INQUIRY & INVESTIGATION

The Virtual Genetics Lab:

A Freely-Available Open-Source Genetics Simulation

BRIAN WHITE ETHAN BOLKER NIKUNJ KOOLAR WEI MA NAING NAING MAW CHUNG YING YU

We have developed the Virtual Genetics Lab (VGL), a computer simulation of transmission genetics in a hypothetical insect. The program presents a student with a genetic phenomenon–the inheritance of a randomly-selected trait. The student's task is to determine how this trait is inherited. He/she constructs a genetic model for the phenomenon by designing his/her own experimental crosses; the results of these crosses are generated by the software. He/she then analyzes the results, revises hypotheses as necessary, and continues investigations until convinced that he/she has determined how the trait is inherited. The open-ended nature of the simulation supports a variety of strategies for genetic analysis. We have found the software to be a useful way to reinforce concepts of genetics as well as to illustrate how scientists test hypotheses. We have used VGL successfully at the introductory undergraduate level; similar software has been used successfully in high schools (Slack & Stewart, 1990) and middle schools (Echevarria, 2003).

Background

Genetics is a core element of modern biology and a part of many high school and college biology classes. In many of these courses, students show that they understand the principles of genetics by being able to solve genetics problems. Because problem solving can often require a deep understanding of the material, it is used as a tool for both teaching and evaluation. Many genetics problems require cause-to-effect reasoning (for example, "Predict the offspring from a cross of . . . "); however, research has shown that some students can successfully solve these problems even when their understanding of fundamental genetic concepts is weak (Stewart, 1983). In contrast, the most challenging problems require effects-to-cause reasoning (for example, "Given these data, how is this trait inherited . . . "). Because these problems require a deeper conceptual understanding, they will be more effective tools for learning

Hypothesis generation and testing are central to the scientific process; thus learning to test hypotheses in a scientific manner is also a common goal of most beginning biology courses. Many science education reform documents call for science students to learn this skill; for example, the NRC's *National Science Education Standards* (NRC, 1996) call for students to know how to "Design and conduct scientific investigations," and (Bransford et al., 1999) strongly encourage science classroom activities where ". . . students design studies, collect information, analyze data, construct evidence, and then debate the conclusions that they derive from their evidence" (p. 171). However, it is often difficult to provide challenging hypothesis-testing activities given the limited class time, student experience, and scientific equipment found in many teaching laboratories.

Several computer simulations of genetics have been developed that offer students a chance to practice effects-to-cause reasoning and hypothesis testing. Each application allows the student to design crosses; the program calculates the results, which the student must then interpret. CATLAB (www.emescience.com/bio-software-catlab.html) is a simulation of the genetics of coat color, tail length, and other traits in cats; use of CATLAB has been shown to require an in-depth understanding of genetics (Simmons & Lunetta, 1993). Biologica (www.concord.org/biologica/) is a collection of genetics simulations involving pea plants and dragons; it too has been shown to be an effective tool for teaching genetics (Hickey et al., 2003). The Virtual Flylab (www.biologylabsonline.com) is a simulation of fruit fly genetics that is used in many classrooms.

The Genetics Construction Kit (GCK, www.bioquest.org/indexlib.html; Soderberg et al., 1994) simulates fruit fly genetics. Studies of GCK have shown that students must use their knowledge of genetics to solve problems with this software (Hafner & Stewart, 1995; Finkel, 1996). Among the simulations described, GCK provides students with the widest variety of possible genetics problems. This is because, each time GCK is launched, the program chooses randomly from a large set of possible genetic models, characters, and traits; these choices can be constrained by the instructor to provide differing levels of difficulty for the students. As a result, each GCK problem is different. GCK has been used extensively in classrooms and for education research (Soderberg et al., 1994; Hafner & Stewart, 1995; Finkel, 1996; Echevarria, 2003). The GCK program was developed in the late 1980s for the Macintosh platform; since that time, computers have substantially increased in power and graphics capability.

