Genomics

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

2. Übungsblatt vom 23. Oktober 2012
Diskussion am 1. November 2012

Exercise 1.

a) Write up the definition of the hamming distane and edit distance. For the edit distance, consider only *insert*, *delete*, and *replace* operations. Each operation has an cost of one.

b) Determine the hamming distances and the edit distance between following pairs of strings.

```
TGGTACTTCTC  TGGTGGTGG  TAGGTGGTG
TAGTTCTTCTT  TGGTGGTGG  TGGTGGTGG
```

c) Write a function, method, or programm in a programming language of your choice to determine the edit distance between two given text strings. Test it with the examples of b)

Exercise 2.

a) What is the purpose of read mapping in a next generation sequencing workflow? Which contraints make it special from more general approximate string matching problems?

b) Give a formal definition of the read mapping problem.

c) Solve the following read mapping problem instances. The distance function is the edit distance. All reads are of good quality. Write down all matches with a distance not greater than 2.

```
Reference: TGGTACTTCTCCTACCCCCCA
Read #1:   TACTT
Read #2:   CTTTC
Read #3:   TCCTC
Read #4:   CCGCC
```

d) This is optional. Think about how a simple read mapper could be implemented, for instance, by reusing the result of 1b). The input to the function could be a reference sequence, a read, a constant k and a distance function. The output should contain locations of the reference sequence where the read matches with a distance not greater than k. There is no need to take efficiency considerations into account. Implement it as a testable function/method in one of your favourite programming languages. Show the correctness of your implementation by comparing it with the result of c).

Exercise 3.

To make yourself familar with various read mappers, you should reproduce the Rabema benchmark that was partly introducted within the lecture. Instructions how to get benchmark suite running are available from `http://www.seqan.de/projects/rabema/`. Especially interesting is to go through "For the Impatient" and the Rabema Manual. It is suggested to consider at least two read mappers. Widely used read mappers are for instance:

- Bowtie 2 from `http://bowtie-bio.sourceforge.net/bowtie2/index.shtml`
- BWA from `http://bio-bwa.sourceforge.net/bwa.shtml`
- Mosaik from `http://code.google.com/p/mosaik-aligner/`
- RazerS from `http://www.seqan.de/projects/razers/`

The first solution to this assignment is a script in one of your favourite scripting language (bash, Python, Perl, etc.) or a makefile that can be used to run the benchmark. It is assumed that all necessary tools are available from the command line path. As a reference sequence, you can use the one supplied with Rabema. Other than that, the script should be system-independent (i.e., don't use any path that is specific to your system).

The second solution to this assignment is the result of running the benchmark.