Genomic Reconstruction: Short-Read Sequencing and Computational Algorithms

Sabrina Eng September 26, 2021

**Abstract**

Sequencing the human genome provides valuable resources for biomedical research and medicinal practice. Modern technology struggles with whole-sequencing methods; most genome sequence techniques involve a short-read process by breaking apart the genome into small read segments. To reconstruct the entire sequence, each read is overlapped as a search for a Eulerian path. Computer science is thus applied to assemble the genome as the data sets are complex. This paper details one computational approach to genome reconstruction using the short-read sequencing described above.

# Introduction

DNA, or deoxyribonucleic acid, is a molecule that carries genetic information and thus is present in all organisms. Genetic information is stored by the ordering of four chemical bases: adenine (A), guanine (G), cytosine (C), and thymine (T). These nucleotides are complimentary pairs; A pairs with T and C with G. Via this complimentary bonding, two polynucleotide chains join to create a three-dimensional double helix structure [7].

The genome is the entirety of the genetic information present within DNA. Every cell contains a complete copy of the genome. Thus, a sample of cells for study contains millions of identical genomes. Furthermore, most genomes are incredibly long; the human genome contains over 3 billion base pairs. These reasons contribute to the difficulty in long-read genome sequencing – modern technology cannot handle the immense number of base pairs [1].

Short-read genome sequencing, to combat this problem, reconstructs the genome from short reads. Using biochemical processes, genomes from the sample are broken into small sections and reassembled by overlapping reads using computing power [8]. This method falls under the field of bioinformatics: an interdisciplinary field combining computer science and biology. With the increasing amount of biological data being discovered, computer programs are required for the analysis of patterns between sequences of molecules. The general bioinformatics solution involves building and testing a computational model. This is a simplified overview of the process that will be covered in this article.

Sequencing the human genome gives scientists important clues to the function of specific sequences, and its application to preventative and therapeutic health care. For example, a genetic variant of the apolipoprotein E gene, involved in the production of a protein that helps carry cholesterol in the bloodstream, has been identified with increasing one’s risk for late-onset Alzheimer’s disease [6]. Genome sequencing holds further implications in understanding human variation, hu- man relation to other organisms, and cell development studies, to name a few.

# Genome Reconstruction by Inspection

For this paper, the genome is simplified to a singular sequence of bases, as we are not concerned with the 3-dimensional structure or the bond between the two polynucleotide chains. These elements do not contribute to the basic genetic information encoded.

Biochemical processes break the sample identical genomes into reads of the same length. The breakage process is random in its start point and thus results in different reads.

AGTCTGCTGACTGACGTTA AGTCTGCTGACTGACGTTA AGTCTGCTGACTGACGTTA AGTCTGCTGACTGACGTTA

AGTC—TGCT—GACT—CACG—TTA A—GTCT—GCTG—ACTC—ACGT—TA AG—TCTG—CTGA—CTCA—CGTT—A AGT—CTGC—TGAC—TCAC—GTTA

Figure 1: Sample genome sequence and resulting 4-base reads

Only the reads of the same length are used for reconstruction. Incorporating the other read lengths would result in redundancy of bases and an inaccurate final sequence.

By observation, reads are aligned by analyzing their base composition similarities.

ACGT, ACTG, AGTC, GACG, CGTT, CTCA, CTGA, CTGC, GACT, GCTG, GTCT, GTTA, TCAC, TCTG, TGAC, TGCT

GTTA CGTT

ACGT**\*** GACG

TGAC CTGA

ACTG GACT

TGAC CTGA

GCTG TGCT

CTGC TCTG

GTCT AGTC

Sequence: AGTCTGCTGACTGACGTTA

Figure 2: Alphabetical list of 4-base reads and representation of overlap method

A suffix, the sequence of bases minus the first base, is overlapped with a matching prefix, the sequence of bases minus the last base as demonstrated in Figure 2. The sequence is elongated until all reads are used.

However, this method is not realistic when compared with the average size of the genome, as the number of reads to organize involves an impossible time investment. Using a computational algorithm solves this issue.

# Computational Formulation

To make the algorithm applicable to various read lengths and sequences, reads will be defined as k- mers for future reference. K-mers are sequences of k letters from the genome, with k as an unknown value. For example, 3-mers from the sequence of Figure 1 could be AGT and TTA (the first and last 3 bases). Multiple k-mers of the same base order can be repeated if they appear multiple times within the genome sequence. Short-read genome sequencing takes a set of k-mers to create a genome sequence. Computationally, then, a list of k-mers run through the ideal algorithm should output the reconstructed genome. Developing the algorithm involves elements of graph theory.

# Graph Formulation

As the genome reads in only one direction, direct graphs are used for modeling. Direct graphs consist of vertices (nodes) and edges. Edges connect two nodes and have a direction, pointing from the tail to the head [2].

