Projects submission
Eman Diyab
Ediyab@purdue.edu

Introduction to Bioinformatics BIOL59500 – Fall 2020

Project 1:

Alignments (DNA + Protein) alignment (40 pts-55 pts):

This program is implemented in Python. I used the package ***BioPython*** to implement the local and global alignments with different options. I have implemented all the required functions in the question (listed below).

To run this program please use the following command:

```
Python align_Ediyab.py <DNA fasta file name> <Protein fasta file name >
```

I have attached in the folder 4 fasta files to test my program, 2 short and 2 long sequences.

In this code, all the following functions are implemented:

1- Read the fasta files and extract the sequence from them.
2- Translate the DNA sequence to Protein sequence (considering the forward reading frames).
3- Calculate the alignment using score matrix (BLOSSUM62).
4- Format the alignment in the pairs of lines format.
5- Print all alignments ordered with the “Alignment # printed in the beginning”
6- Print the alignment score for each alignment.
7- DNA sequence is printed (90 bases) then the corresponding 30 translated amino acids along with the 2nd sequence alignment. Like the following format:

```
DNA_sequence_1 (90 bases)
Translated_sequence_1 (30 amino acids)
|. |. |. |. |. |. |. ||||. |.
Protein_sequence_2 (30 amino acids)
```

8- Both local and global alignments are implemented with default settings (match=1, no penalty for mismatch or gap).
9- Both local and global alignments are implemented with user parameters (independent gaps).
10- Both local and global alignments are implemented with user parameters (affine gaps).

Project 2: (I know it is not from the projects list, but I liked to implement it from scratch ☺)

Needleman- Wunsch global alignment from scratch:

This program is implemented in Python. I didn’t use any packages in this code but ***sys.***

To run this program please use the following command:

```
Python NW.py <sequence1 > <sequence2 >
```

For example: Python NW.py AGGT ACT_G

In this code, all the following functions are implemented in the following sequence:

1- First the dynamic programming table/matrix is created.
2- All cells are initialized to 0.
3- Fill out first column the first row.
4- Fill out all other values in the score matrix by choosing the maximum of:
    a. Match: the score from diagonal cells.
    b. Insert or delete: the score from top, left cells.
5- Trace back the alignment.
6- Create the alignment by adding the correct letter and gaps in place.
7- Print the alignment along with it’s score.

This project was so interesting because I know it is a single line function call if I use ***BioPython.***

Thank you so much for your time and efforts to teach us, I hope you have a nice day ☺
Eman Diyab.