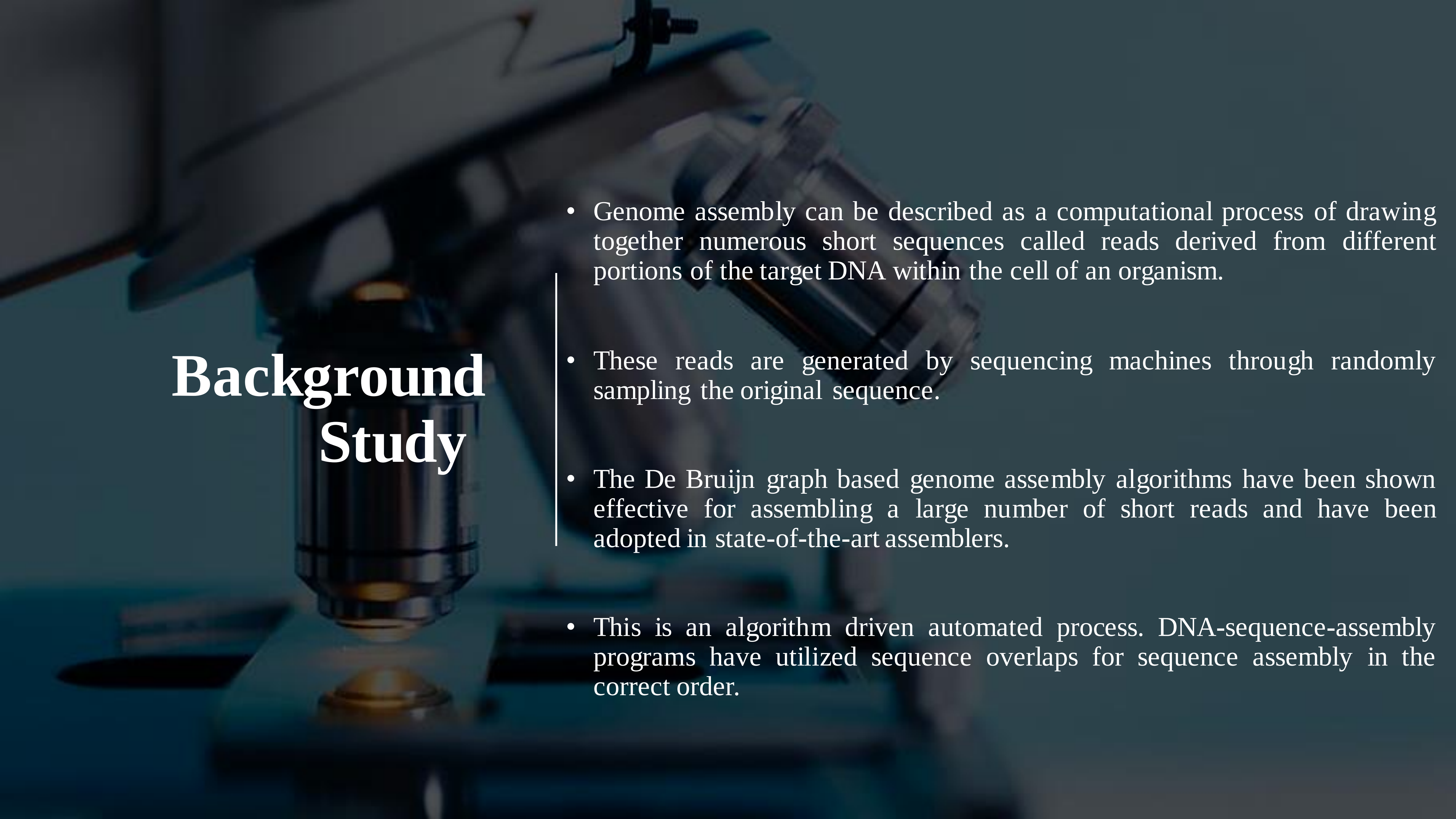
# De Bruijn Graph Construction in Genome Assembly

# Project Summary

- Objectives: To construct De Bruijn Graph in Genome assembly
- Scope: In bioinformatics De Bruijn graphs are used for de novo assembly of sequencing reads into a genome.



# Background Study

- Genome assembly can be described as a computational process of drawing together numerous short sequences called reads derived from different portions of the target DNA within the cell of an organism.

- These reads are generated by sequencing machines through randomly sampling the original sequence.

- The De Bruijn graph based genome assembly algorithms have been shown effective for assembling a large number of short reads and have been adopted in state-of-the-art assemblers.

- This is an algorithm driven automated process. DNA-sequence-assembly programs have utilized sequence overlaps for sequence assembly in the correct order.

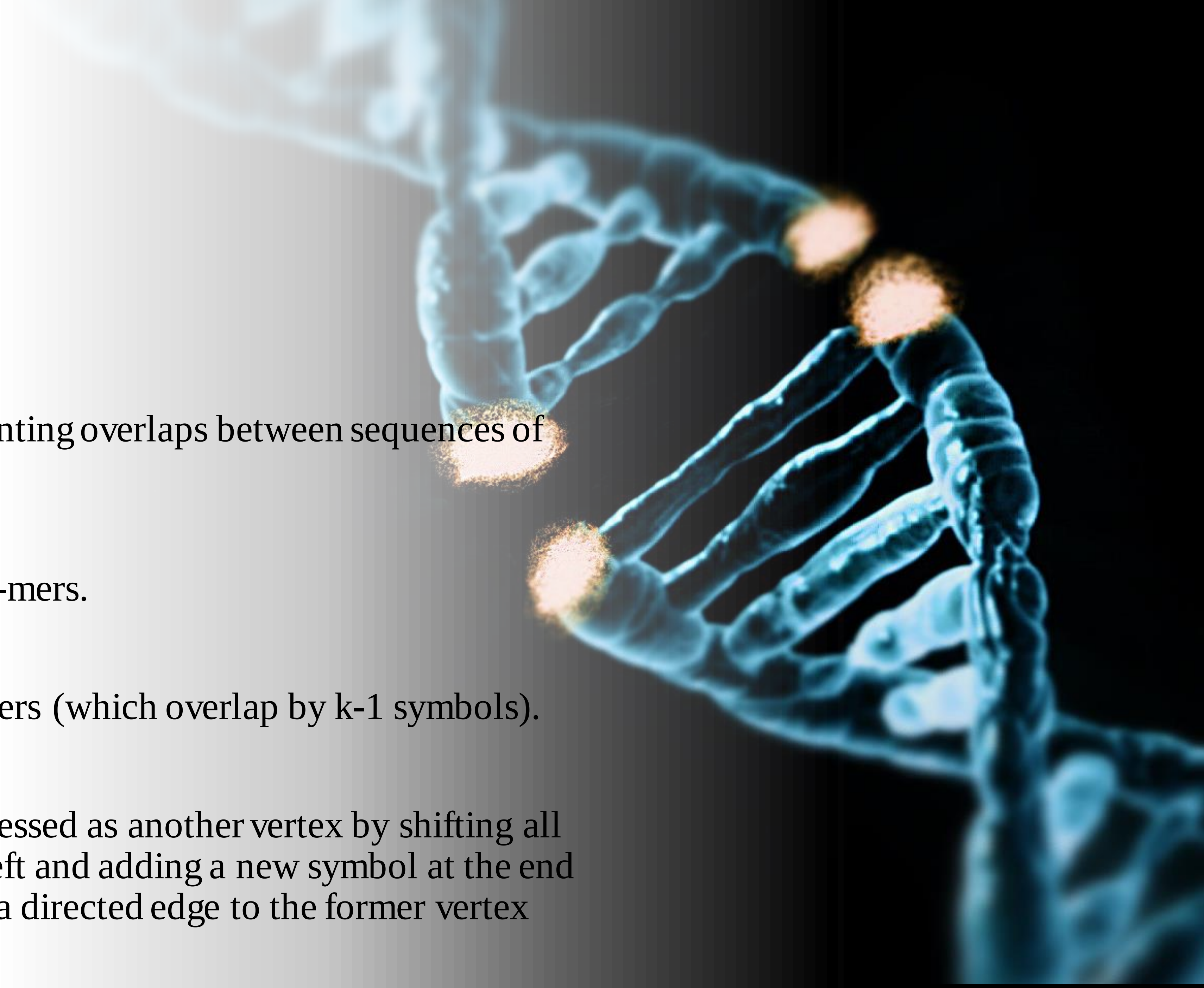
# De Bruijn

- They are directed graphs representing overlaps between sequences of symbols.

