

# CS 2427 - Algorithms in Molecular Biology

## Lecture #2: 13 January 2006

Lecturer: Michael Brudno

Scribe Notes by: Alex Hertel

### 1 Today's Topics:

- Sequencing The Human Genome
- BAC by BAC Technique
- Celera's Technique
- The Role of Graph Theory

### 2 Sequencing The Human Genome

The sequencing of the human genome, also known as the Human Genome Project, is one of the most ambitious and impressive research projects ever attempted. It was completed several times, lastly in 2003, though it may be completed again.

Our genome consists of approximately  $3 \cdot 10^9$  base pairs (b.p.) and is divided into 23 pairs of chromosomes. Each human chromosome is linear; that is to say, each chromosome has a distinct start and end point. This is in contrast with the chromosomes of bacteria, which are circular. Also bacteria typically have only one chromosome.

DNA is a massive molecule which is not only coiled in the well-known double-helix structure, but this structure is further folded over and folded upon itself. A testament to how compressed it is within the nucleus of a cell is the fact that if it were not wrapped, our genome would be about 4 meters long – but instead fits in a micron-sized cell nucleus.

### 3 BAC by BAC Technique

Since the human genome is so long, it clearly cannot be sequenced manually; the entire process must be highly automated for us to stand any chance of success. Unfortunately, our automated methods for taking a strip of DNA and sequencing it can only reliably handle strands that contain about 500 b.p.. In other words, our methods cannot come even remotely close to sequencing one whole chromosome, let alone the entire genome. We therefore are forced to cut up the DNA into smaller pieces and sequence them.

The standard technique for doing this previously was called the BAC by BAC method (BAC stands for Bacterial Artificial Chromosome). This name comes from the fact that during this method, we use e. coli bacteria to perfectly duplicate strands of human DNA by inserting those strands into the circular e. coli genome and letting it divide.

The BAC by BAC technique works as follows:

1. Make many copies of the human genome to be sequenced.
2. Take a number of copies of the human genome, and split them up into much smaller strands of DNA, each consisting of about 150000 b.p.. Each of these strands is called a BAC. The overall idea is that we will sequence the BACs, and then use the fact that they overlap to put all of those sequences together (in the obvious way), and thereby obtain a sequencing of the entire human genome. We therefore must make sure that all of the BACs together not only contain the entire human genome, but that the overlaps are sufficiently large for us to be able to reliably put them together again. Figure 1 below shows an example of how the genome can be ‘tiled’ using many BACs.

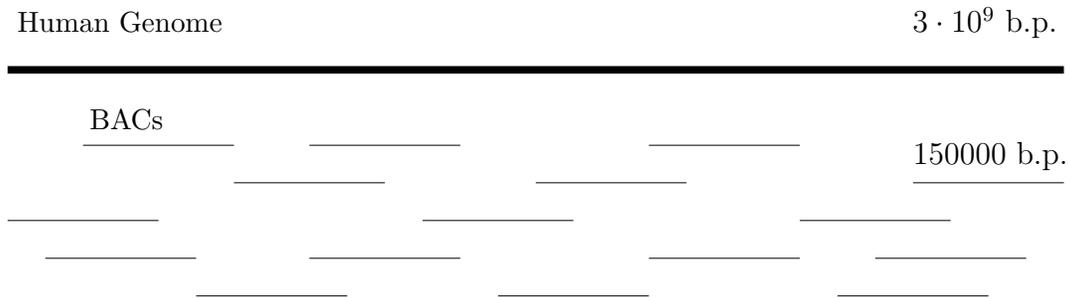


Figure 1: Many overlapping BAC strands compared to the entire genome

3. Once we have the BACs, we have to find out where they came from in the genome, and then choose the smallest set of BACs that completely covers the entire genome. This process is very expensive in human time.
4. The next step is to sequence all of these BACs. It is easy to see that if we correctly sequence the BACs, then that will give us a correct sequencing of the entire human genome. Sequencing the BACs is done analogously to sequencing the genome: we take each BAC, make copies of it, and then randomly cut them up into small pieces which are approximately 4000 b.p. long. We take these and use our automated sequencing technology to sequence approximately 500 b.p. at each end, as shown below in Figure 2. Each of these 500 b.p. sequences is called a ‘read’, and it tells us not only what is on one strand of the DNA, but by complementary base pairing, it immediately gives us the other strand as well. For the smaller 4000 b.p. pieces, the gap is 3000 b.p. long. It is also normal to use longer 40000 b.p. pieces with a much larger gap. One important note is that each read is done from the 5’ to the 3’ end, so the two reads from each fragment are from opposite strands, and opposite ends of the fragment. With

enough overlapping reads, it becomes possible to put them together in order to sequence the BAC.

