Cognitive Tutoring in the use of Bioinformatics Tools

[1]✉MAKOLO, A. and [1]NKEMELU, D.

University of Ibadan
aumakolo@gmail.com, dnkemelu@gmail.com

Abstract

With the proliferation of simple and complex bioinformatics tools, there is the need to teach researchers how to use these tools effectively. To evaluate the potential of cognitive tutoring in the wide-scale adoption of several bioinformatics tools, we designed a simple prototype. We embedded a cognitive tutor, built with the Cognitive Tutor Authoring Tool, on a pre- existing platform, the Gene Adjacency Program, developed by the University of Ibadan Bioinformatics group. Our preliminary tests show that researchers who used the platform with the cognitive tutor embedded showed higher levels of competence and efficiency. These results indicate that cognitive tutors have the potential to teach bioinformatics researchers employing new tools how to efficiently use them and accurately make sense of their results.

Keywords: *Bioinformatics, Cognitive tutoring, Intelligent Tutoring System, Gene Adjacency.*

1. Introduction

A major activity in bioinformatics is to develop software tools to generate useful biological knowledge. There are several bioinformatics tools created with the goal of developing methods for the analysis and interpretation of complex datasets. Bioinformaticians make use of these tools to understand and discover novel solutions to the vast amount of biological problems available. The challenge, however, is on how to teach researchers, especially those with non-technical backgrounds, to make effective use of these tools.

Tutoring done by humans has proven to be a very effective form of instruction and there is evidence that expert tutors produce enormous learning gains [1]. However, with recent advancements in cognitive computing and artificial intelligence, computers have become efficient, flexible, scalable and economic alternatives to expert human tutors [2].

Human tutoring is widely believed to be the most effective form of instruction and tutoring available, and both experimental work and historical evidence confirm that expert human tutors can produce extremely large learning gains. Ever since computers were invented, they seemed capable of becoming both collaborators and alternatives to human experts. In recent years, cognitive tutors have been employed to help tutor people and the gains have been largely significant [3].

According to Aleven *et.al.* [4], a cognitive tutor is a particular kind of intelligent tutoring system that utilizes a cognitive model to provide feedback to users of complex systems as they are working through problems. The Cognitive Tutor programs utilize cognitive models and are based on model tracing and knowledge tracing. Most tutors perform user evaluation by flexibly comparison of their activity against generalized examples of problem-solving actions. Such tutors are referred to as example-tracing tutors. Example-tracing tutors are capable of sophisticated tutoring behaviors; providing step-by- step guidance on complex problems while recognizing multiple user strategies and, where needed, maintaining multiple interpretations of user behavior.

As more bioinformatics tools are developed to solve biological problems, researchers are faced

with the challenge of learning how to use these tools and accurately making sense of their results. The problems faced include: inability to use the tools, inability to effectively use the tools to get proper result and inability to make sense out of its result. The implementation of a cognitive tutor for bioinformatics tools is very important because it helps researchers without technical skills to efficiently use bioinformatics tools thereby helping to eliminate or at least reduce errors resulting from improper use of the tools.

In this work, we developed a cognitive tutor for a bioinformatics tool; the Microbial Gene Adjacency Visualization Software [5]. The inputs processed by the application program were RefSeq files downloaded from brcdownloads.patricbrc.org [6]. A web interface was designed to host the tutor. The cognitive model on which the tutor runs was built using the Cognitive Tutor Authoring Tool⊘R provided by Carnegie Learning [7]; by generalizing and annotating the behaviour graph and creating the Behaviour Recorder file [.brd] using extensible markup language (XML), correct and incorrect behaviours were demonstrated.

Results from the new system show that using the cognitive tutoring approach to teach bioinformatics can increase widespread adoption of tools and further progress in the field.
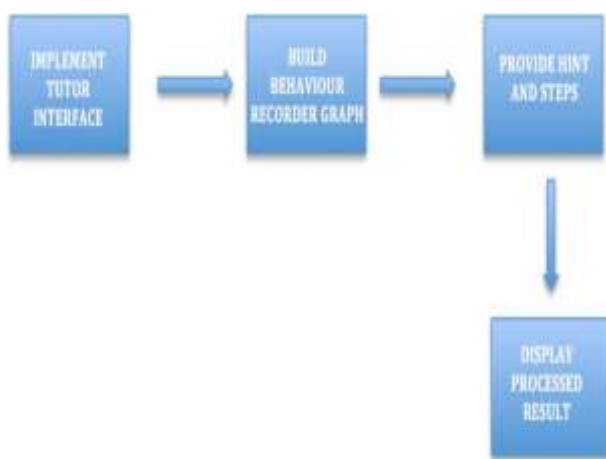

