ARTICLE
Open Source Software Tools for Teaching Neuroscience

Benjamin Latimer[1†], **David Bergin**[2†], **Vinay Guntu**[1], **David Schulz**[3], **& Satish Nair**[1]

[1]Electrical Engineering & Computer Science, [2]Educational, School, & Counseling Psychology,[3]Biological Sciences, University of Missouri, Columbia, MO 65201. †These authors contributed equally to this work.

We developed an interdisciplinary course in computational neuroscience to address the need for students trained in both biological/psychological and quantitative sciences. Increasingly, exposure to advanced math and physics is important to stay on the cutting edge of developments and research in neuroscience. Additionally, the ability to work in multidisciplinary teams will continue to be an asset as the field develops. This course brings together students from biology, psychology, biochemistry, engineering, physics, and mathematics. The course was designed to highlight the importance of math in understanding fundamental neuroscience concepts and to prepare students for professional careers in neuroscience. They learn neurobiology, via a 'biology to model and back again' approach involving wet- and software/modeling-labs, with the latter being the focus of this paper. We presented a subset of the software activities described here at the 2017 Faculty for Undergraduate Neuroscience Workshop.

Key words: Computational Neuroscience; Software; Electrophysiology

---

The 21st century may well be the "Century of the Brain." With recent advances in neuroscience techniques, the mapping of the human genome, and increases in computing power, neuroscience is primed for major advances in the coming decades. To meet this grand challenge, a new generation of biomedical researchers must be trained in an interdisciplinary fashion that unites the fields of engineering, biology, mathematics, physics, and computer science.

The National Academies recommended, in 2003, "a comprehensive reevaluation of undergraduate science education for future biomedical researchers." It calls for a "renewed discussion on the ways that engineering and computer science, as well as chemistry, physics, and mathematics are presented to life science students" (National Academies, 2003). The American Association for the Advancement of Science has said that "interdisciplinary research will have to be taken to new levels to address exceedingly difficult challenges such as autism, infectious diseases, and cancer" (Palmer, 2011).

We developed an interdisciplinary course in neuroscience to help address these needs. Using the framework of Computational Neuroscience, we introduce concepts like differential equations, circuit theory, and basic neuron physiology. Students are exposed to software labs as well as electrophysiology labs to get a broad understanding of the inner workings of neurons. This interdisciplinary approach helps expose life science students to quantitative techniques and engineering students to biological techniques.

Some interdisciplinary neuroscience courses have been developed and reported (Modo and Kinchin, 2011; Reynolds, 2012; McFarlane and Richeimer, 2015). Most cite differing background knowledge as the main challenge in designing an interdisciplinary course. Courses have been developed that bring together students from biology and art (Reynolds, 2012) or science and humanities (McFarlane and Richeimer, 2015). Our course curriculum and open-source software tools are designed to cross-train quantitative science (e.g., engineering, physics and math) and life science (e.g., biology and psychology) majors.

Others have reported on computational neuroscience software simulations that may be useful in undergraduate neuroscience education (Av-Ron et al., 2006; Stuart, 2009; Schettino, 2014; Nichols, 2015). These range from simple graphical user interfaces to simulations where students are involved directly in coding, though most tend toward the former. Software simulations can be invaluable in neuroscience programs, especially in small liberal arts colleges where class sizes tend to be small but there are often space and financial resource constraints (Bish and Schleidt, 2008).

Electrophysiology labs in undergraduate courses have also been reported (Ramos et al., 2007; Wyttenbach et al., 2014). These labs, of course, are more complex in terms of equipment, knowledge, and preparation than software labs. That said, they are critical for exposing students to actual techniques used in neuroscience research. Few, if any, courses have been reported that expose students to both computational neuroscience and electrophysiology.

Life science students and engineering students bring different prior knowledge and experience to the course. Life science students may be uncomfortable with math topics like differential equations, while engineering students usually have very little experience with biological systems. The instructor must be aware of this imbalance in background knowledge and design the course such that prior knowledge is either not necessary or can be learned.