*n*1

*e*1

*n*2

Figure 3: Basic directional graph with 2 nodes; edge tail at node 1, edge head at node 2

In a graph path, the head of edge *ei* must equal the tail of edge *ei*+1 by the end of the edge sequence. Furthermore, an edge cannot be visited twice.

*n*2

*e*2

*n*4

*e*1

*e*3

*e*5

*n*1

*e*4

*n*3

Figure 4: Basic direction graph path with every edge *ei* equaling *ei*+1

The algorithm must combine the sequencing of suffixes and prefixes with paths. Although k- mers can appear multiple times in a genome sequence, when graphing the path, it is illustrated as a single node. This allows for the possibility of loops in the graph, or multiple edges traveling to and from the same nodes.

When graphing the list of k-mers, the intersection of prefixes and suffixes become the nodes, and the k-mers show the connection between the two as edges.

K represents a given list of k-mers, and N(K) represents the list prefixes and suffixes with repeats removed. Example graphical representation of list K:

K = [TCT, CTG, TGC, GCT, CTG, TGA, GAC, ACT, CTG, TGA, GAC, ACG, CGT,] N(K) = [TC, CT, TG, GC, GA, AC, CG, GT]

|  |  |  |
| --- | --- | --- |
| Edge with Label | Tail Node | Head Node |
| *TCT* | *TC* | *CT* |
| *CTG* | *CT* | *TG* |
| *TGC* | *TG* | *GC* |
| *GCT* | *GC* | *CT* |
| *CTG* | *CT* | *TG* |
| *TGA* | *TG* | *GA* |
| *GAC* | *GA* | *AC* |
| *ACT* | *AC* | *CT* |
| *CTG* | *CT* | *TG* |
| *TGA* | *TG* | *GA* |
| *GAC* | *GA* | *AC* |
| *ACG* | *AC* | *CG* |
| *CGT* | *CG* | *GT* |

Table 1: Base data of sample genome sequence for graph usage

*GCT*

*GC*

*TGC*

*ACG*

*CGT*

*AC CG GT*

*CTG*

*GAC*

*TCT*

*CTG*

*TC CT TG*

*TGA*

*GAC*

*CTG*

*TGA*

*GA*

*ACG*

Figure 5: Graphical path of sample genome sequence labeled with the 3-mer as edges

To reconstruct the genome sequence based on the graph, the last nucleotide of each succeeding k-mer is added to the starting sequence. This is done by following the numerical progression of edges.

*e*8

*e*4

*GC*

*e*3

*e*12

*e*13

*e*1

*e*2

*e*5

*e*7

*AC*

*CG*

*GT*

*TC CT TG*

*e*6

*e*11

*e*9

*e*10

*GA*

Figure 6: Graphical path of sample genome sequence labeled with ordered edges

TC + T + G + C + T + G + A + C + T + G + A + C + G + T = TCTGCTGACTGACGT

For a list of k-mers, it is possible to create more than one resulting genome sequence. In the context of the information from Figure 6, Edge 6 could become Edge 3, resulting in a difference in a segment of the sequence - ...TGACTG... compared to the identified ...TGCTGA... segment above.

# Eulerian Paths

Eulerian paths simplify the graphical process and prevent the loss of sequence repeats and Eulerian cycles help to find Eulerian paths. In a Eulerian cycle, each node has the same number of incoming edges i(N) as outgoing edges o(N) [5].

*|i*(*N* ) *− o*(*N* )*|* = 0

The graphical representation of the sample sequence TCTGCTGACTGACGT in Figure 6 is not a Eulerian cycle, as the left and right-most nodes do not have an even degree; there is an odd number as the difference between incoming and outgoing edges. This is the characteristic of a Eulerian path: A path is a Eulerian path if and only if *|i*(*N* ) *− o*(*N* )*|* = 0 for all nodes except the start node, where *|i*(*N* ) *− o*(*N* )*|* = *−*1, and the end node, where *|i*(*N* ) *− o*(*N* )*|* = 1. If the graph fits these conditions, a Eulerian path exists. Then, a temporary edge is added from the end node to the start node to create a Eulerian cycle (as for both the start and end node, *|i*(*N* ) *− o*(*N* )*|* = 0 is now true), and the Eulerian cycle is adjusted to begin with the edge after the temporary edge. The temporary edge is removed, and the Eulerian path is left. An example of this process is shown below:

Sample k-mer list: [TAA, AAT, ATG, TGG, GGG, GGA, GAT, ATG, TGC, GCC, CCA, CAT, ATG, TGT, GTT]

*CA*

*CC*

*GC*

*TA*

*AA*

*AT*

*TG*

*GT*

*TT*

*GA*

*GG*

* 1. Prefixes and suffixes are represented through nodes.

