

ARTICLE**Learning How to Code While Analyzing an Open Access Electrophysiology Dataset****Ashley Juavinett***Division of Biological Sciences, Neurobiology Section, UC San Diego.*

Conducting neuroscience research increasingly requires proficiency in coding and the ability to manipulate and analyze large datasets. However, these skills are often not included in typical neurobiology courses, partially because they are seen as accessory rather than central, and partially because of the barriers to entry. Therefore, this lesson plan aims to provide an introduction to coding in Python, a free and user-friendly programming language, for instructors and students alike. In this lesson, students edit Python code in the Jupyter Notebook coding environment to interact with

cutting-edge electrophysiology data from the Allen Institute for Brain Science. Students can run their own experiments with these data to compare cell types in mice and humans. Along the way, they gain exposure to Python coding and the role of coding in the field of neuroscience.

Key words: coding, electrophysiology, patch clamping, neuronal diversity, Allen Institute, Python, big data, Jupyter Notebooks

Neuroscience datasets are increasing in size, complexity, and access to the public. Various datasets, such as those collected by the Allen Institute for Brain Science, are available for students to see real neuroscience data and conduct their own analyses. Such data can be used in course-based undergraduate research experiences (CUREs), which have proven to be a powerful means of exposing diverse students to real-world research and inviting these students into genomics and molecular biology (Makarevitch et al., 2015; Kjolvik and Schultheis, 2019). CUREs, especially when implemented with free, publicly-available resources, can provide access to the field of neuroscience to *anyone* who is interested.

Recently, various reports have commented on the need for additional training to enable students to analyze large datasets in neuroscience (Akil et al., 2016; Grisham et al., 2016, 2017). Furthermore, there is an increasing demand for students to acquire coding skills for research and non-related career paths beyond our field. It is imperative that we offer undergraduate experiences in both of these domains – big data analysis and coding – to best prepare them for success beyond college.

Despite the importance of these skills, many students hold preconceived notions about what it means to be a “coder” and feel that – even as undergraduates – they’ve already missed their opportunity to learn how to code (Qin, 2009; Lewis et al., 2016a). Programming classes are not required in many biology or neuroscience programs, and at large universities, these classes can exist in very different academic spheres on campus (Pinard-Welyczko et al., 2017). It is imperative that we open the door to coding for our biology and neuroscience undergraduates, even with small doses of exposure in our classes, which have been shown to improve attitudes towards coding (Du et al., 2016). The activities described here introduce students to the Python coding language, which is freely available, relatively easy to learn, and widely used in neuroscience research and beyond (Muller et al., 2015; Perkel, 2015).

The Allen Institute for Brain Science makes available

multiple datasets for public and educational use. Many educators have used these datasets to provide students with interesting, real world examples of neuroscience data, especially data that is difficult to collect in an undergraduate laboratory (e.g., Ramos et al., 2007; Jenks, 2009; Gilbert, 2018). These datasets come with rich metadata as well as an active community of researchers, which makes the use of this data particularly straightforward.

One of these datasets, The Allen Cell Types Database, (<http://celltypes.brain-map.org>) contains whole-cell *in vitro* patch clamp recordings from both mouse and human cortex. These recordings have been used by the Allen Institute as well as other researchers to demonstrate fascinating differences between mouse and human cells, and well as between cell types within a species (Kalmbach et al., 2018; Gouwens et al., 2019).

There are two primary ways to interact with the Allen Cell Types dataset: through their website, or programmatically, through the Allen Institute’s Software Development Kit (SDK). While interacting with the website enables users to visualize the data as well as analyze individual experiments, the SDK enables custom analyses across many individual experiments within the Allen Cell Types dataset. By interacting with the SDK, users can choose species, cell types, or variables to compare, and perform their own “pseudo-experiments” on the data.

Using these data, instructors can introduce students to the diversity of cell types, even just within the cortex (Ascoli et al., 2008; DeFelipe et al., 2013; Jiang et al., 2015). In typical neuroscience textbooks, students are exposed to just one canonical example of an action potential. With recordings from a variety of different cell types in mice and humans, the Allen Institute data demonstrate the diversity of action potential shapes and therefore the cellular mechanisms that generate action potentials. In addition, the dataset contains many pre-computed features that capture passive properties of the membrane (e.g., input resistance and resting membrane potential) as well as metrics derived from action potential trains elicited from an injection of

Article / Video	Link	Brief Description
Allen Cell Types Database: Understanding the fundamental building blocks of the brain	https://youtu.be/1GWyJxzxqll	Brief introduction to the goals and methods of the Allen Cell Types database
Whole cell patch clamp electrophysiology	https://youtu.be/TUoCQTwevVo	Demonstration of how whole cell patch clamping is performed at the institute, particularly in human tissue
Classification of electrophysiological and morphological neuron types in the mouse visual cortex	https://www.nature.com/articles/s41593-019-0417-0	Primary paper related to this dataset; provides an overview of the dataset as well as scientific findings regarding the classification of neuron types
h-Channels Contribute to Divergent Intrinsic Membrane Properties of Supragranular Pyramidal Neurons in Human versus Mouse Cerebral Cortex	https://www.cell.com/neuron/fulltext/S0896-6273(18)30900-0	Study that uses this data to compare mouse and human recordings, demonstrating differences in the expression of h-channels

Table 1. Possible pre-readings and viewings related to this activity. Table is arranged in increasing complexity. For the primary papers, instructors may want to scaffold these papers or select specific figures, depending on the course and student population.

current (e.g., adaptation).

The *Cell Types Lesson* described here introduces students to this rich dataset, exposing them to real electrophysiology data and the types of metrics that can be used to compare and contrast cell types in the mammalian brain. Throughout the readings, lectures, and activities, students are asked to consider how passive membrane properties and intrinsic electrophysiology relate to the composition of the membrane as well as the function of neurons and ultimately behavior (e.g., Sabatini and Regehr, 1997; Toledo-Rodriguez et al., 2004; Cui et al., 2018). The Allen Cell Types dataset also contains information on whether the cells are spiny or aspiny and can therefore be used to introduce students to the role of dendritic spines.

