Introduction to biological data analysis in Python
BI 399, University of Oregon

- Syllabus -

Stilianos Louca

Winter 2023

Contents

Note: This syllabus accounts for the current regulations and restrictions associated with the COVID-19 pandemic, as of December 28, 2022. Some aspects may be subject to change on short notice as the COVID-19 situation continues to evolve.

1 Course description

Computational data analyses have become an indispensable part of biological research and industry. From genome sequencing, electroencephalograms, gene expression microarrays, pedigrees and phylogenetic trees to epidemiological and ecological time series, nearly all domains of biology now necessitate the use of computers. Computer programming is the means by which modern scientists "tell" computers what to do. This course will introduce students to Python, a free and popular computer programming language, as a powerful tool for biological data analysis. Students will learn to load, analyze and visualize datasets using realistic examples taken from the scientific literature and public databases. Examples will cover a broad range of biological domains, including epidemiology (e.g., COVID19 dynamics), conservation, ecology and population biology (e.g., animal tracking data), physiology and evolution. The course will be very hands-on, with the bulk of the learning process happening through programming exercises.

The course will consist of one in-person lecture per week, two in-person computer labs per week, weekly reading assignments and weekly homework exercises. The lectures will cover new programming concepts and examples. Through the labs and homework exercises students will practice programming in Python to analyze/visualize datasets. The computer labs will also provide opportunities to discuss the weekly assignments and resolve questions related to the overall course material. In addition, the GE and the lead instructor will host weekly office hours.

This course is assigned 4 credits and is formally a 300-level course. The course replaces a previous course (Introduction to Python for Biologists) that used to be taught at the University of Oregon at the 400/500 level. The course targets undergraduate students in biology and other natural sciences with an interest in computational approaches.

2 Instructors

Lead instructor: Stilianos Louca, PhD, Assistant Professor.
Email address can be found at www.loucalab.com.
May also be contacted through Canvas, but **not** through Microsoft Teams.
In-person office hours: Wednesdays at 15:30–16:30, in Onyx Bridge 282.

Teaching assistants: Carolyn Delevich, George Vengrovski
In-person office hours (Delevich): Tuesdays + Thursdays at 16:00-17:00.
In-person office hours (Vengrovski): TBD.

3 Schedule

Lecture: Wednesdays 14:00-15:20, in Straub (STB) 251.
First lecture is on January 11, 2023. Last lecture is on March 15, 2023.

Computer labs - cohort 1: Mondays + Fridays, at 13:00–14:50, in Esslinger Lab (ESSL) 116.
First lab is on Friday, January 13, 2023. Last lab is on Friday, March 17, 2023. GE: Vengrovski.

Computer labs - cohort 2: Mondays + Fridays, at 11:00–12:50, in Esslinger Lab (ESSL) 116.
First lab is on Friday, January 13, 2023. Last lab is on Friday, March 17, 2023. GE: Delevich.

Mid-term exam: In-person written examination during one of the lectures, in the week of February 13–17, 2023.

Final exam: In-person written examination during the week of March 20–24, 2023.

Holidays observed: Martin Luther King, Jr., on Monday, January 16, 2023.

4 Zoom instructions

Unless mentioned otherwise, all lectures, computer labs, exams and office hours will be held in person. However, as the COVID-19 pandemic continues to evolve, situations might arise where any of these sessions is moved to a remote Zoom meeting (for example, if the lead instructor is forced to self-isolate). Hence, students are encouraged to familiarize themselves with Zoom prior to the course. Instructions for installing Zoom and joining a Zoom meeting can be found at:
`https://service.uoregon.edu/TDClient/2030/Portal/KB/ArticleDet?ID=101392`

Please join all online lectures and computer labs with video turned on and audio turned off by default. If you'd like to ask a question during the lecture and lab, please behave as in a normal class – either raise your hand and/or discretely draw the instructor's attention by voice (remember, everyone in the meeting can hear you). Please avoid using the Zoom text-chat function during the lecture and lab, since it distracts from the main conversation and may escape the attention of the instructor.

