Int. J. Math. And Appl., 9(2)(2021), 81–90
ISSN: 2347-1557
Available Online: http://ijmaa.in/

International Journal of *Mathematics And its Applications*

Genome Sequencing with Graph Elements

Arnav Gattu[1,*]

1 Newark Memorial High School, Newark, California, United States.

Abstract: DNA sequencing is the process of ordering the nucleotides in a piece of DNA. Each of the four existing nucleotides is identified with one of the four letters A, C, G, T. Sequencing a genome is quite challenging. It requires breaking the DNA of the genome into many smaller pieces, aligning and merging the pieces, and reconstructing them into a one long genome in proper order. Since the completion of the human genome project, technological improvements have accelerated the speed of genome sequencing and made it less expensive. This study presents a method of reconstructing the genome from its smaller sequences or reads using algorithms based on graph theory. Reconstructing a genome is an important step in understanding the pattern of letters of the genome sequence, functions of genes, relation between genes, and how they all work together. This understanding of the genome sequence of different species, including plants leads to many applications, medicines, and vaccines as seen in the current Covid-19 pandemic.

Keywords: DNA sequencing, Covid-19, Graph Elements.

1. Introduction

"The discovery of the structure of DNA transformed biology profoundly, catalyzing the sequencing of the human genome and engendering a new view of biology as an information science" [1]. Deoxyribonucleic acid commonly known as DNA is a molecule that contains all the genetic information required to build and maintain an organism. It consists of two strands that wind around one another forming a double helix shape. Each strand is made of deoxyribose and phosphate groups. Attached to each sugar is one of the bases Adenine (A), Cytosine (C), Guanine (G), and Thymine (T). The bases link across the two strands in a certain manner using hydrogen bonds: cytosine pairs with guanine and adenine pairs with thymine. A human genome contains approximately 3.2 billion nucleotides and about 23,500 genes [2]. Sequencing an entire genome is a challenging task, as it requires breaking the genome into smaller pieces or segments, then sequence the pieces, and reconstruct the entire genome in the proper order.

DNA sequencing efforts were pioneered by Walter Gilbert [3], and Fred Sanger [4] in the 1970s. In the later years and decades these initial methods were refined and sped with technical advances. The Human Genome Project (HGP) automation efforts accelerated the processing time and reduced the cost drastically to sequence an entire human genome. The publicly funded effort took about 13 years to complete and spawned an entire new industry of bioinformatics, and a new era of medicine with significant advances in technology [5]. The cost and processing times have improved a long way from sequencing few base pairs a day to sequencing an entire human genome in a day, and from a cost of millions of dollars to just $100 for a genome [6]. Genome sequencing techniques have evolved from Sanger sequencing, to varied shotgun sequencing methods, to

* *E-mail: arnavgattu@gmail.com*

next generation sequencing (NGS) or also known as massively parallel or deep sequencing technology [7]. Innovation and increasing computational techniques have and will lead to ultra high throughput, speed, and scalability.

DNA sequencing involves multiple complex steps and may vary from platform to platform. In this study, our objective is to learn the sequence of nucleotides in a whole genome, where the sequence is a finite list of nucleotides or letters. Sequence assembly is accomplished by aligning and merging segments from a longer genome sequence to reconstruct the original sequence. In this paper, I have shown an algorithm based on de Bruijn graphs [8] and Sanger sequencing [4], where given an arbitrary order of k-mers from a long genome sequence, a genome can be created by finding its Eularian path. The challenge is always of scaling, K-mer lengths can run into double digits or higher, as the k-mer length increases the complexity of genome reconstruction increases and becomes tedious. The algorithm presented here is simple to construct and scales.

2. Results

In this study to reconstruct a genome, I will describe an algorithm based on elements of graph theory, where given an input list of k-mers, the algorithm will output a genome based on the input list of k-mers.

Definition 2.1. *A directed graph is a set of objects called vertices or nodes which are connected together with edges, where all edges are directed from one node to another. The tail of an edge connects a node to another node with its head.*

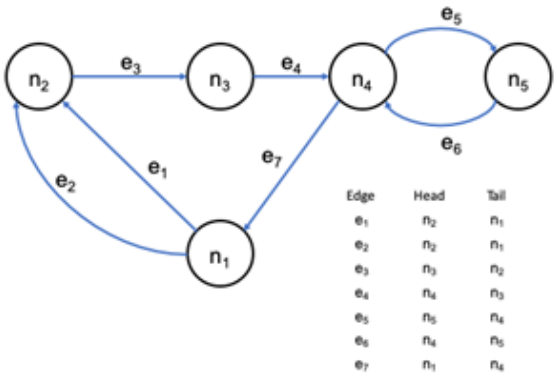

Figure 1: Directed graph

Definition 2.2. *A path is a sequence of different edges $e_1, e_2, e_3, \ldots, e_n$ where the head node of e_{n-1} equals the tail node of e_n and the edge is not repeated.*

Example 2.3. *In figure 1, the sequence $e_1, e_3, e_4, e_5, e_6, e_7$ is a path. On the other hand, the sequence e_1, e_3, e_4, e_7, e_1 is not a path because the edge e_1 is visited twice. Similarly, e_1, e_3, e_4, e_6 is not a path because the head of edge e_4 is not the tail of edge e_6.*

Given a list of k-mers a graph with edges can be created with the following approach.

1. For each k-mer create an edge and label the edge same as as the k-mer; as the k-mers may repeat in a list so edges may also repeat.

2. Create a set S of different words that are prefixes and suffixes of the k-mer.

3. Consider each element of S to be a node.

4. Tail of each edge is the node whose label matches the prefix label of the edge, and the head of each edge is the node whose label matches the suffix of the edge label.

Example 2.4. *List of K-mers ["CTT", "TTA", "TAT", "ATC", "TCA", "CAT", "ATA", "TAT", "ATT", "TTT", "TTG", "TGT", "GTA"]*

The set S of prefixes and suffixes from the above k-mer list are: CT, TT, TA, AT, TC, CA, TG, GT.

The graph can be represented with nodes and edges as illustrated below, where the 2 letters represent the nodes and the arrow represents the edges connecting the two nodes.

$CT \rightarrow TT$ *(edge connecting node CT to TT, tail is at CT and head at TT)*

$TT \rightarrow TA, TT, TG$

$TA \rightarrow AT, AT$ *(2 edges connecting node TA to node AT, so it appears twice)*

$AT \rightarrow TC, TA, TT$

$TC \rightarrow CA$

$CA \rightarrow AT$

$TG \rightarrow GT$

$GT \rightarrow TA$

In a few cases the nodes and edges repeat themselves as in the case of $TA \rightarrow AT$, AT, as these represent k-mers of the same letters, which appear twice in the k-mers list.

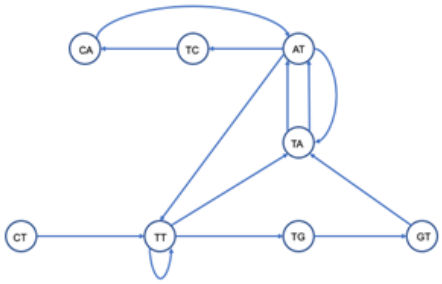

Figure 2: Graph with nodes and edges

In the above graph the edges are not labeled as these can be procured by putting the labels of tail and head together or which are the prefix and suffix of a k-mer. In the graph there is a loop, and there are multiple edges from one node to another.

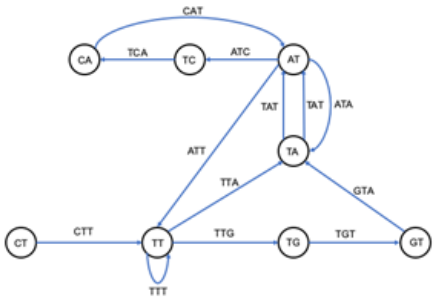

Figure 3: Graph with nodes and labeled edges

Definition 2.5. *A Eulerian path is a path which contains all the edges in the graph.*

From the above two graphs following observation are made:

