Project in Algorithms in Molecular Biology

Genome assembly typically consists of four phases:

1. sequencing error correction
2. contig assembly
3. scaffolding
4. gap filling

Many tools exist for genome assembly. Some of them are pipelines that perform all or several of the above phases, whereas some of them perform only one phase. Choose at least two tools to experiment with. For example you can choose two tools both of which perform the whole assembly pipeline and compare those or you can choose a different tool for each phase. Choose also one phase and implement your own tool for that. Integrate your own tool with the existing tools to produce assemblies for the given two data sets. Once you have assemblies of the two data sets you can run quast[1] to evaluate the assembly of the real *E. coli* genome.

PLAN

Write a short plan of how you are going to complete the project. The deadline for submitting the plan is May 10th. Submit the plan by sending it by email to the lecturer (leena.salmela@cs.helsinki.fi). The plan should include the following:

1. The tools you are planning to use in the project.
2. Which phase are you going to implement yourself? Describe briefly the algorithm you are planning to use. You can use available libraries for example to implement some advanced data structures (like de Bruijn graph).
3. A schedule for your work.

[1] http://bioinf.spbau.ru/quast

FINAL SUBMISSION

The final submission is due May 29th. Submit your project by sending it by email to the lecturer (leena.salmela@cs.helsinki.fi). The final submission consists of

1. A report including
   - Short description of the tools used
   - Short description of the algorithmic ideas used in your own tool
   - Documentation of how the tools were run (e.g. shell scripts and configuration files)
2. Source code of your own tool
3. The two assembled genomes as multi-fasta files
4. Do not submit the code of other tools than the one you implemented yourself

EVALUATION

The assembled genomes are evaluated by running quast[1]. If the reference genome is in genome.fasta and the assembly in assembly.fasta, this can be done as follows:

```
quast.py -R genome.fasta assembly.fasta
```

The statistics of the assembly can now be found in file quast_results/latest/report.txt. We will look at

- # contigs
- Total length
- N50
- misassemblies + local misassemblies
- mismatches + indels per 100 kbp

You can run the evaluation yourself for the real data set.

GRADING

The project will be graded as passed/failed. To pass the course a student should complete a project according to the requirements above. To pass there is no minimum requirement for the evaluation results of the submitted genomes.