Within the supplied notebooks, students will reveal how the presence of spines correlates with the shape of the action potential. Students are also invited to choose a feature (e.g., input resistance, see Appendix 1 for all features) and test whether or not this feature is more similar across mouse and human cells or across spiny and aspiny neurons within a species.

Specifically, the learning objectives for the provided materials are as follows:

- Relate the diverse intrinsic electrophysiological features of neurons to their structure and function
- Compare electrophysiological characteristics of neurons in humans and mice
- Practice using Jupyter Notebooks to run and edit Python code
- Develop a sense of belonging and self-efficacy in coding and neuroscience research

Although many students may initially be wary of learning how to code, these lessons provide a scaffolded, neuroscience-specific introduction to programming that can make the benefits of learning how to code more immediately clear.

IMPLEMENTATION

These activities were first implemented in an upper-division Neurobiology Laboratory course taught at a large, R1 university, but can be adapted for different course needs. Students in our course had previous neuroscience content knowledge but almost no coding experience (see RESULTS for details).

Instructors may choose to use the web-based version of the lesson plan, without incorporating any coding activities (Figure 1). Those that opt to include coding activities can choose how much coding skill-building to expect from students. These activities can be used in lecture classes or in laboratory classes – educators can adapt these activities for their classes as they see fit. All of the materials for these lessons can be found at the companion website for this paper (<http://sites.google.com/ucsd.edu/neuroedu>).

After assigning a pre-reading or viewing (see examples in Table 1), instructors should introduce students to several key concepts that intersect with this dataset, specifically: whole-cell patch clamp recording, characterizing passive

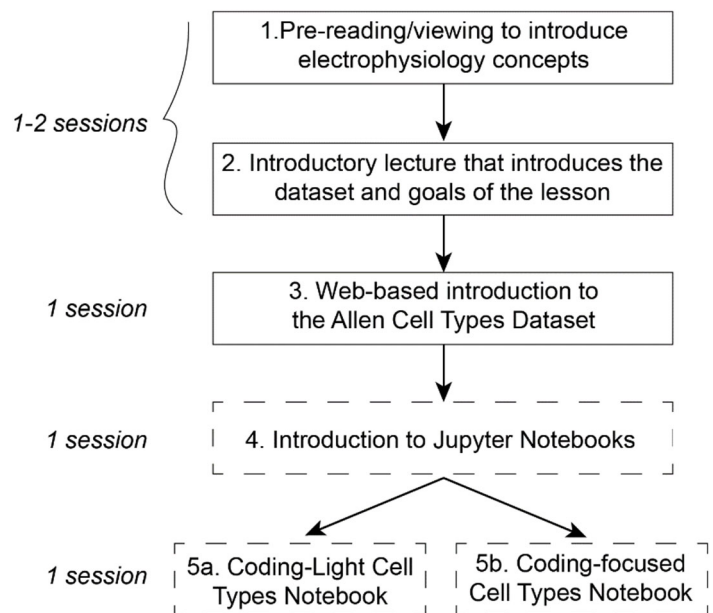


Figure 1. Possible workflows for implementing these activities. Instructors can choose to implement this lesson without any coding components (Steps 1-3). If instructors would like to minimally expose students to coding, they can include the Introduction to Jupyter Notebooks and the Coding-Light Cell Types Notebook (4-5a). To teach students more coding skills, instructors can opt for the coding-focused Cell Types notebook (4-5b). One “session” here is a ~50 minute lecture (or 50 minutes of a lab class).

and stimulus-driven electrophysiology recordings, and dendritic spines. Sample slide decks and additional materials can be found on the accompanying website.

If pursuing the coding versions of this lesson, instructors can also introduce basics of coding (variables, if/else conditionals, and loops) as well as the specific tools (e.g., Jupyter Notebooks) used in this lab. Instructors may also choose to assign coding lessons on free online platforms (e.g., DataQuest, CodeAcademy) before completing the coding activities here.

Web-Based Introduction

With this background, students can then dive into the web-based introduction to the Allen Cell Types dataset (Appendix 2 or companion website). This introduction walks them through the major features of the data and asks the students to look for specific features to compare mouse and human neurons (“Data Notebook,” Appendix 3). Instructors can orient the students to different aspects of the data depending on their course goals. For example, instructors that introduce f/I curves to students may ask them to investigate the f/I curves plotted on the electrophysiology page (Figure 2).

After completing the web-based introduction, students were asked to complete an assessment (“Data Notebook,” Appendix 3) aimed to assess their understanding of the dataset and the methods used to collect the data.

Python Coding in Jupyter Notebooks

The coding aspects of this lesson plan are implemented within Jupyter Notebooks (<https://jupyter.org>). These notebooks are built with an iPython kernel and can be opened and run in a browser window either with a local or cloud installation of Python. Usefully, the notebooks provided here can be entirely run in a web-browser, using either Google Colaboratory (Colab,

<https://colab.google.com>) or Binder (<https://mybinder.org>). Both Colab and Binder create a web-based, online coding environment to run Jupyter Notebooks (see Figure 3 for a comparison of interfaces). If instructors would prefer to run these notebooks locally, students and instructors can install Anaconda (<https://www.anaconda.com/products/individual>) which usefully comes installed with several key packages for scientific computing that are used within these notebooks (e.g., NumPy and Pandas).

The code for all of these notebooks, along with links to load the notebooks in Colab or Binder, can be found on GitHub (<https://github.com/ajuavinett/CellTypesLesson>) or the companion website. For use on a local computer, the notebooks can be acquired using git, or the entire repository of code can be downloaded as a .zip from GitHub.

Recognizing that many biology instructors may themselves be new to coding, the provided Jupyter Notebooks do not assume any coding background and provide an entry level to coding in Python. The “Introduction to Jupyter Notebooks” tutorial teaches students how to write and run code in Jupyter Notebooks, and is a necessary step to completing the subsequent Cell Types Lesson notebook. There are “Task” cells for students to complete throughout the introductory notebook, but there was no assessment directly tied to the “Introduction to Jupyter Notebooks.” This notebook also cites additional resources for students or instructors who would like to learn more about Python coding and using Jupyter Notebooks.

