Foundations in Neuroscience IV – Computational Methods

Course Number:	26:112:611 – 2024/2025
Faculty:	Krekelberg (coordinator), Headley, McGinty, Cole
Last Change:	January 2025.

Time:	**Wednesday Morning (typically 9 am - 11 am)**
Location:	**Aidekman 202**

Course Description

Given the sheer volume and complexity of experimental data, data analysis is a key aspect of many, if not all, subfields of Neuroscience. Python and Matlab are the main tools for the analysis of neuroscientific data. In this course, students develop a conceptual understanding of modern analysis techniques and their practical implementation in Python or Matlab. The student will learn general analysis and programming techniques based on examples from the field of Neuroscience, including the analysis of behavioral data, functional imaging, and spike and local field recordings.

Recognizing the different backgrounds in quantitative analysis techniques, the course is split into two components. The first – Guided Self-Study – is an informal component that prepares the student to benefit optimally from the second part, which consists of lectures, a hackathon, and student presentations.

Course Objectives

Upon completion of the course, students will be able to

- Use Matlab or Python to write, debug, and document code.
- Use Matlab or Python's powerful visualization capabilities for data exploration and to present results in scientific presentations and publications.
- Explain the covered analysis techniques at a conceptual level.
- Use a range of quantitative techniques to analyze neuroscience data.

Part 1 - Guided Self-Study

Because students often start with very different backgrounds in quantitative analysis techniques, the course is based on a multi-semester trajectory that includes guided self-study. This component is not formally part of the for-credit course, but students with a limited background in quantitative analysis need to develop the computing skills needed to participate successfully in the lecture course and hackathon.

Summer

In the summer before their first semester, students master elementary programming skills in Python or Matlab. The Program Director provides students access to cmbn.udemy.com and helps them choose the Python or Matlab learning path. Students choose based on their experience, plans for a career in industry or academia, and the programming languages used in the laboratory they intend to work in during their PhD (the PI should be consulted). The faculty have curated the tutorial videos and assessments in Udemy.

September Bootcamp

In the September boot camp, faculty assess students' progress in self-guided study to identify weaknesses that will need additional attention in the Fall semester.

The format is informal: Students meet with a faculty member and are asked questions or given assignments in a ready-to-go working environment to determine their ability to comprehend, write, and debug code.

The outcome of this assessment is a list of topics that the student will work on during the Fall semester. Some students may need additional training in the curated Udemy Python or Matlab learning paths or additional materials provided by the faculty. Most students will be assigned (parts of) the Signal Processing and Statistics learning paths in Udemy. Some advanced students may not need any additional training in the Fall semester.

Pre-course Semester(s)

During the semester(s) before the start of the course, students work on their assigned learning paths in Udemy. This includes the Signal Processing and Statistics paths. A TA is available to assist and answer questions, and a Slack channel is available for discussion among students, the TA, and faculty.

Note that because the formal course (below) is only offered every other year, some cohorts will have one Fall semester to prepare, while others will have a full year.

Part 2 - The Course

One of the four Foundations courses of the GPN program is offered each semester. Foundations IV, therefore, takes place every other year in the Spring semester.

Lectures

Each lecture starts with a conceptual introduction, covering only a limited mathematical background. The primary goal is to convey the ideas behind the method and illustrate strengths and potential pitfalls using examples rather than formal proof.

Teachers share lecture notes and examples from published neuroscience articles ahead of time. They also provide a list of Python and Matlab packages/toolboxes that implement the method and pointers on how to get started using it. While instructors may rely primarily on either Matlab or Python for their lecture, they will cover the material to benefit users of either programming package.

Students read the articles and other materials before class, identify aspects of the technique they find difficult to understand, and prepare questions about the method's interpretation or implementation.

Project

During the course, students develop a data analysis project based on their data or a public dataset from one of the neuroscience archives (e.g., openneuro.org, dandi.org, human connectome project, etc.).

The project will use Python or Matlab to analyze the data with one or more advanced analysis techniques. During the first few weeks of the project, students develop a project outline/proposal and submit it to the instructor(s) for approval. Once approved, the student continues developing the code in a GitHub repository where instructors have access and can provide feedback. The hackathon sessions are in-person meetings where all students, under the guidance of the instructors, work on the project.

A complete project consists of

- A GitHub repository with all relevant code and documentation.
- A readme file with instructions on downloading and installing the code and the data.
- For Python: a Jupyter Notebook with the primary analysis results.
- For Matlab: an MLX file with the primary analysis results.
- Documented code that runs without errors.

