Jl. of Computers in Mathematics and Science Teaching (2009) **28**(4), 405-421

Bringing Research Tools Into the Classroom

CHARLES SHUBERT, IVICA CERAJ AND JUSTIN RILEY
Massachusetts Institute of Technology, USA
cshubert@mit.edu
ceraj@mit.edu
jtriley@mit.edu

The advancement of computer technology used for research is creating the need to change the way classes are taught in higher education. "Bringing Research Tools into the Classroom" has become a major focus for the Software Tools for Academics and Researchers (STAR) team at MIT. The STAR developers collaborate with faculty researchers to help solve the problems of moving computational tools used for research into the classroom. In the first two years of the program's existence, the STAR team has implemented software to bring high-performance computing (HPC) into material modeling courses, built software for protein structure visualization and hydrological analysis of watersheds, repackaged genomics software for analyzing microarray data, and provided the development environment and computational resources necessary for teaching parallel programming. The goal of this work is to make research software and data easily accessible to students and their faculty in a package that not only promotes use in the classroom, but also creates a path for unscripted discovery. This paper presents five STAR software packages that are currently being used in MIT classrooms. A STAR software package is considered successful if it enables bringing new topics into the classroom that previously could not be taught.

The rapid expansion of research data and the corresponding need for tools that facilitate the understanding of this data are having a significant impact on how contemporary research is being conducted. There is a desire in the educational community to bring students closer to contemporary research. Finding the solutions to the practical problems of bringing research closer to the students is the work of the Software Tools for Academics and Researchers (STAR) group of the Office of Educational Innovation and Technology (OEIT) at the Massachusetts Institute of Technology.

While there is much educational software available, it is mostly focused on course management or mastering basic material. The software designed to bring the student into a deeper understanding of the contemporary topics in a research domain is often too arcane to be used in the classroom. The STAR program seeks to make research software easier to use, easier to access, scalable for peak-period classroom usage such as the night before the homework is due, and to have a sustainable support infrastructure that extends beyond the authors. STAR developers start this process by talking to researchers and asking the question, “What prevents you from bringing your research tools into the classroom?” Every researcher we encounter has no difficulty answering this question.

Initially, we concentrated on making computational resources available to students in the classroom. We discovered in the process of running the research software in a classroom setting that there were a number of operational inefficiencies associated with getting the software to run at all. A significant percentage of hands-on time in the classroom was spent getting UNIX command line options correct, modifying shell scripts, and navigating arcane user interfaces. It became clear to us that research software in the classroom needed to be easier to use.

Currently, the STAR program focuses on five areas of contemporary computation in research. We use StarGP to remove the operational fog of running UNIX command line programs on high performance computing clusters, StarBiochem to remove the complexities of using protein structure visualization tools, StarBiogene to investigate genomics data, StarHydro to explore watersheds using elevation map data, and StarHPC to develop and run parallel programs on high performance computing clusters. Each of these STAR software packages is covered in more detail later in this paper.

For this effort to be truly successful, collaboration between software developers and researchers is essential. We have collaborated with:

- The Computational Biology and Bioinformatics group at the Broad Institute of MIT and Harvard to bring their software into both Biology and Materials Science modeling courses. In our collaborations with the Broad Institute developers, we adapted the GenePattern (Reich et al., 2006) work flow software for use in a cross discipline course on “Introduction to Modeling and Simulation” (Yip et al.,

2002) taught by Professors Nicola Marzari, Markus Buehler, Raul Radovitzky, and Dr. Timo Thonhauser.

- MIT Physics Professor John Belcher to leverage his TEALsim (Belcher, 2003) physics simulation and visualization software for protein visualization. This work was for the Introductory Biology courses taught by MIT Biology Professor Graham Walker and his colleague, Professor Melissa Kosinski-Collins who is now a Professor of Biology at Brandeis University (Chisholm, Walker, Khodor, & Mischke, 2005).
- MIT Hydrology Professors Rafael Bras and Dara Entekhabi to develop software for the undergraduate Hydrology courses.
- The Broad Institute developers to adapt various genomics software from the Broad Institute for MIT's Introduction to Biology courses.
- MIT Earth, Atmospheric, and Planetary Sciences (EAPS) Research Scientist Constantinos Evangelinos to develop an on-demand high performance computing environment for teaching parallel programming.

