CS101 Project Rubric

Introduction

In lieu of exams, you will be completing a coding project that builds off of Lab03: “DNA & RNA Sequencing.” Now that you’ve learned to parse and process DNA strands, your team will be responsible for writing more code to further manipulate, categorize, and visualize bioinformatic data. These exercises are based on the ROSALIND project.

Objectives

❖ Operate in a design team to produce specified deliverables
❖ Apply Python in a medium-scale project
❖ Analyze a system with both analysis and design optimization components

Milestones (100 points)

This project will have three major milestones to complete. For the first two milestones, your group will write and execute code that completes tasks defined in the rubric below. For the third milestone, you will be responsible for consulting with your TA to craft your own unique extension of the project, visualize your work, and submit a report.

Milestone 1 (27 points) — Due on March 10th

Lab03: “DNA” introduced you to writing functions that manipulate DNA strings. We’ll kick off the project with similar problems. Your team will be writing your own code and building off parts of Lab03.

※ Find the Milestone 1 details on Campuswire under Lectures.

Milestone 2 (34 points) — Due on April 7th

This leg of the project will be a continuation of Milestone 1, your team will build upon your existing code with more complex functions.

※ Milestone 2 details to be released on Campuswire at a later date.

Milestone 3 (39 points) — Due on May 5th

For this last Milestone, your team will be reviving the woolly mammoth with a final product of your own. You will be given a collection of mammoth DNA strands, but they are
fragmentary and require some preparation. You need to match adjacent strings (think like matching tree rings) and identify gaps in the DNA. You will then need to search a much larger database for fragments which can bridge the gaps. Your final product should include a hypothetical DNA genome including confidence of each match and any bridging fragments.

※ Milestone 3 details to be released on Campuswire at a later date.

Teams (points vary per Milestone)
The project will be completed in groups of 3-4 members; these will be the same group members you've been working with during labs. Your group will decide on team roles and responsibilities, complete peer evaluations at the mid-/endpoints of the project, and frequently check in with your TA with progress reports.

Roles and Responsibilities
Each team member must fulfill one of the following roles alongside completing milestone content. You will keep track of a task list or spreadsheet that every team member contributes to in a unique way.

- **Facilitator:** You are responsible for keeping the team on task and on schedule. 
  - You should help balance workloads by consulting with the strategist, and checking in on each of your teammates.
  - Each week, you are responsible for creating a task list that you and your team intend to accomplish for the week. You will also submit this task list to the TA.

- **Strategist:** You are responsible for monitoring workload, observing how well the team is functioning, and considering how it could improve in the future.
  - You will work closely with the facilitator on the weekly task list to determine which tasks have been accomplished and how to tackle new tasks.

- **Integrator:** You are responsible for making sure all coding components contributed by each team member are properly organized and combined.
  - You will demonstrate that your code runs by presenting files/doing a demo for the TA during check-ins.
  - You are also responsible for submitting your team code to Gradescope.

- **Recorder:** You are responsible for taking detailed meeting notes, monitoring contribution, and performing quality control to ensure that each team member is contributing fairly.
  - Each week, you are to communicate with your teammates and ensure that their contribution is fulfilling the requirements of the technical rubric, and mark progress on the team task list.
- If there is no 4th member, then the recorder role should be fulfilled by every team member.

(These roles are loosely based on the POGIL team model.)

**Check-ins**

Your TA will schedule weekly check-ins with you. These will be conducted through privately offered project office hours or through your private chat room. All team members are expected to be present and contributing/asking questions during the check-in. This is also the best time to ask any clarifying questions on your current work and new tasks moving forwards.

**Peer Evaluation**

A peer evaluation survey will be released with each Milestone. Your evals will be submitted to your TA and completely anonymous — this is your opportunity to tell us frankly how well your team is working, and for us to give extra encouragement or suggest solutions to any problems that may have arisen since the start of the project. Project grades may be adjusted to reflect your contribution and effort.

- **Late submissions:** 2 points will be deducted per individual (not team) score on the milestone for a missing or late peer evaluation.

**Task List**

Every member should contribute to the task list to keep track of individual and team contributions, progress, and goals. We’ve provided a sample task list for you to use as a template. Feel free to make modifications for your team or find a different method altogether, as long as all the required information is present!

**Organization**

In order to keep track of your code, we recommend that you use some form of centralized file sharing and version control. You can create a Google Drives folder or, even better, give popular version control tools like GitHub a try! If you are looking to minor in CS or find a career related to computing, learning how to use version control tools is vital — here is an introduction to GitHub if you are interested!

**Good Coding Practice**

When writing Python code, it is generally a good idea to follow standardized style guidelines. What this means is that producing clean, professional, and consistent code that others can understand is an important aspect of learning how to code!
The PEP8 (Python Enhancement Proposals) are generally a good guideline for making sure your Python code is consistently and logically styled. We won’t ask you to follow these guidelines exactly, but there are a few things we'll be looking out for when grading your projects:

**Good naming conventions**
Your functions and variables should have descriptive but relatively succinct names that are easy to understand to someone who has never touched your code. Variable names are typically lowercase with words separated by underscores (but you can use camelCase if you want!)

**Reasonable line length**
PEP8 guidelines suggest a maximum line length of 79 characters... Essentially, we just ask that you don't make us scroll horizontally :)

**Well documented code**
You may add comments to your code to help your teammates (and your TA) understand what you are trying to achieve. You don’t need too many! Don’t clutter up your beautiful code.

Additionally, we ask that you practice documenting your functions with a docstring, AKA a string that occurs as the first part of a module, function, class, or method. (See example below)

```python
def example_function():
    """ This function does something.
    Inputs? Outputs? Tell me more :)"
    """
```

**Project Submission**
At each milestone, you will submit a portion of the project on Gradescope. You will receive an invitation to join Gradescope via email. Once you have joined with your UIUC email, note when milestones are released and due. You may submit each milestone as many times as you’d like up until the deadline.

The integrator on the team should submit the code as a Python file and add the group members to the submission. The option to add Group members will appear after the code has been submitted:
Failure to add group members will result in a reduced grade, so please don’t forget this step!

- **Late submissions:** 5 points will be deducted from your total Milestone score per day after the deadline.