Computational Molecular Biology and Bioinformatics

MetaGraph

Malay Bhattacharyya

Associate Professor

Machine Intelligence Unit
Indian Statistical Institute, Kolkata

August, 2025

The de Bruijn graphs

Given a sequence s and the k-mer length k, we can construct a de Bruijn graph with the following characteristics:

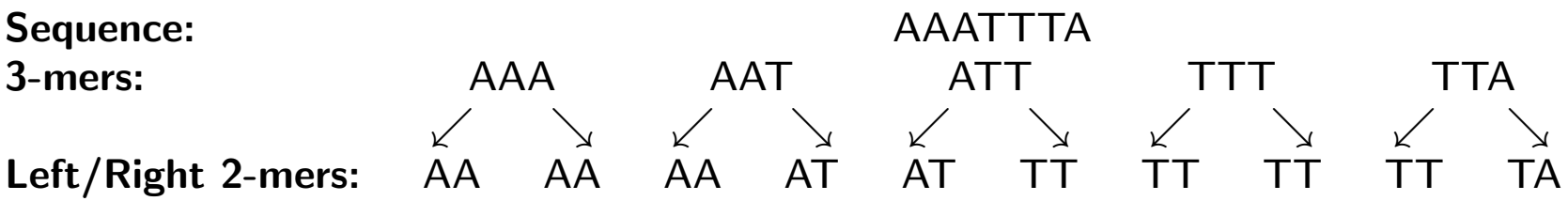
- the nodes are (k-1)-mers, and
- an edge is present between a pair of nodes if they have (k-2)-long overlap.

Consider the sequence `AAATTTA` and the value of $k = 3$. Let us construct the corresponding de Bruijn graph.

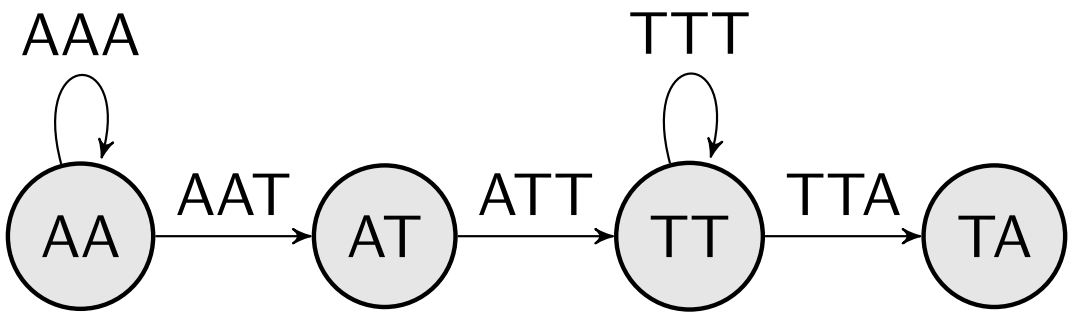
Note: The value of k is taken as odd.

The de Bruijn graphs

We can obtain the left and right (k-1)-mers of each k-mer present in the sequence as follows.

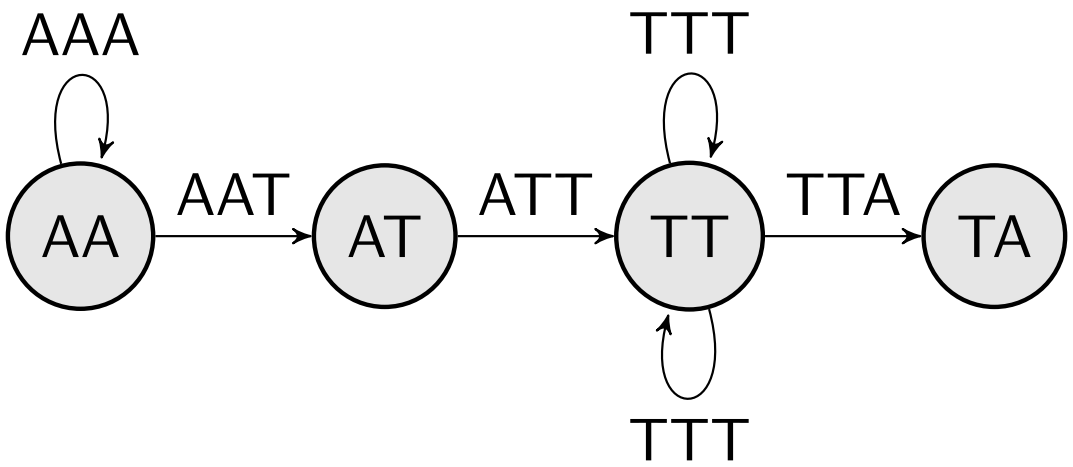


We then construct the corresponding de Bruijn graph as follows.