An interesting side effect of this work has been the cross-pollination of useful software from one domain to other research and educational domains.

STARGP: A FRONT-END FOR HIGH PERFORMANCE COMPUTING CLUSTERS

StarGP is the repackaging of a scientific workflow engine provided by the Broad Institute of Harvard and MIT. It is used for running a wide variety of command line based research software on remote high-performance clusters. It allows research software to be used in the classroom. Students are able to experiment with research tools facilitating the exploring of both scripted and unscripted questions. StarGP addresses the details of setup, distribution, and management of research software. By reducing the typically steep learning curve for running the software, a researcher can introduce these fascinating learning tools into the classroom.

GenePattern as a scientific work flow front-end

The Bioinformatics group at the Broad Institute of Harvard and MIT faced similar challenges for providing the latest tools and methods to researchers who are trying to analyze the mountains of gene expression data. To address this problem, they developed the GenePattern computational software. GenePattern allows users and administrators to create and run reusable scientific work flows. It provides a user-friendly, form-based interface for wrapping individual software tools into "tasks" and combining these into more complex research work flows, called "pipelines."

We found that the GenePattern application works very well for student access to high-performance computing resources. We adopted this tool and built tasks and pipelines that promote creative use of research tools in the classroom. Instead of learning how to run the research software by typing UNIX command line options and editing UNIX shell scripts, students use a more familiar web interface to invoke research software, set options, and manipulate input and output data.

Electron structure calculations made easy

The course "Introduction to Modeling and Simulation," jointly taught by faculty from seven MIT departments, was chosen as the pilot for this framework. We built a scientific work flow around Quantum-Espresso (Baroni et al., 2005), a set of programs for electronic structure calculations using density-functional theory and density-functional perturbation theory. The GenePattern work flows mimic existing lab exercises and have the goal of minimizing the amount of lab time spent explaining how to use Unix commands and how to modify shell scripts. The work flows also achieve the goal of maximizing the amount of lab time used to investigate scientific concepts taught during the course. As an added bonus, the creation of the work flows made the task of preparing and presenting lab exercises much simpler. The use of this web application as the front-end for the HPC cluster standardizes the system administration of the cluster.

The successful adoption of GenePattern for this course creates a model for the use of HPC computing clusters in other domains, demonstrating the value of applying large-scale computational power to undergraduate and graduate teaching environments.

Results

The classroom pilot for this software was conducted during April, 2006. Results showed that this new web-based system is much easier to use and much more effective in conveying the intent of the research software than the command line based experiments used previously. When compared with earlier non-StarGP lab exercises, students using StarGP were able to obtain an order of magnitude more data in 10 minutes than they were able to obtain in 90 minutes prior to using StarGP.

Demonstrated benefits include:

- Removed barriers to introducing research tools into the classroom
- Improved utilization of limited classroom time
- Enabled exploratory learning
- Introduced students to current research technology
- Encouraged testing hypothesis and comparative exploration
- Controlled software dissemination
- Centralized software upgrades and administration
- Lowered the cost of system administration.

STARBIOCHEM: AN INTUITIVE PROTEIN VISUALIZATION TOOL

StarBiochem is a protein visualization tool developed in Java. The StarBiochem project addresses the difficulty experienced by MIT Professors Graham Walker and Melissa Kosinski-Collins in using existing protein visualization tools in the classroom (unpublished correspondence). The StarBiochem code base is derived from MIT Professor John Belcher's TEALsim simulation and visualization software.

The key enhancements to the user interface are:

- Easier selection of protein structural components
- Easier control of the rendering of selected and unselected components
- Introduction of higher level protein structural elements in the same order in which they would be encountered in the classroom.

