

Multimedia in Biochemistry and Molecular Biology Education

Interactive Computer Simulations of Genetics, Biochemistry, and Molecular Biology*

Received for publication, August 1, 2007, and in revised form, September 28, 2007

Brian T. White†§ and Ethan D. Bolker‡

From the †Department of Biology, University of Massachusetts Boston, Boston, MA 02125; ‡Department of Computer Science, University of Massachusetts Boston, Boston, MA 02125

This article describes three interactive software simulations targeted to key domains of modern molecular genetics: genetics, biochemistry, and molecular biology. These simulations allow students to deepen their understanding of key principles in these domains by setting up crosses, designing proteins, and designing genes; the simulations then apply these principles to produce results that the students can interpret. Using this software, students can confront their misconceptions and connect these disciplines in a way that is difficult if not impossible without an interactive environment. We present lab exercises that guide the students as they use the software in a series of activities that begin by introducing the tools and build towards more open inquiry. Preliminary evaluation shows that students enjoy the software and that it promotes active engagement and application of the material. The open-source Java software and the relevant lab manuals are available free of charge on line.

Keywords: Computer simulation, genetics, protein folding, biochemistry, molecular biology.

Genetics, biochemistry, and molecular biology are the core domains of modern molecular genetics and are core parts of biology courses from the high school AP level through post-graduate studies. Gaining a deep understanding of these domains requires the ability to develop and test hypotheses. Although it is possible to develop hypotheses using static presentations, animations, pencil-and-paper exercises, or visualizations, testing hypotheses requires experimental work—either real or simulated. This article describes three interactive computer simulations that allow students to explore these areas of modern biology; they should be useful for high school through intermediate-level undergraduate students. The software as well as sample lab manual sections can be downloaded free of charge from the sites listed at the end of this article.

Previous studies have shown that interactive simulations allow students to employ a more active form of learning than is possible with other teaching techniques since they allow students to ask their own questions and follow their own paths, dictated by their individual learning styles, strengths, and weaknesses [1]. Simulations in chemistry have been shown to help students to confront and eliminate misconceptions [2]. In many cases, simulations are used in Discovery Learning, where students discover for themselves the underlying principles in a partic-

ular domain by conducting simulated experiments. This has been shown to be effective in many cases, especially when it is carefully scaffolded [3, 4]. In the lab exercises described in this article, students use simulation to solve problems, generate results that must be explained, and develop and test hypotheses. These exercises reinforce the basic understanding of transmission genetics, protein structure, and gene expression that students have gained from preceding lectures, hands-on model building, and simpler pencil-and-paper problems. Each simulation acts as a capstone for the corresponding section of the course so that students can take what they have learned to a higher level. The simulations allow the students to explore: to test their understanding by predicting results; to apply multiple approaches to different problems; and to engage with their own ideas, questions, and misconceptions. In this way, these exercises take advantage of many of the features of Discovery Learning—students learning at their own pace and working from their own questions and previous knowledge.

In a previous article, we described the Virtual Genetics Lab (VGL),¹ a simulation of transmission genetics [5]. VGL is modeled on the Genetics Construction Kit (GCK) [6]. GCK has been studied in detail in a variety of educational contexts; researchers have found that students must apply their knowledge of genetics in depth when

*This work was supported in part by NSF grant 9984612.

§To whom correspondence should be addressed. 100 Morrissey Blvd., Boston, MA 02125. Tel.: +617-287-6630. E-mail: brian.white@umb.edu

¹The abbreviations used are: VGL, Virtual Genetics Lab; GCK, Genetics Construction Kit; GX, Gene Explorer; PI, Protein Investigator; MGX, Molecular Genetics Explorer; TA, Teaching Assistant.

