

Computational Neuroscience

Psychology 5270 Spring 2024

January 12, 2024

Course Information

- Class times: TTh 9:30-10:45
- Course websites:
 - [github/comp-neurosci](#)
 - [canvas](#)
- Slack group: uva-comp-neuro
 - [Signup](#)
- Class location: Ridley Hall 123
- Instructor: C Daniel Meliza (cdm8j)
 - Office Hours: by appointment
- Teaching Assistant: Jingrun Lin (jl5xg)
 - Office Hours: Th 11:30-12:30 [zoom](#); Fr 3-4 Gilmer 226J
- Final: in-class presentation

Course Description

To survive and reproduce in a complex and variable physical world, animals have to extract information from sensory inputs, make decisions about how to allocate their resources, and perform tightly choreographed sequences of motor behaviors. Brains have evolved to perform these tasks reliably, flexibly, and efficiently. **Our goal is to understand how these computations are implemented by circuits of neurons.**

The challenge we face is that brains themselves are enormously complex. Even relatively “simple” non-human animals have millions of neurons and billions of connections. Just observing a small fraction of the brain in action can generate enormous quantities of data. Voltage from an intracellular electrode is sampled 40,000 times a second, and EEGs are recorded on multiple channels sampled up to 2,000 times a second. That means almost 2.5 million data points in one minute! With so much data, how can we possibly find the information in all the noise?

The approach we’ll use here is **computational modeling**, which means building toy systems that capture some aspect of how a particular system in the brain is performing its function. As we’ll discuss, there are many different kinds of models and ways of building them, but one of the most useful functions of a model is that it can be used like a lens that can reveal structure in our data and help us filter out the noise.

This course will introduce you to foundational principles of computational modeling in the brain using a hands-on approach with exercises that will also build your expertise in using scientific programming to implement models and apply them to real-world data.

Course Objectives

By the end of the semester, you will be able to:

- identify different types of neuroscience data and the appropriate methods for analyzing them

- understand linear time series models and how to use them to analyze and predict neural activity
- use Python and its numerical libraries to process, analyze, and visualize neuroscience data
- think through computational problems logically and outline the steps of an analysis
- work as part of a team to write and talk about computational research
- feel confident in your ability to use computers in your research and find resources to answer future questions

Course Schedule

This schedule is subject to change based on the interests of the class and our progress through the material, and some readings have yet to be chosen. If there's something you want to cover that isn't listed, let me know! In general, Tuesdays will be used for lecture-based instruction, and Thursdays for in-class work. Readings must be completed before class on Tuesday. Assignments are distributed on Rivanna and must be completed before class on Thursday the following week.

Week	Topic	Readings	Assignments
1/18	Introduction		
1/23	What is computation?	Ewert (1974), McNaughton & Morris (1987)	Pen and Paper Networks
1/30	Working with time series data	Wilson (2014)	Time-varying Data
2/6	I/O for neuroscience; spike train stats	Dayan and Abbott § 1.1,1.2	Data and Spike Stats
2/13	Linear time-invariant systems	Dayan and Abbott § 1.3-1.6	LTI Systems
2/20	Estimating parameters and comparing models	Dayan and Abbott Ch 2	Receptive Fields
2/27	Dimensional reduction	Cunningham and Yu (2014)	Dimensional Reduction
3/5	Spring recess (no class)		
3/12	Encoding vs Decoding models	Holdgraf et al (2017)	Data Exploration I
3/19	Project journal club		
3/26	Dynamical systems theory	Izhikevich (Ch 3–4)	Dynamical Systems
4/2	Phenomenological dynamical models	Izhikevich (Ch 5–6)	Reduced Models
4/9	Network models	Stimberg et al (2019)	Network Models
4/16	Work on final projects		
4/23	Work on final projects (T), presentations (Th)		
4/30	Presentations		

Materials

Computing requirements

Scientific programming is essential to computational neuroscience. Digital computers are good at doing the same thing over and over again, which is exactly what we need to reliably deal with the big data generated in neuroscience. In fact, everyone who works with data can benefit from being able to hand tedious tasks off to a computer. Scientific programming is an important skill for researchers at all levels, and the practice of programming will help you learn how to break down complex problems of all kinds into a set of logical steps.

Anyone who is interested in developing these skills is welcome to take the class regardless of programming experience. (Seriously!) You do need to be committed to learning these essential skills, however, which may require significant out-of-class work to nail down core concepts if you are totally new to this. See the end of this document for some resources on different topics.

Laptops are necessary for this class. You will need to bring your laptop, running Linux, OS X, or Windows, to every class meeting. Almost all the work will be performed in [Jupyter](#) notebooks hosted on UVA's high performance computing cluster. We will spend time in class making sure everyone has a functioning environment.

