International Journal of Arts, Humanities and Social Studies
Abbreviated Key Title: *Int J Arts Huma Social Studies*
Home page: https://ijahss.in/
Volume: 6 Issue:2 (Mar-Apr 2024); Page No: 14-21
DOI: : http://doi.org/10.5281/zenodo.10876095

OPEN ACCESS
ISSN: 2582-3647
Original Article

Graph Theory Meets Genetics: Understanding Genome Assembly Through Euler And De Bruijn Graphs

Ahmed Sohail[1*], Yuhan Guo[1]

[1]School of Science, Zhejiang University of Science and Technology, Hangzhou, 310023, China

***Corresponding Author**
Ahmed Sohail

ABSTRACT

This abstract delves into the intricate intersection of Graph Theory and Genetics, focusing on the pivotal role of Eulerian and De Bruijn graphs in deciphering the complexities of genome assembly. Genome assembly, a fundamental challenge in bioinformatics, involves reconstructing the complete DNA sequence of an organism from short, fragmented reads. By leveraging graph theory concepts, specifically Eulerian and De Bruijn graphs, researchers can navigate through this intricate puzzle of genetic information. Eulerian graphs, pioneered by Leonhard Euler in the 18th century, provide a powerful framework for analyzing interconnected paths within genomic data. These graphs offer a systematic approach to trace the sequence of DNA fragments and identify overlaps, crucial for reconstructing the entire genome accurately. On the other hand, De Bruijn graphs, named after mathematician Nicolaas Govert de Bruijn, offer a more efficient representation of genomic sequences by breaking them into smaller, overlapping k-mers. This abstract explores how these graph structures serve as indispensable tools in genome assembly by enabling researchers to address challenges such as repetitive sequences, sequencing errors, and genome complexity. By dissecting genetic information into manageable components and reconstructing the original sequence through graph traversal algorithms, scientists can unravel the mysteries encoded within the DNA strands. This exploration not only enhances our theoretical understanding of graph-based genome assembly but also provides valuable insights into the practical considerations and applications of Eulerian and De Bruijn Graphs in the ever-evolving landscape of genomics

Key Words: *Graph Theory; Genome Assembly; Eulerian Graphs; De Bruijn Graphs; Bioinformatics*

Literature Review

In 1953, the groundbreaking work of J.D. Watson and F.H.C. Crick [1] culminated in the formulation of the double-helix model, a pivotal representation of the DNA molecule that integrated chemical and physical data. DNA, an acronym for Deoxyribo Nucleic Acid, comprises two antiparallel strands linked by two or three hydrogen bonds, resulting in a helical configuration. These strands serve as the repository for the genetic information of all living organisms, including humans. The DNA structure is characterized by four nucleotide bases: guanine (G), thymine (T), adenine (A), and cytosine (C). Adenine forms complementary pairs with thymine, while guanine pairs with cytosine. Consequently, genome sequencing involves the determination of the arrangement of these nucleotide bases within the genome. The swift progression of genome sequencing technologies has rendered it indispensable in numerous biological studies, as well as in interdisciplinary research domains such as biotechnology, forensic biology, and diagnostics. Profound insights into genome sequencing have become imperative for advancements in these applied disciplines. Notably, Edwin Southern introduced the hybridization-based sequencing (SBH) method in 1988 [2], further contributing to the arsenal of genome sequencing techniques. The fusion of Graph Theory with Genetics has significantly advanced the field of genome assembly, offering a sophisticated framework for deciphering complex genetic sequences. This literature review synthesizes recent research findings and insights on the utilization of Eulerian and De Bruijn graphs in understanding genome assembly.

In terms of Computational Genome Assembly by Graph Theory, In a recent study on computational genome assembly by graph theory (Sarkar, Bijan. (2024)) [3], the challenges of assembling short sequencing reads into a contiguous genome were explored. The foundational knowledge of DNA sequences and the application of graph theory principles were highlighted as essential components in addressing this daunting task. Concerning Application of De Bruijn Graphs in Genome Assembly, A review article emphasized the practical application of De Bruijn graphs in genome assembly(Compeau, P. E. C., Pevzner, P. A., & Tesler, G. (2011))[4]. By breaking down genomic data into k-mer prefixes and suffixes, these graphs provide a powerful representation for reconstructing fragmented genetic information efficiently.