1. The path $e_1, e_2, e_3, \ldots, e_n$ forms a eulerian path where it contains all the edges of the graph.
2. The graph results in multiple paths leading to reconstruction of different genomes, including the original genome, as illustrated in Figure 4.
3. The two different paths are:
   (a). $e_1, e_2, e_3, ..., e_{13}$ this results in the genome CTTGTATCATTTATA which was of interest.
   (b). $E_1, E_2, E_3, ..., E_{13}$ this results in genome CTTATCATTTGTATA.
4. Order of k-mers does not affect the outcome.
5. Even though we are able to reconstruct multiple genomes, the approach is successful as we are able to construct one or more genomes from the same list of k-mers, including the original genome.

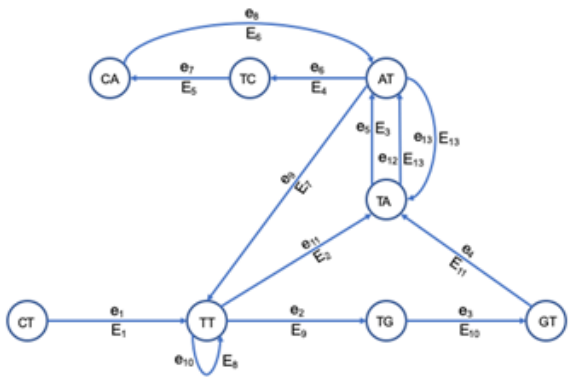

Figure 4: Graphs with multiple paths

Definition 2.6. *Define n to be a node in a directed graph, where inDegree of n, inDeg(n), is the number of edges $e_1, e_2, e_3, ..., e_n$ having n as its head, and outDegree of n, outDeg(n), is the number of edges which have n as its tail.*

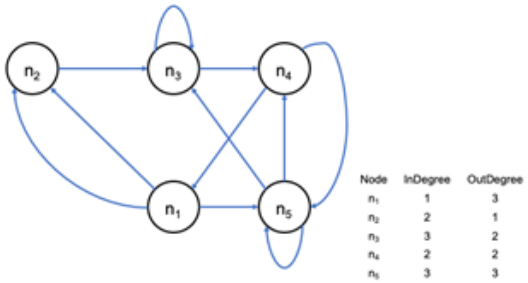

Figure 5: Example of inDegree and outDegree count at each node

Definition 2.7. *In a connected graph for every pair of nodes n_1 and n_2 there is a path that starts with an edge that has node n_1 as tail and ends with an edge that has node n_2 as head.*

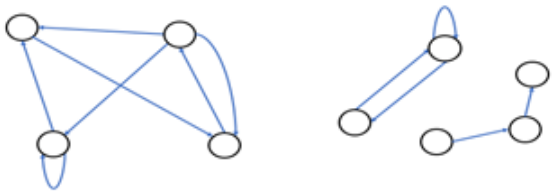

Figure 6: Examples of connected and disconnected graphs

Definition 2.8. *In a connected graph, a Eulerian path $e_1, e_2, e_3, ..., e_n$ is a cycle where the tail of e_1 is equal to the head of e_n.*

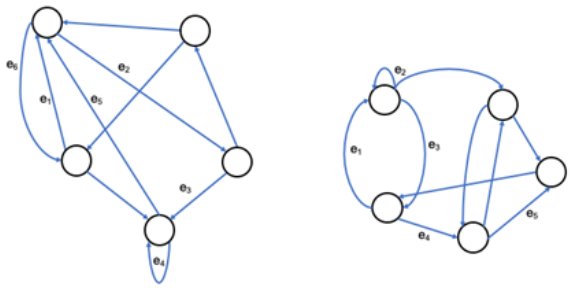

Figure 7: Graph on left is a cycle, where the path $e_1, e_2, e_3, e_4, e_5, e_6$ is a cycle; while the graph on the right is not a cycle e_1, e_2, e_3, e_4, e_5 is not a cycle

Theorem 2.9.

(1). Assume G is a connected graph with nodes N and edges E.

(2). Graph G has an Eulerian cycle, and a Eulerian path that is a cycle, only when $inDeg(n) = outDeg(n)$.