- Vertices/nodes in the graph are k-mers.

- Edges represent consecutive k-mers (which overlap by k-1 symbols).

- If one of the vertices can be expressed as another vertex by shifting all its symbols by one place to the left and adding a new symbol at the end of this vertex, then the latter has a directed edge to the former vertex



# Eulerian Path

- An Euler path, in a graph or multigraph, is a walk through the graph which uses every edge exactly once.
- A graph is called Eulerian if it has an Eulerian Cycle and called SemiEulerian if it has an Eulerian Path.
- The Eulerian Path problem is Polynomial time. String can be reconstructed by finding an Eulerian path in the de Bruijn graph

# Research Design

- The answer to the stated problem now was to find a path through the graph that traverses each edge exactly once, or in other words **Eulerian trail.**
- Reads are broken into smaller fragments of a specified size k.
- k-mers are identified and a de Bruijn graph with (k–1)-mers as nodes and k-mers as edges .
- A Eulerian path is traced through this network resulting in the reconstruction of the original genome sequence.

# Methods

Take all (k-1)-mers from the set of k-mers

Construct a multi-graph with nodes being k-1-mers; draw an edge between two k-1 mers only if the two k-1 mers are taken from the same read.

Graph constructed this way is guaranteed to have a Eulerian trail, follow the trail and connect the nodes to form our original sequence.

This algorithm can be used to assemble k-mer reads

# Algorithm

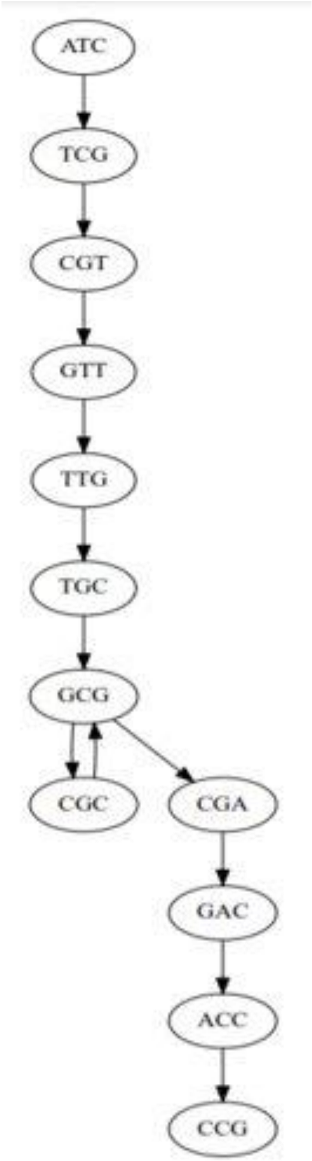
- build_k_mer function is used to generate k-mers from the given input sequence.
- debruijnize function helps to create nodes from the kmers .
- make_node_edge_map function maps the connection between each nodes
- eulerian_trail helps in the tracing of eulerian path through the (k-1)mers
- visualize_debruijn is used for visual representation of de-bruijn graph
- assemble_trail is used to reconstruct the original string from the eulerian path through the de-bruijn graph

# Results and Discussion

- Here we have taken a string "ATCGTTGCGCGACCG" and the kmer value as 4 and have obtained the results.