On the Algorithmic Features for Genome Assemblers, General algorithmic features for genome assemblers were discussed in a comprehensive review(Wajid, B., & Serpedin, E. (2012))[5], emphasizing the extensive use of graph theory in genome assembly. The study highlighted the importance of graph structures consisting of nodes and edges in resolving the genome assembly problem effectively. On Genome Assembly and Eulerian Cycles, Research elucidated the relationship between Eulerian and Hamiltonian cycles with genome assembly(Medvedev P, Pop M (2021))[6]. While Eulerian and Hamiltonian cycles play a crucial role in theoretical discussions, their complexity does not directly impact genome reconstruction algorithms based on De Bruijn graphs.

About the Future Directions in Whole-Genome Assembly,A review on de novo whole-genome assembly categorized assemblers based on De Bruijn graphs into Hamiltonian and Eulerian types(Jang-il Sohn, Jin-Wu Nam (2018))[7]. The discussion emphasized the performance superiority of Eulerian de Bruijn graph-based assemblers like EULER, SPAdes, ALLPATHS-LG, and MaSuRCA in handling large genomes efficiently.

When it comes to advances in Genome Sequencing, the field of genome sequencing has undergone significant advancements in recent years, resulting in increased accuracy, reduced costs, and improved speed. The development of next-generation sequencing technologies has made it possible to sequence an entire genome in a matter of days, at a fraction of the cost of previous methods. This has led to a significant increase in the number of sequenced genomes, enabling researchers to study genetic variation on a large scale (Goodwin, S., McPherson, J. & McCombie, W. (2016)) [8]. Genome sequencing has numerous applications in the fields of medicine, biology, and agriculture. In medicine, genome sequencing is used to diagnose genetic diseases, predict disease susceptibility, and develop personalized treatments(Ashley E. A. (2015)) [9]. In agriculture, genome sequencing is used to improve crop yields, develop disease-resistant strains, and improve food security (Varshney RK, Terauchi R, McCouch SR (2014)) [10].

Despite the many advances in genome sequencing, there are still some challenges and limitations that need to be addressed. One of the biggest challenges is the analysis of large amounts of sequencing data, which requires sophisticated computational tools and algorithms (Duan, J., Shi, J., & Ge, H. (2013)) [11]. Another challenge is the interpretation of genetic variants, as many variants are of unknown significance and their effects on health are still not well understood (Richards, S., Aziz, N., (2015)) [12]. Genome sequencing has revolutionized the field of genetics and has numerous applications in medicine, biology, and agriculture. With continued advancements in technology and analysis tools, genome sequencing is poised to play an increasingly important role in personalized medicine, disease diagnosis, and treatment, and in improving food security.

Understanding Genome Assembly Through Euler Graph

In this particular section, we will indicate how the genome is sequenced using the iconic Eulerian approach method. Whenever there exists an end-to-end closed walk that goes through each and every edge of a connected graph G, then that same graph is affectionately known as an Euler graph. In order for a path to be classified as an Euler path, each edge of a graph must be utilized exactly one time. The start and termination vertices of an Euler path are always distinct and different. Indeed, every edge of a graph must be duly utilized just once in an Eulerian circuit. This same identical vertex dedicates itself as the starting point and endpoint in an Euler circuit route. A connected graph G is Eulerian if and only if all vertices have even degree, and a connected graph G is Eulerian if and only if its edge set can be decomposed into circles.

Objective: Employing an Eulerian approach, the goal is to reconstruct the genome sequence from a given set of reads, denoted as S.

Each unique prefix or suffix within the genome reads is represented as a node. These nodes correspond to substrings, or (l-1)-mers, derived from the l-mers in set S. When a node possesses a prefix v and a suffix w, they are linked together. This connection occurs if the final l-2 elements of node v match the first l-2 elements of node w, and the concatenation of v and w belongs to set S. To reconstruct the shortest sequence string via the Eulerian path, a collection of (l-1)-mer strings, i.e., strings one character shorter than the given strings, is considered.