List of topics

#	Topic	Who	Date
	Data science tools. *GitHub, containers, high-performance computing (HPC) clusters, open data initiatives, and repositories.*	BK	1/22/25

	Basic Mathematics *Linear algebra, derivatives, vectors, matrices.*	BK	1/29
	Visualization *Exploratory data analysis, histograms, scatter plots, best practices for data graphics.*	VmG	2/5
	Significance testing - parametric *T-test, ANOVA, multiple comparisons, interactions, power.*	VmG	2/12
	Significance testing – nonparametric *bootstrap, permutation tests.*	VmG	2/19
	Regression – 1 *ordinary least squares, design matrices, general linear model*	DH	2/26
	Regression – 2 *logistic regression, regularization, generalized linear models.*	DH	3/5
	Mixed Effects Models *non-independence, generalized models, effect size*	BK	3/12
	Hackathon #1 *Discuss project, setup data*		TBA
	Dimensionality Reduction *curve fitting, principal components analysis, singular value decomposition.*	BK	3/19
	Classification *support vector machines, neural networks, linear discriminant analysis, multivoxel pattern analysis*	MC	3/26
	Similarity Analysis *(partial) correlation, representational similarity analysis, multidimensional scaling*	MC	4/2
11	Hackathon #2 *Coding session to develop the project, with hands-on guidance from instructors.*		4/9

12	Hackathon #3 *Coding session to develop the project, with hands-on guidance from instructors.*		4/16
13	Hackathon #4 *Coding session to develop the project, with hands-on guidance from instructors.*		4/23
14	Project Presentation #1 *Brief intro to the data, explanation of the code, demonstration of outcomes.*		4/30
15	Project Presentation #2 *Brief intro to the data, explanation of the code, demonstration of outcomes.*		5/7

Office Hours & Communication

Instructors are available on the course's Slack channel.

Please post your course-related questions on the Slack Channel so every student can read them. Asking course-related questions on Slack allows other participants with the same question to benefit from the responses. Also, review the channel before posting a question; it may have already been asked and answered in previous posts.

Please email the instructors or ask for a face-to-face meeting for personal or confidential questions.

Course Delivery

The self-guided content will be delivered online in instructional videos. The videos will introduce programming and data analysis concepts and introduce assignments. The assignments are crucial to learning in this course; students should use any resources to complete the assignments, as the effort to find a solution is a key part of learning to program.

The lecture course will be in-person.

Textbooks & Materials

There is no textbook for the course. All materials, including videos, assignments, example scripts, functions, and other written materials are available from curated online resources (e.g.,

cmbn.udemy.com, GitHub classroom, etc.) Specific links will be shared at the start of the semester.

Prerequisites

All GPN students are expected to take this course. Other students should contact the Course Coordinator to discuss registration.

Students need access to a computer running a recent version of Matlab or Python.

Grading

In-Class Participation (30%):

Students are expected to

- Read materials provided before each lecture
- Ask questions and participate in discussions during the lectures
- Participate in the hackathon

Project (70%):

Students submit a final project that showcases their data analysis skills. This project

- Available to the instructors on GitHub.com
- Uses properly documented, structured Python or Matlab code
- Runs without errors
- Uses one or more advanced analysis techniques
- Uses appropriate visualization
- Uses the student's own research data or a public dataset.

Using Artificial Intelligence

Students may use all available artificial or biological intelligence to develop their projects. In the instructor's experience, AI tools can be helpful for advanced users, while early learners may be led astray by AI hallucinations. Whether using AI, code from others, or your code, the same adage applies: assume there are bugs and carefully verify the outcomes. Students are expected to be able to explain the function of every line of code and will be asked to explain a portion of their code as part of their final project presentation.

Attendance Policy

Time Commitment

To succeed in this course, we estimate that you will need to commit to at least 12 hours per week for each semester. Some weeks will require more time, some less. Note that this is the standard expectation for a 4-credit course.

In Part 2, the class meets every Wednesday morning, typically from 9 to 11 a.m., with some flexibility to accommodate seminars, etc. Students will also have additional meetings with one or more of the instructors to obtain project feedback and hands-on assistance with project setup.

Dropping the Course

To withdraw from a course, it is not enough to stop attending. In accordance with university policy, students wishing to withdraw from a course must do so formally through the Registrar's Office. It is the student's responsibility to complete all forms. If this is not done, the instructor must assign a grade of F at the end of the semester.

Academic Integrity

Students at Rutgers University are expected to maintain the highest ethical standards. The consequences of academic dishonesty, including cheating and plagiarism, are very serious. Rutgers' academic integrity policy is at academicintegrity.rutgers.edu. When you submit a project or other assignment, you need to abide by the honor pledge of "On my honor, I have neither received nor given any unauthorized assistance on this examination (assignment)."

Serving Students with Disabilities

Rutgers University welcomes students with disabilities into all of the University's educational programs. In order to receive consideration for reasonable accommodations, a student with a disability must contact the appropriate disability services office at the campus where you are officially enrolled, participate in an intake interview, and provide documentation. If the documentation supports your request for reasonable accommodations, your campus's disability services office will provide you with a Letter of Accommodations. Please share this letter with your instructors and discuss the accommodation with them as early in your courses as possible. To begin this process, please complete the Registration form on the Office of Disability Services web site.

Code of Conduct

In the interest of fostering an open and welcoming environment, lecturers, teaching assistants and students pledge to making participation a harassment-free experience for everyone, regardless of

age, body size, disability, ethnicity, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

Examples of behavior that contributes to creating a positive environment include:

- Using welcoming and inclusive language
- Being respectful of differing viewpoints and experiences
- Gracefully accepting constructive criticism
- Focusing on what is best for the community
- Showing empathy towards other community members

Examples of unacceptable behavior include:

- The use of sexualized language or imagery and unwelcome sexual attention or advances
- Trolling, insulting/derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or electronic address, without explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting