Lab 12: Genome Indexing

Background:

The source for the background information for this lab is an article that appeared in a magazine called IEEE Spectrum in July 2013. The article can be viewed at this link for more information:

http://spectrum.ieee.org/biomedical/devices/the-dna-data-deluge

- There are four different types of nucleotides in DNA: adenine (A), thymine (T), cytosine (C), and guanine (G).
- Human DNA has roughly 3 billion nucleotides.
- In June 2000, the Human Genome Project announced successful completion of a draft of a human genome.
- Sequencing a human genome is a computationally intensive process. The first steps begin in a test tube: the DNA strand is split down the middle, it is copied many times, then split into much shorter segments. A machine called a sequencer then identifies the string of nucleotides in each fragment. A computer or a set of computers then takes all of the strands and tries to re-order them using the existing human genome as a reference.
- The difficulty is that there are hundreds of thousands of strands completely out of order with many overlapping and much repetition.
- Developing algorithms to take the sequencer data and sort it accurately and efficiently while keeping the cost down is an active area of research.

Genome Indexing:

In this lab, we will take a look at one of the approaches for sorting the sequencer data based on indexing the genome. Indexing a genome is similar to developing an index for a book. The index of a book contains key words with page numbers where the words are located in the text. In Genome Indexing, we look for key sequences of nucleotides and record their locations in the genome. For example, the sequence 'GATTACA' occurs roughly 697,000 times in the human genome.

The picture on the following page (taken from the IEEE Spectrum article), illustrates the idea. As illustrated in the picture, we scan through the genome looking for some key triplets, in this case: 'AAA', 'ATC', and 'CGG'. When a triplet (codon) is found, the location in the genome is recorded.

Note: the picture makes it appear that we will sort through a matrix, but that is not the case. We will sort through a long vector of nucleotides looking for certain codons.

Genome Indexing

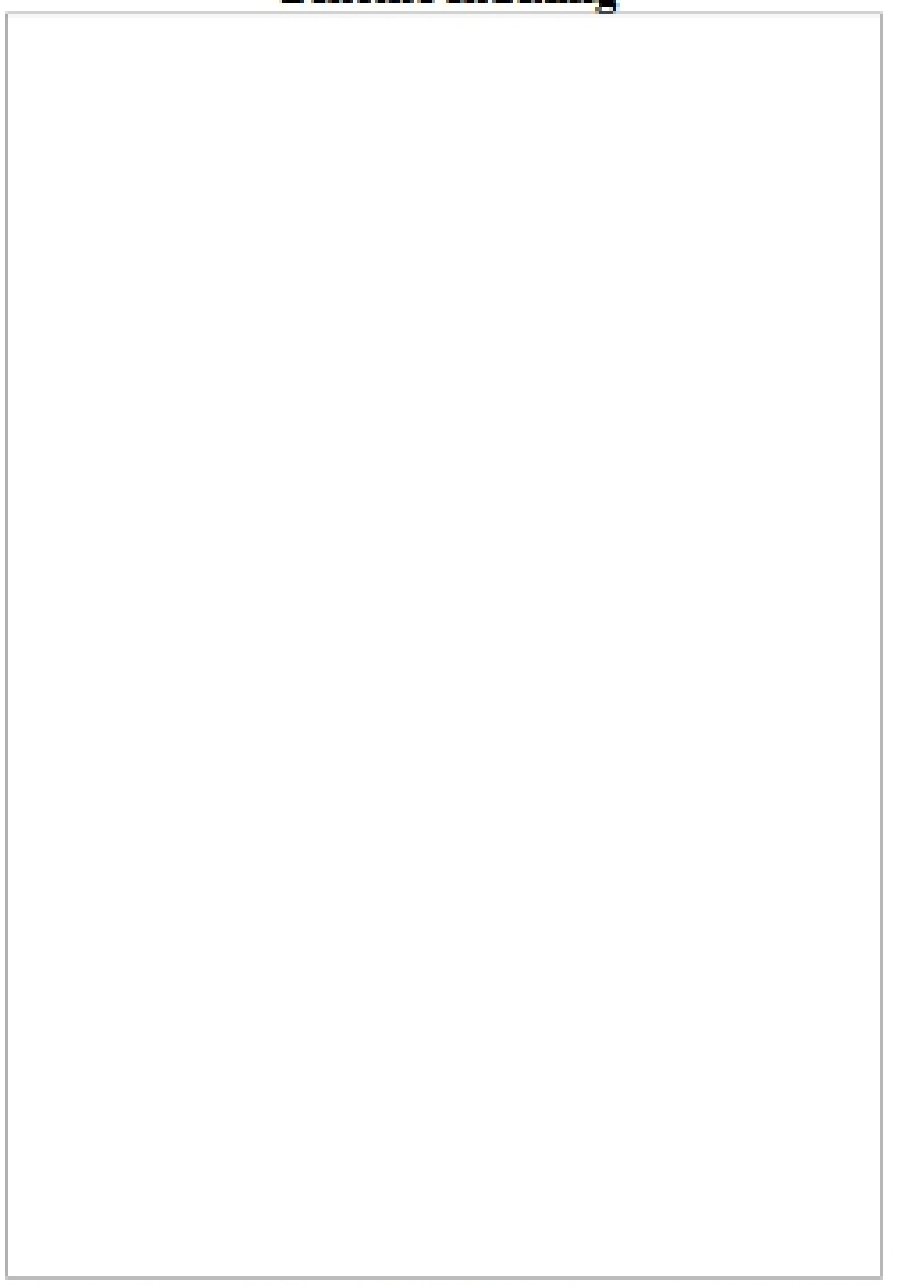

Image Copied from IEEE Spectrum July 2013

Preliminary Exercises:

1. Download the file called sequence_long.txt from Blackboard and put it in your current directory.

2. Run this command. You should get some positive integer back for the variable fid (variable name short for file identifier). If you get -1 back, then the file failed to open so it either isn't in your directory or the name of the file was somehow changed.

```
>> fid = fopen('sequence_long.txt','r')
```

3. Run the next three commands.

```
% Read the text file sequence into MATLAB one character at a time:
>> A = textscan(fid,'%1s');

>> DNA = A{1};  %Creates a cell array of nucleotides.

>> fclose(fid);  %Closes the text file
```

4. Look in your workspace. Record the size of DNA?

5. Run this command to produce the first 15 entries in DNA

   ```
   >> DNA(1:15)
   ```

   Results?

6. Run the following commands and record (and understand) the results

   ```
   >> DNA(1)

   >> strcmp(DNA(1),'C')

   >> strcmp(DNA(1),'A')
   ```

Remember strcmp is the equivalent and replacement of = = for strings. Only use = = for numbers, use strcmp for strings.

Lab Assignment:

Write a script file that will do the following:

- Import the text file sequence_long.txt into MATLAB using the commands in steps 2 and 3 of the preliminary exercises.

- Loop through the DNA array and record all of the locations of the triplets (codons): 'AAA', 'ATC' and 'CGG'. You will need a separate location vector for each triplet in order to store all of the locations for that particular triplet. **Use a while loop for this. If you find a codon then jump ahead in the DNA sequence past this codon. For example:**

  'A' 'T' 'C' 'G' 'G' 'A' 'G'

  ATC is here so don't count the CGG because the C is already part of ATC

- Use an fprintf statement to print out the total number of 'AAA' codons found.

- Use an fprintf statement to print the first 10 locations (offsets) for the 'AAA' codon

- Use an fprintf statement to print out the total number of 'ATC' codons found.

- Use an fprintf statement to print the first 10 locations (offsets) for the 'ATC' codon