Homework Project 3 - 10 Kilobase Sequence Assembly

CS/BIOINFO M122/M222

Due: Tuesday May 25th, 2021, 11:59 pm

This programming assignment is designed to teach you about sequence assembly.

Overview
In this assignment, you are given paired-end reads from an 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.

Your job will be to reconstruct contigs--the longest sequences of the donor that you are confident that you have assembled correctly. The output format is simple, and described on the cm122 site (see: https://cm122.herokuapp.com/ans_file_doc). The approximate length of the donor sequence is 10,000 bases.

Starter Code

Starter code for the project is available at https://github.com/rosie068/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.
The starter code will give you functions to read the input data and write a properly-formatted output file. You will have to write the actual assembler.

Tutorial

We are providing you with the skeleton for one script:
1. `basic_assembly.py` takes in a set of reads, an output file and an output header and outputs the contigs generated by assembly.

Running 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 3.

1. Download your practice and for-grade data from:
10K Spectrum Reads (Not for credit):
http://studentdownloads.s3-us-west-1.amazonaws.com/spectrum_A_1.zip

10K Normal Reads (Not for credit):
http://studentdownloads.s3-us-west-1.amazonaws.com/practice_A_2.zip

10K Normal Reads (For Credit):
http://studentdownloads.s3-us-west-1.amazonaws.com/hw3all_A_3.zip

The commands below assume that you have a folder named practice_A_2 in the HP3 folder. If you download and save things in a different place you'll have to adjust the file paths below.

2. Use `basic_assembly.py` as shown below.

```
python3 basic_assembly.py -r reads_practice_A_2_chr_1.txt -o practice_A_2_output.txt -t practice_A_2_chr_1
```

This will generate a file of contigs in test_output.txt and a zipped version of that file formatted correctly for submission.

*NOTE: -t assigns a genome name to your submission, so that the submission site knows which genome you're attempting to assemble. Your final submission for the project MUST use "-t hw3all_A_3_chr_1" otherwise the submission site will not accept it. Use "python3 basic_assembly.py -h" to see what the -t value should be for each dataset. *

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

I/O Details
https://cm122.herokuapp.com/ans_file_doc should handle most of your questions on reading and writing output.

de Bruijn Graph

The de Bruijn graph using a spectrum of size _k_ is by taking each read, and drawing a directed edge from the chunk of the read from position _j_ to position _j+k_, to the chunk from _j+1_ to position _j+k+1_ for all _j_. Some filtering of this graph needs to be done in the case that the reads have errors, and to filter out data from the "contaminating" sequence.

Other Assembly Algorithms

There are ways to perform assembly that do not involve de Bruijn graphs. You are permitted to attempt a different algorithm. However, be advised that computational efficiency may be an issue.

Grading

To obtain full credit you must obtain the following thresholds.

	Coverage	Accuracy	Contig Size
Undergrad	87	43	5
Grad	92	48	8