After completing the introductory notebook, students can move on to either of the cell types notebooks. There are two versions of the Cell Types notebook: one which asks students to interact with the dataset programmatically but does not ask them to practice coding (“coding-light,” Figure 1), and one that is coding-focused, in which students are students, as well as student interest. With the coding-light version, students will be exposed to assigning variables,

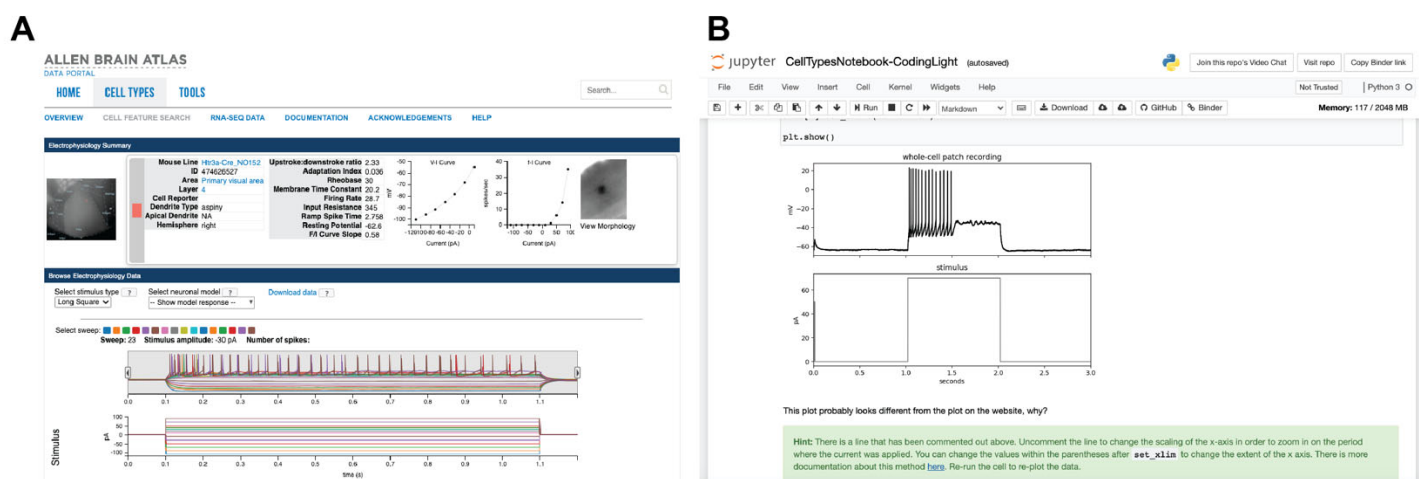


Figure 2. Screenshots of interfaces included in this lesson plan. (A) Sample data on the Allen Institute website, which features a recording from an aspy mouse neuron. The webpage provides information about the neuron, raw sweeps of a recording, as well as some computed electrophysiological features, including an f/I curve (top right). Students can click on “View Morphology” in the top right corner to see the filled cell. (B) Jupyter Notebook interface. Jupyter Notebooks contain both code and descriptive cells, and can show plots (including raw recording traces) within the notebook.

Task: Find the cell specimen ID for the first cell you looked at on the Allen Institute website (hint: it's in the URL), and assign this to `cell_id` below by replacing the `...`. Run the cell.

This might take a minute or two. You should wait until the circle in the upper right is *not* filled to continue.

```
In [ ]: # Enter your cell_id below
cell_id = ...

# Get the electrophysiology (ephys) data for that cell
data = ctc.get_ephys_data(cell_id)
print('Data retrieved')
```

Task: Find the cell specimen ID for the first cell you looked at on the Allen Institute website (hint: it's in the URL), and assign this to `cell_id` below by replacing the `...`. Run the cell.

This might take a minute or two. You should wait until the circle in the upper right is *not* filled to continue.

```
▶ # Enter your cell_id below
cell_id = ...

# Get the electrophysiology (ephys) data for that cell
data = ctc.get_ephys_data(cell_id)
print('Data retrieved')
```

Figure 3. Screenshots of the native Jupyter Notebook or Binder (top) and Google Colaboratory (bottom) code and markdown cells. Note that there is a “Run cell” button to the left of the cell (black circle with white arrow) in the Colab environment. The color of different coding aspects (often called “syntax highlighting”) differs between interfaces as well. Beyond these superficial differences, these interfaces work very similarly.

using Booleans, and editing plots, but will not be asked to generate code on their own. In the coding-focused version, expected to learn basic coding concepts (“coding-focused,” Figure 1). The choice of these two tracks depends on how much time there is to dedicate to troubleshooting with students manipulate more code and also learn the basics of plotting both timeseries data and scatterplots in Python.

Within both notebooks, students are introduced to scatterplots as a means of comparing the waveforms of spiny and aspiny cells. Specifically, students are led through an activity which asks them to plot the upstroke-downstroke ratio of the cell compared to the trough (after-hyperpolarization depth), which demonstrates clear differences in the waveforms of spiny (putative excitatory) and aspiny (putative inhibitory) cells. This demonstrates that neurons can have dramatically different looking action potentials and illustrates that action potential shape can correspond to other features of the cell.

In the second half of the Cell Types Notebook, students can choose from pre-computed electrophysiology features contained in the dataset to compare between mouse and human cells and effectively run their own experiment. These features are either passive membrane properties (e.g., input resistance, resting membrane potential), subthreshold - features (e.g., sag), or are derived from either single spikes (e.g., peak, afterhyperpolarization trough) or from action potential trains elicited by current injections (e.g., adaptation, latency). Many of the variable labels for these features are opaque; see Appendix 1 for a glossary of features that may be of interest. Depending on the existing course content, instructors may choose one or two of these to illustrate the point that intrinsic electrophysiological

properties determine a given neuron’s response dynamics and determine the way it can encode information about external stimuli.

For the coding-light Cell Types Lesson, students are asked to save the plots created throughout the Jupyter Notebook (Figure 4). For the coding-focused Lesson, we also asked students to complete a corresponding Jupyter Notebook that was automatically graded using a tool called NBGrader (<https://nbgrader.readthedocs.io/>). With small classes, these coding assessments could easily be manually graded. Interested instructors should contact the author for assessment materials related to this lesson plan.

