BIOL 4320—Scripting for Biologists
Syllabus—Fall 2023
Utah Tech University

Instructor

Name: Dr. Randy Klabacka
E-mail: randy.klabacka@utahtech.edu
Office: SET 513
Office hours: Tues/Thurs 1–3 pm; Wed 2-3 pm

Course Details

Credits:	3
When:	Tuesday and Thursday 3:00–4:15
Where:	SET 501
Prerequisites:	BIOL 3300
Course materials:	This course will utilize online resources available to Utah Tech students (such as Pluralsight, YouTube, open-source software packages, and github)

Course description

Students will learn techniques in computational biology to apply their computer science skillset to biological data. Specifically, it is focused on learning best-practices to design scripts for computational biology through hands-on coding exercises. These exercises will allow students to refine their ability to analyze data using essential concepts in computer science such as conditionals, loops, functions, classes, regular expressions, and recursion. Topics of version control, code readability, and documentation design are used to highlight the importance of reproducibility in science. While the course is taught using Python, students are allowed to explore and use other scripting languages.

Course Learning Outcomes

The BIOL 4320 course at Utah Tech University has developed a set of program learning outcomes (PLOs). At the successful conclusion of this course, students will be able to:

- CLO 1: Implement best practices in script development for computational biology.
- CLO 2: Design computer programs to analyze their own (and other) biological data.
- CLO 3: Critically review coding scripts used in the primary literature.
- CLO 4: Create reader-friendly documentation to help others implement their code for similar datasets (or to simply reproduce their results.

Program Learning Outcomes

The Biological Sciences Department at Utah Tech University has developed a set of program learning outcomes (PLOs). The key DBS student learning outcomes that we will address in BIOL 4320 are:

- PLO 1: Outline the foundational concepts of biology including cellular, organismic, ecological, and evolutionary biology.
- PLO 2: Evaluate hypotheses, design research, test hypotheses, conduct data analysis, and draw conclusions on biology related problems.
- PLO 3: Integrate knowledge of scientific literacy in oral and written assignments when communicating biological topics.

Course Assignments and Grading

Course points

Item	Points
Participation	20
Scripting exercises	140
Code review	60
Class Project	80
Final Exam	40
Total:	340

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

Participation

Participation includes attendance, engagement in class discussion, and working on projects during class time. Attendance in this course is mandatory. Engagement includes asking and answering questions. Working on projects means that when class time is dedicated to working on projects, the student isn't working on material not relevant to the course. Failure to meet the requirements of participation on a day in the classroom will result in a five point reduction in the student's overall participation grade.

Scripting exercises

Throughout the semester, you will complete assigned coding exercises to hone the skills you are learning in class. 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). Here is a list of the exercises, their points, and their due dates:

Exercise	Topic	points	Due date
Exercise 0	Setup survey	5	Aug 24
Exercise 1	Intro to git	10	Aug 31
Exercise 2	Python best practices	10	Sep 19
Exercise 3	Bash scripting	10	Sep 26
Exercise 4	Debugging	10	Oct 3
Exercise 5	Dataframes & plotting	20	Oct 10
Exercise 6	Reading & writing files	20	Oct 17
Exercise 7	Translation exercise	20	Oct 24
Exercise 8	Regular expressions	20	Nov 7
Exercise 9	Recursive factorials	20	Nov 14

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 September 14, 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?). The final review is **due on Oct 31**.

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

Final project

You will develop a script, or set of scripts, to simulate, process, and/or analyze biological data. **While you will work in a team, each team member will lead their own independent project.** 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 individual 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. **By October 19**, each student must submit their own 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 October 19 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 December 4**. Presentations will be in random order during weeks 13 and 14 (see schedule below). Each presentation should be about 20 minutes long, with some time dedicated to answering questions. A student's grade will be determined based on the quality of the project, the ability of others to replicate their results, their individual contributions to the project (assessed by Git/GitHub's tracking of every line of code written or modified by each contributor), and their collaboration with team members on their projects (assessed by the instructor and informed via student survey). 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

Final Exam

The final exam will be given during its time scheduled by the university (see the schedule below). The exam will consist of a coding and a non-coding portion focused on principles learned during

the course.

