**Lab 13: DNA Pattern Matching**

**A. Introduction**

The human genome project, a project to map all of the human genes (about 24,000) started in October 1990 and was completed in April 2003 ([**http://www.genome.gov/10001772**](http://www.genome.gov/10001772)). There are four DNA nucleotides: adenine (A), thymine (T), cytosine (C), and guanine (G).  A codon is a sequence of three nucleotides. These codons allow the body to code for the 20 nucleic acids that are used to build all of the proteins in the body.  One specific codon, TAC, indicates the start of a gene. Any of the three codons, ACT, ATT, and ATC, can indicate the end of a gene.

**B. Reading and Writing to Files**

MATLAB has the capability to import data from and export data to many different types of files including text files, excel files, and image files such as jpg or gif.

To read or write to a file, you must always open the file first. The generic MATLAB command for this is:

fid = fopen(‘filename.ext’,permission)

* Permission identifies what you are allowed to do with the file. A couple of possible choices are:

‘r’ read only from the file

‘w+’ open or create a file for reading or writing; clear any existing content

‘w’ open a file for writing; clear any existing content

* The variable fid (short for file identifier) is assigned a number corresponding to the specified file. Each time we wish to read from or write to the file, this file identifier must be used. If we are reading and writing to multiple files, we would have to use something like fid1, fid2, fid3, … in order to distinguish between files.
* If a file does not open properly, fid will be assigned 1. Whenever you open a file in a program, you should always check to see if fid is 1 and if it is, output an error message indicating that the file failed to open:

**if** fid == -1, error(‘File failed to open’); **end**

When you are finished writing to a file, you should always close the file using the command:

fclose(fid)

The following MATLAB commands will give you practice with reading and writing to files and also help you get started on the programs you will be writing for this lab

>> fid = fopen(‘test.txt’)

What is fid? What does this value for fid mean?

>> fid = fopen(‘test.txt’,’w+’)

What is fid now?

***Look in your Current Directory Folder. You should see a file now called test.txt. If you double click on it, you will see that it is empty.***

Type all of the following commands in MATLAB:

>> StartPos = 18; EndPos = 38;

>> fprintf(fid,‘Name: ***type your name here*** \n’);

>> fprintf(fid,‘Lab 13: DNA Pattern Matching\n \n’);

>> fprintf(fid,‘Start Position of Gene is: %d \t’,StartPos);

>> fprintf(fid, ‘End Position of Gene is: %d \n’,EndPos);

>> fclose(fid);

Now double click on the ***test.txt*** file in your current MATLAB directory and copy the contents of your text file here:

***Notes:***

1. ***Always close files after you are done reading from or writing to them!***
2. ***If you don’t use fid as the first argument of fprintf, it will simply print to the screen.***
3. ***Remember \n and \t are control characters. \n starts a newline and \t tabs over.***
4. ***To find out more about fopen you can type >>help fopen***

**C. Extremely Useful Exercises**

1. Download the text files titled ***sequence\_short.txt*** and ***sequence\_long.txt***. Save them in the folder where you have been saving all of your MATLAB programs for lab. Make sure the file names are: ***sequence\_short.txt*** and ***sequence\_long.txt***. Double click on the sequence\_short.txt file to open it and you should see a string of DNA nucleotides. Close the file.
2. Now try this sequence of commands. Copy and paste the cell array, codons, in the space below.

>> fid = fopen('sequence\_short.txt','r')

>> C = textscan(fid,'%3s');

>> codons = C{1}

>> fclose(fid)

codons =

1. What does the command: >> length(codons) tell you?
2. The function ***strcmp*** is very useful for comparing strings. Try the following commands and record the results in the space below. If you don’t understand what these commands are producing, talk to your T.A.

>> codons{1}

>> strcmp(codons{1},'TAC')

>> codons{7}

>> strcmp(codons{7},'TAC')

>> **for** pos = 1:8, **if** strcmp(codons{pos},'TAC') disp('Found a start'); **else** disp('Not a start'); **end**; **end**

**D. Finding Number of Genes and Start/End Positions of Genes in Genome File**

**Problem 1:** Write a function called short\_codon that will identify the start position and end position of the gene in the genome file ***sequence\_short.txt***. Your function should create a report called ***report\_short.txt*** somewhat similar to the one below.

Name: Your Name Here

Date: The Date Here

Lab 13: DNA Pattern Matching

Start Position: 7 End Position: 18

**Problem 2:** Write a function called long\_codon that will identify the start position and end position of all of the genes in the genome file ***sequence\_long.txt*** relative to the start of the file. The program should also count how many genes are found. ***This may require some significant modification of the previous program- read the notes below.*** Your function should create a report called ***report\_long.txt*** that includes your name, the date, lab title, ***the start and end position for every gene*** in the sequence and the ***total number of genes*** in the sequence.

*Notes:*

1. *Program 2 will likely require significant modification of your previous program because the short sequence simply had one start codon and one end codon. The long sequence has several thousand start and end codons but only a few of the starts and ends actually indicate the presence of a gene. Here is an illustration (green = start codon and red = end codon):*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

End of Gene 2

End of Gene 1

Start of Gene 2

Start of Gene 1

1. *Your program should begin searching for a start codon. Once a start codon is found, the program should do the following:*

* *Record the start position in the report\_long.txt file*
* *Switch tasks to start looking for an end codon (stop looking for a start)*
* *Continue to move through the codon array until an end codon is found*

1. *Once an end codon is found, the program should do the following:*

* *Record the end position in the report\_long.txt file*
* *Switch tasks to start looking for a start codon (stop looking for an end)*
* *Continue to move through the codon array until a start codon is found*

*.*

1. *This pattern should continue until the codon array has been completely scanned.*
2. *Defining a variable to help keep track of whether you are searching for a start codon or for an end codon would be useful. This variable is often referred to as a flag. It would take on one value while you are in “search for a start” mode and another value while you are in “search for an end” mode.*
3. *Each start-end combination indicates another gene. Remember to count the genes and record the start and end position of each gene in the genome in your report\_long.txt file.*