GAIL: A Genetics Argumentation Inquiry Learning System

Nancy L. Green, Mark Hinshaw, Carl Martensen, Meghana Narasimhan and Tshering Tobgay

Department of Computer Science
University of North Carolina Greensboro
Greensboro, NC 27402 USA
nlgreen@uncg.edu

Abstract

This paper discusses ongoing work to build an argumentation inquiry learning system, GAIL. The purpose of GAIL is to support students in constructing scientific arguments in an undergraduate genetics course in order to facilitate deeper learning and improve argumentation skill. Students can construct argument diagrams using a drag-and-drop graphical user interface. The system constructs arguments on-the-fly to use as a knowledge source for evaluating the learners' arguments and providing intelligent feedback.

Introduction

Argumentation plays an important role in science. There has been significant interest within the field of science education in argumentation. However, students' arguments have been shown to be deficient in a number of ways, e.g., lacking support for claims (Bell and Linn 2000; Jiménez-Aleixandre, Rodríguez and Duschl 2000), failing to provide alternative explanations (Lawson 2003; Schwarz et al. 2003), and using inaccurate or irrelevant support (Zohar and Nemet 2002). Computer-supported cooperative learning systems for argumentation have been developed (Kirschner et al. 2003; Scheuer et al. 2010; Pinkwart and McLaren 2012) but they do not provide human-level expertise in evaluating student argumentation. Furthermore, in larger-enrollment classes human teachers may not have sufficient time to evaluate learners' arguments nor to provide one-on-one feedback as the learner works on an argumentation lesson.

To address this problem we are implementing a prototype genetics argumentation inquiry learning system, GAIL. GAIL will support learning to argue about cases in human genetics. This is a field that applies findings from genetics research to biomedical reasoning. GAIL is designed for use in an introductory genetics course for undergraduates that many biology majors find the most challenging course in the biology core curriculum. We hope that use of GAIL will improve argumentation skill, facilitate deeper learning of genetics, and increase interest and engagement in science.

Each GAIL lesson requires learners to construct arguments for and against certain hypotheses about a genetics case, e.g., about an infant who may have an inherited metabolic disorder or someone who inherited a genetic variant that is associated with increased risk of colon cancer. A prototype user interface is shown in Figure 1 (see last page). Information relevant to the lesson is provided by GAIL on the left-hand side of the screen: the *Problem* (to give a certain argument); *Hypotheses* (which can be used in the argument, but note that some are incorrect); *Data* from medical records about the patient and the patient's biological family; and *Connections*, a list of facts or principles of genetics. The center of the screen shows two arguments constructed by a learner. To construct the arguments, the learner searched for text components on the left-hand side of the screen, dragged them into the workspace in the center of the screen, and connected the components. Arrows point from support to conclusion. The connection between support and conclusion – known as the *warrant* in argumentation theory (Toulmin 1998) – is linked by a line to the arrow.

In Figure 1, the problem is to give two arguments for the hypothesis that the patient (referred to as J.B.) has cystic fibrosis, i.e., has two variant alleles of the *CFTR* gene. The leftmost "chain" of arguments begins with data (at the bottom of the argument diagram) about J.B.'s respiratory problems. The learner used that data to support an intermediate hypothesis that J.B. has thickened mucus in the lungs, which is used to support an intermediate hypothesis that J.B. has abnormal CFTR protein, which is used to support the main hypothesis/conclusion that J.B. has cystic fibrosis. Branching from the right hand side of the diagram, connections (warrants) provide justification for each step of the argument. The second argument for the same hypothesis begins with data about J.B.'s lab test result.

GAIL's innovation is that the system can *generate* arguments for evaluating the correctness of the learner's arguments, rather than requiring the arguments to be constructed by a teacher. Use of the generated arguments enables GAIL to provide intelligent feedback on both the structure and content of the learner's argument.

System Design

The author of an argumentation lesson to be used in GAIL creates an XML-formatted file that contains: (1) strings of natural language text -- the problem, hypotheses, data, and connections -- to be displayed to learners on the left-hand side of the graphical user interface as shown in Figure 1; (2) a specification of an internal causal domain model; and (3) mapping of the natural language strings in (1) to concepts and relations in the domain model. GAIL's **Authoring Tool** provides (1) to the user interface, uses (2) to build an internal Knowledge Base, and stores (3) to enable GAIL's Argument Evaluator to semantically interpret learners' argument diagrams, to avoid the challenge of interpreting natural language input.

