**S13674**

**2017s16415**

**Hendalage D.P.B.**

**BT-3172: Special Topics in Bioinformatics: Computing for Biologists**

**Lab 2: Use of Python data structures and control structures in bioinformatics**

In this practical you will learn how to use Python data structures and control structures to solve biological problems.

DREB1A (Dehydration-responsive element-binding protein 1A) is an important protein which is expressed in response to abiotic stresses such as drought stress and salinity stress in plants. It is important to study this protein and its involved biological pathways when studying plant abiotic stress responses. During this practical, you will work with DREB1A protein in rice.

After using PyCharm to write your scripts, copy the codes to the appropriate space below the questions. Also, submit the Python files separately so they can be tested. Use the following format to name the script: YourIndexNo\_PrimaryQuestionNo\_SecondaryQuestionNo.py.

1. Using control structures to differentiate between DNA and amino acid sequences
   1. Use the UniProt knowledgebase to search for DREB1A protein in rice subspecies japonica. Write its UniProt identifier number below. Is the protein reviewed by the knowledgebase? What is the importance of this reviewed status in the UniProt knowledgebase?

**Q64MA1**

**Yes it is reviewed by the knowledgebase**

UniProtKB/Swiss-Prot (reviewed) is a high quality manually annotated and non-redundant protein sequence database

* 1. Now download the FASTA amino acid sequence for the DREB1A protein in rice and rename it to “OSDREB1A.fasta”. Furthermore, obtain the RefSeq gene sequence record for the DREB1A protein. Make sure you obtain the most updated RefSeq record. Write the RefSeq gene ID below. Locate the mRNA sequence record for this gene and write its accession number (with version included). Obtain its FASTA record and copy and paste it below the FASTA record of the amino acid sequence in the previously downloaded OSDREB1A.fasta file. Shorten the FASTA headers to only keep their identifiers.

[4347620](https://www.ncbi.nlm.nih.gov/gene/4347620)

XM\_015755426.2

* 1. Write an algorithm to differentiate between the two FASTA records in OSDREB1A.fasta file. The algorithm must correctly identify the mRNA record from the amino acid sequence and replace its Thymine (“T”) bases with Uracil (“U”) bases. Then it should save the resulting transcribed mRNA sequence in a new FASTA file called “OSDREB1A\_mRNA.fasta”. The FASTA header of this file should keep the header of the original mRNA record with the word: “transcribed” added to the end.

Import the fasta file

Create a dictionary

Find the nucleotide sequence

Change the ‘T’ to ‘U’

Print the dictionary

Find the mRNA sequence

Change the header

Save the header and sequence to filec

* 1. Implement the above algorithm in Python.

1. Translating mRNA sequences using a Python script.
   1. Write an algorithm to translate a given mRNA sequence (with Uracil bases instead of Thymine) to its amino acid sequence. For this algorithm, ignore the different open reading frames and begin the translation from the first letter. You have to use a while loop when writing the algorithm. Use the amino acid codons table text file to create a dictionary to store codon to amino acid mappings. End the translation when you hit a stop codon.

Open the codon map file and store as variable

Remove the header and empty lines

Set the codon as key and letters as values in the dictionary

Define a empty string

Open the mRNA file and store as a variable

Remove the header and empty lines

Take three bases and compare with codon table iterate by 3

Add the amino acids in to the empty string

When stop codon meet break the propagation

Write the sequence in another file

Calculate the length of peptide chain

* 1. Implement the above algorithm in python using OSDREB1A\_mRNA.fasta file as the example. Write the translated sequence in an output file in FASTA format
  2. What is the length of the translated amino acid sequence?

230aa

1. Using the graph data structure in Python to calculate the degree of a given protein. For this, you will need the NetworkX Python package installed.
   1. First find protein-protein interaction data using the rice DREB1A Uniprot record. Write the STRING database identifier for this protein below.

4530 (4530.OS09T0522200-01)

* 1. Go to the STRING record of the rice DREB1A protein. How many other proteins does it interact with according to the visualization? Now increase the maximum number of interactions to show to 100.

11 proteins

When increase to 100, 63 proteins

* 1. Now, write a Python program to calculate the degree of the rice DREB1A protein. You have to use the NetworkX module for the calculation. First, download the interaction data in tabular format. Use the downloaded TSV file for the calculation. Write the resulting degree of the program.