Searching for Non-Existent Needles in a DNA Haystack

Ben Noland, Dr. Amit Jain (Computer Science), Dr. Tim Anderson (Computer Science), and Dr. Greg Hampikian (Biology)

Boise State University

College of Engineering

Computer Science

Abstract

In an effort to understand the limitations placed on DNA by evolutionary pressure, we have developed a parallel program that searches DNA for non-existent sequences. We have run this on the human genome, which is about 2.87 billion characters long, and have identified 86 length 11 sequences that do not exist. If the human genome was random (no evolutionary pressure), then the probability of this happening is less than 10 to the power of -311.

The program is being run on the Boise State Beowulf Cluster with 122 processors. Using the cluster allowed us to run the program about 40 times faster than would be possible on a single workstation.

The Question

We are looking for a way to find the shortest sequences that do not occur anywhere in a given genome. By a sequence we mean a set of adjacent bases in the order they appear within the genome.

For a given length N, there are 4^N possible sequences of that length.

4^{10}=1,048,576
4^{11}=4,194,304

We were hoping to look at sequences of length 16.
4^{16}=4,294,967,296

The Basic Algorithm

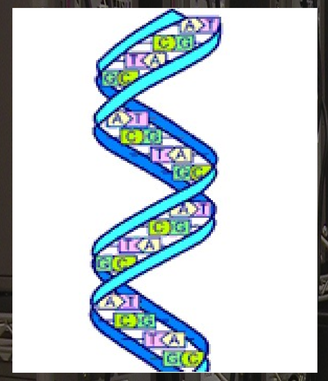
1) Each node of the cluster creates a huge list with one entry for every possible sequences of a given length.
2) Each node reads through a different portion of the genome, keeping track of how many times each sequence has been seen.
3) The nodes all send their data to the master node where they are combined into a master list.
4) Look through the list for sequences that happened 0 times.

Problems with this approach.

Memory(RAM) usage is directly correlated to to size of the list. The size of the list is exponential with respect to the length of sequences. This means that memory usage is exponential.

We can not do sequences longer than 13 due to memory limitations.

The Basics of DNA



DNA is made up of 4 different bases which are abbreviated A, C, T, G.

A small portion of a DNA strand makes up the genome, which is what we are studying.

The human genome has about 2.87 billion base pairs.

We have downloaded the genomes of many different plant and animal species onto the cluster.

Each genome is stored in a set of large text files that contain a long string of A, C, T and G.

As an example of the file sizes we are dealing with, the human genome takes up about 2.87 gigabytes of disk space.

An Extended Algorithm

The key idea here is that each node of the cluster looks for a different range of sequences. For example the first node only looks for sequences that start with AAA, node 2 only looks for sequences that start with AAC etc.

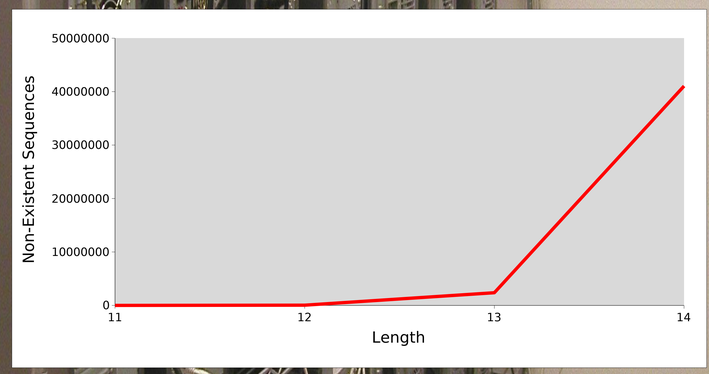
1) Each node creates a list that can hold all possible length 13 sequences
2) Each node reads through the entire genome, but only looks for sequences that start with a certain prefix. It then cuts off the prefix and keeps track of the rest.
3) Each node looks through its list for sequences that didn't occur.

This approach has allowed us to reach our goal of looking at length 16 sequences.

The Cluster

Cluster Configuration
- 61 nodes
- 122 2.4 GHz Intel Xeon CPUs
- 64GB memory
- 3.4TB disk space
- private Gigabit network
- Gigabit connection to campus backbone
- Operating System: Linux
- Parallel Software: MPI, PVM

Results



We have successfully created a program that finds all non-existent sequences in a given genome, and have gathered data on the human for up to length 16.

In the chart above you can see the pattern of sequences that are missing from the human genome for lengths 11-14.

The Future

So far we have only run our program on the human genome. We plan to run it on many more species so we can compare the data.

We are currently in the process of developing a website that will enable the users to compare data from different groups.

We also plan to run the program on all the species we have collected at one time. This will give us an idea of sequences that do not occur in any of the species.