Based on our previous work on modeling genetics (Green 2005), a **Knowledge Base** (KB) describes (i) instances of small set of concepts in human genetics (e.g. genotype, protein, phenotype) and (ii) causal relations between these concepts. Causal relations are defined in terms of influence and synergy relations of a qualitative probabilistic network (QPN) (Druzdzel and Henrion 1993). Different genetics KBs can be constructed automatically from XML-language descriptions of the causal model specified using the Authoring Tool.

Argumentation schemes are descriptions of acceptable, but often defeasible, patterns of reasoning (Walton, Reed and Macagno 2008). Following the same approach as in our previous research on argument generation (Green, Dwight, Navoraphan and Stadler 2011), GAIL's **Argument Generator** creates arguments by instantiating abstract argumentation schemes with concepts and relations from a QPN. The argumentation schemes are formalized in structures including *claim/conclusion*, *data*, and *warrant*. The propositions used as claim or data describe states of variables in a QPN. The warrant expresses formal constraints on the nodes of the QPN in terms of probabilistic influence and synergy relations. The distinction between premises as data and warrant reflects their difference in function and source of information. Data premises refer to a particular case, whereas warrants describe biomedical principles and other generalized knowledge. The *condition* of GAIL's argumentation schemes is used to represent possible exceptions.

For example, the argumentation scheme for *reasoning from effect to cause*, shown in Figure 2, can be instantiated from a KB to create an argument that a patient has HNPCC (a mutation in the *MLH1* gene, a hereditary condition predisposing one to colon cancer) based upon the data that genetic testing showed a variant *MLH1* allele, and the connection (warrant) that having HNPCC typically leads to that test result. (The exception condition for this argumentation scheme asks whether there is an alternative explanation for the data.) An argument diagram representing this argument is shown in Figure 3; to save space, instead of natural language text from the graphical user interface, the diagram uses letters representing propositions, where A is the conclusion, B is the data, and S+(A,B) is the warrant.

Claim: $A \geq a$
Data: $B \geq b$
Warrant: S+(<A,a>, <B,b>)
Condition: ¬ exists C X-({C,A},<B,b>): $C \geq c$

Figure 2. Argumentation scheme.

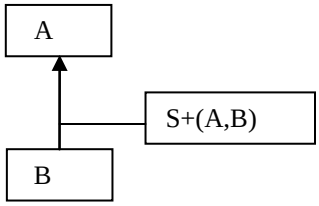

Figure 3. Simple argument.

Figure 4 shows a more complicated argument. The main claim (A=1) is that a patient's mother has exactly one mutated *CFTR* allele. The left-hand subargument is for the hypothesis that she has one or two mutated *CFTR* alleles. That subargument is supported by the hypothesis (D=2) that the patient has cystic fibrosis (two mutated *CFTR* alleles), and is warranted by the synergy relation, X^0(<A=1,B=1>, D=2), i.e., that a child who has two mutated alleles inherited one from the mother and one from the father. Note that the claim D=2 would be supported by another subargument (not shown in Figure 4). The right-hand subargument is for the hypothesis that the mother does not have two mutated *CFTR* alleles. This is supported by the data (¬C) that she does not have cystic fibrosis symptoms, and warranted by the positive influence relation between having two mutated *CFTR* alleles and symptoms of cystic fibrosis.

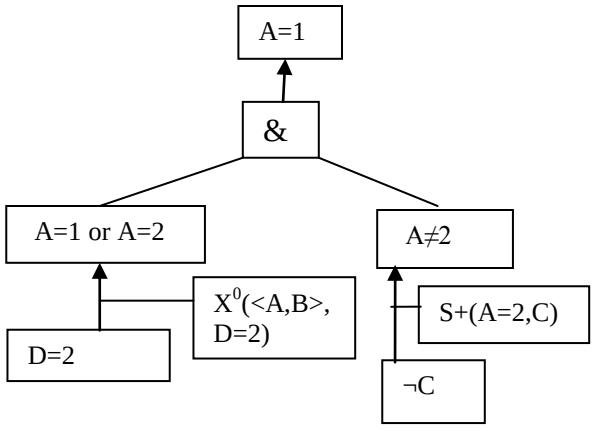

Figure 4. More complicated argument.

