Assignment -1: Central Dogma of Life and Strings

IIIT-DELHI
Due Date: 21th August 12.00 pm
Total(80 points)
Instructor: Debarka Sengupta

Plagiarism: All submitted codes are expected to be the result of your individual effort. You should never misrepresent someone else's work as your own. In case any plagiarism case is detected you will get one grade reduction in final examination. Cite the resource wherever using other's code.

Instructions:

1) Allowed programing language are python and C . No external libraries would be allowed. If you see an ambiguity or inconsistency in a question, please seek a clarification from the teaching staff. Only in Problem 1 you are allowed to use Bio-python for parsing the file.

2) You must submit your working solution on backpack on the deadlines page from where you have downloaded this assignment instructions sheet. No extensions on deadline. If you fail to submit within the time limit then your solution will not be evaluated.

3) Mention your enrollment no and name at starting of the each file . Write a clean code with proper comments at appropriate places as it will be checked.

4) Store the each problem with rollno_problemX_Y.py or rollno_problemX_Y.c where X is the problem no and Y is task no and upload a zip folder with rollno_name_assignmentX.zip containing all the codes and assignment report.Codes won't be checked if they don't follow the guidelines.

Hardware and Software Resources:

It is mandatory that you should do version controlling of all your homeworks. We will be using https://bitbucket.org/ for this purpose. Get a login id for free on bitbucket and create the homework repository. All homeworks should be saved inside the repository named hw_1 with your roll number as suffix. If your roll number is 1234 then the repository name will be hw1_1234. Every homework repository should have access level set as "private". You have to use the "share" option for your homework repository to share it with both of your TAs. You can share using their email ids. We will not entertain any date extension requests if you are not doing version controlling and accidently deleted your homework.You can also use HPC for doing your homework if your machine can't handle large files.

Problem 1: Perform Translation and Transcription. [20 points]

TB is a bacterial disease which in humans is usually caused by an organism called Mycobacterium tuberculosis (M. tuberculosis). TB is an abbreviation of the word Tuberculosis and is how people usually refer to the disease. One-third of the world's population is thought to be infected with TB[**1**]. *You have been provided with genbank file*[**2**] *of Mycobacterium tuberculosis H37Rv, complete genome. The task to do are mentioned below:*

TASK

1] Use a genbank parser library [Biopython,scikit-bio etc] to parse the file and extract all CDS region from the provided Mycobacterium tuberculosis input genbank file(mtb.gb) and store them in the fasta file. A sample output file for is attached with the document titled as problem1_gene.fasta. Write your own parser to read the fasta and genbank files. Compare speed of both the parser and comment on both the methods.**[5 points]**

2] Write a function for transcription of DNA sequence to RNA sequence. No inbuilt function like sorting,searching or any external library function should be used over here.**[3 points**]

3] Using the function of transcription ,output of program 1 and MTB genbank file write the code to transcribe the CDS region into Proteins and store them in fasta file. The codon table is also also provided to you as codon_table.tsv . A sample output file is attached with document titled as problem1_protein.fasta.**[5 points]**

4] Write a code to calculate the GC content of DNA sequence. Report GC content of each CDS sequence **[2 points]**

5] Make a pdf report explaining all the function you wrote for problem 1 along with explaining genbank format ,CDS region and central dogma of life **[5 points]**

References:

[1] http://www.who.int/mediacentre/factsheets/fs104/en/
[2]https://www.ncbi.nlm.nih.gov/Sitemap/samplerecord.html

Problem 2:String Permutation and Counting. [15 points]

String manipulation and counting is one of the important aspect in bioinformatics. The frequency of a set of k-mers in a species' genome, in a genomic region, or in a class of sequences, can be used as a "signature" of the underlying sequence.Complete the task mentioned below

TASK

1] Write a simple enumeration algorithm, either recursive or iterative, that prints all k-mers consisting of 5 letters {-, A, C, G, T}.Compare the time of both recursive and iterative approaches and mention the time complexity of the algorithm **[5 points]**

2] 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 multiple sequences. Use fasta parser created by you.**[5 points]**

3] In molecular biology and bioinformatics, the consensus sequence**[1]** is the calculated order of most frequent residues, either nucleotide or amino acid, found at each position in a sequence alignment. A consensus string c is a string of length n formed from our collection by taking the most common symbol at each position. There can be multiple consensus sequence possible. You will have to print any one of them if there are multiple. Use the fasta parser created by you.**[5 points]**

Input: Input would be a fasta file like

```
>Seq1
TGTAGAA
>Seq2
AGTACCC
>Seq3
AGTGCTC
>Seq4
AGTATAG
```

Output: The program should output :

```
Consensus: AGTACAC
Frequency Matrix:
[[3, 0, 0, 3, 0, 2, 1],
 [0, 0, 0, 0, 2, 1, 2],
 [0, 4, 0, 1, 1, 0, 1],
 [1, 0, 4, 0, 1, 1, 0]]
```

Where the frequency matrix rows are ACGT and cols are the position of the string .

References:

[1]https://en.wikipedia.org/wiki/Consensus_sequence

Problem 3: String Matching [45 points]

A common problem in text editing and DNA sequence analysis: finding strings inside other strings.The string matching problem is this: given a smaller string P (the pattern) that we want to find occurrences of in T.
For example, in the string "TGAGAGAC", the pattern string "GAG" occurs at offset 1 and 3.

TASK

1] Write a naive string matching algorithm. **[5 points]**

2] Make a new version of the naive string matching algorithm that allows up to 1 mismatches per occurrence. **[10 points]**
Example: In the string "TGAGAGAC", the pattern string "GAG" occurs at shifts 1 and 3 and 5 where 5 offset having 1 mismatch

3] Implement Knuth-Morris-Pratt algorithm**[1]**.**[12 points]**

4] Explain the all three algorithms in a pdf file with example and compute the time complexity of them.**[8 points]**

5] Implement pascal triangle using recursion. **[10 points]**

References

1] Knuth, Donald; Morris, James H.; Pratt, Vaughan (1977). "Fast pattern matching in strings". SIAM Journal on Computing. 6 (2): 323–350. doi:10.1137/0206024