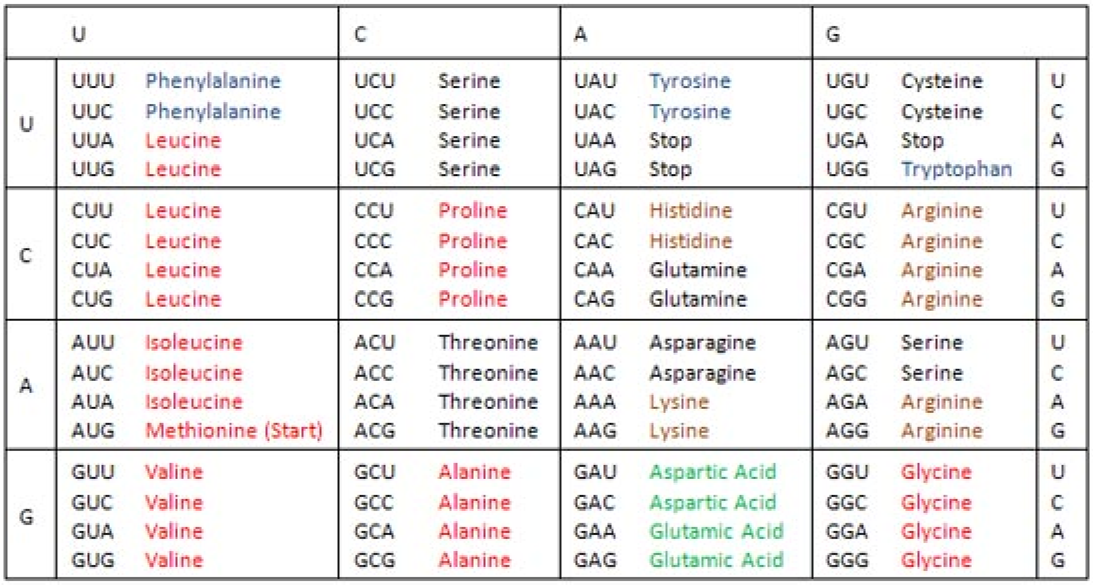
Lab 2: Analysis of DNA and mRNA for MC1R Gene

Background:

The genes in DNA encode the protein molecules that are necessary for specific cell functions. There are four nitrogenous bases in DNA: adenine (a), guanine (g), cytosine (c), and thymine (t). The first step in manufacturing the protein is to transfer the information in the double stranded DNA to a single stranded mRNA (messenger RNA).

mRNA also has four bases: adenine (a), guanine (g), cytosine (c), and uracil (u). The uracil (u) replaces the thymine (t) in the DNA. In mRNA, the bases are grouped together in triplets called codons. These codons specify the amino acids which are then chained together to manufacture a protein molecule.



	U		C		A		G		
U	UUU	Phenylalanine	UCU	Serine	UAU	Tyrosine	UGU	Cysteine	U
	UUC	Phenylalanine	UCC	Serine	UAC	Tyrosine	UGC	Cysteine	C
	UUA	Leucine	UCA	Serine	UAA	Stop	UGA	Stop	A
	UUG	Leucine	UCG	Serine	UAG	Stop	UGG	Tryptophan	G
C	CUU	Leucine	CCU	Proline	CAU	Histidine	CGU	Arginine	U
	CUC	Leucine	CCC	Proline	CAC	Histidine	CGC	Arginine	C
	CUA	Leucine	CCA	Proline	CAA	Glutamine	CGA	Arginine	A
	CUG	Leucine	CCG	Proline	CAG	Glutamine	CGG	Arginine	G
A	AUU	Isoleucine	ACU	Threonine	AAU	Asparagine	AGU	Serine	U
	AUC	Isoleucine	ACC	Threonine	AAC	Asparagine	AGC	Serine	C
	AUA	Isoleucine	ACA	Threonine	AAA	Lysine	AGA	Arginine	A
	AUG	Methionine (Start)	ACG	Threonine	AAG	Lysine	AGG	Arginine	G
G	GUU	Valine	GCU	Alanine	GAU	Aspartic Acid	GGU	Glycine	U
	GUC	Valine	GCC	Alanine	GAC	Aspartic Acid	GGC	Glycine	C
	GUA	Valine	GCA	Alanine	GAA	Glutamic Acid	GGA	Glycine	A
	GUG	Valine	GCG	Alanine	GAG	Glutamic Acid	GGG	Glycine	G

https://askabiologist.asu.edu/plosable/do-you-have-cavemans-brain

The following website has a wealth of information and great illustrations on the translation of DNA to mRNA to Protein.

http://www.nature.com/scitable/topicpage/translation-dna-to-mrna-to-protein-393#

The MC1R gene encodes the instructions for manufacturing a protein (melanocortin 1 receptor) which is then used by specialized cells to produce melanin, a pigment that gives skin, hair, and eyes their color. In lab today, we are going to analyze the reference gene sequence for the MC1R gene. The reference sequence was down-loaded from the National Center for Biotechnology Information: http://www.ncbi.nlm.nih.gov/ which provides access to biomedical and genomic information. All government funded genetics projects must upload their data to this site.

