COMP3353 Bioinformatics

Assignment 3: Variant Calling and Genome Assembly

Due Date: Thursday, 19 Nov 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 0: Prerequisites

1.1 Download the VCF file

This VCF file stores a subset of variations of sample NA12878 chromosome 21 Whole Exome Sequencing (WES) using genome version GRCh38.

```
curl -s
http://www.bio8.cs.hku.hk/comp3353/NA12878_chr21_GRCh38_assignment3.vcf >
NA12878_chr21_GRCh38_assignment3.vcf
```

Learn more about the VCF format 4.2.

Section 1: Know more about the VCF file

Note: You can either 1) use existing tools such as “bcftools”, “vcftools”, 2) use Linux commands, or 3) implement your program to get the results. But try your best to produce your results using only the existing tools, to learn how to make the best of the existing tools, you will need to not only read the README of the tools but also to play with the tools (run the tools without any option so as to show the help info).

Question 1 (5 points):

How many variants are in the VCF file? Show your work.

Question 2 (5 points):

How many variants are in the VCF file with QUAL greater than 40? Show your work.

Question 3 (5 points):

How many transitions were found, regardless of quality? How many transversions? Show your work.

Hint: Have a look at this page. Based on the sequencing method, find out the expected TiTv ratio. I suggest you use `bcftools stats`.

Question 4 (5 points):

What is the ratio of transitions and transversions of this VCF? Is what you would expect? Show your work.

Question 5 (5 points):

How many insertions were found? How many deletions? Show your work.

Question 6 (5 points):

How many SNPs had a total depth (DP in INFO field) greater than or equal to 20? Show your work.

Section 2: Programming, ergo sum

Question 7: Transitions and Transversions (20 points)

For DNA strings s_1 and s_2 having the same length, their transition/transversion ratio $R(s_1, s_2)$ is the ratio of the total number of transitions to the total number of transversions, where symbol substitutions are inferred from corresponding mismatched symbols as when calculating Hamming distance (see Assignment 2 Question 6 Counting Point Mutations).

Given: Two DNA strings s_1 and s_2 of equal length (at most 1000 bp).

Return: The transition/transversion ratio $R(s_1, s_2)$ (round to 2 decimal places).

Sample input (Assume the sample input stores in input.txt):

```
>string1
GCAACGCACAACGAAAACCCTTAGGGA
>string2
TTATCTGACAAAGAAAGCCGTCAACGG
```

Sample Output (Output to the screen):

```
0.71
```

Question 8: Enumerating *k-mers* Lexicographically (20 points)

Assume that an alphabet $\mathscr{A}$ has a predetermined order; that is, we write the alphabet as a permutation $\mathscr{A} = (a_1, a_2, \ldots, a_k)$, where $a_1 < a_2 < \cdots < a_k$. For instance, the English alphabet is organized as (A, B, …, Z).

Given two strings s and t having the same length n, we say that s precedes t in the lexicographic order (and write $s <_{\text{Lex}} t$) if the first symbol $s[j]$ that doesn't match $t[j]$ satisfies $s_j < t_j$ in $\mathscr{A}$.

Given: A collection of at most 10 symbols defining an ordered alphabet, and a positive integer n ($n \leq 10$).

Return: All strings of length n that can be formed from the alphabet, ordered lexicographically (use the standard order of symbols in the English alphabet).

Sample Input (Assume the sample input stores in input.txt):

```
A C G T
2
```

Sample Output (Output to the outputQ7.txt)

```
AA
AC
AG
AT
CA
CC
CG
CT
GA
GC
GG
GT
TA
TC
TG
TT
```

Section 3: Don't worry, be happy

Please choose to answer **either** question 9 or 10.

Question 9:

Write a program which takes a set of reads as input (from inputQ9.txt) and constructs the de Bruijn graph for the reads. Here is a sample inputQ9.txt:

```
TACAGT
AGTCAG
ACAGTC
TCAGAT
```

The program should perform these actions:

1. Print out all distinct k-mers from the reads, for k = 3. (10 points)
2. Construct the de Bruijn graph for k = 3. Don't print out anything for this part. (10 points)

Hint: Each node of the de Bruijn graph represents a distinct k-mer in part (1). There is a directed edge from k-mer x to k-mer y if the last (k-1) characters of x equal the first (k-1) characters of y.

3. Print out the de Bruijn graph in the form of an adjacency list. (10 points)

Question 10:

In Lecture 15, we have learned germline variant calling. In Lab Session 3, we have also gained some hands-on experience in calling germline variants. We know that, for mendelian disease diagnosis, germline variant calling would be enough. But in genetic testing for cancer diagnosis, we will need somatic variant calling. Somatic variant calling is harder than germline variant calling because cancer tissue usually contains a mixture of cancer cells and normal cells, i.e., the sequencing results are a mixture of two types of cells. While germline variant calling in human only needs to decide between three possible statuses at a genome position: homozygous reference (freq. of alt. bases = 0), heterozygous alternative (freq. of alt. bases = 0.5) and homozygous alternative (freq. of alt. bases = 1), somatic variant calling also needs to take the proportion of cancer cells in to consideration (calculated as freq. of alt. bases times proportion of cancer cells). And because the proportion of cancer cells could be any decimal number between 0 to 1 and we usually don’t know the proportion *a priori*, the possible outputs of somatic variant calling at a genome position becomes infinite.

Imagine your supervisor has a cancer sample and wants you to work on somatic variant calling. Please do some investigations and write a short essay (400 words to 800 words) on somatic variant calling to convince your supervisor that you can make it. Please do cite properly with APA format. Hints: According to the latest review papers, what are the best tools for somatic variant calling. How they perform. How many computational resources do we need for a single cancer sample? How might we verify the somatic variant calling results (either dry-lab or wet-lab)?