BT 3051 — Data Structures and Algorithms for Biology

July-Nov 2019

Assignment 1

27th August 2019

Due date: 12th September, 2019 @ 17:00 **Maximum marks: 80**

Instructions: Write Python codes to solve the problems mentioned below. If you need any assistance, feel free to write to me or the TAs via Piazza (private note). Evaluation will be based on the codes and the logic.

Academic Integrity: You are allowed to discuss the problems verbally with your friends, but copying or looking at codes (either from your friend or the Web) is not permitted. Transgressions are easy to find, and will be reported to the "Sub-committee for the Discipline and Welfare of Students" and will be dealt with very strictly. Mention any collaboration (discussions only!) in your solutions.

Late submission penalties: 1 second – 24 h: 20%; 24–48 h: 40%; > 48h: 60%

Early submission bonuses: > 24h: 5%, > 48h: 10%, > 72h: 20%

Evaluation: Assignments will be evaluated by the TAs within one week of the due date. You can check out your marks and contest them, if needed, for at most one more week post-evaluation, i.e. two weeks from the due date of the assignment.

Problem Statement

1a. (30 marks) **Construct a gene class** with the attributes *sequence*, *geneID* and *length*. Define methods to perform the 3 functions as illustrated below:

```
> a=Gene("DNA1","AGCTGCATGTACGTAGTCA")
> b=Gene("DNA2","TCATCGGTAGCAATTT")
> GC_content(a)
47.4
> c=a+b
> c.sequence
"AGCTGCATGTACGTAGTCATCATCGGTAGCAATTT"
> -a
"TCGACGTACATGCATCAGT"
```

(GC content is the percentage of G&C bases in a sequence. Complementary base pairing rule: A-T, G-C)

1b. (50 marks) **Identify Open Reading Frames, Translate mRNA to Protein, Calculate Protein Mass**

(This problem is adapted from http://rosalind.info/problems/orf/ and http://rosalind.info/problems/prtm/).

The 20 commonly occurring amino acids are abbreviated as single letters from the English alphabet. These symbols are used to construct protein sequences. Information to synthesise proteins is encoded in mRNA molecules. A codon is a sequence of three RNA/DNA nucleotides that translates to a specific amino acid or stop/start signal during protein synthesis, according to a genetic code. Either strand of a DNA double helix can serve as the coding strand for RNA transcription. Hence, a given DNA string implies six total reading frames, or ways in which the same region of DNA can be translated into amino acids: three reading frames result from reading the string itself, whereas three more result from reading its reverse complement. An open reading frame (ORF) begins with a start codon, ends with an end codon, and doesn't have any other stop codons in between.

Given: A file with DNA sequences in FASTA format (each of length at most 10kbp). See http://rosalind.info/glossary/fasta-format/ for details on the FASTA format.
Return: Every distinct candidate protein string that can be translated from ORFs of the DNA sequences, along with the total mass of the protein strings. The strings can be returned in any order.

Consult the DNA codon table from the file `DNA_TABLE.txt` and the monoisotopic mass table from the file `PROT_MASS.txt`. Do not use biopython.

(Can you reuse any piece of code from the previous question?)

Your code should read something like this (use the template files uploaded):

```
# Homework Header as usual
#
#
#

import sys
import doctest

def read_FASTA(fname):
        """ (str) -> (list of tuples)
        # function body with documentation
        """
        return sequences # a list of (sequence_name , sequence) tuples

def identify_orfs(dnaStrand):
        """ (str) -> (list of strings)
        # function body with documentation
        """
        return frames # a list of orf strings

def translate_DNA(dnaStrand,  translation_table = 'DNA_TABLE.txt'):
        """
        # function body including documentation and test cases
        >>> translate_DNA('AUGUAUGAUGCGACCGCGAGCACCCGCUGCACCCGCGAAAGCUGA')
        MYDATASTRCTRES
        """
        return protein # the protein string

def compute_protein_mass(protein_string):
        """
        #function body including documentation and test cases
```

```
    >>> compute_protein_mass('SKADYEK')
    821.392
    """
    return mass # the mass of the protein string as a float

if __name__ == '__main__':
    #DO NOT CHANGE THE FOLLOWING STATEMENTS
    for seq_name , seq in read_FASTA("hw1b_dataset.faa"):
        print (seq_name+”:”)
        for orf in identify_orfs(seq):
            protein=translate_DNA(orf)
            print (protein,compute_protein_mass(protein))
```

How to Submit your Homework

- Use the template files provided.
- Submit your assignment ONLY via the submission link: http://tinyurl.com/bt3051-submit.
- You should not be signed into Dropbox while uploading this file (or use an incognito window to open the link), so that you can enter the following details during submission, instead of Dropbox auto-filling it:
  - First Name: Roll Number
  - Last Name: Your Full Name
  - E-mail: Your `smail` id
- Save your solution files as `hw1a.py` and `hw1b.py`. Do not use different filenames!
- Each of your submission files, `hw1a.py` and `hw1b.py` should begin with the **header information** shown below — the number of the assignment, your roll number, your collaborators' roll number(s), and approximately how much time you took to solve the problems in that part of the assignment.
- **Submissions not adhering to any of the above instructions will not be evaluated**.
- Also do not send the files by e-mail — obviously, they will not be evaluated.

```
#BT3051 Assignment 1a
#Roll number: BE13B001
#Collaborators: CH12B001, EE13B001
#Time: 1:15
```

Attention: This assignment is fairly simple; the main purpose of this assignment is to ensure that you can write simple Python programs, document them well, and write sensible test cases.