COMP3353 Bioinformatics

Assignment 2: DNA sequencing alignment

Due Date: Sunday, 18 Oct 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 Prerequisites

1.1 Download the FASTQ files

The FASTQ files are resulting from an Illumina paired-end sequencing run (run means "a round of sequencing", or "round") of genomic DNA from Staphylococcus aureus (details). Since this is from a paired-end sequencing run, there are two files - one for each end of each DNA fragment. As such, the two files are named `ERR2337147_1.fastq.gz` and `ERR2337147_2.fastq.gz`.

```
curl
ftp://ftp.sra.ebi.ac.uk/vol1/fastq/ERR233/007/ERR2337147/ERR2337147_1.fastq.g
z > ERR2337147_1.fastq.gz
curl
ftp://ftp.sra.ebi.ac.uk/vol1/fastq/ERR233/007/ERR2337147/ERR2337147_2.fastq.g
z > ERR2337147_2.fastq.gz
gunzip ERR2337147_1.fastq.gz
gunzip ERR2337147_2.fastq.gz
```

Conveniently, the sequence for each end of each fragment is consistently ordered in each file.

For example, let's look at the first line of each file:

```
head -n 1 ERR2337147_1.fastq
```
@ERR2337147.1 M02625:113:000000000-D2C2J:1:1101:11910:4314/1

```
head -n 1 ERR2337147_2.fastq
```
@ERR2337147.1 M02625:113:000000000-D2C2J:1:1101:11910:4314/2

Notice that aside from the /1 and /2 the sequence IDs for each record are identical in each file, indicating that they came from the two 5' ends of the same clonally amplified clusters on the Illumina flow cell.

1.2 Install tools

1.2.1 Install Conda

Conda is an app store for both Linux and Mac. While the system's default app installers including both the "apt-get" in Ubuntu and "Yum" in RedHat require administrator (root) privilege, Conda installs apps in your home folder and solves the libraries dependencies automatically. One thing to note is, in Conda, not all apps that are available in Linux would also be available in Mac, so if Conda can't help you with installing a tool in MacOS, you are always welcomed to use the CS academy server or install a virtual Linux with VirtualBox for Mac.

To install Conda, please follow the guide in this page. If you use the CS academy server, you can use the below commands:

```
curl https://repo.continuum.io/miniconda/Miniconda2-latest-Linux-x86_64.sh >
Miniconda2-latest-Linux-x86_64.sh
sh Miniconda2-latest-Linux-x86_64.sh
```

Use the down arrow to go through the license, type **"`yes`"** to agree with the license. When being prompt for the installation path, press enter to install Conda to your home folder by default. When being asked "Do you wish the installer to prepend the Miniconda2 install location to PATH in your …", type **"`yes`"** to agree. After the installation, run command **"`source ~/.bashrc`"** to load the updated environment variable **"`$PATH`"**. Now you should be able to use `conda` by using the command **"`conda`"**.

1.2.2 Add the "bioconda" channel to Conda

"bioconda" is one of the many available Conda channels. It hosts most of the commonly used bioinformatics tools. Please use the following commands to add the "bioconda" channel to Conda:

```
conda config --add channels defaults
conda config --add channels conda-forge
conda config --add channels bioconda
```

For more details of bioconda, please visit this page.

1.2.3 Install seqtk

Install seqtk through bioconda:

```
conda install -c bioconda seqtk
```

1.2.4 Install fastqc

Install bioawk through bioconda:

```
conda install -c bioconda fastqc
```

Section 1: Know more about the FASTQ file

Note: You can either 1) use existing tools such as "seqtk", "fastqc", etc., 2) use Linux commands, or 3) implement your own program to get the results. But try you 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 pairs of paired-end sequences resulted from this run?

Question 2 (5 points):

How many nucleotides were sequenced in total for this run?

Question 3 (5 points):

What is the overall GC content of the two FASTQ files? (we might use "fastqc" to solve the problem, use command "fastqc -h" for more details, fastqc produces a webpage HTML file for each FASTQ file, please open the HTML files for more details, in the CS academy server, you can click the "Home" icon on your desktop to access the HTML files, or you can use the "scp" command to download the HTML files to your local computer and open them with a web browser) Please write down in your answer the steps (like the commands you have ran) you used to get the overall GC content. Please upload a screenshot of the GC content summary in the HTML file generated by fastqc or other tools. If you wrote your own script, please upload it. Please search the expected GC content of the Staphylococcus aureus genome at https://www.ncbi.nlm.nih.gov/genome. How does the overall GC content of the two FASTQ

files compare to the expected GC content of the Staphylococcus aureus genome from the website?

Question 4 (10 points):

In the left end reads (the file suffixed "_1.fastq", those reads suffixed "/1"), how does the average Phred quality score for the first read (sequence) position compared to the average Phred quality score for the last position? Why is this? (We might use "seqtk fqchk" to solve the problem? Mind the "-q" option, we need to consider all bases)

Question 5 (5 points):

Is the length of every sequence in the FASTQ files the same? How do you know?

Section 2: Mutations

Question 6: Counting Point Mutations (15 points)

Given two strings s and t of equal length, the Hamming distance between s and t, denoted $d_H(s,t)$, is the number of corresponding symbols that differ in s and t. Notice that "Hamming distance" was not taught in class, but it is actually a simplified version of "Edit distance". Please check the example below.

In this figure, the different symbols are in red.

GAGCCTACTAACGGGAT
CATCGTAATGACGGCCT

Given: Two DNA strings s and t of equal length (not exceeding 1000 base pairs).

Return: The Hamming distance $d_H(s,t)$.

Sample Input:
```
GAGCCTACTAACGGGAT
CATCGTAATGACGGCCT
```
Sample Output:
```
7
```

Question 7: Calculating Protein Mass (15 points)

In a weighted alphabet, every symbol is assigned a positive real number called a weight. A string formed from a weighted alphabet is called a weighted string, and its weight is equal to the sum of the weights of its symbols.

The standard weight assigned to each member of the 20-symbol amino acid alphabet is the monoisotopic mass of the corresponding amino acid.

Given: A protein string PP of length at most 1000 amino acid.

Return: The total weight of PP. Consult the monoisotopic mass table.

Sample Input:
```
SKADYEK
```
Sample Output:
```
821.39192
```

Section 3: BWT Encoding

Please complete EITHER question 8 OR 9.

Question 8 (40 points):

Implement a BWT encoder. Please submit your code.

Question 9 (40 points):

1. Please write down the steps to **encode** "TGCAAG" into a BWT string.
2. Please write down the steps to **decode** the BWT string "GGT$CCA".

Note: No limitation is imposed on how you want to show the steps. You can write down the steps on papers and submit the photo captures, or you can draw on your iPad Pro with a virtual pencil, or you can just make a table like below.

An example to show steps **encoding** "ACACGT".

Burrows-Wheeler Transform				
1. Input	**2. All rotations**	**3. Sort**	**4. Take the last column**	**5. Output**
ACACGT$	ACACGT$ $ACACGT T$ACACG GT$ACAC CGT$ACA ACGT$AC CACGT$A	$ACACGT ACACGT$ ACGT$AC CACGT$A CGT$ACA GT$ACAC T$ACACG	$ACACG**T** ACACGT**$** ACGT$A**C** CACGT$**A** CGT$AC**A** GT$ACA**C** T$ACAC**G**	T$CAACG