June 14, 2021

Assignment 9
Algorithms for Sequence Analysis, Summer 2021

Algorithmic Bioinformatics · Prof. Dr. Sven Rahmann

Hand in date: Monday, June 21, before 20:00

Exercise 1: Ukkonen's speed improvement (4 Theory)
For $P =$ `babb` and $T =$ `aababbaaaabab` and edit distance threshold $k = 1$,

(a) find the endpoints of all k-approximate matches of P in T using Ukkonen's improved DP algorithm,

(b) show exactly which cells of the matrix are computed, and

(c) show the $last_k(j)$ boundary as a function of j.

Exercise 2: Shift-And for k-approximate matches (4 Theory)
For the same P, T, k as in the previous exercise, use the NFA and Shift-And algorithm to find the endpoints of all k-approximate matches of P in T. Illustrate which states are active after each text character (using figures or bitvectors).

Exercise 3: Neighborhood (4 Theory)
Determine the edit-distance-1-neighborhood of $s =$ `babba`, i.e., the set of all strings that have edit distance ≤ 1 to s. You might use an NFA to systematically enumerate them.

Exercise 4: Number of co-optimal edit alignments (4 Programming)
We know how to compute the total number of *possible* alignments between two sequences of lengths m and n, respectively. Modify the idea of that algorithm to count the number of optimal alignments of s and t: First, compute the DP matrix including traceback pointers. Then, efficiently count the number of co-optimal paths.
Implement your algorithm as follows: It should take two strings as command line arguments. The output should be edit distance and the number of co-optimal global alignments (with minimal edit cost), separated by space.

Example: Your program should work as follows:

```
> ./program APPARENTLY PARENTHESIS
7 20
> ./program AAAAA AAAABAAAA
4 56
```

In the first example, the edit distance is 7, and there are 20 co-optimal alignments.
In the second example, the edit distance is 4, and there are 56 co-optimal alignments.

Hint: Your are allowed to use and modify the global alignment and traceback code that becomes available in the lecture of Tuesday June 15 (note min here vs. max there!). In this way, you only need to think about counting the co-optimal paths and not about implementing traceback, etc.