Illustrative Example 1: Consider a multiset M containing 3-long nucleotides extracted from a genome sequence: {TGT, AAT, TGG, ATG, TGC, ATG, TAA, ATG, GTT, CAT, CCA, GGA, GCC, GAT, GGG}. We derive all distinct (k-1)-mers from our set of k-mers, where k=3. For instance, from TGC and TGG, we obtain TG, GC, and GG. The process begins with the construction of a multi-graph comprising nodes representing (k-1)-mers. Subsequently, edges are drawn between two (k-1)-mers if they originate from the same genome read, such as AAT & ATG (refer to Figure 3).: Let's take H = {AGT, TCC, TTT, TTC , GCT, CCA, CTA, TCC, TAG ,AAT, TGC, CAA} as a multiset containing all 3-long nucleotides of a genome sequence. By establishing nodes for every unique prefix/suffix, such as CTA, we identify the prefix CT and the suffix TA. This process is illustrated in Table 1. Upon identifying all distinct prefixes and suffixes, we obtain the set:

V = {AT, GC, CT, CC, AA, TG, TT, CA, AG, GT, TC, TA}.

Table 1. Prefix and Suffix of Example 1

3-LONG NUCLEOTIDES	PREFIX	SUFFIX
AAT	*AA*	*AT*
TGC	*TG*	*GC*
CAA	*CA*	*AA*
GCT	*GC*	*CT*
CCA	*CC*	*CA*
CTA	*CT*	*TA*
TCC	*TC*	*CC*
TAG	*TA*	*AG*
AGT	*AG*	*GT*
TCC	*TC*	*CC*
TTT	*TT*	*TT*
TTC	*TT*	*TC*

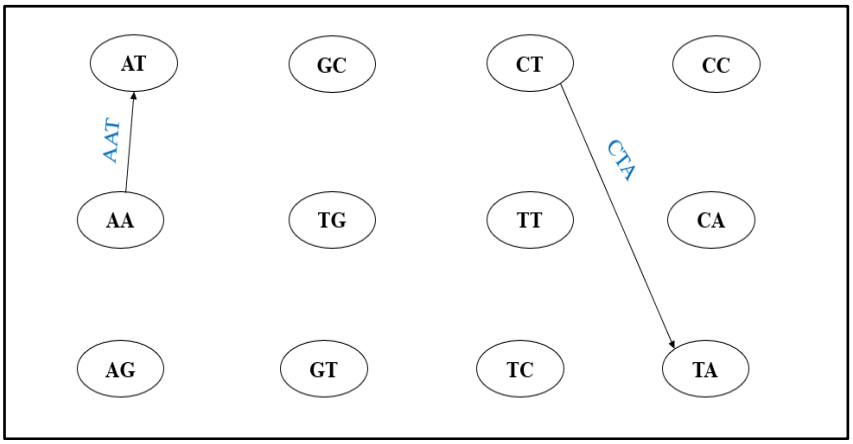

Figure 1. multigraph with AAT and CTA

As demonstrated earlier (refer to Figure 1), the prefix AA links to the suffix AT via an AAT edge, mirroring the DNA read. Similarly, the prefix CT connects to the suffix TA through a CTA edge, reflecting the DNA read as well. By completing the diagram and connecting these prefixes to their corresponding suffixes, we create the graph depicted in Figure 2. In terms of connectivity, it's essential for the graph to remain connected. If it isn't, it can be broken down into its connected components, each of which can then be individually examined for Eulerian properties. One crucial property is the even degree requirement: every vertex in the graph must have an even degree. The degree of a vertex signifies the number of edges incident to it. This requirement arises from the fact that when traversing an edge, we enter and exit a vertex, thus contributing 2 to the vertex's degree. Consequently, if there are any vertices with an odd degree, it becomes impossible to find a circuit that traverses every edge exactly once.

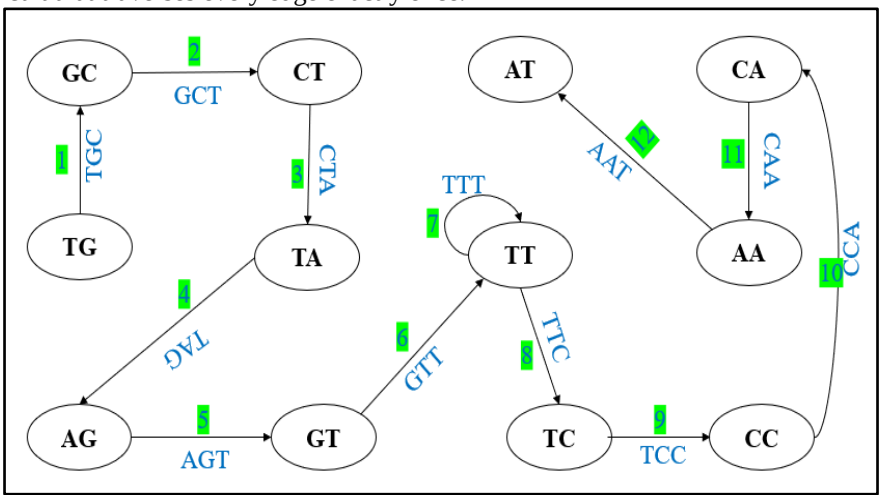

Figure 2. Complete Eulerian path of H.

The Eulerian path depicted in Figure 2 guides our genome reconstruction process from k-mer DNA reads, as indicated by the numbered markers. By leveraging overlaps in DNA reads, we generate the path table illustrated in Table 2. Despite its efficacy, the Eulerian approach is not without limitations. It is particularly sensitive to errors within the input data, such as sequencing errors and genome rearrangements. To mitigate these challenges, various methods have been devised, including error correction algorithms and hybrid assembly approaches that amalgamate different sequencing technologies. Euler's theorem establishes that a graph is Eulerian if and only if each vertex possesses an even degree. This property facilitates the tracing of a path through the graph, ensuring that each edge is visited exactly once, thereby aiding in the assembly of sequences into a longer contiguous sequence.

Table 2 demonstrates the reconstruction of our original genome through the alignment of DNA reads, yielding the Eulerian path TGCTAGTTTCCAAT. Contrasting the Hamiltonian and Eulerian approaches, the key disparity lies in computational efficiency. Computers can rapidly identify Eulerian cycles, whereas Hamiltonian cycles pose a more challenging computational task. Since then, significant progress has been made in this field. Notably, the introduction of de Bruijn graphs has revolutionized genome reconstruction by segmenting DNA sequences into k-mers, each represented as a node in the graph. This strategy proves particularly effective for larger genomes and has spurred the development of numerous genome assemblers.Table 2. Eulerian path table of H.

	T	*G*	*C*											
		G	*C*	*T*										
			C	*T*	*A*									
				T	*A*	*G*								
					A	*G*	*T*							
						G	*T*	*T*						
							T	*T*	*T*					
								T	*T*	*C*				
									T	*C*	*C*			
										C	*C*	*A*		
											C	*A*	*A*	
												A	*A*	*T*
GENOME	**T**	**G**	**C**	**T**	**A**	**G**	**T**	**T**	**T**	**C**	**C**	**A**	**A**	**T**

Understanding Genome Assembly Through De Bruijn Graph

Genome sequencing using De Bruijn graphs is a computational approach employed in bioinformatics to reconstruct the complete DNA sequence of an organism. It is particularly effective for handling the large datasets generated by high-throughput DNA sequencing technologies, such as Next-Generation Sequencing (NGS).

Here's a simplified explanation of how genome sequencing using De Bruijn graphs works:

1) Fragmentation of DNA:
   - The first step involves breaking down the DNA into smaller fragments through a process called DNA sequencing. These fragments are usually short sequences of nucleotides (A, T, G, C) that can be read by sequencing machines.
2) Construction of De Bruijn Graphs:
   - Each fragment is represented as a node in the De Bruijn graph, and overlapping sequences between fragments are connected by edges. The De Bruijn graph is constructed based on k-mers, which are substrings of length k from the DNA fragments.
   - For example, if we have the DNA sequence "ATGC," and we choose k=2, the corresponding k-mers would be "AT," "TG," "GC."
   - The nodes in the De Bruijn graph represent all possible k-mers present in the DNA fragments, and edges connect overlapping k-mers.
3) Graph Simplification:

- De Bruijn graphs can be quite complex due to repetitive regions in genomes. To simplify the graph and facilitate analysis, some redundant nodes and edges are often collapsed or removed.

4) Path Traversal:
   - Genome assembly involves finding a path through the simplified De Bruijn graph that covers all unique k-mers, essentially reconstructing the original DNA sequence.
   - This path traversal corresponds to the reconstruction of the genome, where each unique path represents a potential arrangement of the DNA fragments.
5) Handling Errors and Gaps:
   - DNA sequencing technologies may introduce errors, and some regions of the genome might not be covered by sequencing reads. Strategies are employed to address errors, such as error correction algorithms, and to fill gaps in the genome assembly.
6) Validation and Refinement:
   - The assembled genome is then validated and refined through various computational and experimental methods to ensure accuracy. This may involve comparing the assembled genome to reference genomes or using additional sequencing data.