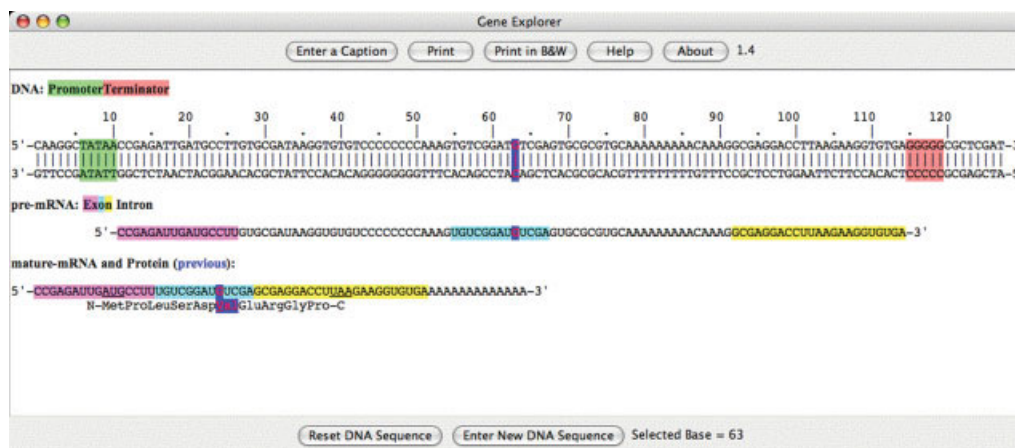


Fig. 1. **The Gene Explorer.** Base 63 in the DNA is highlighted in blue; GX highlights the corresponding bases in the pre-mRNA, mRNA, and the corresponding amino acid in the protein.

working with the software [7, 8]. This article describes three new software applications: the Gene Explorer (GX), which simulates eukaryotic gene expression; the Protein Investigator (PI), which simulates the process of protein folding; and the Molecular Genetics Explorer (MGX), which combines VGL, GX, and PI to connect genetics, protein folding, and gene expression in a unified exploration of a single biological phenomenon. In each application, students develop hypotheses and test them by setting up crosses (VGL and MGX), designing proteins (PI and MGX), or designing genes (GX and MGX).

Informal, anecdotal classroom evaluation of this software has been positive. It shows that the students actively engage in the activities and apply their knowledge thoughtfully. Several of the applications are being used at other institutions. We are in the process of conducting formal evaluations of some of this software at University of Massachusetts Boston and elsewhere.

SIMULATION 1: THE GENE EXPLORER

The GX is an application where students can explore, edit, or create DNA sequences, which are then transcribed, spliced, and translated based on a simplified model of eukaryotic gene expression. When a base in the DNA is selected, any corresponding base in the pre-mRNA and mature mRNA is automatically highlighted along with any corresponding amino acid in the protein. Figure 1 shows the results of selecting base 63 in the DNA. When the student edits the DNA sequence, GX automatically updates the mRNA and protein as appropriate. Students can also enter a DNA sequence of their own design and observe how it is expressed. Finally, the displayed gene can be printed.

GX simulates eukaryotic gene expression in a series of steps that model the process in living cells. The simulation begins with transcription: it searches for a promoter sequence (5'TATAA3') and a terminator sequence (5'GGGGG3') in the DNA; if both are found, it creates a pre-mRNA starting at the promoter and ending at the terminator. Next, GX splices the mRNA: it searches for introns (5'-GUGCGCAAAG-3') and removes them.

It then adds a 3' poly-A tail. Finally, GX translates the mature mRNA by searching for the 5'-most AUG and continuing until it reaches the first in-frame stop codon or the 3' end of the mRNA. The most important differences between gene expression in GX and the actual expression of eukaryotic genes are as follows: actual gene sequences are much longer than 100 nucleotides, recognition sequences are usually longer and allow for some mismatches; poly-A tails are generally longer than 13 nucleotides; and there are sequences surrounding the AUG that are also necessary for initiation of translation. Nevertheless, GX does embody the most pedagogically relevant features of eukaryotic gene expression. It is also possible to configure GX as a web-page that simulates either prokaryotic or eukaryotic gene expression (see under "How to Obtain and Use the Software").

GX fills a gap between pencil-and-paper gene expression problems and professional gene analysis tools. Students learn the details of transcription, translation, and mutation by working through simple cases by hand. However, pencil-and-paper techniques quickly become tedious and error-prone for longer genes, mRNA splicing, and complex mutations. On the other hand, professional tools often take shortcuts—translating DNA sequences, for example—that, while simplifying analysis for professionals, can lead to confusion when used by beginning students.