BRIAN WHITE is Assistant Professor of Biology at UMass Boston, Boston, MA 02125; e-mail: Brian.white@umb.edu. ETHAN BOLKER is Professor of Mathematics and Computer Science at UMass Boston. NIKUNJ KOOLAR, WEI MA, NAING NAING MAW, and CHUNG YING YU were students in Professor Bolker's software engineering class when they wrote VGL.

VGL is a derivative of GCK that takes advantage of these improvements in technology to incorporate added functionality. It features a color user interface with large icons and pictorial representations of phenotypes, ZZ/ZW sex-linkage in addition to XX/XY, compatibility with most modern operating systems, free distribution, and open-source customizability. With VGL, students can practice the logic of genetic analysis without the distractions of wet labs. It should not replace a wet lab–the technical details of lab are an essential part of appreciating and understanding genetics. VGL allows students more opportunities to test hypotheses in a limited time. We have used this program for several semesters at UMass Boston with great success. Fully-functional VGL (not a demonstration version or a trial license) along with documentation, sample lab manuals, and java source code is available for free download at http://intro.bio.umb.edu/VGL/index.htm .

The VGL Program

The VGL user interface has been designed to be as intuitive as possible. When VGL is launched, the user selects *New Problem Type* . . . from the *File* menu. Her choice of problem type sets constraints on the range of genetic models that VGL will choose from when it generates an actual problem. VGL problems concern one gene with two or three alleles. The underlying genetic model is specified in terms of:

Type of dominance

- Two alleles
  - Simple dominance (heterozygote phenotype is identical to that of the dominant homozygote)
  - Incomplete dominance (heterozygote phenotype is different from either homozygote)
- Three alleles
  - Hierarchical dominance (A dominant to B and C; B dominant to C; C is recessive to all)
  - Circular dominance (A dominant to B; B dominant to C; C dominant to A)

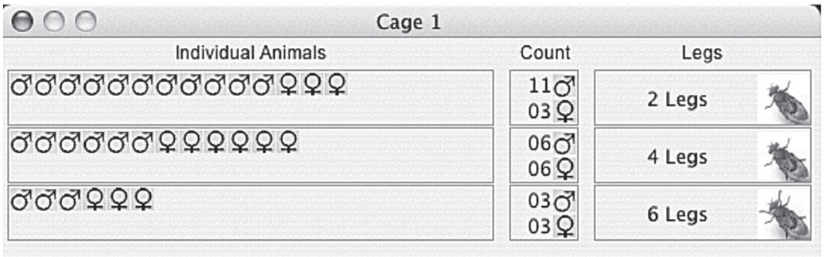
Type of Linkage

- Autosomal (not sex-linked)
- Sex-linked with XX females and XY males
- Sex-linked with ZZ males and ZW females

The type of dominance and the type of sex-linkage are independent choices; this yields a maximum of 12 possible underlying genetic models in the most advanced mode.

Once the user has chosen the problem type, VGL randomly selects a specific genetic model honoring the constraints on type of dominance and type of linkage. Then it chooses randomly from a large list of characters (body color, number of legs, wing shape, etc.) and from a list of traits for that particular character. The traits are chosen so as not to reveal the nature of the underlying model. For example, having legs will not necessarily be dominant to lacking legs; likewise, orange will not necessarily be the heterozygote of red and yellow. Because the character and traits are chosen randomly each time VGL is launched, red body color may be X-linked dominant in one problem but autosomal recessive in another. Therefore, the students must solve problems using their knowledge of genetic analysis, rather than their knowledge of specific traits. VGL first generates Cage 1 containing creatures with randomly-selected genotypes. Only the phenotype of the creatures is visible; the student must determine the underlying genetic model. Figure 1 shows a sample Cage 1 for a problem type specification that allowed all six possible genetic models.

