Opening a Window in the Black Box: Improving bioinformatics tools by exposing their innards to biologists

Erik Pukinskis
Indiana University
epukinsk@indiana.edu

ABSTRACT
The design process is described for Underbelly, a sequence alignment tool for biologists that allows them to explore how the Smith-Waterman algorithm works. Interviews were conducted with 2-3 biologists in each of six labs, which showed that while computational tools are common, few biologists know how they work, which can cause errors in their use. A paper prototype, and later a fully interactive prototype, were developed. The interactive prototype was tested with six biologists. The prototype piqued their curiosity, allowed them to answer their own questions, and helped them think critically about the algorithm and the software they use.

INTRODUCTION
As the field of Bioinformatics expands, large numbers of computational tools are being developed for a diversity of Biological pursuits. In areas like Evolutionary Biology where research goals have a strong genomic component, the ability to use computational tools for database searching, sequence alignment and phylogeny has become an all but requisite skill for biologists. However, it seems that biologists see computational tools as peripheral to their work--a means to an end. In most cases, biologists are only motivated to learn "just enough" about a computational tool to perform the operations necessary to get the results they need.

In most situations this works well, but in some cases incorrect use of computational tools can lead to erroneous results. Since the emergence of large, unified databases like NCBI's GenBank (Benson et al, 2005) or the Gene Ontology (Ashburner et al, 2000), these results often enter, unvetted, into a larger pool. Once in these databases, other researchers infer other relationships from the bad data, causing mistakes to be amplified.

The volume of results in these databases which are derived from experiments conducted entirely in silico can be quite large. In the case of the Gene Ontology, 95% of the term assignments were inferred from sequence similarity or other computational analyses (personal correspondence). This volume of computationally-derived annotations demonstrates the massive potential for erroneous reporting. [BAD]

Additionally, these errors can sometimes slip through the peer review process. Journal submissions often come with "60-plus pages of supplementary data", while accompanying data sets "may contain millions of discrete data". (Nicholson, 2006) This volume of experimental data make it difficult for reviewers to assess the validity of computational results.

Lastly, biologists seem to be highly constrained by the limitations of their knowledge of computational tools. It can take years for biologists to identify an optimal toolchain and to learn to use it to its fullest extent, if they reach that point at all. Typically, biologists simply make do with what they have: a handful of tools and a subset of their capabilities that were introduced to them through their social network.

Taken together, there is clearly a strong need to reduce errors in the use of Bioinformatics tools. Our research is focused on improvement biologists' understanding of how their software tools work.

METHOD
We began our work by conducting in-depth interviews with biologists and observations of their work as a form of contextual inquiry. The study was conducted in six labs with a range of research focuses. 2-3 individuals were interviewed in each lab, who were either graduate students, postdocs, or research associated. Participants were asked open-ended questions about the biological aims of their research, the computational tools they used, how they used them, what parameters they manipulate, their understanding of how their tools work, and how their educational and research history has formed their understanding of their tools. In addition, they were asked to demonstrate common tasks that they perform.

STUDY
While there were differences in what software was used and how, there was much in common between the relationship between biologists and their software. When asked how their software works, the biologists interviewed invariably described the end-product of the software. None of the participants knew more than the most basic details about how their software actually carries out their tasks. Often, participants didn't know why they were carrying out certain tasks.

For example, none of the three participants who used Mega knew what bootstrapping was or what the difference was between different bootstrapping methods. Another researcher who uses ARB did not know the difference between the different models it uses, although she knew that it was important that the right model was chosen for each sequence. She saw this automatic selection feature as a selling point of ARB. All of the participants seemed to feel that the fewer parameters they had to provide, the better.

Three common approaches to learning how to use their software were observed. In many cases, the researcher would simply ask their lab partners how they use the tools and then imitate that workflow. In other instances, researchers tried multiple parameters and then compare the results. If the results appeared identical, the biologist assumed that the parameters did not matter, and that is was safe to choose either. Worst of all, in some cases biologists will try multiple methods for achieving some output, compare the results, and then choose the result that looks most correct to them.

