2021 10th International IEEE/EMBS Conference on Neural Engineering (NER)
Virtual Conference, May 4-6, 2021

Multi-platform simulations facilitate interdisciplinary instruction in undergraduate neuroscience

David W. Donley*, Ziao Chen*, David Bergin,
David J. Schulz, Satish S Nair, *Senior Member, IEEE*

2021 10th International IEEE/EMBS Conference on Neural Engineering (NER) | 978-1-7281-4337-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/NER49283.2021.9441407

Abstract— **Neuroscience is a highly interdisciplinary field, but more collaboration among STEM disciplines is needed to advance undergraduate neuroscience education. This paper reports the development of code-based virtual laboratories to cross-foster ideas from engineering and biology in the neuroscience classroom. The simulations use a combination of NEURON and Python code to model neurophysiological processes. We report that the use of computational tools in the classroom increases student self-reported comfort in participating in a computational research project. The tools we developed have potential to increase persistence and retention of undergraduate students by encouraging interdisciplinary thinking and reducing barriers to entry.**

I. Introduction

Quantitative tools are increasingly important for biologists and should be incorporated into the undergraduate curriculum to facilitate a deeper understanding and provide opportunities for independent research [1], [2]. The *Vision and Change* initiative identified competencies related to computation and the need for quantitative reasoning and modeling [3]. These trends in undergraduate higher education create opportunity for collaboration among STEM fields. The field of neuroscience is, by nature, a highly interdisciplinary field. Many neuroscience programs exist as collaborations between psychology and biology. More recently, there has been a recognition that engineering, mathematics, and computer science concepts enhance neuroscience degree programs [4].

Despite the need for increasing collaboration, there are many barriers to entry for incorporating computation into the life science classroom at the undergraduate level. First, STEM fields place a premium on laboratory experiences that provide students concrete, discipline-specific skills. Techniques learned in the life science laboratories provide a foundation for students seeking summer research internships or mentored research experiences during the academic year. Second, computer-based learning using simulations or virtual "lab" exercises have traditionally not been viewed as an effective method to promote student learning. In part, this is due to the fact that many simulations and computer models oversimplify biological processes. Computer-based tools may be used as a cost-saving measure but if they do not recapitulate the biological system, learning may be compromised. Third, programming languages often seem inaccessible to novices. To solve these issues and open more opportunities for interdisciplinary course offering, tools are needed that address these barriers.

There have been several studies that demonstrate the usefulness of virtual laboratories or simulations in the life science classroom [5]–[7]. Particularly when used in combination with other teaching modalities, simulations and virtual demonstrations can increase student interest and facilitate inquiry-based learning [5], [7]. In the future, undergraduate instructors will need a computational toolbox on which to draw that addresses barriers to entry for computation.

Herein, we describe the implementation of virtual labs that simulate central nervous system functions. The virtual labs use Jupyter Notebooks as a method of distribution. The underlying physiology is implemented using NEURON [8]. Python is used to implement interactive portions of the code without the need to know how to write code. Together, these tools provide a method for engaging students in inquiry-based exploration of neuroscience processes. Additionally, we report that computational tools have potential to engage students and promote inclusion in the research community similarly to students who have a traditional laboratory experience.

II. Methods

CRediT Taxonomy Assessment.

The CRediT Taxonomy (**Table 1**) designates contributor roles in a research project; it was adapted as an assessment instrument from a previously published study [9]. In this study, students were asked to envision a future experiment in neuroscience using either primarily computational tools or primarily 'wet-lab' tools. Students were asked to rate their comfort with each CRediT role using the following scale:

0 = I would be most comfortable with no responsibility

1 = I would be comfortable with very little responsibility or would only be comfortable if I had a lot of supervision

2 = I would be comfortable with moderate responsibility and could work somewhat independently in this role if I had a mentor or collaborator

3 = I would be comfortable with primary responsibility in this area and would be comfortable and competent with little to no supervision

* Equal contributions. DW Donley is with the Department of Biology, Harding University, Searcy, AR 72143. Z Chen and SS Nair are with Electrical Engineering and Computer Science, D Bergin is with the College of Education, and D Schulz is with Biology, University of Missouri, Columbia MO 65211 (corresponding author DWD Ph: 501-279-4151; fax: 501-279-4706; e-mail: ddonley@harding.edu). This research was supported in part by NSF grants EEC-1801666, DBI-1659831, OAC-1730655, an NIH grant MH122023 to SSN, and by a grant to DWD by the Arkansas INBRE grant program from NIGMS (P20 GM103429) at NIH.

