(Revised) Nitrogen Fixation Gene Analysis Project Proposal

The big idea - What is the main idea of your project? What topics will you explore and what will you generate? What is your **minimum viable product**? What is a **stretch goal**?

> The main idea of our project is to identify genes responsible for nitrogen fixation within bacterial genomes. Out of the topics we've explored in this class, this project will mainly involve data analysis. The minimum viable product for this project is an extended gene-finder that finds genes in a genome that are similar to a given gene and returns a percentage similarity; for data visualization, a dot plot could be employed. Our stretch goal is to create code that prints the genes (as base pairs) of the nitrogen fixing bacteria and shows the number of nitrogen fixing gene matches.

> **Learning goals** - What are your individual learning goals for this project?

> For this project, Erica would like to be able to have a nice, efficient, optimized code and utilize good coding practices.

> Rebecca would like to enhance her understanding of molecular biology and computational methods to interpret genetics.

> Liv would like to learn more advanced data analysis that could be applied to identifying different kinds of genes. She would also like to learn to write more elegant, advanced code.

Implementation plan - This will probably be pretty vague initially. Perhaps at this early juncture you will have identified a library or a framework that you think will be useful for your project**.** If you don't have any idea how you will implement your project, provide a rough plan for how you will determine this information.

> At this point we don't fully have a sense of what we're going to use for the project, but we imagine we will end up using pypy because the project requires a lot of looping, as well as the initial code from gene_finder, which requires use of random, amino_acids, load_seq, and load_contigs. Otherwise we're not certain how the code will play out.

Project schedule - You have 6 weeks (roughly) to finish the project. Sketch out a rough schedule for completing the project. Depending on your project, you may be able to do this in great specificity or you may only be able to give a broad outline. Additionally, longer projects come with increased uncertainty, and this schedule will likely need to be refined along the way.
I
> Week 1- know exactly what the project entails; project proposal, outline of the code and
>> delegation
> Week 2 - Finish an extremely simple implementation of the code that can be built upon.
(start working on the algorithm)

Week 3 - Finish a implementation of the code that finds nitrogenase matches.
Week 4 - Finish a code that can find nitrogenase matches. (% matches)
Week 5 - Debugging.
Week 6 - Debugging what we have, final touches (such as making the program more user friendly.)

Collaboration plan - How do you plan to collaborate with your teammates on this project? Will you split tasks up, complete them independently, and then integrate? Will you pair program the entire thing? Make sure to articulate your plan for successfully working together as a team. This might also include information about any software development methodologies you plan to use (e.g. agile development). Make sure to make clear why you are choosing this particular organizational structure.

Work independently and come together to go over code during team meetings. Heavily utilize github, floobits, and messenger/text

Risks - What do you view as the biggest risks to the success of this project?

Scheduling could hurt the project, it seems each member of our team has limited time during the week to meet. Another big risk is limited coding experience, which may slow down the time it takes to code and introduce more confusion with debugging. Along that vein, project complexity could be an issue, though that seems to be a lesser worry.

Additional Course Content - What are some topics that we might cover in class that you think would be especially helpful for your project?

It might be useful to cover more code optimization material to help us code elegantly.