Currently, GAIL employs seven argumentation schemes. Arguments such as that shown in Figure 4 can be generated by chaining and/or conjoining subarguments. Arguments are represented internally as directed acyclic graphs. Unlike previous educational systems in which all possible arguments to be used by the system had to be encoded by an author in natural language (e.g. Woolf et al. 2005) or in propositional logic (e.g. Yuan et al. 2008), GAIL's arguments are generated by the system on-the-fly. Since the argument generator and schemes do not encode domain-specific or patient-specific content, they can be used to generate arguments in any domain whose domain knowledge can be represented in a similar format.

After a learner has created an argument diagram, GAIL's **Argument Evaluator**'s task is to evaluate the acceptability of the structure and content of the learners' argument diagram. First, the learner's diagram is translated into an argument structure containing KB concepts and links. The translation process uses the correspondences between text the learner sees on the screen and KB concepts and mappings provided via the Authoring Tool from (3). The translated structure is in the same representation as arguments produced automatically by the Argument Generator. Then the internal representation of the learner's argument is compared to all possible arguments created for the given problem by the Argument Generator.

GAIL's **Feedback Generator** can respond to the following types of errors, where components of the learner's argument are enclosed in brackets below:

- <main claim> does not match the claim to be argued for in the problem.
- <hypothesis> is unsupported (i.e. no argument is provided for it).
- <data or hypothesis> does not support the given <conclusion>.
- <data or hypothesis> does not support the given <conclusion> directly; one or more hypotheses are missing between it and the given <conclusion>.
- Additional data or hypothesis must be conjoined to the given <data or hypothesis>.
- <data or hypothesis> is given as supporting <data or hypothesis 2> but it should be conjoined to <data or hypothesis 2>.
- The warrant is missing between the given <data> and <conclusion>.
- The given <warrant> is irrelevant to the given <data> and <conclusion>.

Note that, unique to GAIL, most of the above types of errors are semantic in nature.

For each type of error, the author of a GAIL lesson or a system developer can provide a severity code and three levels of feedback message templates in an XML-formatted file. In the current implementation of GAIL, a student is allowed three tries to construct an acceptable argument. After each try, the Feedback Generator selects the most general (lowest level) unused message for an error; each time the student makes the same error on a subsequent try, the next more specific (next higher level) message is selected. A positive message is generated when an error is corrected on the next try. Currently, the Feedback Generator displays only the message for the most serious error to the student, but writes all of the detected errors to a logfile for inspection by the teacher.

Feedback Example

To illustrate the learner's interaction with GAIL, suppose the problem was to give an argument for the hypothesis that J.B.'s brother might have malnutrition and poor growth. Internally, GAIL generates a chained argument beginning with the data that J.B.'s brother has been diagnosed as having cystic fibrosis, which supports an intermediate hypothesis that his CFTR protein is abnormal, which supports an intermediate hypothesis that he might have pancreatic abnormality, which supports the main hypothesis that he might have malnutrition and poor growth. (The warrants of GAIL's argument are not described here to save space.) However, on the first try the learner's argument contains the main claim that J.B. (rather than J.B.'s *brother*) has cystic fibrosis, which does not match the problem. Since this type of error has been given the highest severity code, GAIL would tell the student that the main claim of his argument does not match the problem. On the second try, the student fixes the main claim and constructs a new argument. GAIL informs him that the problem noted on the last try has been fixed. However, the student's argument is missing the intermediate hypothesis that J.B.'s brother might have pancreatic abnormality, so GAIL also informs the student that one or more intermediate hypotheses are missing between J.B.'s brother having abnormal CFTR protein and J.B.'s brother having malnutrition and poor growth. On the third try, the student adds the missing hypothesis but provides an irrelevant warrant. GAIL would inform the student that he has made progress but that the warrant he just added is irrelevant to that subargument.