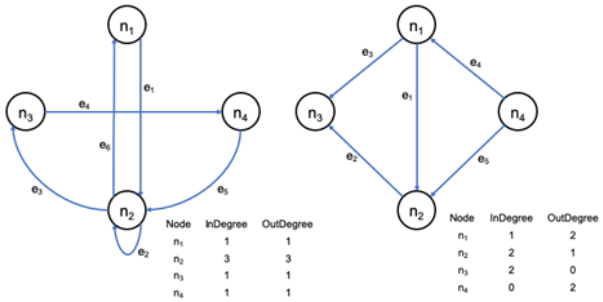

Figure 8: Graph on left is where $inDeg(n) = outDeg(n)$ for every node, it also shows the Eulerian cycle. The path $e_1, e_2, e_3, e_4, e_5, e_6$ is an Eulerian cycle; while the graph on the right is not an Eulerian cycle

Theorem 2.10.

(1). Assume directed and connected graph G with nodes N and edges E.

(2). G will have a Eulerian path that is not a cycle when there exists:

(a). a node x where $outdeg(x) - indeg(x) = 1$ and

(b). a node y where $indeg(y) - outdeg(y) = 1$ and

(c). for rest of the other nodes n, $indeg(n) = outdeg(n)$

(d). here, any Eulerian path starts with an edge having node x as its tail and ends with an edge having node y as its head.

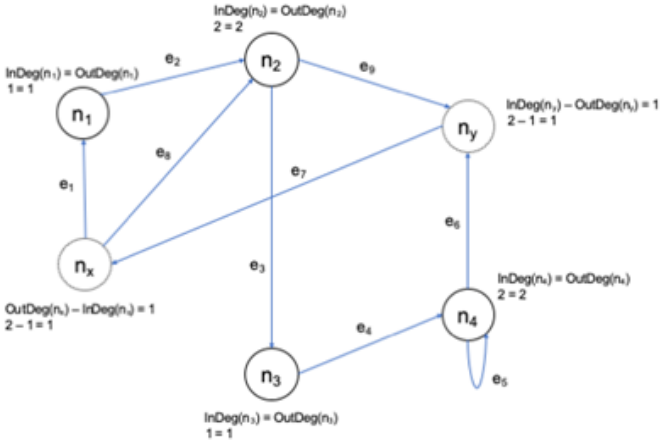

Figure 9: A graph satisfying conditions for a Eulerian path to exist. The order of edges in the graph is the Eulerian path

3. Methods

Algorithm to find Eulerian cycle in a graph

Let's assume that a Eulerian cycle exists in a graph. Now, let's describe an algorithm to find this Eulerian cycle.

1. Pick any edge and call it e_1 as shown in the figure below

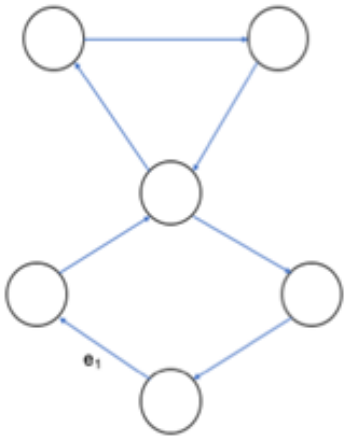

Figure 10: A graph with edge e_1

2. Create a path until it is no longer possible to move ahead as shown in the figure below.

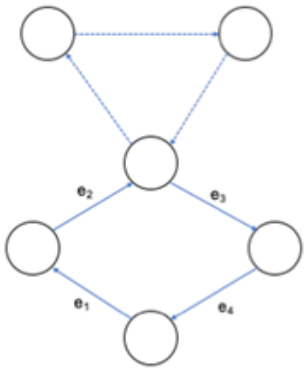

Figure 11: A graph with a dead end path

3. If we had visited all the edges by the time we had to stop because we could not continue, we would be done. But many times, as in above example, this is not the case. The cycle path e_1, e_2, e_3, e_4 does not contain all the edges of the graph above. Nevertheless, since the graph is connected, there should be at least one node in the cycle that is the tail of an edge that is not in the cycle. The next step is to go around the cycle and stop at one of these nodes. In the figure this is shown as filled node with the tail of the dashed edge. This filled node should be the first visited node.

