Algorithms for Bioinformatics (autumn 2010)

Exercise 2 (Mon 20.9, 10-12, C222)

1. **Continuing with Python.**

   Write a Python program that implements $TotalDistance(v, DNA)$ -function from the lecture.

2. **Partial digest.**

   Consider partial digest

   $$L = \{1, 2, 3, 3, 4, 5, 5, 6, 8, 9\}.$$

   Solve Partial Digest problem for L (i.e. find X such that $\Delta X = L$).

3. **Motif finding using black box program.**

   You have access to a program X that, given set S of DNA sequences, motif length m, and threshold k, finds all motifs $A = a_1 a_2 \cdots a_m$ that occur with at most k mismatches in each of the DNA sequences in S. Program X outputs each motif with a list of all its occurrences. You are studying a set of genes for which earlier studies indicate that there might be a transcription factor that binds to a motif that consists of two *half-sites*, i.e., having the structure $a_1 a_2 \cdots a_{m'} NNNNNb_1 b_2 \cdots b_{m'}$, where N is any symbol, B is the reverse complement of A, and A and B can have together at most k' mismatches in their occurrences in each DNA sequence in S. You try to run program X with parameters $m = 2m' + 5$ and $k = k' + 5$ but it takes too long to run. How would you proceed in finding your motif? Does your approach allow the amount of symbols N to vary?

4. **Modifying your own motif finder I.**

   Modify `BranchAndBoundMedianStringSearch()` pseudocode studied at the lecture so that it finds motifs consisting of half-sites as in assignment 3. (You may use `TotalDistance()` metric instead of the one used in assignment 3., if you define that N against any symbol scores 1.)

5. **Modifying your own motif finder II.**

   A suffix tree -based approach was described at the lecture for finding exact motifs. Modify it to find motifs consisting of half-sites (assignment 3. with $k' = 0$).

6. ***Voluntary extra programming exercise.***

   Write a Python program that implements $BranchAndBoundMedianStringSearch(DNA, t, n, l)$ -function from the lecture.