Hannah Machado

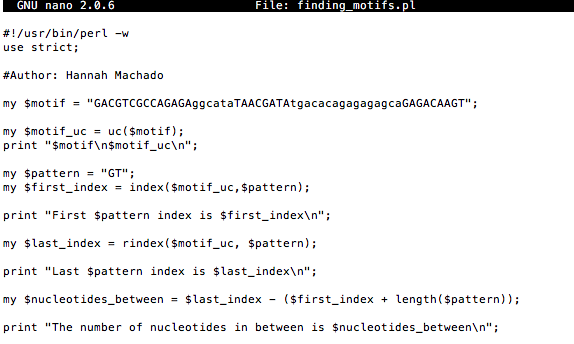
CSC 210

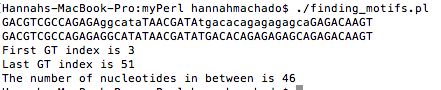
Apr. 20th, 2017

Homework 11

**1. Finding motifs:** Create a Perl program that computes and prints the number of nucleotides that separate the first and last appearance of the motif AGAG in lower case, upper case, or combination, in the DNA sequence

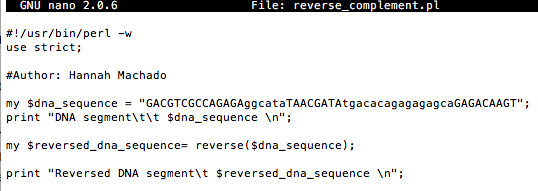
GACGTCGCCAGAGAggcataTAACGATAtgacacagagagagcaGAGACAAGT





**2. Reverse complement:** Create a Perl program that computes the reverse complement of the DNA sequence below and prints the sequence and its reverse complement:

GACGTCGCCAGAGAggcataTAACGATAtgacacagagagagcaGAGACAAGT

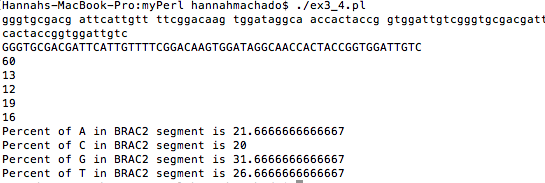
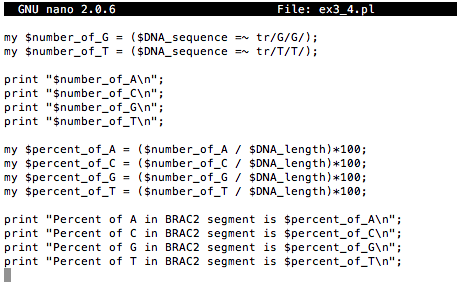


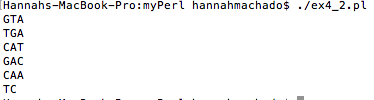
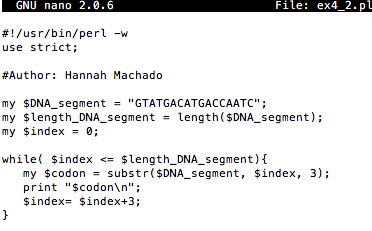


3. **Counting base pairs with tr:** The tr operator returns the number of characters that were changed (transliterated). For example, the Perl statement

$number\_of\_C = ( $DNA\_sequence =~ tr/C/C/ );

captures the numbers of Cs in the string $DNA\_sequence and and records it in the variable $number\_of\_C. Now, write a Perl code that prints out the percentages of the four nucleotides A, C, G, T in the following segment of the breast cancer gene [BRAC2](http://www.ncbi.nlm.nih.gov/nuccore/224004157?report=genbank):  
gggtgcgacg attcattgtt ttcggacaag tggataggca accactaccg gtggattgtc



4. **Codons in array:** Create a Perl program that stores a DNA sequence of any length, extracts all codons in the sequence, and stores them in an array. Print the sequence and the codons, one per line, comprising it. Your program should handle the case where the sequence length is not a multiple of 3. Test your program with several sequences.

5. **Generating random DNA sequences**: Generate a random DNA string 101 bases long, store it in a string variable, and print it. To do that, use an array of nucleotides, and the [rand](http://perldoc.perl.org/functions/rand.html) Perl function, to generate random numbers. Other tools that you can use include: string concatenation, the [int](http://perldoc.perl.org/functions/int.html) Perl function, and a loop.

Run your program several times. Do you get the same random sequence? Unlikely! For certain simulations, it may be necessary to use the same random sequence. Consult the documentation on how to set the seed of the [srand](http://perldoc.perl.org/functions/srand.html) Perl function.