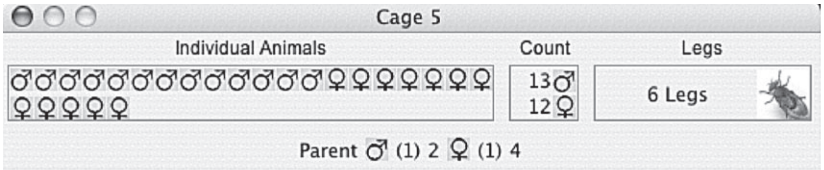
Figure 1. A typical starting population.



Cage 1

Individual Animals	Count	Legs
♂♂♂♂♂♂♂♂♂♂♂♀♀♀	11♂ 03♀	2 Legs
♂♂♂♂♂♂♀♀♀♀♀♀	06♂ 06♀	4 Legs
♂♂♂♀♀♀	03♂ 03♀	6 Legs

Figure 2. The results of the cross described in the text.



Cage 5

Individual Animals	Count	Legs
♂♂♂♂♂♂♂♂♂♂♂♂♂♀♀♀♀♀♀♀ ♀♀♀♀♀	13♂ 12♀	6 Legs

Parent ♂ (1) 2 ♀ (1) 4

The individuals are sorted by sex and phenotype. In this case, there are 11 males and 3 females with two legs; 6 males and 6 females with four legs; and 3 males and 3 females with six legs. Clicking on the small phenotype images at the right of each row pops up a window with a large color illustration of the corresponding phenotype. Since the data show three different leg-number phenotypes, and given that the problem type in this case is limited to only two alleles, it is reasonable to suspect incomplete dominance is at work in this problem.

At the left of the cage, the individual male and female symbols represent individual creatures that can be selected for crossing. To perform the cross, the student clicks on the two parent symbols and clicks *CROSS* at the top of the VGL window. VGL then generates a new cage of offspring by randomly combining pairs of alleles from the two parents. As a result, the numbers of offspring will usually approximate, but not exactly match, the expected Mendelian ratios. Figure 2 shows the results of a cross of a male with two legs and a female with four legs.

This cage shows the offspring as well as the phenotype and cage-of-origin of the parents (bottom row). These results are also consistent with incomplete dominance and suggest that six-legs is the heterozygote of two-legs and four-legs. The results also do not suggest sex-linkage. The student can then select any of the creatures in any cages for further crosses to confirm or disconfirm this hypothesis. He/she continues until he/she has convinced him/herself, teacher, and/or classmates that he/she has discovered the correct genetic model. Here, as in research, it is not possible to look behind the scenes to see the "right answer."

Because the student designs the crosses him/herself, VGL accommodates a variety of experimental approaches as does GCK (see Slack & Stewart, 1990). Some students make many crosses until they see a pattern emerging while others prefer to analyze each cross carefully before choosing what to do next. Either strategy can lead to a successful outcome. Moreover, some VGL crosses will be more informative than others, in contrast to most teacher-generated problems that tend to present only relevant information. For example, a cross of red X blue that gives some red and some blue offspring is consistent with either red or blue being dominant. In contrast, a cross of green X orange that gives all orange offspring is only consistent with orange being dominant. The student must therefore decide how best to use each result. This requires careful coordination of theory, experiment, and evidence.

Although the genetic model and individuals' genotypes are invisible in VGL's normal mode, there is a practice mode for beginning students that allows access to this information. In practice mode, clicking on a button on Cage 1 reveals the underlying genetic model, and *Balloon Help* shows the genotype of each individual creature. Practice mode can be enabled or disabled in the problem type specification.

VGL also incorporates two features that make the students' work easier. Students can save their work in progress and come back to the same problem and all of their crosses at a later date. It is also possible to generate a "printer-friendly version" of their work that lists their crosses and offspring for incorporation into a lab report or other documentation.

Our VGL Lab Exercises

We have used VGL for several semesters in General Biology I (Bio 111, the first semester course for biology majors; intro.bio.umb.edu/111-112/) at UMass Boston. After instruction in the principles of genetics and problem solving, students reinforce their understanding of genetics as they apply the principles they have learned in two virtual lab exercises. Students work in groups of two or three at each computer workstation. Relevant sections of our lab manual can be downloaded from the VGL site.