The de Bruijn graphs

Note that, if we add one more `T` to our input sequence such that s becomes `AAATTTTA`, and reconstruct the corresponding de Bruijn graph, we get a *multiedge* in the graph as follows.



The de Bruijn graphs

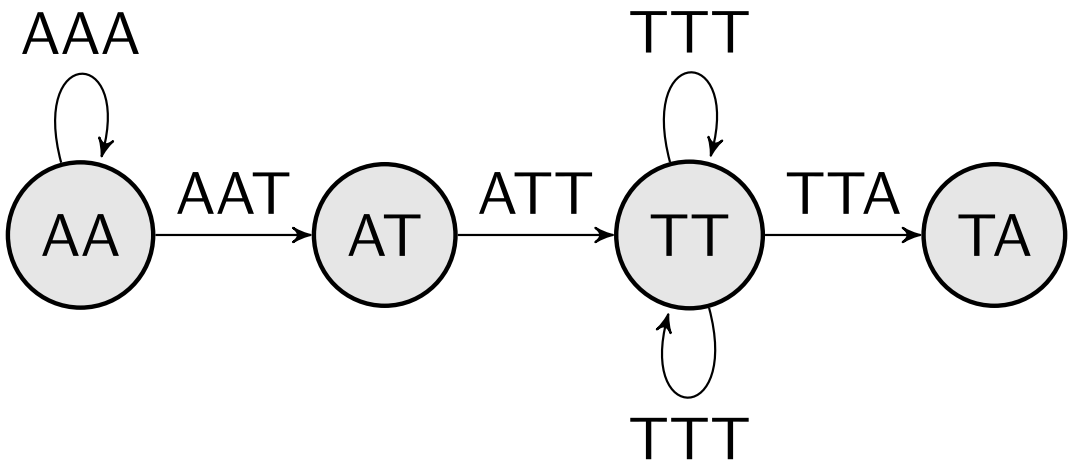
A graph is *connected* if there is a path between every pair of vertex. An *Eulerian walk* visits each edge exactly once. A directed, connected graph is *Eulerian* if it contains an Eulerian walk.

A node is *balanced* if its in-degree = out-degree. A node is *semi-balanced* if its |in-degree - out-degree| = 1. A directed, connected graph is Eulerian if and only if it has at most 2 semi-balanced nodes and all other nodes are balanced.

Note: For simplicity, we do not distinguish Eulerian from semi-Eulerian.

The de Bruijn graphs

Note that, the following graph is Eulerian.



Arguments:

1. AA $\rightarrow$ AA $\rightarrow$ AT $\rightarrow$ TT $\rightarrow$ TT $\rightarrow$ TA
2. AA and TA are semi-balanced, AT and TT are balanced

The de Bruijn graphs

For genome assembly, each k-mer is recorded in *twin* nodes – one node in forward direction and one node in reverse complement.

With perfect sequencing, this procedure always yields an Eulerian graph unless the genome is circular. Hence, we just need to find out the Eulerian walk in the graph.