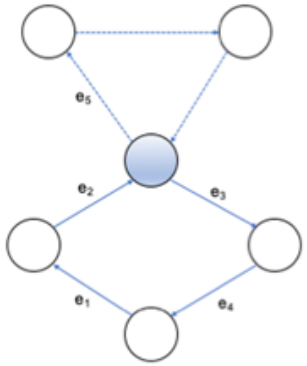

Figure 12: Creating a new edge

4. Re trace the cycle again, but starting at the filled node, and change the label of the edges accordingly.

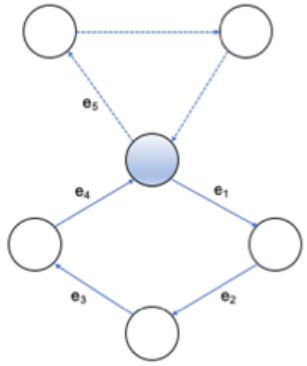

Figure 13: Retracing the edges with ordered labels

5. Because we always end at the same node we start when tracing a node, we end at the filled node. But the filled node was selected because it has an edge coming out that is not in the cycle. That means that we do not need to stop, we can continue and end with a cycle that contains all the dashed edges of the cycle and some new edges. This is illustrated below. In the case below, we ended up with an Eulerian cycle and we are done. This is not always the case, but we just have to continue repeating this strategy and we eventually do end with an Eulerian cycle.

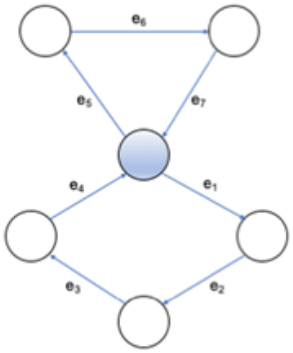

Figure 14: A graph traced with Eulerian cycle

Algorithm to find a Eulerian path in the graph

Our objective was to find a Eulerian path and not a Eulerian cycle. To reach this objective, let's follow these steps:

1. Find a node y where $inDeg(y) - outDeg(y) = 1$.
2. Find another node x where $inDeg(x) - outDeg(x) = -1$.
3. Ensure for rest of the nodes $inDeg(n) = outDeg(n)$ or the difference is zero
4. Add an edge with node y as tail and node x as head.
5. Find a Eulerian cycle in this new graph
6. Retrace the cycle starting with the edge that comes after the added edge and just stop before the turn of this new edge. This path is an Eulerian path.

The steps are illustrated below:

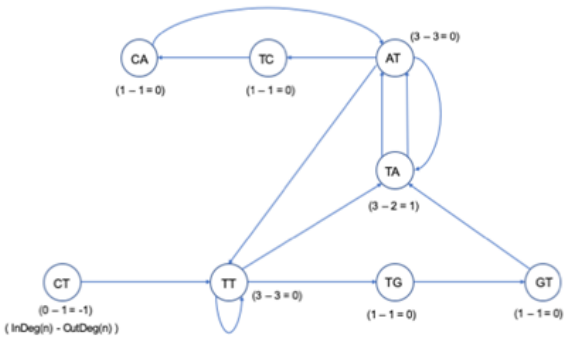

Figure 15: A graph with nodes and edges, and nodes with inDegree and outDegree difference