RESULTS

This lesson plan has been implemented in various iterations with five different neurobiology lab classes, ranging in size from 20-37 students. This neurobiology lab course typically includes additional coding activities after these lessons, which serve as the introduction to coding and the use of big data in neuroscience. We assessed attitudes towards this lesson in two different classes, with 10 respondents from each class.

Before completing these lessons, most of our students (75%, $n=20$) had no experience coding in Python, and none of the students (0%, $n=20$) had experience working in Jupyter Notebooks. Even still, we found that with careful explanation and tutorials about how to manipulate Python code in Jupyter Notebooks, students quickly gained proficiency in working through these materials.

After completing either the coding-light or coding-focused version of the Cell Types Notebook, two different

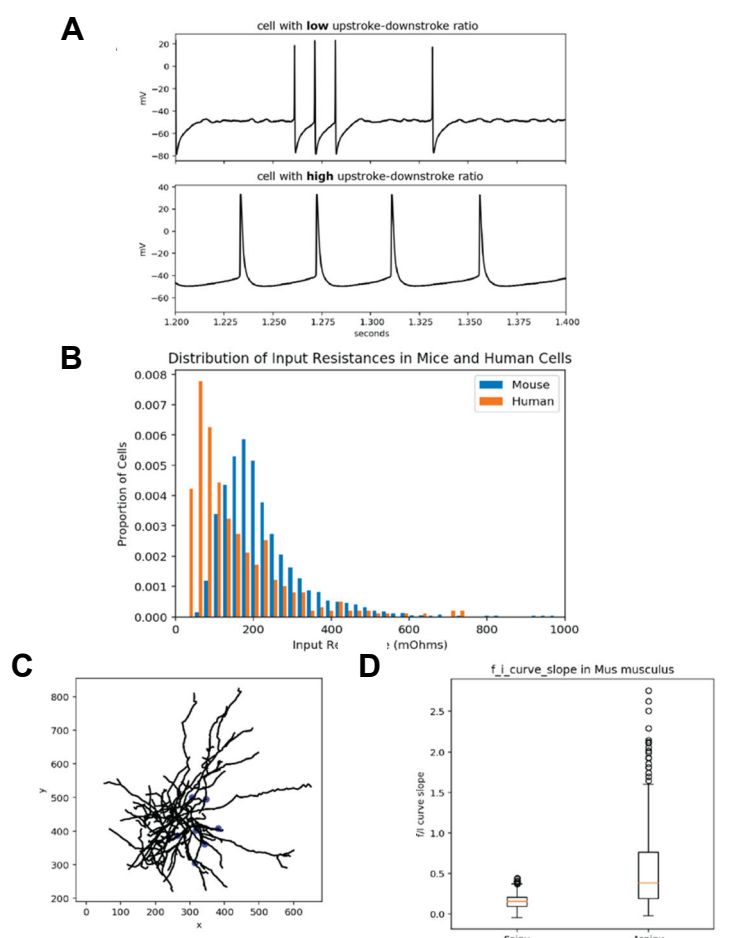


Figure 4. Sample plots that can be generated with the cell types notebooks. Within the notebooks, students can (A) observe the differences in action potential shapes based on the ratio of the action potential upstroke to downstroke (B) generate a 2D image of the morphology of the same cell (C) compare a chosen parameter (here, input resistance) between mouse and human cells and (D) compare a chosen parameter (here, slope of the f/l curve between spiny and aspiny neurons).

classes were asked to report on what they liked most and least about the lesson (Figure 5). Regardless of the lesson's coding level, most of the respondents reported liking the exposure to Python coding and Jupyter Notebooks, and many appreciated that this lesson involved real data (Figure 5A). With the coding-light version of the laboratory, students largely fell into two camps: students who wished the lesson contained more coding training, and students who were content with the amount of coding involved (Figure 5).

In open-ended responses to the coding-light lesson, students expressed a range of opinions. In line with their survey responses (Figure 5), several expressed frustration about their first attempt at coding, whereas others expressed frustration that there wasn't much depth. One student expressed initial frustration with the Jupyter interface: *"The layout of Jupyter notebook was tricky to figure out at first."* Still, many students recognized the utility of learning how to code, and the difficulty that accompanies learning any new skill. For example, one student reported, *"It was more tedious than I thought it would be. The program is very detail-oriented which was, at times, difficult but I*

learned a lot."

On the whole, many students felt that they had learned some programming and several reported an increase in interest in coding. In response to the question, "Any other feedback you'd like to provide about this lecture or lab?" one student wrote, *"It was really cool to be introduced to coding in a way that's applicable to neurobiology/neurophysiology. I've always wanted to learn how to code but it would take away time from my classes, but now that it's incorporated into one of my classes it was a really cool and enjoyable experience."*

DISCUSSION

This article describes a lesson plan and set of resources for instructors to empower their students to learn a bit of coding and interact with cutting-edge neuroscience data. As programming is increasingly embedded in neurobiology and adjacent fields, it is increasingly important to expose our students to these skills. This lesson plan builds on a growing body of ideas from other educators on how to integrate computational modeling and big, open access data analysis in our classrooms (Ramos et al., 2007; Grisham et al., 2016, 2017; Gilbert, 2018; Gorur-Shandilya et al., 2018).

Implementation of These Lessons

These resources are modular and can be implemented in a variety of classrooms by instructors with a variety of backgrounds (Figure 1). Given ample time and student enthusiasm (instructors should consider asking students ahead of time how interested they are in coding), it is certainly possible to introduce even a completely naïve coding group to some elementary concepts via these lesson plans.

The ideal implementation of these lessons depends on the instructor's background and programming resources available on campus. At our university, the use of a campus JupyterHub has proven exceptionally helpful for implementing these and similar coding lessons for both biology and computer science classes. As others have noted, JupyterHub can usefully be configured with a specific version of Python and with dependencies installed (Laderas, 2018, <https://laderast.github.io/2018/01/17/what-we-learned-teaching-python-to-neuroscience-students/>).