Fig. 1. Steps in setting up a cognitive tutor on an interface using the Cognitive Tutor Authoring Tool⊗R

2. Related Works
Wang and Vincent [10] defined cognitive computing in terms of cognitive informatics, as a multidisciplinary field that applies how the brain processes information and copes with decision making. The focus of cognitive computing is on mimicking the mechanisms of the brain to endow computer systems with the faculties of feeling, thinking and knowing such as is evident in intelligent tutoring systems and cognitive tutors [11]. In the following subsections, we will review existing literatures in the subject matter and introduce concepts in our overall work.

2.1 Intelligent Tutoring Systems (ITS)
An Intelligent Tutoring System (ITS) is a computer system that has the objective of providing immediate and customized instruction or feedback to learners, usually without intervention from a human teacher [12]. ITSs have the common goal of enabling learning in a meaningful and effective manner by using a variety of computing technologies. According to Joseph and Sharon [12], ITSs are being used in both formal education and professional settings and have demonstrated their capabilities and limitations. Modern day ITS tries to replicate the role of a teacher or a teaching assistant and involves problem generation, intelligent automatic feedback generation with a high recall value.

2.2 Cognitive Tutor
Anderson *et al.* [13], defines a Cognitive Tutor as a kind of intelligent tutoring system that utilizes a cognitive model to provide feedback to users as they are working through problems. This feedback immediately informs users of the correctness, or incorrectness, of their actions in the tutor interface. In addition, it also has the ability to provide context-sensitive hints and instruction to guide users toward reasonable next steps. According to them, Cognitive Tutors were originally developed to test ACT-R theory for research purposes and they are developed also for other areas and subjects such as computer programming and science.

The Cognitive Tutor programs are based on model tracing and knowledge tracing. Model tracing and knowledge tracing are essentially used to monitor students' learning progress, guide them to the correct path to problem solving, and provide feedback by checking actions like button clicks, and value entering. This enables learners to develop a complex problem-solving skill through practice. Typically, cognitive tutors provide such forms of support as: (a) a problem-

solving environment that is well designed and allows for "visible thinking" (b) step-by-step feedback on student performance (c) feedback messages specific to errors (d) context-specific next- step learning hints at student's request, and (e) individualized problem selection. Cognitive Tutors accomplish two of the principal tasks characteristic of human tutoring:

1) Monitors the student's performance and provides context specific individual instruction, and.
2) Monitors the student's learning and selects appropriate problem solving and learning activities.

2.3 The Cognitive Tutor Authoring Tool (CTAT ⊗R)

Cognitive Tutors and Example-Tracing Tutors, the two types of tutors supported by CTAT⊗R, represent different trade-offs between ease of authoring on the one hand and generality and flexibility of the resulting tutors on the other. Cognitive Tutors are rooted in the ACT-R theory of cognition and learning. They are capable of interpreting users problem-solving behavior by employing a cognitive model that captures, in the form of production rules, the skills that the user is expected to learn [4]. Authoring a Tutor involves the following development steps:

(a) Create the graphical user interface (GUI) used by the students,
(b) Demonstrate alternative correct and incorrect solutions,
(c) Annotate solutions steps in the resulting behaviour graph with hint messages,
(d) Feedback messages, and labels for the associated concepts or skills,
(e) Inspect skill matrix and revise.

CTATs Example-Tracing Engine uses the Behavior Graph to guide a user through a problem, comparing the students' problem-solving behavior against the graph. It provides positive feedback when the users behavior matches steps in the graph, and negative feedback otherwise [14].

2.4 Cognitive Tutoring in Bioinformatics and the Gene Adjacency Program

Bioinformatics can be described as a discipline that integrates computers, software tools, and databases in order to address biological questions. According to Makolo and Raphael [5], visualization models are critical to understanding and making sense of big and complex data generated from genomic research. The Gene Adjacency Program is a model for the visualization of neighbourhood genes and their representation as binary codes. This concept of using binary code for modelling is derived from computational thinking techniques which simulates problems using computer logic of applying abstraction and pattern matching to extract hidden patterns aimed at knowledge discovery. The binary representation enables easy pattern matching of the different gene component and the comparative analysis of multiple genomes and prediction of transcriptional units which are the basis of biomolecular network or biosynthetic pathways. We will embed a cognitive tutor to this tool for our present research.