Readings

The assigned texts will come from the primary literature and from the following textbooks:

- Dayan, P and Abbott, LF (2001) Theoretical Neuroscience. MIT Press
- Sterratt D, Graham B, Gillies A, and Willshaw D (2011). Principles of Computational Modelling in Neuroscience. Cambridge University Press.
- Eugene Izhikevich, Dynamical Systems in Neuroscience. MIT Press.

The textbooks can be accessed through the UVA library as electronic texts. Use [VIRGO](#) to search. PDF copies will be shared through the File tab in Canvas.

Evaluation

Engagement (40%)

Due to the fact that we complete much of the work in class in pairs and small teams, attendance and active participation are critical. Unless otherwise arranged with the instructor, you may only miss one class without penalty. If you are unable to attend class in person due to a COVID-related isolation or quarantine, notify the instructor so that you can attend via zoom. Additional absences may be excused due to religious holidays, UVA-required extracurricular activities (e.g., competitions or performances), or legitimate academic reasons; requests must be made and approved at least one week in advance.

How you can succeed: Come to class with a laptop. Read or watch out-of-class material and come prepared with questions and thoughts so you can participate in discussions. Talk to your programming team to make sure you both understand the tasks. Ask questions in class or on the class slack if anything is unclear or if you are getting an error you and your team can't figure out.

Homework (30%)

Each week, you will be given assignments that will immerse you in theoretical or practical aspects of computational neuroscience. Almost all of these assignments will involve programming. Programming isn't just about making sure you've closed all your brackets; it's a new way of thinking, and these problems will give you practice with this type of thinking without your console giving you errors.

How you can succeed: Think hard about the problem and how to construct a series of explicit, logical steps that solve it. Write down your best thoughts on the solution, and read through it carefully to catch any leaps of reasoning you've made. Your process is more important than whether every step is correct. Remember, there are usually multiple ways to solve any problem.

Data Exploration (30%)

As part of a semester-long project, you and your programming team will have the opportunity to choose one of the neuroscience data sets we'll be working with in class. As we discuss various topics in data science, you will be applying these to your data set and building toward a complete research project. Along the way, you and your team will gain expertise on your data set and be a resource for other students in our class interested in working with that type of data. There will be milestones and chances for revision over the course of the semester. Important dates and more detailed information are laid out in the project description.

How you can succeed: Engage with the data set you have chosen. Explore the structure of your data set and read the methods section of the associated publication. Think about which methods we discuss can be applied to your data and what kinds of questions could be asked with them. Test and document functions you write to make them as easy as possible for you and the rest of the class to use. Take advantage of peer and instructor feedback to improve your final product.

Timeline

- Week 5: Choose a data set that interests you. Describe the experimental conditions and how the data were collected.
- Week 7: Write a module with functions to read raw data from the files in the data set and parse into a flexible data structure.
- Week 10: Develop a question you can answer about your data set (or a comparison between your data set and another) using one or more of the methods we have been working with.
- Week 13: Generate a set of preliminary figures that capture the most important features of your results.
- Week 14: Finalize your analyses and figures. Prepare a short presentation on your results.

Course policies

Work must be turned in on time to receive full credit. There is a penalty of 5% per day for late work unless an extension is arranged **before** the due date or an emergency prevents submission.

Group work must represent equal effort by all members. Do not put anyone's name on an assignment unless they made a substantive contribution. If your schedules are not compatible, if there is an emergency or illness, or if you need to come to a different arrangement for splitting the

work, contact me so we we can make sure there is an equitable solution that maximizes everyone's opportunity to learn.

Students with COVID-related symptoms or a positive test result within the last 5 days, or who are waiting on test results after a close contact must not attend in person. Notify me to make arrangements for attending virtually. If you are displaying symptoms in class, you will be asked to leave and will lose participation credit for the day.

I don't typically answer email or Slack messages outside of normal working hours. If I do write, I don't expect an immediate response.

My goal is for everyone in the class to have an equal opportunity to learn and to demonstrate their knowledge. Students with disabilities are entitled to reasonable accommodations. Contact the Student Disability Access Center (434-243-5180) for more information or to arrange accommodations.

Resources

Scientific programming and Python

- If you're new to programming in general, check out the [Python for Data Science](#) DataCamp tutorial.
- VanderPlas J (2016) Python Data Science Handbook. O'Reilly. The full text of this book is available [here](#) as a complete set of Jupyter notebooks that you can run locally or in [Google Colab](#). The first four chapters are highly recommended.
- Langtangen HP (2016) A Primer on Scientific Programming with Python. 5th ed. Springer: Berlin Heidelberg.

General mathematics and statistics

- Spence, Insel, and Friedberg, Elementary Linear Algebra. Prentice Hall.
- Hiranabe, The Art of Linear Algebra. <https://github.com/kenjihiranabe/The-Art-of-Linear-Algebra>
- Hirsch, Smale, and Devaney, Differential Equations, Dynamical Systems, and an Introduction to Chaos. Springer.