**BT-3172: Special Topics in Bioinformatics**

**Lab 13: Introduction to the Unix shell for biological applications.**

**Name : Anushka Udara**

**Index number: s14234**

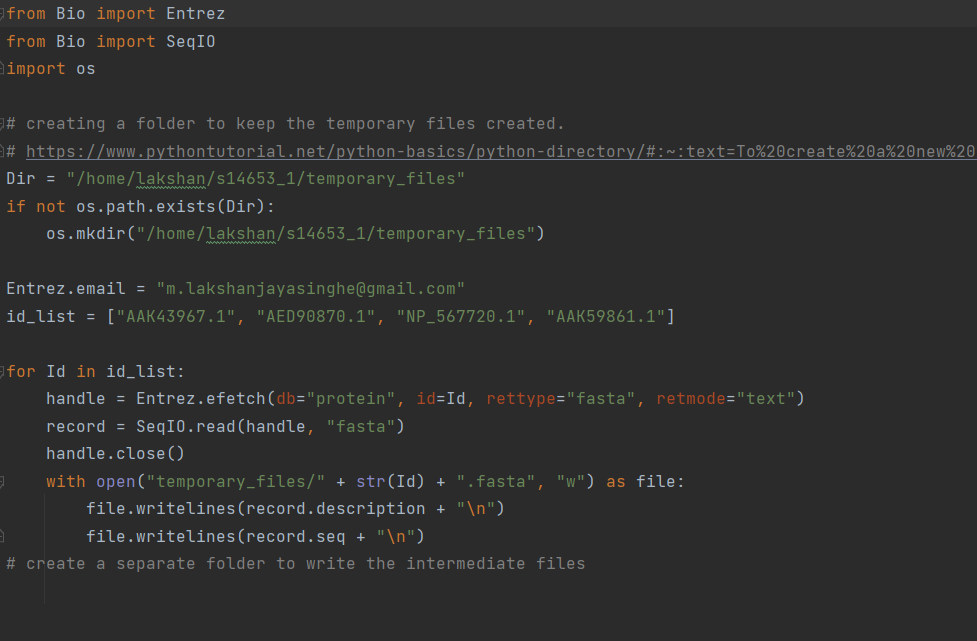
In this practical, you will learn how to use basic Unix shell commands to solve simple biological questions.

Create 2 folders for the two questions and name them in “Your\_index\_question\_no” format. Use the Unix shell/command line to implement your commands. Follow the question-specific instructions and save the necessary Python and shell script files in their folders. Also, make sure all the outputs specified in the questions are in the folders. Finally, compress the folders and upload them to the LMS. **You must write the Unix shell commands and Python scripts in the space below each question for evaluation.**

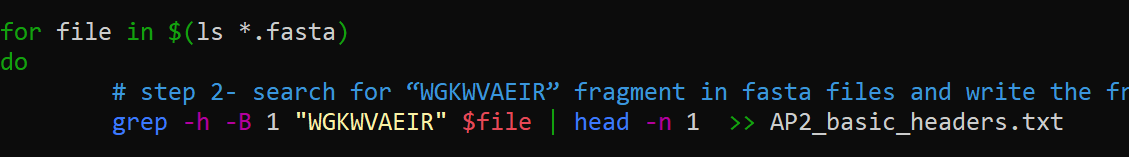
1. Processing multiple FASTA files using the Unix shell.

In this problem, you will be working with APETALA2/ETHYLENE RESPONSIVE FACTOR (AP2/ERF) family transcription factors (Zhouli, et al., 2019) which contain the AP2/ERF DNA-binding domain.

.

* 1. Write a Python script to retrieve the GenBank records for the following accessions: "AAK43967.1","AED90870.1","NP\_567720.1", and "AAK59861.1", and save their amino acid sequences in **separate** FASTA files. The FASTA files can be named by their respective accession number. You can use the Biopython module when writing the script. Save the script as “Your\_index\_multi\_fasta.py”.
  2. Use the grep command to search for the “WGKWVAEIR” amino acid sequence fragment from the AP2/ERF domain in above retrieved sequences and output the FASTA headers of the sequences which contain the domain fragment in a separate file called “AP2\_basic\_headers.txt”. Write the headers in the space below.

