COMP90016

Group Assignment: Group Feedback

This document attempts to cover most of the things encountered while "marking" your group assignments.

You all had a genuine attempt at the assignment. I was pleased with the level of participation for this voluntary assignment and with the programming competency displayed. Most programs were functioning as expected and produced results in accordance or close to the assignment spec. Here is the group feedback for the group assignment.

1. The program:
   Surprisingly, there are very few solutions that agree with my sample solution (see LMS). My solution is not very efficient, since it is using BioPython (slow!) and implements string matching manually (for the teaching purposes). It introduces some object-oriented principles by using classes for alignments and the aligner. This is by no means necessary in your future submissions but may be useful to you.
   The most common things that I saw go wrong are:
   - No command line arguments: The file names to the inputs should be passed through the command lines and read by the script with *sys.argv.*
   - Wrong output format: The output should be written exactly as prescribed with a single tab-stop between each of the five fields.
   - Wrong reverse strand coordinates: Many submissions did not take into account (this would probably be due to bad wording in the spec?) the correct coordinate for reverse strand alignments. According to spec:
     All alignments are reported with respect to the forward strand of the reference. That is, if a read aligns to the reverse strand, the position should still reflect how far away the read is from the 5' end of the reference.
     For those that mapped the read to the reverse complement of the reference, this means that coordinates have to be projected (the start becomes the end and vice versa).
   - Wrong coordinates: The coordinates are supposed to be 1-based.
2. The discussion:
   These are the alignment statistics:
   Reads aligning 0 times (unmapped): 23 (10.550459%)
   Reads aligning 1 time (unique match): 88 (40.366972%)
   Reads aligning 2 or more times (multi-mapped): 107 (49.082569%)
   Why don't 23 of the reads not align to the reference? The most obvious reasons would be differences between the sequenced DNA and the reference genome (such as SNPs) and sequencing errors. Since we are only considering perfect matches, any such differences would cause an unmapped read.
   There is some code in the sample solution that tries to match all but the last base of a read to the reference in case of an unaligned read. This confirms that 16 of the 23 reads have a 1bp difference to the reference in the last position. Single-nucleotide mismatches (due to variation or error) are therefore the most likely cause for unmapped reads.

Why do most of the reads multi-map? The reads are of length 6bp. 4^6=4096. The length of the reference is about 1.3kbp. Arguably, there is enough sequence diversity in 4^6 strings to have mostly unique matches in a short reference. However, the reference is not random and repeating patterns may occur leading to an inflated number of multi-mappers.

We should further consider that the reference has two strands. While these are not independent, they result in additional sequence that reads can align to, and therefore increase the chance of multi-mapping reads.

The expected occurrence of any string of length 6 within the reference is about 0.3 times, so most reads multi-mapping is surprising.