To address this, we begin the course with a refresher on basic math concepts such as functions, slope, and time constant (Table 1). These concepts can be intimidating for life science students, so care is taken to emphasize conceptual patterns over mathematical jargon and notation. We demonstrate visual models before introducing mathematics. To assist students lacking adequate background in math, we encourage peer-to-peer teaching between students from different disciplines (Reynolds, 2012) and provide weekly help-sessions throughout the semester. Access to our materials is freely available via the

Focus of week	Monday (Lectures)	Wednesday (Labs)	Friday
1: Introduction	Welcome & Introduction	Software Experiment 0: Getting started with Math	Lecture: Nernst Equation and GHK-Eqn.
2: Biology	Electrophysiology and data acquisition	Biology Experiment 0: Electrophysiology and DAQ setup	Lecture: Resting mem. potential and GHK-eq.
3: Engineering	Holiday	S-Experiment 1: How do we model a passive cell membrane?	Review for week
4: Biology	Details related to electrophysiology setup	B-Experiment 1: How does the electrophysiology setup work?	Review for week
5: Engineering	Hodgkin-Huxley models	S-Experiment 2: What makes a neuron spike?	Review for week **QUIZ 1**
6: Biology	Action potential	B-Experiment 2: Resting membrane potential	Review for week
7: Engineering	Bursting neurons	B-Experiment 3: Action Potential	Review for week
8: Biology	Voltage gated channels & bursting neurons	S-Experiment 3: What is adaptation? Bursting?	Review for week
9: Engineering	Creating models using NEURON	Description of Modeling and Biology projects – due at end of course	**MIDTERM** **Exam**
10: Biology	Synaptic transmission and processing	B-Experiment 4: Synaptic transmission – Earthworm preparation	Review for week
11: Engineering	Modeling of synapses	S-Experiment 4: How do neurons communicate via synapses?	Review for week
12: Biology	Motor-networks	B-Experiment 5: Local field potentials using Micro-Electrode-Arrays (Virtual Lab)	Review for week
13: Engineering	Motor-networks	S-Experiment 5: How do tetrapods decide when to walk, trot or gallop?	Review for week **QUIZ 2**
14: Engineering	Simple neural networks	S-Experiment 6: Modeling simple neural networks	Biology Project presentations. <u>Report due</u>
15: Engineering	Review	Modeling Project presentations + Report due	*No class* *Finals – next week*

Table 1. Schedule of our interdisciplinary undergraduate neuroscience course, with alternating instruction in Engineering & Biology Labs.

Canvas site described in Figure 1.

Figure 1. The software and accompanying curriculum is freely available through our Canvas site. To download this software, simply go to courses.missouri.edu, scroll to the bottom of the page, and click "Canvas Guest & Visitor Login." Enter username *cns* and password *workshop* to access the materials.

The students tend to be juniors and seniors from engineering (~75%) and biological/psychological sciences (~25%). Our instructional goals focus on gains in learning neuroscience concepts using model-based engineering approaches. The neuroscience concepts include Nernst potentials, resting potential, action potentials, bursting, and synaptic transmission. The model-based engineering content includes concepts from high school calculus (differentiation and integration), first order ordinary differential equations (ODEs), basic electrical circuit elements, derivation of equations using constitutive laws, solution of first order ODEs by hand and using software, first order response characteristics including time constant and gain, and programming concepts. The software NEURON facilitates explanation of the neuroscience concepts using engineering principles.

MATERIALS AND METHODS

The interdisciplinary course (Table 1) was developed as a collaboration between engineering and biology and has been offered annually since 2008. The lectures cover general topics in neurobiology like Nernst potential, action potential, bursting, synapses, and neural networks. Each of these topics is associated with a software and/or wet lab for

students to explore the topics in greater depth. The software labs, the focus here, are performed using tutorials we developed with the open source software NEURON (Carnevale and Hines, 2006) which permits illustration of both neuroscience and engineering concepts (cited earlier) in an integrated manner.