5 Expected learning outcomes

Students will learn the basic concepts of scientific programming and computational data analysis in the popular language Python. The following topics will be covered, roughly in this order:

- Variables, in particular for storing text ("strings") and numbers ("floats")
- String formatting and printing text to the screen
- If statements (i.e., conditioning of program flow)
- Lists and sets (e.g., lists of numbers)

- For loops (i.e., automated task repetition)
- Reading and writing data from/to files (including fasta and CSV/TSV files)
- Batch-processing of multiple data files (e.g., large numbers of genomes)
- Numpy arrays, including indexing, slicing and mathematical operations
- Visualization (scatterplots, curve plots, bar plots, histograms, heatmaps, box plots)
- Creating custom functions
- Boolean arrays and boolean indexing
- List comprehension
- Pandas dataframes for working with multivariate multi-type data
- Randomness, statistical hypothesis testing, P-values

Learning outcomes will be evaluated through homework assignments, lab sessions, discussions in class and written exams (details in Section 11).

6 Prerequisites

Recommended prerequisites for this course are:

- High school calculus (e.g., functions, logarithms, square roots, what is a derivative).
- High school statistics (e.g., what is a probability, mean and standard deviation).
- High school biology (e.g., what is a nucleotide, gene and genome).
- Basic familiarity with computers (e.g., installing software, downloading files, writing reports).
- Students will need to have access to a regular computer (e.g., laptop or desktop) to follow the course examples and complete their homework. If this is an issue, see Section 13 for possible accommodations.

No prior programming experience is required, but will be of benefit if existent. The formal prerequisites BI 213 or 214 or 283 are mostly listed due to technical limitations of our course scheduling system, and will almost surely be waived by the instructor upon request by the student.

7 Materials and equipment

Canvas: Unless otherwise mentioned, material (reading assignments, homework assignments, announcements etc) will be posted on the UO Canvas course page. Students should check for announcements and assignments posted on Canvas at least every 2-3 days; students may also choose to set up automated email notifications in Canvas.

Textbook: The full reading material for this course will be provided for free as a textbook on Canvas. Students should download the latest version of this document from Canvas at least once a week, since material might be added or modified throughout the course. A not-regularly updated version of the textbook can also be found here, and students are free to start reading it prior to the course.

Software: The course revolves entirely around the programming language Python, which is freely available for all major operating systems including Windows, Mac and Linux. Students are free to use their own laptops during the computer labs, instead of those provided in the lab. In fact, students are strongly encouraged to install Python (version 3.9 or later) on their own devices prior to the course, as the computer lab sessions alone are unlikely going to be sufficient for practicing the material and completing the homework assignments. Instructions for installing Python can be found in the aforementioned book, as well as on numerous locations on the internet. Students may also choose to install JupyterLab on their device, which allows running Python code through a web browser in a more interactive fashion, however this is not strictly necessary for succeeding in this course.

Copyright and privacy protection: Do not share or post online any of the course material, including lecture notes, assignments and official solutions, exams, or recordings of online Zoom meetings.

8 Weekly reading assignments

Students will be expected to complete weekly reading assignments from the freely available textbook mentioned in Section 7. These assignments will be posted on the course's Canvas webpage prior to the week in which they are supposed to be completed. Reading assignments will largely cover the material discussed in class during the same week. By having reading assignments that overlap with the lecture material, students will be better prepared to ask questions and participate in discussions during the lectures.

9 Weekly exercises

Students will be expected to complete weekly homework exercises, which will generally be in the form of Python programming tasks. Exercises will be based on previously covered material and will be graded. In general, exercises will be graded for the completeness and correctness of the computing results as well as the computer code itself, although the relative importance of the two will vary considerably between exercises.

Weekly exercise assignments will be posted on Canvas each Wednesday, and will be **due on the Tuesday of the following week, at 23:59**. Solutions to selected exercises will be subsequently discussed in class during the weekly computer labs. Students are urged to start working on their assignments as soon as possible, as some may require extensive troubleshooting.

Solutions must be submitted via Canvas as a single PDF file prior to the deadline. Only PDF files will be accepted; please do not submit your solutions in Word document format. Solutions in paper format will not be accepted. Students may choose to use Jupyter Notebooks, but must convert the notebook (including its output) into a single PDF for submission.

Students are expected to complete most of the assignments on their own. Consultation and collaboration with peers is permitted, but every student should fully understand the solutions produced and must individually write up their computer code and report (if applicable). Assignment solutions are graded individually. Solutions must be clearly legible and must adhere to common professional standards (e.g., proper axes labeling in figures).