AAK43967.1 putative AP2 domain-containing protein [Arabidopsis thaliana]

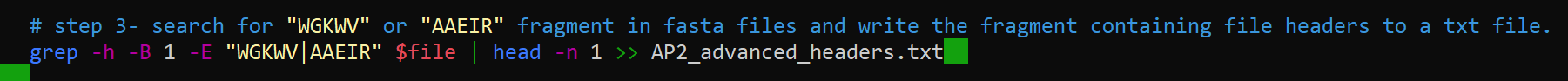
AED90870.1 DRE-binding protein 2A [Arabidopsis thaliana]

* 1. Now, modify the above search term to include a REGEX expression to search for “WGKWV/AAEIR” amino acid fragment in the sequences and output the FASTA headers of the sequences which contain the domain fragment in a separate file called “AP2\_advanced\_headers.txt”. Write the headers in the space below.

AAK43967.1 putative AP2 domain-containing protein [Arabidopsis thaliana]

AAK59861.1 At1g53910/T18A20\_14 [Arabidopsis thaliana]

AED90870.1 DRE-binding protein 2A [Arabidopsis thaliana]

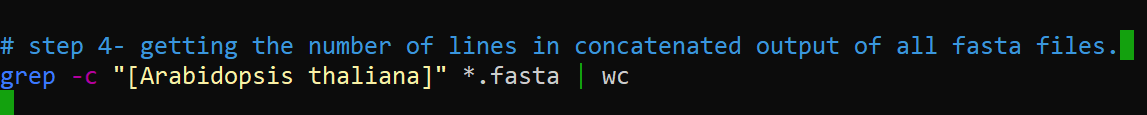


* 1. Write a shell command to concatenate the FASTA files downloaded in question (I) and count the number of FASTA files in the concatenated output. Write the count below.

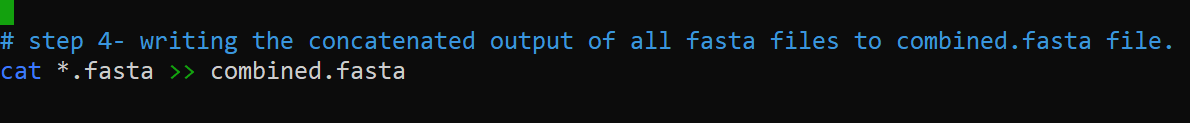
Hint: Use the cat, grep, pipe and wc commands appropriately.

Using grep command to search for Arabidopsis thaliana name, we can get the number of fasta files. wc command outputs the**number of lines, word count, byte and characters count**in the given input. Since, Arabidopsis thaliana is mentioned only once in the fasta file’s header, it can be used to get the number of fasta files in the concatenated output.