Hands-on activities foster classroom engagement (Wang et al., 2014), which has three dimensions: affective, behavioral, and cognitive. Affective refers to positive emotions, behavioral refers to observable behavior, and cognitive refers to mental effort to understand. For example, one question we pose in lecture is "Can you describe conceptually how the Nernst equation is derived, or describe the ionic flow during an action potential?" The students struggle to respond. Some find this frustrating, but others find it invigorating and cognitively engaging. Active learning in lecture and lab tends to foster attention and achievement (Prince, 2004). All our software tutorials involve active learning, e.g., in the tutorial 'Burster' students explore on their own how each current channel contributes to the overall burst phenomena. Note that these software Labs are also closely linked to the wet-labs (not the focus here) where the students see the waveforms in experiments, but can explore how the waveforms happen only via explorations using the software tutorial.

Students work on the labs and projects in groups. Engineering students are paired with biology/psychology majors which provides opportunities to develop soft skills such as the ability to communicate across disciplines and to work in a team to accomplish complex tasks. Biologists are typically comfortable with neuroscience concepts while engineers are better trained in model-based concepts. However, both groups lack soft skills (Mohan et al., 2010; Franklin et al., 2012) making such an arrangement useful.

Software Labs
There are seven software labs and one 10-week modeling project, all of which use the free and open-source software NEURON. The tutorials are designed to be user-friendly and require minimal or no knowledge about coding. In the 10-week programming project, students learn how the various components of the course fit together via active programming involving a half-center oscillator model. The students first study the biology literature on half-center oscillators such as those that control blood flow in a leech heart. They then develop an electrical circuit model with all components, implement the model in NEURON, and then tune the parameters to reproduce the alternating oscillation pattern in the neurons.

Figure 2 shows what a student would see during one of the software labs. In the upper left panel is the membrane potential, and the three panels below that are voltage gated ion channel conductances, currents, and gating variables respectively. The idea of this particular exercise is to explore how voltage gated ion channels contribute to the action potential.

Software Experiment 1 – Passive Membrane
The purpose of the first software lab is to emphasize the concept of the resting membrane potential and the time constant of a cell membrane. Students are provided with the code to create a single Hodgkin-Huxley-type cell with sodium, potassium, and leak channels. They are then instructed to block the sodium and potassium channels by setting the conductances to zero.

They work through a series of questions that help them investigate the concept of time constant. For instance, they are instructed to plot the membrane voltage for three different current injections and determine the membrane time constant. During this exercise, the first order equation model of a passive membrane is emphasized.

Software Experiment 2 – Action Potential
The second software experiment is used to illustrate the concept of an action potential including how it is generated and the underlying mathematical model that can be used to describe the mechanism. We developed a user-friendly interface where students can manipulate parameters and run the simulation of a spiking Hodgkin-Huxley cell. Visuals of activation/inactivation gates are provided as well as the accompanying ionic currents. This helps students connect the ideas of gates and currents with the resulting action potential.

As a part of the accompanying assignment, students are asked to block only the sodium channel or only the potassium channel and describe the effect on membrane potential. Emphasis is placed on the theory and mathematics of the Hodgkin-Huxley equations. For example, we ask students to calculate the activation probability m and inactivation probability h and then compare their hand calculation with the model values. This is another example of active learning. Students are also asked to explore the relationship between depolarizing current injection and firing frequency.

Software Experiment 3 – Bursting and Adaptation
Once students have a strong grasp of how voltage-gated ion channels function, they move on to the third tutorial, which is about bursting (the phenomenon seen when neurons fire a discrete group of spikes followed by a period of quiescence) and firing adaptation. Students simulate a bursting cell that includes slow and low-threshold calcium channels, potassium channels, and a potassium-activated calcium channel (Liu et al., 1998). Plots of the ionic currents are shown to illustrate the currents responsible for the slow-wave modulation of bursting activities compared to the fast currents responsible for the action potentials themselves.

Like the other labs, the corresponding assignment instructs students to block certain channels by setting the conductance equal to zero and describe the effects seen in both the membrane potential and the underlying currents. In this way, they are able to decouple the slow oscillation that underlies the bursting activity and the fast oscillations that are the action potentials themselves.