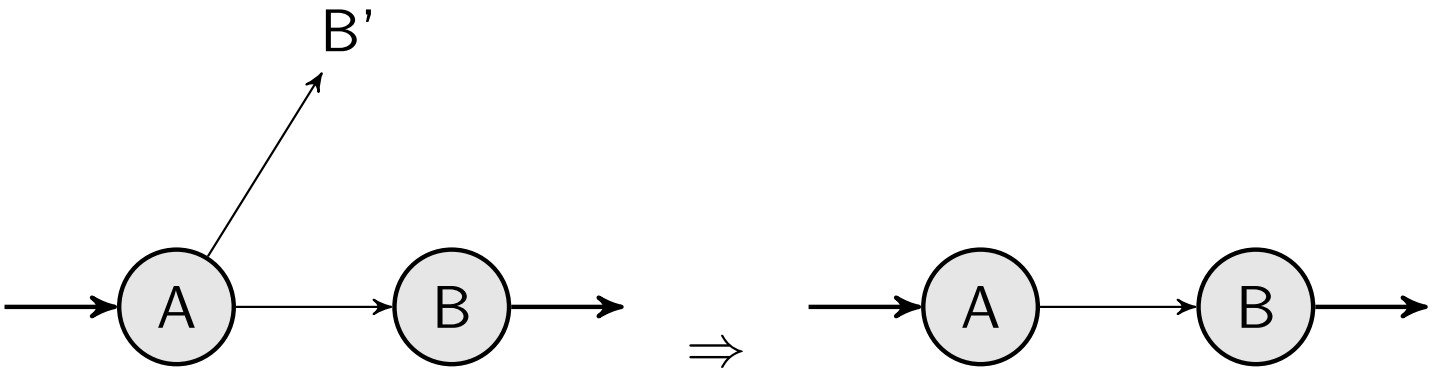
Note: No node can be its own reverse complement because k is odd.

The de Bruijn graphs

Note that, there are further challenges to deal with as listed below.

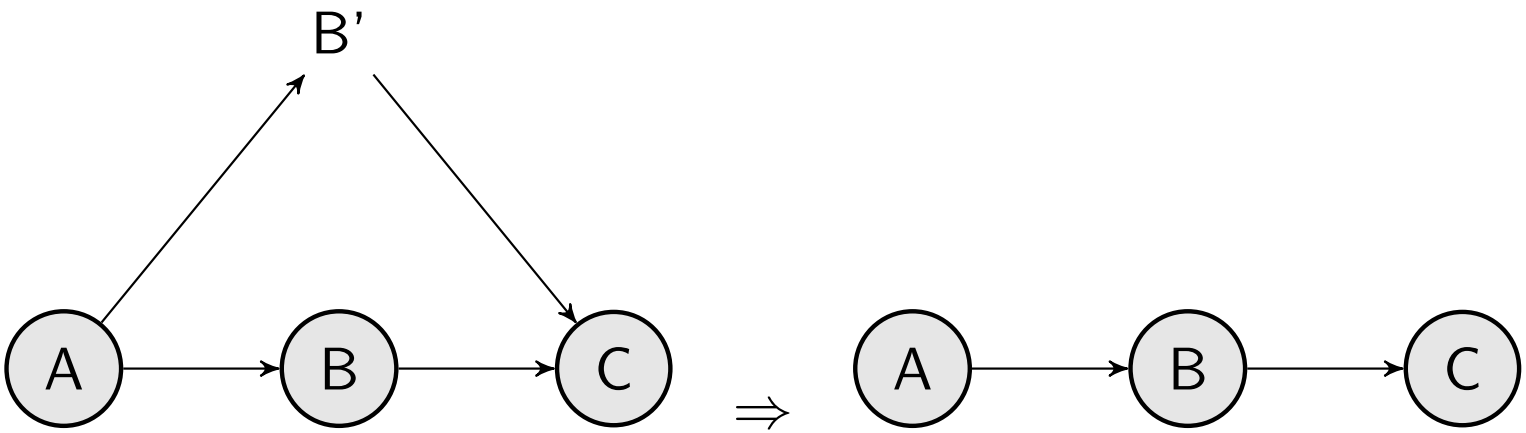
1. A de Bruijn graph can have multiple Eulerian walks, *only one* of which corresponds to original superstring.
2. Having gaps in the coverage can lead to a disconnected de Bruijn graph. The connected components are individually Eulerian, but not the overall graph.
3. Differences in coverage might lead to non-Eulerian de Bruijn graphs.
4. Errors and differences between chromosomes might lead to non-Eulerian de Bruijn graphs.