Part A: Counting the Nitrogenous Bases in a DNA Strand

1. Go to the metasite for Blackboard. Download the MC1R.mat file and the SampleCode.m file. Save both of these to your current folder in MATLAB.

2. For loops can be used to scan through a set of data and count the how many times something occurs. Open up the SampleCode.m file and look at the section of code labeled %% Part A. See if you can figure out what the code does. Then check by running the code in that section.

3. At the MATLAB command prompt, type the following commands:

```
>> clear; load MC1R;
```

4. You should now see two things in your Workspace Window: DNA (1x10099 cell array) and mRNA (3366x1 cell array). Double click on the DNA in your Workspace Window which will open it up in the Variable Editor Window. Don't edit the sequence – simply take a look at what is there: 10099 entries with one of the 4 nitrogenous bases: a, c, g, and t. Close the Variable Editor Window.

5. At the MATLAB command prompt, type the following command which will display the first seven entries in the DNA cell array: `>> DNA(1:7)`

6. Notice that the entries in DNA are strings. When dealing with strings, we do not use the relational operator: == but instead use the string compare function (strcmp). Try the following commands in MATLAB:

```
>> if strcmp(DNA(1),'g'), disp('Found a g'); else disp('Not a g'); end

>> if strcmp(DNA(4),'g'), disp('Found a g'); else disp('Not a g'); end
```

7. Open a new script file. Add a comment line with your name, your Professor's name, and your section number. Start a new section labeled: %% Part A: Counting Nitrogenous Bases

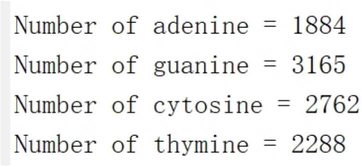
8. Add these two lines of code to your script:

```
clear all; close all;
load MC1R;
```

9. Now write a program that will count and display (fprintf) the total number of each nitrogenous base: adenine (a), guanine (g), cytosine (c), and thymine (t) in the MC1R DNA strand.

10. Run the section and check your results with your T.A.

PASTE RESULTS HERE:



```
Number of adenine = 1884
Number of guanine = 3165
Number of cytosine = 2762
Number of thymine = 2288
```

PASTE MATLAB CODE FOR PART A HERE:

```
%% Part A:Counting Nitrogenous Bases
clear; close all;
load MC1R.mat;
count_a = 0;
count_g = 0;
count_c = 0;
count_t = 0;
for i = 1:length(DNA)
    if strcmp(DNA(i),'a')
    count_a = count_a+1;
    elseif strcmp(DNA(i),'g')
    count_g = count_g+1;
    elseif strcmp(DNA(i),'c')
    count_c = count_c+1;
    elseif strcmp(DNA(i),'t')
    count_t = count_t+1;
    end
end
fprintf(['Number of adenine = %i\nNumber of guanine = %i\n'...
         'Number of cytosine = %i\nNumber of thymine = %i\n'],...
          count_a,count_g,count_c,count_t)
```

Part B: Locating and Counting Start and End Codons in the mRNA Strand

1. In your Workspace Window, double click on the mRNA cell array to open it up in the Variable Editor Window. Notice the bases are combined into triplets which are called codons. Most codons specify an amino acid as shown in the table on page 1 of this lab. However, there are special codons that start or stop the protein manufacturing process:

   Start Codon: AUG
   Stop Codon: UAA or UAG or UGA

2. Close the Variable Editor Window. For this part of the lab, you will need to record the locations of the start and stop codons in the mRNA strand. Open up the SampleCode.m file and take a look at the section labeled: %% Part B. See if you can figure out what the code is doing then run that section to check.

3. At the MATLAB command prompt, type the following commands:

   ```
   >> clear; load MC1R;
   ```

4. At the MATLAB command prompt, type the following command which will display the codons in entries 179 to 183 of the mRNA cell array.