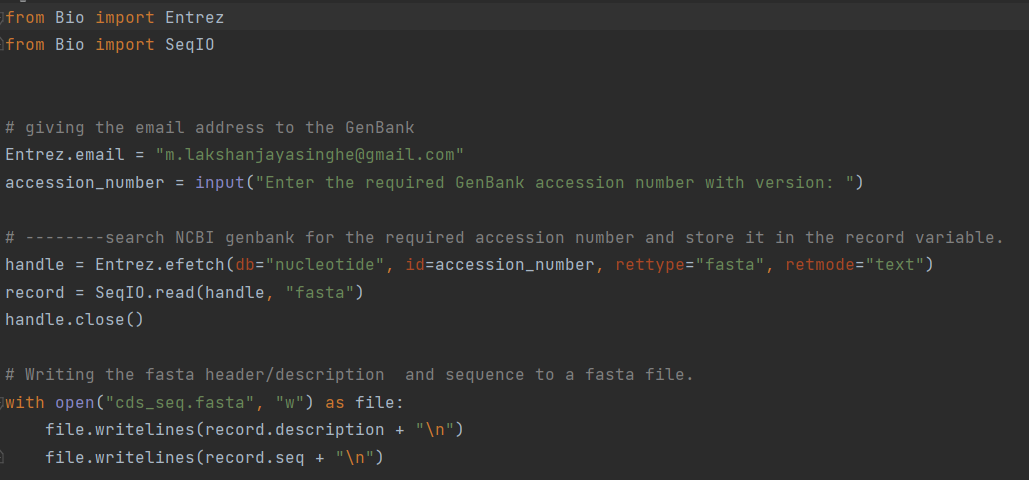
4 4 77



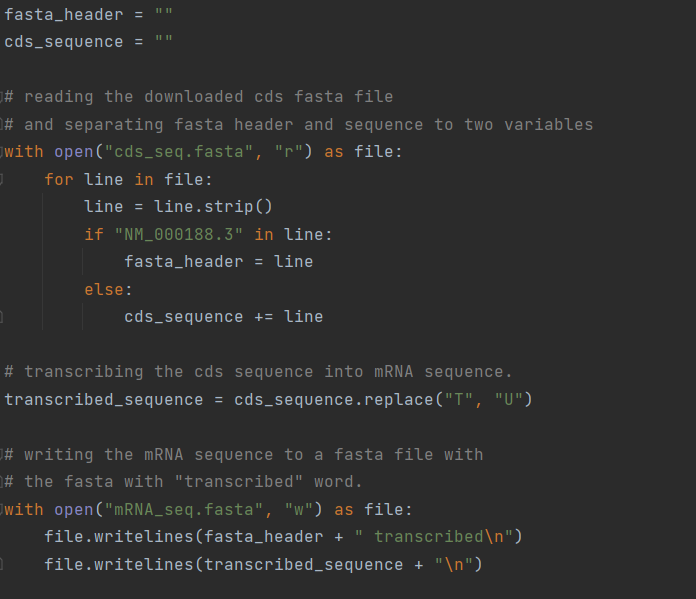
* 1. Write a shell command to concatenate the FASTA files downloaded in question (I) into a single FASTA file called “combined.fasta”.



1. Write the following Python scripts and assemble them into a bioinformatics pipeline using a shell script. You can use Biopython package when writing the Python scripts. Because you are writing a bioinformatics pipeline, Python scripts must be properly commented and an introduction should be given for each script.
   1. First, write a Python script (cds\_seq\_retrieve.py) to retrieve the GenBank record for an accession number (with version) of a protein coding DNA sequence or a reverse-transcribed mRNA complement given by the user and save the sequence in FASTA format (cds\_seq.fasta). The script must prompt the user to input the accession number (with version).

