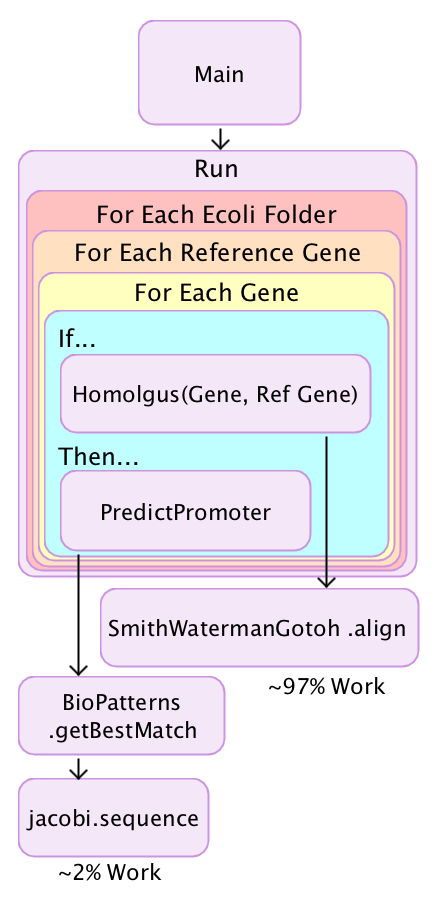
**Manual parallelization of a Sequential program**

CAB401 Semester 2 Assignment

By Ian Daniel – n5372828

**The Sequential Program - Simple Overview**



The program searches for Sigma70 in E. coli genes. Sigma 70 is a housekeeping sigma factor also known as the primary sigma factor. This Sigma keeps essential genes and pathways operating. The algorithm in use is called the Smith-Waterman algorithm and it has the Gotoh’s improvement. It is implemented in Java by Ahmed Moustafa.

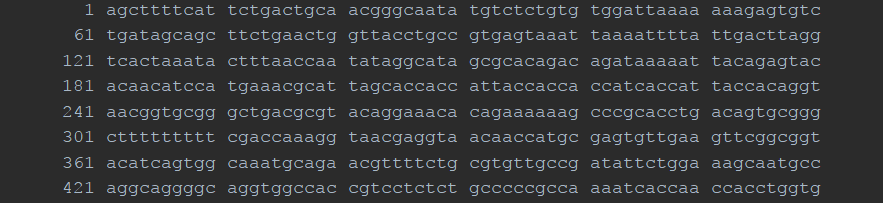
The general structure of the program is simple. The Java program starts with the Main method in class Sequential and this does is call the Run method in the same class with the path to “referenceGenes.list” and the “Ecoli” folders.

In the Run method, there is 3 nested for loops. The first two are simply preparing for the inner for loop by loading the Ecoli Genes and Reference Genes. The inner for loop compares these two and if a match is found it will then search for the Predict the best match between the two.

The bulk of the work in this program, approximately 97% of it, happens in the Homologous method which calls the SmithWatermanGotoh.align method. As seen in the above flow diagram a small amount of work also happens in while finding the best match.

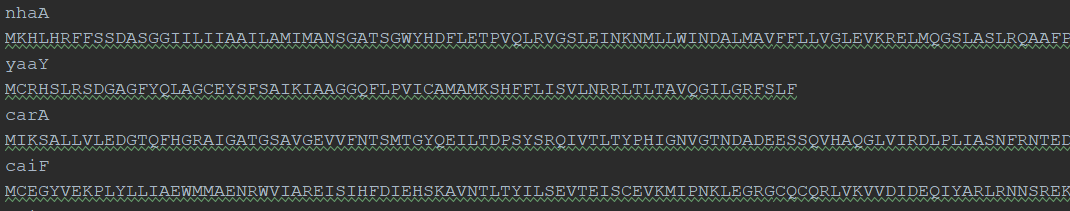
**The Data**

The data consists of two parts; the Genes and the Reference Genes. The Genes come from the four folders in the Ecoli folder. Each of these folders contains a GBK file which has reference material and the gene data itself. An example of the gene data can be seen below.



There is around 77000 lines in each file, these is parsed to peptide sequences and saved the in GenbankRecord object called record.

The reference genes are stored in the single file referenceGenes.list. This list contains the eight different genes such as nhaA, carA and folA. Here is an example of the reference genes.



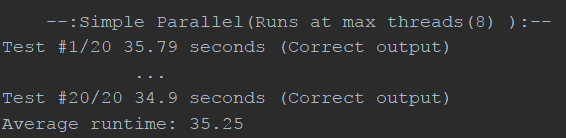
These are parsed into a List of type Gene called referenceGenes and is iterated of in the second for loop and then compared in the Homologous if statement in the inner for loop.

**Hardware and Software**

The hardware used to run the experiments for the various implementations is a Xeon E5-1630 v3 that runs at 3.7 GHz. The system has 32GB Ram, an Nvidia GTX980 GPU and a Samsung SSD SM871 512GB SATA3 540 read/ 500 write. This is running on Windows 10 version 1709 and the IDE is IntelliJ IDEA 2018. The CPU profiler is JProfiler, this program shows how many threads are being used, CPU utilization and which part of the program does all of the work.

**Tests**

Each class is tested twenty times and the time taken is averaged for the final result. Each test’s output is also compared to the unedited original to confirm it is correctly working.



The right shows a truncated example of the twenty tests and the resultant average time. Later when describing the results of the parallel implementation I will also the minimum, maximum and the standard deviation.