

Algorithms in Bioinformatics 2, SoSe2007

Assignment sheet # 9

Christian Rausch

June 17, 2007

To solve this, please download the following

`www-ab.informatik.uni-tuebingen.de/teaching/ss07/albi2/assign/files/assign09.zip`

and modify the file `albi2.assembly.Contigger`.

This download contains the `jloda` library that provides a graph class `jloda.graph.Graph`. Documentation for this can be found in the `java/doc` directory. Also, this download provides `albi2.assembly.ForwardSimulator` that can be used to simulate shotgun sampling reads from the forward strand of a target DNA sequence. You can use the class `albi2.assembly.Contig` and `albi2.assembly.FragmentPosition` to represent contigs, if you wish.

The goal of this week's assignment is to implement a "contigger", that is, a program that takes as input a collection of reads and produces as output one or more contigs.

To keep this simple, we will assume that all reads have been sampled from the forward strand of the target DNA sequence. Moreover, we will assume that the only sequencing errors that can occur are random nucleotide replacements, but not insertions or deletions.

1 Design of a Contigger (2 points)

Suppose we are given a collection of reads R and the percent sequencing error rate. Please design a contigger, discussing the details of the following points:

- How to find overlaps between reads. Do we need to do dynamic programming under the given conditions?
- How to represent the fragments and overlaps in a graph.
- How to represent contigs.
- How to extract contigs from the graph.

2 Implementation Contigger (7 points)

The program should have the following command-line arguments:

```
-i <String> : Input file
-o <String> : Output file
-e <int>    : percent sequencing error rate
```

The input file is assumed to contain the reads in FastA format. Please write the contigs to the output file using the following format, as already implemented:

```

Contig name=c1 fragments=4 length=2030
Frag id=7 length=630 start=0 end=630
Frag id=19 length=770 start=420 end=1190
Frag id=11 length=840 start=910 end=1750
Frag id=3 length=490 start=1540 end=2030
Frag Contig name=c2 fragments=3 length=1500
Frag id=9 length=560 start=1890 end=2450
...

```

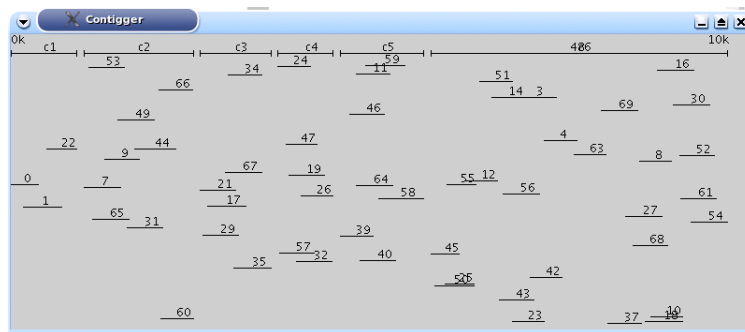
The program should operate as follows:

- First, read in the reads.
- Build the overlap graph, keeping any overlap that has length ≥ 20 and whose error rate does not exceed the specified percent error rate.
- In the overlap graph, find all contigs.
- Write the contigs to the output.
- Draw the contigs using the `Contigger.paint()` method.

Please apply your program to the following files:

- `chain8.frg`: contains 8 fragments that overlap in the order that they occur in, with perfect overlaps,
- `permuted8.frg` contains the same 8 fragments, but in permuted order,
- `sequence5x.frg` contains fragments sampled from about 10kb of sequence at 5x coverage, error rate 1%; they should assemble into a single contig,
- `sequence3x.frg` contains fragments sampled from the same target sequence as the previous, at 3x coverage, error rate 5%, they will assemble into multiple contigs.

The `Contigger.paint()` method should produce a picture like this:



with the long lines near the top representing contigs and the shorter lines floating below at random heights indicating the reads that make up the different contigs. Contigs and reads are labeled by their names.

3 Adaption of the Contigger Implementation to Work with the WGS Simulator of the Previous Assignment Sheet

Which changes would be necessary? (1 point)

Implementation of the necessary changes. (4 optional points)

Assignments due: Monday, June 25, 10 am