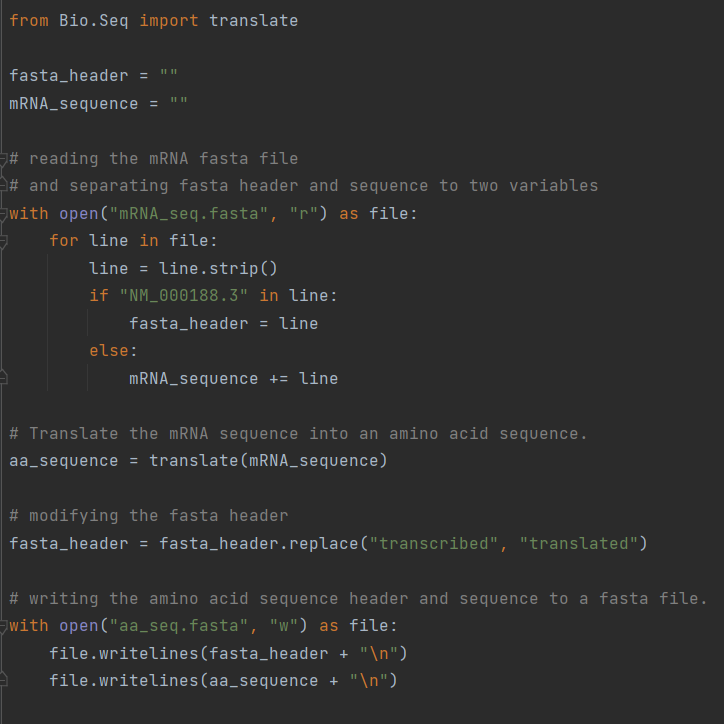


* 1. Then, write a Python script (transcribe.py) to transcribe the above sequence in the FASTA file and save the transcribed mRNA sequence in another FASTA file (mRNA\_seq.fasta). The FASTA header must contain the added word “transcribed” at the end. The program should read the “cds\_seq.fasta” file.

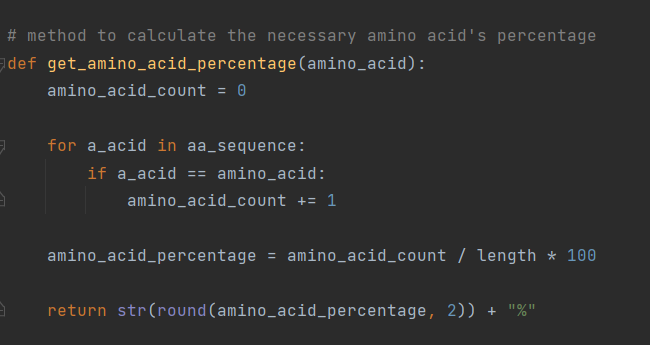


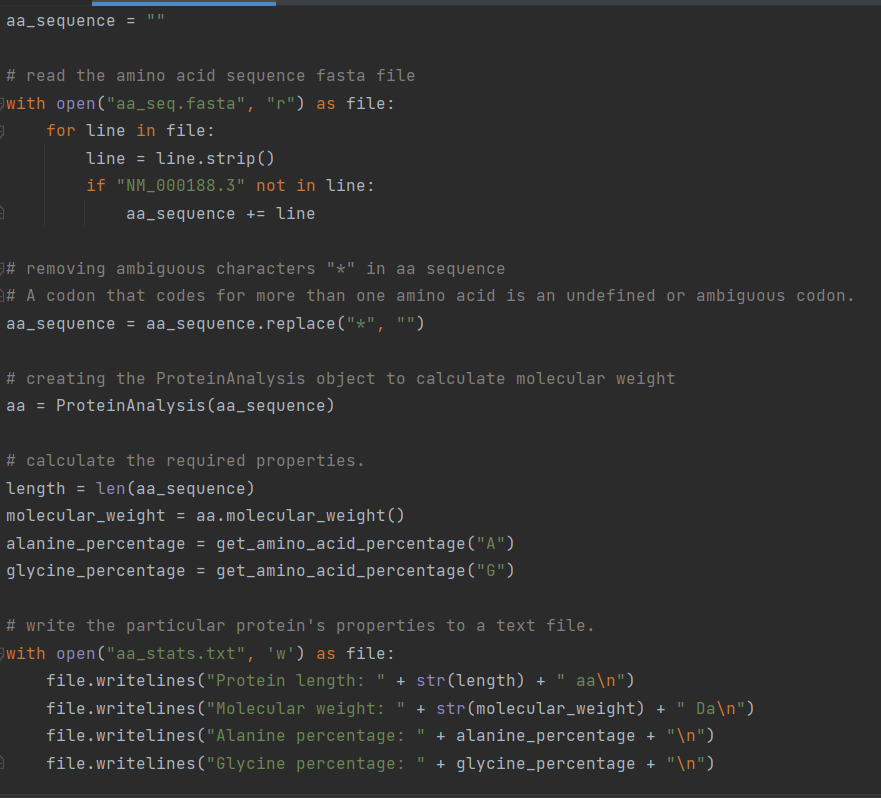
* 1. Then, write a Python script (translate.py) to translate the above sequence in the FASTA file and save the translated amino acid sequence in another FASTA file (aa\_seq.FASTA). The FASTA header must contain the added word “translated” at the end. The script must read the “mRNA\_seq.fasta” file.