Conclusion

This paper discusses ongoing work to build an argumentation inquiry learning system, GAIL. The purpose of GAIL is to support students in constructing scientific arguments in an undergraduate genetics course in order to facilitate deeper learning and improve argumentation skill. The system generates arguments on-the-fly to use as a knowledge source for evaluating the learners' arguments and providing formative and summative intelligent feedback.

All of the components described in this paper have been implemented in Java. Future work includes

improvements to the user interface and the Feedback Generator. The Feedback Generator will be made more intelligent to address certain types of errors that we have observed in our formative evaluation studies. For example, a learner "flattened" a chained argument into a one-level structure by conjoining together all of the data and intermediate hypotheses. Note that in this case, the learner has selected the correct content but has just not structured the argument properly into subarguments and shown how one subargument builds upon another subargument. Because the Feedback Generator has access to arguments constructed by GAIL's Argument Generator, the Feedback Generator will be able to detect this type of error and provide more meaningful feedback than systems that do not have access to content. After these improvements are made, we plan to evaluate GAIL's effectiveness in an undergraduate genetics course.

Acknowledgments

Former graduate students Mark Hinshaw, Carl Martensen, Meghana Narasimhan, and Tshering Tobgay contributed to the implementation of GAIL for their MS Projects. Former graduate students Benjamin Wyatt and Chris Cain also contributed to the implementation of GAIL. Wyatt and Martensen received support from a UNCG Regular Faculty grant and Cain received support from the Computer Science Department. Dr. Malcolm Schug of the UNCG Department of Biology has provided helpful feedback on the project.

References

Bell, P., and Linn, M. 2000. Scientific Arguments as Learning Artifacts: Designing for Learning from the Web with KIE. *International Journal of Science Education*, 22(8), 797-817.

Druzdzel, M.J. and Henrion, M. 1993. Efficient Reasoning in Qualitative Probabilistic Networks. In *Proc. of the 11th Nat. Conf. on AI (AAAI-93)*, pp. 548-53.

Green, N.L. 2005. A Bayesian Network-Based Coding Scheme for Annotating Biomedical Information Presented to Genetic Counseling Clients. *Journal of Biomedical Informatics*. 38, 130-144.

Green, N., Dwight, R., Navoraphan, K., and Stadler, B. 2011. Natural Language Generation of Biomedical Argumentation for Lay Audiences. *Argument and Computation* 2(1), 23-50.

Jiminez-Aleixandre, M., Rodriguez, M., and Duschl, R.A. 2000. 'Doing the Lesson' or 'Doing Science': Argument in High School Genetics. *Science Education*, 84(6), 757-792.

Kirschner, P.A., Buckingham Shum, S.J., and Carr, C.S. (Eds.) 2003. *Visualizing Argumentation*. London, UK: Springer.

Lawson, A. 2003. The Nature and Development of Hypothetico-Predictive Argumentation with Implications for Science Teaching. *International Journal of Science Education* 25(11): 1387-1408.

Pinkwart, N. and McLaren, B.M. (Eds.) 2012. *Educational Technologies for Teaching Argumentation Skills*. Sharjah: Bentham Science Publishers

Scheuer, O., Loll, F., Pinkwart, N., and McLaren, B.M. 2010. Computer-Supported Argumentation: A Review of the State of the Art. *Computer-Supported Collaborative Learning* 5(1): 43-102.

Schwarz, B., Neuman, Y., Gil, J., and Ilya, M. 2003. Construction of Collective and Individual Knowledge in Argumentative Activity. *Journal of the Learning Sciences* 12(2): 219-256.

Toulmin, S.E. 1998. *The Uses of Argument*, Cambridge, UK: Cambridge University Press.

Walton, D., Reed, C., and Macagno, F. 2008. *Argumentation Schemes*. Cambridge, UK: Cambridge University Press.

Woolf, B., Reid, J., Stillings, N., Bruno, M., Murray, D., Rees, P., Peterfreund, A., and Rath, K. 2002. A General Platform for Inquiry Learning. In S.A. Cerri, G. Gouardères, and F. Paraguaçu (Eds.), *ITS 2002, LNCS 2363*, 681-697. London: Springer.

Yuan, T., Moore, D., and Grierson, A. 2008. A Human-Computer Dialogue System for Educational Debate: A Computational Dialectics Approach. *International Journal of AI in Education* 18, 3-26.

