Genomics

Freie Universität Berlin, Institut für Informatik
Knut Reinert, Peter Robinson, Sebastian Bauer
Wintersemester 2012/2013

3. Übungsblatt vom 29. Oktober 2012
Diskussion am 8. November 2012

Exercise 1.

If you haven't already done so in the previous exercise, implement a DP algorithm for solving the read mapping problem using one of your favourite programming languages. If m is the length of the read/pattern/query to be sought, the implemented algorithm shouldn't use more than $O(m)$ space. That is, its additional space requirements should be independent of the reference sequence. Ideally, the actual results are provided to clients via an iterator, via a callback, or a similar mechanism.

Exercise 2.

Using Ukkonens algorithm for k-differences matching, find all occurrences of the pattern $P = \texttt{tcaa}$ in the text $T = \texttt{atcatcaatc}$ with up to $k = 2$ differences. Show the dynamic programming matrix, the value of $lact$ for each column and do not compute unnecessary cells, etc.. For one column of your choice, keep track of the auxiliary variables Cn and Cp as well as the whole column vector C (so as to understand their meaning).

Exercise 3.

Prove the correctness of the following observations mentioned in the lecture. C is a dynamic programming matrix computed using the edit distance.

$$\begin{aligned}
\text{horizontal adjacency property} \quad & \Delta h_{i,j} = C_{i,j} - C_{i,j-1} \in \{-1, 0, +1\} \\
\text{vertical adjacency property} \quad & \Delta v_{i,j} = C_{i,j} - C_{i-1,j} \in \{-1, 0, +1\} \\
\text{diagonal property} \quad & \Delta d_{i,j} = C_{i,j} - C_{i-1,j-1} \in \{0, +1\}
\end{aligned}$$

(Hint: induction on $i + j$).

Exercise 4.

Conclude from Exercise 3 (you may use it even if you have not done the proofs) that the value of $lact$ (in Ukkonen's algorithm for string matching with up to k differences) can increase in one iteration by at most one.

Exercise 5.

This time use Myers' bit-vector algorithm for pattern and text in Exercise 2. Describe how to extend Myers' bit vector algorithm for approximate string matching with patterns larger than the word length. Describe you approach as pseudocode.