Software Experiment 4 – Synapses
The communication between neurons is a critical topic in an introductory neuroscience course. The lecture for this tutorial includes information about different types of synapses (excitatory, inhibitory, non-channel,
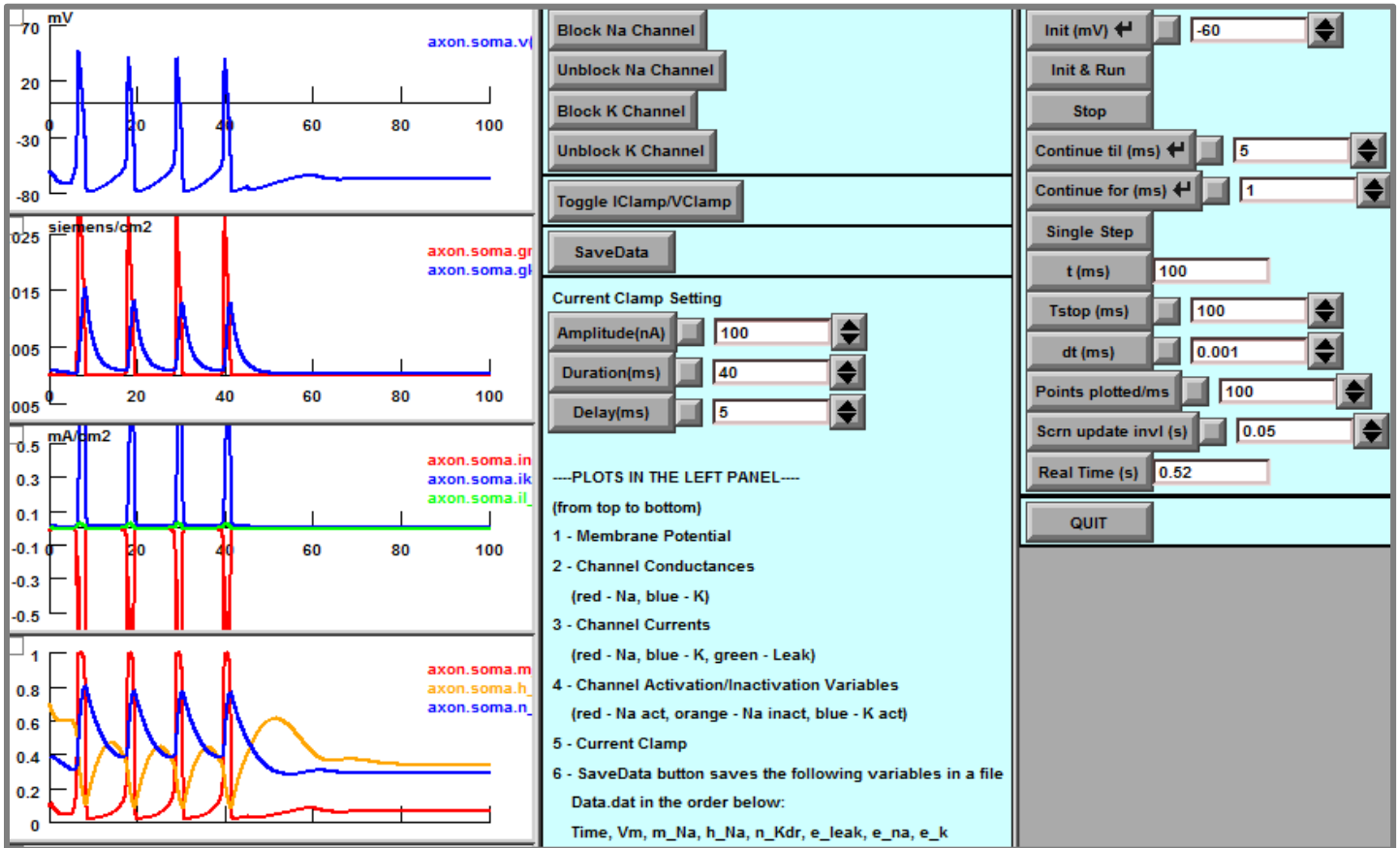
Figure 2. Student's view of the Action Potential tutorial. The Hodgkin-Huxley equations are used to teach the concepts of voltage-dependent ion channels. Students can zoom in on graphs and explore any parameter in the model.
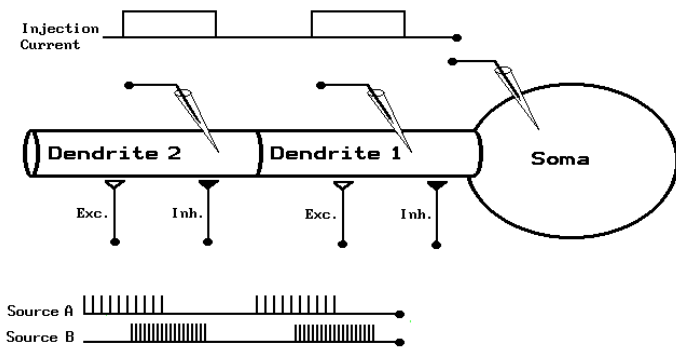

