CS 3005: Programming in C++

DNA Sequencing Assembly

The Human Genome Project was a landmark 13 year project to map the structure of human DNA. The project made it possible for many advances in medicine. It is also useful for other fields, such as agriculture and anthropology.

The basic process used was to extract the DNA from the cells of a person. Unfortunately, the whole DNA (about 3 billion nucleotides long) can’t be extracted in one piece. So, the DNA is copied many times, and then these copies are broken up into a very large number of small pieces (each about 100 to 500 nucleotides long). The small pieces are processed in a sequencer machine capable of detecting the strand of nucleotides. The result of processing one strand then is a sequence of letters describing the type and order of nucleotides in the strand. The strands only contain four nucleotide bases: adenine (A), guanine (G), cytosine (C), and thymine (T). So, strands are represented by a text string containing the letters AGCT.

In order to recreate the order of the full DNA strand, these pieces must be stitched together in the correct order. To do this, the ends of two incomplete DNA strands are compared to see if they match when overlapped. If so, it is possible that these DNA strands are two pieces of a larger whole. The strands are matched and merged repeatedly until an approximation of the full DNA string is reproduced.

Here is an example of two DNA strands that match when overlapped:

<table>
<thead>
<tr>
<th>Strand 1: ACGGACATAGTCATT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strand 2: CATAGTCATTTCATG</td>
</tr>
<tr>
<td>Combined: ACGGACATAGTCATTTCATG</td>
</tr>
</tbody>
</table>

Assignment

In this step of the assignment, you will implement the greedy match and merge assembly algorithm of DNA sequencing. The algorithm’s input is an ordered collection of strands. The strands will be in ascending order (as determined by the less than operator). The algorithm compares all pairs of strands, looking for the largest overlap. If more than one overlap has the largest size, keep the one that has the first left strand. As in, it is first in the vector of strands. If the same left strand is tied, keep the overlap with the right hand strand that is earliest in the list. If the largest overlap is at least some minimum number of nucleotides long, then the algorithm will remove the left and right strands from the collection, and add their merged result to the collection, then sort the collection again. This greedy process continues until there is only one strand left, or, there are no overlaps that meet the minimum size requirement.

Requirements

- Your algorithm must be declared in `sequence.h` and implemented in `sequence.cpp`.
- The signature for the function is `std::vector<Strand> sequence(std::vector<Strand> strands, const size_t minimum_overlap_size);
- Note that this algorithm can, and must, use the `std::vector` class from the STL.
- Do not use `using namespace std;` in your header file.
- A unit test program must be created. This program must exercise the sequence algorithm with various input cases to assure that the output is correct.
- A Makefile must be provided for this assignment. At minimum, it must have a target `unittest` that will be sure your unit test program is built correctly, and then run `valgrind` on your unit test program.

Show Off Your Work

To receive credit for this assignment, you must upload the source code (`sequence.h`, `sequence.cpp`, `Strand.h`, `Strand.cpp`) and your unit test file(s) and the Makefile to the CIT submission system linked at the top of the page.

Additionally, the programs must build and run. In addition to your unit tests, your `sequence` function will be tested by a grading unit test. Any incorrect performance or memory errors will be counted against the
assignment score.