These enhancements allow StarBiochem to be used in classroom to visualize and explore the biochemical concepts of protein structure.

StarBiochem's picture is worth a 1000 words

StarBiochem is a tool that creates and controls the rendering of protein structures. It gives the user fine-grained control of the resulting visual experience. Its Java implementation and the TEALsim simulation framework allow it to run on most popular platforms (including Linux, Windows and Mac OS X). StarBiochem can take advantage of the hardware acceleration of a platform's graphics card via Java3D, a Java package for creating three-dimensional visualizations.

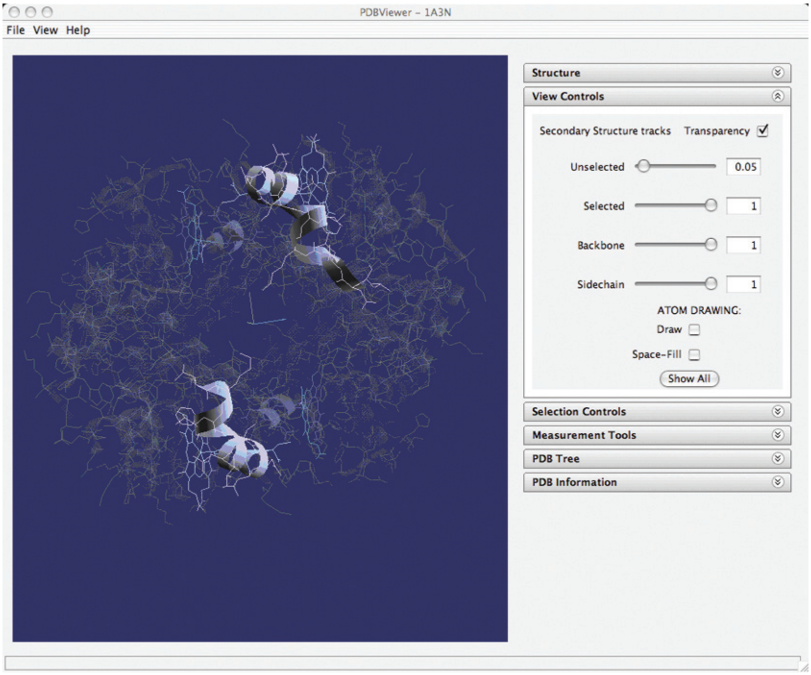
The key concepts of protein structure are the basis of StarBiochem's user interface. Usability and simplicity are achieved by developing novel and flexible methods for selecting elements of protein structure visualization and manipulating their visual state. In StarBiochem, a selection and its surrounding context can be independently emphasized or de-emphasized. This allows the user to create visual snapshots corresponding to protein structures of interest. These snapshots help formulate, communicate, and visualize important aspects and properties of a protein's structure. StarBiochem can access the thousands of molecules in the Protein Data Bank (Bernstein et al., 1977). It does this by querying the Protein Data Bank web service. This creates an environment in which up-to-date information is available to the end-user.

How does Heme dock on Hemoglobin?

The test bed for StarBiochem in the classroom setting was the course „Introductory Biology," a requirement for all MIT students. The faculty and instructors used StarBiochem to present complex structural information about proteins to the students in an easier, more understandable, visually rich, and more intuitive way. Students explored a representative set of molecules in a homework problem set having no prior training on the software. StarBiochem was downloaded from the STAR web site by the students as a Java WebStart application.

Students can use this interactive tool to formulate hypotheses. As an example of how StarBiochem might be used, consider the case of a student looking at the molecule 1A3N (Deoxy Human Hemoglobin). Examining the molecule reveals that there are four Heme molecules attached to the Hemoglobin molecule. By displaying the secondary structures, the student can see that each Heme is attached to a Hemoglobin helix motif (Figure 1).