Schedule

Week 0: Getting Started			
	Aug 22	Do in class:	Syllabus introduction & Installations
	Aug 24	Turn in before class:	Exercise 0
		Watch before class:	Intro to Bash
		Do in class:	Review of Bash shell
Week 1: Introduction to git			
	Aug 29	Watch before class:	all 4 "Intro to git" videos & ssh keys
		Do in class:	git interactive discussion
	Aug 31	turn in before class:	Exercise 1
		Do in class:	Python syntax interactive discussion pt 1
Week 2: Overview of python			
	Sep 5	Do in class:	Python syntax interactive discussion pt 2
	Sep 7	Do in class:	Python best practices
Week 3: Introduction to bash scripts			
	Sep 12	Do in class:	Interactive discussion on reading/reviewing bioinformatics literature
		Do in class:	Interactive discussion on bash scripts
	Sep 14	Turn in before class:	Selected manuscript for code review
		Do in class:	Interactive discussion on shell scripting best practices
Week 4: Pseudocode & Debugging			
	Sep 19	Turn in before class:	Exercise 2
		Do in class:	Interactive discussion on debugging python module
	Sep 21	Watch before class:	How to write pseudocode
		Do in class:	Interactive discussion on writing pseudocode
Week 5: Text Processing & Plotting			
	Sep 26	Turn in before class:	Exercise 3
		Watch before class:	Text processing
		Do in class:	Text processing interactive discussion
	Sep 28	Watch before class:	Plotting with python
		Do in class:	Plotting with python interactive discussion
Week 6: RegEx & Biopython			
	Oct 3	Turn in before class:	Exercise 4
		Watch before class:	Regular expressions video
		Do in class:	Interactive discussion on regular expressions
		Do in class:	Interactive discussion on Biopython package
Week 7: Using Random Numbers			
	Oct 10	Turn in before class:	Exercise 5
		Watch before class:	Random numbers
		Do in class:	Random numbers interactive discussion
	Oct 12	NO CLASS	Fall Break
Week 8: Project Development			
	Oct 17	Turn in before class:	Exercise 6
		Watch before class:	Recursion
		Do in class:	Recursion interactive discussion
	Oct 19	Turn in before class:	Project Proposals
		Do in class:	
Week 9: Recursion			
	Oct 24	Turn in before class:	Exercise 7
		Do in class:	Using factorials in population genetics
	Oct 26	Do in class:	Work on projects
Week 10: Common packages in comp bio			
	Oct 31	Turn in before class:	Code Review
		Do in class:	Using the modules of biopython
	Nov 2	Do in class:	Work on projects
Week 11: Code documentation			
	Nov 7	Turn in before class:	Exercise 8
		Do in class:	Best practices for creating code documentation
	Nov 9	Do in class:	Work on projects
Week 12: Work on projects			
	Nov 14	Turn in before class:	Exercise 9
	T & R	Do in class:	Work on projects
Week 13: Flex week			
	Nov 21	Do in class:	Dynamic programming introduction
	Nov 23	NO CLASS	Thanksgiving Break
Week 14: Project presentations			
	T & R	Do in class:	Work on projects
Week 15: Project presentations			
	Dec 4	Turn in before class:	Final Project
	T & R	Do in class:	Project presentations
Week 16: Finals Week			
	Dec 12	Final Exam	SET 501 (3:00 – 4:50 pm)

Course Policies

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

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 instructor 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 instructor will not be tolerated. If you experience any form of discrimination in this course, please report your concern at https://utahtech.edu/report-a-concern/.

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.

Makeup Policy
Students who miss the final exam or final course assignment deadline will need to contact the instructor and turn in the valid excuse within 48 hours from the time that the exams were given. The makeup exam schedule is determined by the instructor and will need to be done within ONE week (5 work days) from the time that the exams were given. The format, questions and difficulty-level of make-up exams are not guaranteed to be same as the normal exam, which are at the discretion of the instructors. The student and the instructor will agree upon a date and time for the makeup exam. Valid excuses include: 1). illness documented by a physician. 2) evidence of personal or family emergency. 3) official university excuses (e.g., university functions).

Instructor Communication Policy
It is your responsibility to read course announcements sent by your instructor (over Canvas and through email). Your Utah Tech University email address is the university-approved form of communication between instructors and students. Students should give the instructor 48 hours to get back to them on any communication, and one week for grading turnaround time on major assignments.

