into mRNA and translation from mRNA to proteins are taught extensively in high school biology classes, "novice biology students often get mired in all the details of genetics problems and protein synthesis, and never see the relationship between the two topics" (Mulvihill 1996).

Levels of Organization

Like evolution, an the understanding of the DNA-protein relationship suffers because students must bridge the relationship gaps between different *levels*, namely the *macroscopic* level (multicellular organisms), the *microscopic* level (single cells), and the *sub-microscopic* level (molecules within the cell). Marbach-Ad and Stavy (2000) suggest that it is difficult for students to make such a large jump in levels, noting that students often relate to humans and animals when talking about traits (larger organism for a larger concept), but shift to bacteria when talking about DNA and proteins (smaller organism for a smaller concept). In a study by Lewis, Leach, and Wood-Robinson in the UK (2000a), only 4 out of 400 students were able to correctly order the organism, cell, nucleus, chromosome, gene, and DNA by relative size. Many students also consider genes, DNA, and chromosomes to be separate entities (Saka 2006).

Ontological Errors

The major *ontological error* that students have regarding DNA and proteins is a view of genes as *matter* rather than *processes* (Venville and Treagust 1998, Tsui and Treagust 2004). As matter, genes are only passive objects passed on from parent to offspring. However, as a process, DNA is a code that actively determines the production of a protein. Thinking of genes as simple entities to be shuffled around leads to misconceptions such as the idea that each type of cell only contains the specific information needed to perform its particular function (Lewis, Leach, and Wood-Robinson 2000b). However, encouraging the *radical conceptual change* to switch between the two ontologies allows students better to understand the intricate connection between DNA, proteins, and traits (Tsui and Treagust 2004).

Presence of Drivers

The role of *randomness* is heavily downplayed in the teaching of molecular processes, with even basic concepts of diffusion and osmosis being attributed to the presence of *drivers*. In reality, many key components of DNA and protein function depend on random movements. Current theory suggests that RNA polymerase finds its target largely due to a combination of facilitated diffusion movements (Halford and Marko 2004). Ribosomes fluctuate randomly between an unlocked and locked position, with certain chemical reactions ensuring unidirectional translation in the presence of mRNA (Spirin 2004). Additionally, many systems effectively "gauge" concentrations of ligand through random sampling (a larger concentration of ligand leads to a higher probability of random receptor protein binding, which leads to a higher frequency of receptor response).

Yet even after two or three lecture-based college level biology classes at MCDB/UC Boulder, out of roughly 500 students in five different classes, many were still unable to answer the following multiple choice question: "Imagine that you are an ADP molecule inside a bacterial cell. Which best describes how you would manage to "find" an ATP synthase so that you could become an ATP molecule" (Garvin-Doxas and Klymkowsky 2008). Of the answers – a) I would follow the hydrogen ion flow, b) the ATP synthase would grab me, c) my electronegativity would attract me to the ATP synthase, d) I would be actively pumped to the right area, and e) random movements would bring me to the ATP synthase – the correct answer (e) was consistently the least or second least answer chosen in every class.

Centralized Mindset

Perhaps the biggest problem above all other problems in the teaching of the DNA-protein relationship is that the importance of proteins as the working machines behind the functioning of all living things is not emphasized. Instead, students only see isolated proteins functioning as enzymes or membrane channels. In a UK study of 482 students (aged 14-16) asking, "Why is DNA important?", 59% said DNA "defined" living things, 14% recognized its importance in providing information, and only 4 responses made any mention of the production of proteins (Lewis, Leach, and Wood-Robinson 2000a). Even undergraduates in an introductory development course had trouble understanding the concept of combinatorial control, failing to "recognize how protein interactions with other cellular components can lead to specific cellular responses" (Douglas 2008).

David Bolinsky, one of the creators of the acclaimed animated video, "The Inner Life of a Cell," says in a 2007 talk that a goal of his animation was to create a mental image for students of the cell as "a large, bustling, hugely complicated city that's occupied by micromachines." He goes on to say:

> *These micromachines really are at the heart of life. These micromachines, which are the envy of nanotechnlogists the world over, are self-directed, powerful, precise, accurate devices that are made out of strings of amino acids. And these micromachines power how a cell moves; they power how a cell replicates; they power our hearts; they power our minds.*

Without this mental image of an intricate, intertwined protein community, students cannot hope to appreciate how a sequence of nucleic acids can lead to the complexities of the many organisms roaming this earth.

2.3. Problem-Based Learning

We often think of classrooms as places where students go to learn, but we never question what they are learning. We assume they should be learning facts, but in reality, they should be learning *how to learn*. Mitchel Resnick, the creator of StarLogo, once stated in an article:

> *Teachers cannot simply pour information into the heads of learners; rather, learning is an active process in which people construct new understandings of the world around them through active exploration, experimentation, discussion, and reflection. In short, people don't* get *ideas; they* make *them.* (2002)

Particularly in the sciences, where fact can always be questioned and ideas often come from out-of-the-box thinking, teachers must work to dispel notions of absolute right and wrong and, instead, encourage creativity.

Studies have shown that learning through open-ended problem-solving significantly changes student understanding of the *nature of science*. Students who engaged in 13 open problems relating to genetics understood science to be a problem solving process, rather than a series of serendipitous discoveries. Theories were seen as dynamically changing, rather than hard unwavering facts. Student view of scientists changed from crazy old men in labcoats working alone, to regular people working creatively as a group (Orcajo and Aznar 2005). Only by understanding the nature of science can students then effectively partake in it.

Key to problem-based learning is the step where students deconstruct their own misconceptions, as well as the misconceptions of others. After developing a solution, students should present their ideas and defend them against targeted questions from their teacher and peers. Argumentation in the classroom can often be a powerful way to learn, as students defending their ideas will often end up seeing the loopholes in their own arguments (Stern and Ben-Akiva 2007). One activity using this method required students to view a video of various pedestrians voicing their (mistaken) understandings of antibiotic resistance and then to analyze the errors in each argument (Cloud-Hansen *et al.* 2008). Another activity had students discussing works of science fiction and finding the scientific inaccuracies (Bixler 2007). Critical thinking skills are built when students observe whether their current ideas are sufficient to explain a phenomenon and are then given the opportunity to revise and test new ideas as a response (Perry *et al.* 2008).

StarLogo TNG is a modeling medium that is easy-to-use and designed to encourage student creativity. This makes it ideal for students to explore their own hypotheses and take control of their own learning. Any unexpected behavior of a model challenges students to inspect their own thinking for bugs. Frustration can be used productively in learning how to slow down when faced with a brick wall.

Student blogs about using TNG for physics simulations revealed that students appreciated the problem-solving approach:

> *"Problem solving is the best way to learn because we are forced to understand every angle. The program also triggers my creativity, which I believe is not as important in school. Instead of knowing a certain way to figure something out like math problems, we have to discover different ways to [find the answer]. This type of learning is out of the ordinary, and I believe it is a necessity!"* (Klopfer *et al.*, in press).

TNG is a powerful tool, with which students can take control of their own learning. When integrated into the biology classroom, TNG can help students explore solutions to problems and, by doing so, challenge their own ideas and misconceptions.

3. Curriculum Design

3.1. On the Molecular and Cellular Scales

The initial goal of the curriculum was to provide an aide for teaching about DNA and proteins. I feel that the biggest oversight in high school biology courses is a lack of emphasis on the importance of proteins as the backbone of cellular processes. I, myself, did not recognize the vast complexity of protein networks and their importance until my undergraduate cell biology course. Additionally, while a few projects already existed in StarLogo demonstrating other difficult biology concepts like ecology, evolution, and diffusion; none dealt with DNA, proteins, or protein systems. With this in mind, I outlined the original curriculum as a set of units detailing the full process of how basic genetic code leads to the physical expressions of cells, tissues, and full organisms. The flowchart of the key relationships that should be taught together (Offner 1992) is found in Figure 2.

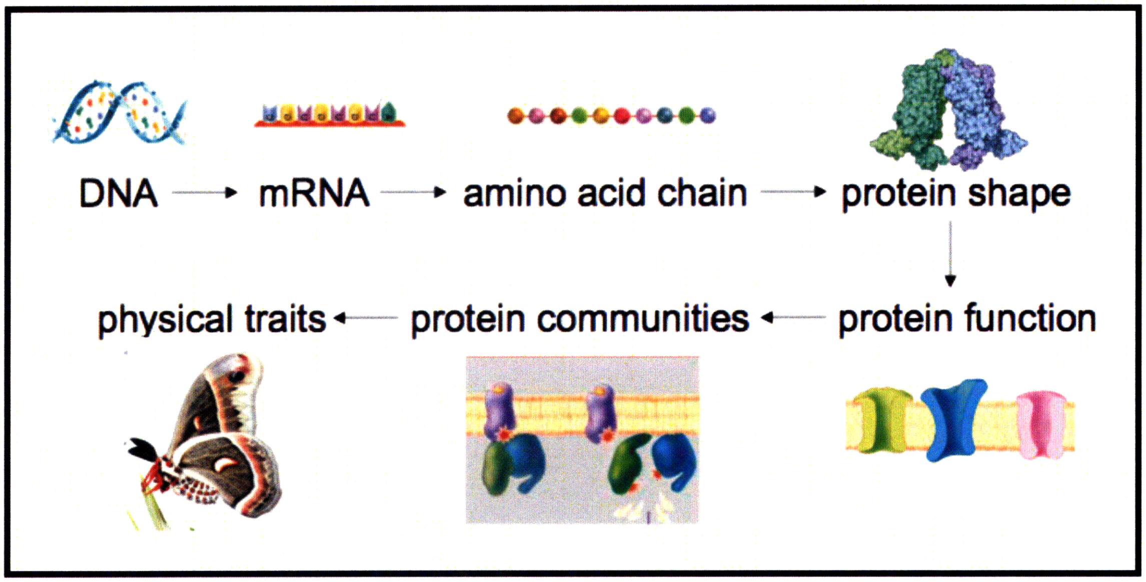


Figure 2: Key relationships between DNA and traits.

[Images taken from *Biology* by Johnson and Raven and *Inside the Cell* from the National Institutes of Health and National Institutes of General Medical Sciences.]

3.1.1 Current Techniques

The current educational literature chronicles many creative and engaging ways to depict the first three steps of the flowchart, where DNA is transcribed to mRNA and mRNA is translated to an amino acid chain. Using Lego^{TM} blocks as alternate representations of various molecules for transcription and translation is a technique that is continually being perfected (Sprehn 1993, Templin and Fetters 2002, Roberg 2004), as is the technique of engaging students in song and dance (Stencel and Barkoff 1993) or as actors in an elaborate molecular play (Garbarino-Asmus 2007). The physicality of building tangible models or moving around the room helps students remember basic ideas and processes (which would otherwise be just a mess of scientific names) by bringing the molecular movements to life at a macro scale.

Although the current techniques are effective in helping students visualize the processes of transcription and translation, they do not put the processes in any particular context. Students understand the first three steps of the flowchart, but lose track of the other four. Instead, students should be able to see the processes at work in the context of a cell, which in turn works together with other cells in the context of an organism. It is important that the entire system is functioning concurrently, such that transcription and translation are not isolated events, rather continuous processes. Nobody tells a cell to produce a particular protein at a particular time, but the genes controlling the production are switched on and off in response to a combination of environmental cues (Lewis, Leach, and Wood-Robinson 2000b, Wood 2008). As in evolution, molecular processes should be considered as states of *equilibrium* and not as *events*.

3.1.2. Proposed Techniques

In the curriculum, I wanted to take the pros of the current educational techniques and remedy the cons. In StarLogo TNG, students manipulate "molecules" that are represented by agents, whose shapes they can choose, and handle the molecular processes on a visible scale. However, they gain the added benefit of being able to place those processes in the context of a system, with many agents performing similar actions simultaneously and interacting with other agents.

Recalling the misconceptions from the literature, I then attempted to design the units to address the following learning goals:

1) *Bridging the gap between the macroscopic level, microscopic level, and sub-microscopic level.* The curriculum walks through the different scales and emphasizes how processes in one affect processes in another.
2) *Depicting genes as processes and not matter.* The DNA in the simulation does not move (unlike in real life), but it is still able to dictate the creation of complicated webs of proteins. When students create mutations in the base pairs, large failures can occur in the whole system.
3) *Highlighting the randomness of molecular movement.* In all of the units, the agents are moving randomly, yet they are all able to accomplish their tasks.
4) *Emphasizing the importance of proteins as the backbone of cellular function.* The proteins in the lessons are depicted as workers. There are many types and they are responsible for all of the basic functionality within the cell.

