## Algorithms for Bioinformatics (Autumn 2012)

## Exercise 2 (Thu 20.9, 10-12, B119, Niko Välimäki)

1. Simulating improved breakpoint reversal sort.

Perform the improved breakpoint reversal sort algorithm (page 30 at lecture slides) with $\pi=34658172$ and show all intermediate permutations. Is this the optimal solution to this instance of reversal sorting problem?

## 2. Correctness of improved breakpoint reversal sort.

Prove that the improved breakpoint reversal sort algorithm (page 30 at lecture slides) works correctly. That is, define concepts increasing strip and decreasing strip formally and then prove that if permutation $\pi$ contains one or more decreasing strips, then there is always a reversal that decreases the number of breakpoints $b(\pi)$ at least by one.

## 3. Translocations via reversals.

Page 32 at lecture slides gives an example of a translocation simulated by three reversals. Define formally what translocation operation does for a permutation $\pi_{1} \pi_{2} \cdots \pi_{n}$ and prove that any translocation can be replaced by three reversal operations.

## 4. Implementing improved breakpoint reversal sort.

Write a Python program that implements improved breakpoint reversal sort and analyse the running time of your implementation.

## 5. Shortest approximate superstring.

Let $\mathbf{S}=S_{1}, S_{2}, \ldots, S_{n} \subseteq \Sigma^{*}$ be a set of strings from alphabet $\Sigma$. Given a threshold parameter $k$, an approximate superstring of $\mathbf{S}$ is defined as a string $T$ such that for each $S_{i} \in \mathbf{S}$ it holds $d_{H}\left(S_{i}, T\left[j_{i} \cdots j_{i}+\left|S_{i}\right|-1\right]\right) \leq k$ for some $j_{i}$, where $d_{H}()$ denotes the Hamming distance.

A greedy approximation algorithm for finding the shortest approximate superstring can be derived as follows. Let an approximate overlap of $A=\alpha \gamma, B=\gamma^{\prime} \beta \in \mathbf{S}$ be pair of strings $\left(\gamma, \gamma^{\prime}\right)$ such that $d_{H}\left(\gamma, \gamma^{\prime}\right) \leq k$ and the length of the overlap $|\gamma|=\left|\gamma^{\prime}\right|$ is maximum among all ways to to write $A$ and $B$ in parts $A=\alpha \gamma$ and $B=\gamma^{\prime} \beta$. Iterate the following until there is only one string in set $\mathbf{S}$ : (1) Choose $\alpha \gamma, B=\gamma^{\prime} \beta \in \mathbf{S}$ with maximum approximate overlap; (2) remove $A$ and $B$ from $\mathbf{S}$ and insert $\alpha \gamma \beta$ into $\mathbf{S}$.

Simulate the above greedy algorithm with $k=1$ on the set \{ACACGATC, ATGACAAA, TAATAAGA, CAGGATCA\}.

Is the solution of your simulation a valid approximate superstring? Does the algorithm always find a valid approximate superstring? If not, give a modification so that it does.