Refinement of de Bruijn graphs



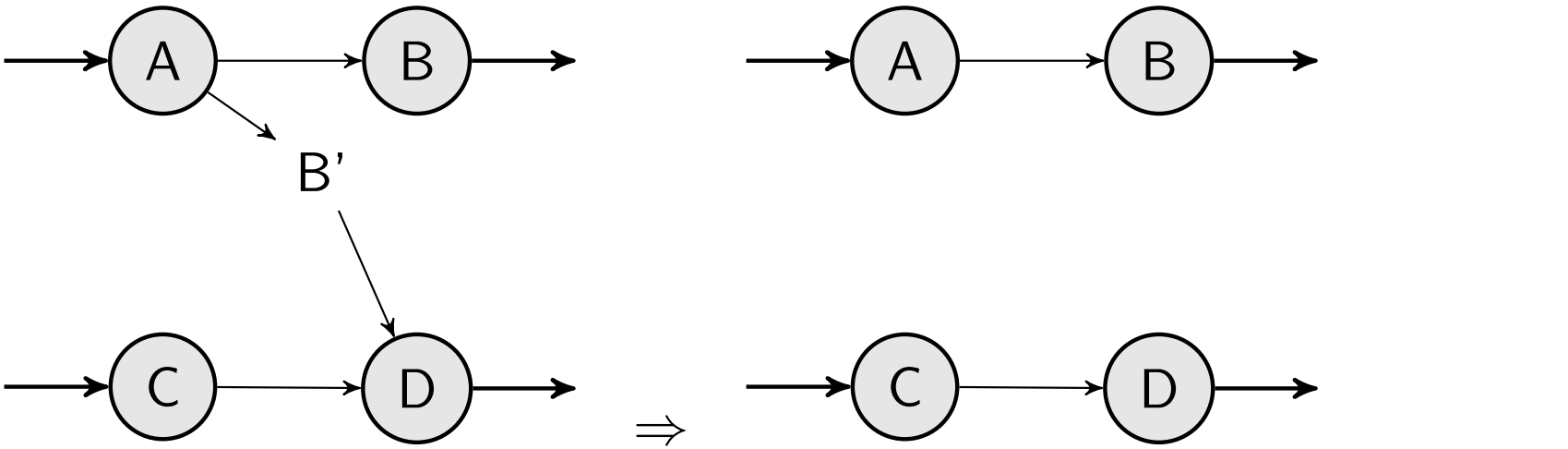
If errors are present at the end of read, remove ‘dead-end’ tips

Refinement of de Bruijn graphs



If errors are present in the middle of read, remove 'bubbles'

Refinement of de Bruijn graphs



If chimeric edges are present, remove short and low coverage nodes

Limitations of de Bruijn graphs

The de Bruijn graphs suffer from the following limitations.

- It cannot resolve repeats as well as overlap graphs because reads are immediately split into shorter k-mers.
- It cannot deal with complicated errors because only a specific type of overlap is considered.
- Some paths through de Bruijn graphs are inconsistent with respect to input reads. Hence, the read coherence is lost.

What is MetaGraph?

MetaGraph is a methodological framework that enables us to scalably index large sets of DNA, RNA or protein sequences using annotated de Bruijn graphs [1].

