Bachelor Thesis

Optimized pattern matching in genomic data

Report

Martin Westh Petersen - mqt967
Kasper Myrtue - vkl275

20. April 2015

Indholdsfortegnelse

1 Program structure

A list of actions our program should do in order to execute a pattern search:

- Read and parse the input line. E.g. "scanFM 'ATTGCCCC[0,1,2]' 'data.txt'". Possibility of writing "$->$ 'output.txt'" which results in the matches not being displayed in the terminal but written the the specified file.
- Parse the pattern into units and save them as different types (objects of different classes that inherit from a common PUNIT class), e.g. EXACT_PUNIT, AMBI_PUNIT etc.
- Choose the order of which to search for the patterns and create a state that readies for this search, for example a list of the punits in correct order with some way of keeping track of the different positions the punits have to with respect to each other.
- Search for the punits in the order chosen by simply calling a ".search()" method for each PUNIT-object. The PUNIT-object's search method invoked is unique for each different type of PUNIT, and returns either True or False. If the search for each PUNIT returns True the match is saved.
- The saved matches are either displayed in the terminal or written in a file, depending on the call of scanFM.

Types of PUNITs that we need

- EXACT - A PUNIT of this type consists of either of the letters 'A', 'C', 'G' and 'T' or the wildcards.
-