However, newer tools such as Google Colaboratory and Binder have proven to be very powerful click-and-go options for Jupyter Notebooks, and can be used without any on campus support.

Reflections on Student Responses

Based on our cohorts of students that completed these activities, students were overall very positive about their experience, the introduction to these electrophysiology principles, and the exposure to coding in Python (Figure 5). After reflecting on our final course evaluations, it was clear that these activities, along with several similar lessons, helped our students develop a better understanding of the diversity of neurons in the brain, the means by which we can characterize them, and the use of coding as a tool in neuroscience research.

Although some students found the programming

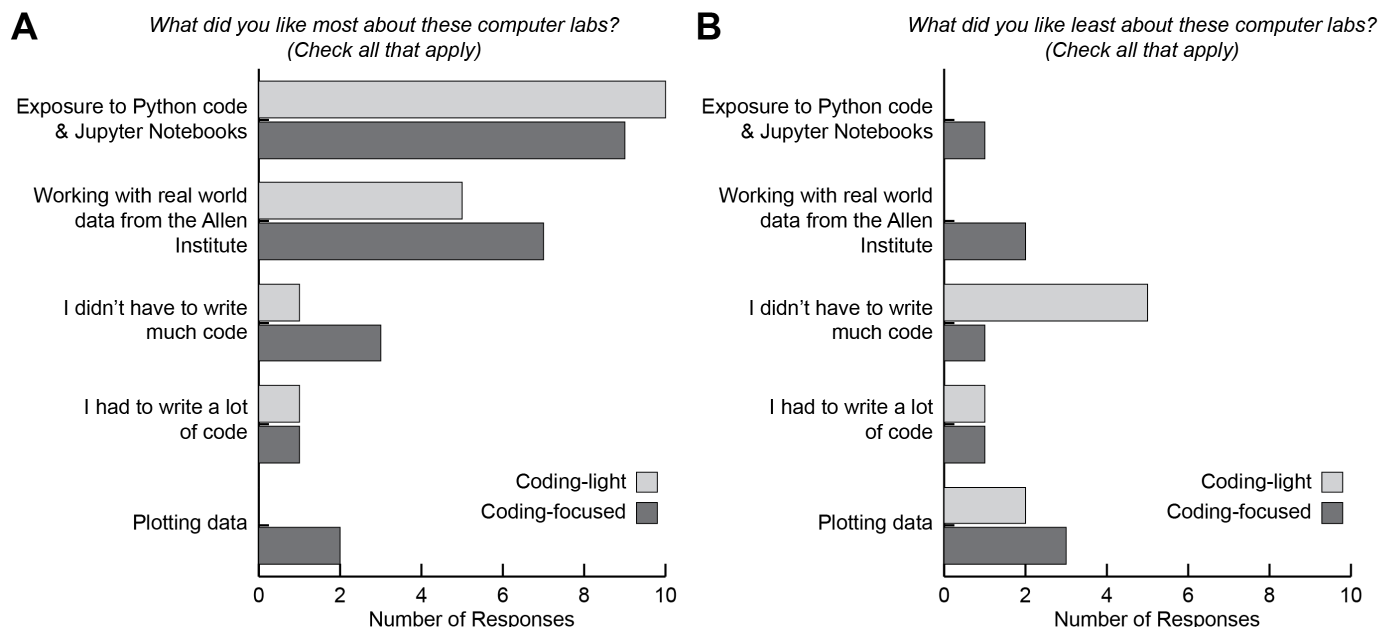


Figure 5. Two different groups of students ($n = 10$ each) received different versions of the Cell Types Lesson (coding-light and coding-focused) and were asked which aspects of the lab they liked (A) or disliked (B). Students could choose multiple options for each question, as well as provide open-ended responses.

environment “tricky at first,” they quickly developed proficiency. With both coding-light and -focused versions of this activity, we observed students’ perceptions about coding change, largely in a positive direction, but also in unexpected ways. One student’s comments that the notebooks were “tedious” highlights a frequent mismatch between student expectations and the reality of coding: many students expect coding to be math-heavy and full of jargon, but *actual* coding requires sometimes “tedious” attention to detail and debugging with limited *actual math*.

After initial cohorts of students had completed the coding-light activities, it was clear that many students had hoped for more coding challenges. We therefore developed the coding-focused track, not without its own difficulties. Students in this track were varied in their comfort with the lesson. Several students reported feeling frustrated with the plotting aspects of the lesson, likely because this is where the majority of the challenge in these notebooks originated. Further, many students have experience plotting data in a graphical interface such as Microsoft Excel, and therefore may be frustrated by the seemingly opaque way plots are manipulated in Python.

On the whole, students seemed to appreciate the exposure to “real world” neuroscience, and several students in the class went on to pursue internships at the Allen Institute. With real data that is actively being collected and analyzed by real neuroscientists, this lesson plan enables students to experience real neuroscience research.

Extensions of These Materials

Given the number of pre-computed features in the Cell Types dataset, there are many neuroscience principles that

can be taught with these resources. For example, instructors can explain the implications of the adaptation index and interspike interval for sensory coding or learning (e.g., Reich et al., 2000), emphasize the importance of injecting “noisy” in addition to square wave currents, or discuss the implication of other morphological features, such as the cell’s surface area. While cell surface area and rheobase are not readily obtained via the primary lesson plan described here, there is an additional notebook on GitHub (CompareCellFeatures.ipynb) for students to work with a prepared dataset to compare surface area, input resistance, tau, and the rheobase of mouse and human cells.

Lastly, the Allen Institute has recorded from cells from over a hundred different transgenic mouse lines, which selectively mark cells based on their expression of a particular promoter (Madisen et al., 2010; Harris et al., 2014). This genetic strategy can be used to selectively mark excitatory or inhibitory neurons, or neurons in a specific layer of the cortex. As an extension of the notebooks here, students could compare specific populations of transgenic cell types. For example, students could compare PV+ inhibitory cells to Emx1+ excitatory cells (Gorski et al., 2002). There are additional Jupyter Notebooks to accomplish this type of analysis on GitHub.

