Engineering 105: Introduction to Scientific Computing

Day 1 – Course overview & introduction

Wednesday 12 January 2022

Course overview

- Introduction to computation and data analysis using Python, a versatile and powerful programming language widely used in engineering fields.

OBJECTIVES

(1) Students will learn how to code in Python and how to generalize what they learned to MATLAB; they will learn about variables, objects, functions, decisions, iteration, and recursion in the process.

(2) Students will learn scientific computing methods, including numerical experiments, simulations, statistical analysis and presentation of experimental data, image processing, and the creation of graphical user interfaces.

(3) Students will learn how to answer questions with data and how to think successfully in a data centric world.

Why Python?

- Open-source
- Intuitive syntax, readable code
- Increasingly important for science & tech jobs (and other fields!)
- Compatibility with many platforms and systems
- Large standard library
- Multiple programming paradigms

Why MATLAB?

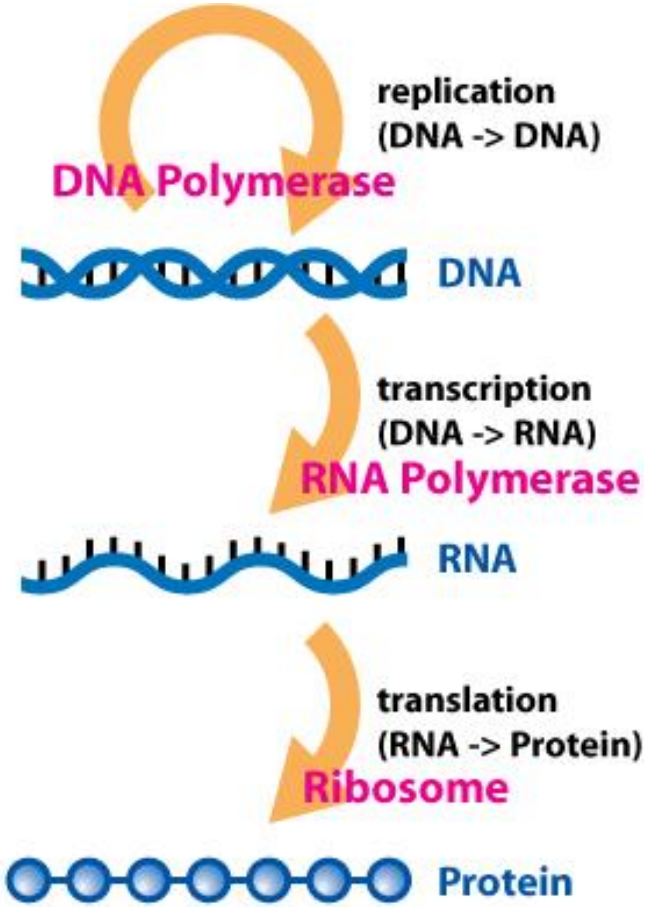
- Has not been replaced by Python used in some engineering fields/jobs
- Lots of built-in functionality
- Used in some upper-level engineering courses

- Learning one programming language makes it easier to learn other languages
- The last portion of the course will focus on applying your Python skills to MATLAB so you are ready for any class or field

How we learn to code

- Practice
  - In class & recitation
  - Homework assignments
  - Projects
- Collaboration
  - With classmates
  - Through searching (Stack Overflow, etc.)
- Problem solving
  - Focus on learning to code around relevant datasets