TABLE 1: CREDIT TAXONOMY FOR CONTRIBUTOR ROLES IN A RESEARCH PROJECT. STUDENTS WERE ASKED TO RATE THE EXTENT TO WHICH THEY WOULD BE COMFORTABLE TAKING ON EACH ROLE IN A HYPOTHETICAL FUTURE RESEARCH PROJECT USING EITHER PRIMARILY COMPUTATIONAL TOOLS OR PRIMARILY TRADITIONAL 'WET-LAB' TOOLS

Role	Definition
Conceptualization	Ideas; formulation or evolution of overarching research goals and aims
Data curation	Management of activities to annotate, clean data, and maintain research data for initial and future use
Formal analysis	Application of statistical, mathematical, computation techniques to synthesize study data
Funding acquisition	Acquisition of the financial support for the project leading to publication
Investigation	Conduct of research process, specifically performing the experiments or data collection
Methodology	Development of methodology; creation of models
Project administration	Management and coordination responsibility for research activity planning and execution
Resources	Provision of study materials, reagents, patients, laboratory samples, instruments, computing resources, and/or other analysis tools
Software	Programming, software management; implementation of computer code; testing or use of existing code components
Supervision	Oversight and leadership responsibility for the research activity planning and execution, including mentorship of others on the core team
Validation	Verification; evaluating replication / reproducibility of results or research output
Visualization	Preparation, creation, and/or presentation of the work, specifically data presentation
Writing – original draft	Preparation and writing of the initial draft that reports the work
Writing – reviewing, editing	Critical review, commentary, or revision of reports on the work

Demographic data from students was gathered as a part of this survey. This survey was approved by the Harding University Institutional Review Board (#2020-141). Data was gathered from three different upper-division course-based undergraduate research experiences. Students anonymously completed the rating scale, and no student grades were impacted by this survey.

Development of Inquiry-based Software Labs

Previously, we reported the development of 'Software Labs' using NEURON [10]. Laboratory exercises were based on the NEURON programming language. These Software Labs used and illustrated both neuroscience and engineering principles. Software labs involve active learning strategies that promote student engagement [11].

Development of the multi-platform notebooks

Software labs built in NEURON were adapted to Python-based software tutorials. Python was used at the command line to run the NEURON simulations and generate graphs in a single notebook. This structure requires students to become familiar with distributors to run Jupyter Notebooks or web-based notebook environments such as Google Colab as well as learn basic Python and NEURON commands.

To facilitate student engagement, each Software tutorial allows students to engage with the simulations either through command-line manipulation or through Python widgets. All source code and notebooks can be found at https://courses.missoui.edu/ under the 'Canvas Guest and Visitor Login' link. The username for these tools is 'cns' and the password is 'workshop'. (without quotes)

III. RESULTS

To understand how well computational laboratory exercises prepared students for entry into the scientific community, the CRediT Taxonomy survey was used to assess how students responded to research-based experiences. Students rated the extent to which they would feel comfortable in various roles in a future research project in any topic of neuroscience.

Students reported their comfort level in each role using a 0-3 rating for a project using primarily computational methods and a research project using primarily traditional 'wet-lab' techniques (**Fig. 1**). All students surveyed were majoring in the life sciences; 45% of responses were from students majoring in Biology and 55% from students majoring in Molecular Biology. Students were asked about the resources used in their coursework. There was no difference in student scores on the CRediT Taxonomy for a project that used traditional 'wet-lab' techniques.

However, students who had previously used computational tools to in courses or laboratories reported that they would be equally comfortable in a primarily computational project or a primarily 'wet-lab' project. These result are somewhat unsurprising given that students with more experience using a technique are more likely to be comfortable using that techniques in the future. Interestingly, these students would still feel comfortable in a 'wet-lab' research project, suggesting that the laboratory experiences in a typical undergraduate curriculum provide students a baseline level of confidence. These data also demonstrate that students who have experience with command-line interfaces are more comfortable with the idea of a computation-based research project. Students' scores were similar for common aspects of the research process such as Formal Analysis, Data Curation, Visualization, and Writing. These results further illustrate the ability of virtual tools to teach some aspects of the scientific process. The largest differences between student ratings were in project-specific roles such as Conceptualization and Investigation. Students overall felt more comfortable in a Conceptualization role for wet-lab experiments irrespective of research project methods (1.7±0.6 for primarily computation projects versus 2.5±0.4 for primarily wet-lab research). However, prior experience with virtual tools increased student comfort with Investigation using computational tools (from 1.4±0.5 for students without experience to 2.3±0.4 for students with prior experience). Students were overall uncomfortable in roles that are not usually experienced in an undergraduate classroom such as Funding Acquisition and Project Administration and Supervision.