Teaching Discipline-Based Coding

Just as important as the tools, however, is an understanding of the preconceived notions about coding that our students bring into our classrooms (Qin, 2009; Donohue et al., 2012; Du et al., 2016; Lewis et al., 2016b; Gaudier-Diaz et al., 2019). While this work begins to assess students’ attitudes

towards coding, much more work needs to be done with these and other teaching materials to fully understand how we should teach discipline-based coding to diverse learners who have categorically decided that they are not “math people.” Additional work should be done to assess students’ sense of belonging and self-efficacy in the intersection of neuroscience and computation, as has been done in other fields of biology and physics (Hazari et al., 2010; Trujillo and Tanner, 2014). As the job force continues to move towards big data and analytics, understanding the perspectives of our students – particularly those from under-resourced backgrounds – is not only a question of neuroscience education, but also of equity.

Like learning any new language, learning how to code takes time and patience. As with other complex new skills, working deliberately through materials and clearly setting expectations for students will ease concerns and enable learning (Trujillo and Tanner, 2014). However, introducing coding within the context of neurobiology can invite additional students into an otherwise very daunting field. It is important that instructors approach these lessons with care, explaining to students that errors are inevitable and that the computer is frustratingly literal, but ultimately, these may be important skills to learn in our ever-evolving and expanding field.

REFERENCES

- Akil H, Balice-Gordon R, Cardozo DLL, Koroshetz W, Posey Norris SMM, Sherer T, Sherman SM, Thiels E (2016) Neuroscience Training for the 21st Century. *Neuron* 90:917–926.
- Ascoli GA et al. (2008) Petilla terminology: Nomenclature of features of GABAergic interneurons of the cerebral cortex. *Nat Rev Neurosci* 9:557–568.
- Cui Y, Yang Y, Ni Z, Dong Y, Cai G, Foncelle A, Ma S, Sang K, Tang S, Li Y, Shen Y, Berry H, Wu S, Hu H (2018) Astroglial Kir4.1 in the lateral habenula drives neuronal bursts in depression. *Nature* 554:323–327.
- DeFelipe J et al. (2013) New insights into the classification and nomenclature of cortical GABAergic interneurons. *Nat Rev Neurosci* 14:202–216. Available at <http://www.nature.com/doi/10.1038/nrn3444>.
- Donohue SE, Darling EF, Mitroff SR (2012) Links between multisensory processing and autism. *Exp Brain Res* 222:377–387. Available at <http://link.springer.com/10.1007/s00221-012-3223-4>.
- Du J, Wimmer H, Rada R (2016) Hour of Code™: Can it change students’ attitudes toward program-ming? *J Inf Technol Educ Innov Pract* 15:52–73. Available at <http://www.jite.org/documents/Vol15/JITEv15IIPp053-073Du1950.pdf>.
- Gaudier-Diaz MM, Sinisterra M, Muscatell KA (2019) Motivation, Belongingness, and Anxiety in Neuroscience Undergraduates: Emphasizing First-Generation College Students. *J Undergrad Neurosci Educ* 17:A145–A152. Available at <http://www.ncbi.nlm.nih.gov/pubmed/31360130%0Ahttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC6650254>.
- Gilbert TL (2018) The Allen Brain Atlas as a Resource for Teaching Undergraduate Neuroscience. *J Undergrad Neurosci Educ* 16:A261–A267. Available at <http://www.brain-map.org/tutorials/index.html>.
- Gorski JA, Talley T, Qiu M, Puelles L, Rubenstein JLR, Jones KR (2002) Cortical excitatory neurons and glia, but not GABAergic neurons, are produced in the Emx1-expressing lineage. *J Neurosci* 22:6309–6314.
- Gorur-Shandilya S, Hoyland A, Marder E (2018) Xolotl: An intuitive and approachable neuron and network simulator for research and teaching. *Front Neuroinform* 12. Available at <https://pubmed.ncbi.nlm.nih.gov/30534067/>.
- Gouwens NW et al. (2019) Classification of electrophysiological and morphological neuron types in the mouse visual cortex. *Nat Neurosci* 22:1182–1195.
- Grisham W, Brumberg JC, Gilbert T, Lanyon L, Williams R, Olivo R (2017) Teaching with Big Data: Report from the 2016 Society for Neuroscience Teaching Workshop. *J Undergrad Neurosci Educ* 16:68–76.
- Grisham W, Lom B, Lanyon L, Ramos R (2016) Proposed Training to Meet Challenges of Large-Scale Data in Neuroscience. *Front Neuroinform* 10:1–6. Available at <http://journal.frontiersin.org/Article/10.3389/fninf.2016.00028/abstract>.
- Harris J a., Hirokawa KE, Sorensen S a., Gu H, Mills M, Ng LL, Bohn P, Mortrud M, Ouellette B, Kidney J, Smith K a., Dang C, Sunkin S, Bernard A, Oh SW, Madisen L, Zeng H (2014) Anatomical characterization of Cre driver mice for neural circuit mapping and manipulation. *Front Neural Circuits* 8:1–16. Available at http://www.frontiersin.org/Neural_Circuits/10.3389/fncir.2014.00076/abstract.
- Hazari Z, Sonnet G, Sadler PM, Shanahan MC (2010) Connecting high school physics experiences, outcome expectations, physics identity, and physics career choice: A gender study. *J Res Sci Teach* 47:978–1003. Available at <https://onlinelibrary.wiley.com/doi/full/10.1002/tea.20363>.
- Jenks BG (2009) A Self-Study Tutorial using the Allen Brain Explorer and Brain Atlas to Teach Concepts of Mammalian Neuroanatomy and Brain Function. *J Undergrad Neurosci Educ* 8:A21-5. Available at <http://www.ncbi.nlm.nih.gov/pubmed/23493964>.
- Jiang X, Shen S, Cadwell CR, Berens P, Sinz F, Ecker AS, Patel S, Tolias AS (2015) Principles of connectivity among morphologically defined cell types in adult neocortex. *Science* (80-) 350:aac9462–aac9462. Available at <http://www.sciencemag.org/cgi/doi/10.1126/science.aac9462>.
- Kalmbach BE et al. (2018) h-Channels Contribute to Divergent Intrinsic Membrane Properties of Supragranular Pyramidal Neurons in Human versus Mouse Cerebral Cortex. *Neuron* 100:1194-1208.e5.
- Kjelvik MK, Schultheis EH (2019) Getting Messy with Authentic Data: Exploring the Potential of Using Data from Scientific Research to Support Student Data Literacy Gardner S, ed. *CBE—Life Sci Educ* 18:es2. Available at <https://www.lifescied.org/doi/10.1187/cbe.18-02-0023>.
- Lewis CM, Anderson RE, Yasuhara K (2016a) “i Don’t Code All Day”: Fitting in computer science when the stereotypes don’t fit. In: *ICER 2016 - Proceedings of the 2016 ACM Conference on International Computing Education Research*, pp 23–32. New York, NY, USA: Association for Computing Machinery, Inc. Available at <https://dl.acm.org/doi/10.1145/2960310.2960332>.
- Lewis CM, Anderson RE, Yasuhara K (2016b) “i Don’t Code All Day”: Fitting in computer science when the stereotypes don’t fit. In: *ICER 2016 - Proceedings of the 2016 ACM Conference on International Computing Education Research*, pp 23–32. New York, NY, USA: Association for Computing Machinery, Inc. Available at <https://dl.acm.org/doi/10.1145/2960310.2960332>.
- Madisen L, Zwingman TA, Sunkin SM, Oh SW, Zariwala HA, Gu H, Ng LL, Palmiter RD, Hawrylycz MJ, Jones AR, Lein ES, Zeng H (2010) A robust and high-throughput Cre reporting and characterization system for the whole mouse brain. *Nat Neurosci* 13:133–140.
- Makarevitch I, Frechette C, Wiatros N (2015) Authentic research