Each of these approaches is unsettling. Imitating colleagues' practices could lead to perpetuation of bad practices. And conducting limited explorations of the effects of different parameters is a dangerous strategy. While a parameter might have no effect on one dataset, it might have a transformative effect on another. And looking at multiple results and choosing the result which appears most correct is certainly not a scientifically valid approach.

Another finding was that although colleagues often shared knowledge about commonly used tools like BLAST, in two cases interviewees indicated that for a particular less common tool or method, they felt they were the only member of the lab, or sometimes the only person in the university who knew how to use that tool. In these cases, not even the professor running the lab knew how to use the software tools in question.

All of these findings seem to indicate that biologists almost always treat their software tools as "black-boxes". They understand what goes in and what comes out, but not what happens in between. These practices clearly raise ethical questions about scientists' responsibility in reporting computational results, but it also represents an opportunity for exploring ways to help biologists to become more familiar with how their software works. These observations seem to suggest that there is room for improvement in the way Bioinformatics software provides learning opportunities for biologists.

It was also observed that in some cases biologists were reticent to take advantage of the documentation provided with software. One interviewee walked through the process of doing a BLAST search, and when asked if the help was helpful, opened it up and indicated that he did not understand any of the information that provided. When we discovered that there were a variety of interactive tools that demonstrate the sequence alignment process (BiBiServ, 2006, Setoft, 1999, Sumazin, 2003, etc) we began asking researchers to attempt to learn something from one of them--BiBiServ's Sequence Alignment Applet. Three of the four researchers who did this were unable to extract any meaningful information from the application except to identify the two sequences being aligned. The fourth took it as a challenge to decipher the applet, and was able to discern that the scores represented some kind of match and that the algorithm was searching for "diagonals" but was not able to decipher how it would do that.

The difficulty in both of these cases seems to be that help documentation is often written in language and presentations that biologists don't understand. As a result, the time investment required for biologists to learn a small amount about the software they use can be prohibitive. It seems that because biologists are committed first to solving biological problems and only secondarily to solving computer problems, they sometimes avoid learning unnecessary details about their tools.

REQUIREMENTS AND DESIGN PROCESS

Our goals was to design a new tool that would provide immediate educational payoff to biologists who want to invest a small amount of time into learning about the software, and to encourage and support critical thinking about the function of software. Based on our contextual inquiry and discussions with bioinformaticists, we identified Sequence Alignment as an important computational task in biology that is simple enough for anyone to understand. Several possible approaches were explored as sketches, and one of these was developed into a paper prototype.

The paper prototype represented a sequence alignment tool using the Smith-Waterman algorithm (Smith and Waterman, 1981) taking a unique approach to bioinformatics education. Rather than providing external documentation or separate educational materials, it exposed the inner workings of the sequence alignment process to biologists. It did this by providing representations of the data that the algorithm manipulates, a pseudocode version of the algorithm itself, and references between the code and the data. By using a step forward button, users could walk through the execution of the algorithm step by step.

To get an indication of whether this prototype could provide some educational benefit, a pilot study was conducted with two bioinformaticists and a biologist. The paper prototype was explained to each participant, and they were invited to try to use it and then offer feedback. It became immediately clear that users would need a greater amount of interactivity than what the paper prototype afforded, so we went on to develop a fully interactive prototype that provided the ability to navigate forward and backward through the code and watch results being calculated. (Figure 1) This interactive application was named "Underbelly" because it allows biologists to explore functional parts of the software normally hidden from view.

