

# Homework 4

Skylar Chan

2023-2-17 (due 2023-2-27)

## Contents

<b>1</b>	<b>Instructions</b>	<b>1</b>
<b>2</b>	<b>Standard error</b>	<b>1</b>
<b>3</b>	<b>Regex practice</b>	<b>1</b>
<b>4</b>	<b>Grep</b>	<b>2</b>

## 1 Instructions

You must write all your answers in single Bash script. Any lines that are not commands should start with a # to indicate that they are a comment. The shell won't execute commented lines.

## 2 Standard error

1. Although it looks like it, `some_command 2>&1 > /dev/null` does not redirect standard error to `/dev/null`. Why not?

## 3 Regex practice

What do the following regular expressions match? Provide 3 example words that match.

2. `^#\??\**\++`
3. `[0-9]{3}-[0-9]{3}-[0-9]{4}`
4. `((UC)+G?C*)+`

Write regular expressions to search for the following. You may wish to reference the fasta file format for the one-letter representations of DNA and amino acids, as well as the codon table for converting between DNA and amino acids.

5. Poly-A tails, which are repeated As at the end of a string.
6. The DNA sequence of a protein that has a start codon, at least one amino acid, and a stop codon. Recall that there is one start codon and several stop codons.
7. The amino acid sequence of the same protein in question 6.

## 4 Grep

`grep` is the classic Unix tool to search files for text.

8. For each the regular expressions in question 6 and 7, write a `grep` command that searches for the regex using `grep -E`. You do not have to escape special characters.
9. **Extra Credit:** same as question 8, except you cannot use `grep -E` or `-P`. You must escape every special character.
10. Revisit the fasta file in `~/bsci238g/public/hw3`. Create a Bash script that prints the number of A,C,T, and Gs in that particular fasta file. The script should also print the number of sequences present. You may need to use more than 1 command to do this.
11. I have two files with lines of text. For each line in file 1, I want to know if it's present in the file 2. Use a single `grep` command to accomplish this.
12. Same as question 11, but use `sort` and `comm` in a pipeline.