Through the lessons, students traverse the entire path outlined in the flowchart in Figure 2, while simultaneously engaging in creative problem solving, model building, and data analysis. The different scales of the system tie the units together, while each individual unit is rooted in problem-based learning.

3.1.3. Scales

It was important to create a curriculum that portrayed the *different scales* that are affected by molecular processes, integrating rather than separating them (Marbach-Ad and Stavy 2000). However, rather than simply building a giant system that zoomed in and out, I built a different level for each scale (Figure 3) to keep student focus on one concept at a time. Only by finishing one level can a student move on to the next. The three levels step through transcription (DNA to mRNA) at the nucleus scale, translation (mRNA to amino acid chain to protein shape) at the cytoplasm scale, and signaling chains (protein shape to protein function to protein communities) at the cell scale. Students gain immediate visual feedback while moving between levels. A fourth scale shows many cells functioning together (protein communities to physical traits), but the logistical details make it easier to portray the fourth scale separately from the first three.

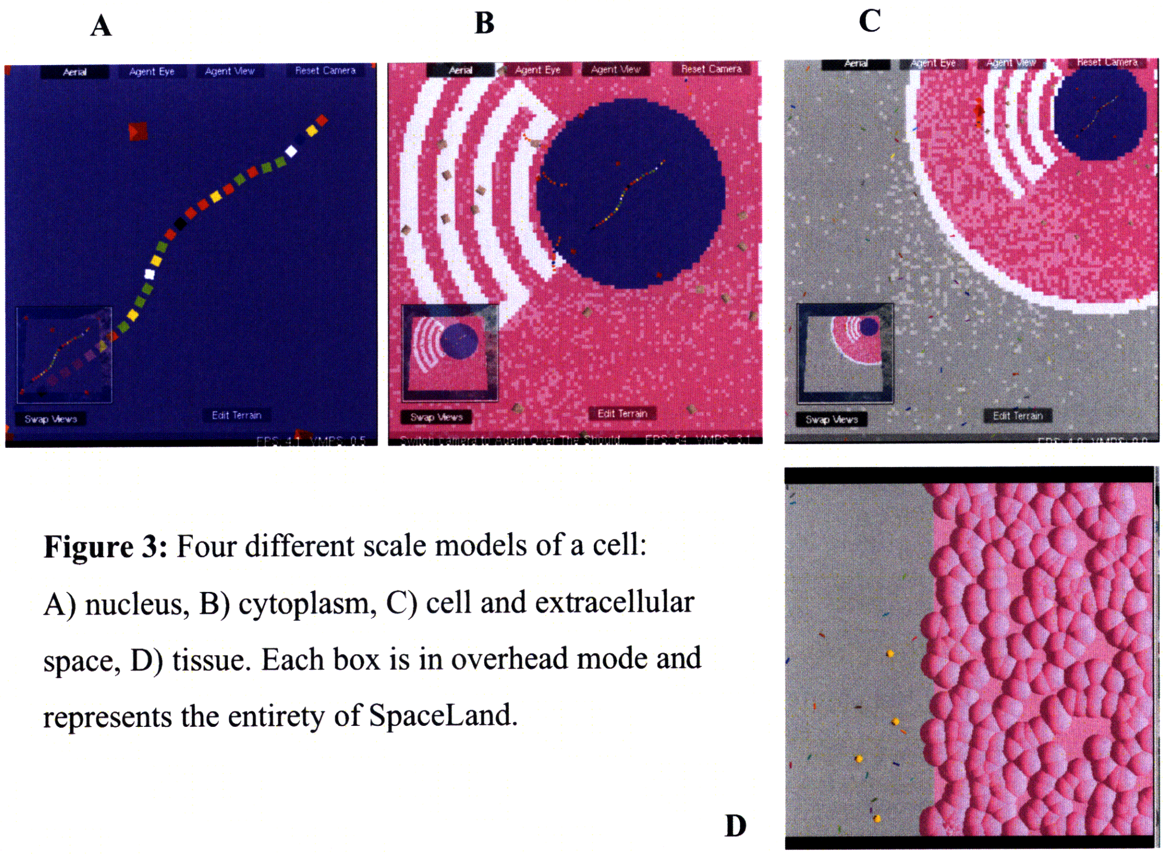


Figure 3: Four different scale models of a cell: A) nucleus, B) cytoplasm, C) cell and extracellular space, D) tissue. Each box is in overhead mode and represents the entirety of SpaceLand.

Using the different levels, students can work through topics related to each idea, while keeping in mind the big picture. Certain aspects of each level are not activated until programmed by the students, such that the students can understand how single actions by one agent can affect the function of the overall system. Once students finish building the entire functioning cell, they can "mutate" DNA base pairs and see small changes reverberate through the system to create large effects. Multiple representations that are linked allow students to "manipulate processes at different but dynamically related levels of genetics and visualize the changes made" (Tsui and Treagust 2004).

3.1.4. Problem-Based Learning

Within each unit, the activities require students to engage in problem-based learning. There are many unsolved mysteries and partially understood processes in biology that can be explored in such a way. At the transcription level: "How does RNA polymerase 'find' the gene promoter?" At the translation level: "How do proteins fold?" At the cellular level: “What is the purpose of the complexity of signaling chains?” At the tissue level: “How do cancers form?” Because these questions do not have a right or wrong answer, students can brainstorm creative answers that not only help them understand the concepts at a higher level of Bloom’s taxonomy, but also train their minds to think in a way that prepares them for research later in life.

The TNG project files for the units are designed to be tools for students to test their hypotheses. It is difficult to expect students to program small parts into an already developed system, but this can be partly remedied by creating specialized procedures as an interface abstraction. For example, one way in which I predicted students might ask the RNA polymerase to find a promoter is to program it to recognize an unspecific encounter with the DNA strand and walk along it towards the promoter. I programmed a “follow DNA” procedure that students can use to explore this option, which hides some messy details from student code but maintains flexibility, since students can combine the procedure with other commands. Of course, a teacher cannot predict everything the student comes up with, but specialized procedures can help lower the complexity.

Testing hypotheses can often yield surprising results. Finding unexpected behavior from seemingly minor changes can lead to rich discoveries by students and teachers alike. As an example, while building the tissue project I was surprised that a non-cancerous mutation led to the cancerous proliferation of the mutated cell. My cells were only supposed to replicate if there was a void nearby that needed to be filled. After thinking about the situation, I realized that the liquid movement of my cells increased the likelihood that any particular cell would encounter a void at some given time. By reducing the mobility of my cells, the cancerous activity ceased.

Because the model can never be 100% accurate to reality, an important aspect of every unit should be a discussion of the pros and cons of the model (Westerling 2008). After each activity there should be a debriefing time in which students point out the inaccuracies of the model and argue whether or not these inaccuracies invalidate the conclusions drawn from the model. Topics like transcription and translation, which involve many details that cannot be easily portrayed, should be taught before the activity so that students can fill in the blanks themselves while working with the models.

3.2. On the Organism and Population Scales

Though the DNA-protein units traverse many different scales to help students connect the interplay of molecular processes with larger observable outcomes, if we present them outside of the context of an entire population, we miss the opportunity to emphasize the way in which the same processes have affected ecosystems over the entire history of the earth. It is because of mutations in DNA – which cause differences in proteins produced, which then alter the functions of cells, which then create variations amongst organisms – that the diversity of life can exist.

Therefore, in order to take advantage of these learning possibilities, I sandwiched the DNA-protein units in between two sections: ecology and evolution. This has three major benefits:

1) Students are introduced to TNG programming and decentralized systems concepts at a familiar organism scale, without the additional task of processing content on a molecular scale.
2) The way in which organisms interact is very similar to the way in which molecules interact, creating a strong analogy between the two that allows students to relate to the molecules as agents just like rabbits or wolves.
3) The ecology units set up ideas of population dynamics that combine with the ideas of molecular genetics in the DNA-protein units to drive evolution. Hence, the evolution models that close the curriculum draw from themes woven through the previous units and tie them together.

If these themes are highlighted throughout the curriculum, students can really develop a systems mindset and see the connections between all of the elements that make up the world around us.

3.2.1. Introductory Activities

I chose to use two different introductory units to ease students into building an ecological model from scratch. Without this introduction, students have no basis for understanding decentralized systems and programming in TNG, becoming easily overwhelmed with the details of the language instead of seeing the big picture. The first unit focuses on simulations and scientific experimentation, while the second focuses on understanding decentralized systems. In the first unit, students become familiar with the aspects of TNG SpaceLand, while in the second unit, students become familiar with the TNG programming blocks.

The first unit highlights Spaceland as the *environment* where the agents live, focusing only on carrots and the abiotic factors controlling carrot growth. Students can get used to the look and feel of StarLogo by changing only sliders and not actual code. In the later ecology units, students will build on this carrot model to create their own ecosystem of carrots and rabbits. In the unit, students learn how changes in different parameters of a system, such as amounts of sunlight or nutrients, can greatly affect the way the system functions.

The second unit, VANTS (virtual ants), asks students to program a single procedure in which an agent walks and stamps a color on the ground. First, students observe a pattern created by a single agent and try to figure out what it is doing. Hal Scheintaub, who teaches physics and biology at the Governor's Academy in Newbury, Massachusetts and has been working with StarLogo for many years, has implemented this unit many times in his physics classes over the years. His classes correctly deconstructed the agent's movement with varying degrees of success, with some understanding the simple commands immediately and others creating long lists of rules over more than thirty minutes. Nevertheless, all classes felt satisfied with themselves once they figured out the mystery.

After students understand the existing code, they modify it to create their own patterns. Because each agent begins with a random color, the movements of many agents in parallel create colorful patterns on the floor that can be changed with tiny changes in the code, like a kaleidoscope. The starting conditions of the agents also have profound impacts on the final pattern. With more advanced techniques to raise and lower the terrain as well, students can even lift the pattern into three dimensions. VANTS is a creative and fun activity while providing a simple introduction to both code blocks and decentralized systems.

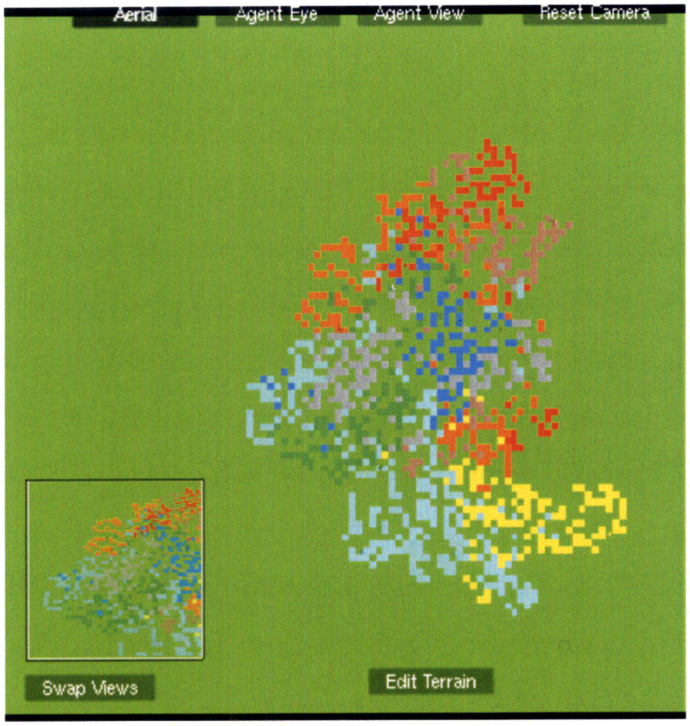


Figure 4: The original pattern created by 10 VANTS.