Illustrative Example 2: Consider a multiset M containing 3-long nucleotides extracted from a genome sequence: {TGT, AAT, TGG, ATG, TGC, ATG, TAA, ATG, GTT, CAT, CCA, GGA, GCC, GAT, GGG}. We derive all distinct (k-1)-mers from our set of k-mers, where k=3. For instance, from TGC and TGG, we obtain TG, GC, and GG. The process begins with the construction of a multi-graph comprising nodes representing (k-1)-mers. Subsequently, edges are drawn between two (k-1)-mers if they originate from the same genome read, such as AAT & ATG (refer to Figure 3).

Table 3. Prefix and Suffix of Example 3

3-LONG NUCLEOTIDES	PREFIX	SUFFIX
TGT	*TG*	*GT*
AAT	*AA*	*AT*
TGG	*TG*	*GG*
ATG	*AT*	*TG*
TGC	*TG*	*GC*
ATG	*AT*	*TG*
TAA	*TA*	*AA*
ATG	*AT*	*TG*
GTT	*GT*	*TT*
CAT	*CA*	*AT*
CCA	*CC*	*CA*
GGA	*GG*	*GA*
GCC	*GC*	*CC*
GAT	*GA*	*AT*
GGG	*GG*	*GG*

Table 3 above shows the 3-long nucleotides being split to their respective prefix and suffixes.

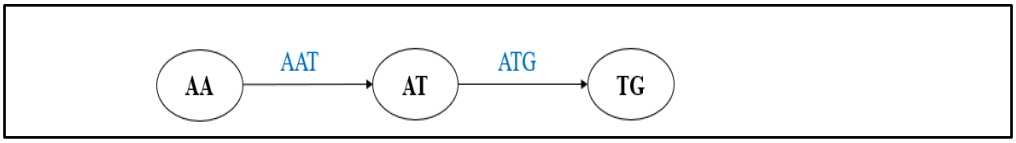

Figure 3.multigraph with AAT and ATG.

By adhering to this method, we ensure the graph possesses an Eulerian trail. Following this trail and connecting the nodes accordingly enables the reconstruction of the original genome sequence. A representation of such a graph will be provided, as depicted in Figure 4.

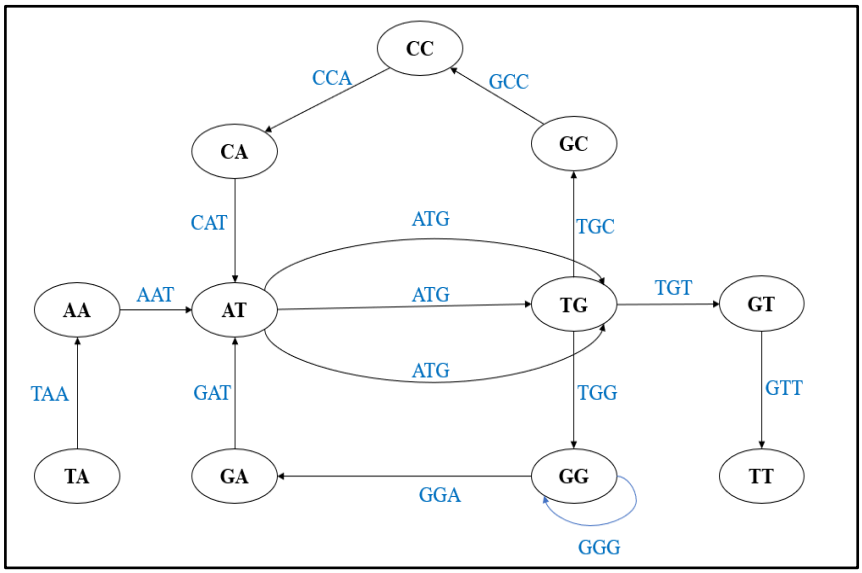

Figure 4. complete Debruijn graph of M.

As depicted in the illustration above (refer to Figure 4), each edge in this graph corresponds to a length-3 input string. By traversing this network via an Eulerian path, we can effectively reconstruct our original genome sequence, as shown in Table 4. The process involves following an Eulerian walk that traverses each edge exactly once, ultimately leading to the reconstruction of the genome sequence TAATGCCATGGGATGTT.

