Bachelor Thesis
Optimized pattern matching in genomic data

Synopsis

Martin Westh Petersen - mqt967
Kasper Myrtue - vkl275

23. Februar 2015

1 Problem definition

Is it possible create a program with the same core functionality as scan_for_mathces, but with an equal or increased performance every time?

2 Limitations

We will be implementing the core functionality of scan_for_matches which we define as follows:

- Constructing literal pattern units (e.g. "AGUUG") that are used for finding a specific sub-sequence in the database sequence.
- Constructing ambiguous pattern units (e.g. "4..8") that match any sub-sequence with a possibility for a flexible range.
- Combining pattern units to a full pattern that defines the full search criteria when scanning the database sequence.
- Allowing any pattern unit a specified number of insertions, deletion and mismatches (e.g. "AGGUAAA[2,0,3]").
- Defining variables for referencing pattern units (e.g. "p1=5..6") that can be used to find related patterns that are unknown before search (e.g. p1=5..6 p1[1,0,0])
- The ∼ symbol can be inserted in front of any pattern unit to indicate that we are looking for the reversed complement of that pattern unit.

Beside these already existing features, we will be implementing a more optimized way of searching for matches, given complex patterns consisting of multiple pattern units. E.g. the order of which the different pattern units are searched can be optimized to increase performance instead of going through the pattern units from one end to the other.

In case of excess time, secondary features may be implemented:

- Logical "or" between patterns (e.g. "(AUUG | AGGG)") that matches either of the sub-sequences.
- A possibility for defining custom "pairing rules" (e.g. "r1=au,ua,gc,cg,ga,ag") that can be used for for defining allowances when comparing a reversed complement pattern unit (e.g. "r1∼p1").
- An analysis of the complexity and running time of the program.

Every feature from scan_for_mathces beyond what has been mentioned will not be implemented.

3 Motive

Pattern matching functionality for strings in genomic data is very useful, but requires a good performance due to huge amounts of data. Scan_for_matches serves this purpose, but big improvements in performance can be made. On top that, the code is poorly documented, lacks version control and the code is hard to read and maintain.

4 Tasks and schedule

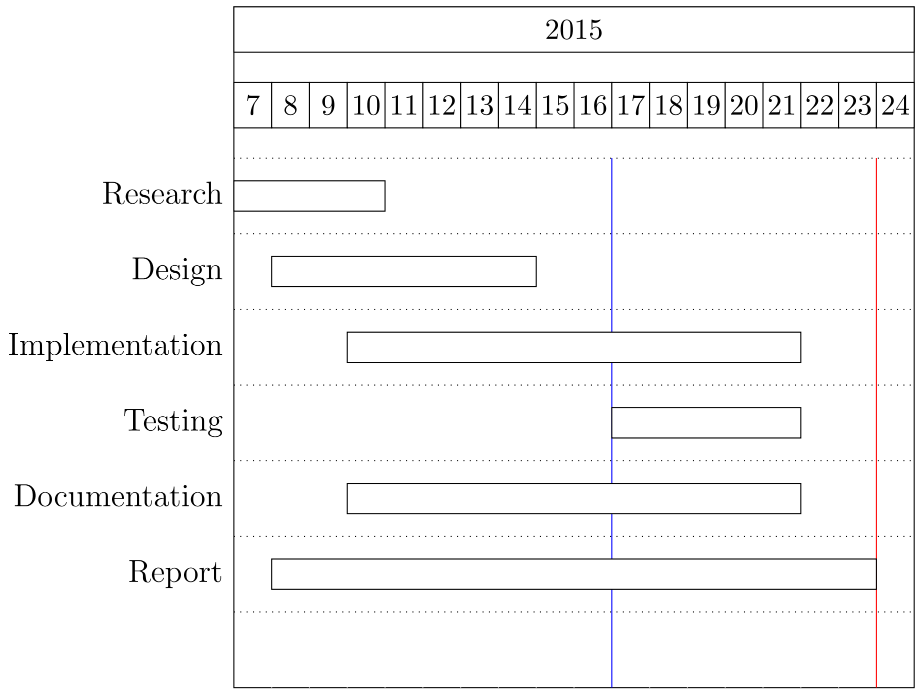

Figure 1: Schedule