3.2.2. Ecological Models

After gaining a foundation in StarLogo, students can embark on building an actual ecology model. A model of rabbits and grass was detailed in the original StarLogo (Resnick 1999). The student who created it noticed that under certain parameters all of the rabbits died, and he realized that this was because the population oscillated below zero, after which there was no chance for the rabbits to repopulate. The feedback cycle of predator and prey populations is a central theme in ecology and can be demonstrated nicely without very complicated code. Walking students through the steps in building the code gives them the opportunity to learn many modeling concepts and programming constructs. The piece-by-piece design of the different rabbit procedures is based on "Program a Bunny," a joint project between the MIT Teachers' Education Program and Simbiotic Software where students learn to program simple commands for a single rabbit using a limited set of code blocks (http://simbio.com/Maine). However, it extends the program with more complicated rabbit movement and the ability to sense a carrot and track it down.

3.2.3. Analogies

Analogies are useful for introducing certain concepts because they allow students to organize new ideas within memory using a filing system already built by other life experiences (Biermann 1988). A strong analogy can be made to describe protein systems as micro-ecosystems, and hence, many of the concepts learned in ecology can then be applied to the environment within the cell, reiterating and reinforcing systems concepts.

Within the first unit on carrots and the later unit on predator-prey interactions, there is a rich parallel to the units on protein signaling and cell growth. The agents in each unit follow similar code, leading to similar behavior. Proteins have a certain half-life for degradation, while carrots and rabbits have a certain age for death. Cell growth is density dependent, limited by the surrounding nutrients, as is the growth of carrots. A comparison of their carrying-capacity graphs is shown in Figure 5A. Enzymes are produced in response to the presence of food, which effectively breakdown the food and eliminate the signal for more enzyme synthesis. This presents a feedback loop similar to that of rabbits and carrots. A comparison of these predator-prey graphs is shown in Figure 5B. Each organism in an ecosystem, in conjunction with the abiotic environment, holds a specific niche, just as each protein and organelle has a specific function.

To better highlight this analogy, the agents in the unit on protein signaling are not just generic protein shapes; rather, they take on the form of characters representing their respective responsibilities. The messenger protein is in the shape of a car, the enzyme is in the shape of a Pacman, and various other proteins take on the forms of well-known cartoon and game characters.

3.2.4. Evolution

The final units on evolution bring together why molecular changes in DNA provide the raw material necessary for the visible changes in populations. The major concepts in evolution are shown as a map in Figure 5 (Ferrari and Chi 1998, Anderson, Fisher, and Norman 2002).

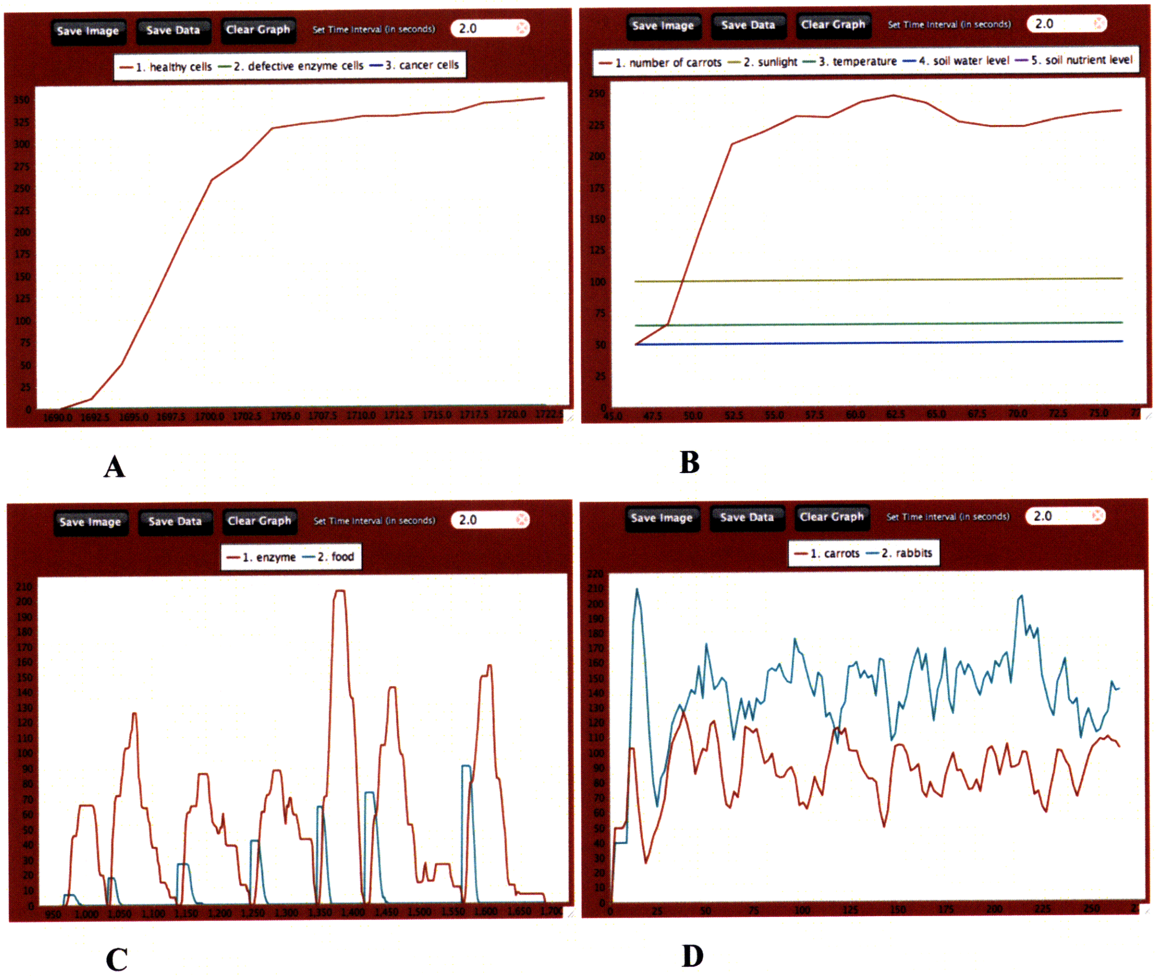


Figure 5: Correlation between micro-scale dynamics and macro-scale dynamics. A) Healthy cells in a confined space and B) carrots in SpaceLand all grow quickly until they reach carrying capacity. C) The input of substrate causes production of enzymes, which break down the substrate and therefore prevent further production of enzyme; D) similarly, rabbit populations grow in response to more carrots, but as the rabbits overeat the carrots, competition increases and more rabbits die and fail to reproduce.

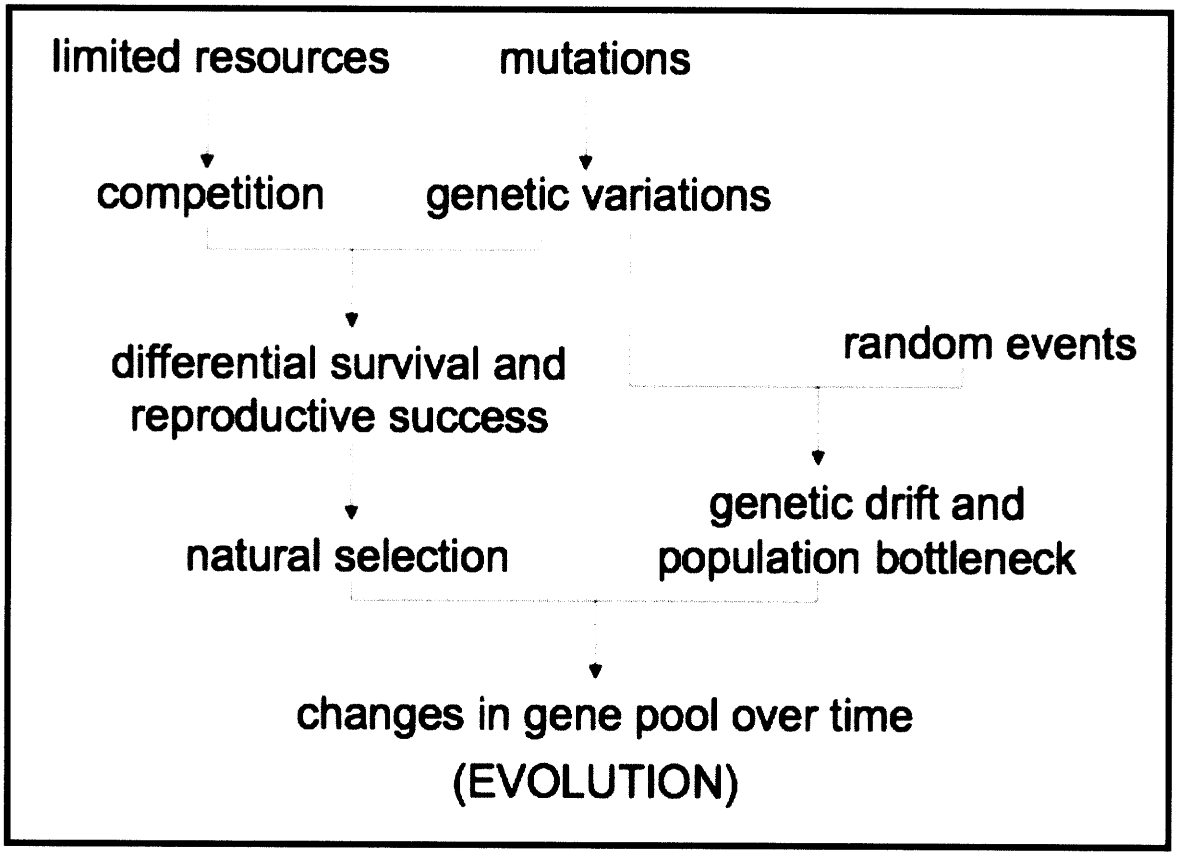


Figure 6: Map of concepts in evolution.

There already existed several models of evolution in the original StarLogo that provided the main ideas for the models in TNG. The StarLogo models use the idea of a "color gene" that may or may not affect a rabbit's fitness. The color gene provides an abstraction from details of the molecular units, but since the evolution units come after students already understand genes and mutations, it is easy for them to make the connection between the physical expression of color and the DNA code controlling it.

With the evolution units at the end of the curriculum, all of the essential elements outlined in Figure 6 are discussed. The idea of limited resources creating competition is covered in the ecology section. The idea of mutations causing genetic variation is explained in the DNA-protein section. The color gene activity highlights both random events and differential survival/ reproductive success as causes for evolution over time in a population's gene pool.

Recalling the misconceptions from the literature, I then attempted to design the units to address the following learning goals:

1) *Differentiating between properties of individuals and properties of a population.* As students run the models, they will notice that rabbits themselves do not change color. However, the average color gene in a population changes with the average frequency of each allele.
2) *Emphasizing evolution as an equilibration process rather than an event.* Because students code for small parts of the model, they know that the rabbits are all following the same rules in parallel. The rabbits do not strive for any particular goal color, but the interactions of many rabbits and carrots causes the net effect.
3) *Highlighting genetic drift and population bottlenecks as important phenomena causing changes in the gene pool.* The first of the two evolution units deals explicitly with genetic drift and population bottlenecks, showing that evolution still occurs without the presence of a driver like natural selection.

One additional theme I included in the evolution unit was the concept that "fitness" can only be defined by a particular situation. While one individual may have a certain evolutionary advantage in a certain set of conditions, a change in the surroundings may render another individual more evolutionarily "fit."

In the evolution units, the color gene can do one of three things: provide only cosmetic effect, dictate a rabbit's speed, or control a rabbit's speed *and* metabolism. When the color gene has only a cosmetic effect, the rabbits often still evolve due to genetic drift, counter to student intuition. However, the evolution does not take on an expected outcome with every trial, as the course of the evolution is determined by random changes in the system. On the other hand, a color gene that is tied to rabbit speed creates an evolutionary advantage for those rabbits that can outrun their competitors and save more energy. Finally, when the color gene is tied to both rabbit speed and metabolism, the two characteristics balance each other out, and it is not clear which rabbits have the advantage.

The final activity deals with the concept of "fitness." Previously, the rabbit's ability to survive depended only on the presence of carrots and the relative abilities of other rabbits, and it was obvious that a rabbit that could find carrots faster than another would be better equipped to survive. However, natural selection is determined by the characteristics of a particular environment (Anderson, Fisher, and Norman 2002), and what is considered beneficial in one environment might be considered harmful in another. To show this in the final activity, rabbits that run fast also lose more energy, and energy loss exponentiates at lower temperatures. Therefore, rabbits that run fast have an advantage at high temperatures, where the benefit of speed outweighs the detriment of energy loss, but are at a disadvantage at low temperatures, where their excessive energy loss causes them to die out quickly.

