CSCI1820 HW4

Due: March 4 11:59pm

Choose to complete either the Biology **or** Computational problem 1. Biologists do not need to complete the De Bruijn graph problem.

This homework is scored out of 100 (or more with extra-credit). For all problems, you will only receive full credit if you document how you obtained your solution; in most cases, commented code is the best way to do this. In some cases a brief description is adequate. The other common cause of points subtracted is failing to show the data or mathematics which justify your statements.

Problem 0 Mimivirus (25)

Unintelligent Design, Charles Siebert.
`http://discovermagazine.com/2006/mar/unintelligent-design`

To provide some background for this problem, please read the article on "Unintelligent Design" located online.

We would like you to support some of the claims presented in the article using bioinformatics tools. More specifically, we would like you to show that the Mimivirus is related to each of:

1. the Avian Bird Flu virus (Influenzavirus A)
2. Ebola virus (ebolavirus)
3. a bacteria of your choosing
4. a more complex cellular organism, possibly animal

The entire genome of the Mimivirus is conveniently located in the BLAST database. (You can find it by searching for "Mimivirus" under the category "Genomes" on NCBI's main page.) The Influenzavirus A genome and the ebolavirus genome are also in the BLAST database. We would like you to choose amino acid and/or protein sequences from the Mimivirus genome that you can use to show that the Mimivirus is related to both other viruses and bacteria. Feel free to use alignment code, dot plots, BLAST, and NCBI Genome/Nucleotide/Proteins/Genes/PubMed databases, etc. You will be graded on the strength of your argument – we will approach it with skepticism! Your argument should include at least statistics, similarity, and biology as bases for plausibility.

Problem 1 BIOLOGY Phase Transitions (25)

In this problem we will investigate the importance of selecting adequate alignment parameters to produce a biologically meaningful result. We've uploaded the Mathematica notebook AlignmentStats.nb to the homework page. Familiarize yourself with the code. Adjust the parameters for the probability of A, C, G, T using the sliders generated from the Manipulate Mathematica function and **describe how varying the probabilities of nucleotides affect the phase transition from a logarithmic-length local alignment to a linear-length local alignment**. Use as much or as little of the existing functionality as you'd like. Please provide support for your claims using Mathematica.

Also, for a particular set of parameters, investigate how the alignments look around the phase transition. **Please record at least one set of alignments representing the three phases (logarithmic local alignment, transitional phase, and linear local alignment). How do the parameters affect the alignments?** We recommend using the DynamicStringAlignment.nb Mathematica notebook included on the homework page of the website to visual alignments.

You may also want to experiment with AlignmentStats.nb. For this, you will have to adjust the sliders for the probabilities of each nucleotide in the Manipulate function and also in the alignstats function: Adjust numberOfAlignments which is the number of sequences to generate. Be careful because if this value is x then this function will evaluate $\binom{x}{2}$ alignments. The lengthOfSequence variable is the length of the sequences being aligned and has less of an impact on the runtime. After evaluating all of these cells, there is code that will plot important relationships relating the local and global alignment scores with similarity matrix parameters.

Problem 1 COMPUTATIONAL Probability and RNAseq (25)

The bacterial organism $\sum K\Omega 9$ has a 262,144bp length genome. (Only in this class would an organism have a power of 2 as genomic size). You've been given the task of identifying the coordinates of the genes of this organism. To infer the presence of a gene, we need to identify an mRNA transcript for the genomic sequence. However, in practice this is done using RNAseq. mRNA from cell extract are cut into many fragments (assume randomly), then one end of the fragments is sequenced up to a read length r. For this problem, $r = 8$.

Your task is this: we give you the genome of $\sum K\Omega 9$. Then we give you 2000 8-mer reads of mRNA transcripts. Assume that the genomic sequence is the template strand, thus the mRNA is in the same orientation as the genome, only with "U" substituted for "T". Output your inferred coordinates of the genes in the genome, given that this bacteria is intron-less. (Note: There is no "correct" answer as you might not have the read fragment for the ends. In fact, since the same end of the mRNA is sequenced each time, it's almost impossible to infer one of the termini!)

Document all of your methods, and provide code. For each gene you identify, give a statistical argument for that prediction. This may consist of one overarching paragraph, then an equation for each gene. Keep in mind that the mRNA reads will randomly match with other scattered 8-mers along the genome. Your solution must therefore first answer the following questions: What is the probability of a random 8-mer match by chance? (Assume equal probabilities of A,C,T,G.) What is the probability of k matches occurring in a "gene" region? Address any caveats in your probabilistic model, or additional assumptions you make. You are graded on the strength of your model and the quality of argument.

Hint: This could help in a simple model: MATLAB function binocdf(x, n, p) gives the probability of $\leq x$ successes in n trials with p probability of success.

Extra-credit: Allow for sequencing error by implementing an inexact match of up to k mismatches. You will also need to add complexity to your probabilistic model in your argument.