At present, the most striking attempt at applying cognitive tutoring in bioinformatics is the CTAT Genetics which contained a total of 12 lessons developed for the SimBioSys genetics tutor project at CMU, developed primarily by Albert Corbett, Benjamin MacLaren and Linda Kauffman.

2.5 Limitations of the Cognitive Tutor

Despite the commercial successes of Cognitive Tutor, Aleven [15] identified a design limitation associated with it. Its complexity demands that designers spend 100 of hours per instructional hour to create the program. Furthermore, Cognitive Tutors may not account for the flexible, complex and diverse ways humans create knowledge as cognitive model is based on assumptions about how learning occurs which dictates the chosen instructional methods such as hints, directions and timing of the tutoring prompts. Thus, human tutors may outperform Cognitive Tutor if it provides a higher level of responsiveness to student errors as they are capable of providing more effective feedback and scaffolding to learners than Cognitive Tutors. This indicates that the cognitive model may still be incomplete but has the potential for massive improvement [16].

3. Methodology

Most users of bioinformatics tools depend on human tutors and/or technical documentation for understanding the flow and how to use the tools. However, with the proliferation of bioinformatics tools [8], human tutors are no longer sufficient to properly tutor the large number of researchers and biologists who are interested in using the tools for research and other breakthrough findings.

Another challenge with human tutors is the idea that they may not fully cover the overall features user. The problem of miscommunication by human tutors is also a difficulty [9].

The proposed system involves the implementation of a cognitive tutor on the Gene Adjacency Program [5]. The implementation of a cognitive tutor for bioinformatics tools is very important because it ensures learning by doing as hints are provided based on current learning challenge and next steps. It would also help researchers without technical skills to efficiently use the program, thereby eliminating or at least significantly reducing errors resulting from improper use of the program.

The design objectives include:
- Designing a host user interface for the tutor in the bioinformatics application.
- Configuring the Cognitive Tutor that would determine the flow of tutoring.
- Setting up the cognitive tutor on the selected bioinformatics tool.

of the tool at every teaching instance and may not be readily available at the convenience of the
- Providing clear and unambiguous hints to users where necessary.
- Testing the application with real data and users to ensure that it guides users to carry out the primary function of the host tool.

3.1 Software Requirements

The Cognitive Tutor Authoring Tool⊗R is platform independent, hence it can be configured on Mac OSX, Windows and Linux distributions. MAMP Server (Mac OSX, Apache, MySQL and PHP Server) was used on Mac OSX to process all requests and display results. The system was developed using HTML5 (Hypertext Markup Language) for building the web interface, JavaScript for embedding the cognitive tutor, Object Oriented PHP for coding the system logic and XML (Extensible Markup Language) for extending the Behaviour Recorder File (.brd) needed by the cognitive tutor.
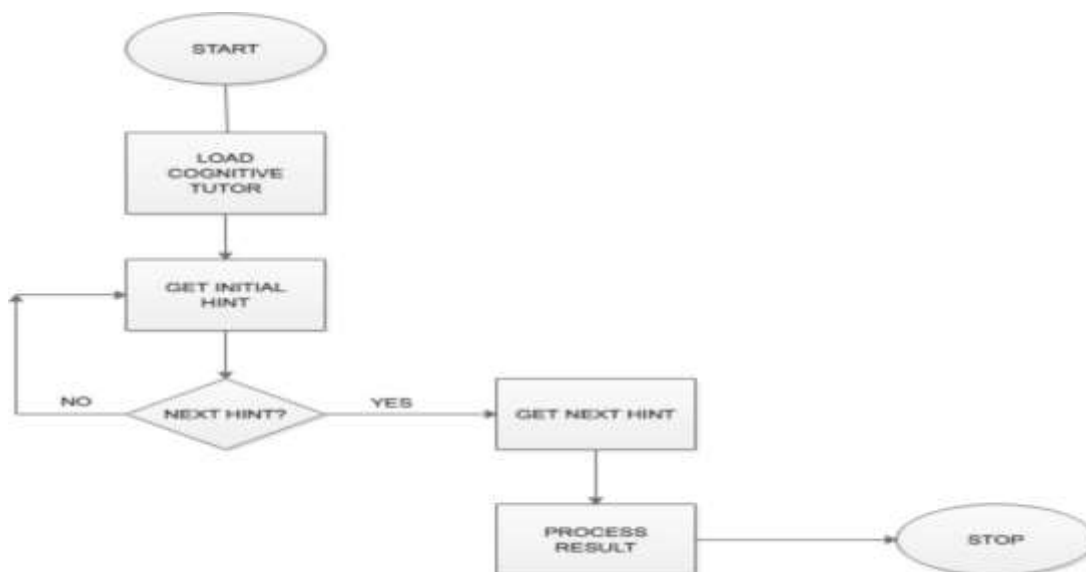

Fig. 2. Flow diagram showing how to use the cognitive tutor.