The system begins with all identical rabbits, which survive at the default medium temperature but die out at low temperatures. This serves to highlight genetic variation as a buffering mechanism against change. Whereas the word "mutation" brings to mind undesired effects such as disease and cancer, it also has positive effects in preserving populations.

3.3. The Gaming Element

After the main ideas were outlined, working with students from Lawrence schools helped mold the details of the curriculum to fit the game-like feel of TNG. I facilitated at six 45-minute TNG sessions in a freshman biology class and a junior/senior urban ecology class at Lawrence High School (LHS) last spring, as well as four 90-minute TNG sessions in a summer test preparation program for eighth graders at Lawrence Family Development Charter School (LFDCS). The average group size for all classes was approximately 15. The result of these sessions was not only a better understanding of the challenges in introducing TNG to a classroom, but also two games incorporated into the ecology and DNA-protein units.

3.3.1. LHS

During the LHS sessions, we proceeded through some introductory activities and attempted to implement some basic rabbit-building exercises. Students were able to program their rabbits to eat carrots by going around at random angles. However, because the sessions were not structured around any particular goal or theme, it was difficult to motivate the activities. The sessions were held only once a week, and the class periods were too short and far apart to implement a cohesive curriculum.

However, we did discover that students were most excited when they discovered breed and terrain editing. We decided to spend one session teaching students to build a game from scratch. The game was simple, where a user-controlled Pacman moved around eating treasures. For the next session we created a pre-programmed game, where Mario had to traverse a maze of obstacles to save Luigi, who was being held prisoner by Wario on the other side (Figure 6). The key was that students had to program Mario to conquer the obstacles by shooting Warios, learning to swim, donning volcano boots, climbing walls, and using teleports. Students were given instruction sheets that guided them through each obstacle but gradually became less detailed, allowing students to figure out which of the previous concepts to use.

The effect of this approach varied greatly between the two classes. The younger students were frustrated and confused, requiring a lot of help from facilitators and become unfocused when unattended to. However, those that worked through the activity became comfortable with the techniques and eventually had fun playing the game. The older class had only four students due to end of the year activities, and each student therefore received much more individual attention. These students were hardly bored and stayed occupied, even in the midst of a speaker announcement.

Figure 7: “Saving Luigi” was a game where students had to program Mario to overcome obstacles in a maze to reach Luigi on the other side.

We believed a lot of the difficulty in maintaining student focus stemmed from lack of a clear goal at the beginning of the sessions. Additionally, because the projects used during the first sessions were of higher complexity, students never became comfortable with their abilities to manipulate the program on their own. For this reason, the biology curriculum builds in complexity, using VANTS to foster a familiarity with the program and the rabbit-carrot guided programming to teach the different blocks.

3.3.2. LFDCS

The workshop activities at LFDCS tied together the two topics that were seemingly unrelated in the LHS lessons: simulations and games. In the first half of every session, students worked with models by changing slider values of certain parameters. Then, after the break, they connected the simulations to games by three methods:

1) Using the simulation to understand techniques to beat the game.
2) Programming new elements into the simulation to turn it into a game.
3) Modifying parameters using knowledge from the simulation to change the difficulty of a game.

Students enjoyed personalizing their games. They made giant doctors and hills that reached to the sky, pushing the software to its limits. In the process, they learned what made a game good and what made it bad: that an easy game is boring because it provides no challenge, while the hardest game possible is also boring because it is too difficult and therefore frustrating.

This particular organization, in which students worked with a model first to create context for the topic at hand and later developed a game relating to the model, seemed to work well. The sessions were often filled with laughter and excitement, as students ran from one computer to another trying to see what others had done.

3.3.3. Games in Biology

Because the games were so powerful in motivating student interest, activities were added at the end of each major section of the biology curriculum that turned the concepts into games. The activities build on the previous lessons while introducing new problems for students to contemplate.

For ecology, students program a first-person hunter character that helps to control an overpopulation of wolves by shooting some when the population gets too high (Figure 8). Students can maximize their scores (wolves shot) by only shooting enough wolves to maintain the population. If they get greedy and shoot all of the wolves as fast as they can, there will no longer be any wolves to repopulate the system. This activity brings up many questions regarding the role that humans play in an ecosystem.

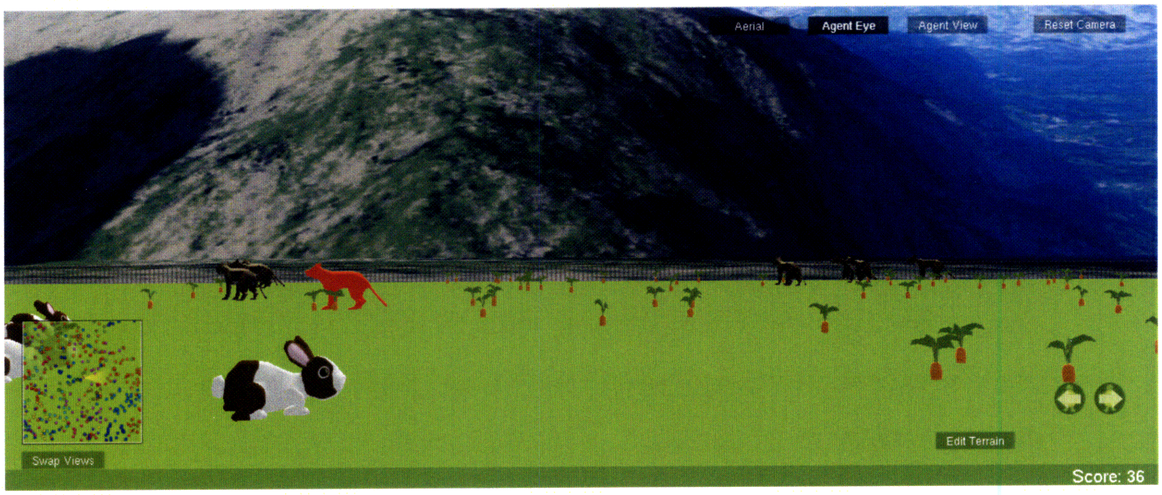


Figure 8: Students program a hunter to control wolf populations.

For cell biology, students control a first-person doctor character in an attempt to beat cancer. The doctor has many tools at his convenience, including a scalpel for excision of a growth, drugs for targeted destruction of cancer cells, and radiation for chemotherapy. Students control the difficulty of their game by changing parameters, such as the growth speeds of the tumor cells, the effectiveness of particular treatments, and the “terrain” of the body, which includes blood vessels and nerves that cannot be crossed, limiting the operability of certain cells. The game requires students to understand what makes cancers grow, the methods with which they are treated, and limitations on each of these methods.

4. Implementation Details

This section presents the details of the biology curriculum. In each lesson, I will address the lesson plan, main concepts to be learned, and some technical details of the model. An overview chart of the lessons is provided in Table 1, and detailed teacher guides with student worksheets are available online at the StarLogo TNG website: http://education.mit.edu/drupal/starlogo-tng/biologycurriculum.

For each activity, teachers should abide by three mottos:

1) *Student ideas are never wrong.* While students may propose theories that are considered scientifically implausible, we must remember we once thought the earth was flat until it was proven otherwise. It is more important to encourage students to examine their ideas and determine if they are supported by experimental evidence. For this reason, teachers should write student ideas on the board during brainstorming sessions, which can also help tie together concept between lessons.
2) *Models are never 100% accurate.* After each lesson, teachers should ask students to assess where each model correctly represented reality, and where it fell short. This teaches students always to critically assess their experiments and prevents students from harboring misconceptions due to model simplifications.
3) *Draw connections whenever possible.* The main theme in the curriculum is decentralized systems, and it is important to thread this theme into the lessons to keep a consistent anchor to refer back to. Other similarities, like the graphs of carrying capacity and feedback oscillations, should be highlighted as well.

Table 1: Overview of the biology curriculum.

Unit	Overview	Key Concepts
1	Students observe the density-dependent growth of carrots and predict changes in behavior due to various abiotic factors by changing amounts of each through sliders. This lesson encourages students to form and test hypotheses methodically.	- abiotic elements - limiting factors and carrying capacity - exponential vs. logistic growth - density dependent vs. density independent regulation
2	Students try to characterize the behavior of an agent as it creates seemingly complicated pattern. Then they modify the agent code to create their own patterns. This lesson introduces programming.	- agent-based modeling - decentralized systems and emergent behaviors - systematic changes
3	Students build a model of carrots and rabbits through step-by-step guided programming. Through this, they learn the essential procedures that the rabbit model follows (hop, eat, reproduce, die) and observe the predator-prey cycle that results. This lesson teaches model building techniques.	- producers and primary consumers - predator-prey - energy transfer and efficiency - movement, sense, aging, eating, metabolism, reproduction, death
4	Students disturb a stable population of carrots, rabbits, and wolves, causing wolves to overpopulate. They then program a first-person hunter character to shoot the wolves and discuss the role of humans in ecosystems. This lesson teaches gaming techniques.	- secondary consumers - small populations - overpopulation - role of humans
5	Students program the RNA polymerase to "find" its target DNA, with the goal of having the most efficient search algorithm. Then they discuss the plausibility of their method for real proteins.	- nucleus, cytoplasm - transcription and translation - indicator and signal sequences - random movements
6	Students play with the protein folding program, generating random sequences and seeing if they fold under certain rules.	- protein folding - covalent vs. noncovalent bond - shape = function
7	Students create another gene sequence to code for an enzyme that is only produced in the presence of its substrate in the extracellular space, creating a feedback loop. They then mutate the gene sequence to see that the protein no longer folds properly.	- extracellular space - receptor, ligand, signaling - positive and negative feedback - types of mutation
8	Students grow a single cell into a full organ. They notice that mutating an enzyme gene after growth only affects a single cell, while mutating before affects all progeny cells. Then they mutate a replication gene and see that it causes cancer. They program a first-person doctor to treat the cancer.	- cell division and cell death - mutations in somatic cells vs. mutations in gametes - cancers and tumors - cancer treatments
9	Students begin with rabbits with genetic variation in a "color gene." The gene has no selective advantage, but the population still converges to a single color due to genetic drift. Then students link the gene to speed and energy loss for directional and stabilizing selection. Floods show population bottleneck.	- genetic variation - competition - genetic drift - population bottleneck - natural selection (directional, stabilizing, and disruptive)
10	Students vary mutation rates in the population and the temperature. They see what is considered "fit" in one set of conditions may be "unfit" in another.	- mutations as basis for evolution - adaptation - "fitness"

Unit 1: Abiotic Factors (Carrots)

Goals:

The goal of this introductory project is to understand the value of using models and simulations to test hypotheses that may be difficult to test in real life. Students should learn to formulate hypotheses, perform methodical tests, and present data-driven results. In particular, students should realize the importance of sequentially isolating parameters to pinpoint the source of a phenomenon. In terms of understanding ecosystems, students should realize the importance of abiotic factors in ecosystem functioning and how these factors can limit possible growth.

Lesson Plan:

The lesson begins with students discussing the elements that make up an ecosystem. Of these, the lesson focuses on the interactions between producers and the abiotic environment. Students are introduced to the idea of using a model to simulate carrot growth and discuss the importance of models and simulations in scientific inquiry.

The model incorporates six factors that affect carrot growth: physical crowding, sunlight, temperature, soil water level, soil nutrient level, and soil nutrient distribution. By default, the limiting factor on carrot growth is physical space, which causes the population size to exhibit the classic graph of logistic growth. Students are then asked to investigate how each of the other factors affects the population size. Though it is not explicitly stated that they should only vary one factor at a time, they may quickly find that it is difficult to track changes with too many variables in play.

sunlight 10.0 (0.0–100.0)
temperature 65.0 (0.0–100.0)
soil water level 50.0 (0.0–100.0)
soil nutrient level 100.0 (0.0–100.0)
soil nutrient distribution 25.0 (0.0–50.0)

Figure 9: Carrot simulation after varying the amount of sunlight in the system.