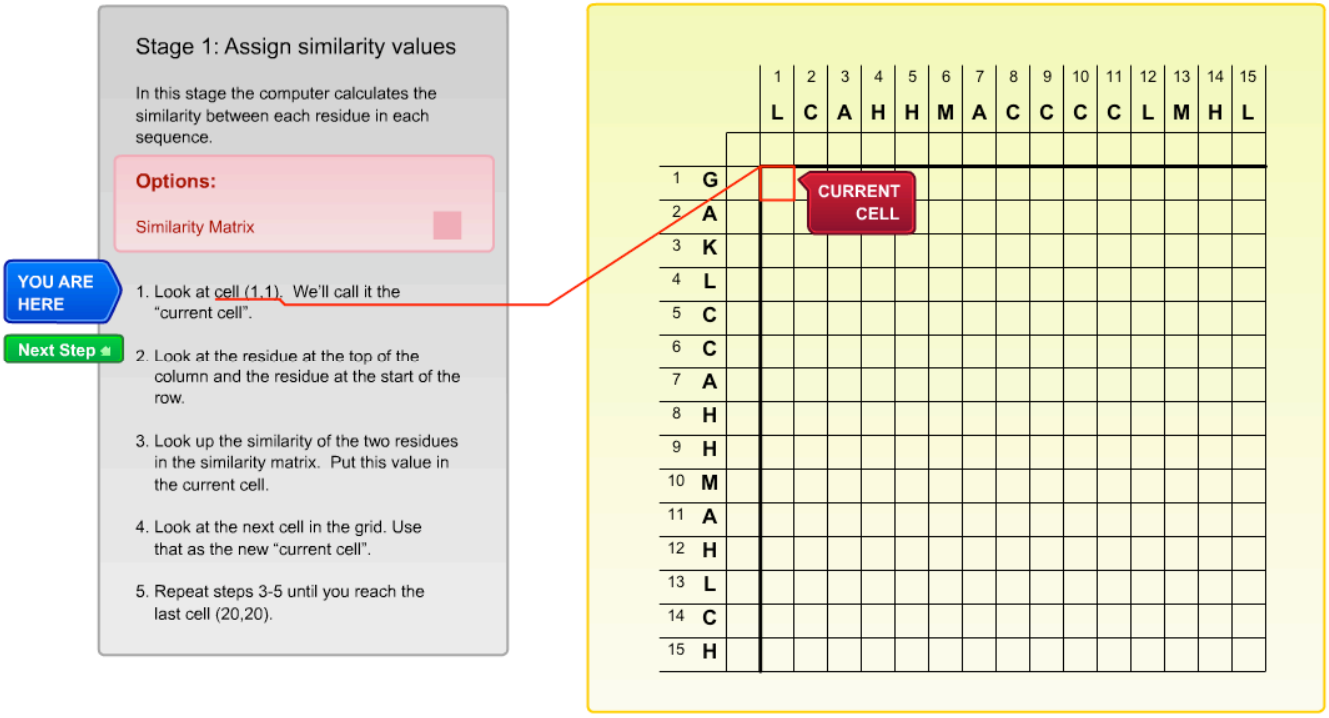

Figure 1: First interactive Underbelly prototype

Finally, a third prototype (Figure 2a) was developed after further user testing integrated this experience into the sequence alignment process, automatically "playing" the algorithm as results are being calculated and offering the ability to pause the action (Figure 2b), and to observe the algorithm at multiple levels of detail (Figure 3).

There are five critical aspects to the design of Underbelly:

First, Underbelly makes exposure to the inner workings of the algorithm unavoidable. The biologist is free to ignore what is happening onscreen between the time they enter their data and the time they see the results, but they must at least register the fact that some calculation is happening and that it is at least somewhat accessible to them. The interface is presented in as friendly a way as possible, so that biologists can see how easy it is to step inside the algorithm and see what is happening.

Second, the algorithm is presented at multiple levels of granularity. By default, the algorithm is represented in high-level pseudocode (Figure 3) which is readily understandable by biologists. No initiative is required to read this overview while the user is waiting for the algorithm to do its work. However, if the user's curiosity is piqued, they can click the "Show me" button next one of the stages, and a more detailed representation of that particular subroutine will be shown (Figure 3). In this way, Underbelly offers both a very low barrier to entry, and the opportunity for a very deep kind of educational exploration not currently afforded by most documentation.

