ABIN Assignment 2

Dr. Debarka Sengupta

14/09/2023

Guidelines

1. Deadline: 18th September 2023, 11:59 pm (Midnight).
2. Late submission: 19th September 2023, 11:59 pm (Midnight) (50% of the obtained marks will be deducted).
3. Further late submissions after the 19th of September will be awarded zero.
4. All coding assignments must be submitted as .ipynb files with proper comments.
5. Standard IIIT-D plagiarism policy applies.
6. The assignments have to be submitted in the following manner:
   (a) Create a single Jupyter notebook with proper demarcation of questions and responses (text, code, command, explanation, output, graphs). The accepted language is Python.
   (b) A PDF file comprising the entire Jupyter Notebook and a separate Jupyter Notebook file needs to be uploaded (zip format) at the assignment link on Google Classroom before the deadline.
   (c) One group should submit only from the registered submission ID during group formation. Rest of the members must turn in the assignments with private comments mentioning the name, roll, and submission id. Any violation of above will disqualify your submission from evaluation.
   (d) The name of the PDF and jupyter notebook file should be a combination of the group number and assignment number. For example, `group1_1.pdf`, where 'group1' is the group number and '1' is the assignment number.
7. No shift of the deadline is allowed.

article

Question 1 (5 Points)

Write a function called `count_kmers` which takes as input a string and an integer k, and returns a dictionary as output. The keys of the dictionary should be all the k-mers that were seen, and the values should be the number of times each k-mer occurs in the string. Here input would be a fasta file containing nucleotide sequences.

Question 2 (5 Points)

Given two strings s and t (of possibly different lengths), the edit distance $dE(s,t)$ is the minimum number of edit operations needed to transform s into t, where an edit operation is defined as the substitution, insertion, or deletion of a single symbol. The latter two operations incorporate the case in which a contiguous interval is inserted into or deleted from a string; such an interval is called a gap. For the purposes of this problem, the insertion or deletion of a gap of length k still counts as k distinct edit operations.

Given: Two protein strings s and t in FASTA format (each of length at most 1000 aa).

Return: The edit distance $dE(s,t)$.

Sample Dataset: $s = \text{PLEASANTLY}, t = \text{MEANLY}$

Question 3 (5 Points)

An alignment of two strings s and t is defined by two strings s' and t' satisfying the following three conditions: 1. s' and t' must be formed from adding gap symbols "-" to each of s and t, respectively; as a result, s and t will form subsequences of s' and t'. 2. s' and t' must have the same length. 3. Two gap symbols may not be aligned; that is, if $s'[j]$ is a gap symbol, then $t'[j]$ cannot be a gap symbol, and vice-versa.

We say that s' and t' augment s and t. Writing s' directly over t' so that symbols are aligned provides us with a scenario for transforming s into t. Mismatched symbols from s and t correspond to symbol substitutions; a gap symbol $s'[j]$ aligned with a non-gap symbol $t'[j]$ implies the insertion of this symbol into t; a gap symbol $t'[j]$ aligned with a non-gap symbol $s'[j]$ implies the deletion of this symbol from s.

Thus, an alignment represents a transformation of s into t via edit operations. We define the corresponding edit alignment score of s' and t' as $dH(s', t')$ (Hamming distance is used because the gap symbol has been introduced for insertions and deletions). It follows that $dE(s,t) = \min dH(s', t')$, where the minimum is taken over all alignments of s and t. We call such a minimum score alignment an optimal alignment (with respect to edit distance).

Given: Two protein strings s and t in FASTA format (with each string having length at most 1000 aa).

Return: The edit distance $dE(s,t)$ followed by two augmented strings s' and t' representing an optimal alignment of s and t.

Sample Dataset:

$s = \text{PRETTY}, t = \text{PRTTEIN}$