Hint: for this example, it is not needed to start the amino acid sequence with methionine. Simply translate the mRNA sequence using translate() function in Biopython.

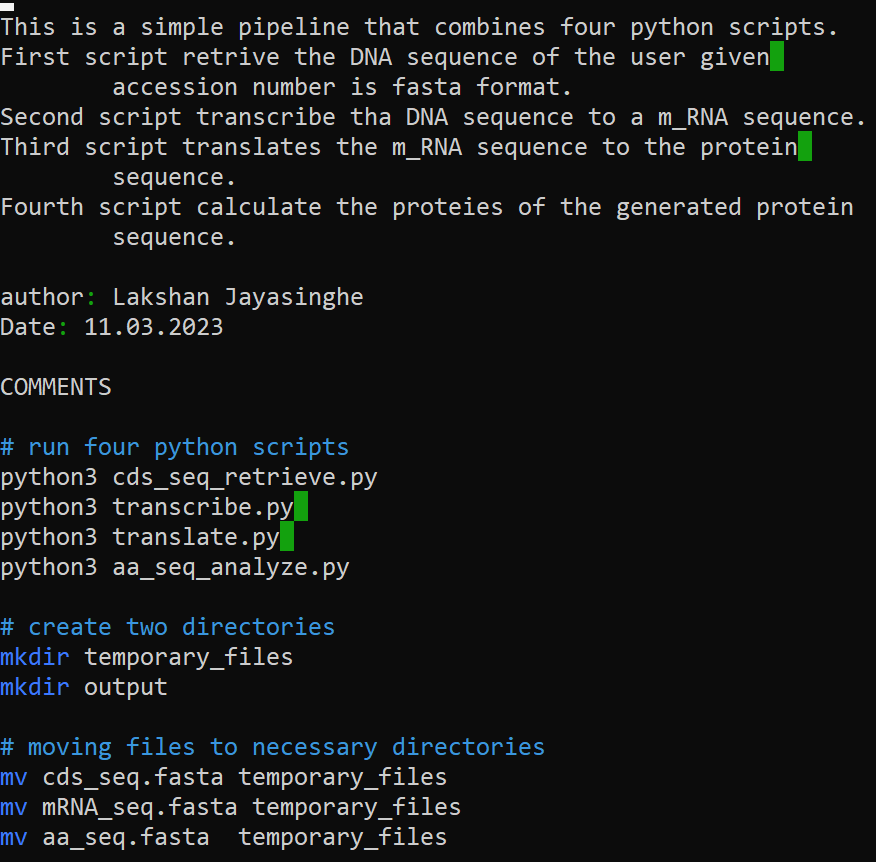


* 1. Finally, write a Python script (aa\_seq\_analyze.py) to analyze the aa\_seq.fasta file and calculate the length, molecular weight, alanine percentage, and glycine percentage of the sequence. Save the calculated parameters in a new text file called “aa\_stats.txt”. The script must read the “aa\_seq.fasta” file as the input.

Hint: You can use Biopython for above calculations. Find out the specific sub module for protein sequence analysis.



* 1. Now, using a shell script, build a simple pipeline to combine the above 4 scripts in the given order. Further, using the same shell script, create two folders: intermediate\_files and output. Move the “cds\_seq.fasta”, “mRNA\_seq.fasta”, and “aa\_seq.fasta ” into the intermediate\_files folder and the final output file: “aa\_stats.txt” into the output folder. Save the shell script as “your\_index\_bi\_pipeline.sh”. Use the “NM\_000188.3” accession as the input to the pipeline, which is for human hexokinase 1 gene. Write the amino acid statistics you calculated below, which would be in the output folder.



Protein length: 1147 aa

Molecular weight: 128469.92550000083 Da

Alanine percentage: 6.54%

Glycine percentage: 6.8%

**References**

* Xie, Zhouli, et al. "AP2/ERF transcription factor regulatory networks in hormone and abiotic stress responses in Arabidopsis." *Frontiers in plant science* 10 (2019): 228.