Project Year

2016-2017

Project Title

Interactive Tutorials and Assignments for Biomedical Engineering

Project Team

Eileen Haase, Whiting School of Engineering, Biomedical Engineering, Faculty
Richard Chen, Whiting School of Engineering, Biomedical Engineering, Fellow
Andrew Tsai, Whiting School of Engineering, Biomedical Engineering, Fellow

Audience

The intended audience of our project is all BME students at Johns Hopkins University. Our project is meant not only for freshman and sophomore students who want to follow the latest trends in biomedical research, but also for junior and senior students who want to adapt to BME coursework that is becoming increasingly computational-heavy with machine learning. Though our project will affect primarily the BME undergraduate curriculum, the resources built from our solution will be applicable to students in other majors where scientific programming is an incredibly desired skill such as Chemical & Biomolecular Engineering, Computer Science, Biophysics, and Applied Mathematics & Statistics.

Pedagogical Challenge

As BME is becoming an increasingly computational-heavy field, there is a need for greater emphasis on scientific computing in the undergraduate curriculum. Courses such as *Molecules & Cells* and *Systems Bioengineering* are structured such that the course material can adapt to leading-edge research within the field. For example, this year in *SBE II*, instead of learning about neural excitability via cable theory, we learned it using Poisson processes and generalized linear models. Though new and exciting, many course assistants could not properly instruct on the material, and many students were unable to complete the assignments without verbatim snippets of code from the professor. Currently, there exists only a MATLAB tutorial for BME students. While MATLAB is powerful, it limits the diversity of problems that students encounter in the classroom. In order for the new course material to properly engage students, the tools we use to tackle these problems must also be fresh and up-to-date.

Solution

Our solution is two-fold. It will:

1) Provide interactive tutorials in Python and R

    In Python and R, it is possible to write interactive, online tutorials that allows the student to view the output of programming lines like a PowerPoint. Our solution would allow students to rapidly teach themselves the foundations of scientific computing.

2) Introduce programming assignments that will help enrich the course material taught in BME courses such as *Molecules & Cells*, and *SBE I & II*.

    -Assignments in network biology inference and differential expression analysis using R in *Molecules & Cells*. These assignments will not only add variety to a class that is mainly qualitative, but prepare students for research in systems biology

- Assignments in machine learning using Python in *SBEII*. These assignments will not only help illustrate neural excitability in the class, but also prepare students for research in predictive modeling

Assessment Strategy

To assess our solution after implementation in a classroom setting, we would read every course evaluation after the course was completed, and look for feedback on the programming assignments. We would first implement our solution in *Molecules & Cells*, which is taught in the fall, and then in *SBEII* in the spring. For both courses, we would quantify our success based on the number of favorable comments of our solution. In addition to comments from course evaluations, we would communicate with the course assistants as well, and listen to their feedback on the quality of our solution. After a first implementation of our solution in *Molecules & Cells*, we would make any improvements on our solution before we implement it in *SBEII*.

Faculty Proposal

Dr. Haase has led a number of successful Technology Fellowships, which have had a huge positive impact on learning, not only within BME, but also throughout university and the world. Our most recent project, a series of over three dozen tutorials on building with and using electronics, *BME Builds*, receives over 10,000 hits per month, already over 80,000 hits since it was posted last June (https://www.youtube.com/channel/UCK9uRj5u4afQGQWWFtU0jPA).

Dr. Haase worked with three BME students to develop a suite of online mini-lectures and computer simulations for *Molecules and Cells* (580.221) (available on YouTube through JHU MolCell, which had over 11,000 views as of 2/2016 - https://www.youtube.com/channel/UCFt4pZ6tKZZ4gxnPfku0KrQ). Our original Technology Fellowship was used to work with three BME seniors in the development of an online Matlab tutorial used in *Freshmen Modeling and Design* (580.111) and other BME courses to introduce novice programmers to Matlab (available on YouTube through JHU Matlab: https://www.youtube.com/channel/UCAzVE1csErV_ufPIkZo3t6A).

Over the years, our BME students have done a phenomenal job of pinpointing student needs and developing the resources necessary for student success. Our biomedical engineering students bring unique analytical skills to hundreds of research labs each year. Incorporating these analytical skills into our courses will help our students become even better prepared to immediately contribute to physiological and cell systems modeling and data analysis. In addition to Matlab, many students learn other languages, such as Python and R. It is important for our students to understand how different programming languages might be best suited for different types of situations. Above all, a unifying theme of analyzing real-life physiological data should be one of the hallmarks of our BME program. Faculty do not have the time, or the expertise, to include teaching programming in their courses. A combination of tutorials and assignments would greatly benefit faculty, students, and the BME program as a whole.

Due to the popularity of our previous video series, our IT department has now developed a “branding” video which will be added to the beginning and end of each tutorial. This video identifies this top quality work as the product of our JHU BME department. Previous video series have been posted using various names (“BME Builds”, “JHU MoleCell”, “JHU Matlab”) which make it difficult to identify as a product of

our Johns Hopkins BME students. Credit will always be given to the Center for Educational Research for their Technology Fellowships in initiating these projects.

Blackboard allows “Statistics Tracking” to count the number of times students use the links to the videos posted on YouTube. However, since students can bypass Blackboard and access YouTube directly, we will need to use a different mechanism of evaluation, such as end-of-semester surveys and comparison of the depth and quality of homework assignments before and after the availability of these tutorials.