Figure 3. Virtual experimental setup of Software Experiment 4 (Bower and Beeman, 1998). Each synapse receives a user-determined excitatory (*source A*) or inhibitory drive (*source B*).

neuromuscular junction, and electrical synapses), how neurotransmitters bind to receptors, and the concept of summation (temporal and spatial). The software interface consists of a single cell innervated by four synapses: two inhibitory and two excitatory (See Figure 3). The synaptic stimulation parameters of frequency (of excitatory and inhibitory drive) and synaptic weights can be adjusted by the students so they can explore how these interact to determine the spiking characteristics of the neuron via spatial and temporal summation.

Software Experiment 5 – Central Pattern Generator
This tutorial is the first that moves out of the cellular domain and up to the systems level by discussing central pattern generators (such as those that control heart rate, breathing, etc.) where stereotypical rhythmic firing patterns can be generated solely by different synaptic connectivity. The simple network of four identical cells in Figure 4 (of the CPG tutorial) is designed to mimic a hypothetical network in a tetrapod locomotion system including 'walk,' 'trot,' and 'gallop' solely by different combinations of excitatory or inhibitory synaptic connection types and their weights (with no change to the single neuron models). For instance, the 'walk' mechanism shown in Figure 4 begins with stimulation to L2 which propagates to L1, R2, and R1. These four nodes model legs on a tetrapod, with spikes indicating a raised leg. Students need to adjust the configuration of the network so that the expected output pattern is generated.

Software Experiment 6 – Three Neural Networks
In this final software exercise, students explore some of the most common neural networks seen in biology: short-term memory (STM), winner-take-all (WTA), and half-center oscillator (HCO). We provide a GUI for the STM exercise, but for the WTA and HCO, students must learn how to use the NEURON software in the way a researcher would. As mentioned earlier, the HCO project is a ten-week project

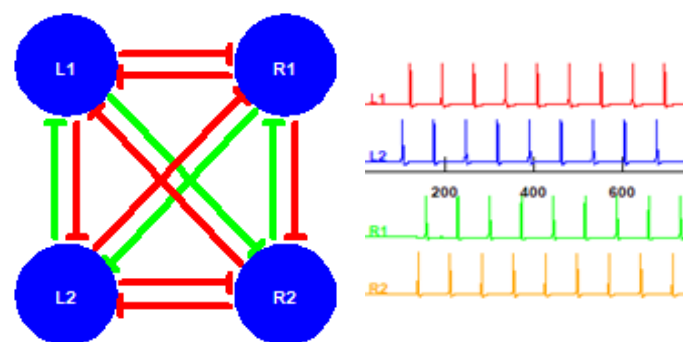involving programming using NEURON, where we actively encourage interactions with other students.