Academic Integrity
Academic dishonesty is an offense that will be reported to the UTU Academic Integrity Committee and handled by the instructor following University Policy 555. Acts of academic dishonesety include group work on quizzes/exams, sharing notes, communicating quiz/exam materials, and plagiarizing materials (including using machine learning natural language processing such as ChatGPT). Please refer to the following document for further information regarding academic honesty: UTU Academic Integrity Committee

Accessibility
Students who need accommodations are asked to electronically submit their approved accommodations to the UTU Disability Resource Center. If you need accommodations but have not established them, make an appointment with Disability Resource Center, North Plaza Building (Beside the Testing Center), 435-652-7516.

Generative AI

Generative AI models, such as language learning models, are recognized as valuable tools for enhancing the learning process in this course. These models are approved for guiding your learning experience, aiding in creating practice problems, and facilitating solution exploration. However, it's important to emphasize that while generative AI can support your understanding, it must not be employed to directly answer questions or complete assignments. Use of generative AI to develop practice problems and work through their solutions is encouraged as a study method. Misuse of generative AI will be considered an act of academic dishonesty in this course. If you have any uncertainties or inquiries regarding the appropriate utilization of generative AI, please reach out to the instructor. Open discussions about responsible AI implementation are welcomed, ensuring that we collectively uphold the educational value of these technologies. (This section was written with the help of chatGPT).

Important University Dates

See the UTU Academic Calendar

- Aug 22: Date classwork begins
- Aug 22: Tuition and fees due
- Aug 26: Last day to add without instructor signature
- Aug 31: Drop/Audit fee begins
- Sep 2: End of 100% Refund Period
- Sep 12: Pell Grant Census
- Sep 12: Last day to drop
- Sep 12: Last day for refund
- Sep 19: Last day to add/audit
- Oct 12: Midterm grades posted
- Oct 13-14: Fall Break (no classes)
- Oct 18: Last day to withdraw from class
- Nov 23-25: Thanksgiving Break (no classes
- Dec 9: Date classwork ends
- Dec 12-16: Finals week
- Dec 20: Final grades posted

Useful Links

Disability Resource Center
IT Help Desk
Library
Testing Center
Tutoring Center
Writing Center

Title IX Statement

Utah Tech University affirms its commitment to the promotion of fairness and equity in all aspects of the educational institution. Harassment and discrimination – including sex/gender discrimination, gender identity, gender expression, sexual harassment, sexual misconduct, gender-based violence, dating violence, domestic violence, stalking, pregnancy or parental , family or marital status and or retaliation –not only disrupts our commitment to maintaining an environment in which every member of the University community is treated with respect and dignity, but may also violate University policy and federal, state, and/or local law.

Should you or someone you know experience behavior that is coercive, discriminatory, harassing, and or sexually violent in nature, or if you or someone you know has questions about their rights and options regarding such behavior, you are encouraged to contact:

– Hazel Sainsbury, Dir. Of Equity Compliance, Title IX Coordinator: 435-652-7747 (ext. 7747) hazel.sainsbury@utahtech.edu ; titleix@utahtech.edu

Incidents may also be reported directly to law enforcement, either separately or in conjunction with any report made to the University's Title IX Coordinator, and the University will aid in making contact if requested.

– Utah Tech University Police: 435-275-4300 or by calling 9-1-1.

Maintaining a safe and inclusive University community is a shared responsibility. For more information on how Title IX protections can benefit you and help us keep a productive campus environment, visit titleix.utahtech.edu to learn more.

Utah Tech Email Disclaimer

You are required to frequently check your Utah Tech email account as important class and university information will be sent to this account, including bills, financial aid/scholarship notices, notices of canceled classes, reminders of important dates and deadlines, course information, and other information critical to your success at UT. To access your Utah Tech email account, visit mail.utahtech.edu. Your email account username is Digital-ID@utahtech.edu (e.g. D12345678@utahtech.edu). If you don't know or have forgotten your Digital-ID or password, please visit changepassword.utahtech.edu.

References and Changes to Syllabus

The instructor will reference the syllabus when students have questions on course objectives, policies, and grading. The instructor can make changes to the syllabus during the course of the semester, but these changes will always be reasonable and announced to the class.