In terms of enhancing the accuracy of genome assembly, the de Bruijn graph approach proves invaluable. This approach aids in rectifying errors within sequencing data, thereby contributing to more precise genome assemblies. By leveraging the overlap among k-mers within the graph, the de Bruijn graph approach adeptly identifies and corrects sequencing errors, thereby refining the accuracy of the resulting genome assemblies.

Table 4. Recreating the initial genome sequence layout.

	T	*A*	*A*													
		A	*A*	*T*												
			A	*T*	*G*											
				T	*G*	*C*										
					G	*C*	*C*									
						C	*C*	*A*								
							C	*A*	*T*							
								A	*T*	*G*						
									T	*G*	*G*					
										G	*G*	*G*				
											G	*G*	*A*			
												G	*A*	*T*		
													A	T	G	
														T	G	T
															G	T	T
																T	T
GENOME	T	A	A	T	G	C	C	A	T	G	G	G	A	T	G	T	T

Conclusion

Differences between Euler and De Bruijn Graphs in Genome Sequencing

Eulerian graphs and De Bruijn graphs are both graph-based approaches used in genome sequencing, but they have distinct characteristics and serve different purposes.Below are the key differences between Eulerian graphs and De Bruijn graphs in the context of genome sequencing:

1) Purpose and Nature:
   - Eulerian Graphs: These graphs are used to find Eulerian paths or cycles in a graph. Eulerian paths traverse each edge exactly once, while Eulerian cycles traverse each edge exactly once and end at the starting node. Eulerian graphs are applicable to genome assembly where the task involves finding a path that covers every edge (corresponding to DNA fragments) exactly once.
   - De Bruijn Graphs: Specifically designed for genome sequencing, De Bruijn graphs represent overlaps between short DNA sequences (k-mers). They are directed graphs where nodes correspond to k-mers, and edges connect overlapping k-mers of length (k-1). De Bruijn graphs are used to reconstruct the entire genome by finding a path through the graphChikomana, S. and Hu, X. (2023) [13].
2) Graph Structure:
   - Eulerian Graphs: Eulerian graphs are undirected graphs, and they are characterized by the existence of an Eulerian path or cycle.
   - De Bruijn Graphs: These are directed graphs representing overlapping sequences of DNA fragments. Nodes in a De Bruijn graph correspond to k-mers, and edges represent overlaps between adjacent k-mers.
3) Overlaps Representation:
   - Eulerian Graphs: Do not inherently represent overlaps between sequences. They focus on finding paths that traverse each edge exactly once.
   - De Bruijn Graphs: Explicitly represent overlaps between DNA sequences by connecting k-mers that share a common (k-1)-mer prefix or suffix.
4) Handling Repetitive Sequences:
   - Eulerian Graphs: May struggle with repetitive sequences in genomes, as Eulerian paths can become ambiguous when traversing regions with repeated patterns.
   - De Bruijn Graphs: Are more robust in handling repetitive sequences due to the use of short k-mers, which helps distinguish between similar but distinct regions in the genome.
5) Application in Genome Sequencing:
   - Eulerian Graphs: Have been historically used in genome assembly, but they may face challenges with the complexity of modern genomes.
   - De Bruijn Graphs: Are widely adopted in modern genome sequencing techniques, such as de novo assembly((Xingyu Liao, Min Li,(2019))[14]using high-throughput sequencing data. They handle large datasets efficiently and are well-suited for current genomics research.

Similarities between Euler and De Bruijn Graphs in Genome Sequencing

Euler graphs and De Bruijn graphs, although distinct in their approaches to genome sequencing, share some similarities in their application within this domain. Here are the key similarities based on the provided search results:

Shared Characteristics:

- Graph Theory Basis: Both Euler graphs and De Bruijn graphs are rooted in graph theory, utilizing nodes and edges to represent genetic sequences for genome assembly.
- Efficiency: While Eulerian paths in De Bruijn graphs solve complex graph problems efficiently by visiting each node exactly once without simplification, Eulerian cycles in De Bruijn graphs correspond to single genome reconstructions, showcasing efficiency in solving genome assembly challenges.
- Genome Reconstruction: Both approaches aim to reconstruct the original genetic sequence by representing DNA sequences in a graph format, aiding in resolving repeats and assembling genomes efficiently.
- Applications: Tools like SPAdes and Velvet leverage both Eulerian de Bruijn graph-based assembly methods and De Bruijn graph-based assembly techniques for genome assembly tasks, demonstrating the practical application of these graph structures