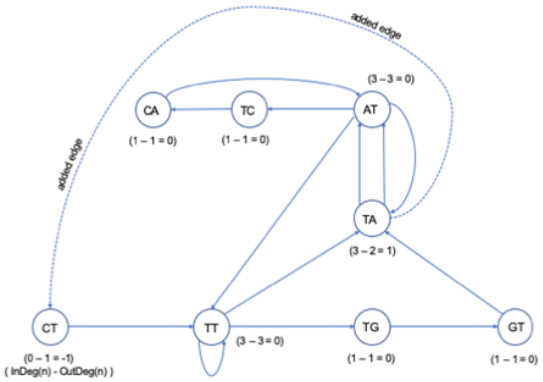

Figure 16: A graph with added edge, shown as dashed line

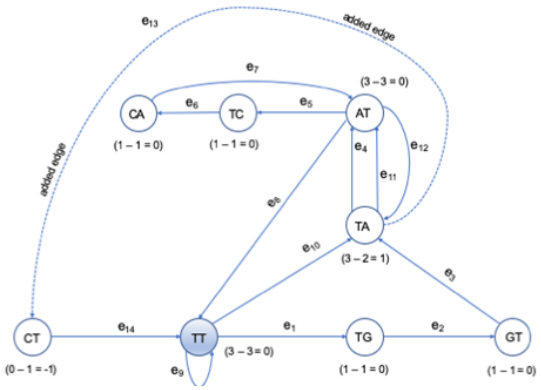

Figure 17: Finding Eulerian cycle in the graph

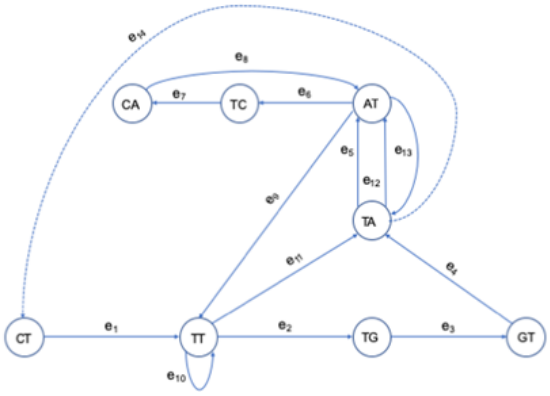

Figure 18: Retracing the Eulerian cycle to get the Eulerian path starting with edge e_1

Finding the Genome

Following the above algorithm and tracing the edges starting from e_1 will lead to the genome: CTTGTATCATTTATA

4. Conclusion

In the field of bioinformatics, the challenge of DNA sequencing is a race to find cheaper, accurate, and faster techniques. As the number of k-mers grow exponentially along with the size of the k-mers, it becomes computationally challenging to

build an accurate genome sequence. Genome reconstruction is a challenging task. After genome sequencing is complete, the segments or fragments of reads have to be assembled back into one long genome sequence. There's no particular order in which these reads and k-mers are sourced. While doing genome sequencing, it is imperative to maintain a low error rate. In this study I have provided an output genome which is sourced from a given list of k-mers. In the algorithm I have shown, given an arbitrary order of k-mers, a genome can be reconstructed by finding a Eulerian path. As the data challenges with genome sequencing continue to grow, algorithms like these will help in faster computation and allow us to keep the cost down.

Acknowledgements

I want to thank my mentor Prof. Guillermo Goldsztein, GeorgiaTech, for his guidance and patience in answering my questions and concepts, and his encouragement to continue this study.

References

[1] L. Hood and D. Galas, *The digital code of DNA*, Nature, 421(2003), 444-448.

[2] A. J. Marian, *Sequencing Your Genome: What Does It Mean?*, Methodist Debakey Cardiovascular Journal, 10(1)(2014), 3-6.

[3] A. M. Maxam and W. Gilbert, *A new method of sequencing DNA*, Proc. Natl Acad. Sci. USA, 74(1977), 560-564.

[4] F.Sanger and A. R.Coulson, *A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase*, J. Mol. Biol., 94(1975), 444-448.

[5] H. Chial, *DNA sequencing technologies key to the Human Genome Project*, Nature Education, 1(1)(2008), 219.

[6] A. Regalado, *China's BGI says it can sequence a genome for just \ $100*, MIT Technology Review, 26(2020).

[7] S. Behjati and P. S. Tarpey, *What is next generation sequencing?*, Arch Dis Child Educ Pract Ed., 98(6)(2013), 236-238.

[8] N. G. de Bruijn, *A Combinatorial Problem*, Proceedings of the Section of Sciences of the KoninklijkeNederlandse Akademie van Wetenschappen te Amsterdam, 49(7)(1946), 758-764.