With StarBiochem, it is easy to select the four Heme molecules and their subsequent helices, and to make the unselected part of the Hemoglobin molecule transparent. This ability to select specific molecular components and make the unselected components partially or wholly transparent is unique to the StarBiochem user interface. It allows the student to ask interesting questions and potentially important research questions such as "Why are two of the Heme molecules attached to short straight helices while the other two Heme molecules are attached to longer bent helices?" The questions are formulated because StarBiochem makes it easy to look at very specific structural components of a molecule.



Secondary Structure tracks Transparency ☑

Unselected 0.05

Selected 1

Backbone 1

Sidechain 1

ATOM DRAWING:

Draw □

Space-Fill □

Figure 1. Highlighted Heme's in Deoxy Human Hemoglobin

Results

As of December, 2007, StarBiochem has been successfully used by approximately 1300 MIT students in the Introduction to Biology course, roughly 400 students at Brandeis University, and hundreds of high school students. StarBiochem is freely available on the STAR web site (http://web.mit.edu/star/biochem).

STARHYDRO

StarHydro is a Java application for watershed delineation and analysis. Its intended purpose is to serve as an easy to use, high quality analysis and visualization package for exploring watersheds. It captures the major concepts of the Hydrology research domain and provides an interactive visualization of the watershed map. It allows for manipulation of relevant parameters for the analysis of a watershed, including the stream accumulation threshold and drainage velocities. StarHydro was built to serve not only as a classroom tool to explore the hydrology of a watershed, but also to export the relevant analytical results as well as the original data used to produce them. The next extension to the StarHydro platform is to integrate, render, and analyze the results of the watershed data generated by popular hydrology research software.

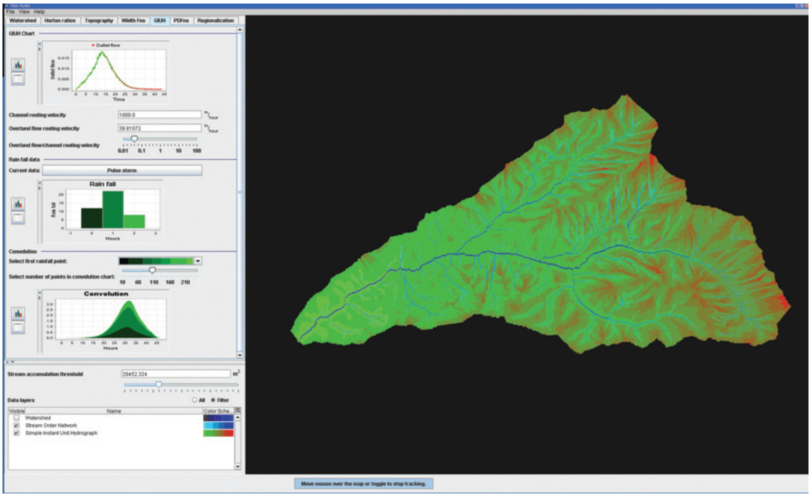
Flood of understanding from convoluted rainfall

Convolution and the distributed nature of real-world problems are two very important concepts that are hard to explain. In the Hydrology course at MIT, students analyzed the experimental watershed named Little Washita. Using StarHydro, students were able to see the watershed color-coded with the run-off travel times and the watershed outlet response to a three hour storm (Figure 2).

StarHydro makes it easy to load different digital elevation maps and analyze a watershed. The analyses available in StarHydro include Horton ratios, hypsometric curve, link concentration function, slope-area relationship, width functions, instant unit hydrograph including storm convolution and others.

Results

The first use of this software was in an undergraduate Hydrology course at MIT in Fall, 2007. It has now become a regular tool used in the course. The course curriculum will be extended to include distributed hydrology that was previously too difficult to teach. There is additional interest in using the software at other institutions and for high school environmental courses. StarHydro is freely available on the STAR web site (http://web.mit.edu/star/hydro).