Problem 2 COMPUTATIONAL: De Bruijn Graphs (35)

Given an integer k, the De Brujin for sequence assembly is constructed by creating 4^k nodes each representing a unique k-mer (piece of DNA of length k). Let a and b be two nodes with DNA sequences of $a_1, a_2, ..., a_k$ and $b_1, b_2, ..., b_k$ respectively. We place a directed edge from node a to node b if $a_2, a_3, ..., a_k = b_1, b_2, ..., b_{k-1}$, in other words, if the last $k-1$ characters of a overlap with the first $k-1$ characters of b.

Write a program that takes as input a FASTA sequence representing a genome and a k. Construct the De Bruijn graph for the input k and output a visual representation of the graph. Use the input genome to add the edges to the De Bruijn graph. You can use whatever external libraries to draw the graph but we'll also accept the graph as text output in DOT (`http://www.graphviz.org/doc/info/lang.html`) or Mathematica (look up GraphPlot function) format. The De Bruijn graph of the full $\sum K\Omega 9$ genome is likely too large to visualize so we've provided a reduced-sized $\sum K\Omega 9$ genome on the website. Feel free to try to use a large segment of the $\sum K\Omega 9$ genome or any other biological sequences as long as you can make sense of the output graph. Adjust the input k and describe how varying k perturbs the structure of the graph. The ultimate goals is to use De Bruijn graphs for de novo assembly; however, in this scenario, DNA sequencing errors greatly complicate the construction of the assembly. Can you think of how DNA sequencing errors might manifest themselves in the De Bruijn graph (an example would be great!) and suggest (*not code*) a method to correct them?

Extra-credit: Mathematica implementation *and/or* show the path through the graph representing the reconstruction of the input DNA sequence.

Problem 3 Information theory (15)

We will present a number of problems that will lead us to the concepts of information and entropy. This will help us understand the use of relative entropy in the de-noising of statistical significant pairwise alignments with substitutions matrices such as PAM matrices that we presented in class.

As a start, let us think of a probability space and events in that space. An event that is highly probable conveys little information: e.g., "the sun will raise tomorrow" or "I have to work tomorrow". On the other side, events that are very unlikely convey much information, e.g., class will be canceled tomorrow due to an alien invasion.

Informally, in English language, information is associated with with "surprise" and "meaning". In information theory, information describes "surprise" - the concept of meaning, as in meaning of an English sentence is not part of the mathematical notion of information, just "surprise" which is defined via events with low probability.

We obtain *1 bit of information* when we get a truthful answer to a yes or no question.

Problem: Weighing coins (5)

- Suppose you have 9 coins, eight of which are identical and one is lighter. You possess a scale and the only action you may take is to weigh two sets of coins A and B and determine whether $weight(A) > weight(B)$, $weight(B) > weight(A)$, or $weight(A) = weight(B)$. What is the minimum number of weightings required to determine which coin is the light coin?
- Extra points (2). Generalize the problem to n coins with one lighter coin
- Extra points (4). Generalize the problem with n coins with $m \leq n$ lighter coins

Problem: 20 Questions game (10)

- Alice and Bob are playing a game. Alice secretly picks a number between 1 and 2^{20} = roughly 1 million. Bob asks questions and Alice has to answer truthfully but only "yes" or "no" answers. If, in 20 questions or less, Bob finds Alice's secret number, Bob wins; otherwise Alice wins.
- In fact the game can be played with Alice picking a secret word from the English dictionary, known to be of size roughly 2^{20}. And Bob asking questions for which Alice gives again yes or no answers.
- Prove that Bob can always win by describing the winning algorithms for these two games.
- Extra points (10). Assume Alice is allowed one lie. How many questions does Bob need to guarantee a victory?

Problem Extra credit: COMPUTATIONAL: Hirschberg Algorithm (40)

Apolipoprotein L, 3 (APOL3) and apolipoprotein L, 4 (APOL4) are believed to be paralogs. Two genes are paralogous if they are related by a gene duplication event. One hypothesis concerning gene duplications is that the new copy of the gene can evolve freely while the old copy can preform its normal functionality; however the genes should still be related at the sequence level. To examine this hypothesis, we want to align these two gene sequences and analyze the results. APOL3 and APOL4 are very long genes and are difficult to align using the standard Needleman-Wunsch algorithm (using 1 Byte to represent each cell would yield a DP table of $>$ 3.2GB).

In order to align these two sequences, you are tasked to write a program that performs Global Alignment using Hirschberg's algorithm. Your program should take as input sequences s_1 and s_2 and similarity matrix M and should use no more than $O(n)$ space. Please implement Hirschberg's algorithm **without** affine gap penalties. We've uploaded the APOL3 and APOL4 gene sequences to the homework page as well as class notes and a book chapter on the algorithm. Analyze the alignment. What would you conclude about the relationship between the two sequences? Is global alignment the appropriate alignment algorithm to use in this case?

The sequence alignment applet linked from the resources page (`http://drp.id.au/align/2d/AlignDemo.shtml`) is a great resource for this problem. Because space is less of an issue with this algorithm, Mathematica implementations earn super extra-credit and a Pastiche pie award!!!!.