Genomics

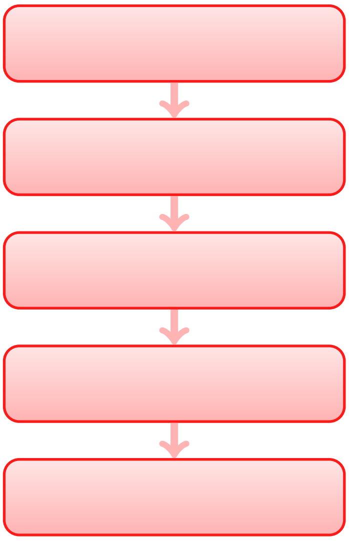
Freie Universität Berlin, Institut für Informatik
Knut Reinert, Peter Robinson, Sebastian Bauer
Wintersemester 2012/2013

1. Übungsblatt vom 18. Oktober 2012
Diskussion am 25. Oktober 2012

Exercise 1.

You should design a workflow for seeking for mutations in patient DNA. The workflow should consist of five major processes. The five processes are called 1) **Whole Genome Mapping**, 2) **Raw Data Analysis**, 3) **Annotation**, 4) **Raw Data Generation**, 5) **Variant Calling**. Fill in the terms in an order that actually makes sense.

Workflow Description Input



Furthermore, associate following phrases with each of the processes: 1) **Sample preparation and sequencing**, 2) **Base calling**, 3) **Alignment to a reference genome**, 4) **Detection of genetic variation**, 5) **Linking variants to biological information**.

Associate following input types with each of the processes: 1) **Fastq files**, 2) **Prepared samples**, 3) **SAM files**, 4) **VCF files** 5) **Intensity files**.

Exercise 2.

a) What kind of measure is the PHRED quality score, how is it defined, and in which step of the workflow does it come into play? How are the scores usually encoded in ASCII files produced by the Illumina pipeline?

b) Suppose that a sequence run produced a read sequence with following bases and PHRED-like scores

```
G A T T T G G G G T T C A
G E D E E E E C C B B ; 9
```

Note that in this exercise, the PHRED-like scores are encoded using an ASCII offset of 33 corresponding to encoding of Sanger and Illumina 1.8+ FASTQ format.[1] Using the definition of the PHRED scores (see slides 58 of the first lecture), what is the probability that the entire read is accurate, i.e., doesn't contain any wrongly called base? What is the expected number inaccurate reads if a total of 1,000,000 sequences of similar quality were generated?

c) Now, write down a function, method, or program in a programming language of your choice that returns the probability that a sequence with a given PHRED string in ASCII representation of given length is accurate. For instance, in the C programming language the prototype could look like:

```
double prob_of_read_being_accurate(const char *quality, int len);
```

Test this function by calling it with the example in b).

Exercise 3.

a) What is a CIGAR string and where in the workflow above does it come into play?

b) Find the CIGAR string of following alignment. The position location shall be given one-based. Use only M, I, and D operators.

```
Reference:  C  C  A  T  C  C  T     G  A  A  C  T  G  A  C  T  A  A  C
                     |  |  |  |     |  |  |     |  |  X  |  |
Read:                T  C  C  T  A  G  A  A     T  G  G  C  T

Pos:
CIGAR:
```

c) Reconstruct the alignment of the following read that has been mapped to the given part of the reference sequence.

```
Reference:  A  G  C  A  T  T  A  C  T  A  C  T  A  A  A  T  T  T
Read:       C  A  T  C  G  A  T  A  C  T  A  A  A
Pos:        3
CIGAR:      4M1D1M1I7M


Reference:


Read:
```

d) Now, write down a function, method, or program in a programming language of your choice that returns the CIGAR representation of a given alignment. For instance, in the C programming language the prototype could look like:

```
char *cigarize(const char *reference, const char *read);
```

Parameter `reference` and `read` are null-byte terminated strings of the same length. Gaps in the respective sequences resulting from the previously calculated alignment are indicated using spaces (just as in b)).

[1] see `http://en.wikipedia.org/wiki/FASTQ_format` for other possibilities