- experience and “big data” analysis in the classroom: Maize response to abiotic stress. *CBE Life Sci Educ* 14:1–12.
- Muller E, Bednar JA, Diesmann M, Gewaltig M-O, Hines M, Davison AP (2015) Python in neuroscience. *Front Neuroinform* 9:11. Available at <http://journal.frontiersin.org/article/10.3389/fninf.2015.00011/abstract>.
- Perkel JM (2015) Programming: Pick up Python. *Nature* 518:125–126.
- Pinard-Welyczko KM, Garrison ACS, Ramos RL, Carter BS (2017) Characterizing the Undergraduate Neuroscience Major in the U.S.: An Examination of Course Requirements and Institution-Program Associations. *J Undergrad Neurosci Educ* 16:A60–A67. Available at <http://www.ncbi.nlm.nih.gov/pubmed/29371843>.
- Qin H (2009) Teaching computational thinking through bioinformatics to biology students. In: *SIGCSE'09 - Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, pp 188–191. New York, New York, USA: ACM Press. Available at <http://portal.acm.org/citation.cfm?doid=1508865.1508932>.
- Ramos RL, Smith PT, Brumberg JC (2007) Novel in silico Method for Teaching Cytoarchitecture, Cellular Diversity, and Gene Expression in the Mammalian Brain. *J Undergrad Neurosci Educ* 6:A8–A13. Available at <http://www.ncbi.nlm.nih.gov/pubmed/23493835>.
- Reich DS, Mechler F, Purpura KP, Victor JD (2000) Interspike intervals, receptive fields, and information encoding in primary visual cortex. *J Neurosci* 20:1964–1974. Available at <https://www.jneurosci.org/content/20/5/1964>.
- Sabatini BL, Regehr WG (1997) Control of neurotransmitter release by presynaptic waveform at the granule cell to Purkinje cell synapse. *J Neurosci* 17:3425–3435. Available at <https://www.jneurosci.org/content/17/10/3425>.
- Toledo-Rodriguez M, Blumenfeld B, Wu C, Luo J, Attali B, Goodman P, Markram H (2004) Correlation maps allow neuronal electrical properties to be predicted from single-cell gene expression profiles in rat neocortex. *Cereb Cortex* 14:1310–1327.
- Trujillo G, Tanner KD (2014) Considering the Role of Affect in Learning: Monitoring Students' Self-Efficacy, Sense of Belonging, and Science Identity. *CBE Life Sci Educ* 13:6. Available at <https://www.lifescied.org/doi/full/10.1187/cbe.13-12-0241>.

Received June 16, 2020; accepted September 9, 2020.

Acknowledgements. I would like to thank Kaitlyn Casimo and Saskia de Vries (Allen Institute for Brain Science) for help with using these datasets and for useful feedback on the teaching materials, as well as Catherine Hicks (Khan Academy) and Katie Petrie (UC San Diego) for helpful comments on this manuscript.

Address correspondence to: Dr. Ashley Juavinett, Division of Biological Sciences, Neurobiology Section, UC San Diego, York Hall 4070C, Mail Code 0355, 9500 Gilman Drive, La Jolla, CA, 92037. Email: ajuavinett@ucsd.edu

Copyright © 2020 Faculty for Undergraduate Neuroscience

www.funjournal.org

APPENDIX 1: GLOSSARY OF FEATURES AVAILABLE VIA THE `get_ephys_features()` METHOD

Note that several of the variable names contain additional information about whether the measurement is in time (`_t`) or voltage (`_v`) and in response to which type of stimulus (`_short_square` = 3ms; `_long_square` = 1s; or `_ramp` = 25 pA/1s. “Trough” was chosen rather than “after-hyperpolarization” since in many cases the membrane potential did not hyperpolarize below the baseline membrane potential. Additional features can be computed from the data using the `EphysSweepFeatureExtractor`. See <https://allensdk.readthedocs.io> as well as the Allen Institute Whitepaper for more information: http://help.brain-map.org/download/attachments/8323525/CellTypes_Ephys_Overview.pdf.