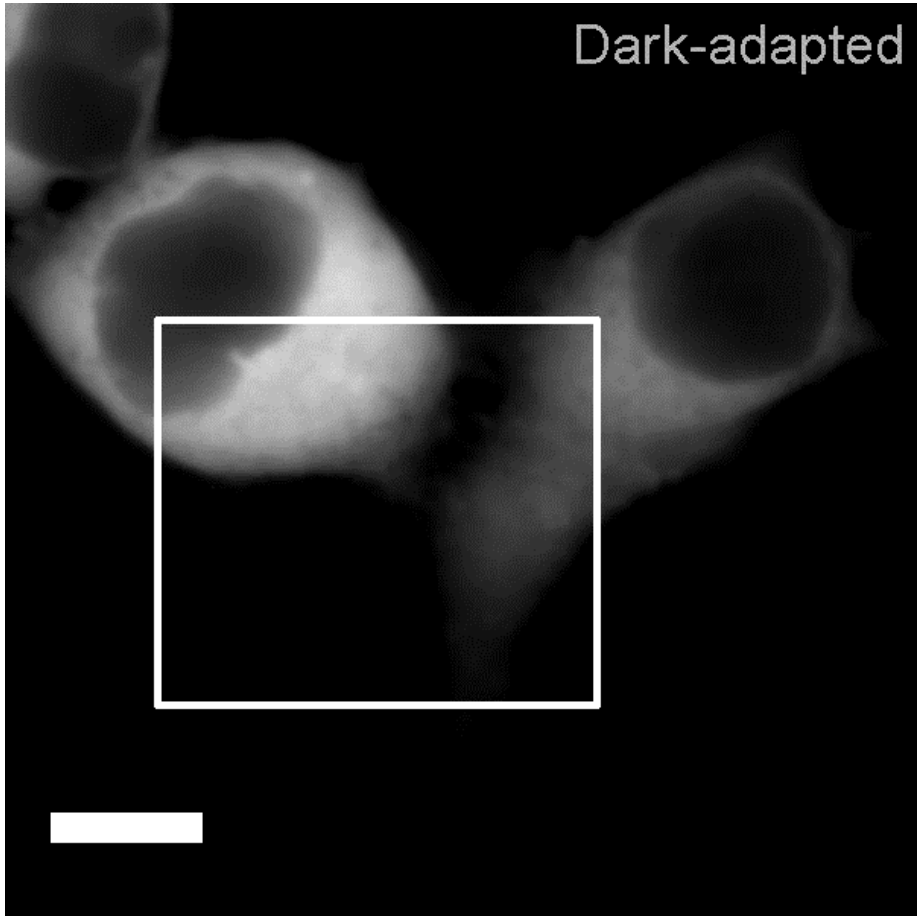
Dataset 1: the central dogma of biology



- How genetic information controls cell activities
  - Reading DNA, RNA, and protein sequences
  - Analyzing sequences and calculating gene and protein properties
  - Comparing sequences across species
  - Pulling genomic data from repositories
  - Measuring and predicting codon usage probabilities
  - Simulating transcription and translation timescales and outcomes

Image source: Wikimedia Commons, accessed 12/26/2021

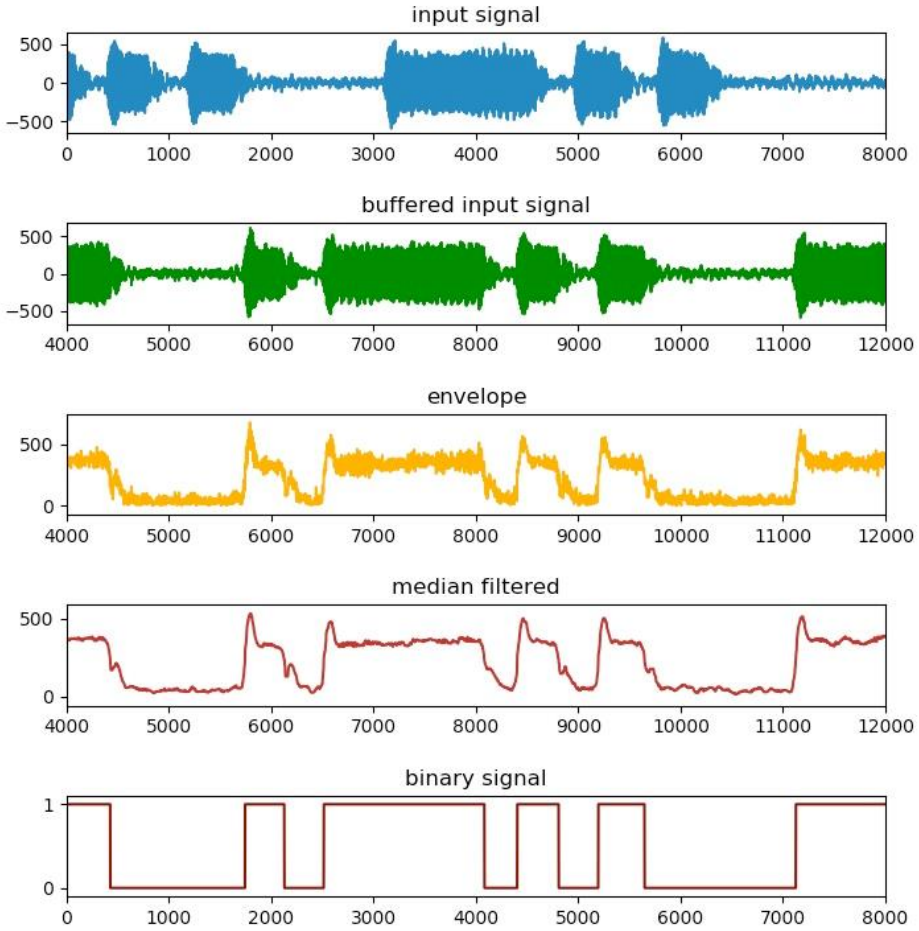
Dataset 2: tracking cell morphology & movement