Zohar, A. and Nemet, F. 2002. Fostering Students' Knowledge and Argumentation Skills through Dilemmas in Human Genetics. *Journal of Research in Science Teaching* 39(1): 35-62.

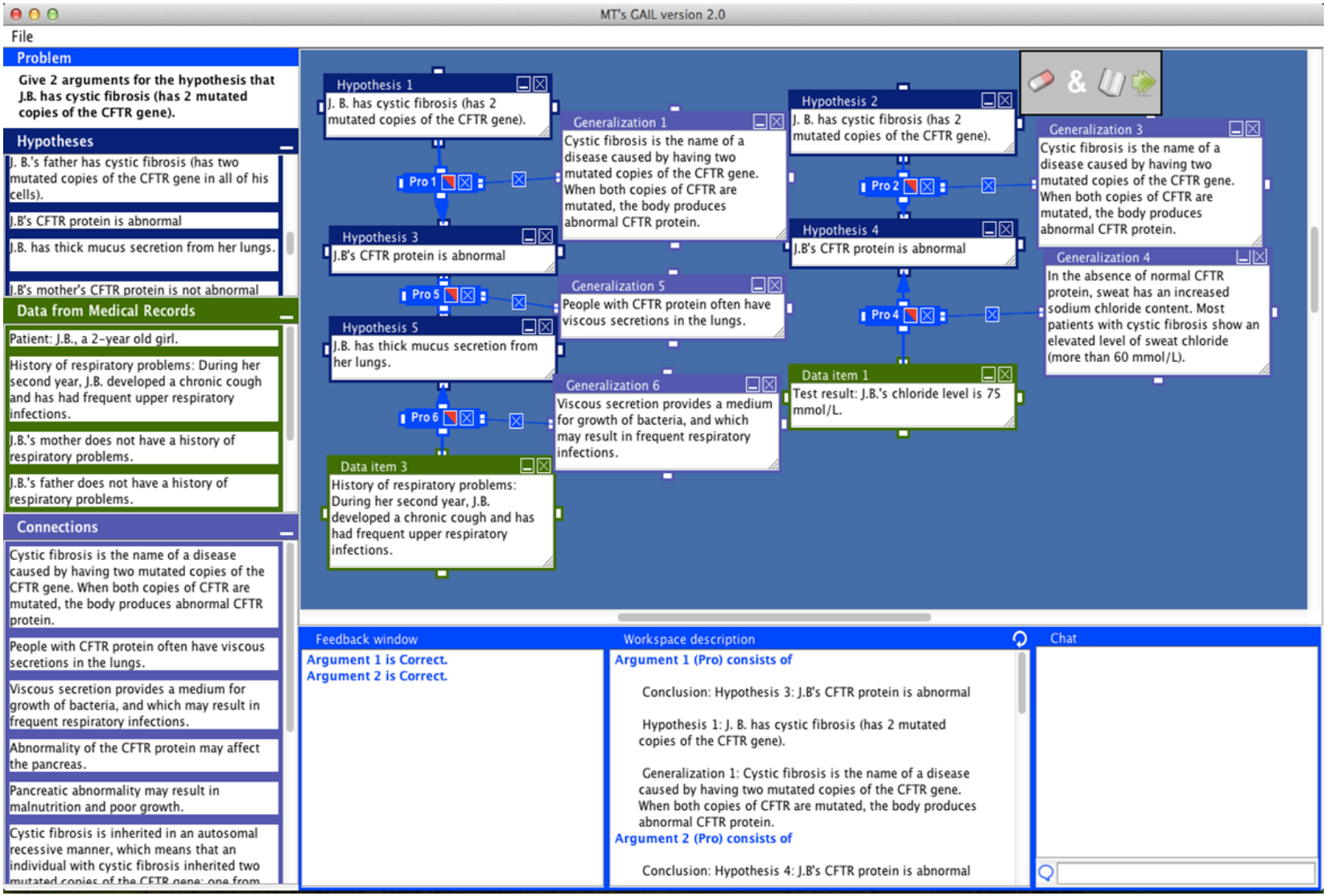

Figure 1. Screen shot of prototype GAIL user interface