The GX lab exercises follow lectures on gene expression and a lab where students use LEGO models to explore DNA structure, DNA replication, transcription, and translation, followed by some pencil-and-paper transcription, translation, and mutation problems. Students begin the 3-hr GX lab session by exploring a simple eukaryotic gene. They find parts of the DNA that do not correspond to bases in the pre-mRNA (extragenic regions), the mature mRNA (introns), or amino acids in the protein (5' and 3' UTRs). They then construct a map of the given gene showing the promoter, terminator, introns, exons, start codon, and stop codon. Next, they make particular mutations (missense, silent, nonsense, splicing, etc.) and explain their effects. Finally, they are asked to devise mutations with specific effects: a muta-

- (1) **Student 2:** We don't want to mess around with the stop just {after} the start one
- (2) **Student 1:** Yah there is an AUG there so this is a potential start so if we delete this would slide over and start here {meaning that, if they deleted the start, translation would then just start at the next AUG in the mRNA}.
- (3) **Student 2:** But we want it to not start at all
- (4) **Student 3:** We want no protein you know what I mean
- (5) **Student 1:** Yah
- (6) **Student 2:**What else can we do?
- (7) **Student 3:** I dunno...the only I can think of is move the stop closer to the start
- (8) **Student 2:** Alright so try that
- (9) **Student 1:** What?
- (10) **Student 2:** Go after the AUG... ok what is the first letter.. we have a CCU...we can't do anything to stop it
- (11) **Student 3:** Doesn't it seem the most logical I dunno; I don't like this
- (12) **Student 2:** How about you do something in the promoter
- (13) **Student 3:** I dunno if you can
- (14) **Student 2:** Just try itoh YAH
- (15) **Student 3:** Wow so we changed something in the TATA box
- (16) **Student 2:** Yah the 1st A in the TATA Box
- (17) **Student 1:** Number 7
- (18) **Student 3:** So we changed it to a TCTAA
- (19) **Student 1:** Went from AT to a CG...draw the resulting gene
- (20) **Student 2:** So you prevented it from starting

FIG. 2. Excerpt of transcript of students using the Gene Explorer.

tion that changes the mRNA but not the protein; and one that results in no mRNA or protein being made. Lastly, they must find the mutation that leads to a particular mutant protein sequence. For the lab report, students must design and test a gene of their own that contains one intron and produces a protein of at least five amino acids.

We have used GX with success for 4 years. Anecdotal reports from students and TAs are positive and students are able to complete the lab report with a high rate of success. For example, in the fall of 2006, of the 281 students in the class, 244 (87%) attended the GX lab; of those, 237 (97%) turned in a GX lab report; the average lab report score for those attending the lab was 19.27 out of 20 points (15 points for a gene that produces a 5 amino acid protein; 5 points for including an intron).

The section of classroom video transcript shown in Figure 2 is an example of the kind of experience that GX facilitates. The students were trying to make a single-base-pair mutation that would completely prevent the formation of mRNA and protein. Line 1 refers to their previous attempt when they put a stop codon immediately after the start codon. The GX showed them that this would still result in the production of both mRNA and protein leading them to discover that this was not a viable strategy. In Line 2, they realize that deleting the start codon would only lead to translation initiation at a downstream site. Starting with line 3, they move towards a correct answer. Note especially Lines 12–14, where the students decide to try a promoter mutation even though they are unsure if it is possible. They immediately recognize that they have succeeded (Line 14) and correctly explain why (Line 20). Although promoter mutations had been described previously in lecture, GX helped the students to “rediscover” them at a point when they were ready to absorb the information.

SIMULATION 2: THE PROTEIN INVESTIGATOR

The PI is a simulation of protein folding. Students enter a short sequence of amino acids and the program calculates the folded shape of this polypeptide. Amino acids are modeled as planar disks with a fixed common radius. A protein folds itself by arranging its amino acids on a two-dimensional hexagonal grid. The specific shapes of the amino acids' side chains are ignored, but their hypothetical interactions due to ionic, hydrogen, and disulfide bonding and the hydrophobic effect contribute to the energy of the molecule when amino acids occupy adjacent hexagonal cells or are exposed to the surrounding water. The PI is shown in Figure 3. The Amino Acid table in the upper left shows the 20 amino acids and their abbreviations as they would appear in a folded protein. Each disk shows the properties of the side chain of the corresponding amino acid: the darker the shading, the more hydrophobic the side chain. Anionic, cationic, and polar uncharged side chains are indicated by “-,” “+,” and “*” symbols respectively. There are two Folding Windows, each capable of folding and displaying a folded protein; sample folded proteins are shown in each.