3.2 How to Use the Program

1) *Load the application which activates the cognitive tutor*
2) *Click on the HINT button to get the first instruction from the tutor.*
3) *Follow the instruction by clicking on CHOOSE FILE button of the Gene Adjacency Program and selecting your downloaded RefSeq file.*
4) *After selecting the file, click on NEXT pointer to get next instructions. At this point, the PREV pointer is activated to allow user go over previous steps.*
5) *Click on the PROCESS FILES button to view and download processing result.*
6) *Click DONE to complete the tutoring process and get your results.*

4. Results and Discussion

On launching the application, the first interface displays the cognitive tutor. At this point, it loads the logic in the Behaviour Recorder File onto the index web interface of the Gene Adjacency Program. After successfully loading the tutor, the homepage displays a brief textual explanation of what the program is about and an instruction to follow the Cognitive Tutor for directions on how to use the program. The user clicks on the HINT button to get the first learning instruction from the tutor. The user then follows the instruction and clicks on the CHOOSE FILE button of the Gene Adjacency Program and selects the file. It is expected that the user has downloaded a RefSeq file with the extension RefSeq.cds.tab that will be processed by the application from [6].

After selecting the desired file, click on the NEXT button on the cognitive tutor to get the next learning instruction. If the user wishes to process more than one RefSeq file, click on the ADD FILE button and repeat Step 3 to add a new file for processing, else, the user can click on the NEXT button to proceed.
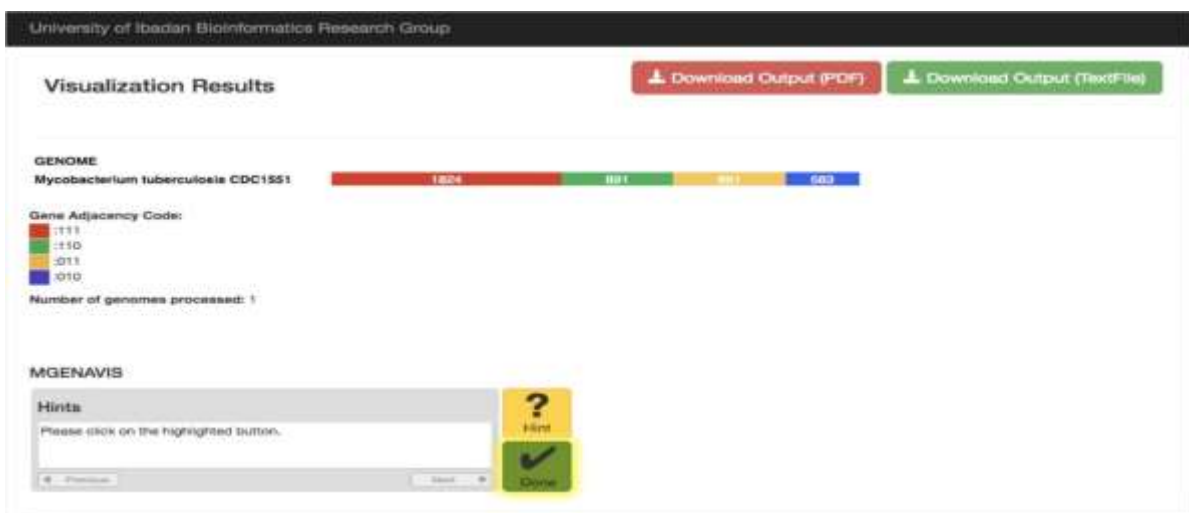

Fig. 3. User interface of the Gene Adjacency Program showing the cognitive tutor in the bottom left.

Fig. 4. The cognitive tutor displaying learning instructions to the user.

Notice that the PREVIOUS button is activated to allow the user go over previous steps either to correct mistakes or to confirm mastery of the steps. Click the NEXT button for more learning hints. Then click on the PROCESS FILES button to view processing result. Notice that you can download result in PDF or txt formats. Click on the DONE button to complete the tutoring session.

a) Preliminary Evaluation: To preliminarily evaluate our system, we carry out a small user testing experiment with 6 bioinformatics researchers having varying experience levels and all having no prior experience using the Gene Adjacency Program. The group was divided into 2 subgroups: test and control. Each member of the test group was given the application with the cognitive tutor embedded while each member of the control group was given the application without the cognitive tutor.