Channel routing velocity 1000.0 m/hour

Overland flow routing velocity 39.81072 m/hour

Overland flow/channel routing velocity

Current data: Pulse storm

Select first rainfall point:

Select number of points in convolution chart:

Stream accumulation threshold 29452.324 m²

Data layers ○ All ● Filter

Visible	Name	Color Sche...
□	Watershed	
☑	Stream Order Network	
☑	Simple Instant Unit Hydrograph	

Figure 2. Rainfall run-off travel times in Little Washita

STARBIOGENE

StarBiogene is a repackaging of a set of visual and analytic software tools, developed by the Bioinformatics Group at the Broad Institute of MIT and Harvard, for the analysis of genomic gene expression data generated through microarray analysis. This software allows students to compare the expression of multiple genes across a range of biological samples (cell lines, tumors, etc) via a centralized web interface. It allows students to process and explore various microarray datasets and also comes prepackaged with the data sets of a seminal paper (Golub et al., 1999) in cancer genomics.

Analysis and Data Visualization

StarBiogene currently consists of a set of visual and analytic software tools for the analysis of microarray gene expression data. Analytic "pipelines," or work flows, were created around these tools to allow cancer datasets to be analyzed using the provided data visualizers at the press of a button. To view a dataset, students choose one of these pipelines and simply upload their data file. StarBiogene takes care of processing and viewing the data.

Students examine the data by using one of the following data visualizers:

- *Original data viewer:* presents the data to the student in its original form. It allows the student to examine the data before processing.
- *Microarray viewer:* analyzes a microarray dataset and determines each gene‘s correlation to a type of cancer. The genes are sorted by their correlation score and plotted on a graph. A spreadsheet listing the genes in the dataset, their correlation scores, and other relevant information is shown below the plot.
- *Heatmap viewer:* analyzes a microarray dataset and visualizes it in the form of a heatmap. Using this view, students are able to look at the expression of multiple genes across a range of samples simultaneously.
- *Clustering viewer:* analyzes a microarray dataset in a similar fashion to the heatmap but it additionally clusters the samples according to their similarity in gene expression. The clustering view of the data exposes cases where a sample may be correlated to a specific cancer type but is more similar in gene expression to samples of another cancer type.

Collaboration with the Broad Institute of MIT and Harvard

One of the primary goals of StarBiogene is to make the software both easy to use and accessible. In collaboration with Dr. Megan Rokop of the Outreach Program at the Broad Institute, the STAR group assessed the usability issues associated with the data visualizers included in StarBiogene. In order to bring StarBiogene into the classroom, it was necessary to make modifications to both the GenePattern web interface and the data visualizers to create a more user-friendly experience. For the data visualizers, this required reorganizing various configuration menus to provide a more intuitive user interface as well as adding additional preferences and searching capabilities to the data visualizers. For the GenePattern web interface, modifications were needed to support approximately 400 students using the software the night before the homework was due.

In collaboration with the GenePattern software developers at the Broad Institute, the STAR group implemented these changes for the Introduction to Biology courses at MIT. These changes were then incorporated into the next release of GenePattern. As a result, StarBiogene provides an example of how the educational usage of research software ultimately serves to improve existing research software.

How do we judge which type of leukemia a patient has?

The problem set provided with StarBiogene guides students through the analysis of a set of tumor samples in relation to two forms of leukemia: ALL and AML. Since the tumor type of these two forms is indistinguishable when viewed through a microscope, the software is used to determine the type of tumor a given patient has. This analysis is important, as one tumor type will respond to a particular treatment while the other won't. Furthermore, giving a patient the wrong treatment could be fatal.

Results

StarBiogene was used for the first time during Fall, 2007, with approximately 300 students in MIT's Introduction to Biology course. Our usage statistics indicate that StarBiogene had a maximum load of 160 students the night before the homework was due. StarBiogene can be accessed via the STAR web site (http://web.mit.edu/star/biogene).

