Celine Al-Noubani

BIOL 7200 Exercise 5

**Question 1 (100 points)**

In many cases, the formats used to store biological data are not easy for humans to read and extract information from. Therefore, it is valuable to be able to write code to transform commonly used formats into human-readable representations.

An example of such data is sequence alignments. Sequence alignments are often represented in FASTA format. Alignments look much like the FASTA files you have been working with so far in this course, but they can include gap characters to indicate positions that are absent in one or more sequence.

For a human to look at a FASTA formatted alignment and identify the shared and distinct positions is quite difficult. Have a look at the following alignment, for example.  
A close up of a dna

Description automatically generated  
Identifying bases that are the same requires you to scan through and compare every position in the two sequences manually. This becomes even more intolerable if you recall that FASTA format allows sequences that can be millions of bases long and split over multiple lines. Manual comparison of FASTA format sequences in those cases is impossible.

Luckily, we can convert FASTA formatted sequences to more comfortably viewed forms. We can line up the sequences for easy viewing and can adda additional information to assist with identifying which bases are the same and which are different. For example, with a fairly simple transformation, we can render the above alignment like this:  
A close up of a number

Description automatically generated  
It is much easier to view that alignment and spot the positions that differ.

Your task this week is to write a script to perform the above transformation. Your script should do the following:

1. Read a file whose path is provided as a command line argument. You should assume the file contains  
   aligned DNA sequences in fasta format (i.e., the sequences are the same length).
2. Print the sequences to the terminal without headers
3. As seen in the above example, add a pipe symbol between bases that are identical and a space between bases that differ.

Your script must take command-line input. Do not hard-code the path to the sequence file. The usage of your script should be

<script name>.py <FASTA file>

In order for your script to have the described usage, you need a shebang. Unlike Bash, you can't assume that Python is in the user's /usr/bin/ directory. You must use a portable shebang (see the lecture slides for an example). From now on, incorrect shebangs carry penalties! This is true for every executable script you submit.

You do not need to worry about representing long sequences. Specifically, you don't need to account for how lines wrap in a terminal window. You can assume that each sequence will fit on a single line in the terminal.

Contents of Script:

#!/usr/bin/env python

import sys

fa\_file = open(sys.argv[1])

contents = fa\_file.readlines()

new\_list = []

for i in range(1, len(contents), 2):

    new\_list.append(contents[i].strip())

x,y = new\_list

print(x)

counter = 0

for char in x:

    if char == y[counter]:

        print("|", end = "")

    else:

        print(" ", end = "")

    counter += 1

print()

print(y)

fa\_file.close()

Running in Terminal:

