Harvard-MIT Division of Health Sciences and Technology
HST.508: Genomics and Computational Biology

Problem Set 2

Please make sure to show your work and calculations and state any assumptions you make in answering the following questions. Include the names of the people you worked with at the top of your problem set.

Problem 1: Genome sizes and data storage (35 points total)

The NIH 's National Center for Biotechnology Information (NCBI) provides a huge repository and a multitude of databases for biological information. NCBI Entrez's Genome page (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Genome)is a good starting place for resources on genome projects. Many biology textbooks also commonly discuss genomic size and its biological basis.

1 (a) Find the approximate size of the West Nile viral genome, the microbial *Escherichia coli* K12 genome, the *Caenorhabditis elegans* haploid genome, the haploid human genome and the *Amoeba dubia* genome in base pairs. (1 pt each, total of 5)

1(b) Find out the estimated total number of genes in each of the above organisms. (1 pt each, total of 5)
Is the size of genome proportional to the total number of genes? Give at least one reason why this is or is not the case.(4 points)
Is it always true that the more complex the organism, the large genome it has? Give an example if your answer is no and explain why.(4 points)

1(c) What is the minimum number of bytes required to store the genomes listed above? To store the human genome in its diploid rather than haploid form? Show your calculations! (6 points)

1(d) What is the minimum number of bytes needed to store all human genomes? All such genomes can be represented as a single individual's genome plus the variations, or polymorphisms, seen in all other human genomes. Assume that the human population is ~ 6 billion, which was the population reached in October 1999, and that polymorphic sites tend to be simple single nucleotide polymorphisms (SNPs) such as "A" in one genome and "C" in another) and occur about once every 3 kb (4pts).

1(e) How many double-sided DVDs would it take to store the genomes listed above given your bit conversions above? How many 80GB hard disks would it take to store all human genomes in the world, again given your calculations above (4pts)?

1(f) Some nucleotide sequence data have to be stored at more than 2 bits/base. Could you think of a reason why this would be the case? (3 points)

Problem 2: Sequence occurrences (30 points total)

2 (a) At how many *sites* would you expect "CG" to occur in 4.6 Mbp (mega bp) in a *double-stranded* genome? How about "CTAG"? And "GATTACA"? Assume all nucleotides have an equal probability of occurring. (2 pts for each, total of 6 pts)

> Hint: "CG" is a palindromic sequence. When it occurs on one strand, it also occurs on the complement.
>
> 5'-*CG*-3'
> 3'-*GC*-5'
>
> At this site, you find 2 occurrences of "CG." (For palindromic sequences, you find 2 occurrences of the sequence at 1 site.)
>
> "GATTACA" is not palindromic.
>
> 5'- GATTACA -3'
> 3'- CTAATGT -5'
>
> For non-palindromic sequences, you find 1 occurrence of the sequence at each site.

2 (b) Run the following perl program, parse.pl, with the ~4.6 Mbp* E. coli K12 *genomic sequence as input. (Obtain the genomic sequence file, "E.coli_K12.txt," from the following link: *ftp://ftp.ncbi.nih.gov/genbank/genomes/Bacteria/Escherichia_coli_K12/U00096.fna* ***)***

```
#!/usr/bin/perl -w
$fname = "E_Coli_K12.txt";
open(FH, $fname);

$text =  SeqFromFile($fname);
close FH;

$string = $temp = "ctag";
$match = $text=~ s/$string//gi;
$a = $match * ($temp =~ s/a//gi) + ($text =~ s/a//gi);
$c = $match * ($temp =~ s/c//gi) + ($text =~ s/c//gi);
$g = $match * ($temp =~ s/g//gi) + ($text =~ s/g//gi);
$t = $match * ($temp =~ s/t//gi) + ($text =~ s/t//gi);
$n = $match * ($temp =~ s/n//gi) + ($text =~ s/n//gi);
print "$string: $match\na: $a\nc: $c\ng: $g\nt: $t\nn: $n\n";


#Converts sequence files to a string. Assumes the first line is
#the header, which it removes, and it removes linebreaks.
sub SeqFromFile {
  my $fname= $_[0];
  open(FH, $fname);
  local $/;
  my $data = <FH>;
  $data =~ s/\>.+?\n//;
  $data =~ s/\n//g;
  return $data;
}
```

Step by step instructions for students with fas accounts:

1. Send the *E. coli K12* genome sequence to your fas account and save it in the proper directory as “E_Coli_K12.txt”.
2. 2. Login to fas.harvard.edu.
3. 3. Type "pico" at the command line. This brings up a text editor called pico.
4. 4. Paste or type the program (above) into the pico window.
5. 5. Press “Ctrl-x” to exit pico. Type “y.” Type “parse.pl” to name the program. Press “return.”

2(b) What is the program output? (1 pt)

2(c) What is the ratio of the observed incidence of CTAG in the E. coli K12 genome to the expected value? (1pt)

2(d) In part a, you should have assumed that each nucleotide has an equal probability of occurring, but you know from part c that this is not actuality the case. Based on the output from part c, re-calculate the expected incidence of CTAG in the E. coli K12 genome. (2 pts)

2(e) Re-calculate the ratio of observed to expected values for CTAG. You can exclude N's from the length of the sequence. (2 pts)

2(f) Why might the observed incidence be so different from the expected? Speculate what this may mean in a biological sense. (2pts)

2(g) This version of the E. coli K12 genomic sequence did not contain any N's (Note that an N represents any nucleotide at that particular position). If we were analyzing a sequence with N's, could we simply remove them from the sequence at the beginning of the program ($text = <>; $text =~s/n//ig;) ? (2pts)

2(h) Explain what the program does. Use any of the recommended Perl resources or texts to explain what each line does (12 pts).

2(i) Modify the program, parse.pl, so that it counts the number of times the string "TCAGGACT" occurs in the E. coli K12 genome. Find occurrences on both the sense and the antisense strands. Show your code and the output. (2 pts)

Problem 3: Sequence Alignments (35 points total)

3(a) Briefly describe the differences between global and local alignment and between pairwise and multiple sequence alignment. Bioinformatics (Mount, 2001) covers these in detail. (4pts)

3(b)(i) Compare BLAST to the Smith-Waterman algorithm. What are the advantages and disadvantages of BLAST? (3pts).

ii. There are two major implementations of the BLAST algorithm initially developed by Altschul et al (J. Mol. Biol, 1998) , NCBI BLAST and WU-BLAST (Washington University). While WU-BLAST is commonly used for searching genome sequences, NCBI BLAST is the more widely used of the two. Here you will become familiar with NCBI BLAST .