STARHPC

StarHPC is a cloud computing project that significantly lowers barriers for teaching parallel programming in the classroom. StarHPC provides a dynamic, on-demand, remote compute cluster for developing parallel programs in both OpenMP (Multi-Processing) and OpenMPI (Message Passing Interface) frameworks, and also provides the development environment and computational resources necessary to help teach students the concepts of parallel computing. StarHPC can be used in advanced computer science courses to teach students about the challenges of parallel programming. It also serves as the STAR group‘s first look at using Amazon‘s Elastic Compute Cloud (EC2) web service as a solution to the issues associated with bringing computational resources into the classroom.

Bringing computational resources into the classroom

In general, owning computational resources is costly and involves a lot of overhead. In addition to the servers, there are costs associated with housing, powering, cooling, and administering a traditional compute cluster.

Aside from the cost, it is an operational challenge to deliver a traditional compute cluster in a classroom setting. Setting up user accounts, providing remote access outside the class, and configuring the various software packages needed can prove to be formidable obstacles. Unless there exists a resident expert with the time and expertise to address these issues, it can be difficult for faculty to manage these resources effectively in addition to creating the problem set(s).

Even when these obstacles can be overcome, there is an issue regarding resource utilization. Typically there is a one to two week "peak-period" during the course when the cluster experiences its largest load. Typically, this is when a problem set is due. However, outside of this peak-period usage, there is a lot of idle time unless the cluster has a second purpose outside the classroom.

It is also possible that multiple courses utilize the same resources, creating an overlap in peak-period usage. This overlap can pose a problem, as students have to compete for resources. Depending on how intensive the calculations are for each course utilizing the cluster, limited resources can determine what the student can accomplish.

Living in clouds with StarHPC

StarHPC remedies the problems associated with bringing computational resources into the classroom by providing a dynamic, on-demand, compute cluster hosted by Amazon‘s EC2 web service. EC2 provides a web service that allows a user to request a number of virtual machines to be started on computers in Amazon‘s data centers collectively called "the cloud." Once a virtual machine has been started, the user pays only for what they use, at $0.10/hr until it‘s shutdown. Comparing this to the cost of owning traditional compute clusters, EC2 presents a very appealing service with an affordable pricing model.

Amazon also provides an extensive API for managing these virtual machines. With the API, the user can dynamically adjust to their current load by adding or removing machines as necessary. It can also be used to capture the working state of a virtual machine into a virtual machine "image." The image can then be reloaded at a later time to produce an identical configuration. This is useful for capturing the software and configuration needed for a particular course. Using the Amazon EC2 API for starting and stopping machines and creating virtual machine images, it is possible to automate administrative overhead such as creating user accounts, setting up workspaces, configuring remote access, and so on.

Amazon's EC2 web service addresses many of the issues associated with bringing computational resources into the classroom. First, there are no concrete resources to house, power, and cool as with traditional compute clusters. Also, the setup and configuration associated with the resources are captured in a virtual machine image which can be loaded at will. This means administrative work only needs to be done once, and not every semester. Second, it provides a way to eliminate under-utilized resources since the user only pays for what they use. There is no need for an EC2 cluster to be up and running outside peak-period usage. Finally, overlaps in peak-period usage due to multiple courses needing computational resources are eliminated by the ability to launch multiple independent clusters. Furthermore, by creating separate virtual machine images, each course can have its own set of configurations and software packages completely independent of each other.

Results

StarHPC was used in the "Parallel Programming for Multi-core Machines Using OpenMP and MPI" course during an Independent Activities Period (IAP) at MIT. The course was a two-week crash course in both MPI and OpenMP programming techniques for writing software that runs on high performance computing systems. Throughout the course, students were guided through implementing increasingly complex set of parallel programming exercises as homework. Students logged-in remotely to the StarHPC cluster to develop, compile, and run their source code from virtually anywhere, even from a web browser if necessary. The course lasted two weeks and had a total of ten users actively developing on the cluster. The estimated cost for using StarHPC hosted on Amazon‘s EC2 web service was around $15 per student. Comparing this to the cost of a physical cluster, cloud computing presents an affordable solution to supporting peak-period computing needs in the classroom. To learn more about StarHPC, visit the project web page at (http://web.mit.edu/star/hpc).