Figure 4. Central Pattern Generator of Software Experiment 5 (Bower and Beeman, 1998). The figure on the right is the membrane potential of each cell.

Biology Labs

Wetlab Experiment 1 – Resting Membrane Potential
The electrophysiology labs start with a lab measuring the resting membrane potential of a crayfish abdominal extensor muscle. Students measure the potential of a cell at different concentrations of extracellular potassium. This allows them to investigate the effect of potassium ion concentration on resting potential, as discussed in the first few lectures. They also get experience using micromanipulators, microscopes, and amplifiers (see Baierlein et al., 2011).

Wetlab Experiment 2 – Action Potential
The second electrophysiology lab has students using a leech preparation to view an action potential (AP) and characterize it, including identifying the different phases of the AP. We handle the dissection ahead of time so that students can focus on performing the sharp intracellular electrode recording. During this lab, we also inject depolarizing current to induce action potentials and hyperpolarizing current to experimentally derive the time constant of a membrane (see Titlow et al., 2013.)

Wetlab Experiment 3 - Synapses
The final wet lab is an experiment using an earthworm to investigate synaptic transmission. The worm is placed on a custom-made bed of pins that are connected to an amplifier made from commonly available hardware components. The output voltage can be viewed using LabView or other software. Earthworm skin is very conductive, so students can view the action potentials that occur from touching the worm and then its behavioral response. This helps students learn about temporal and spatial summation characteristics of synapses. For more details see Kladt et al. (2010).

Low-cost kits have recently been advertised by BackyardBrains.com to perform a variety of neuroscience experiments. They have a well-documented experiment using a cricket and nicotine that we are planning to integrate into our wet-lab sequence. We are also in the process of integrating another recent resource, brainfacts.org of the

Society of Neuroscience, that has interactives and ideas for teaching neuroscience.

OUTCOMES & CONCLUSIONS
The course has experienced an average participation of about 25 students each fall semester. This is the maximum number of students the electrophysiology lab can support. We have also disseminated the materials via summer camps to neuroscience faculty from 2 and 4-year institutions (Nair et al., 2017), and they are beginning to use the software experiments in their own courses.

Through a summer course funded by the NIH BRAIN initiative, we have also used these materials to train PhD students, post docs, and faculty from research universities across the US (Nair et al., 2017). In order to assess the effectiveness of the software and curriculum for learning neuroscience, we have tracked data on pre- and post-test scores, self-reported surveys, and anecdotes from participants in our courses, all of which utilize the same computation-focused curriculum. An analysis of those data is given in Latimer et al. (2018), briefly: Gains in learning for students in our course was quantified based on responses to similar questions on mid-terms and final exam (n=19) and were significantly higher on the final. Their rating of the importance of model-based approaches and the utility of software modules were very high, 3.9/5 and 3.95/5, respectively. The participants in the cited workshops also had similar learning gains and ratings.

Though the challenges of offering an interdisciplinary course are many, the reward is also great. To meet the challenge of understanding the brain, it is important to integrate mathematics and computational concepts into the neuroscience curriculum. The active, hands-on, and flexible learning provided by interactive software tutorials on individual laptops has been found to enhance this important integrated approach to teaching neuroscience. Another positive outcome of the undergraduate course at MU has been the high participation of women. Engineering typically attracts very few women, but we found that 40-50% of the participants in our course were women, primarily from bioengineering and biology.

REFERENCES
Av-Ron E, Byrne JH, Baxter DA (2006) Teaching basic principles of neuroscience with computer simulations. J Undergrad Neurosci Educ 4(2):A40–A52.

Baierlein B, Thurow AL, Atwood HL, Cooper RL (2011) Membrane potentials, synaptic responses, neuronal circuitry, neuromodulation and muscle histology using the crayfish: student laboratory exercises. J Vis Exp (47) pii:e2322, doi:10.3791/2322.

Bish JP, Schleidt S (2008) Effective use of computer simulations in an introductory neuroscience laboratory. J Undergrad Neurosci Educ 6(2):A64–A67.