To fold a novel protein, the student types or edits the desired protein sequence in the Amino Acid Sequence window using the single-letter amino acid code; he/she then chooses whether the protein will be folded in oxidizing (“Disulfide Bonds ON”) or reducing (“Disulfide Bonds OFF”) conditions and clicks “Fold.” PI then folds the protein, trying many different conformations to find the one with the lowest energy. For each conformation, the energy is calculated as a weighted sum of the hydrophobic index of each amino acid multiplied by the number of edges it has exposed to the solvent, the number of possible ionic bonds, the number of possible hydrogen bonds, and the number of possible disulfide bonds. An ionic, hydrogen, or disulfide bond is considered “pos-

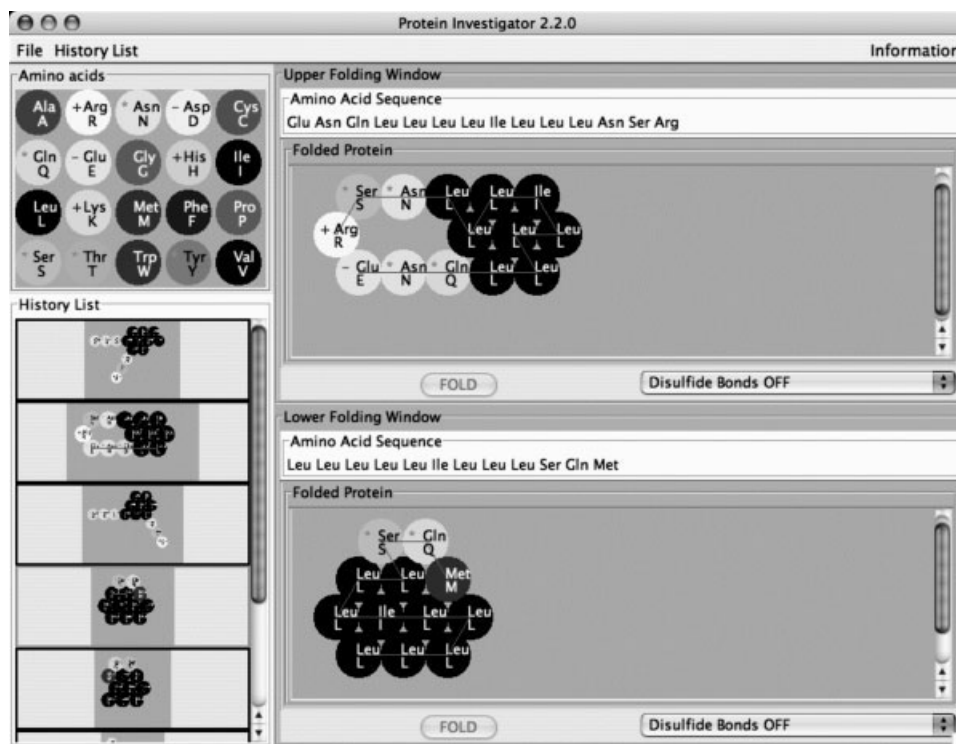


FIG. 3. **The Protein Investigator.** This screenshot shows two folded proteins and the History List.

sible” if two appropriate amino acids are in contact in the two-dimensional folded structure. In the overall energy calculation, each bond type is weighted by its relative strength: disulfide bonds, if enabled, are strongest, followed by ionic bonds, and then hydrogen bonds; the weakest are hydrophobic interactions. The backbone is indicated by a magenta trace; disulfide bonds, if present, are shown as yellow lines. Once folded, the protein is shown in the Folding Window and an entry is added to the History List. Proteins can be moved from the History List to either Folding Window for comparison with related proteins. The History List can be saved between sessions, printed, or exported as a web page of sequences and structures for printing or incorporation into a lab report.

VGL and GX implement recognizable (although simplified) models of the biological phenomena they are designed to teach. The PI takes greater liberties with protein biochemistry in order to meet our pedagogical goals. The important lessons the student must learn are that the biological activity of a protein depends directly on the shape of the molecule, that the shape is determined by the sequence of amino acids when it folds itself in space to minimize its energy, and that a small change in the amino acid sequence can sometimes cause a large change in the folded shape. Ultimately, the behavior of our simulated proteins must be explicable in terms of the properties of the amino acids involved.

Software that attempts to predict the three-dimensional shape of a polypeptide given the amino acid sequence does exist, but the algorithms are computationally intensive. It takes a super-computer several hours to predict the fully folded shape of even a small protein. Even if this were practical in a 3-hr teaching lab, a student would

be hard put to compare two complex three-dimensional protein molecules to observe the effects of a change to their amino acid sequence. PI avoids this difficulty by implementing our simplified model of protein folding.

Even with these simplifications, folding a long sequence by exploring all possible configurations to find the one with minimum energy would take too long, since there are about 4.25^n planar configurations for a chain of length n . To reduce folding time, PI employs an incremental algorithm, at each step tentatively placing the next eight acids in the chain so as to minimize the energy of the folded molecule up to that point. Then it makes the placement of the first four permanent, and proceeds to the next step.