Variable	Description	Related neuroscience principles
<code>adaptation</code>	The rate at which firing speeds up or slows down during a stimulus, defined as: $\frac{1}{N-1} \sum_{n=1}^{N-1} \frac{ISI_{n+1} - ISI_n}{ISI_{n+1} + ISI_n}$ where N is the number of interspike intervals (ISIs) in the sweep	Learning and memory, integration of inputs in single cells, sensory adaptation, habituation vs. adaptation, information transmission
<code>avg_isi</code>	The mean value of all interspike intervals in a sweep	Mechanisms of an action potential, integration of inputs in single cells, information transmission
<code>fast_trough</code>	timestamp (<code>_t</code>) or voltage (<code>_v</code>) of the trough within in the interval 5 ms after the peak in response to a short square stimulus (<code>_short_square</code>), a long square (<code>_long_square</code>), a ramp (<code>_ramp</code>).	Mechanisms of an action potential, the role of ion channels in the shape of an action potential
<code>f_i_curve_slope</code>	slope of the curve between firing rate (f) and current injected; see the Allen Institute Whitepaper for details	The relationship between input and firing rate
<code>input_resistance_mohm</code>	input resistance of the cell, in mega ohms	Passive membrane properties, Ohm's law
<code>latency</code>	time for the stimulus onset to the threshold of the first spike	Integration of inputs in single cells, sensory integration
<code>peak</code>	timestamp (<code>_t</code>) or voltage (<code>_v</code>) of the maximum value of the membrane potential during the action potential (i.e., between the action potential's threshold and the time of the next action potential, or end of the response) in response to a (<code>_short_square</code>), a long square (<code>_long_square</code>), a ramp (<code>_ramp</code>) stimulus	Mechanisms of an action potential, the role of ion channels in the shape of an action potential
<code>sag</code>	measurement of sag, or the return to steady state divided by the peak deflection	I_h currents, subthreshold properties
<code>slow_trough</code>	timestamp (<code>_t</code>) or voltage (<code>_v</code>) of the membrane potential in the interval between the peak and the time of the next action potential in response to a short square stimulus (<code>_short_square</code>), a long square (<code>_long_square</code>), a ramp (<code>_ramp</code>)	Mechanisms of an action potential, the role of ion channels in the shape of an action potential
<code>tau</code>	membrane time constant, in ms	Passive membrane properties, Ohm's law, membrane time constants, capacitance
<code>upstroke_downstroke_ratio</code>	the ratio between the absolute values of the action potential peak upstroke and the action potential peak downstroke.during a short square (<code>_short_square</code>), a long square (<code>_long_square</code>), a ramp (<code>_ramp</code>) stimulus	Mechanisms of an action potential, the role of ion channels in the shape of an action potential
<code>vrest</code>	resting membrane potential, in mV	Mechanisms of establishing and maintaining a resting membrane potential, role of ions and channels in the resting membrane potential

APPENDIX 2: ACCESSING THE ALLEN CELL TYPES DATASET ON THEIR WEBSITE

Before we go behind the scenes, we'll look at a few cells on the Allen Institute for Brain Science's online interface to get a feel for what the data looks like.

1. Go to <http://celltypes.brain-map.org/>. Read through "About Electrophysiology" and "About Morphology" to answer questions 1 and 2 in the [Data Notebook](#).
2. Scroll down to "Download Single Cell Data and Models" and the section "Morphology and Electrophysiology" and click on Electrophysiology Page. This page shows all of the electrophysiology data for one sample cell.
3. On the top, you'll see a Mouse Line, Brain Area, and Layer where this cell comes from. Note them in Table 1.
4. The Mouse Line tells us the Cre-driver line that was used — in other words, the cells that the Allen Institute researchers targeted had that gene, and therefore they also expressed Cre-recombinase. Go to <https://www.ncbi.nlm.nih.gov/gene/> and search for the name of this gene (without -Cre) to answer question 3. This page also gives us some key details about the cell. Note the resting membrane potential in Table 1.
5. Click through the stimulus sweeps (the colored boxes) to find the first one that elicited an action potential. Record the minimum stimulus amplitude required to elicit an action potential in this cell in Table 1. **Note:** You should notice that this value is either very close or identical to the rheobase of the cell, as reported on the table on the top. As a reminder, the **rheobase** is defined as the minimum current needed to elicit an action potential. When the current is below the rheobase, an action potential will never occur regardless of the length of the stimulation.
6. Click through to a stimulus sweep with a higher current injection. Does the cell adapt to the stimulus? In other words, does the space between spikes increase? Is there a metric here that would help quantify this? Record that metric in Table 1.
7. Use the dropdown menu on the left to change the stimulus type to "Short square." Look at how the current injection trace changes — now, it's a shorter pulse. Record the minimum current needed to elicit an action potential at this stimulus, and answer question 4.

Wondering what these 'square' pulses mean? Square simply describes the shape of the stimulus, and it's a common way to inject current into cells. There are more details on the stimuli the Allen used [here](https://raw.githubusercontent.com/ajuavinett/CellTypesLesson/master/docs/stimuli.png): <https://raw.githubusercontent.com/ajuavinett/CellTypesLesson/master/docs/stimuli.png>.
8. Use the dropdown menu on the left to change the stimulus type to "Noise." Take a look at the current injection trace to get an idea of what this stimulus looks like. Note the differences between the cell's response to this stimulus versus the square wave pulses. Answer question 5.
9. Click on View Morphology on the right, to check out the morphology of this cell. Take note of the distribution of the axon and the dendrites. Close this window.
10. Click on Cell Feature Search on the top right corner. Here, you can look through other cells in this dataset. Find a cell from human tissue that is in the **same Layer** and has the **same Dendrite Type** as the first cell you looked at. Use the data on that page to fill out Table 1.

APPENDIX 3: DATA NOTEBOOK

1. Which method are the Allen researchers using to record from cells?
2. What are the cells filled with to visualize their morphology?
3. Which neurotransmitter receptor does this gene code for?
4. Why would a longer stimulus require less current to elicit an action potential?
5. Why would you want to inject a noisy current into the cell? *(Not like 60 Hz noise, what we've been talking about before. "Noisy" current here is random, fluctuating around a certain value but not a perfect sine wave. To answer this question, consider what happens in a real neural circuit. Is the input as simple as a square wave?)*

Table 1.

Property	Mouse	Human
Cre Line		N/A
Area		
Layer		
Dendrite type		
Resting potential		
Minimum current to elicit an AP with the long square wave		
Minimum current to elicit an AP with a short square wave		
Adaptation index		