Motif Finder

Finding Relevant Promoter Motifs "Upstream" of Genes

In your last Perl program, you built a simple gene finder. This programming assignment assumes that you have already located a specific gene but now you want to investigate the DNA sequences "upstream" (just prior to or to the left of) that gene, as shown below:

First a few helpful facts about genes and their related "promoter" regions:

1. Functional genes have sequences upstream ("to the left of") the start DNA triplet "ATG" (or "AUG" in the RNA). The upstream area is called the "promoter". The function of the promoter region is to help direct the cell to either transcribe the gene or not. This is a very important decision for multi-celled organisms like humans. For example, since each cell has all of the DNA, the cells in our elbows have the same genes as the cells in our eyes. The reason the eyes are not erupting from our elbows is because promoter information is dictating, "Do not transcribe this eye gene in the elbow!" In this programming assignment, you will be designing and implementing a **promoter motif searcher**. Promoter motifs often are repetitive DNA sequences upstream of genes. You will focus on certain categories of repetitions.

2. Remember, genes "code for" ("are the instructions to build") proteins. Proteins are used throughout the cell (and body) to run chemical reactions and form structures.

3. Upstream of (in the promoter region of) many genes are simple well-known motifs that are needed to begin any transcription, no matter what the cell. These motifs do not specify elbow *vs*. eye (for example) but are more general in use. A general motif that you will seek is: the 8-mer "TATA box" which is defined as some permutation of:

```
TATA(A or T)A(A or T)(A or G)
```

OUTPUT

A sample output is shown on the last two pages and is described here. Your program will start with a DNA sequence as returned from the getDNA() subroutine. See the sample input files `test1.fna` and `test2.fna` in the starter kit. We will assume this DNA is a potential promoter region upstream of a gene (Note: this does not include the gene, only the upstream region of a gene). Your job is to write a program to search this upstream region and produce a report. Of course, your program should work on *any* sequence, so take care to write your program to handle the *general* case. For example, do not use *magic numbers* in your program (such as look in location 53) but rather require your program to compute and save the lengths and locations of strings and substrings in variables and then use those variables in your calculations. In short, use lots of variables and think carefully about when to prevent division by zero by using `if-else` statements.

ALGORITHM

(1) Determine if the starting DNA upstream sequence contains a TATA-box. If it does not, report that and end the program. Otherwise, continue with the remainder of this report.

(2) Assuming you have found a valid 8-mer TATA-box, report the actual TATA-box motif and its location.

(3) Clip out the substring to the left of (upstream) of the TATA-box. Print that substring, its length, and the percentage of this substring's length as compared to the entire sequence. Careful! It *is* possible that the TATA-box appears right at the beginning of this sequence, thus there is no upstream region. Or it *is* possible that the TATA-box appears at the very end of this sequence, thus there is no downstream region. You must insert `if-else` statements to handle these situations in an appropriate way. For example if there is no upstream region, you would not do step #4 for the upstream region.

(4) In the substring to the left of the TATA-box (assuming there is an upstream region),
   (a) report *all* Direct Repeats (DR) motifs found with a total length of 4 to 12 bp and their location within the substring (e.g., GCGC, AAATTCAAATTC, etc).
   (b) sum the lengths of each DR found and report the percentage of the sum of all DRs as compared to the entire substring to the left of the TATA-box.
   (c) report *all* Mirror Repeats (MR) motifs found of total length 6 bp and their location within the substring (e.g., ACGGCA)
   (d) sum the lengths of each MR found and report the percentage of the sum of all MRs as compared to the entire substring to the left of the TATA-box.
   (e) report *all* occurrences of *your favorite motif* that are found and their location within the substring. (Note in the sample output our favorite motif was GC-rich sequences of length [4-7]). You cannot use this as your favorite motif. Be creative!
   (f) sum the lengths of each of your favorite motifs found and report the percentage of the sum of all your favorite motifs as compared to the entire substring to the left of the TATA-box.