The PI lab exercises follow lectures on the three-dimensional structure of proteins and are followed by a lab exploring the three-dimensional structure of the enzyme lysozyme [9]. At the start of the PI lab, students learn to use the software by working with some simple protein sequences. Students are asked to predict their folded shapes, check their predictions against the results computed by PI, and to explain any discrepancies. Students then go on to make a simple protein and predict as well as observe the effects of specified amino acid substitutions. Next, students are asked to use mutations to demonstrate the role of particular noncovalent bonds in a protein they have designed. Finally, they are asked to design proteins with specified shapes of increasing complexity. The activities fit into a 3-hr lab session.

We have used the PI for two semesters in General Biology I (Bio 111), the first semester majors’ core biology course at University of Massachusetts Boston, with positive results. Students like using the program and

(1) **Student 2:** Um, okay, now we need the circling arms.
 (2) **Student 1:** Um hmm.
 (3) **Student 2:** Um, so we need to put, um, a positively charged one on one end, and a negatively charged one on the other end. So let's put a charged one on this end.

(4) **Student 1:** Which one...
 (5) **Student 2:** Okay, that's R, right? Yeah.
 (6) **Student 1:** And one more.
 (7) **Student 2:** And on the other end we should put a negatively charged one.
 (8) **Student 1:** This one?
 (9) **Student 2:** Which one?
 (10) **Student 1:** This one? I don't know.
 (11) **Student 2:** I don't know if the polar ones will work. Let's try it. T. No, it's not T, it's Y. Okay. No, that didn't work.
 (12) **Student 1:** Maybe... which one is that... H? Sorry.
 (13) **Student 2:** Oh, I think histidine and arginine have the same charge.
 (14) **Student 1:** Yeah.
 (15) **Student 2:** If they have a different charge, then they'll attract each other. Let's um. Try this one. Lysine. Cause it has a positive charge. That's, uh, K. K, for some reason. (pause) Oh, come on!
 (16) **Student 2:** Maybe our chains are too long. Wait no, it still says they have the same charge. Wait, lysine still has a positive charge. That explains it. Arginine's positive, lysine's positive, histidine's positive. We should try one of the ones that's not positive (laughs).
 (17) **Student 1:** (laughs) (pause) Which one...
 (18) **Student 2:** I dunno. That's um...
 (19) **Student 1:** Same thing.
 (20) **Student 2:** Not charged. Glutamic acid is...
 (21) **Student 1:** Oh, I got the wrong one.
 (22) **Student 2:** E.
 (23) **Student 2:** That's huh not charged.. glutamic acid is
 (24) **Student 1:** Oh I got the wrong
 (25) **Student 2:** Ya.....ah "e"
 (26) **Student 2:** Alright.. lets shorten this up.
 (27) **Student 2:** Ahhhhh!

Target Shape: Starting Protein:



RSNQLLLILLSQN

Their Protein:



RSNQLLLILLSQNY

Their Protein:



RSNQLLLILLSQNK

Their Final Protein:



RSNQLLLIQNE

FIG. 4. Excerpt of transcript of students using the Protein Investigator.

often download it for home use. In the fall of 2006, the PI lab followed a lab using molecular visualization, which has been shown to be effective at teaching protein structure that explores the three-dimensional structure of lysozyme [9]. We asked the students which labs should be included in the syllabus for the next year; out of 281 students enrolled, 241 (86%) responded. Of these responses, 5% chose Visualization only, 32% chose PI only, 29% chose Visualization followed by PI (as they had experienced it), 32% chose PI followed by Visualization, and 2% chose none. These results show that, while both approaches were appreciated, students preferred PI (93% selected an option involving PI) over the Visualization (66% selected an option involving Visualization). We are conducting a more formal evaluation of the learning outcomes of these two approaches. Our study will also look for specific misconceptions that may be generated by the PI lab.

The transcript of a videotaped lab session shown in Figure 4 provides an example of the kind of thought process that the PI encourages. The structures the students refer to are shown opposite the corresponding text. The transcript shows the students using their knowledge of how amino acid side chains interact as they construct a protein with "circling arms" (Line 1). Note that they make several mistakes (Lines 11 and 15), but are able to use

the simulation and their understanding of protein folding to correct those mistakes and arrive at the desired shape (Line 15).