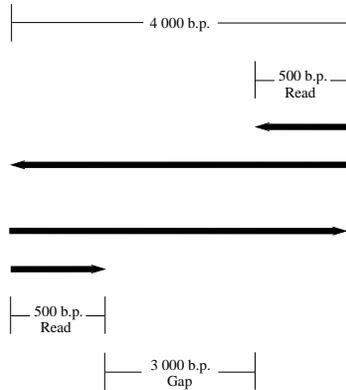


Figure 2: Each DNA fragment consists of two strands, each of which is read from the opposite direction.

## 4 Celera’s Technique

Celera Genomics, a private company, and its founder Craig Venter developed a technique which can be automated to a greater degree than the traditional method. Celera’s idea was to cut out the intermediate step of having to sequence the BACs; instead, they split the original DNA sequence into 40,000 and 4000 b.p segments directly (these are called cosmids and plasmids, respectively), and from there to read them and then assemble the reads for the whole genome. Since the splitting of the genome into the shorter pieces is relatively random, ensuring that the entire genome gets covered requires the sequencing of base pairs approximately ten times the size of the genome, or  $3 \cdot 10^{10}$  b.p.. This translates to approximately  $6 \cdot 10^7$  reads.

This entire process is called ‘whole genome shotgun sequencing’, since the process of fragmenting DNA strands is reminiscent of the tiny shot pellets coming out of a shotgun. With a sufficient number of reads, it is not hard to see that the entire problem of sequencing the human genome has been reduced to the combinatorial problem of amalgamating semi-overlapping strings correctly.

## 5 The Role of Graph Theory

It turns out that the problem of putting all of the reads back together to either form a BAC or the human genome itself can be expressed as a problem in graph theory. Simply take every individual read, and create a node corresponding to its sequence. For any two nodes / reads that have an overlapping segment, create a directed edge from the first node to the second one. It is not hard to see that a Hamiltonian Path (a simple path containing each node exactly once) in the graph will yield a correct amalgamation of the genome fragments, and therefore give us a correct sequencing of the BAC / genome. We will call the graph with reads as nodes and overlaps as edges “string graphs”.

### Bidirected Graphs

There is a problem, however: although each of the two reads from a fragment is done in the 5' to 3' direction, this tells us nothing about which way the fragment was originally oriented in the genome. In other words, the problem of putting all of the pieces together is complicated by the fact that we don't know which strand each read came from; in fact, exactly half of the reads are from one strand, and half are from the other, so we can picture half of the reads going from left to right, and the other half going from right to left. We shall refer to a read going from left to right as a 'right read', and one going from right to left as a 'left read'. This means that there are four possible ways for reads to overlap: R-R, R-L, L-R, and L-L.

In order to model this properly, we use a generalized notion of a string graph called a 'bidirected string graph'. A bidirected graph is similar to a directed graph except that instead of just having one type of directed edge, bidirected graphs have four types of edges. The four different types of reads as well as their corresponding bidirected edges are shown below in Figure 3. It is worth noting that the first and last edges are identical, and are analogous to a standard directed edge.

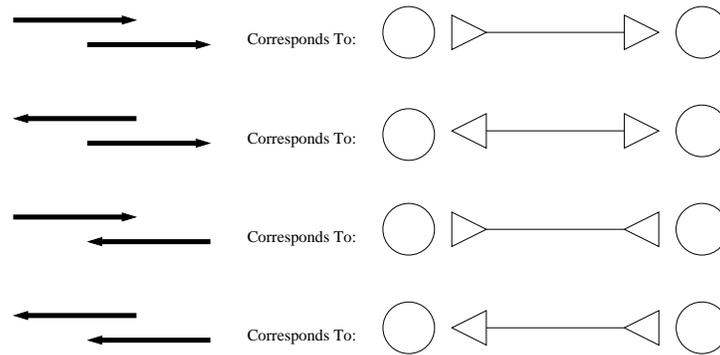


Figure 3: The four different ways reads can overlap with their corresponding bidirected edges; the first and last edges are indistinguishable

The endpoint of each edge in a bidirected string graph can be thought of as either entering or leaving a node. The Hamiltonian Path problem for bidirected graphs is similar to the standard Hamiltonian Path problem with two particularities that are worth mentioning: firstly, one does not have to follow the directions of the edges. For example, one can enter a node through an edge endpoint that is leaving it, and one can leave a node through an edge endpoint that is entering it. The second particularity is that for each vertex, if you enter a node via an in-arrow, then you must leave it via an out-arrow, and vice versa. Using these rules, some Hamiltonian Path in the bidirected string graph (there may be multiple paths) will correspond to a valid assembly of the genome the reads of which were used to build it.

