# *Syllabus—Spring 2022*
# *BIOL 4XXX—Scripting for Biologists (3 cr)*

**Instructor**
Randy Klabacka
E-mail: klabacka.randy@gmail.com
Office hours: MW 2:30–3:00 pm

## Time and place

MW from 1:00–2:15pm.

## Course prerequisites

BIOL 3030 — Principles of Genetics

## 1 Course materials

- **Students are required to have a computer on which they can install a Unix-like shell, Python, and Git.** Any modern computer running Linux, Windows, or MacOS will work. We will help you install these requirements during the first week of class.
- **If you do not have access to a computer, please let us know as soon as possible.**
- Students will need to create a free account on GitHub (github.com), if you don't already have one.
- **There is no textbook required for this course.**

## 2 Course description

In this course, you will learn computational skills that will allow you to approach biological questions in new ways. You will learn how to use a computer scripting language to simulate, acquire, process, and analyze biological data. Similar to learning a new spoken language, learning to code requires regular practice. Accordingly, in this course, we will spend as much time as possible actively coding. This will involve "active-coding" lectures, scripting exercises, and a class project. I will teach using the Python language, but students are allowed to explore and use other scripting languages.

## 3 Course Objectives and Outcomes

***The primary goal of this course is for you to become fluent enough in a scripting language to enable you to probe your own research interests in new ways.*** By the end of this course, our goal is for you to learn the following skills:

- You will understand basic computer science concepts (e.g., loops, conditionals, functions, data types, and regular expressions) and be able to use them to simulate, process, and analyze biological data.
- You will learn best practices for writing and maintaining computer code, including version control.
- You will learn to critically review scripts used in the primary literature.

The key DBS course learning outcomes that we will address in this course are:

1. Students will be able to effectively locate, evaluate, and summarize published information in the life sciences. They will be able to identify appropriate sources for specific information needs and use appropriate search tools and search strategies to access needed information.

2. Students will be able to communicate effectively to the appropriate audience in the written genres common to biological sciences (e.g., lab reports, research articles). Components of written communication effectiveness include logical organization of data and ideas, appropriate language use, and correct use of scientific citations.

3. Students will be able to apply the scientific method to formulate testable hypotheses, gather data that address the hypotheses, and analyze the data (statistically, graphically) to assess the degree to which their scientific work tests their hypotheses and draw appropriate conclusions from the data.

4. Students will develop metacognitive skills and be able to distinguish between broad categories of metacognition as applied to their major. In particular, they will distinguish between foundational (i.e., knowledge recall) and higher order (i.e., creative, analysis, synthesis) metacognitive skills.

# 4 Course website

The syllabus, grades, announcements, and some class materials will be posted on the Canvas website for this course.

Some class materials will be available elsewhere on the web (e.g., https://klab-ut.github.io/coursemat/S4B.html).

# 5 Course Assignments and Grading

## 5.1 Course points

| Item | Points |
|---|---|
| Participation | 20 |
| Scripting exercises | 140 |
| Code review | 60 |
| Class Project | 80 |
| Total: | 300 |

## 5.2 Letter grade distribution

| Percentage | Grade |
|---|---|
| >= 90.0 | A |
| 80.0 – 90.0 | B |
| 70.0 – 80.0 | C |
| 60.0 – 70.0 | D |
| <60.0 | F |

## 5.3 Scripting exercises

Throughout the semester, you will complete assigned coding exercises to hone the skills you are learning in class. Each exercise is worth 10 points (3% of your final grade). Each exercise will be graded at the instructors discretion depending on the tasks required (some exercises only require completion, others require correct formatting and structure).

| Exercise | Topic | points | Due date |
|---|---|---|---|
| Exercise 0 | Setup survey | 5 | Jan 19 |
| Exercise 1 | Intro to git | 10 | Jan 31 |
| Exercise 2 | SSH key | 5 | Feb 7 |
| Exercise 3 | Python best practices | 10 | Feb 21 |
| Exercise 4 | Debugging | 10 | Feb 28 |
| Exercise 5 | Dataframes & plotting | 20 | Mar 14 |
| Exercise 6 | Reading & writing files | 20 | Mar 21 |
| Exercise 7 | Translation exercise | 20 | Mar 28 |
| Exercise 8 | Regular expressions | 20 | Apr 4 |
| Exercise 9 | Advanced topics | 20 | Apr 25 |

### 5.4 Code review

As a scientist, you will be asked to peer-review research manuscripts and associated code to assess the quality and reproducibility of the experiments to address the research question and the accuracy of their interpretations of the results. To practice these critical thinking skills and to assess your ability to apply the class concepts, you will choose a manuscript from the primary literature that utilized a significant scripting component to simulate, process, or analyze their data.

