Molecular Sequence Algorithms, Spring 2004
Exercise 7, March 8, 2004, at 12.15–14 in E16–17

The in-class course exam originally scheduled for March 15 is replaced by a home exam. Written solutions to home exam assignments shall be returned by March 15 at noon. The questions and instructions are available at the course homepage.

1. Present the Smith-Waterman dynamic programming table for computing optimal local alignments between the strings “Helmin” and “kemia”. Apply a scoring scheme that scores matches by +2, and mismatches and insertions/deletions by −1. What is the optimal local alignment of these strings?

2. (~Gusfield, Ex. 11.11) The number of optimal alignments between two strings can grow exponentially with respect to the length of the strings. Thus it is not possible to enumerate all of them in worst-case polynomial time. Present a dynamic programming algorithm that computes the number of optimal alignments between given strings \( S_1[1\ldots n] \) and \( S_2[1\ldots m] \) in \( O(nm) \) time (either for edit distance or similarity).

3. The computation of optimal alignments with affine gap weights would seem to require four complete dynamic programming tables \( V(i, j), E(i, j), F(i, j), \) and \( G(i, j) \). Outline a method that computes the value \( V(n, m) \) of an optimal alignment using less space.

4. Show that a center string can be found in polynomial time. What is the complexity? What does this give as the total complexity of computing a multiple alignment by the center star method?

5. FASTA and BLAST exclude from closer examination those parts of the database that do not contain exact occurrences of “k-tuples” or “words”. Such a strategy can be formally justified under a unit-cost scoring scheme which only penalizes the number of single-character differences (mismatches, insertions and deletions): Let a pattern \( P[1\ldots n] \) be broken into \( d+1 \) disjoint blocks \( B_1 = P[1\ldots i_1], B_2 = P[i_1 + 1\ldots i_2], \ldots, B_{d+1} = P[i_d + 1\ldots n] \). Then a substring \( T' \) of text \( T \) can be an approximate occurrence of \( P \) with at most \( d \) differences only if some block \( B_h \) has an exact occurrence in \( T' \). Explain why this holds.

6. I’ll post a course evaluation form at the course homepage. Fill and return the form. Feedback on a “beta-test version” of a course like ours is especially valuable for improving the course and for deciding about its possible future.