We have recently added a "Game Mode" to the PI. In this mode, students design their own proteins to match one of a series of pre-designed "target shapes"; the PI then determines if the student's guess matches the shape of the target. This allows students to practice protein engineering with feedback from the software.

SIMULATION 3: THE MOLECULAR GENETICS EXPLORER

The MGX combines VGL, PI, and GX in a common framework that allows students to explore a single biological phenomenon from the perspectives of genetics, biochemistry, and molecular biology, providing a connection between topics that students often experience as separate units. It is intended to be used several times during a semester-long course to provide a common thread linking these perspectives. It can also serve as a capstone for each of these course sections, providing an advanced application of what students have learned in a hypothesis-testing situation.

The MGX exercises in Bio 111 begin with a biological phenomenon, a hypothetical diploid self-fertile hermaphroditic flowering plant that produces flowers of several

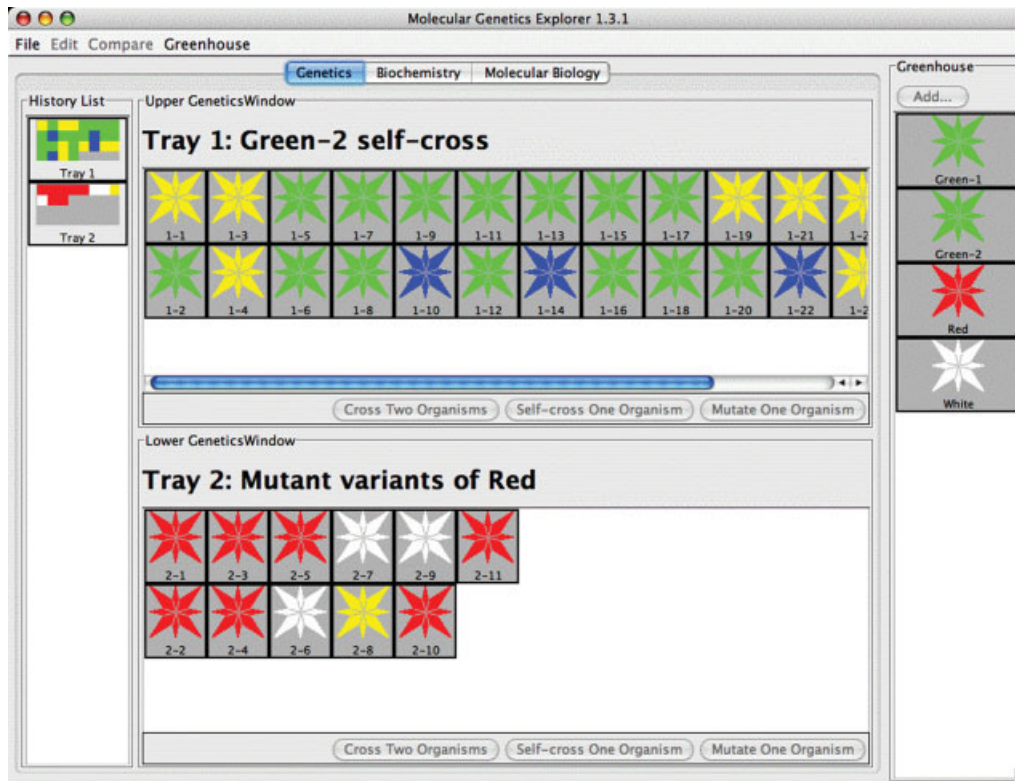


FIG. 5. **The Molecular Genetics Explorer.** This screenshot shows the Genetics Workbench with the results of a self-cross in the upper window; the lower window shows the results of one round of mutant generation.

different colors. The opening MGX screen, shown in Figure 5, shows a starting set of plants in the Greenhouse; these represent a “field population” of plants that have been chosen to provide a suitable problem space. In the narrative that accompanies these exercises, the students work for a flower-breeding company that wants to produce a pure-breeding purple flower of this species. The field samples do not contain any purple flowers. The students must use MGX to study the genetics, biochemistry, and molecular biology of flower color in these plants in order to build the pure-breeding purple flowers. Along the way, they deepen their understanding of these three disciplines and the connections between them. Figure 6 shows the kinds of questions students can explore with MGX; it is based on Botstein’s Triangle [10]. The boxes represent the three disciplines and a representation of two alleles using the symbols of that discipline. The questions originate from a particular discipline and connect to another. For example, if a student was thinking in genetics terms, he/she might ask, “Why is A dominant?”; this question can only be answered in biochemical terms. The dashed lines indicate questions that are typically addressed in general biology courses; the solid lines indicate the additional questions that MGX allows students to explore.