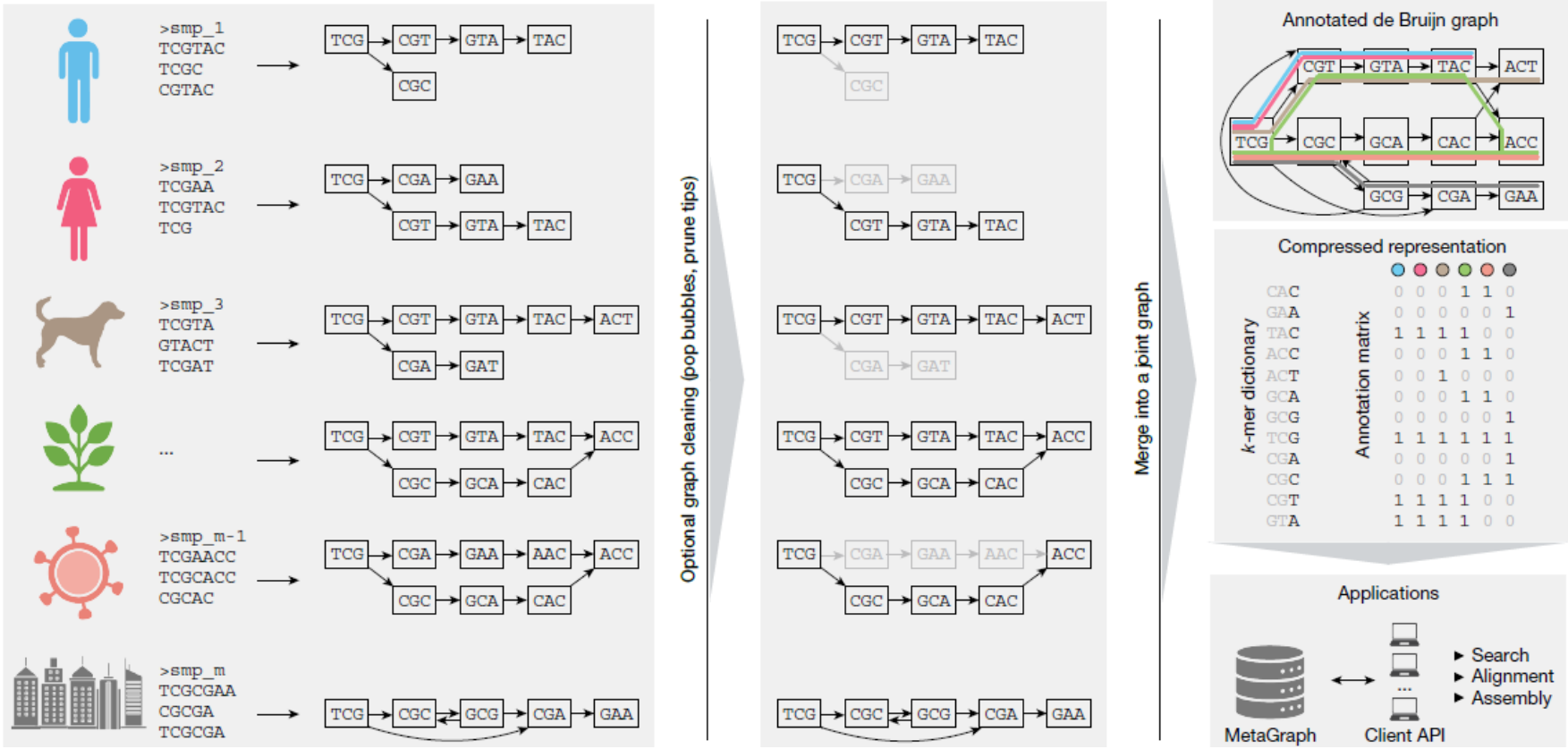
MetaGraph can index biological sequences of all kinds, such as raw DNA-sequencing and RNA-sequencing (RNA-seq) reads, assembled genomes and amino acid sequences.

Representing de Bruijn graphs

MetaGraph provides several data structures for storing k-mer sets, which are used as a basis to implement different representations of the de Bruijn graph abstraction.

In addition to the simple hash table, the k-mers may be stored in an indicator bitmap [2] (a binary vector represented as a succinct bitmap of size $|\Sigma|^k$ indicating which k-mers are present in the set) or in the BOSS table [3]. Hence, the implementations based on the data structures HashDBG, BitmapDBG and SuccinctDBG together.

The MetaGraph framework



References

1. Karasikov, M., Mustafa, H., Danciu, D., Kulkov, O., Zimmermann, M., Barber, C., Rätsch, G. and Kahles, A., Efficient and accurate search in petabase-scale sequence repositories. Nature, In press, 2025.
2. Conway, T.C. and Bromage, A.J., Succinct data structures for assembling large genomes. Bioinformatics, 27(4):479-486, 2011.
3. Bowe, A., Onodera, T., Sadakane, K. and Shibuya, T., Succinct de Bruijn graphs. In International workshop on algorithms in bioinformatics (pp. 225-235). Berlin, Heidelberg: Springer Berlin Heidelberg, September 2012.