EVALUATION OF STAR SOFTWARE

The process of creating a STAR product begins during the initial discussion with a researcher. The pivotal question that we ask is "What is the biggest problem you have bringing your research into the classroom?" The answer to this question determines what the STAR team will attempt to solve. We explicitly encourage the researcher to think in terms that are very broad as our goal is to find projects that enable teaching of important research concepts that are currently too difficult to teach due to some technical issue. This discussion defines the scope of STAR projects.

The STAR team then turns the researcher's definition into a prototype that is evaluated by the researcher to determine if it captures a solution that addresses the problem. This development/evaluation cycle is executed until the solution is determined viable in the context of what the researcher wants to accomplish in the classroom. The researcher then develops an initial set of classroom exercises (problem sets) to capture the intent of software use in the classroom.

The STAR team then goes through a process of assessing the usability, scalability, and robustness of the solution using standard software development techniques. The team works closely with the MIT Usability Lab and student volunteers to unearth flaws in the software and in the initial problem sets. They test the scalability of the software to determine if it is usable with the expected student load the night before the homework is due. Extensive testing is then conducted to ensure that new software has no significant flaws that would make classroom use problematic. A final review is made by the developers to determine if the software is classroom ready.

This is just the first step in assessing the value of STAR products. There are three more types of evaluations that we are planning to undertake with our current work. The first is student assessment of the value of the software. The second is an assessment of the benefit the student carries away from a course using the software. The third assesses the long-term benefit from using the software.

In talking with our students and looking at survey results, we believe that the students enjoy using the software and feel that using the software was valuable, but survey participation is low, and getting opinions verbally was anecdotal. Moving forward, we will work with assessment experts at MIT to extract more useful information.

Even more difficult is assessing what students take away from a course using the software. The STAR team doesn't control the classroom use of the software and different courses use the software in different contexts. Consequently, this remains a challenge for the team. Our current thinking is that the closest that we can come to useful information is to make this assessment during our usability tests in the lab. The downside of this is the small sample size.

We are skeptical that assessment of long-term benefit is possible. Research by its very nature is constantly changing focus. The best that we can hope for is that researchers feel the software we develop is relevant and helps them bring their research and thinking into the classroom. During one of our trials with high school students using StarBiochem, our structural biology collaborator, Prof. Melissa Kosinski-Collins, remarked “I’ve always been able to see these protein structures in my head. Now I can show them to the students.”

FUTURE WORK

We will continue our efforts to lower barriers to the exploration of scientific and engineering topics. Our emphasis will be on providing better user experiences and hiding non-essential system complexities.

In the near future, we will look to extend our software development in Biology and work with the MIT Biology Department to make Computational Biology ubiquitous in the Biology curriculum. We plan to integrate high performance computing into much of the software we are developing. This will provide a foundation for a rich, visual experience coupled with significant computational power. We believe that this is a framework for rich exploratory learning in classroom and homework settings and also a positive influence on the development of research tools.

We will formalize the inclusion of research domain experts in both the development and dissemination phases of STAR product development. The MIT Teaching and Learning Laboratory will assist us in formal assessment of STAR software effectiveness and help us develop a framework for assessing the impact of our software on teaching and narrowing the gap between research and teaching.

CONCLUSIONS

The Software Tools for Academics and Researchers (STAR) project has developed a framework for deploying streamlined user interfaces and building work flows for running experiments in several areas of research and education that are both sustainable and scalable.

StarGP provides a form-based web interface for running command line tools where complex input and output formats are the norm. In addition to low overhead for setting up students, StarGP allows students to harvest computing resources for running complex scientific simulations.

StarBiochem is tackling the problem of intuitive user interfaces. It provides an environment for students to explore complex visual structures and create snapshots that help them understand how structure relates to function.

