COMP3353 Bioinformatics

Assignment 1: Dealing with sequence

Due Date: Thursday, 27 Sep 2020 @ 11:59 pm

Submission:
File formats including 1) Microsoft Word document, 2) Latex PDF and 3) Markdown are acceptable. Please put your 1) name, 2) university number, and 3) email address on the first page of your submission. Please number your answers to each problem and sub-problem properly. If the program code in your submission is written in indentation-sensitive languages such as Python, please consider submitting your code separately with the filename properly numbered to the question. You can use Latex or Markdown that supports code-embedding as well. Please submit your answers in Moodle. Email submission is not suggested, but in case you have problems with Moodle, please email the answers to the TA.

You can click on the hyperlinks to get a more detailed definition or explanation of the word or concept.

Section 1: Sequence Conversion

Question 1 (5 points): Reverse complement of a DNA string:
In DNA, symbol `‘A’` and `‘T’` are complements of each other, as are `‘C’` and `‘G’`. Reverse complement of a DNA string *s* is the s^c formed by firstly reversing the symbols of *s* (`“CATT”` -> `“TTAC”`), and then substituting the symbols with respective complements (`‘A’` -> `‘T’`, `‘T’` -> `‘A’`, `‘G’` -> `‘C’`, `‘C’` -> `‘G’`)

Given: A DNA string *s* of length at most 1000 bp.
Return: The reverse complement s^c of *s*.
Sample input: “`GAACTATT`”
Sample output: “`AATAGTTC`”
Requirements: There exists no limitation on how you take in the input and generate the output. 1) Pipe-chained Linux commands, 2) scripts and, 3) source code are all accepted.

Question 2 (5 points): Translating RNA into Protein:
Use the RNA codon table to encode (translate) the RNA sequence into protein string. Each codon consists of three nucleotides, translating to a single amino acid (e.g., 'AUG' in an RNA

sequence translates to a protein symbol 'M', the abbreviation of 'Methionine'). The translation does not always start from the first nucleotide of a given sequence. Instead, it starts at the first occurrence of 'AUG', with 'M' as the first output of the translation. The translation does not always stop at the last nucleotide of a given sequence either. Instead, it ends at a stop codon, with the remaining nucleotides in the sequence untranslated. A stop codon does not translate to any amino acid, it does only one job – stop the translation.

Given: An RNA string *s* corresponding to a strand of mRNA (of length at most 10 kbp).
Return: The protein string encoded by *s*.
Sample input:
`“GAUGGGGAGUACCCGUUAAAACGGGAUGGCCAUGGCGCCCAGAACUGAG”`
Sample Output:
`“MGSTR”`
Requirements: There exists no limitation on how you take in the input and generate the output. 1) Pipe-chained Linux commands, 2) scripts and, 3) source code are all accepted.

Section 2: Pattern Searching

Question 3: Pattern Searching of a DNA string (the questions required an answer are underlined and in bold):

a. Download the human hg38 chromosome 22 sequence file in FASTA format:
   http://hgdownload.cse.ucsc.edu/goldenPath/hg38/chromosomes/chr22.fa.gz
   This FASTA file is in 5′ to 3′ direction.

b. Restriction enzymes cleave DNA molecules at or near a specific sequence of bases. For example, the HindIII (pronounced “Hin D Three”) enzyme cuts at the "/" in either this motif: 5'A/AGCTT3'. **How many perfectly matching HindIII restriction enzyme cut sites are there on chr22?** Don't worry about sites that span two lines - just care about sites that are **fully contained** on a single line. No need to consider the reverse strand. (5 points)

c. **How many HindIII cut sites are there on chr22, assuming that a mutant form of HindIII will tolerate a mismatch in the second position?** Think about ways in which you could best test for all the possible DNA combinations. There are a few valid approaches. Don't worry about sites that span two lines - just care about sites that are **fully contained** on a single line. No need to consider the reverse strand. (15 points)

d. Given the chr22 FASTA file we have used for the rest of this homework, devise a conceptual strategy (no need for source code) for using Unix commands you know of so far (and possibly a wish list of methods that you are not yet aware of) to describe how you would conduct an *in-silico* digest of chr22 using HindIII. **That is, if HindIII cuts at the sites described above, how could you predict exactly the sequences and their lengths that would result from chr22?** (10 points)

e. **Describe in prose an approach to finding all occurrences of the motif “`ATTCCGAATCAGGGT`” on chromosome 22. You also need to be able to find any motif that is one mismatch away (e.g., “`TTTCCGAATCAGGGT`”) from this motif.** (10 points)

f. **What is the longest consecutive simple repeat of CAG (consider both capital letters and small letters) occurring on a single line in the FASTA file?** Hint: can try regular expression (regex) matches with grep. (10 points)

g. **How many distinct gaps (The number of “N” in a gap should be ≥1.) are there on chromosome 22?** Show your work by providing the script you used. The “N” strings at the very beginning and the very end are not gaps. (10 points)

Section 3: Don’t worry, be happy

Please complete EITHER Q4 OR Q5.

Question 4 (30 points):

Given a DNA string t, its GC-content is defined as the percentage of nucleotides in t that are either ‘C’ or ‘G’, i.e. $(\text{number of }'\mathtt{C}' + \text{number of }'\mathtt{G}')/|t|$. For example, the GC-content of “AGCTATAG” is 3/8 = 37.5%. Given a DNA string s and some fixed length k ($k < |s|$), we can see that s has exactly $|s| - k + 1$ substrings of length k. In this question, we define a substring to be *high-GC* if its GC-content is > 70%. Please include "N” when you calculate the GC-content.

Hereafter, we take s to be the human hg38 chromosome 22, which can be downloaded from http://www.bio8.cs.hku.hk/comp3353/chr22.fa

1. Let $k = 100$. How many length-k substrings of s are high-GC?
2. We want to “filter away” those high-GC substrings found in part (a) from s using a masking procedure. Precisely, for every high-GC length-k substring, we replace all its k nucleotides in s by ‘N’ or ‘n’ (preserving the original letter case). You should keep the first line of the file, which contains the chromosome name, unchanged. Compress your output file using ‘gzip’. Rename the compressed file as “chr22.masked.fa.gz” and upload it to Moodle.

Question 5 (30 points):

Please write an essay (more than 400 words and less than 800 words) on how the completion of human genome project could benefit your research or work in the future.