HINT: When using regular expressions with the match (`m`) operator, surround your entire regex within parentheses. This will allow you to "capture" the result of your match in an efficient way. Note that this extra set of parentheses (the entire match) will be "remembered" in $1, thus other sets of parentheses *within* your regex (such as you'll need for repeats) must start at two (`\2`), e.g., `\4\3\2` for the mirror repeat regex that you'll need, as shown below:

```
print "Searching for Mirror Repeats (MR) of length 6 bp \n\n";

$sum = 0;
while ( $upstream =~ m/((.)(.)(.)\4\3\2)/g )
{
    $answer = $1;  # $1 holds the entire substring found

     # since pos holds the location just after the last match
     # do the math to indicate the starting location
    $DRloc  = (pos $upstream) - length($answer);
        :
}  # while still more upstream to search
```

DOCUMENTATION

Your program should be well documented. Here is a reminder of some of the things that a professional program includes as part of the documentation as well as some specific instructions for you to follow when documenting this program.

_____ Your **Name** found *in* source code.
_____ At least some attempt to document your progress in "**date last modified**" section.
_____ **Status** message given in documentation, e.g., "This program works!" or "Errors in DR section"

_____ Submission materials **stapled**, including source code and a sample of *your* program's output.
_____ Source code printed in **landscape** mode.
_____ Sample **output** included.

_____ All **variables** declared together at the **top** of the program.
_____ All variables have associated **# comments** explaining their meaning and use.

_____ Use lines to break your code into sections, for example
#------------------------- Searching for DRs to left of TATA-box ----------------------------

_____ Write comments *within* your program to explain to the reader what your code is doing at that point.

In addition, include a section of documentation at the top of your program where you show and explain to the reader the **regular expressions** and associated explanations that your program uses to find:

TATA-box, Direct Repeats, Mirror Repeats, and your favorite motif

For example:

```
# ================================================================
# REGEX patterns used:
#
#    TATA box:            …
#
#    Direct repeats:      …
#
#    Mirror repeats:      total length 6 bp where the first three
#                         nucleotides are "mirrored" by the last
#                         three nucleotides, e.g.:   ACGGCA
#
#                         regex used:  ((.)(.)(.)\4\3\2)
#
#    My favorite repeat: …
#
# ================================================================
```

SAMPLE OUTPUT (using **test1.fna**)

```
Successfully opened and read the file:
/Users/mleblanc/Desktop/_DNA242_Fall2007/Programs/a4_motifFinder/test1.fna
>test1.fna

We are assuming we are searching the upstream region of a certain gene.

Region (69 bp) being searched is:
GCGGCGACCACCTTATATGGTTCAGCAGGCCAAGCCTTATAAAAAGGCGGGCGGCTTCGCGAGGACTTG

================================================================
Found a TATA-box:  TATAAAAA
at location 37

---------------------------------------------------------
 UPSTREAM of TATA-box
---------------------------------------------------------
Region upstream of the TATA-box:
GCGGCGACCACCTTATATGGTTCAGCAGGCCAAGCCT
is of length 37 (bp) or 53% of the entire upstream region.

---------------------------------------------------------
Searching for Direct Repeats (DR)

Found another DR: GCGGCG
      at upstream location: 0
Found another DR: ACCACC
      at upstream location: 6
Found another DR: TATA
      at upstream location: 13
Found another DR: CAGCAG
      at upstream location: 22

Percent of upstream region that is comprised of DRs is: 59%
---------------------------------------------------------
Searching for Mirror Repeats (MR)

Found another MR: GCGGCG
      at upstream location: 0

Percent of upstream region that is comprised of MRs is: 16%
---------------------------------------------------------
---------------------------------------------------------
Searching for runs of GC-rich regions of length [4-7]

Found another GC-rich region of length {4-7}: GCGGCG
      at upstream location: 0
Found another GC-rich region of length {4-7}: GGCC
      at upstream location: 27

Percent of GC-rich region is: 27%
---------------------------------------------------------
================================================================
```