Homework 2

Parallel sequence alignment program using MPI

You are a data scientist who is working with genetics scientists. The biologists have isolated a number of genes and they want to track how often and where those genes occur in the mouse chromosome (DNA) and how many times they are repeated in the chromosome (if any).
The labs prepared two groups of files: *a)* files for the genes they want to search for: read (seed) files, and *b)* files for the chromosomes they want to search into.

Your program should be able to take multiple seed (read) input files, and one reference genome files (a chromosome) and print out a single file containing the indices and counts of all the matched reads in those chromosomes. For a simple example, if the read file contains:

ACGT

AGGT

ATTTT

ATTTT

and the reference genome files were:

>chr1

ATTTTTACGGTAACGTAGGT

the output file would contain:

chr1, 0, 2

chr1, 12, 1

chr1, 16, 1

Your program should be able to take multiple seed (read) input files, and one reference genome files (a chromosome) and print out a single file containing the indices and counts of all the matched reads in those chromosomes. For a simple example, if the read file contains:

ACGT

AGGT

ATTTT

ATTTT

and the reference genome files were:

>chr1

ATTTTTACGGTAACGTAGGT

the output file would contain:

chr1, 0, 2

chr1, 12, 1

chr1, 16, 1

@ERR192339.1 HWUSI-EAS493_0001:2:1:995:9863/1
NGAGCCGTCACAGCCTGCCGTGGGAAACCTCNCCCCNGNN
+
#)+*)-)),'3-335AAAA887207A55A##########
@ERR192339.2 HWUSI-EAS493_0001:2:1:995:17601/1
NCAGAATTTGCATCATGAACGATGAGCTGATCGTGANGNN
+
#)))'())(+A7AAA0)00.8-8-8AA-AA##########
@ERR192339.3 HWUSI-EAS493_0001:2:1:995:5061/1
NCACAATCTGCTTCCCAGCACTGACAGCCAAGTCACNTNC
...

where the first value on each line is the chromosome, the second is the index where there was a match and the third is the number of reads matching the sequence at that index.

I. Your program needs to:

1. Divide the reads across each process. Each process should get a read file or a portion of a read file to work on. The process should take each read and put it into an unsorted dictionary (*dictionary_name<string,int>)*, where the string is the sequence, and the int is the number of times it occurs in the file.
   1.1. You can ignore any reads with *Ns*.

1.2. You can ignore some lines, the data file looks as shown in Box1 below. The lines in bold are the reads, you can ignore the other lines.

2. Each process should read the chromosome file after reading the seed file. The easiest way to do this would be to have a string and just append each line of the file to it. I would recommend creating some smaller sized test files to start with (Note1).

> **Note1:** If you use the *head -n 100 <file>* command it will print the first 100 lines of the file to the screen, this is an easy way to get a smaller version of a file

3. After the reads have been read into the unsorted_dictionary, and the chromosome file has been read, the process should start at index 0 and check to see if the first 40 characters are in the unsorted dictionary, if they are, it needs to asynchronously send a message to the master process (which will handle writing the sequences to the output file) -- use **MPI_Isend** for this; it should do this for every index in the chromosome. You can ignore any sequences in the reference genome with *Ns*.

1. The master process should repeatedly receive the incoming messages from the other processes, putting the results into another unsorted dictionary, adding to the count if different processes have reads at the same index.

2. When all the other processes have completed, they should send a finish message to the master process (*Note:* you can use a different tag for this), and when the master process receives a finish message from all the other processes then it should write the contents of its dictionary to the output file.

Submit your completed homework as a zip file called '<your_last_name>_hw2.zip'.

A template is created for the program where some of the lines are missing. Complete the missing lines as instructed in the script comments.

II. Rubric

1. **(10%)** Slicing the seed file to slices (chunks) to send them to worker**.**
2. **(20%)** Sending the seed slices with the chromosome file to the workers.
3. **(20%)** Receiving results from works.
4. **(20%)** Sending termination messages to the worker to finish the program.

5. **(10%)** Write the final results in a file as shown in the example mentioned above.
6. **(20%)** Run the program using different number of processes, log the time taken by each run, and plot the results (number of processes on the x-axis and time on the y-axis).

III. Bonus

- **BONUS (10%):** Have the fastest implementation.
- **BONUS (5%):** Have the second fastest implementation.
- **BONUS (20%):** The N's are mysterious genes which the scientists could not identify. They can be any of the genes (A,C, G, T). If you want to help them more, you can consider the combinations of those genes when you replace them N's with them: Have your program handle up to X Ns in each read, and up to Y Ns in each reference genome subsequence. These should be command line parameters.
- **BONUS (10%):** Have your program handle multiple chromosome files as well.

```
http://hgdownload.cse.ucsc.edu/goldenPath/mm10/chromosomes/
```

IV. GETTING THE DATA FILES:

- You can download these files using wget from the command line. This way you can easily download them to remote accounts, instead of downloading them to your computer and then uploading them to the remote account. You can also use the -c command to resume a partially downloaded file, if the download quit for any reason, for example:

```
wget -c http://hgdownload.cse.ucsc.edu/goldenPath/mm10/chromosomes/chr1.fa.gz
```

  will download chr1.fa.gz to the current directory, and continue the download if it was previously broken.

You can get the seed files (the reads) from:

www.ebi.ac.uk/ena

enter in ERP001953.

- To download them from the command line (from the above website):

```
wget -c ftp://ftp.sra.ebi.ac.uk/vol1/fastq/ERR192/ERR192339/ERR192339.fastq.gz
wget -c ftp://ftp.sra.ebi.ac.uk/vol1/fastq/ERR192/ERR192340/ERR192340.fastq.gz
...
```

- You can download the mouse reference genome here:

you want to download the files chr*.fa.gz, which are gripped files of each mouse chromosome in FASTA format, e.g.:

```
wget -c http://hgdownload.cse.ucsc.edu/goldenPath/mm10/chromosomes/chr1.fa.gz
wget -c http://hgdownload.cse.ucsc.edu/goldenPath/mm10/chromosomes/chr2.fa.gz
```

- The files are compress and can be extracted using:

```
gunzip <file_name>
```