**By February 7, you must have your chosen manuscript approved by the instructors.** Failing to submit your selected manuscript will result in a deduction of 10 points.

You will write a two-page review of the article, focused on the associated code and your ability to reproduce the results presented in the published manuscript (papers significantly longer or shorter than two pages may lose points based on the instructor's discretion). Fully reproducing the results may not be feasible, but you should still discuss whether the authors provided what is necessary to do so (and if not, what was missing?). We will have review check-points where you will present the progress of your review with your peers. Each check-point will be announced one week before it occurs. The final review is **due on April 4**.

Code Review Rubric

| **Item** | **Points** |
|---|---|
| Summary of manuscript objectives | 20 |
| Language(s) used | 2 |
| Number of scripts | 2 |
| External packages used and why | 6 |
| Purpose of each script | 10 |
| Examination of code | 35 |
| Documentation critique | 7 |
| Code commenting  documenting | 7 |
| Following best practices | 7 |
| Reproducibility critique | 7 |
| References to specific lines of code | 7 |
| Conclusion | 5 |
| Summary of critiques | 5 |
| Total: | 60 |

You may need to look up best practices for other code if scripts aren't in python

### 5.5 Final project

You will develop a script, or set of scripts, to simulate, process, and/or analyze biological data. Team members will work collaboratively via version-control software (Git) and host your code in a repository on GitHub. At the end of the semester, each team will give a short presentation in the form of a workshop/tutorial. The goal is for the team to guide the rest of the class through an example of how to use their script(s). The tutorial should have enough documentation to be self-contained; i.e., other students should be able to complete the tutorial without the presentation.

If possible, we want this project to be applicable to your research. To enable this, we will be flexible about how students assemble into teams. Teams can consist of 1–6 students.

**By February 16**, each team must submit a short project proposal to be approved by the instructor. The project can change after the proposal is approved, but a change MUST be approved by the instructor. The proposal will be submitted as documentation within a GitHub page, with all team members included as collaborators. Failure to submit a proposal by February 16 will result in a 5 point deduction from the final grade.

A link to the code and documentation for the project is **due before class on April 18**. Presentations will be in random order during the weeks of **April 18 and 27**.

A student's grade will be determined based on the quality of the project, the ability of others to replicate their results, and their individual contributions to the project (assessed by Git/GitHub's tracking of every line of code written or modified by each contributor). More specifically, the break down of your grade for the project will be as follows:

Class Project Rubric

| **Item** | **Points** |
|---|---|
| Documentation | 25 |
| Summary of project objectives | 5 |
| Language(s) used | 3 |
| Number of scripts | 2 |
| External packages used and why | 5 |
| Purpose of each script | 10 |
| Code | 30 |
| Appropriate use of classes, types, methods, functions, variables, conditionals | 10 |
| Following best practices (including commenting, documenting, modularity) | 10 |
| Reproducibility | 10 |
| Presentation | 25 |
| Documentation walk-through | 5 |
| Example run (using abbreviated data if needed) | 5 |
| Answering questions | 5 |
| Clarity and smoothness of presentation (no bugs) | 5 |
| Staying within time limit | 5 |
| Total: | 80 |

You may need to look up best practices for other code if scripts aren't in python

## 5.6 Participation

Participation includes attendance, engagement in class discussion, and activity during breakout rooms. Engagement includes asking and answering questions. You may ask/answer questions verbally or using the zoom chat option. Activity during breakout rooms includes asking questions to peers, answering questions, and presenting code.

## 5.7 Late Policy

Scripting exercises will be reduced by 1% for each day they are late. Code reviews will be reduced by 5% for each day they are late. Late final projects will not be accepted.