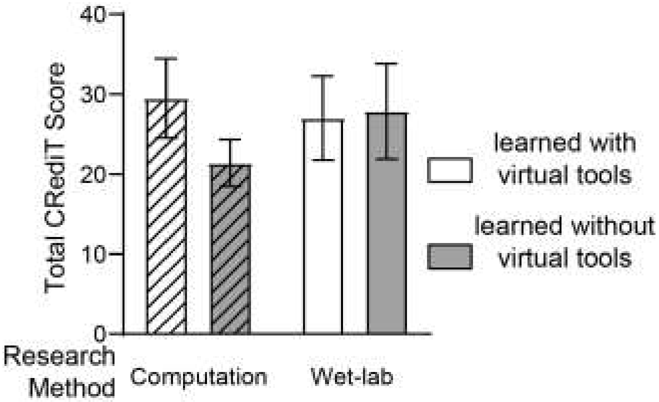

Figure 1. Total ratings from the CRediT Taxonomy from life sciences students who have previously used computational or virtual tools that require the use of one or more programming languages to learn biology (white bars) compared to student who have not had computation experience (grey bars). Students rated their level of comfort for each CRediT role for a hypothetical future study using primarily computational tools (hatched bars) or primarily wet-lab tools (open bars). N=20. Bars represent means ± standard deviation.

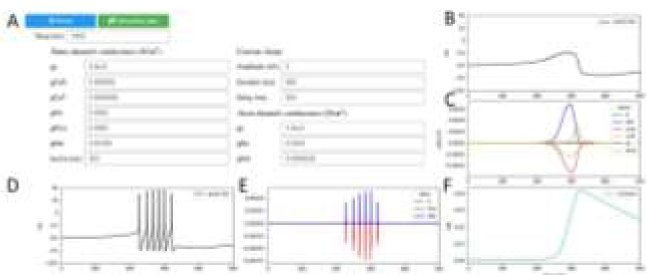

Figure 2. Representative images of the ‘Burster’ tutorial using Python and NEURON programming languages. **A**. Widget options for students to perform inquiry-based manipulations of a model of a bursting hippocampal neuron. **B**. Soma membrane potential plot. **C**. Plots of the currents through ion channels in the soma. **D**. Axon membrane potential plot. **E**. Current graphs through ion channels in the model axon. **F**. Total calcium pool in the neuron.

We approached the incorporation of computational models for neuroscience education with the hypothesis that students should be responsible for understanding the underlying biology of the system. Previously we reported the development of “Software labs” and their incorporation into an interdisciplinary neuroscience course [11]. These simulations use biophysical characteristics of neurons to model biological processes. We now report improvements to these simulations to use a multi-platform environment that encourages student inquiry. The new “Software tutorials” allow students to use Python-based widgets to manipulate biophysical parameters of neurons and gain understanding. (**Fig. 2A**) As student gain experience with the environment, they have access to edit programming commands. Each ‘Software tutorial’ contains questions to guide learning and direct the manipulations of the models. The models provide graphs that plot dependent outcomes including membrane voltage, current, and ion concentrations. (**Fig. 2B-F**) Students must read and interpret the graphs to answer questions and develop new questions. Therefore, these tutorials develop quantitative literacy. In the example shown in **Figure 2**, students work with a bursting neuron. In this model, students begin to construct mental models of the relationships between biophysical parameters in the soma versus axon of a neuron.

The use of computational models is well received and contributes to the goals of interdisciplinary courses.

