

BMI/CS 576 Fall 2009

Homework #2

Prof. Colin Dewey

Due Thursday, October 15th, 2009 by 11:59pm

The goal of this assignment is to become more familiar with the algorithms for sequence alignment.

To turn in your assignment, copy all relevant files to the directory:

`/u/medinfo/handin/bmi576/hw2/USERNAME`

where `USERNAME` is your account name for the BMI network. You must submit a file named `README` to this directory, which gives directions on how to compile (if necessary) and run your programs. For each question below, the `README` file should list the files relevant to that question (e.g., code, other files with written answers).

1. Briefly describe a *greedy* algorithm for producing a *global* alignment of two DNA sequences. Give one example for which this algorithm gives a poor (far from optimal) result.
2. Suppose we wish to partition a DNA sequence into segments, such that each segment is predominantly composed of a single base. We decide to create such a partition, p , by maximizing an objective function

$$f(p) = a \cdot n(p) + b \cdot m(p) + c \cdot x(p),$$

where $n(p)$ is the number of segments created by p , $m(p)$ is the number of positions that match the base assigned to their segments, and $x(p)$ is the number of positions that do not match the base assigned to their segments. In general, b will be a positive value and a and c will be negative values. For example, the score of the partition

AATGTACA|CGCC|TTGTT|CCC|GAGGGT

would be $5a + 18b + 8c$ (the segments in this partition would be labeled, from left to right, as A, C, T, C, G).

- (a) How many possible partitions are there for a sequence of length L ? (Ignore the labels of each segment)

- (b) Give an optimal partition and its score for the sequence

CGCCATTAT

with $a = -2$, $b = 2$, and $c = -1$.

- (c) Describe a *dynamic programming* algorithm for computing the optimal partition score of a DNA sequence of length L , given values for a , b , and c . You do not need to describe how to find a partition that gives the optimal score. (Hint: break the problem into $4L$ subproblems, four subproblems for each prefix of the sequence.)
3. Write a program, `AlignLocal`, that takes as input two protein sequences, an amino acid substitution matrix, a gap score, and a space score, and outputs an optimal *local* alignment for the *affine gap* scoring scheme (i.e., the score for a gap of length $k > 0$ is $g + sk$).

The syntax for running the program should be:

```
AlignLocal seqs_file matrix_file g s
```

where `seqs_file` is the name of a file containing two protein sequences, one per line, `matrix_file` is the name of a file containing an amino acid substitution matrix, `g` is an integer gap score, and `s` is an integer space score. The program should print to standard output an optimal local alignment, with one sequence per line.

Your program should print out only one optimal alignment in cases where there are multiple optima. If there are multiple starting points for the traceback, you should start from the lowest, rightmost element in the M matrix. For example, if the entries in the M matrix with the maximum score are $(4, 5)$, $(5, 3)$, and $(5, 4)$, you would start at $(5, 4)$. During the traceback, you should take the *highroad* alignment path. Also, you should follow the convention that the rows of DP matrix correspond to the characters of the first sequence given, and the columns of the matrix correspond to characters of the second sequence given.

Sample input and output files, as well as a substitution matrix file (to give you the format and scores to work with) are provided at: <http://www.biostat.wisc.edu/bmi576/hw2/>