StarHydro provides an easy to use, high quality analysis and visualization package for exploring watersheds.

StarBiogene has enhanced existing genomics research software for microarray analysis from the Broad Institute by making the software more usable and accessible. With StarBiogene, genomics research software can be accessed from a centralized web interface instead of needing to install the software locally.

StarHPC provides the development environment and computational resources necessary to teach parallel programming in OpenMP and OpenMPI. StarHPC represents the STAR team‘s first attempt to use Amazon‘s EC2 web service to address the issues with bringing computational resources into the classroom.

References

Baroni, S., Dal Corso, A., de Gironcoli S., Giannozzi, P., Cavazzoni, C., & Ballabio, G. et al. (2005). *Quantum Espresso*. Retrieved April 1, 2008, from http://www.quantum-espresso.org/

Belcher, JW. (2003). *Transferring knowledge and experience in innovative educational transformation* [MIT Course]. Retrieved from http://gallery.carnegiefoundation.org/gallery_of_tl/transferring_knowledge_and_experience_in_innovative_educational_transformation.html

Bernstein, FC., Koetzle, TF., Williams, GJB., Meyer, EF., Brice, MD., & Rodgers, JR., et al. (1977). *RCSB Protein Data Bank* [Research Project]. Retrieved from http://www.rcsb.org

Chisholm, SW., Walker, G., Khodor, J., & Mischke, M. (2005). *Introductory Biology, 2005* [MIT Course] Retrieved from http://ocw.mit.edu/OcwWeb/Biology/7-014Spring-2005/CourseHome/

Golub, SW., Slonim, DK., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, JP., Coller, H., Loh, M., Downing, JR., Caligiuri, MA., Bloomfield, CD., Lander, ES. (1999).

Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring, 1999 Science Vol. 286 no. 5439 pp 531-537

Reich, M., Liefeld, T., Gould, J., Lerner, J., Tamayo, P., Mesirov, JP. (2005). *GenePattern 2.0*, Nature Genetics, Retrieved from http://www.broad.mit.edu/cancer/software/genepattern/index.html

Yip, S., Powell, A., Bazant, M., Cao, J., Carter, CW., Hadjiconstantinou, N., et al. (2002). *Introduction to Modeling and Simulation* [MIT Course]. Retrieved from http://ocw.mit.edu/OcwWeb/Nuclear-Engineering/22-00JSpring-2006/CourseHome/

Author's Note

The initial work for the "Bringing Research Tools into the Classroom" initiative was funded by MIT'S Academic Computing and Information Services and Technology (IS&T) "Big Initiative."

Acknowledgements

Prof. Graham Walker, Prof. Melissa Kosinski-Collins, Dr. Michelle Mischke, and Dr. Megan Rokop for being outstanding collaborators; their enthusiasm helped StarBiochem attain the success it has. Prof. Walker received support for his participation from the Howard Hughes Medical Initiative (HHMI).

Our collaborators from the Broad Institute of MIT and Harvard. In particular, Dr. Jill Mesirov, Ted Liefeld, Joshua Gould, and Michael Reich for their suggestions on how to utilize GenePattern and for their help with modifying the data visualizer software for StarBiogene.

Prof. John Belcher for his insight regarding pedagogical aspects of software, and the TEALsim developers at the MIT Center for Educational Computing Initiatives - Andrew McKinney, Phillip Bailey, and Michael Danziger - for being great collaborators. Prof. Belcher's TEALsim project received support from the Davis Educational Foundation.

Prof. Nicola Marzari, Prof. Markus Buehler, and Dr. Timo Thonhauser for being exceptional clients and allowing us to bring their fascinating research codes into the classroom.

Prof. Rafael Bras and Prof. Dara Entekhabi have provided wonderful support for our work developing software for Hydrology.

Dr. Vijay Kumar, Associate Dean and Director of the Office of Educational Innovation and Technology (OEIT), and Prof. Dan Hastings, MIT Dean of Undergraduate Education (DUE) for their on going support.