**Example**

Figure 4 below shows an example of a set of reads that form a cycle. To the right we have the corresponding bidirected graph. It is not hard to see that  $g, h, c, d, e, f, b, a$  is a legitimate Hamiltonian path in the graph.

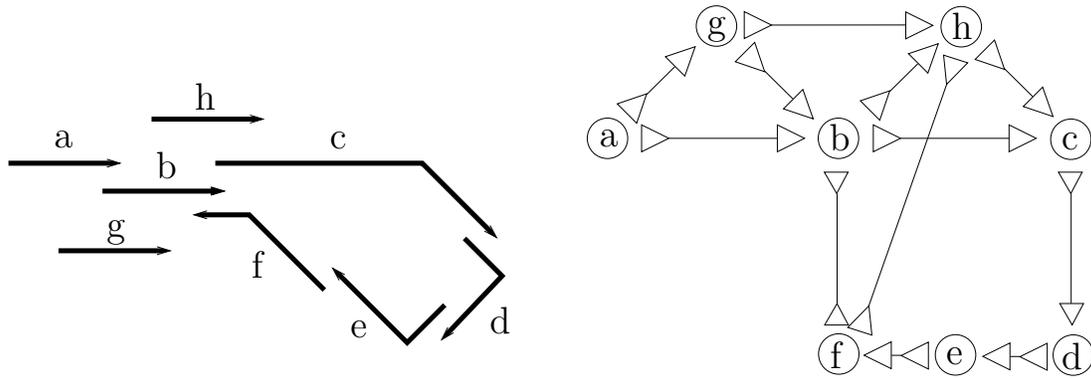


Figure 4: An example of a circular set of overlapping reads and its corresponding bidirected graph

### Transitive Edge Removal

If you now consider nodes  $a$ ,  $b$ ,  $c$ ,  $g$  and  $h$  in our original string graph you will realize that they all actually come from the same sequence, and if we were to allow for every node to be used more than once then can be represented by a single path. Removing such unneeded edges simplifies the problem of finding a Hamiltonian Path in our graph, so we want to prune any and all edges possible. The pruning step is called ‘transitive edge removal’. Consider three nodes  $a$ ,  $b$ , and  $c$ , as shown below.

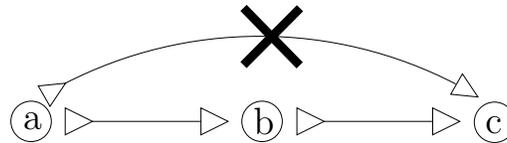


Figure 5: Transitive Edge Removal

The idea is that if there is an edge from  $a$  to  $b$ , from  $b$  to  $c$ , and from  $a$  to  $c$ , then we can delete the edge from  $a$  to  $c$ , since any Hamiltonian Path that uses this edge can use  $a$  to  $b$  and  $b$  to  $c$ , as long as we allow every edge to be used more than once. We therefore formulate the Generalized Hamiltonian Path problem in bidirected graphs as the path that goes through every node, but is allowed to visit nodes more than once.

Figure 6 below shows the previous bidirected graph after having its transitive edges removed.

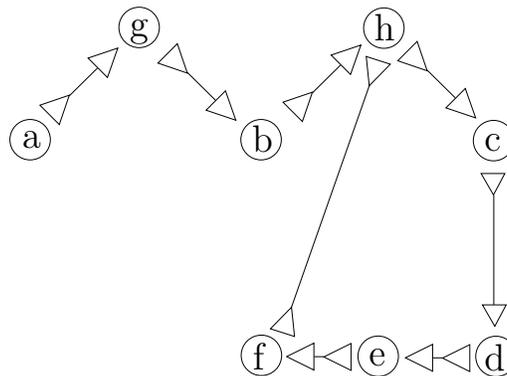


Figure 6: Transitive Edge Removal