Computer Science 6915 - Winter 2020
Assignment 1

Reproduce from scratch in Python3 the KNN classifier for predicting transcription factor (TF) binding preferences discussed in class on Wednesday January 15th. The corresponding slides are in Lecture 2, pages 21 to 27. If you wish, you can experiment with different distance functions and number of neighbours. “Implementing from scratch” means that you need to write the necessary code to perform KNN yourself. You are allowed to use functions available in packages such as Pandas or NumPy (for example, to handle the data and perform vector operations).

Your program should be called TF_KNN.py and it should run in Linux. You program should take three command-line arguments (given in the following order):

1. A filename specifying a tab-delimited plain-text file containing the attributes (X) of the training data (i.e., a string per TF representing the protein sequence of each TF). The file has two columns: the name of the TFs and the sequence of the TFs.
2. A filename specifying a tab-delimited plain-text file containing the output (Y) of the training data. Each column corresponds to the output vector of one TF. Each row corresponds to a specific string of eight characters representing a unique DNA sequence. This file contains a header row indicating the name of the TF.
3. A filename specifying a tab-delimited plain-text file containing the unseen instances for which your program needs to predict the corresponding output vector. The file has two columns: the name of the TF and the sequence of the TF.

For example, we might execute your program as follows:

```
$python3  TF_KNN.py X_train.txt Y_train.txt X_unseen.txt
```

where the $ indicates the terminal prompt. Your program should create a tab-delimited text plain-text file containing the predicted output (Y') for the unseen instances (provided in the file X_unseen.txt). Each column corresponds to the output vector per TF. The file should contain a header row indicating the name of the TF. You can assume all the input files are in the working directory.

In D2L, two files are provided: one with 150 TF sequences (called TF_sequences.txt) and one with the corresponding 150 output vectors. To test your script you will need to remove some of the output vectors and give to your program the corresponding names and sequences in the “X_unseen.txt” file.

For this assignment, submit through D2L the following (one submission per team):

a) Your python code in a single file called TF_KNN.py
b) A PDF file containing a one-page description of your implementation including the pseudo-code of your KNN implementation, the definition of the distance function you used, and explaining how you tested your code to make sure your implementation was correct.

The assignment will be graded based on the correctness of your KNN implementation (including prediction performance, code correctness and meeting specifications), and clarity of the description.

If you need, you can use these computer labs: https://www.mun.ca/computerscience/ugrad/labschedule.php