MGX is structured to facilitate this interconnected investigation. Flowers from the Greenhouse can be analyzed in any of three Workbenches: the Genetics Workbench, the Biochemistry Workbench, or the Molecular Biology Workbench. In the Genetics Workbench, based on VGL, flowers can be self-crossed, out-crossed, or

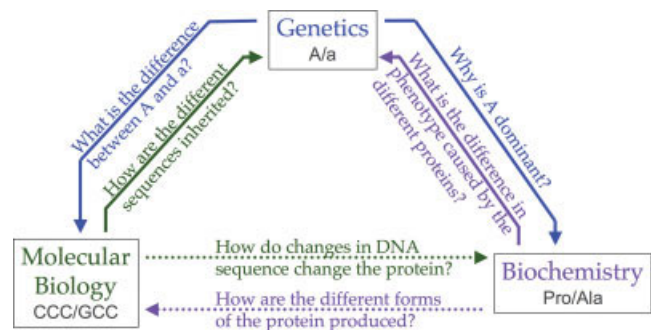


FIG. 6. **Connections between disciplines available via the MGX.** The three main disciplines of molecular biology are shown above linked by questions that illustrate the connections between them. All of these questions can be explored with MGX; dotted lines indicate questions commonly addressed in general biology courses.

subjected to random mutagenesis; the Genetics Workbench is shown in Figure 5. The program simulates crossing by randomly choosing from the parent’s alleles and determining the resulting color when constructing the offspring. In the Biochemistry Workbench, based on PI, pigment proteins present in flowers can be viewed, edited, and their colors predicted. In the Molecular Biology Workbench, based on GX, genes encoding these pigment proteins can be examined and edited; the engineered genes can then be incorporated into new flowers that can be sent to the Greenhouse for further study in other Workbenches.

Students begin by using the Genetics Workbench to identify the color alleles present in the field samples. They then cross these samples and their offspring to determine how the alleles interact to give an overall flower color. For example, they can discover that white color is recessive to the other colors. Although they can describe the interactions between alleles, the Genetics Workbench cannot help them explain, for example, why the white allele is recessive or how the white allele differs from the colored alleles. Students can create a purple flower by combining red and blue in the Genetics Workshop, but the resulting heterozygote is not pure-breeding. Although their understanding of the genetics of color in these flowers is complete, they do not yet know enough to achieve their goal.

A few weeks later in the semester, in the Biochemistry section of the course, the students revisit MGX after they have used PI. This time, they look at the pigment proteins present in each of the Greenhouse strains. With the Biochemistry Workbench, they can determine the color of each protein present and explain, in biochemical terms, why the alleles interact as they did in the Genetics lab. They find that there are two different white alleles: one that makes a colorless protein and one that makes no protein at all. This begins to explain why the white phenotype is recessive to the other colors: no pigment is present in white strains. By looking at the structures of the colored and uncolored proteins and by designing proteins of their own, students can develop and test hypotheses about the structural features that govern color formation. With the MGX "Compare" function that aligns protein sequences, students can find the differences in amino acid sequences that result in the observed protein colors. They infer that proteins with a hydrophobic core of seven amino acids surrounded by six hydrophilic amino acids will be colored; the particular aromatic amino acids present in the hydrophobic core determine the particular color. Using this information, they can design a purple protein and use the Biochemistry Workbench to verify that their understanding of the biochemistry of color in these flowers is correct. As before, their understanding of the genetics and biochemistry involved is complete, but they lack an understanding of the relevant molecular biology to construct a pure-breeding purple flower.

Finally, after they have used GX in the Molecular Biology section of the class, they use the Molecular Biology Workbench to explore the DNA of the Greenhouse strains. First, they find the differences in DNA sequence among the different alleles that they found in the Genetics Lab. They then observe how these differences lead to the different proteins they observed in the Biochemistry Lab. They can then explain how the two white alleles can have the same color although they have different DNA sequences. They next design and test a gene that expresses the purple protein sequence they built in the Biochemistry Lab. They then create a flower that is homozygous for this purple protein, save it to the Greenhouse, and use the Genetics Workshop to show that it is pure-breeding. In the lab report, students must first explain their design process—how color is determined in

these flowers and how they used this information to build a pure-breeding purple flower. They are then asked to connect the three parts of the course by explaining the difference between gene and allele in terms of genetics, biochemistry, and molecular biology.