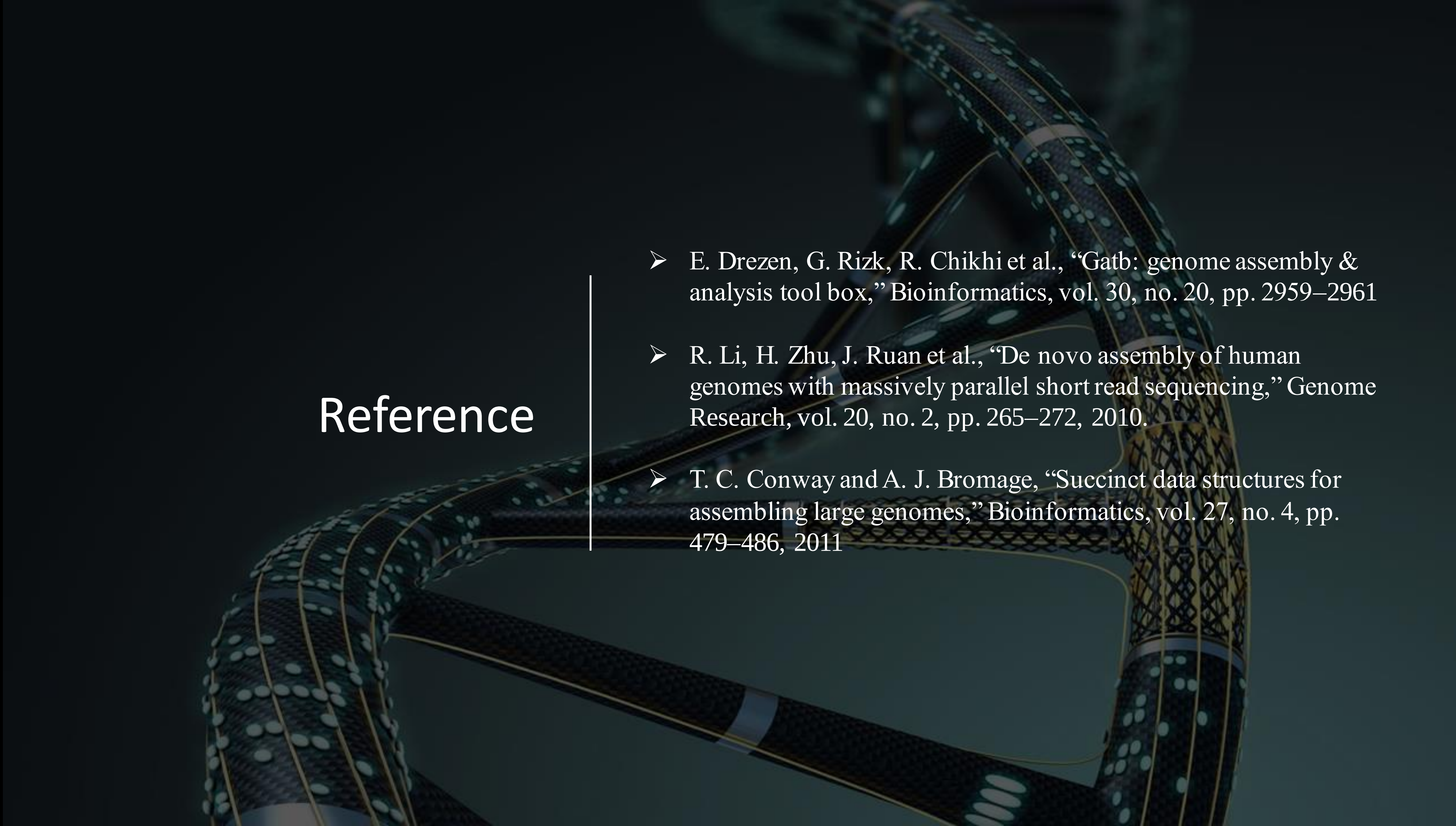
- The reads are generated using this read, and from those reads, we have reconstructed the original string using de bruijn graph

- The original and the reconstructed strings can then be compared. The de bruijn graph was constructed from the reads



# Conclusion

- In this project we were able to understand and implement the concepts of de-Bruijn graphs in genome assembly.
- We were able to reconstruct the genome from the given input string and k-mer composition which when compared to the initial input string is found to be the same, hence concluding with successful implementation.
- But the De Bruijn graphs do not preserve positional information.
- Valuable context information stored within the reads is lost for assembly, because the k-Mers have to be shorter than the actual read length.



# Reference

- E. Drezen, G. Rizk, R. Chikhi et al., “Gatb: genome assembly & analysis tool box,” Bioinformatics, vol. 30, no. 20, pp. 2959–2961

- R. Li, H. Zhu, J. Ruan et al., “De novo assembly of human genomes with massively parallel short read sequencing,” Genome Research, vol. 20, no. 2, pp. 265–272, 2010.

- T. C. Conway and A. J. Bromage, “Succinct data structures for assembling large genomes,” Bioinformatics, vol. 27, no. 4, pp. 479–486, 2011