- How do we quantify cell changes and movement over time?
  - Reading in images and detecting objects within them
  - Measuring velocities and calculating angles of movement
  - Simulating cell movement given constraints
  - Modeling cell velocities over time
  - Calculating differences and assessing significances of differences between cell types

Video source: Berlew, et al. 2021 *Advanced Biology*

Dataset 3: Signal processing



- What clinically significant information can we learn from signal data?
  - Visualizing and comparing signal data
  - Filtering, buffering, and processing signals to extract information
  - Calculating clinically relevant parameters from signals
  - Modeling and simulating signal data

Course topology

- Section 1: Introduction to data science (weeks 1-4)
- Section 2: Univariate statistics (weeks 5-8)
- Section 3: Foundations of machine learning (weeks 9-10)
- Section 4: Transitioning to MATLAB & final projects (weeks 11-14)

Class structure

- Short (~5 minute) lecture segments separated by interactive coding practice
  - Please bring a laptop to class—if that's an issue for you, let us know!
- Practice in pairs using Google Colaboratory notebooks
- Instructor and TAs on hand to answer questions and troubleshoot

Recitations

- Fridays 1:45-2:40 PM
  - On Zoom for weeks 1-2 (1/14 and 1/21)
  - In person for the rest of the semester (Towne 315)
- Led by TAs
- Attendance is optional but strongly encouraged!
- No new material covered, but a great place to ask questions and get more practice

Student/office hours and 1:1 meetings

- Instructor student hours: M W 11:15-12:30 PM in [room TBD]
  - Encouraged to sign up for meeting times on Calendly
- Questions about course material, assignments, course logistics, grades
- Questions about major/career choices, professional development, personal concerns
- To set up a 1:1 meeting: Email me with a brief description of the topic and a few times that work for you (and Zoom or in person)

Course staff – instructor

- B.S. in Chemistry – Haverford College (2015)
  - Thesis: “A bioinformatics driven investigation into the structural diversity of type II polyketides”
- Americorps – City Year Philadelphia (2015-2016)
- Ph.D. in Bioengineering – University of Pennsylvania (2021)
  - Thesis: “Single-component optogenetic tools for cytoskeletal rearrangements”
  - Adviser: Brian Y. Chow
- Research interests: Optogenetics, protein engineering, cell signaling, computational data analysis
- Personal interests: Mystery novels, my cat (Oscar), running, cycling, mental health, musical theater

Course staff – teaching assistants

Course participants – students

- Capped at ~40 students
- Priority given to Bioengineering majors
- No prerequisites; will draw upon some basic calculus
- Current enrollment: 30 first-years, 6 sophomores, 3 juniors, 2 seniors

COVID-19 and this semester

- My priorities: your learning & the health of our class / Penn / Philadelphia
- Masks required for class, recitations, office hours, indoor meetings
- Extra masks & hand sanitizer available at the door
- If you are not feeling well or have a red OpenPass, STAY HOME.
- Dr. Berlew will share her OpenPass at the start of each class and get COVID tested weekly.
- Course staff will remain open to discussing student COVID concerns and course policy suggestions.

