**Project #3**

**Objective**

To combine all that youve learned in the course towards the accomplishment of a high-throughput computing task.

**Due**

Friday May 4th at 5pm PST. Please create a new Project03 folder on your GitHub repository and upload all files. Please include a REAME.md file that fully explains how to reproduce the final output using your solution.

**Background**

This project uses the NCBI sequence similarity tool BLAST. If you are unfamiliar with it, you can learn more about it here: <https://en.wikipedia.org/wiki/BLAST>. BLAST is already installed on Kamiak.

**Tasks**

Given any FASTA file containing over 10,000 protein sequences, split that file into 10 equally sized files. Using Kamiak, execute the BLAST **blastp** program using the SwissProt as the database. You must execute blast on all 10 files, but you must automate submission of those 10 files in some way. You cannot have 10 different Kamiak submission scripts. You can use the /data/hort503\_01\_s18/example-data/all.pep file to test your script (It contains over 66K *Oryza sativa* (rice) proteins).

You can find the SwissProt database here: /data/hort503\_01\_s18/example-data/swissprot. The output of your blast results should be in tab-delimited format and should contain the following columns in this order:

1. Query accession
2. Subject accession
3. Subject title
4. Percent identity
5. Alignment length
6. Number of mismatches
7. Number of gap openings
8. Start of alignment in query
9. End of alignment in query
10. Start of alignment in subject
11. End of alignment in subject
12. Expect value
13. Bitscore

Next, generate a tab delimited summary file of all the results containing two columns, the first being the protein name, the second being the total number of alignments matches for that protein. Sort the file such that the gene with highest number of hits occurs at the top of the list.

Once your code is completed you should upload all scripts to your GitHub repository. Additionally, you should generate a README.md file to be include with your GitHub submission as well. The README.md file should describe to a reader what your project 03 submission does, and how to execute each step. The instructions should be sufficient such that someone not in the class could use code to execute a split BLAST job. The README.md file can contain plain text, or it can use markdown to “prettify” the text. Information about how to use markdown in your REAME.md file can be found here: <https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>

There are no other requirements. Choose the best solution that makes sense to you by combining the knowledge and tools covered in this class. You can use Python, BASH, Pandas, command-line data wrangling, Nextflow, Kamiak, etc. There will be multiple avenues so there is no right way to do this. But the final output should be correct.

**Grading**

You will be graded on the following scale

1. Every portion of your solution runs without any errors (e.g. scripts, Kamiak submissions, command-lines, etc): 70 points.
2. Your program generates a properly formatted output file and plot: 15 points.
3. Your README.md file is sufficient detailed to allow someone outside of the class to run your program (5 points).
4. Your script follows the criteria specified above: 10 points.
5. 5 points will be subtracted for each day late.

**Extra Credit (10 points)**

Rather than split the input FASTA file into 10 equally sized files, split it into files no larger than 5000 sequences each. So, for the 66K rice protein example file you would be creating 14 files. Adjust your workflow such that it can accept any number of files rather than just 10 and someone could provide any size input file and the script will work.