The third key element of our design is that the user can explore the execution space of the algorithm very freely. Not only can they play and pause the algorithm, they can step forward and backward through it, jump to specific points in their data, and move between different stages of

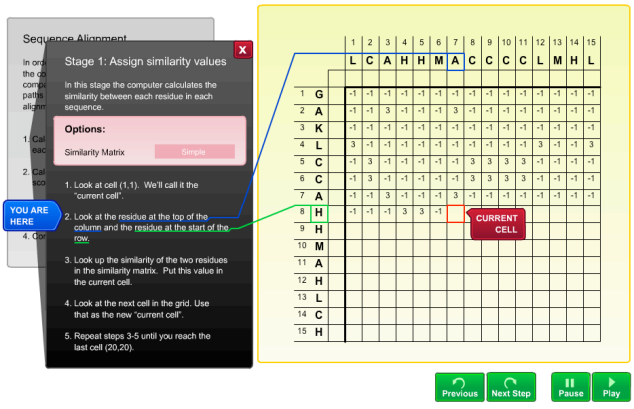

Figure 2a: Second interactive Underbelly prototype

Figure 2b: Code playback controls

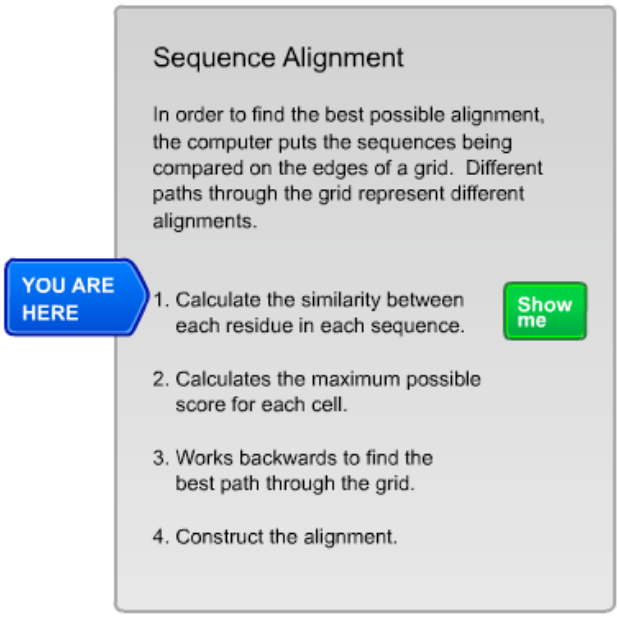

Figure 3: Pseudocode

the algorithm. Eventually, we would like to provide the opportunity for users to modify the data being manipulated and watch results change.

It is also important to note that Underbelly works with a user's own, real data. Although the current implementation is a research prototype, and thus somewhat simplified, it is not meant to be a toy application used only for educational purposes. It is indented to be a real, usable application that provides integrated learning and exploration. It is our hope that such a configuration will overcome the barrier to entry problems of current documentation and educational materials.

Lastly, Underbelly does not provide any representations that are unique to the sequence alignment process. Many of the existing interactive sequence alignment demonstration programs represent the sequence alignment process with a special visual language, representing the grid as a field of arrows, for example[1]. We wanted to design Underbelly in such a way that it might feasibly be generated automatically from code that had been annotated with pseudocode descriptions and some hints about how to break it into meaningful steps. We are proposing Underbelly as a template for a new way of presenting software to biologists, and as such we wanted to make it as generic as possible. Generating an Underbelly-like application from source code is a non-trivial task, but we wanted to make sure that it was at least possible in theory. Future work will explore what would be necessary to make this happen.

USER STUDY

Versions of the interactive prototype were evaluated with six biologists, ranging from masters students in biology to research associates and postdocs. Participants were shown how the Underbelly works and given the opportunity to explore freely. As they explored they were asked to use the Thinkaloud technique (Van Someren et al, 1994) to describe their thoughts. The data collected was largely qualitative. Users were asked to speak aloud their immediate reactions, to describe what the algorithm was doing, and to speculate about what they might use the software for. When users would have a difficult time understanding certain concepts, their thoughts would be documented and then the area of confusion would be explained to them. This allowed us to continue to observe their explorations, which would otherwise be thwarted by misunderstanding. Each user spent 20-30 minutes with the software.

