Part 1 - CS MAJOR

This programming assignment involves introducing the Biopython package to perform pairwise sequence alignment to determine if two sequences from two different organisms are similar or related to each other.

About Biopython:

Biopython is a set of libraries to provide the ability to deal with ”things” of interest to biologists working on the computer. In general this means that you will need to have at least some programming experience (in python, of course!) or at least an interest in learning to program. Biopython’s job is to make your job easier as a programmer by supplying reusable libraries so that you can focus on answering your specific question of interest.

One thing to note about Biopython is that it often provides multiple ways of doing the same thing. This can be frustrating since one often wants or needs to know just one correct and efficient way to do something. However, having multiple ways to accomplish the same task can be a real benefit because it gives you lots of flexibility and control over the libraries.

Instructions:

1. If Biopython is not already installed in Python 3.x on your computer, then in a shell window, execute the following command to install it.

```
pip3 install biopython
```

2. Read instructions about performing pairwise local and global alignment using Biopython at http://biopython.org/DIST/docs/api/Bio.pairwise2-module.html

3. Create a Python module named <your initials>_pairwise_alignment.py that reads in the protein sequences in the files named seq1.txt and seq2.txt and both globally and locally aligns them using the BLOSUM62 matrix. Make sure to use the “format_alignment” function in the Biopython package to print out the alignment results.

ic_pairwise_alignment.py ×

```python
with open("seq1.txt", "r") as seq1_file:
    lines = seq1_file.readlines()
    seq1 = lines[1].strip()

with open("seq2.txt", "r") as seq2_file:
    lines = seq2_file.readlines()
    seq2 = lines[1].strip()
```

```python
# Global alignment using BLOSUM62 matrix
# and print the results to the console (using format_alignment function)
print("Global alignment starting====================")
for a in pairwise2.align.globaldx(seq1, seq2, matrix):
    print(pairwise2.format_alignment(*a))
print("Print global successful====================")


# Local alignment using BLOSUM62 matrix
# and print the results to the console (using format_alignment function)
print("Local alignment starting====================")
alignments = pairwise2.align.localdx(seq1, seq2, matrix)
for a in alignments:
    print(pairwise2.format_alignment(*a))
print("Local alignment successful====================")
```

4. Which alignment, global or local, performed the best?

Both global and local seem to have performed the same as the score is the same for both a global alignment and local alignment. This is interesting and can be caused if the two sequences are very similar.

```
Global alignment starting====================
M--A---ALSGGGG------GG-AE---P-GQALFNGDM-E----P--EAGAGAGAA-AS--SAA--DP--AIPE--EVWNIKQM--I-KLTQEHIEAL--LDKFGG
|  |   | .         || |    | |.   | .  |    |  |  .| |   ||  .    |   |.    .  || |   . ||      |  |
MEVAVEKA-A----AAAAPAGGPA-AAAPSGE---N-E-AESRQGPDSE--SG-G--EASRLN--LLD-TCAV--CHQ--NI-Q-SRVPKL-------LPCL-----
  Score=142
```

```
Local alignment starting====================
1 M--A---ALSGGGG------GG-AE---P-GQALFNGDM-E----P--EAGAGAGAA-AS--SAA--DP--AIPE--EVWNIKQM--I-KLTQEHIEAL--L
  |  |   | .         || |    | |.   | .  |    |  |  .| |   ||  .    |   |.    .  || |   . ||      |  |
1 MEVAVEKA-A----AAAAPAGGPA-AAAPSGE---N-E-AESRQGPDSE--SG-G--EASRLN--LLD-TCAV--CHQ--NI-Q-SRVPKL-------LPCL
  Score=142
```

5. Use tools at the NCBI portal (you should now know or have an idea how to do this without being told which specific tool to use) to identify the type (i.e., name) of protein and organisms represented by the sequences in seq1.txt and seq2.txt

Sequence 1 in seq1.txt is the transcription intermediary factor 1-alpha isoform 1 protein sequence. This specific protein sequence can be found in Mus musculus (house mouse).

transcription intermediary factor 1-alpha isoform 1 [Mus musculus]

NCBI Reference Sequence: NP_659542.3

Identical Proteins FASTA Graphics

Go to:

```
LOCUS       NP_659542               1051 aa            linear   ROD 10-SEP-2023
DEFINITION  transcription intermediary factor 1-alpha isoform 1 [Mus musculus].
ACCESSION   NP_659542
VERSION     NP_659542.3
DBSOURCE    REFSEQ: accession NM_145076.4
KEYWORDS    RefSeq; RefSeq Select.
SOURCE      Mus musculus (house mouse)
  ORGANISM  Mus musculus
```

Sequence 2 in seq2.txt is the serine/threonine-protein kinase B-raf isoform 2 protein sequence. This specific protein sequence can be found in Homo sapiens (humans).

serine/threonine-protein kinase B-raf isoform 2 [Homo sapiens]

NCBI Reference Sequence: NP_001341538.1

Identical Proteins FASTA Graphics

Go to:

```
LOCUS       NP_001341538             767 aa            linear   PRI 15-OCT-2023
DEFINITION  serine/threonine-protein kinase B-raf isoform 2 [Homo sapiens].
ACCESSION   NP_001341538 XP_005250102
VERSION     NP_001341538.1
DBSOURCE    REFSEQ: accession NM_001354609.2
KEYWORDS    RefSeq.
SOURCE      Homo sapiens (human)
  ORGANISM  Homo sapiens
```

6. Submit your code and this worksheet. NOTE: do not use absolute paths in your code. Your submission should work "relative" to any Python working directory that your code is placed and run.

As long as seq1 and seq2 are named respectively, this should allow access if in the same directory.

```python
# Read in the sequences from the files: seq1.txt and seq2.txt
# and store them in variables
with open("seq1.txt", "r") as seq1_file:
    lines = seq1_file.readlines()
    seq1 = lines[1].strip()

with open("seq2.txt", "r") as seq2_file:
    lines = seq2_file.readlines()
    seq2 = lines[1].strip()
```