The first VGL exercise follows introductory lectures on Mendel. Its purpose is to familiarize the students with VGL and to reinforce the logic Mendel used. The problems are limited to autosomal traits with simple dominance; practice mode is disabled. When the members of a group have decided that they have found a model consistent with their data, they ask another group of students to serve as reviewers of their work. The group designates a particular cross for the reviewers to perform and describes the expected offspring. The reviewers perform the cross and if the results match the prediction, the group receives credit for solving the problem. In this way, the students must convince an independent group of their peers that they have found the correct model. Each student prepares a lab report for this exercise that describes the solution to one of the problems he/she worked in lab. The students are provided with a lab report template that guides them in constructing a sound scientific argument.

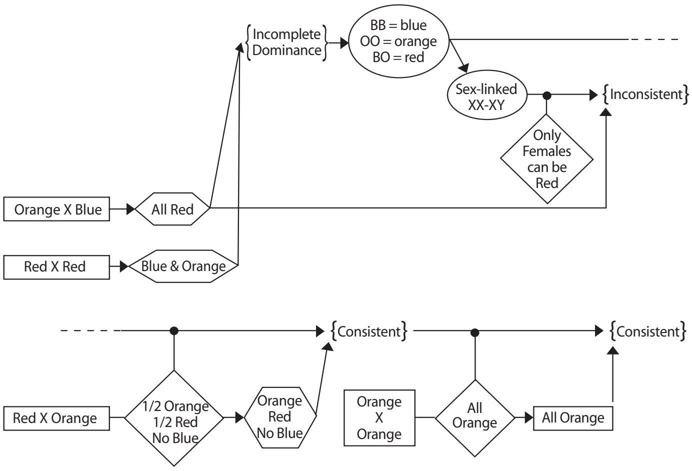
The second VGL exercise acts as a capstone for the genetics section of the course; problems here include all possible types and practice mode is disabled. Here, the TA serves as the reviewer for these more sophisticated problems. The lab report for this session follows the format of a scientific paper (abstract, introduction, methods, results, conclusions) where each student presents an argument for his/her genetic model.

Students need not use VGL in a laboratory; instead, they can download the software and work problems at home on their own computers.

Hypothesis Testing Using VGL

We recorded the speech and actions of several groups of students solving several VGL problems and then analyzed the recordings to explore the logical process the students followed. This process can be diagrammed using a Reasoning Map (White, 2004); Figure 3 shows a sample from the second VGL lab. Reasoning Maps diagram elements of the hypothesis-testing process: hypotheses (ovals), tests (rectangles), predictions (diamonds), results (hexagons), and conclusions (text in {braces}) as well as the connections between them (arrows).

Figure 3. A typical Reasoning Map for students solving a VGL problem.



This group of students began with two crosses performed in the absence of a particular hypothesis (orange X blue and red X red). The results of these crosses led them to conclude that incomplete dominance was involved. This in turn led them to a specific model for the inheritance of body color in their problem (incomplete dominance where red is the heterozygote of blue and orange). They discussed the possibility that the trait could be XX/XY sex-linked; that model predicted that only females could be red. Based on their observation of red males from the first cross, they ruled out this hypothesis. They did not explore ZZ/ZW sex-linkage and (correctly) considered only autosomal models from then on. The following two crosses served to confirm their model; with all the data from four crosses explained, they were convinced that they had the correct model. Interestingly, they did not attempt to demonstrate that their model was the only one consistent with their data.

Several conclusions follow from this and other such maps. First: the students we observed apply their knowledge of genetics to generate hypotheses, design crosses, make predictions, and draw conclusions. Second: these students can use VGL to test hypotheses in a logical and systematic manner. Third: the conclusions they draw are, in general, consistent with their data as well as being correct. Finally, based on our examination of other groups' work, many possible paths to a correct solution are possible. Thus, VGL provides a powerful tool for students' application of genetic principles.

