Algorithms for Bioinformatics (Autumn 2011)

Exercise 4 (Thu 6.10, 10-12, BK107, Niko Välimäki)

1. **Understanding costs and scores.**

   Consider the alignment below:

   ```
   ACGATGAT--CT
   A-GA-CATAAAT
   ```

   What is the cost of the alignment in the unit cost edit distance model? What is the global alignment score the alignment defines, with the mismatch and indel penalties -1 and match premium $+1$? What is the best local alignment score inside the given global alignment?

2. **Understanding matrix filling.**

   Compute the edit distance between `ACGTA` and `AGAA` by filling the dynamic programming matrix, and output the optimal alignment(s).

3. **Simulating small parsimony.**

   a) Solve the small parsimony problem using Sankoff's algorithm, sequences `ACAC,ATAT,CTCT,GTGT` being the leaves (from left to right) of a balanced binary tree.

   b) In the large parsimony problem the leaves can be in any order and the tree shape is not fixed. Does any other tree give better parsimony score for our example?

4. **Overlap alignments: tricks with zeros.**

   We are interested in *overlap alignments* of strings A and B such that suffix of A is aligned against prefix of B. For example, an overlap alignment of `ACGATGAT` and `GACATAAAT` is

   ```
   ACGATGAT
     GA-CATAAAT
   ```

   a) Derive a variant of global alignment recurrence that gives the best scoring overlap alignment of A and B.

   a) Derive a variant of edit distance recurrence that gives the overlap alignment of A and B with minimum cost, with the restriction that overlap should be at least of length ℓ. (Why is such restriction required?)

5. **Reverse translation (1-2 points).**

   An alternative to spliced alignment for aligning protein against DNA is to directly model *reverse translation*. Assume that you have the complete DNA sequence of some eukaryotic organism and the amino acid sequence of some protein suspected to be coded by a gene in the organism DNA (the protein could be predicted from another organism). The gene coding the protein is unknown and your task is to locate it from the DNA.

Give a dynamic programming algorithm to locate a gene (a continuous region in DNA consisting of exons and introns) containing *fewest number of introns*, such that the concatenation of exons can be translated into the given protein sequence. What is the running time of your algorithm?

Can you plug in other constrainst to the gene (overall span in DNA, maximum/minimum distance between consecutive exons, minimum size of exons, start and end codons, `GT` and `AG` dinucleotides in intron ends, some amount of edits to align exon concatenation into one translating into the given protein, etc.)?

An example alignment with 3 exons is given below (recall exercise 1.5 and codon to amino acid translation table).

```
ATGGTTACAAGAGT...AGCAAGCACTCGT...AGCTTCCCTAA

  M  V  T  R         Q  A  L         L  P  *
```