Programming Assignment 2 - Megabase-scale alignment

BIOINFO M260

Part A Due: Monday, January 30th, 2017, 11:59 pm
Part B Due: Wednesday, February 8th, 2017: 11:59 pm

This programming assignment is designed to expand your understanding of sequencing and the difficulty of mapping insertions and deletions.

Overview

In the first two programming assignments for this class, you will solve the computational problem of re-sequencing, which is the process of inferring a donor genome based on reads and a reference.

You are given a reference genome in FASTA format, and paired-end reads.

The first line of each file indicates which project that the data relates to. In the reference file, the reference genome is written in order, 80 bases (A's, C's, G's, and T's) per line.

The paired end reads are generated from the unknown donor sequence, and 10 percent of the reads are generated randomly to mimic contamination with another genetic source. These reads are formatted as two 50 bp-long ends, which are separated by a 90-110 bp-long separator.

Project 2A

This project (especially the grad level) takes a long time to run. To help you stay on deadline, by January 30, you need to have run your code and have submitted it with a nonzero score that will yield some credit on the project.

Starter Code

Starter code for the project is available at `https://github.com/michaelbilow/BIOINFO_M260B`. Remember to start a new branch so you can pull in any changes that I make to the code. As with PA1, you should read the content of PA2, and see if you can understand what it is doing. You should also look to see where your input/output is going to go. This will generate an alignment file in the data folder from which it was executed.

Run `python complex_pileup.py`. This will generate a file of changes and a zipped version of that file formatted correctly for submission.

Download the datasets from the cm124 site, and edit the functions in `basic_hasher.py` and `complex_pileup.py`. You should develop your code on the warmup-difficulty genome, which will allow you to see the answer sets.

You can submit your results as many times as you want to achieve a passing score.

I/O Details

`https://cm124.herokuapp.com/ans_file_doc` should handle most of your questions on reading and writing output.

Pileup

For the purpose of this class, alignment and pileup can be thought of as completely separate processes.

To do this project well, you will likely have to rewrite the complex pileup script. The current script clips the reads every 100 positions, and tries to perform pileup in 100-base chunks. This is not a sensible choice (though it works ok).

Here's an outline of a better algorithm:

```
1. Align reads to reference.
2. Construct consensus sequence
3. Compare the consensus sequence to the reference genome.
4. There will be long sequences (>50 bp) where the consensus matches the reference.
5. Cut out the aligned reads in and around the short chunks where the consensus does not match the refe
6. Assemble the reads into a
7 (optional). Throw out any very long mismatching sequences; these are likely caused by repeated sequen
```

Smith-Waterman Reconstruction

For more enrichment on variant calling using the Smith-Waterman Algorithm, see UCLA Professor Chris Lee's lecture here: `https://www.youtube.com/watch?v=EWJnDMKBEv0`

Grading

SNP Score	No Credit	Full Credit
Undergrad	55	75
Grad	70	90

Indel Score	No Credit	Full Credit
Undergrad	3	13
Grad	15	25

Your total score will be the average of (Your Score - No Credit Score)/(Full Credit Score - No Credit Score) for both SNPs and Indels.