By observation and oral feedback, we find that the test group showed higher levels of competence and efficiency using the system without the need of an expert tutor compared to the control group. While the control group users needed human help to navigate the program and took longer times to get results, the test group were satisfied with the system and confirmed that the cognitive tutor improved their ability to use the Gene Adjacency Program; and that the learning instructions were simple and direct, allowing them navigate through the application with ease.

b) Future Work: The result of this project is of immense use to Bioinformaticians and biologists and it has a large capability for expansion. Further expansions could be made such as:
• Using the cognitive model instead of example-tracing models in order to cover a much broader scope of bioinformatics tools and use cases.
• Data analysis of user performance and experience should be made in order to appropriately and statistically quantify the overall impact of the system.

5. Conclusion

In this work, we focused on exploring the concept of cognitive tutoring and how it can be applied to better use and understand bioinformatics tools. We showed that cognitive tutors, which have been previously applied in other fields such as Mathematics, Economics and Programming, could be applied in Bioinformatics learning and research with considerably large gains. We implemented this by applying a cognitive tutor to the Gene Adjacency Program developed by the University of Ibadan Bioinformatics Group and demonstrating how tutors can be useful in helping non-technical users adapt to new bioinformatics tools and technology by providing intelligent hints and learning instructions.

References

[1] Chi, Michelene TH, Stephanie A. Siler, and Heisawn Jeong. "Can tutors monitor students' understanding accurately?." Cognition and instruction 22, no. 3 (2004): 363-387.

[2] Chen, Min, Francisco Herrera, and Kai Hwang. "Cognitive computing: architecture, technologies and intelligent applications." IEEE Access 6 (2018): 19774-19783.

[3] VanLehn, Kurt. "The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems." Educational Psychologist 46, no. 4 (2011): 197-221.

[4] Aleven, Vincent, Bruce M. McLaren, Jonathan Sewall, and Kenneth R. Koedinger. "The cognitive tutor authoring tools (CTAT): Preliminary evaluation of efficiency gains." In International Conference on Intelligent Tutoring Systems, pp. 61-70. Springer, Berlin, Heidelberg, 2006.

[5] Makolo, Angela, and Raphael Isokpehi. "Interactive Visual Representations of Gene Transcriptional Direction Patterns in Microbial Genomes." In Proceedings of European Molecular Biology Organization Conference on Visualization of Biological Data, VISBI, 2015.

[6] URL:http://brcdownloads.patricbrc.org/patric2/genomes

[7] Blessing, Stephen B. "The cognitive tutor: successful application of cognitive science." (2011).

[8] Kanehisa, Minoru, and Peer Bork. "Bioinformatics in the post-sequence era." Nature genetics 33 (2003): 305.

[9] Cummings, Michael P., and Glena G. Temple. "Broader incorporation of bioinformatics in education: opportunities and challenges." Briefings in bioinformatics 11, no. 6 (2010): 537-543.

[10] Wang, Yingxu, and Vincent Chiew. "On the cognitive process of human problem solving." Cognitive systems research 11, No. 1 (2010): 81-92.

[11] Gutierrez-Garcia, J. Octavio, and Emmanuel Lpez-Neri. "Cognitive computing: A brief survey and open research challenges." In Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence (ACIT-CSI), 2015 3rd International Conference on, pp. 328-333. IEEE, 2015.

[12] Joseph Psotka, Sharon A. Mutter (1988). Intelligent Tutoring Systems: Lessons Learned. Lawrence Erlbaum Associates. ISBN 0-8058-0192-8.

[13] Anderson, John R., Albert T. Corbett, Kenneth R. Koedinger, and Ray Pelletier. "Cognitive tutors: Lessons learned." The journal of the learning sciences 4, no. 2 (1995): 167-207.

[14] Koedinger, Kenneth R., John R. Anderson, William H. Hadley, and Mary A. Mark. "Intelligent tutoring goes to school in the big city." International Journal of Artificial Intelligence in Education (IJAIED) 8 (1997): 30-43.

[15] Aleven, Vincent. "Rule-based cognitive modeling for intelligent tutoring systems." In Advances in intelligent tutoring systems, pp. 33-62. Springer, Berlin, Heidelberg, 2010.

[16] Scandura, Joseph M. "What TutorIT Can Do Better Than a Human and Why: Now and in the Future." Technology, Instruction, Cognition and Learning (TICL) 8, no. 53 (2011): 175-227.