Genomics

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

4. Übungsblatt vom 5. November 2012
Diskussion am 15. November 2012

Exercise 1.

The basic pigeonhole principle is defined as: if n discrete items are put into m pigeonholes with $n > m$, then at least one pigeonhole must contain more than one item. Proof this principle.

Exercise 2.

Proof the following generalization of the pigeonhole principle: If n discrete items are put to m pigeonholes, then at least one pigeonhole must hold no more than $\lfloor n/m \rfloor$ objects with $\lfloor x \rfloor$ denoting the floor function.

Exercise 3.

What are the purposes of filtering and verfication phases of a reading mapping procedure? When are filtering algorithms most effective?

Exercise 4.

A text T of length 500 contains an occurrence Occ of a pattern P of length 100 with at most 6 errors. In how many pieces do you have to split P (i.e. $P = P_1 P_2 \dots P_k$), to make sure that ...

1. T contains one of the P_i without error?
2. T contains one of the pieces with at most 2 errors?
3. Prove following variant of the pidgeonhole principle: Let Occ be an approximate occurrence of a pattern P with k errors. Let $P = p^1, \dots, p^j$ be the concatenation of pieces of P and $a_1, \dots, a_j$ be non-negative integers with $A = \sum_{i=1}^{j} a_i$. Then there exists an $i \in 1, \dots, j$, such that there exists a substring of Occ that matches p^i with no more than $\lfloor a_i k / A \rfloor$ errors.
4. Why is there the need of the a_i values and how are they used in the PEX algorithm?

Exercise 5.

Find the pattern $P =$ `filter` in the text $T =$ `pex_hierarchical_verification_filter` with at most $k = 2$ errors. Determine the verification costs of non-hierarchical filtering directly following the pidgeonhole principle given in Exercise 4. That is, split the pattern into $k + 1$ subpatterns and search for perfect matches. Compare the costs with the verification costs implied by the PEX algorithm that is run on the same input.