FINDINGS

The main findings of the study were threefold. First, roughly half of the participants were very curious about Underbelly and how the sequence alignment was working, and these participants were able to attain a decent understanding of the algorithm in the 20-30 minutes they spent with it. Second, the extent of participants curiosity seemed to depend on the depth of their previous experience with sequence alignment tools. Lastly, in two cases the tool supported spontaneous critical thinking about the function of the software and whether it was the appropriate tool for specific research tasks.

Although none of the participants knew what to expect before seeing Underbelly, half of the participants tested became highly curious about the algorithm when they realize that Underbelly made it accessible to them. Two masters students in particular had done a significant amount of sequence alignment and genomic search and seemed to be genuinely fascinated by Underbelly. They had become somewhat familiar with the idea of the sequence alignment process, but they had resigned themselves to the idea that only computer scientists and bioinformaticists would be allowed to understand how it works. Realizing that they could actually peer into that box that had been permanently labelled "off-limits" seemed to excite them. Two older postdocs who had not done much sequence alignment, or had only done very simple alignments found it to be largely irrelevant to their goals as a biologists.

There were two clear examples of participants engaging in fairly deep critical thinking about the alignment algorithm without prompting. One research associate became concerned when he started to understand how simple the sequence alignment algorithm actually is. He said he trusted it less now that he knew how simple it was. He then started pointing at the sequences, and said he would like to be able to change the sequences and see what would happen. He asked what would happen if you ran the algorithm with one sequence and a second sequence that was exactly the same except that it had the first half swapped with the second half. He felt that kind of swapping was biologically feasible, but he seemed to have

an intuition that the algorithm would not handle that situation gracefully. In fact, he was right. The Smith-Waterman algorithm would give that sequence a rather low score, because only half of the sequence would be able to match.

A second participant spent about ten minutes exploring the algorithm and then started describing a new kind of tool that would allow him to choose between different alignments for a different parts of a sequence: "I wonder: why there isn't a program that could have a window, and you could click.... if you don't like the way it's lined up, it would give you several options, and then score values for that region." This kind of critical thinking about what the software is doing and what is possible is exactly the kind of thinking described in the project's design goals.

What remains to be seen is whether biologists would actually invest the time in working with Underbelly. In our user tests, they were asked to explore the software and given support. In some cases, users ran developed misunderstandings that they were unable to overcome. For example, one user developed the conviction that the Maximum Match score was an indicator of similarity between two residues. She could see that this was not the case, but she was unable to figure out why. It is not clear how Underbelly could be improved in this situation.

CONCLUSION

The user study showed that tools like Underbelly can provide learning opportunities to biologists that were previously unavailable, and it lets them do so quickly, with a small amount of effort. It uses a set of innovative methods for visualizing software in a way that is attractive to users, and has the potential to make a contribution towards solving a serious educational problem in biology.

REFERENCES

Ashburner,M., Ball,C.A., Blake,J.A., Botstein,D., Butler,H., Cherry,J.M., Davis,A.P., Dolinski,K., Dwight,S.S., Eppig,J.T. et al. (2000) Gene Ontology: tool for the unification of biology. Nature Genet., 25, 25–29.

Benson DA, Karsch-Mizrachi I, Lipman DJ, Ostell J, Wheeler DL. (2005) GenBank. Nucleic Acids Res. Jan 1;33(Database issue):D34-8.

BiBiServ (2006) Sequence Alignment Applet. http://bibiserv.techfak.uni-bielefeld.de/media/seqanalysis/align-applet.html

Nicholson, Jeremy K. (2006) Reviewers peering from under a pile of 'omics' data. Nature 440, 992 (20 April 2006)

Setoft, Peter (1999) Java biosequence alignment applet. http://www.dina.kvl.dk/~sestoft/bsa/bsapplet.html

Smith, TF, and Waterman, MS, (1981) Identification of Common Molecular Subsequences, Journal of Molecular Biology, 147:195-197, 1981.

Sumazin, Pavel (2003) Computing Optimal Alignment. http://web.cecs.pdx.edu/~ps/CapStone03/dynvis/SimilarityApplet.html

Van Someren, M. W., Barnard, Y.F., & Sandberg, J.A.C. (1994). The Think Aloud Method: A Practical Guide to Modeling Cognitive Processes. London: Academic Press.