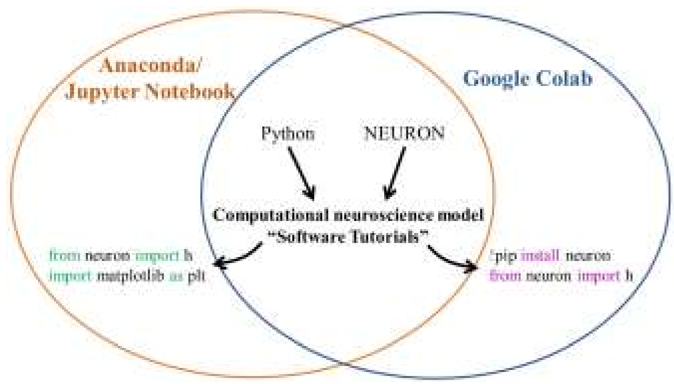

Figure 3. Schematic demonstrating the multi-platform nature of the ‘Software tutorials’. Each notebook utilizes both Python and NEURON code (black) but utilizes specific aspects of the distributor platform. For example, using an Anaconda distributor with Jupyter Notebooks, students use Python to import the needed databases. They need to have NEURON separately installed on the computer. In contrast, Google Colab requires students to install NEURON in their Colab environment using !pip. Each approach has advantages and disadvantages but encourages students to develop cross-disciplinary skills.

In total, we have developed six complete ‘Software tutorials’ that can be implemented in courses as classroom demonstrations, complete modules, or as the beginnings of a student project such as a senior capstone. The tutorials are conceptually progressive but independent. The first tutorial illustrates basic biophysical properties of neurons, followed by a simulation of action potential development plotted against channel gating variables in context of membrane potential changes. These foundational concepts prepare students for the third and fourth tutorials that allow students to explore the bursting activity in neurons (**Fig. 2**) and synaptic function. In these activities, students explore larger biological concepts such as summation in the context of mathematical and engineering principles. Students then are able to engage in circuit and network level thinking with tutorials five and six focused on modeling a central pattern generator and a memory circuit.

The interdisciplinary nature of the tutorials facilitates the development of a diverse skill set. Inquiry-based assignments that use computational tools require students to learn how to properly install and run programming environments as well as how to import and use programming libraries. These skills allow students to have an active role in their learning experience. In addition to teaching discipline-specific concepts, students learn concrete, applicable skills in programming. Together, this multi-platform framework develops a diverse skill set while maintaining a focus on the core content. (**Fig. 3**)

IV. Discussion and Conclusion

More tools are needed to facilitate cross-disciplinary thinking and the use of computational tools in the life science classroom. The tools we developed use a strong mathematical and engineering foundation to develop simulations for the undergraduate neuroscience classroom. This structure opens opportunities for Just-in-Time Teaching (JiTT). The JiTT approach is based on a model where instructors lecture or lead class discussions only after students generate questions on the topic [12]. The JiTT model is an iterative method of engaging students in the learning process. The use of code and mathematics-based tools similar to these has been used to engage students across disciplinary boundaries and encourage a team-based approach to learning [13], [14].

Our survey data suggests that the assumption among many biologists that virtual laboratory exercises are inferior may not be correct (**Fig. 1**). Certainly, more work is needed to fully elucidate the impact of virtual laboratory projects. However, our data indicates that students are, at a minimum, more open to the idea of computational research when they have prior experience with virtual tools to learn biological concepts. The results presented in **Figure 1** are consistent with previous reports that suggest virtual tools are effective in promoting learning, interest, and curiosity [7], [15].

There are many challenges to offering interdisciplinary courses and using interdisciplinary tools in the classroom. We report the development of ready-to-use tutorials that reduce barriers to entry. Namely, our tools run on freely available platforms such as Anaconda and in web-based notebook environments such as Google Colab. By virtue of the flexibility, these tools allow students to access simulation and engage in virtual, inquiry-based laboratory exercises in neuroscience without having a strong background in computer science or programming (**Fig. 3**). These tools are adapted for classroom use while maintaining a high level of scientific rigor (**Fig. 2**). Importantly, these tools support a mechanistic understanding of neuronal physiology from both a biological and engineering perspective.

Providing updated tools that reduce barriers to entry, encourage critical thinking, and support interdisciplinary engagement will facilitate persistence and retention across STEM disciplines. More work is needed to evaluate these tools and to develop best practices for their implementation. We propose that interdisciplinary tools harness the power of diversity in the classroom and facilitate a team-based approach to science. We propose a model where students engage in interdisciplinary, inquiry-based learning in the classroom then progress to independent research projects. We hypothesize that interdisciplinary tools such as our simulations and models can help attract underrepresented groups to STEM fields [16], [17]. Interdisciplinary neuroscience has potential to attract, for instance, women in the life sciences to engineering and mathematics-based disciplines. Women and other minorities have higher representation in the life sciences; thus, the tools that foster inclusive learning environments can attract students from underrepresented groups. More tools are needed to reduce the perceived difficulty of “learning to code”. Multi-platform tools that run in a variety of environments provide the opportunity for instructors to implement simulation in the classroom without promoting educational inequality. Herein we have discussed one approach to solving these issues but more work is needed to expand access to tools that promote interdisciplinary collaboration.

