Construction and Visualization of a De Bruijn Graph

Yifei Wang[1]

[1]Department of Biomedical Informatics & Data Science, Yale School of Medicine, Yale University

ABSTRACT

Introduction De Bruijn graphs provide a fundamental framework for representing overlaps between sequencing reads and are widely used in genome assembly. In this project, we construct a De Bruijn graph from a given set of short sequencing reads under ideal conditions, where all reads originate from the forward strand and contain no sequencing errors.

Methods Given a fixed k-mer size of $k = 5$, each read is decomposed into all possible k-mers. Graph nodes correspond to unique $(k-1)$-mers, while directed edges correspond to k-mers connecting their prefix and suffix. Formally, for a k-mer $s = s_1 s_2 \dots s_k$, the prefix node is defined as $s_1 \dots s_{k-1}$ and the suffix node as $s_2 \dots s_k$. An edge is added from the prefix node to the suffix node, and the multiplicity of each edge is recorded as the total number of occurrences of the corresponding k-mer. The resulting graph is output in Graphical Fragment Assembly (GFA) format, including only node (S-line) and edge (L-line) records.

Results The constructed De Bruijn graph accurately represents the connectivity of all observed $(k-1)$-mers derived from the input reads. The GFA output is directly visualized using Bandage, producing a clear graphical representation of the graph structure and branching patterns. This visualization enables intuitive inspection of overlaps and potential assembly ambiguity.

Conclusion This project demonstrates a complete workflow for constructing and visualizing a De Bruijn graph from sequencing reads. The results highlight the effectiveness of De Bruijn graphs in encoding read overlaps and provide a foundation for understanding graph-based genome assembly methods.

Key words: De Bruijn graph; genome assembly; k-mer; GFA format; graph visualization

1 RESULTS AND DISCUSSION

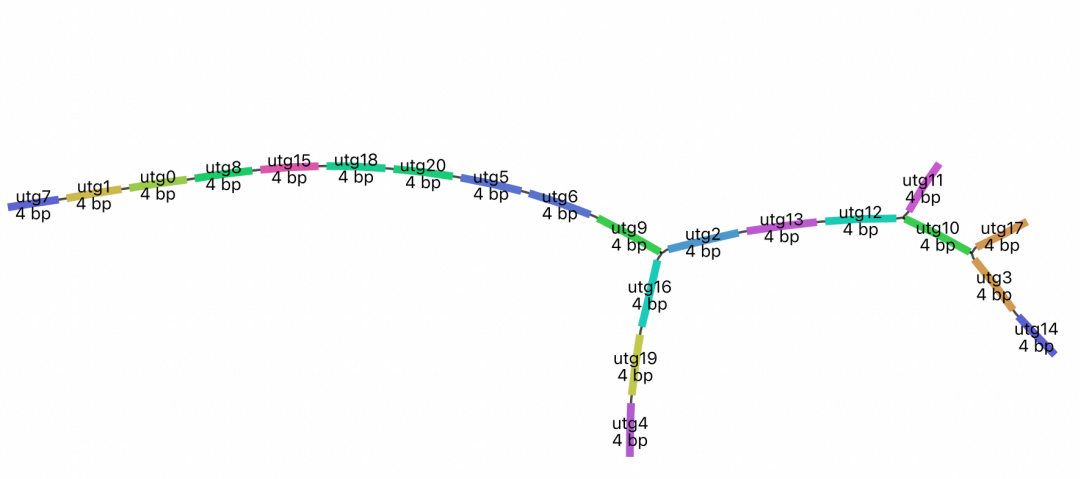

Figure 1: Visualization of the constructed De Bruijn graph generated using Bandage. Each node represents a unique $(k-1)$-mer ($k = 5$), and directed edges indicate overlaps between nodes derived from observed k-mers.

Figure 1 shows the De Bruijn graph constructed from the input sequencing reads. Each node corresponds to a unique $(k-1)$-mer, and directed edges represent overlaps defined by shared k-mers. The graph structure reveals clear linear paths as well as branching points, reflecting alternative overlaps among the reads. Such branching patterns indicate potential assembly ambiguity, which is a characteristic feature of De Bruijn graph-based genome assembly.

2 DATA AND CODE AVAILABILITY

2.1 Data Availability

All data used in this project are simulated sequencing reads provided as part of the course assignment. No real biological or patient-derived data were used in this study.

2.2 Code Availability

The source code used to construct the De Bruijn graph and generate the GFA output is publicly available at the following GitHub repository:

https://github.com/wangyf1125/cbb5800