Midterm Report

Louis S. Revor[1]
Hardin-Simmons University
Abilene, Texas

supervisor :
Dr. Michael Minkoff
Computing and Telecommunications Division
Argonne National Laboratory

18 October 1991

[1]Participant in the Fall 1991 Student Research Participation Program at Argonne National Laboratory. This program is coordinated by the Division of Educational Programs.

Since arriving at Argonne, I have been working with Dr. Michael Minkoff (CTD) and Dr. Radoje Drmanac (BIM) on the simulation of partial sequencing with respect to the human genome. This project is headed by Dr. Drmanac and Dr. Radomir Crkvenjakov (BIM). My side of the project has been the writing of FORTRAN code to test and implement an algorithm for the simulation as directed by the biologists (Drmanac and Crkvenjakov).

The first step in the process of writing the algorithm was the development of code on both sequential and parallel computers for the random creation of sample data for testing the algorithm. This sample data consists of a random genome (sequence), fragments of the genome with random overlap, and probes - all possible combinations of a genomic sequence of a minimal length. All of the data (sequence, fragments, and probes) are of specified length. Once the sequence, fragments, and probes were generated, a table was generated to show which probes were contained in each of the fragments. The only data that is needed for the algorithm is the table, but to show the integrity of the table, the sequence, fragments, and probes were all given as output so the table could be checked manually.

Two different methods were implemented for generating the table. The first method was to see if each probe was in each of the fragments by pattern matching. This became quite a time consuming process as the length of the data increased. Therefore, a new method was implemented. The new method used the fact that the probes were every combination of the letters 'A', 'T', 'C', and 'G'. Since probes consisted of only these four letters, then they could be considered to be a base-four number. This fact was used to correlate the probe to an index in the table. Now we could scan each fragment, looking at each probe length segment, and mark the result in the appropriate table. This dramatically decreased the time, because it no longer involved pattern matching, and that the loop only executed as many times as positive results - instead of all possible results.

This code was run originally on a Sun Workstation, but the time was still a hindrance. The code was then transported to Argonne's Advanced Computing Research Facility's Alliant FX/8 parallel computer. The code was then run without any changes in sequential fashion on the Alliant. A significant decrease in time was noted. The code was then modified for parallel execution and run on the Alliant using 1, 2, 4, and 8 concurrent processors. It was run on a various number of processors to produce a speed up chart which shows the effectiveness of the parallelization of the code. I ran the code using various lengths of the segment, fragments, and probes to show the effect of the length on the time of execution. Once this was done,

I could move on to the first step in the development of code to do the actual reconstruction.

In the reconstruction of the genome, one must look at all of the probes that are contained in a given fragment, then overlap the probes one at a time with maximum overlap. The overlap that we are looking for at first is the overlap of $n-1$ character of the probe, where the probes are n characters long. A branch is possible during each extension, because for every probe there are four probes that match with the overlap as stated above. However, since all four of these probes are not necessarily contained in the fragment, then 1, 2, 3, or 4 branch(es) are possible at every extension. No matches to the current probe is also possible, in which the end of the fragment has been reached. Since there are possible branches, and all must be considered, a tree like structure begins to emerge.

My first task in this reconstruction process was to take a given set of sample data (generated to minimize the processing time), and obtain data on the number of branching points that occur, and the number of distinct results that occur because of this branching. I looked at how the number of extensions affected these numbers and found that as the number of extensions that occur increase, the number of distinct results increase in a basic exponential fashion. This information is imperative, because it shows the biologists (and myself) that the number of possibilities that will occur in the actual problem will be great. It raises the question if their algorithm is efficient enough to be implemented. It is obvious that if the length of the probes is increased, then these numbers would decrease significantly, but since the number of probes is 4^n, then the amount of data needed is significantly increased as the length of the probe is increased.

As of now, this is as far as the project has come. I consider it to be a considerable distance, since it was a new project when I arrived. As we have more meetings with the biologists, the direction this project will take will come into clearer focus. As of right now, we are not sure what direction will take, but wherever it goes it should be interesting.