# 6 Schedule

| | | | |
|---|---|---|---|
| Week 0: Introduction | | | |
| | Jan 12 | Do in class: | Syllabus introduction & Installations |
| Week 1: Getting Started | | | |
| | Jan 17 | NO CLASS | |
| | Jan 19 | Turn in before class: | Exercise 0 |
| | | Watch before class: | Intro to Bash |
| | | Do in class: | Review of Bash shell |
| Week 2: Introduction to Python | | | |
| | Jan 24 | NO CLASS | (watch "Intro to git" videos) |
| | Jan 26 | Watch before class: | all 4 "Intro to git" videos & ssh keys |
| | | Do in class: | git interactive discussion |
| Week 3: Python syntax and storage | | | |
| | Jan 31 | turn in before class: | Exercise 1 |
| | | Watch before class: | Intro to python part 1 |
| | | Do in class: | Python syntax interactive discussion |
| | Feb 2 | Watch before class: | Intro to python part 1 |
| | | Do in class: | Types/operators interactive discussion |
| Week 4: Flow control & functions | | | |
| | Feb 7 | Turn in before class: | Exercise 2 |
| | | Watch before class: | Intro to python part 3 |
| | | Do in class: | Python flow control interactive discussion |
| | Feb 9 | Watch before class: | Intro to python part 4 |
| | | Do in class: | Writing functions interactive discussion |
| Week 5: Pseudocode & Debugging | | | |
| | Feb 14 | Turn in before class: | Selected manuscript for code review |
| | | Watch before class: | Python best practices |
| | | Do in class: | Python best practices interactive discussion |
| | Feb 16 | Watch before class: | Using python packages |
| | | Do in class: | Python packages interactive discussion |
| Week 6: Best practices | | | |
| | Feb 21 | Turn in before class: | Exercise 3 |
| | | Watch before class: | Python debugging |
| | | Do in class: | Python debugging interactive discussion |
| | Feb 23 | Turn in before class: | Project proposal |
| | | Watch before class: | How to write pseudocode |
| | | Do in class: | Pseudocode interactive discussion |
| Week 7: RegEx & Plotting | | | |
| | Feb 28 | Turn in before class: | Exercise 4 |
| | | Watch before class: | Text processing |
| | | Do in class: | Text processing interactive discussion |
| | Mar 2 | Watch before class: | Plotting with python |
| | | Do in class: | Plotting with python interactive discussion |
| Week 8: NO CLASS: | Spring<br>Break | | |
| Week 9: Classes & Packages | | | |
| | Mar 14 | Turn in before class: | Exercise 5 |
| | | Watch before class: | Regular expressions |
| | | Do in class: | Regular expressions interactive discussion |
| | Mar 16 | Watch before class: | Creating python classes |
| | | Do in class: | Creating classes interactive discussion |
| Week 10: Random Numbers & Recursion | | | |
| | Mar 21 | Turn in before class: | Exercise 6 |
| | | Watch before class: | Random numbers |
| | | Do in class: | Random numbers interactive discussion |
| | Mar 23 | Watch before class: | Recursion |
| | | Do in class: | Recursion interactive discussion |
| Week 11: TBD / Work on projects | | | |
| | Mar 28 | Turn in before class: | Exercise 7 |
| | Mar 23 | | |
| Week 12: TBD / Work on projects | | | |
| | Apr 4 | Turn in before class: | Exercise 8 |
| | Apr 6 | | |
| Week 13: TBD / Work on projects | | | |
| | Apr 11 | Turn in before class: | Code review |
| | Apr 13 | | |
| Week 14: Project presentations | | | |
| | Apr 18 | Turn in before class: | Final Project |
| | Apr 20 | | |
| Week 15: Project presentations | | | |
| | Apr 25 | Turn in before class: | Exercise 9 |
| | Apr 27 | | |

# 7 Policies

## 7.1 Class inclusiveness

It is our intent that students from all backgrounds and perspectives be well-served by this course. We are committed to creating an inclusive space that fosters diversity along its many axes: ethnicity, race, sex, gender, disability, age, socioeconomic status, nationality, and culture. As your instructors and as a student in this class, it is our shared responsibility to develop and maintain a positive learning environment for everyone. Any type of discrimination or aggression toward your peers or instructors will not be tolerated.

## 7.2 Excused absences

Students are granted excused absences from class for the following reasons: illness of the student or serious illness of a member of the student's immediate family, the death of a member of the student's immediate family, trips for student organizations sponsored by an academic unit, trips for university classes, trips for participation in intercollegiate athletic events, subpoena for a court appearance, military orders, and religious holidays. Students who wish to have an excused absence from an exam for any other reason must contact the instructor in advance of the absence to request permission. The instructor will weigh the merits of the request, and render a decision. When feasible, the student must notify the instructor prior to the occurrence of any excused absences, but in no case shall such notification occur more than one week after the absence. Appropriate documentation for all excused absences should be provided by the student, when possible.

## 7.3 Disability accommodations

Students who need accommodations are asked to electronically submit their approved accommodations through AU Access and to make an individual appointment with the instructor during the first week of classes, or as soon as possible if accommodations are needed immediately.

## 7.4 Academic honesty

All academic honesty violations or alleged violations of the SGA Code of Laws will be reported to the Office of the Provost, which will then refer the case to the Utah Tech University Academic Integrity Committee.

## 7.5 Classroom behavior

The Auburn University Classroom Behavior Policy is strictly followed in the course, including the discussion forums on Canvas.

## 7.6 Emergency contingency

If normal class and/or lab activities are disrupted due to illness, emergency, or crisis situation, the syllabus and other course plans and assignments may be modified to allow completion of the course. If this occurs, an addendum to your syllabus and/or course assignments will replace the original materials.