*CC*

*CA* ?

? *GC*

?

?

?

*TA AA AT*

?

?

?

?

?

?

*TG*

*GT*

*TT*

?

*GA*

?

*GG*

?

?

*T*

* 1. Eulerian cycle formed with unknown edge order.

*e*11

*CC*

*CA*

*e*10

*GC*

*e*1

*e*2

*e*12

*TA AA AT*

*e*3

*e*8 *e*13

*e*9

*e*14

*e*15

*TG*

*GT*

*TT*

*e*7

*e*4

*GA*

*GG*

*e*6

*e*5

* 1. *e*1 begins after temporary edge, Eulerian cycle adjusted accordingly and temporary edge removed, Eulerian path discovered.

*TA* + *A* + *T* + *G* + *G* + *G* + *A* + *T* + *G* + *C* + *C* + *A* + *T* + *G* + *T* + *T* = *TAATGGGATGCCATGTT*

* 1. Using the Eulerian path, adding each suffix after the first node creates the genome sequence.

Figure 7: Eulerian path process for genome reconstruction from sample k-mer list

Again, the resulting genome sequence in Figure 7 is not the only possible sequence, but this paper does not cover the consequences of multiple paths, only whether a genome sequence can be

created from a list of given k-mers. These steps, from the separation of prefixes and suffixes of the k-mers to the graphing is all computed by algorithms. The algorithms are capable of computing much larger lists and sequences, making this a viable method for genome reconstruction.

# Limitations

Although the use of Eulerian paths in short-read sequencing reduces the challenge of repetitive sequences, amplification bias is still high. The short length of the read also contributes to the lack of quality in the reconstructed genome. The shorter the read sequence, the less accurate the reconstructed genome will be, and the more uncertainty there is in mapping. The relative position of a read in a complex, repetitive genome is difficult to determine with short sequences. Furthermore, short reads can create multiple Eulerian paths when mapping the genome, due to the repetition of sequences. As there is only one correct sequence in genome reconstruction, the presence of multiple Eulerian paths decreases the reliability of the method [3].

In contrast, long-read sequencing takes much longer DNA sequences for a potential in improved de novo genome assembly. Currently, short-read sequencing is more accurate in its application to genome reconstruction, and long-read segments have a much higher error rate in read sequences, with every eighth to ninth base incorrect. However, sequenced accurately, long-read sequencing provides a more complete map of the genome structure and resolves the issues of short-read sequencing. The algorithms discussed in this paper would further the benefits of long-read sequencing, increasing knowledge on contextualization for read segments in reference to the whole genome [4].

# Conclusion

Computer science is essential to the analysis of the genome. Technology is continually evolving, increasing the accuracy and reliability of the methods used to create reads. Beginning with mil- lions to billions of reads, algorithms and graphs are applied, resulting in knowledge that benefits human healthcare and well-being. Although this paper only addresses the foundations of genome reconstruction, it is valuable as an introduction into the importance of data science to the field of biology.

# References

1. Shanika L Amarasinghe, Shian Su, Xueyi Dong, Luke Zappia, Matthew E Ritchie, and Quentin Gouil. Opportunities and challenges in long-read sequencing data analysis. *Genome biology*, 21(1):1–16, 2020.
2. Gary Chartrand. *Introductory graph theory*. Courier Corporation, 1977.
3. Kishore R Kumar, Mark J Cowley, and Ryan L Davis. Next-generation sequencing and emerging technologies. In *Seminars in thrombosis and hemostasis*, volume 45, pages 661–673. Thieme Medical Publishers, 2019.
4. Mohit K Midha, Mengchu Wu, and Kuo-Ping Chiu. Long-read sequencing in deciphering human genetics to a greater depth. *Human genetics*, 138(11):1201–1215, 2019.
5. Dev Patel. Russian bridges, eulerian circuits, and genome assembly?, Jul 2021.
6. Judes Poirier, P Bertrand, S Kogan, S Gauthier, J Davignon, and D Bouthillier. Apolipoprotein e polymorphism and alzheimer’s disease. *The Lancet*, 342(8873):697–699, 1993.
7. Lisa A Urry, Michael Lee Cain, Steven Alexander Wasserman, Peter V Minorsky, and Jane B Reece. *Campbell biology in focus*, volume 10. Pearson Boston, MA, 2014.
8. Nava Whiteford, Niall Haslam, Gerald Weber, Adam Pru¨gel-Bennett, Jonathan W Essex, Pe- ter L Roach, Mark Bradley, and Cameron Neylon. An analysis of the feasibility of short read sequencing. *Nucleic acids research*, 33(19):e171–e171, 2005.