Technical Details:

The model deals with carrots whose growth rates are determined by five changeable (and one preset) parameters. These parameters can be categorized as either density dependent or density independent. Carrot energy production rate and carrot reproduction rate is determined by the density dependent factors (sunlight, soil water level, soil nutrient level, soil nutrient distribution), while carrot death rate is determined by the density independent factors (temperature, soil water level). The carrots use the following procedures:

- Increase age.
- Decrease energy.
- Reproduce at certain energy unless limited by space.
- Produce energy according to availability of density-dependent factors.
- Die at a certain age or with no energy, or more with density-independent factors.

Density dependent factors:

- *Reproduction:* Carrots can reproduce if space allows.
- *Energy production:* Carrots can produce a certain maximum energy per iteration. However, low sunlight, soil water, and soil nutrients can limit the energy production to a fraction of the maximum. Of those three parameters, whichever creates the lowest fraction is the limiting factor, and only that much energy is produced.

Density independent factors:

- *Death:* Carrots die from old age or lack of energy. They also die with some probability according to soil water level and temperature. To implement this probability, each carrot is assigned a "survival" value between 0 and 1 for each parameter. At each parameter value, threshold values are calculated according a function of desired behavior, and carrots with survival values above the threshold die.
  - Temperature: Carrots grow best at 60-70 degrees. Thresholds are determined by a Gaussian distribution centered around 65 degrees, so all carrots survive at 65 degrees, and survival rates drop off as the temperature gets hotter or colder.
  - Soil water levels: Though soil water is necessary for growth, too much water in the soil deprives carrots of oxygen, effectively drowning them. Thresholds are determined by a distribution that drops off at high soil water levels.

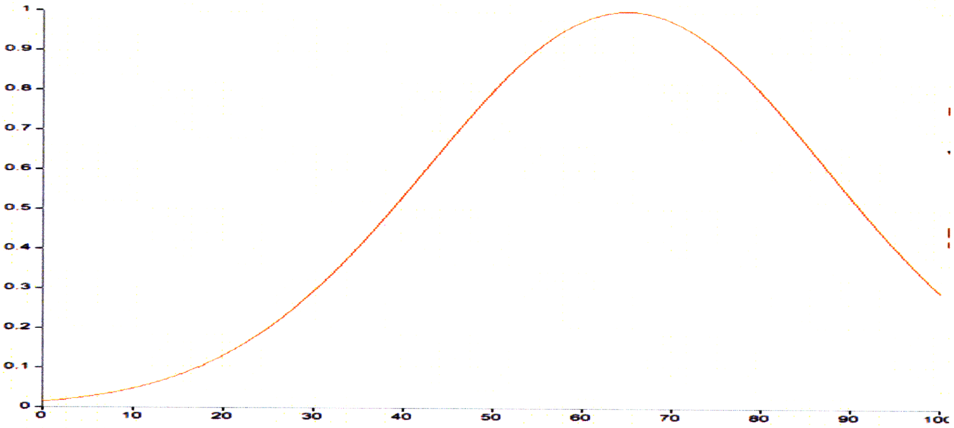


Figure 10: Probability of carrot survival at different temperatures from 0-100.

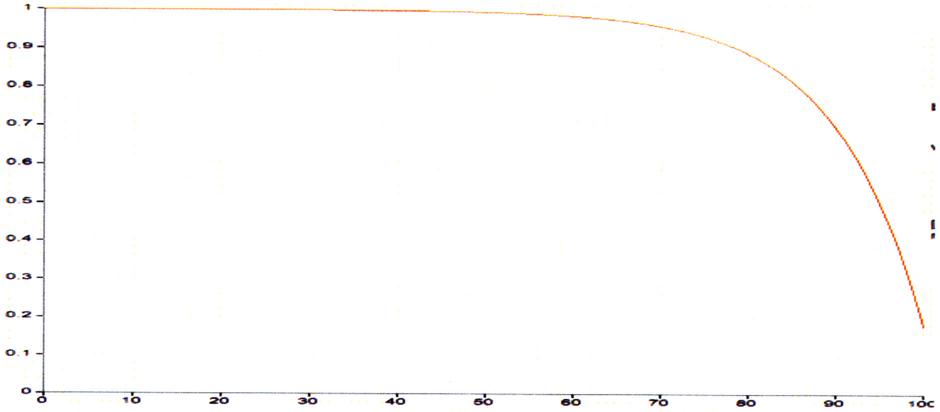


Figure 11: Probability of carrot survival at different soil water levels from 0-100.

Unit 2: Emergent Behaviors (VANTS)

Goals:

The goal of introducing VANTS in the second lesson is to allow students to understand the concept behind decentralized systems and also become familiar with using StarLogo blocks to program. Learning the basic commands and setup in StarlogoTNG during VANTS will also allow students to gain a basic understanding that will make the next activity, programming a rabbit, easier to implement. Students can see large changes arise from simple behaviors and minute changes in the program. Additionally, VANTS allows students to be creative, explore, and produce their own individual products.

Lesson Plan:

The lesson begins with students viewing the behavior of a single virtual ant and trying to determine the code dictating this movement. Students can often be extremely surprised and skeptical that there are actually only six lines of code. Students then change the model slightly to incorporate more than one agent and see how their combined movements can lead to larger patterns that the agents themselves do not design.

Finally, students get the opportunity to experiment with various blocks in the program, altering the agent movement code to generate different patterns. This last section allows students to be creative and gives them a take-home product in the end.

Technical Details:

The initial code involved in VANTS is very simple. An agent moves by walking forward, then stamping their color and turning right if the current patch color is green, or stamping green and turning left if the current patch color is not. The pattern is easily perturbed by changing the initial conditions (number of agents, their headings, and their locations), or changing their movement code (stamping different colors, raising or lowering the terrain, turning and moving different amounts).

Unit 3: Predator-Prey (Carrots and Rabbits)

Goals:

The goal of this programming project is for students to gain an understanding for how to build their own models. The program is relatively simple, and the activity should reinforce understanding of block commands that they already know and introduce them to new ones that they don't. Additionally, the procedures that the rabbits are used in later models by various species, so learning them gives the students a better general understanding of how those models work.

Lesson Plan:

The lesson begins with students discussing the processes defining an organism's lifespan. Organisms (rabbits in this particular model) move, sense, age, eat, metabolize, reproduce, and die. Students work through a step-by-step guided programming activity that walks them through creating a rabbit breed and writing procedures to bring the breed to life. After each step, students can run the program and see how the rabbit behavior has changed. The constant feedback from the program with each minor change provides opportunities for discussions about various programming concepts and how to debug an unexpected behavior. Finally, students can discuss the graph that results from the fully functioning carrot and rabbit system, particularly the dynamics of predator and prey.

Technical Details:

The model uses a modified version of the carrot code, where carrots are only limited by the factor of physical crowing. The rabbits use the following procedures:

- Turn toward any nearby carrots if possible and move forward.
- Increase age.
- Decrease energy.
- Reproduce after a certain age and energy.
- Die at a certain age or with no energy.
- Eat carrots during collisions, gaining 50% of the carrot's energy.

Unit 4: Human Impact (Hunters)

Goals:

The goal of this project is to focus on the question of where humans fit in the ecosystem. Ethical and philosophical questions about what is natural for an ecosystem may become a source of debate, and the discussion generated will be valuable for students to form their own opinions about a topic that has no right or wrong answer. Additionally, the gaming element of the activity should be fun while introducing the concept of how to program keyboard controls and score (should the students want to use the program to make games in the future).

Lesson Plan:

The lesson begins with a model similar to the carrot and rabbit model, except with the addition of a secondary consumer: wolves. The system is relatively stable, but occasionally all of the rabbits and wolves (and sometimes carrots too) die out. If all students run the model, it is likely that at least one trial will result in species extinction, which provides a good opportunity for discussing why random events have larger impacts on small populations, such as that in the TNG model. In fact, this is the difficulty in making more complicated ecological models in TNG, since the stability of the models depends largely on the interplay of various parameters involved. The more parameters there are, the more difficult it is to achieve balance.

Students then perturb the model by changing the energy that wolves need to reproduce. This causes the wolves to overproduce offspring, resulting in too many wolves depleting the resource of rabbits and the ultimate demise of the ecosystem. In order to counter this effect, students program a first-person hunter character that they control via keyboard strokes. The simulation becomes a game, as students try to achieve the highest score, calculated by the number of wolves that they have killed in the game. Through this, students may notice that the only way to kill many wolves is to make their actions sustainable, killing enough wolves to maintain the rabbit population but leaving enough wolves to repopulate.

The resulting discussion of this lesson investigates the niche of humans in the ecosystem. This raises many of the following questions: Is hunting good or bad, or neither? Is it fair to allow killing of as many wolves as possible, as long as the population is sustainable, or should killing be limited to what is necessary for the ecosystem? Is it the human's "job" to maintain the population when it would otherwise die out, or is it only the human's job to correct changes caused by human activities? What does it mean for ecosystems to be in their "natural" state, and are humans and human activities considered part of this state?

Technical Details:

The model adds a wolf breed on top of the carrot and rabbit model. Carrots and rabbits follow the same code as before, except that rabbits have the additional sense to run from wolves. The wolves follow commands similar to those of rabbits. Once the hunter is programmed, it shoots bullets that travel on their own and automatically update the score when they hit any animals. Positive points are given for killing wolves, and negative points are given for killing rabbits.

Unit 5: Transcription and Translation (DNA, mRNA, and Protein)

Goals:

The goal of this project is to introduce transcription and translation in the context of TNG. The setup of the project, which is also used in Unit 7, is such that students progressively move from the nucleus of a cell through various scales up to the organ level, seeing various processes happen in their given contexts. Students will get a chance to discuss one of the many open problems in biology, which is how RNA polymerase tracks down a gene promoter to initiate RNA transcription. They program their ideas into the system and, by doing so, set up the receptor and messenger proteins that are used in Unit 7.

Lesson Plan:

In this lesson, students deal with transcription in the nucleus and translation in the cytoplasm and endoplasmic reticulum. Prior to the lesson, the students will have already learned the details of the two processes, but students generally do not understand just how much is still unknown. In the project file, everything is setup for RNA polymerase to transcribe a strand of mRNA, which goes to either a free or bound ribosome (depending on the gene) and gets translated into a protein. However, the polymerase itself does not move, and it is this search procedure that the students must program.

Current theories about this process include polymerase randomly moving until it binds with a promoter; polymerase randomly moving until it finds a strand of DNA, after which it walks randomly up and down the DNA molecule until it hits a promoter; or polymerase randomly moving until it collides with DNA, which it then uses as a reference, “bouncing around” until it finds a promoter. For this reason, some helper procedures, such as “follow DNA forwards,” are pre-programmed for students to use. Additionally, students may give the polymerase “senses,” attracting it to any nearby promoters. Students gauge the effectiveness of their search methods by the speed with which their mRNA is produced. Discussion following the activity should bring up whether or not their strategies are plausible models for real proteins (for example, a polymerase molecule does not have the ability to sense a particular DNA sequence from far away).

Once the mRNA molecules are created, the students can move from the nucleus to the cytoplasm, where the mRNA is automatically translated into protein that is transported to the cell membrane. While this activity is only for show, it provides students with a visual for the processes in the context of the cell. Also, the proteins that are translated are membrane receptor and messenger proteins that are used in later activities.

Technical Details:

The model consists of three levels that are traversed: nucleus, cytoplasm, and cell and extracellular space. It begins at Level 0 (nucleus) with a single strand of DNA and 5 RNA polymerase. The DNA consists of four types of nucleic acids (red, green, blue, yellow) and two indicator sequences (white for a promoter and black to indicate "stop"). The RNA has a boolean variable "reading?" to indicate whether it is actively searching for a promoter or rather trying to transcribe the mRNA after finding one.

When students complete the transcription task, they can move to Level 1 (cytoplasm), where there are scattered ribosomes that read the mRNA and translate it into protein. In order to show that proteins only take on a particular shape after folding, the mRNA is first translated into a chain of amino acids, which are then in turn replaced by a separate protein breed after translation is complete. For simplicity, there are only two nucleotides per codon, with amino acids matching the color of the first nucleotide. Completed proteins move to Level 2 (cell and extracellular space) to embed in the membrane.