References

[1] M. M. Chen, S. M. Scott, and J. D. Stevens, “Technology as a tool in teaching quantitative biology at the secondary and undergraduate levels: a review,” *Letters in Biomathematics*, vol. 5, no. 1. Taylor and Francis Ltd., pp. 30–48, 01-Jan-2018.

[2] T. Aegerter-Wilmsen and T. Bisseling, “Biology by numbers - Introducing quantitation into life science education,” *PLoS Biology*, vol. 3, no. 1. Jan-2005.

[3] AAAS, *Vision & change in undergraduate biology education*. 2011.

[4] E. H. Chudler and K. C. Bergsman, “Brains-computers-machines: Neural engineering in science classrooms,” *CBE Life Sci. Educ.*, vol. 15, no. 1, pp. 1–7, Mar. 2016.

[5] R. S. Pearlman *et al.*, “Virtual Lab Demonstrations Improve Students’ Mastery of Basic Biology Laboratory Techniques,” *J. Microbiol. Biol. Educ.*, vol. 10, no. 1, pp. 51–57, May 2009.

[6] L. E. de Vries and M. May, “Virtual laboratory simulation in the education of laboratory technicians–motivation and study intensity,” *Biochem. Mol. Biol. Educ.*, vol. 47, no. 3, pp. 257–262, May 2019.

[7] N. R. Dyrberg, A. H. Treusch, and C. Wiegand, “Virtual laboratories in science education: students’ motivation and experiences in two tertiary biology courses,” *J. Biol. Educ.*, vol. 51, no. 4, pp. 358–374, Oct. 2017.

[8] M. L. Hines, N. T. Carnevale, and M. L. Hines, “The NEURON Simulation Environment,” pp. 1–9, 2002.

[9] M. Honoré, T. E. Keller, J. Lindwall, R. Crist, L. Bienen, and A. Zell, “Contributions Made by Undergraduates to Research Projects: Using the CREDIT Taxonomy to Assess Undergraduate Research Experiences,” *Scholarsh. Pract. Undergrad. Res.*, vol. 4, no. 1, pp. 41–51, 2020.

[10] B. Latimer, D. Bergin, V. Guntu, D. Schulz, and S. Nair, “Open Source Software Tools for Teaching Neuroscience,” *J Undergr. Neurosci Educ*, vol. 16, no. 3, pp. A197–A202, 2018.

[11] B. Latimer, D. A. Bergin, V. Guntu, D. J. Schulz, and S. S. Nair, “Integrating model-based approaches into a neuroscience curriculum - An interdisciplinary neuroscience course in engineering,” *IEEE Trans. Educ.*, vol. 62, no. 1, pp. 48–56, 2019.

[12] K. A. Marrs and G. Novak, “Just-in-Time Teaching in Biology: Creating an Active Learner Classroom Using the Internet,” *Cell Biol. Educ.*, vol. 3, no. 1, pp. 49–61, Mar. 2004.

[13] Y. V. Tra and I. M. Evans, “Enhancing interdisciplinary mathematics and biology education: A microarray data analysis course bridging these disciplines,” *CBE Life Sci. Educ.*, vol. 9, no. 3, pp. 217–226, 2010.

[14] I. Karsai, J. Knisley, D. Knisley, L. Yampolsky, and A. Godbole, “Mentoring interdisciplinary undergraduate students via a team effort,” *CBE Life Sci. Educ.*, vol. 10, no. 3, pp. 250–258, 2011.

[15] L. Stadtlander, M. Giles, and A. Sickel, “The Virtual Lab: Research Outcomes Expectations, Research Knowledge, and the Graduate Student Experience,” vol. 3, no. 1, pp. 1–19, 2013.

[16] M. Estrada *et al.*, “Improving underrepresented minority student persistence in stem,” *CBE Life Sci. Educ.*, vol. 15, no. 3, Sep. 2016.

[17] R. Lansiquot, R. Blake, and J. Liou-Mark, “Enhancing Diversity in STEM Interdisciplinary Learning,” 2013, pp. 237–267.