6.047 / 6.878 Computational Biology: Genomes, Networks, Evolution
Fall 2008

6.047/6.878 Fall 2007 Problem Set 1

Due: Monday September 15 at 8pm

1. **Evolutionary distance and whole-genome duplication**. In this problem, you will implement the Needleman-Wunsch algorithm for pairwise sequence alignment, apply it to the protein-coding sequences of related genes from several mammalian genomes, and use the results to learn about their evolution.

   (a) On the class web site, we have provided a skeleton program `ps1-seqalign.py`, which you will complete. We provide a traceback routine, but you will write the code to fill in the score and traceback matrices. The skeleton program specifies a substitution matrix and gap penalty. If you so choose, you may rewrite the program in any programming language. Please submit (1) the portion of the code that you wrote; (2) an optimal alignment the two sequences `AGGTGAT` and `AGTAA`, and corresponding score matrix F with the optimal path indicated; and (3) the score of the alignment of the human and mouse HoxA13 genes, which we also provide on the web site.

   The Hox cluster is a set of genes that are crucial in determining body plan formation during embryo development. They are found in all bilateral animals, in species as distant as the fruit fly. The fruit fly has one Hox cluster, while most vertebrates have four. It is thought that vertebrates have undergone two rounds of whole-genome duplication, giving rise to four Hox clusters from the ancestral one.

   In the remainder of this problem, you will use your Needleman-Wunsch alignment program to analyze the sequences of several Hox genes, and estimate the date of the most recent vertebrate whole-genome duplication. In particular, we are interested in using the N-W alignment score as a distance metric between two sequences.

   (b) Make minor adjustments to your alignment program so that the score it computes can be interpreted as a distance metric. For example, the score of a sequence aligned with itself should be zero, and sequences that are more dissimilar should give a score with a greater magnitude. Describe the changes you made in your handin; no code is necessary.

   (c) Apply your modified program to compute a distance between the human HoxA13 gene and the mouse HoxA13 gene. The fossil record shows that human and mouse diverged about 70 million years ago.

   (d) The modern mammalian genes HoxA13 and HoxD13 arose from a single ancestral gene by whole-genome duplication, long before the human-mouse divergence. We provide the sequences of the human and mouse HoxD13 genes on the web site. Use your distance metric and your results from part (c) to estimate the date of the whole-genome duplication that gave rise to HoxA13 and HoxD13. Make sure to state the assumptions underlying your estimate.

2. **Dynamic programming for multiple sequence alignment**. Give a dynamic programming recurrence for computing the optimal global alignment of three sequences. You do not need to describe how to fill in the dynamic programming table. Assume that you have a function, $s(i, j, k)$, that will provide the score of aligning three nucleotides and/or gaps.

3. **Random projections**. Random projections allow us to perform inexact motif matching. We consider two sequences of equal length to match if a randomly selected subset of their positions match. Here we will analyze some properties of random projections.

   Given two sequences v and w, define the *Hamming Distance*, $d_H(v, w)$, as the number of positions in which v and w differ.

   (a) If the Hamming distance between two sequences v and w of length d is $d_H(v, w)$, what is the probability that k positions selected at random (with replacement) will agree?

   (b) Plot the probability of two sequences having the same random projection when $d = 10$ and $k = 3$. How does this plot change as k increases?

   (c) We can also compute several random projections to compare sequences. Let l be the number of random projections we perform. Calculate the probability of v and w having at least one equal projection.

4. **Sequence hashing and dotplot visualization**. As you have seen in problem 1, sequence alignment is a quadratic time algorithm. Full sequence alignment is therefore only feasible for sequences near the length of a single gene. To align larger regions of a genome, heuristic approximations are typically used. In this problem, you will use hashing techniques to guide the alignment of a 1 megabase (1 million nucleotides) region surrounding the HoxA cluster in human (`human-hoxa-region.fa`) and mouse (`mouse-hoxa-region.fa`). You will use dotplots to visualize the performance of various hashing methodologies.

   The code provided (`ps1-dotplot.py`) finds all 30-mers in the human that also appear in mouse. On a dotplot, each of these matches is represented as a single dot at (x, y), where x is a coordinate for the beginning of a 30-mer in human and y is a coordinate for the beginning of a matching 30-mer in mouse. We provide a plotting function that will produce dotplot images. The format of the image is determined by the file extension (*.ps, *.png, *.jpg). There is also code for heuristically judging the *specificity* of the matches (the fraction of matches that occur near the diagonal of the dotplot).

   (a) Run the script unchanged to generate a dotplot for all exact matching 30-mers. Describe what you see. How many hits are there and what percentage fall near the diagonal? Do you observe any structure in the off-diagonal hits? What types of genomic elements could cause such a pattern? Why are matches that are close to the diagonal more likely than off-diagonal matches to represent "correct", or orthologous, alignments?

   (b) Make the following modifications to the script and report how the plot changes qualitatively and quantatively (how many hits, what percentage are near the diagonal). Also briefly describe how you implemented each change.

   i. Modify the script to find all *exact* matching 100-mers
   ii. Modify the script to find all 60-mers that match every *other* base
   iii. Modify the script to find all 90-mers that match every *third* base
   iv. Modify the script to find all 120-mers that match every *fourth* base
   v. Modify the script to find all 100-mers that allow a *mismatch every third base*

   (c) Although parts a, b.ii, b.iii, and b.iv require the same number of matching bases ($30 = 60/2 = 90/3 = 120/4$), one of them is more specific to the diagonal. Explain why this might be so.

   (d) Explain the trade-off you see between number of hits near the diagonal (sensitivity) and the percentage of hits near the diagonal (specificity). How is the trade-off affected by the hashing parameters?

   (e) Modify the script to also detect inversions. An inversion occurs when a stretch of DNA is spliced out and reinserted in reverse orientation. For example,

   CGT[GATT]AGA
   ⇓
   CGT[AATC]AGA

   The `human-hoxa-region-modified.fa` file contains a version of the Hox region with an artificial inversion. Use the dotplot to locate the inversion in human. (Note: ignore the sensitivity measure)

5. **(6.878) Genome-scale sequence alignment**. LAGAN is one of the most popular programs for genome alignment. Read the following paper (a copy is provided on the course web site), and describe how it incorporates many of the ideas for genome alignment seen in lecture to increase speed and sensitivity. What other ideas are used and how do they affect performance?

   Michael Brudno, Chuong Do, Gregory Cooper, Michael F. Kim, Eugene Davydov, Eric D. Green, Arend Sidow and Serafim Batzoglou (2003). LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA. *Genome Research* 13(4):721-31.

   Optional (extra credit): Download and run LAGAN on the 1Mb sequence provided for problem 3. Overlay the path of the actual alignment to your plots from problem 3. How does it compare to the dotplots you have generated? Do you find differences in the alignment quality between regions of high match density from problem 3, and regions with sparser matches?