The model introduces three concepts that are difficult to implement in TNG: scale and linking. The scales portray the different levels from nucleus to organ; linking makes it possible to have chains of nucleic acids and amino acids, as well as bonds between molecules; and color compartments make it easy to restrict movement of molecules within certain parts of the system.

A **B**

Figure 12: Screenshots of A) DNA being transcribed in the nucleus and B) vesicles escorting proteins from ribosomes in the endoplasmic reticulum to the cell membrane.

SCALES

In order to portray concepts of scale in the model, it is important for each level to maintain consistency in colors, as well as relative sizes and locations of objects. This is achieved in the projects by creating shared variables that are reset upon initiation and can be referenced at every level. The only things that now change between each level are its center of focus and its scale, which are stored in a list for each level. Because the levels are referenced by string and not index, an extra procedure is used to parse through levels and retrieve the necessary information.

When moving between levels, agents recalculate their locations by checking that the level has changed, then updating their coordinates and sizes according to the new center of focus and scale. If the terrain is modified, a special "observer" agent can reset each level's terrain according to the scale. The design of levels in TNG is such that agents exist only in one level at a time, which means that scaling can only be achieved if users change levels with the specialized procedures.

LINKING

Agent linking is another concept that is not available in TNG. However, a simple way for this to be implemented is for agents to have an agent variable, “attachment ID,” to keep track of other agents to whom they are linked. A value of -1 indicates that the agent is not linked. A procedure, “link [agent 1] [agent 2],” updates this variable for both agents involved in a link, and another procedure, “delink,” sets the value to back -1.

Once an agent is linked to another agent, it is very easy for them to communicate. An agent can tell its link to do something by using the command “ask agent [ID]” with the given attachment ID. The agent can also follow its link by setting its location to be the same as that of the link and then stepping backward.

A similar idea allows agents to be chained. Chained agents have two variables, “agent ID ahead” and “agent ID behind,” which are set upon creation. When any agent receives information that pertains to the entire chain, the agent passes the information in both directions like a game of telephone. The agents on the end are distinguished by one of the two variables set to a value of -1. In this way, it is very easy to create a leader that is followed by the rest of the chain. Additionally, any agent that comes in contact with the chain can move along the chain by referencing the agent ID ahead/behind values.

COMPARTMENTALIZATION

By default, agents can traverse any terrain in SpaceLand. However, by giving each agent a “list of ok colors,” they now know where they can and cannot step. Another procedure, “forward-limited,” forces agents to check that their destinations are acceptable locations for them to be and turns them around otherwise. This can be used in all future code without further concern for movement restrictions.

Unit 6: The Folding Problem (Amino Acid Sequences)

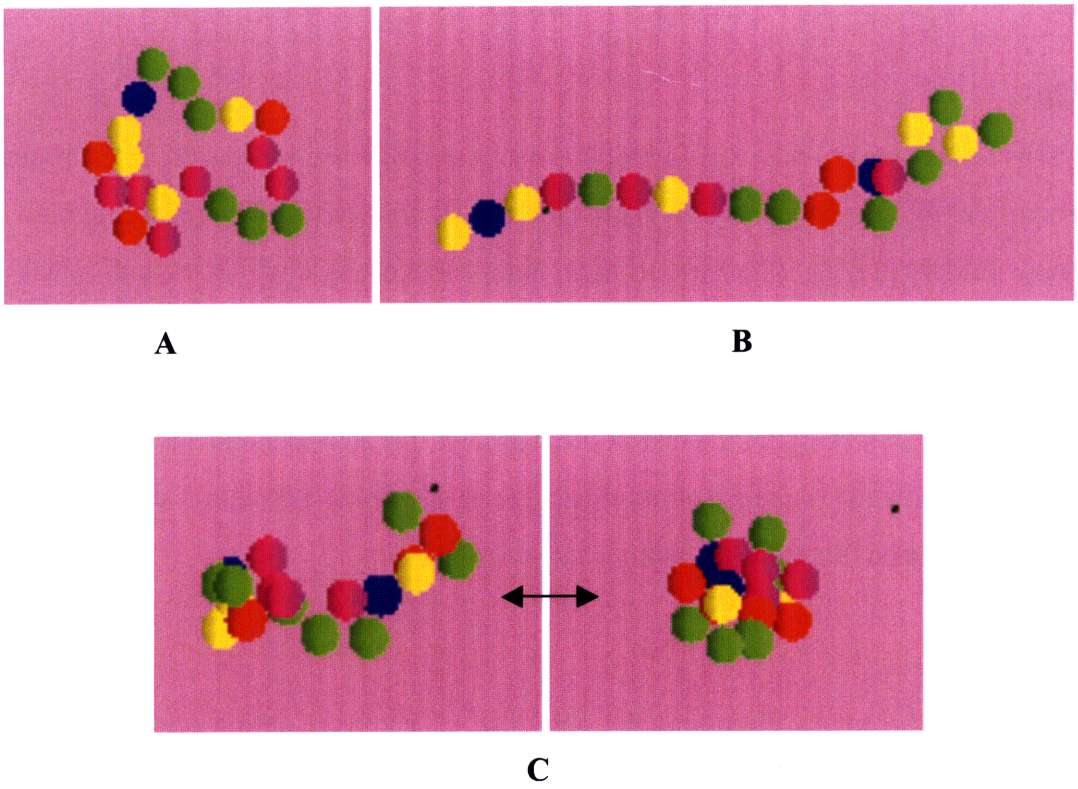
Goals:

This lesson aims to take a closer look at a key step that is glossed over in the previous unit: protein folding. Students will get to tackle another open problem regarding how proteins fold by playing with a model of an amino acid chain in which the individual amino acids have certain non-covalent bonding rules. In creating different chains with different bonding rules, students can partake in "experiment by synthesis," drawing conclusions by observing trends that occur. Through this activity, students should come to appreciate how the sequence of amino acids can determine a protein's shape, and how the protein's shape therefore determines its function.

Lesson Plan:

The lesson begins by discussing the step in which chains of amino acids become functioning proteins. Some students may know the "shape = function" mantra, but it may not be obvious why proteins take on certain shapes. Teachers should then introduce to students the concept that every amino acid residue has a certain property. Some are ionic, being attracted to residues of opposite charges and repelled by residues of the same charge. Some are hydrophobic or hydrophilic, clustering together to hide from water or reaching outward to form hydrogen bonds with water. Some actually come together to link as a covalent bond. It is important to note that proteins minimize the energy required to maintain these connections (like animals huddling together to keep warm).

In the project file, the sequence of amino acids follows very simple rules: red and green repel, and blue and yellow attract. By specifying a length, the student can generate a random sequence of amino acids and see how it folds under these rules. Some sequences may not fold into any stable conformation, but here it is important for students to understand that many amino acid sequences do not fold in real life, but they are selected against as non-functional proteins. Sequences that are particularly interesting can be saved if students create a special variable and copy the temporary sequence values. These can then be used to regenerate the chain for demonstration later. Some interesting structures are shown in Figure 13.


A

B

C

Figure 13: Interesting protein conformations generated by the protein folding model. A) A stable "star" shape that rotates in a circle around its center. B) A stable "flagella" shape that travels the terrain with a tail that follows. C) Two conformations of a protein that does not fold into a single stable shape.

Students can also change the rules easily, or perhaps add extra elements such as water or oil. A discussion following the session may focus why protein folding is still an open problem. In particular, real proteins are thousands of times larger than those in the model, yet they reach their low-energy conformations in very little time – much less time than would be required by trial an error. There are other proteins, such as chaperonins, which help guide proteins in folding and prevent inefficiencies that may occur at local energy minima. However, the basic ideas remain that a lot of protein folding is primarily driven by basic interactions between the amino acid residues, and in this sense the model is a practical reflection of real life.

Technical Details:

The model generates random sequences of five amino acid types (red, green, blue, yellow, pink). These amino acids are chained together in the same way that DNA and RNA are in the previous unit. The method by which they move while following certain rules is that each amino acid picks one of the non-covalent bonding rules it is given to move closer or farther apart from another amino acid, then moves towards the agent ahead until it resets its covalent bond link distance. To ensure that each amino acid maintains the proper distance, the first residue in the chain moves first, then passes the task to the agent behind it.

There are currently three supported procedures for students to use in programming their amino acids. In all three, the amino acid gets a list of agents within its sensing radius that it wants to either move toward or away from. The first strategy is a greedy algorithm, where the amino acid picks the first agent off the list and moves toward or away from it. In the second strategy, the amino acid parses through the list to determine the closest agent (and hence the one with the strongest force) and moves toward or away from it. The third strategy is more complicated, with the amino acid summing the forces from every other amino acid, dividing by the distance from that amino acid, and moving according to the net direction of attraction and repulsion. The first strategy, while simple, yields dynamic structures which move across the screen and pulse as real proteins do, but it is less accurate. The last strategy is more accurate, but the resulting proteins often converge to a static structure. The middle one is, as expected, somewhere in the middle. However, students should be given the opportunity to experiment with all of these, and discussion can focus on the model's accuracy to real life.

Unit 7: Signaling and Feedback (Protein Systems and Cells)

Goals:

The goal of this project is to demonstrate some of the complexity in protein signaling, as well as highlighting feedback as a mechanism of transcriptional control. Because of the setup of particular "gene sequences," students can then mutate single nucleotides and notice which mutations cause an error in protein folding. This gives a bigger picture of all of the processes working together in the cell, as well as introduces the concept of mutation at the cellular level, which will then apply at an organismal level and then a population level in the Units 8 and 10.

Lesson Plan:

In this lesson, students will be guided through adding another sequence to the genome that encodes an enzyme for breaking down substrate. The transcription of this enzyme is turned on in the presence of substrate. The enzyme must break down the substrate in the extracellular fluid, yet the substrate cannot cross the cell membrane. Because of this, there is a signaling chain consisting of the receptor protein and the messenger protein from a previous unit. The receptor protein can only exist in the membrane, and the messenger protein can only exist within the cell, but together, the receptor protein can recognize substrate outside the cell and tell the messenger, who then passes the message into the nucleus, allowing DNA to be transcribed and enzyme to be produced.

Key to this lesson is the fact that the enzyme produced breaks down the substrate, thereby eliminating the signal for production of more enzyme. This is a typical feedback loop that controls many cellular processes. As students observe the graph, they may notice that the amount of enzyme produced plateaus with more food input. When they deconstruct this idea, they should realize that the enzyme production is limited by the number of receptors, which can only bind a limited amount of substrate at once.

After observing this feedback loop, students can then "mutate" base pairs in their DNA. The mutations are all point mutations, but they can cause silent mutations, missense mutations, or nonsense mutations. The proteins with errors will not change into their respective "breeds," and will cause the system to be unable to break down substrate.

By mutating any of the three proteins (receptor, messenger, enzyme), students can see that any breakdowns in the production chain can prevent the final output of product.

Technical Details:

The model builds upon the model from Unit 5, extending the capabilities of the receptor and messenger proteins, and introducing enzymes and food. Additionally, the mutations change the colors of single base pairs of DNA. All processes take place on Level 2 (cell and extracellular space). Students program the enzymes by adding a gene to the system. Receptor and messenger control message sending with the boolean variable "has message?" and promoters control transcription with the boolean variable "received message?" A DNA base pair is mutated by simply setting its color to one of the four random nucleotide colors.

A **B**

Figure 14: Screenshots of A) receptor and messenger proteins and B) enzymes exiting the cell to break down the substrate.