Although this description of the MGX exercises begins with genetics and ends with molecular biology, it is likely that the workbenches in MGX could be used productively in a different order. For example, students could begin by analyzing the pigment proteins and their color combining properties in the Biochemistry Workbench. They could then explore how these are inherited in the Genetics Workbench and finish by constructing a pure-breeding purple flower in the Molecular Biology Workbench.

MGX was successfully tested in the summer of 2007 by 35 science educators at the Summer 2007 BioQUEST workshop; several of them plan to use MGX in their courses in the fall of 2007. We will be using MGX this fall in Bio 111; we plan to continue development of different genotype–phenotype relationships as well as using MGX to model evolution.

HOW TO OBTAIN AND USE THE SOFTWARE

All of the applications described in this article, along with sample lab manual sections, are available free of charge via the Internet from <http://intro.bio.umb.edu/BW/software.html>. All applications are available in both Mac OS X and Windows format. Computers running Windows will require Java; it is available free from <http://www.java.com/en/download/manual.jsp>. None of the applications requires a powerful computer; they work satisfactorily on a 664MHz PC with 512 MB RAM running Windows XP or 98.

Additionally, VGL and GX along with a series of exercises are available as part of a textbook of practice problems [11]. GX runs as a stand-alone application; it can also be embedded as a Java applet in a web page. As an applet, many GX parameters can be set including initial DNA sequence, promoter, terminator, and splice site sequences as well as whether the gene is prokaryotic or eukaryotic.

SUMMARY

We have developed three interactive simulations that allow students to explore genetics, biochemistry, and molecular biology. Based on our experience and others' work with simulations, these programs should allow students to strengthen their understanding of these key concepts by exploring them at their own pace and based on their own questions and misconceptions. Preliminary evaluation suggests that they are highly successful; we are beginning more detailed evaluative studies.

Acknowledgments—MGX and its components are the result of a multiyear collaboration between the two authors and the second author's computer science students. We thank (in alphabetical order) the following for their contributions to the code: Sumana Adma, Vinod Aggawal, Bogdan Calota, Ruchi Dubey, Tao-Hung Jung, Pradeep Kadiyala, Chitra Karki, Prasoona Kejriwal, Nikunj Koolar, Wei Ma, Naing Naing Maw, David A. Portman, Namita Singla, Chung Ying Yu, and Ziping Zhu. We

also thank Bess Thaler and Lois Luberic for survey analysis and video transcription. Finally, we thank the people of Bio-QUEST for their constant support, encouragement, and inspiration during the development of these programs.

REFERENCES

- [1] T. de Jong (2006) Technological advances in inquiry learning, *Science* **312**, 532–533.
- [2] M. Steiff, U. Wilensky (2003) Connected chemistry—Incorporating interactive simulations into the chemistry classroom, *J. Sci. Educ. Technol.* **12**, 285–302.
- [3] T. de Jong, W. R. van Joolingen (1998) Scientific discovery learning with computer simulations of conceptual domains. *Rev. Educ. Res.* **68**, 179–201.
- [4] K. M. Lee, G. Nicoll, D. W. Brooks (2004) A comparison of inquiry and worked example web-based instruction using physlets, *J. Sci. Educ. Technol.* **13**, 81–88.
- [5] B. White, E. Bolker, N. Koolar, W. Ma, N. Maw, C. Yu (2007) The virtual genetics lab: A freely-available open-source genetics simulation, *Am. Biol. Teacher* **69**, 694–697.
- [6] P. Soderberg, J. R. Jungck (1994) Genetics construction kit: A tool for open-ended investigation in transmission genetics, *J. Comput. Higher Educ.* **5**, 67–84.
- [7] R. Hafner, J. Stewart (1995) Revising explanatory models to accommodate anomalous genetic phenomena: Problem solving in the “context of discovery.” *Sci. Educ.* **79**, 111–146.
- [8] E. A. Finkel (1996) Making sense of genetics: Students’ knowledge use during problem solving in a high school genetics class, *J. Res. Sci. Teaching* **33**, 345–368.
- [9] B. White, S. Kim, K. Sherman, N. Weber (2002) Evaluation of molecular visualization software for teaching protein structure: Differing outcomes from lecture and lab. *Biochem. Mol. Biol. Educ.* **30**, 130–136.
- [10] D. Botstein, G. Fink (1988) Yeast: An experimental organism for modern biology. *Science* **240**, 1439–1443.
- [11] B. White, M. Mischke (2006) *A Problems Approach to Introductory Biology*. American Society for Microbiology, Washington, DC.