Perform a standard nucleotide BLAST search (http://www.ncbi.nlm.nih.gov/BLAST/) with the following sequence, using the default settings. Describe the output (1pts) and explain the meaning of the associated measures associated with the output alignments (score and E-value) (2pts). What gene do you think this sequence is from and why (2pts)? What possible homologs are there in other species and why (2pts)? (7pts total)

```
>Unknown sequence
gagtgcttgg gttgtggtga aacattggaa gagagaatgt gaagcagcca ttcttttcct
gctccacagg aagccgagct gtctcagaca ctggcatggt gttgggggag ggggttcctt
ctctgcaggc ccaggtgacc cagggttgga agtgtctcat gctggatccc cacttttcct
cttgcagcag ccagactgcc ttccgggtca ctgccatgga ggagccgcag tcagatccta
gcgtcgagcc ccctctgagt caggaaacat tttcagacct atggaaactg tgagtggatc
cattggaagg gcaggcccac caccccgacc ccaaccccag ccccctagca gagacctgtg
ggaagcgaaa attccatggg actgactttc tgctcttgtc tttcagactt cctgaaaaca
acgttctggt aaggacaagg gttgggctgg ggacctggag ggctgggggg ctggggggct
gaggacctgg tcctctgact gctcttttca cccatctaca gtcccccttg ccgtcccaag
caatggatga tttgatgctg tccccggacg atattgaaca atggttcact gaagacccag
gtccagatga agctcccaga atgccagagg ctgctccccg cgtggcccct gcaccagcag
ctcctacacc ggcggcccct gcaccagccc cctcctggcc cctgtcatct tctgtccctt
cccagaaaac ctaccagggc agctacggtt tccgtctggg cttcttgcat tctgggacag
ccaagtctgt gacttgcacg gtcagttgcc ctgaggggct ggcttccatg agacttcaat
gcctggccgt atccccctgc atttcttttg tttggaactt tgggattcct cttcaccctt
tggcttcctg tcagtgtttt tttatagttt acccacttaa tgtgtgatct ctgactcctg
tcccaaagtt gaatattccc cccttgaatt tgggctttta tccatcccat cacaccctca
gcatctctcc tggggatgca gaacttttct ttttcttcat ccacgtgtat tccttggctt
ttgaaaataa gctcctgacc aggcttggtg gctcacacct gcaatcccag cactctcaaa
gaggccaagg caggcagatc acctgagccc aggagttcaa gaccagcctg ggtaacatga
tgaaacctcg tctctacaaa aaaatacaaa aaattagcca ggcatggtgg tgcacaccta
tagtcccagc cactcaggag gctgaggtgg gaagatcact tgaggccagg agatggaggc
tgcagtgagc tgtgatcaca ccactgtgct ccagcctgag tgacagagca agaccctatc
```

iii. Repeat the search as a translated nucleotide query of all protein databases (BLASTX). What differences do you observe in the output and why (2pts)? Which organisms have possible homologs to the above sequence given this search (1pt)?

Which search would you use, the standard nucleotide BLAST or the translated BLAST, if searching for possible homology in future searches (1pt)? (4pts total)

iv. Repeat the search in iii using the PAM30 matrix rather than BLOSUM62. Describe any differences in output that you observe. *(3 points)* In BLASTX, you have 5 matrices to choose from: PAM30, PAM70, BLOSUM45, BLOSUM62 and BLOSUM80. If you want to find more divergent sequences, which two should you use and why?*(3 points)*

(c) Global alignment with Needlman-Wunsch (6pts total).

Back in 1969, S. Needleman and C. Wunsch came up with an efficient method of obtaining an optimal global alignment (Needleman, S. B., Wunsch, C. D., *J. Mol. Biol.* (1970) 48:443-453). Although not known to the authors, the proposed algorithm followed the principle of dynamic programming introduced some 12 years earlier by Richard Bellman, and later popular alignment algorithms (such as Smith-Waterman local alignment) were based on this Needleman-Wunsch method. The Needleman-Wunsch algorithm can be written as follows:

$$S_{ij} = \max \{ S_{i-1,\, j-1} + s(a_i b_j),\ \max(S_{i-x,\, j} - w_x),\ \max(S_{i,\, j-y} - w_y) \}$$

or, to simplify:

$$S_{ij} = \max \{ S_{i-1,\, j-1} + w(\text{match, mismatch}),\ S_{i-1,\, j} + w(\text{gap}),\ S_{i,\, j-1} + w(\text{gap}) \}$$

Although the Needleman-Wunsch algorithm was initially used to align amino acid sequences, the algorithm can be applied to strings of an arbitrary alphabet. For the purposes of this question, we will consider only DNA sequences. In *E. coli* promoter sequences, the -35 signal TTGACAT is well-known to have functional significance. *Align the -35 signal TTGACAT to the sequence GTTGTACTT* using the Needleman-Wunch algorithm with the following weight function for the DNA sequence alignments in the problems below:

w(match) =2
w(mismatch) =-1
w(gap) =-3

Show the optimal global alignment(s) as well as the matrix that scores all possible alignments.

(d) How would you (or Smith and Waterman) modify the Needleman-Wunsch algorithm to yield optimal local rather than global alignments (2pts)? Perform your alignment again with the local alignment algorithm. Show the optimal local alignment as well as the matrix that scores all possible alignments. (3 points)