Elements of Bioinformatics (autumn 2010)

Exercise 1

Group 1: Tue 9.11, 14-16, BK107; Group 2: Thu 11.11, 8-10, B222

Choose any 5 assignments from below (each assignment gives 1 point, 5 points is maximum for each week; you can do more for better learning of course).

1. **Dynamic programming and Python.**

   Python code in `www.cs.helsinki.fi/u/vmakinen/elements10/ex1_1.py` computes the global alignment of two protein sequences using BLOSUM62 matrix. The sequences are assumed to be in files `alpha.fasta` and `beta.fasta`. Find fasta files of human hemoglobin subunits alpha and beta in NCBI sequence databases `http://www.ncbi.nlm.nih.gov/sites/entrez?db=Nucleotide` and run the program.

   a) What is the output?

   b) Modify the program so that it computes the local alignment. What is the output?

2. **Affine gap penalties.**

   How should the tables/recurrences M and G be initialized so that $\max(M_{m,n}, G_{m,n}) = S_{m,n}$? Here M, G, and S are those defined at the lecture slides 54-55 at the first lecture.

   Modify the above Python code to compute the alignment with affine gap penalties.

3. **Traceback.**

   Modify one of the Python codes of previous assignments to compute the optimal alignment instead of just the score.

4. **Restricting alignments.**

   Modify the two-table algorithm for computing alignment under affine gap penalties so that the algorithm finds an optimal alignment among aligments where an insertion cannot be directly followed by a deletion and vice versa.

5. **Alternative affine gap cost algorithm.**

   Develop an alternative to the two-table affine gap cost computation exploiting the fact that $S_{i',j'} - \alpha - \beta(j - j' + i - i' - 1)$ can be written as $S_{i',j'} - \alpha - \beta(-j' - i' - 1) + \beta(j + i)$. That is, one can keep row maxima of the *invariant* values $S_{i',j'} - \alpha - \beta(-j' - i' - 1)$ and column maxima of row maxima. Then proceeding column-by-column and updating these maxima, value $\beta(j + i)$ can be added to the current column maxima so that the value is identical to maximum of $S_{i',j'} - \alpha - \beta(j - j' + i - i' - 1)$ among $i' \leq i$, $j' \leq j$, $i' + j' < i + j$. Formalise this idea as a working recurrence. The algorithm should work in $O(mn)$ time.[1]

[1] Algorithms for logarithmic gap cost model are extensions of this idea.

6. **Justification for score matrices.**

   Prove that Kullback-Leibler divergence (see lecture slides) is always non-negative.

7. **Progressive multiple alignment I.**

   Find out what are the three different hierarchical clustering methods commonly used. Which one is most related to bioinformatics?

8. **Progressive multiple alignment II.**

   Write a recurrence for aligning two multiple alignments following the example given at the lectures under either the SP score or the entropy score (assuming gaps treated as other symbols).

9. **Search space pruning and multiple alignment.**

   Program called `MSA` (`http://www.ncbi.nlm.nih.gov/CBBresearch/Schaffer/msa.html`) implements the search space pruning idea presented at the lectures. Try out the program on the four human hemoglobin subunits. Report the multiple alignment it finds (use suitable parameters to obtain optimal alignment).