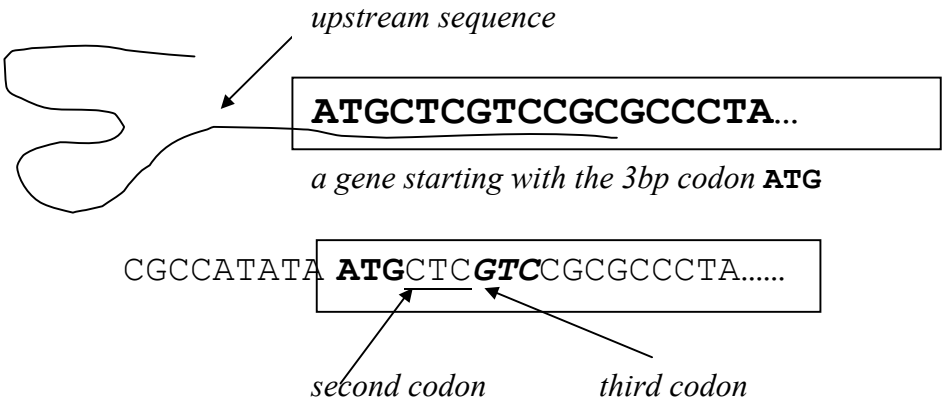
DNA

Playing with Strings

Write a Perl program to print a report about a specific DNA sequence:

cgccatataatgctcgtccgcgcccta

The sequence (starting in lowercase) is a "snip" of DNA that includes some upstream (intergenic) sequence and the beginning (but not all) of a gene (genic sequence). Since all genes begin with the three nucleotide code ATG, we can "easily" find the start of a gene. All the sequence to the left (upstream) of ATG is the upstream (intergenic – between the gene) sequence and the sequence to the right (and including) ATG is the genic (gene) sequence. Note: the entire gene sequence is not included in this small sequence.



Within a gene, every three nucleotides is called a codon. ATG is the initial codon in every gene, but other codons vary depending on the particular gene. In this example, CTC is the second codon and GTC is the third codon and so on.

Starting with a lowercase sequence, your program should print

- starting sequence, the same sequence in uppercase, as well as its length.

Then your task is to locate and report the position of the ATG codon within the starting sequence and then take substrings (substr) of the entire sequence to form two other strings that hold the upstream and genic sequences. Your program should then print

- the locations and three-character codes of the first three codons in the gene (your program can assume that ATG and the first and second codons *will* be there)
- upstream and genic sequences and their respective lengths (see a sample report below)
- the reverse complement of (only) the genic sequence.

Note: your program should *not* only work on this sequence but on *any* sequence that contains ATG, therefore, use variables (e.g., $positionATG) rather than 9. For example, change your starting sequence to see if your program works on a different sequence.

Finally (once you have all of the above working), print

- the original sequence with only the ATG start codon in capital letters (uppercase) while the rest of the sequence is in lowercase.

Also, to try and get a feel for whether the upstream region is AT-rich, print

- a count of the number of As and Ts in the upstream region.

A partial **sample output** is shown below. Your program's output need not be exactly identical but you should print out the information in the same order and obviously your answers should agree. Note: biologists start sequence at one (1bp), but Perl numbers all strings (DNA or otherwise) at character zero (0). You should always format your output according to what your boss wants. In this course, we will assume that your boss is a biologist, thus all sequences start at base pair one (1).

```
++++++++++++ Upstream and Genic Report ++++++++++++++++++++++++

Starting sequence is:    cgccatataatgctcgtccgcgcccta
Converted to uppercase:  CGCCATATAATGCTCGTCCGCGCCCTA

Length of starting sequence is: 27
----------------------------------------------------------------

ATG start codon begins in position (bp) 10
    followed by codon CTC in position (bp) 13
    followed by codon GTC in position (bp) 16
----------------------------------------------------------------

Upstream sequence is:  CGCCATATA

Upstream length (bp): 9

----------------------------------------------------------------

Gene sequence is:  ATGCTCGTCCGCGCCCTA

Gene length (bp): 18

----------------------------------------------------------------

Gene + Strand:  ATGCTCGTCCGCGCCCTA

Gene - Strand:  TAGGGCGCGGACGAGCAT
           :
           :
           :
      you complete the last steps …
----------------------------------------------------------------
```

This program is **due on Thursday September 17** (specifically, dropped to onCourse by 5am on Friday, Sept 18). You must submit your Perl program via onCourse (moodle). Also submit a landscaped printout of your program in class on FRI Sept 18. No late submissions are accepted. (Read that last sentence again).

Submit:
(1) one hardcopy of your Perl program (use a good filename, e.g., **`LeBlanc_a1.pl`**)
(2) one hardcopy page that shows the **OUTPUT** of your program.
(3) one hardcopy of your README documention (.pdf file created by pod2pdf).
(4) **You** *must* staple your Perl (`LeBlanc_a1.pl`), README, and your output together.

Here is some help getting started ... (**Download the a1 Starter Kit** from the onCourse site)

```
#!/usr/bin/perl

use strict;
use warnings;
```

Every program must have some "**pod**" documentation at the top

```
=pod
=head1 NAME

my_a1.pl

=head1 DESCRIPTION

=head2 SUMMARY

This Perl program splits a sequence of DNA into the upstream and genic
regions.It then performs a number of string manipulations on both those
sequences. A report (see Output below) is printed.

=head2 INPUT

 No input. A string of DNA is manually set in a variable.

=head2 OUTPUT

To CONSOLE (screen):
A report of the first three codons and their bp locations, the upstream
and genic regions and their lengths, the reverse complement of the genic
portion, and the number of As and Ts in the upstream region.

=head1 AUTHORS

your name goes here


=head1 MODIFICATION HISTORY

=head3 August 4, 2009 (mdl) --

  back from fishin', just getting started

=head3 August 11, 2009 (mdl) --

  finished the starter kit

=head3 (add your modification dates and notes here)

=cut
```

```
# ----- code starts here ----------------

print "+++++++++++++ Upstream and Genic Report +++++++++++++++++\n\n";
```

 # continues on next page …

```
print "+++++++++ Upstream and Genic Report +++++++++++++++\n\n";

my $someSequence;       # upstream and start of a gene ...

$someSequence     = "cgccatataatgctcgtccgcgcccta";

print "Starting sequence is:    $someSequence \n";

# you convert all nucleotides to uppercase here


print "Converted to uppercase:  $someSequence \n\n";

# you determine the length of this sequence here

print "Length of starting sequence is: $seqLength \n";

print "-----------------------------------------------------\n\n";


# get the position of the start codon "ATG"
my $ATGPosition;

$ATGPosition = index( ……
:

print "-----------------------------------------------------\n\n";

my $upStream;
$upStream = substr( ……

print "Upstream sequence is:  $upStream \n\n";

      :
      :
#     you continue the program
```