   ```
   >> mRNA(179:183)
   ```

5. Again, the entries are strings which means we will use the string compare function (not ==). Try the following at the MATLAB command prompt:

```
>> if strcmp(mRNA(179),'aug'), disp('Found a start'); else
disp('Not a start'); end
```

6. Start a new section labeled: %% Part B: Locating and Counting Start and End Codons

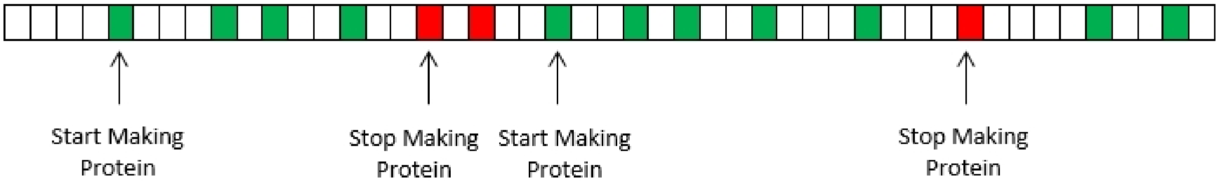
7. Add these two lines of code to your new section:

```
    clear all; close all;
    load MC1R;
```

8. Now write a program that will count and display (fprintf) the total number of start and the total number of stop codons and will also give the location of the start and stop codons in the mRNA strand. ***Note: don't count the 3 stop codons separately – an occurrence of either one of the three stop codons should increase the stop_count by 1.***

9. Run the section. At this point, your program should have found 41 start codons and 90 stop codons in the mRNA strand.

10. Your program now needs an adjustment because not all start codons actually start the protein manufacturing process and not all stop codons actually end the process. Look at the illustration below: The green squares indicate start codons and the red squares indicate stop codons.



As you can see from the illustration, a start codon starts the protein manufacturing process. Once a protein is started, the process continues until a stop codon is found in the mRNA – any extra start codons in the sequence simply produce the amino acid methionine. Similarly, once a protein is completed, the transfer RNA searches for the next start codon in the strand and ignores any extra stop codons in the sequence.

11. In your program, right before the for loop, create a variable called flag and set it equal to 0. You will use this variable to control when you are looking for a start codon and when you are looking for a stop codon. This variable, flag will only take on two possible values, 0 or 1.

flag = 0 means you are looking for a start codon
flag = 1 means you are looking for a stop codon

12. Using your flag variable, adjust your conditional statements so you are only looking for a start when flag is 0 and you are only looking for a stop when flag is 1. Also, adjust your code so that once you find a start codon, you then begin searching for a stop and once you find a stop codon, you go back to looking for a start codon.

13. When finished, run the section. If you have done this correctly, your program should have found 28 start codons and 28 stop codons.

PASTE RESULTS HERE:

```
Number of start codons for protein synthesis:   28

Number of stop codons for protein synthesis:  28

Start Codon Locations

129   179   203   343   389   783   803   911   968   1130   1181   1341   1500   1686
1760   2021   2053   2245   2417   2485   2488   2581   2639   2811   2835   2915   3022
3134

Stop Codon Locations

155   183   223   375   718   798   877   923   986   1156   1214   1352   1506   1727
1794   2038   2075   2249   2440   2487   2558   2582   2649   2812   2891   2940   3062
3216
```

PASTE MATLAB CODE FOR PART B HERE:

```
%% Part B Locating and Counting Start and End Codons
clear all; close all;
load MC1R;
start_count = 0;
stop_count = 0;
flag = 0;
for k = 1:length(mRNA)
    if strcmp(mRNA(k),'aug') && flag == 0
        start_count = start_count + 1;
        start_loc(start_count) = k;
        flag = 1;
    elseif
```

```
(strcmp(mRNA(k),'uga')||strcmp(mRNA(k),'uaa')||strcmp(mRNA(
k),'u ag'))&& flag == 1
        stop_count = stop_count + 1;
        stop_loc(stop_count) = k;
        flag = 0;
    end;
end;
fprintf('Number of start codons for protein synthesis:   %i
\n',start_count);
fprintf('Number of stop codons for protein synthesis:  %i
\n',stop_count);
fprintf('Start Codon Locations \n');
fprintf('%i \t',start_loc);
fprintf('\n');
fprintf('Stop Codon Locations \n');
fprintf('%i \t',stop_loc);
fprintf('\n');
```

What to Turn In:

- This word doc (or pdf) with your results and code pasted in where indicated
- Your script (.m) file