10 Estimated student workload

In addition to attending the lectures and computer labs, students will be expected to complete weekly reading assignments from a textbook (details in Section 7) and weekly exercise assignments (details in Section 9). A summary of the typical workload needed to succeed in this course is given below:

Lectures: 10 lectures $\times$ 1.5 hours $=$ 15 hours

Computer labs: 20 labs $\times$ 2 hours $=$ 40 hours

Assignments: 10 weeks $\times$ 6.5 hours per week $=$ 65 hours

Total hours: 120

11 Grading

All homework and exam scores are counted on a percentage scale. The final grade G will be calculated on a percentage scale using the following formula: $G = 0.5 \cdot H + 0.5 \cdot E$, where H is the average score from the homework assignments (averaged over all assignments) and E is determined by the mid-term and final examination scores, as follows. If the final exam score is greater than the mid-term exam score, then E is equal to the final exam score. If the final exam score is equal to or less than the mid-term exam score, then E is equal to a weighted average of the two exam scores, with the mid-term exam weighting 30% and the final exam weighting 70%. The motivation for this grading scheme is that it gives an opportunity to the student to fully make up for a bad mid-term score at the finals, thus encouraging effort to improve even if a bad score is achieved in the mid-term. There will be no opportunities for extra credit, no grade bumps and no grading on a curve.

Letter grade: The following rubric will be used to convert the final grade G from a percentage scale to a letter scale.

Percentage (P)	Letter
$97\% \leq P \leq 100\%$	A+
$93\% \leq P < 97\%$	A
$90\% \leq P < 93\%$	A-
$87\% \leq P < 90\%$	B+
$83\% \leq P < 87\%$	B
$80\% \leq P < 83\%$	B-
$77\% \leq P < 80\%$	C+
$73\% \leq P < 77\%$	C
$70\% \leq P < 73\%$	C-
$67\% \leq P < 70\%$	D+
$63\% \leq P < 67\%$	D
$60\% \leq P < 63\%$	D-
$0\% \leq P < 60\%$	F

12 Course policies

Students are strongly encouraged to attend all lectures and computer labs. However, attendance will not be taken into account for the grade. It is ultimately the student's responsibility to stay up to date with the material taught and homework assignments. Students that miss a class will be able to download the covered course materials from Canvas, as part of the freely provided textbook. Assignment solutions and all important announcements will also be posted on Canvas.

Late delivery of solutions to exercise assignments will not be accepted in the absence of adequate justification. If you anticipate a delay in your solution delivery and in the presence of adequate justification please contact the lead instructor as early as possible (preferably well ahead of the deadline) to discuss options. Adequate justification may include, for example, a serious illness or natural disaster. Having too much academic workload (e.g., term papers for other courses) does not constitute adequate justification. Students are responsible for checking notifications and assignments posted on Canvas in time.

Students may share their personal lecture notes and may discuss their homework assignments with each other. However, students **must not** share exams, exam solutions, or solutions to the weekly exercise assignments online (including through Canvas). Students must not share audio or video recordings of the lectures, computer labs or office hours without prior permission from the lead instructor.

Please arrive in class on time. Late arrivals distract the instructor and the other students. Please turn off or silence your cell phones during classes. Ask questions if you did not hear or understand something. Students may use laptops or tablets during class (e.g., to take notes or test coding examples), but must do so in a way that does not distract other class participants (e.g., don't display distractive videos or play user interface sounds).

Students are expected to adhere to the University's guidelines on academic integrity as outlined in the University of Oregon Student Conduct Code:
https://policies.uoregon.edu/vol-3-administration-student-affairs/ch-1-conduct/student-conduct-code

13 Accessibility and other issues

If there are aspects of this course that result in unfair barriers to your participation, please let us know as early as possible. You may also contact Accessible Education Services in 164 Oregon Hall, by phone at 541-346-1155 or via email at uoaec@uoregon.edu.

Students may request to borrow a Chromebook laptop from the University of Oregon, as described here:
https://service.uoregon.edu/TDClient/2030/Portal/Requests/ServiceDet?ID=42589

Students may also remotely access the UO virtual computer lab:
https://service.uoregon.edu/TDClient/2030/Portal/Requests/ServiceDet?ID=42572
The virtual computer lab provides free access to various Windows software, including Python (as part of Anaconda).

Students will also have access to a remote JupyterHub, which they can access online to write and run Python code. This resource will be accessible via any standard computer, including those available at the UO libraries. Information on this resource will be disseminated at the start of the term.