Problem Type Files & the VGLAdmin Program

The range of problems that VGL generates can be constrained by configuring specific problem type files. Specifically, these files determine the range of possible models, the probability that each type of model will occur, and the availability of practice mode. Some sample problem type files are included with the VGL download; these range from simple dominance with no sex-linkage and practice mode enabled (Level 1) to all possible models allowed with practice mode disabled (Level 10). The instructor can create custom problem type files using the VGLadmin program (downloaded separately); a users' guide is provided in the download.

Technical Specifications

System Requirements

VGL is compatible with Windows 98 and higher, Mac OS-X, and Linux/UNIX. It is not compatible with Mac OS 9 or lower. VGL requires the Java Runtime Environment (JRE) version 1.4.2 or higher. This is available for free download from java.sun.com and comes pre-packaged with many modern operating systems. VGL runs well on low-end PCs (266MHz Pentium II with 198Mb RAM running Windows 98 or RedHat Linux 9) and Macs (333MHz PowerPC with 384Mb RAM running OSX).

Open Source

VGL was written by a group of students in the Masters Program in Computer Science at UMass Boston. The source code is available online at the VGL site according to the terms of the Gnu Public License (www.gnu.org/licenses). Briefly, the source code can be modified and recompiled by others as long as credit is given to the authors and that all improvements and changes are released freely with the same license. This means that VGL can be used as is, locally adapted, and developed by a community of users. The user interface can be changed, traits can be added or removed, and new genetic models can be added. We hope to develop a community of users and programmers who will help to develop VGL further. Please contact the lead author if you have comments, bugs to report, or would like to participate in development of VGL.

Conclusion

VGL was developed based on successful experience with other open-ended genetics simulations. It retains some of the key features of the existing software and contains adaptations that increase its usability in the classroom. Since VGL is open-source, it will be possible to continue its development as the needs of the user community evolve. We feel that it will make a valuable addition to the toolkit of introductory genetics teachers at many levels.

Acknowledgment

This work was, in part, supported by NSF grant #9984612.

References

Bransford, J.D., Brown, A. L. & Cocking, R.R. (Editors). (1999). *How People Learn: Brain, Mind, Experience, and School*. Washington, DC: National Academy Press.

Echevarria, M. (2003). Anomalies as a catalyst for middle school students' knowledge construction and scientific reasoning during science inquiry. *Journal of Educational Psychology, 95*(2), 357-374.

Finkel, E.A. (1996). Making sense of genetics: Students' knowledge use during problem solving in a high school genetics class. *Journal of Research in Science Teaching, 33*(4), 345-368.

Hafner, R. & Stewart, J. (1995). Revising explanatory models to accommodate anomalous genetic phenomena: Problem solving in the context of discovery. *Science Education, 79*(2), 111-146.

Hickey, D.T., Kindfield, A.C.H., Horwitz, P. & Christie, M.A. (2003). Integrating curriculum, instruction, assessment, and evaluation in a technology-supported genetics learning environment. *American Educational Research Journal, 40*(2), 495-538.

National Research Council. (1996). *National Science Education Standards*. Washington, DC: National Academy Press.

Simmons, P.E. & Lunetta, V.N. (1993). Problem-solving behaviors during a genetics computer simulation: Beyond the expert/novice dichotomy. *Journal of Research in Science Teaching, 30*(2), 153-173.

Slack, S.J. & Stewart, J. (1990). High school students' problem-solving performance on realistic genetics problems. *Journal of Research in Science Teaching, 27*(1), 55-67.

Soderberg, P. & Jungck, J.R. (1994). Genetics construction kit: A tool for open-ended investigation in transmission genetics. *Journal of Computing in Higher Education, 5*(2), 67-84.

Stewart, J. (1983). Student problem solving in high school genetics. *Science Education, 67*(4), 523-540.

White, B. (2004). Reasoning Maps: a generally-applicable method for characterizing hypothesis-testing behavior. *International Journal of Science Education, 26*(14), 1715-1731.