Project background

Project background

You can work on the project without reading the following description, but it might help you get a context for the assignment.

Your computational problem stems from the Spruce genome assembly project which aims at inferring the genome of Norway spruce ("gran", *Picea abies*), which is the tree you find all over Sweden. The spruce genome is large (about 20 Gbp long, compared to 3 Gbp for human and most vertebrates), contains 12 chromosomes of roughly equal size, and is highly repetitive. The high degree of repetition and low complexity regions makes assembly very difficult and the current ambition is therefore to primarily assemble the gene-containing regions. The genes seems to be fairly distributed over the genome.

Using several sequencing techniques and several assembly methods, we have a large set of contigs which we would like to combine. The contigs range in size from 1 kbp to 20 kbp and have a large number of overlaps. There is today no software to combine the overlapping contigs in a reasonable time (we have tried!) so we want to break up the problem in smaller pieces. Due to the low-complexity regions, there is a large number of *false* overlaps, i.e., overlaps that are due to the same subsequences appearing in many places in the genome, not because two reads were sampled from approximately the same position.

Spruce is a quite heterozygous diploid species. About 1% of positions differ in any two individuals (comparable to the difference between human and chimp), and affects the sequencing data we have. In this genome project, there is no information on the sequence level from which chromosome a contig comes from and there are therefore many contigs that have significant overlap, but with some differences because they come from two different versions of the same chromosome.

The data

You are given a graph *G* where a node represent a segment of DNA and an edge means that two segments look like they overlap and should probably be merged into one longer segment. The real chromosomes from the genome that gave rise to the graph should, in principle, be embedded in *G* and correspond to a set of paths.

Unfortunately, the overlap detection is difficult because the original DNA data is (in some places) very repetitive. The effect is that there are *many* edges that are wrong, so many that the graph has become too large for some standard genome assembly tools.

It is to some extent possible to guess where the problems are. If the data had been "easy" (here: the DNA contains no repetitions), each node in our graph should have on average about 3-5 neighbors. The number neighbors is the outcome of a random experiment, so it is not surprising with up to 10 or 20 neighbors once in a while. In the data you get, there are however some nodes with far more neighbors than that.

It should be possible to simplify *G* by throwing away suspicious nodes and/or edges! But at the same time, we don't want to throw away too many because that might mean we are throwing away data.

Genome project status

The current genome project status is that better data is being generated using new and better sequencing instruments, but we had hoped to make use of the graph that you find described in this project. Specifically, we wanted to partition the graph into manageable components and assemble the sequences in each component. For the partitioning, one would need to find a way to identify "bad" overlaps/edges and then compute connected components.