Unit 8: Diseases and Cancer (Cell Communities)

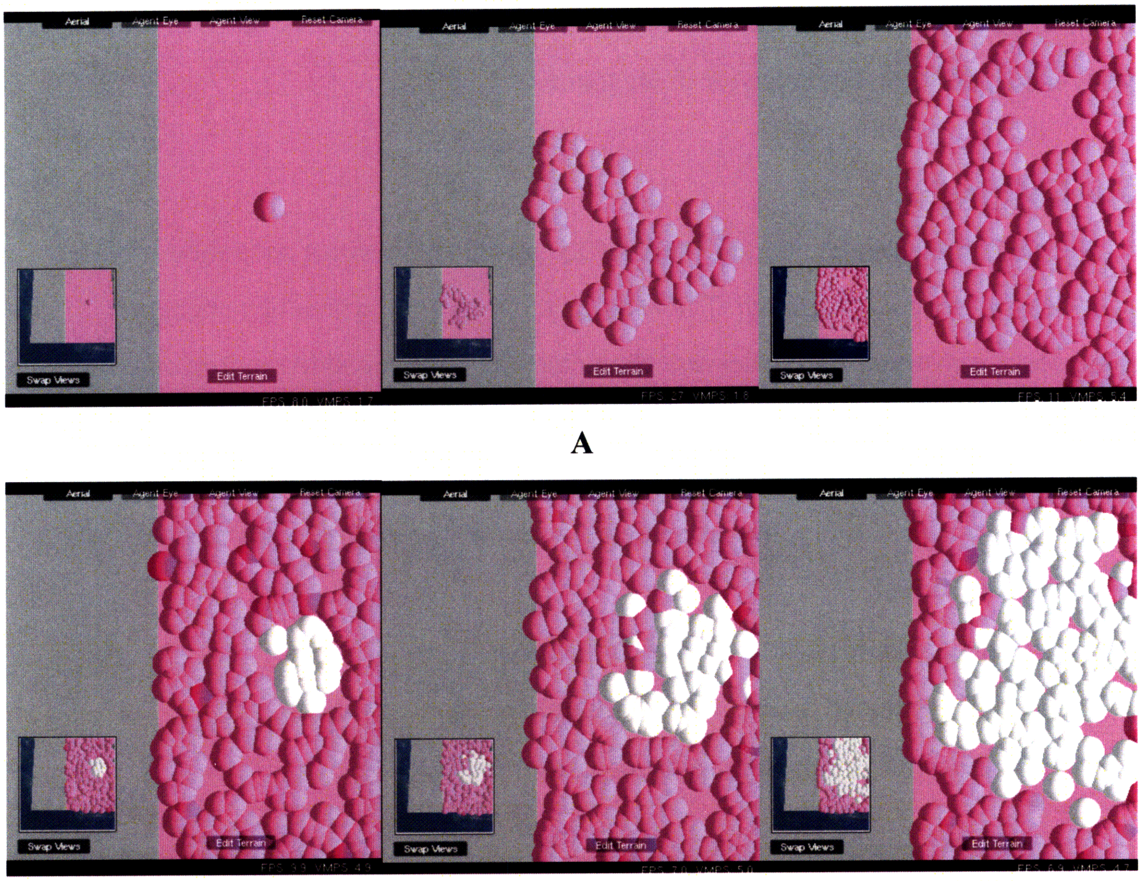
Goals:

The goal of this project is to focus on a level higher than mutations in a cell and understand the impact of mutations in a community of cells. This builds on the previous unit, but it focuses on the difference between a mutation in a somatic cell (or alternatively, a differentiated cell) and mutation in a gamete (or alternatively, a stem cell). This comes down to a higher-level understanding of why mutations can cause diseases to arise in offspring but do not generally cause spontaneous disease in individuals themselves. The exception to this is cancer, which is then discussed in greater length.

Lesson Plan:

In this lesson, students "grow" an organ from a single stem cell. After growth, they then mutate the enzyme gene in one of the cells and see that it has almost no effect on the organ's ability to break down substrate, since it is a loss-of-function mutation and there are many more cells to compensate. However, if students mutate the single cell prior to growing the organ, all of the progeny cells that comprise the organ have the same mutated DNA. Now none of the cells can process substrate. Hence, if mutations arise in gametes or early on in development, they can cause much greater effects in organisms than mutations in the fully developed adult. However, mutations accumulate in adults over time and lead to problems, which is suspected to account for part of the phenomenon of aging.

Students then mutate a different gene controlling replication, which causes a single cell to replicate uncontrollably. The particular gene causes the cell to give birth to three offspring whenever it should give birth to one. Additionally, cancer cells push other cells out of the way very slightly to give themselves more room. Students may notice that many of their cancer cells do not cause tumors and wonder why. This is because the cancer cells are never given the opportunity to actually replicate. It is considered that most cancer cells have mutations that render them unable to *stop* replicating, but not necessarily to *start* replicating. In the simulation, the cells that do grow into tumors happen to be, by chance, next to a cell that dies and leaves an open space to be filled.



A

B

Figure 15: Stages of cell growth: A) Healthy cells from single cell to tissue. B) Cancer cells from a mutation in a single cell to the growth of an entire tumor.

After discussing the growth of the cancer cells, students can change various behaviors of the cells to see which ones lead to a higher risk of cancer. Such parameters include the mobility of the cells, the speed of replication, the amount of space they physically take up, how big of a gap in cells triggers replication, and the age after which they die. Students then use what they learn from these experiments to design levels of a game.

In the game, students program a "doctor" character that goes into the system and cures the cancer. The doctor has three tools: a scalpel, targeted drugs, and chemotherapy. What makes the game difficult is the rapid proliferation of cancer cells in the presence of empty space, so doctors must be swift in taking "all or none" of the cells. Students can control this difficultly level in their game by altering the parameters that they tested. They can also limit the movements of the doctors by drawing a maze of impassable blood vessels and nerves. Because the cells only grow on pink patches, the color of the terrain serves as a code for agent behavior in the game.

Technical Details:

The cells in the model operate similarly to carrots in the carrot and rabbit model, under density-dependent growth. The cells send out enzyme when they encounter molecules of substrate. An "observer" character wanders around to random cells and mutates them if the "mutate" button is pressed. Pink cells are healthy, dark pink cells are dying, purple cells have a mutation in the enzyme gene, and white cells are tumor cells.

The doctor moves according to keyboard controls. It has three tools that are also operated by keyboard:

- Use scalpel (kill all cells with given radius)
- Send drugs (hatch drugs, which are attracted to nearby cells).
- Administer chemotherapy (ask one in three cells to die).

Unit 9: Competition, Selection, and Genetic Drift (Gene Populations)

Goals:

This lesson begins the two evolution units, looking at the effects of competition within an existing gene pool. Though the students touched on the importance of chance events in Unit 4 when talking about small populations, this activity should bring the role of random events in population dynamics into focus. In particular, it is a common misconception that evolution is driven solely by this idea of "survival of the fittest." After this activity students should realize that evolution can also arise from pure chance as well as competition for resources..

Lesson Plan:

This lesson begins with a student discussion of the how the ecology unit was related to the DNA-protein unit. Students should note many parallels, like the carrying capacity and predator-prey graphs, but they should also note the relationships between the levels. The key point to focus on is the fact that all of the agents of a certain breed within the carrots-rabbits-wolves model were exactly the same. In fact, we wanted them that way to highlight that random events caused each agent to act in a slightly different way, although their code was identical. In reality, though, every individual is different due to the genetic variations that arise from mutations in DNA. Students should then discuss how they think genetic variations affect a population. Many may know about evolution but believe in a notion of "survival of the fittest."

When students begin the activity, they are introduced to a "color gene" in the rabbits. The gene has no effect on the rabbit's abilities, only the physical color it appears. Students should hypothesize what will happen to the average color of the rabbits over time, then run the model multiple times to see the outcome. The rabbit population will generally converge to a single color, but the surviving colors will probably be different for most students between each run. Although there were no selective pressures dictating which genes survived, many random events caused genetic drift to happen.

Students then link the color gene to speed, which is done by simply changing the code for how fast the rabbit runs. The rabbits that run faster are able to track down more carrots, and after many runs, the lighter colored rabbits almost always survive, providing an example of directional selection. Then students link the color gene to a second factor, energy loss. The experiment is no longer so clear-cut, as the faster rabbits tend to also lose more energy. Many times there will be a stabilizing selection, but occasionally the graph will show disruptive selection or random behavior. Hopefully the large quantity of experiments that the class performs will collectively show a certain trend.

Finally, students program a population bottleneck, which kills all of the agents in a certain area of SpaceLand. This shrinks the population and therefore exacerbates the effects of random events in the resulting gene pool. The concept emphasizes the unique characteristics of small populations that first discussed in Unit 4.

Technical Details:

The starter code contains the same rabbit and carrot ecosystem from Unit 3. However, the rabbits in this version have an added feature: a color gene. This gene ranges from 1-100 and dictates the shade of blue that the rabbit will appear. To link the color gene to speed, students change the stepsize of the rabbits to a fraction of the color gene. To link the color gene to metabolism, students change the energy lost to a fraction of the color gene. Additionally, the "flood" button simply wipes out the populations on the left side of Spaceland to create a population bottleneck.

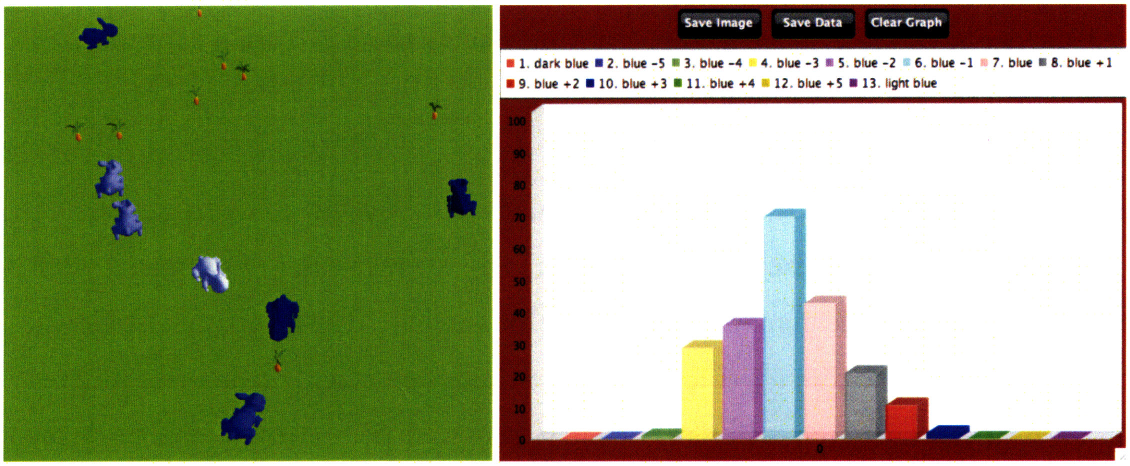


Figure 16: Rabbits in the color gene simulation before linking the gene to speed and metabolism. The graph indicates the relative frequencies of each allele.

Unit 10: Evolution (Mutations and "Fitness")

Goals:

This final lesson brings together the ideas from the previous few units. Genetic mutations cause differential protein expressions (Unit 7), which then cause differences in organisms if occurring at an early developmental state (Unit 8), which then cause variation in the gene pool that leads to changes in allele frequencies due to natural selection and genetic drift (Unit 9). The goal of the lesson is for students to understand that genetic differences are actually important in maintaining populations, particularly in the face of environmental changes, and that the concept of "fitness" can actually only be defined in the context of a particular environment.

Lesson Plan:

The lesson begins with a discussion about whether mutations are good or bad. In all of the previous units, mutations have led to defects and disease and did not give individuals any particular advantage. Students are directed to recall the color gene from the previous lesson and note that the genetic diversity among the rabbits in the simulation must have initially arisen from mutations.

In the project, the rabbits are faced with an extra variable: temperature. Temperature ties the unit back to Unit 1, which considered the importance of abiotic factors in shaping an environment. In this particular system, however, the temperature controls rabbit metabolism. From the previous unit, rabbits that run faster also lose energy quicker. At low temperatures in this project, this effect is exponentiated.

Students then experiment with the system to see how the rabbit population responds to different temperatures and mutation rates. All of the rabbits in the system begin with a single color gene. If the temperature is brought to 0 with no mutation, all of the rabbits will die because they are not equipped to live in the low temperature. At high temperatures, the rabbits survive. If students increase the mutation rate, they will notice that the average value of the color gene increases slightly at higher temperatures and decreases greatly at lower temperatures. Additionally, if students vary the mutation rate first to create variation and then set it back to zero, there may be enough initial variation for the rabbit population to survive the change in temperature.

During the discussion period, students should present their findings and see if they can predict the actual model behavior (e.g. how the rabbit growth was affected by temperature). While they may not figure it out, they may notice that the relationship between energy loss and color gene is not linear with temperature. Additionally, they should discuss the definition of evolutionary "fitness" and note that the mutations were important in cushioning the population in the case of a change in the environment.