COVID-19 and this semester

- Currently, we are online until week 3 (1/24)
- If things change again, we will work together to adjust expectations and the course calendar – expect check-in emails.
- Our priority is teaching you to code and keeping you healthy!

Attendance policy

- Good programming requires practice and collaboration – lecture attendance is strongly encouraged
- 20% of final grade is based on attendance & effort
- Don’t come to class if you have a red pass or aren’t feeling well – you won’t be penalized for these absences and we will work out how to make up lost time
- Classes will be recorded and posted on Canvas

Course materials

- Canvas – used for announcements, grade recording
- Github – used for code, assignments, course documents
- Google Colaboratory – online python notebooks with starting code
- Python (downloaded) – can be used for homeworks and projects
- MATLAB (downloaded) – used for unit at end of course

Grade breakdowns

- 50% assignments [weekly homework assignments]
- 20% attendance and effort [including classwork assignments]
- 20% final project
- 10% midterm portfolio

Assignments: Daily work

- Each class will consist of a mixture of lecture and hands-on programming practice, usually in a Google Colab notebook.

- Submit to Github within one week of class
  - Example: Classwork from Friday 1/14 is due by Friday 1/21 at 6 PM EST

Assignments: weekly homeworks

- Integrate current programming and data science concepts by creating something to apply to a bioengineering problem
- Due on Fridays at 6 PM on GitHub
- Homework 1 due Friday 1/28

Grading policy & late work

- Grades maintained on Canvas by instructor and TAs
- If you believe something was misgraded, please let Dr. Berlew know within two weeks of receiving your grade
- The entire assignment will be re-graded, with the possibility of the initial grade increasing or decreasing

- One-week grace period for classwork assignments
- If you think you need an extension, email Dr. Berlew ASAP
- Will grant extensions for homework assignments liberally in light of global dysfunction
- Without an extension, 20% penalty per day late

Midterm portfolio

- No midterm exam
- Instead, revise and/or expand upon a previous homework assignment and demonstrate a deeper understanding of the associated data science and programming topics
- Submit the portfolio assignment and a self-assessment on your progress in the course
- Meet with Dr. Berlew to discuss your progress and portfolio

Final project

- Applying course concepts to a scientific problem you find interesting
- Grading based on brief proposal, submitted code and project writeup, and a brief in-class project description talk
- More information will be provided later in the course

Academic integrity & collaboration

- Collaboration on coding in class is encouraged!
- Submitted homework and projects must be your own code and reflect your understanding of course material
- Detail any collaboration or resources consulted at the top of your assignments
- Please ask questions about acceptable collaboration and documentation as they arise!

Course accommodations

- Students who require accommodations in this course for disabilities or learning differences, health, religious creed, or other reasons are encouraged to schedule a meeting with Dr. Berlew as soon as possible to discuss your needs.

What should we call each other?

- I am fine with Erin or Dr. Berlew.

Homework 0

Submit a short voice memo or video introducing yourself—we want to know…

- What name you prefer and how to pronounce it
- Common pronunciation mistakes to avoid
- Pronouns, if you would like to share them
- Anything else to keep in mind when addressing you

Navigate to our course GitHub

https://github.com/eberlew/ENGR105-002_S22 [linked on Canvas]

└ In-class_Exercises

└ Day_1_Course_introduction.ipynb

└ ∞Open in Colab

Saying hello

- The `print` command can be used to display text, numbers, or Boolean (true/false) values on the screen.
- To use:

```
print( to display here )
```

Put what you want

Examples:

Input	Output
`print(108)`	`108`
`print("ENGR")`	`ENGR`

Practice: Printing “Hello, world!”

Image source: Wikimedia Commons, accessed 12/26/2021

Writing a function