Homework Project 2A - Megabase-scale alignment: 1-Million Length Genome

BIOINFO M122/M222

Due: Sunday April 21st, 2019, 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.

Starter Code

Starter code for the project has been pushed to the Github repository at https://github.com/eeskin/CM122_starter_code. Use `git pull` in the CM122_starter_code repository you cloned to get the updated code, or you can just redownload the files directly from the link. As with HP1, you should read the content of the HP2 code, and see if you can understand what it is doing. You should also look to see where your input/output is going to go.

Tutorial

There are two scripts to look at, similar to homework project 1:

1. `basic_hasher.py` takes in a reference genome, a set of reads and an output file and outputs the reads aligned to the reference genome into the output file. Unlike the trivial algorithm in project 1, it uses a hashing approach to align the reads.
2. `complex_pileup.py` takes in aligned reads and an output file and outputs the SNPs and indels called based on the aligned reads. Unlike the pileup in project 1, this code uses an edit distance approach to call variants.

Running each of the above scripts with the `-h` option should be self explanatory, but here is an example of running them to create a file that can be submitted on the website for the 10K length genome practice data provided for project 2.

1. Download the 10K practice data from https://cm124.herokuapp.com/h2_data_files into the HP2 folder and unzip it. The commands below assume that you have a folder named practice_E_1 in the HP2 folder. If you download and save things in a different place you'll have to adjust the file paths below.
2. Use `basic_hasher.py` to align reads to the genome.

```
python basic_hasher.py -g practice_E_1/ref_practice_E_1_chr_1.txt \
-r practice_E_1/reads_practice_E_1_chr_1.txt -o test_hasher.txt
```

The step above may take ~30 minutes!

3. Use `complex_pileup.py` to call variants. This step uses the output file called `test_aligner.txt` generated by `basic_hasher.py`

   ```
   python complex_pileup.py -a test_hasher.txt -o test_pileup.txt -t practice_E_1_chr_1
   ```

This will generate a file of changes in test_pileup.txt and a zipped version of that file formatted correctly for submission. On the 10K practice genome this takes 3-4 minutes.

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 call variants in 100-base chunks. This is not a sensible choice (though it works ok). Consider that insertions and deletions may span the boundaries of these 100-base chunks and confound your results. Also, try to take advantage of the fact that a large portion of the reference and donor genome are actually identical or only contain SNPs.

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

Chapter 5 in the textbook also goes over these ideas.

Homework Project 2B - Megabase-scale alignment: 100-Million Length Genome

BIOINFO M122/M222

Due: Sunday April 28th, 2019, 11:59 pm

Project 2B is the exact same problem as project 2A, except now the size of the genome is length 100 million. This makes the hashing approach in `basic_hasher.py` very memory intensive, so you will have to make changes to this script in addition to the changes you made to `complex_pileup.py` for project 2A. You may want to consider using Burrows Wheeler Transform to be able to handle this size of genome. You should also consider that this project may take a long time to run even with a relatively efficient algorithm, so make sure to get started early!

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.