Technical Details:

The code contains the same rabbit and carrot ecosystem from Unit 9. However, there is an added element in this model of temperature. The temperature does not affect the carrots (unlike in Unit 1), but it does affect rabbit metabolism. At high temperatures the energy loss has a relatively linear dependence on color gene value (with low values having low energy loss and high values having high energy loss), but at low temperature the energy loss is more exponential. In this way, the difference in energy loss between gene values is greatly amplified at low temperatures (Figure 17). The specific parameters are such that rabbits with color gene 50 survive at temperature 50 degrees but die out at temperature 0.

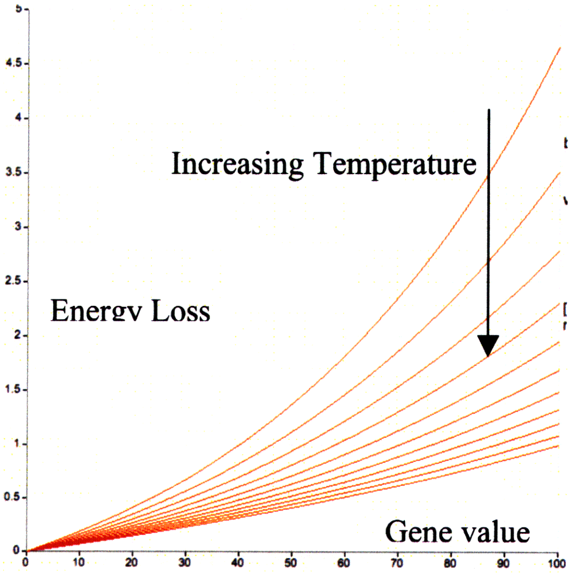


Figure 17: The difference between rabbit energy loss for different gene values is greatly enhanced at low temperatures.

5. Results and Future Work

5.1. Teacher Evaluations

The ecology and evolution portion of the biology curriculum was presented to five teachers from various Massachusetts schools and teaching centers who came to MIT as participants in the one-week StarLogo summer Imagination Toolbox teacher workshop. The teachers participated in various starter activities to learn the basics of StarLogo, then spent two days working specifically on the biology track. After the workshop, the teachers wrote plans for implementing TNG in their classrooms.

All of the teachers expressed interest in implementing at least parts of the curriculum, with some expressing desire to adapt the projects and make their own simulations as well. Unfortunately, the timeline of this thesis project did not coincide well with the course schedules of the teachers. One teacher presented TNG to other teachers at his school and said they were "anxious to implement a couple of the units this spring." Of the biology topics, the evolution units sparked a lot of interest. Another teacher will begin implementation of the curriculum in February 2009.

Hal Scheintaub, our resident StarLogo teacher, implemented a modified rabbit and carrot model that included grass in his biology class. After the activity, he asked the students to blog about how the rhythmic cycle of rabbit and carrot populations arose. Student responses showed an understanding for predator-prey relationships, with many student responses along the same lines as the following: "The rhythm is controlled by both the bunnies and the carrots. As the bunnies eat the carrots the amount of carrots go down, but as the amount of carrots go down, so do the amount of bunnies because there are no more carrots for the bunnies to eat." However, students did note that the cycles were also controlled in part by the programmer, who determined the rates of death and reproduction. This answer displays a certain sensitivity to the nuanced differences between models and real life.

In addition, other users have downloaded the projects from the StarLogo TNG website and worked through the projects themselves. One contacted me via e-mail with minor bugs in the projects, which were subsequently fixed. The user commented that he particularly enjoyed the predator-prey tutorial, which he considered a good idea and "very well written."

5.2. Limitations and Future Work

For many teachers, such as Dr. Scheintaub and those attending the summer workshop, the main goal is to get TNG in the classroom over the course of the entire year. In this way, students really learn to use the tool and can therefore build more complicated models on their own. However, the demands of state standards and school restrictions make this difficult for many teachers. Additionally, teachers require a significant ramp-up time to become comfortable enough with the software themselves before teaching it to students. It is for this reason that the curriculum presented comes packaged for delivery. Teachers with more flexibility can use the concepts and techniques presented here, which I believe will promote deeper student learning in the various topics in biology, as a springboard for their own ideas and activities.

The limitation of this packaged approach is that teachers must walk a fine line between too much information and too little information. A complicated pre-built model can provide a great visual but may seem to students like a black box. Building a model from scratch may give students a feeling of ownership over their programs but may also frustrate them if they don't know how to implement what they want. Straying from the line can cause students to become bored and lose focus. Teachers who work with StarLogo all year have an advantage because students have a better grasp on how to implement what they want.

Though I have attempted to give students some flexibility while dealing with more complicated models, future work would need to be done to test how well the balance is maintained. This would require facilitated in-class workshops in which the entire curriculum was implemented as written. Facilitators would observe student interactions to see where things do not go according to plan. Additionally, pre- and post- tests would have to be administered to determine to what degree the student misconceptions outlined in this thesis were corrected.

Additional work would have to be done on the technical side of some of these models after revision to the actual StarLogo TNG software. A major redesign of variables and lists would eliminate the need for some specialized procedures that clutter the projects. The ability to organize code pages to hide back-end procedures and highlight important ones would help keep large projects manageable and serve as an additional communication tool between teachers and students. These features have been logged and will be revisited in later versions of TNG.

6. Conclusion

The growing field of bioinformatics demands that students entering biology research be familiar with systems concepts, be comfortable using computers to build and testing models, and be able to approach problems creatively. Current high school biology teaching largely ignores these needs, instead focusing on teaching isolated ideas, without the use of computers, and in a classical teacher-student lecture format. This approach has particular effects on the teaching of ecology, evolution, and the DNA-protein relationship, which are all topics that require an understanding of relationships between many different agents operating at many different scales.

This thesis outlined a 10-unit biology curriculum implemented in StarLogo TNG that covers these topics, focusing on helping students make the connections between nucleotide base pairs and diversity in populations. By making analogies between proteins and animals, the curriculum emphasizes proteins as the workers behind all of the molecular processes governing life. By traversing different scales, from molecules to cells to organisms to populations, students can see how changes in one level can have drastic effects on another. Using TNG, students can approach their learning with an open mind and test ideas creatively. Tying ecology, evolution, and the DNA-protein relationships together, the students can see how the topics they learn in biology class are all connected to give rise to the complexity of life on earth.

References

Allen, D., and Tanner, K. (2002). Approaches to cell biology teaching: questions about questions. *Cell Biology Education.* 1, 63-67.

Anderson, D. L., Fisher, K. M., and Norman, G. J. (2002). Development and evaluation of the conceptual inventory of natural selection. *Journal of Research in Science Teaching.* 39, 952-978.

Biermann, C. A. (1988). The protein a cell built (and the house that Jack built). *The American Biology Teacher.* 50(3), 162-163.

Bishop, B. A., and Anderson, C. W. (1990). Student conceptions of natural selection and its role in evolution. *Journal of Research in Science Teaching.* 27(5), 415-427.

Bixler, A. (2007). Teaching evolution with the aid of science fiction. *The American Biology Teacher.* 69(6), 337-340.

Bloom, B. S., Englehart, M. D., Furst, E. J., Hill, W. H., and Krathwohl, D. R. (1956). *A Taxonomy of Educational Objectives: Handbook 1, Cognitive Domain.* McKay: New York.

Bolinsky, D. (2007). "Fantastic voyage inside a cell." Technology, Entertainment, Design. Posted July 2007. Accessed Jan 30, 2009. http://www.ted.com/index.php/talks/david_bolinsky_animates_a_cell.html.

Cloud-Hansen, K. A., Kuehner, J. N., Tong, L., Miller, S., and Handelsman, J. (2008). Money, sex, and drugs: A case study to teach the genetics of antibiotic resistance. *CBE–Life Sciences Education.* 7, 302-309.

Douglas, K. R. (2008). A kinesthetic model demonstrating molecular interactions involved in anterior-posterior pattern formation in Drosophila. *CBE–Life Sciences Education.* 7, 74-81.

Ferrari, M., and Chi, M. T. H. (1998). The nature of naïve explanations of natural selection *The International Journal of Science Education.* 20(10), 1231-1256.

Garbarino-Asmus, E. (2007). Protein structure. *The American Biology Teacher.* 69(1), 38-40.

Garvin-Doxas and Klymkowsky (2008). Understanding randomness and its impact on student learning: Lessons learned from building the biology concept inventory (BCI). *CBE–Life Sciences Education.* 7, 227-233.

Halford, S. E., and Marko, J. F. (2004). How do site-specific DNA-binding proteins find their targets? *Nucleic Acids Research.* 32(10), 3040-3052.

Klopfer, E., Scheintaub, H., Huang, W., Wendel, D., Roque, R. (2009). The simulation cycle – Combining games, simulations, engineering and science using StarLogo TNG. In press for the Journal of E-Learning.

Klopfer, E., Colella, V., Resnick, M. (2002). New paths on a StarLogo adventure. *Computers & Graphics.* 26, 615-622.

Lewis, J., Leach, J., and Wood-Robinson, C. (2000a). All in the genes? Young people's understanding of the nature of genes. *Journal of Biological Education.* 34, 74-79.

Lewis, J., Leach, J., and Wood-Robinson, C. (2000b). What's in a cell? Young people's understanding of the genetic relationship between cells within an individual. *Journal of Biological Education.* 34, 129-132.

Marbach-Ad, G., and Stavy, R. (2000). Students' cellular and molecular explanations of genetic phenomena. *Journal of Biological Education.* 34, 200-205.

Mulvihill, C. (1996). Making the chromosome-gene-protein connection. *The American Biology Teacher.* 58(6), 364-366.

Offner, S. (1992). Teaching biology around themes: Teach proteins & DNA together. *The American Biology Teacher.* 54(2), 93-101.

Orcajo, T. I., and Aznar, M. M. (2005). Solving problems in genetics II: conceptual restructuring. *The International Journal of Science Education.* 27, 1495-1519.

Perry, J., Meir, E., Herron, J. C., Maruca, S., and Stal, D. (2008). Evaluating two approaches to helping college students understand evolutionary trees through diagramming tasks. *CBE–Life Sciences Education.* 7, 193-201.

Resnick, M. (1999). Decentralized modeling and decentralized thinking. In *Modeling and Simulation in Precollege Science and Mathematics*, edited by W. Feurzeig and N. Roberts. Springer: New York. 114-137.

Resnick, M (2002). Rethinking learning in the digital age. In *The Global Information Technology Report: Readiness for the Networked World,* edited by G. Kirkman. Oxford University Press: Oxford.

Roberg, E. (2004). Genetic building blocks. *The Science Teacher.* 71(6), 24.

Saka, A., Cerrah, L., Akdeniz, A. R., Ayas, A. (2006). *Journal of Science Education and Technology.* 15(2), 192-202.

Spirin, A. S. (2004). The ribosome as an RNA-based molecular machine. *RNA Biology.* 1(1), 3-9.

Sprehn, J. L. (1993). Protein building blocks. *The Science Teacher.* 60(7), 22.

Stencel, J., and Barkoff, A. (1993). Protein synthesis: role playing in the classroom. *The American Biology Teacher.* 55(2), 102-103.

Stern, L., and Ben-Akiva, I. (2007). Tails' inheritance: challenging secondary school students' ideas about the inheritance of acquired traits. *The American Biology Teacher.* 69(2), 76-79.

Templin, M. A., and Fetters, M. K. (2002). A working model of protein synthesis using Lego^{TM} building blocks. *The American Biology Teacher.* 64(9), 673-678.

Tsui, C. Y., and Treagust, D. (2004). Conceptual change in learning genetics: an ontological perspective. *Research in Science and Technological Education.* 22, 185-202.

Venville, G. J., and Treagust, D. F. (1998). Exploring conceptual change in genetics using a multidimensional interpretive framework. *Journal of Research in Science Teaching.* 35, 1031-1055.

Wang, K., McCaffrey, C., Wendel, D., Klopfer, E. (2006). 3D game design with programming blocks in StarLogo TNG. *Proceedings of the 7th International Conference on Learning Sciences.* 1008-1009.

Westerling, K. E. (2008). Using playing cards to simulate a molecular clock. *The American Biology Teacher.* 70(8). 37-42.

Wood, W. B. (2008). Teaching concepts versus facts in developmental biology. *CBE–Life Sciences Education.* 7, 10-16.