In essence, while Euler graphs and De Bruijn graphs have distinct methodologies and applications in genome sequencing, they converge in their utilization of graph theory principles to represent genetic sequences efficiently and reconstruct genomes accurately. These shared characteristics highlight the versatility and effectiveness of graph-based approaches in addressing the complexities of genome assembly

REFERENCES

[1] Watson, J. D., & Crick, F. H. (1953). Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid. Nature, 171(4356), 737-738. https://doi.org/10.1038/171737a0

[2] Southern, E. (1998) Analyzing Polynucleotide Sequences. International Patent Application PCT/GB89/00460.

[3] Sarkar, Bijan. (2024). A Study of Computational Genome Assembly by Graph Theory. Annals of West University of Timisoara - Mathematics and Computer Science. 60. 1-24. 10.2478/awutm-2024-0001.

[4] Compeau, P. E. C., Pevzner, P. A., & Tesler, G. (2011). How to apply de Bruijn graphs to genome assembly: a mathematical concept known as a de Bruijn graph turns the formidable challenge of assembling a contiguous genome from billions of short sequencing reads into a tractable computational problem. Nature Biotechnology, 29(11), 987+. https://link.gale.com/apps/doc/A273078609/AONE?u=anon~6461e809&sid=googleScholar&xid=26e2767b

[5] Wajid, B., &Serpedin, E. (2012). Review of general algorithmic features for genome assemblers for next generation sequencers. Genomics, proteomics & bioinformatics, 10(2), 58–73. https://doi.org/10.1016/j.gpb.2012.05.006

[6] Medvedev P, Pop M (2021) What do Eulerian and Hamiltonian cycles have to do with genome assembly? PLoSComput Biol 17(5): e1008928. https://doi.org/10.1371/journal.pcbi.1008928

[7] Jang-il Sohn, Jin-Wu Nam, The present and future of de novo whole-genome assembly, Briefings in Bioinformatics, Volume 19, Issue 1, January 2018, Pages 23–40, https://doi.org/10.1093/bib/bbw096

[8] Goodwin, S., McPherson, J. & McCombie, W. (2016)Coming of age: ten years of next-generation sequencing technologies. Nat Rev Genet 17, 333–351. https://doi.org/10.1038/nrg.2016.49

[9] Ashley E. A. (2015). The precision medicine initiative: a new national effort. JAMA, 313(21), 2119–2120. https://doi.org/10.1001/jama.2015.3595

[10] Varshney RK, Terauchi R, McCouch SR (2014) Harvesting the Promising Fruits of Genomics: Applying Genome Sequencing Technologies to Crop Breeding. PLoS Biol 12(6): e1001883. https://doi.org/10.1371/journal.pbio.1001883

[11] Duan, J., Shi, J., & Ge, H. (2013). Challenges in genome sequencing analysis. Genomics, Proteomics & Bioinformatics, 11(5), 317-323https://doi.org/10.1016/j.gpb.2013.09.006

[12] Richards, S., Aziz, N., Bale, S., Bick, D., Das, S., Gastier-Foster, J., ... & Rehm, H. L. (2015). Standards and guidelines for the interpretation of sequence variants: a joint consensus recommendation of the American College of Medical Genetics and Genomics and the Association for Molecular Pathology. Genetics in Medicine, 17(5), 405-424. DOI: https://doi.org/10.1038/gim.2015.30

[13] Chikomana, S. and Hu, X. (2023) Genome Sequencing Using Graph Theory Approach. Open Journal of Discrete Mathematics, 13, 39-48. doi: 10.4236/ojdm.2023.132004.

[14] Xingyu Liao, Min Li, You Zou, Fang-Xiang Wu, Yi-Pan, Jianxin Wang. (2019) "Current challenges and solutions of de novo assembly" , Quantitative Biology, https://doi.org/10.1007/s40484-019-0166-9

Author Contribution Statement

Ahmed Sohail: Conceptualization, Data curation, Investigation, Methodology, Formal analysis, Validation, Writing – original draft, review & editing.

Yuhan Guo: Funding acquisition, Discussion.