Essay 1

Introduction and overview: As a graduate student in bioinformatics, one extremely important skill is technical communication: learning to express complex ideas to colleagues, as well as explaining your work to those with a less developed background. In order to foster and improve this skill, this course will include several essays. Each essay in this class will be technical in nature, but about a quarter of your grade on each essay will also include grammar and professional writing style. (Because style and manner of communication also matters!) In addition to addressing the question asked, please also try to learn something new and to enjoy conveying this information. This ability to communicate technical ideas in an interesting and understandable tone is one of the one of the strongest skills you will use in any career path.

Assignment details: For the first assignment (in order to warm up to the material and to let me get to know your backgrounds and interests a bit better), I'd like you to review chapters 2 and 3 of the textbook, and then write a 1-2 page essay that either reflects on what you have learned or used about algorithms in your bioinformatics work so far, or considers how you will need to use algorithms in future research projects

I know you have a range of experiences, so I'm deliberately leaving this open - ideally, you'll be able to talk about bioinformatics algorithms that you have used in your research, but feel free to also discuss internship experiences or even class projects that have used biology **and** algorithms in a non-trivial way.

To get you started, here are a few questions you might reflect on before writing:

- In your own research project or industry experience or class project, what bioinformatics algorithmic problems have you seen or worked with? (Alternatively, if you haven't done research projects or BCB course projects, you're welcome to discuss your planned research projects, and discuss the same decisions or tradeoffs that are coming.)
- What toolkits or codebase have you used, or did you develop your own code? Why did you choose these particular tools for your project?
- Did you understand the background behind the algorithms, or read at all about tradeoff between various choices? If you didn't, this is a good opportunity to revisit and compare options! Go research your tools a bit, and gain at least a bit of an idea of the ideas behind them.
- If you understood the background and/or designed your own: were they recursive, iterative, brute force, or other types of algorithms? Did you uncover any hidden flaws or drawbacks when either using or developing these algorithms?
- If you did not delve as much into designing algorithms you used: How much did you need to dive into background in order to use (or choose) them effectively? How (if at all) do you think a deeper understanding of algorithms might have helped your work?

Note that I don't want you to copy/paste these, nor are you required to answer all of them! The goal is a narrative, and these are just talking points to get you started.