Bower JM, Beeman D (1998) The book of GENESIS: exploring realistic neural models with the GEneral NEural Simulation System. New York: Springer-Verlag/TELOS

Carnevale NT, Hines ML (2006) The NEURON book. Cambridge, UK: Cambridge University Press.

Franklin CC, Mohan A, Merle D, Lannin J, and Nair SS (2012) Perceptions of professional skills by graduate students - a

comparative study between engineering, education and biology. Int J Eng Educ 28(3):588-598.

Kladt N, Hanslik U, Heinzel HG (2010) Teaching basic neurophysiology using intact earthworms. J Undergrad Neurosci Educ 9(1):A20-A35.

Latimer B, Bergin D, Guntu V, Nair SS (2018) Integrating model-based approaches into neuroscience. Submitted for publication.

Liu Z, Golowasch J, Marder E, Abbott LF (1998) A model neuron with activity-dependent conductances regulated by multiple calcium sensors. J Neurosci 18:2309-2320.

McFarlane HG, Richeimer J (2015) Using the humanities to teach neuroscience to non-majors. J Undergrad Neurosci Educ 13(3):A225–A233.

Modo M, Kinchin I (2011) A conceptual framework for interdisciplinary curriculum design: a case study in neuroscience. J Undergrad Neurosci Educ 10(1):A71–A79.

Mohan A, Merle D, Jackson C, Lannin J, Nair SS (2010) Professional skills in the engineering curriculum. IEEE Transactions on Education 53(4):562-571.

Nair SS, Schulz DJ, Bergin D (2017) NSF Summer Neuroscience Workshop for 2- and 4-year College Faculty. http://engineering.missouri.edu/neuro/outreach/neuroscience-workshop/.

Nair SS, Schulz DJ, McClellan A, Bergin D (2017) NIH BRAIN initiative summer course: interdisciplinary research training in computational neuroscience - for PhD students, post-docs, medical students, residents and faculty. http://engineering.missouri.edu/neuro/outreach/nih-neuroscience-course/.

National Academies (2003) BIO2010: Transforming undergraduate education for future research biologists. Washington, DC: National Academies.

Nichols DF (2015) A series of computational neuroscience labs increases comfort with MATLAB. J Undergrad Neurosci Educ 14(1):A74–A81.

Palmer J (2011) Experts at U. S. symposium urge efforts to encourage and support interdisciplinary research. Washington, DC: American Association for the Advancement of Science.

Prince M (2004) Does active learning work? A review of the research. J Eng Educ 93(3) 223-231.

Ramos RL, Moiseff A, Brumberg JC (2007) Utility and versatility of extracellular recordings from the cockroach for neurophysiological instruction and demonstration. J Undergrad Neurosci Educ 5(2):A28–A34.

Reynolds ER (2012) Creating cross-disciplinary courses. J Undergrad Neurosci Educ 11(1):A72–A75.

Schettino LF (2014) Neurolab: a set of graphical computer simulations to support neuroscience instruction at the high school and undergraduate level. J Undergrad Neurosci Educ 12(2):A123–A129.

Stuart AE (2009) Teaching neurophysiology to undergraduates using Neurons in Action. J Undergrad Neurosci Educ 8(1):A32-A36.

Titlow J, Majeed ZR, Nicholls JG, Cooper RL (2013) Intracellular recording, sensory field mapping, and culturing identified neurons in the leech, Hirudo medicinalis. J Vis Exp (81): e50631 doi: 10.3791/50631.

Wang Z, Bergin C, Bergin DA (2014). Measuring engagement in fourth to twelfth grade classrooms: The Classroom Engagement Inventory. Sch Psychol Q 29(4):517-535. (Supplemental)

Wyttenbach RA, Johnson BR, Hoy RR (2014) Crawdad: an online lab manual for neurophysiology. Sunderland, MA: Sinauer Associates, Inc.

Address correspondence to: Dr. Satish S. Nair, Electrical Engineering and Computer Science, Naka Hall, University of Missouri - Columbia, Columbia, MO 65201. Email: nairs@missouri.edu