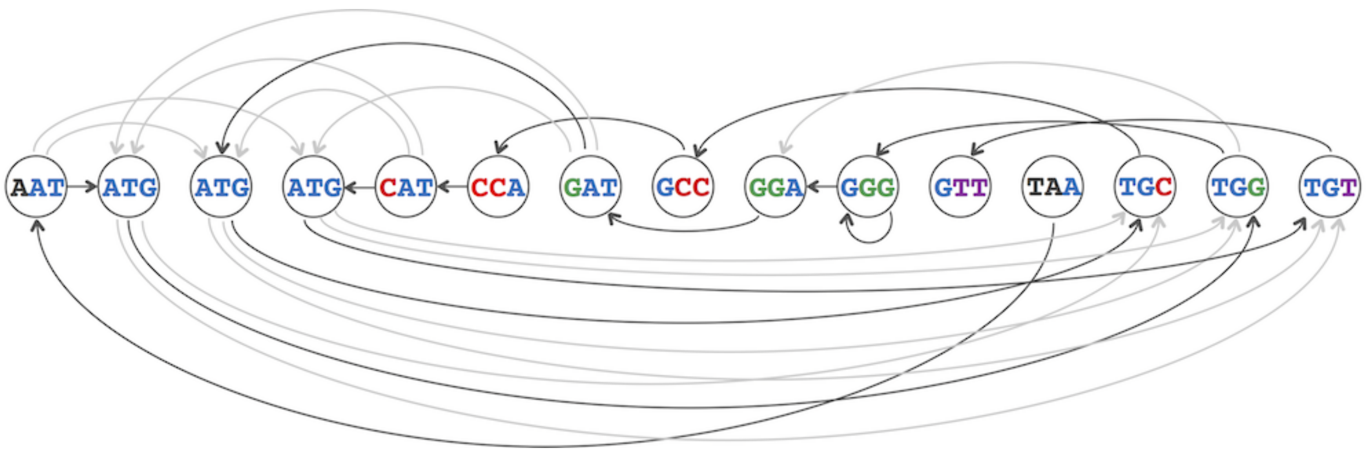
CS4054
Bioinformatics

Spring 2025

Rushda Muneer

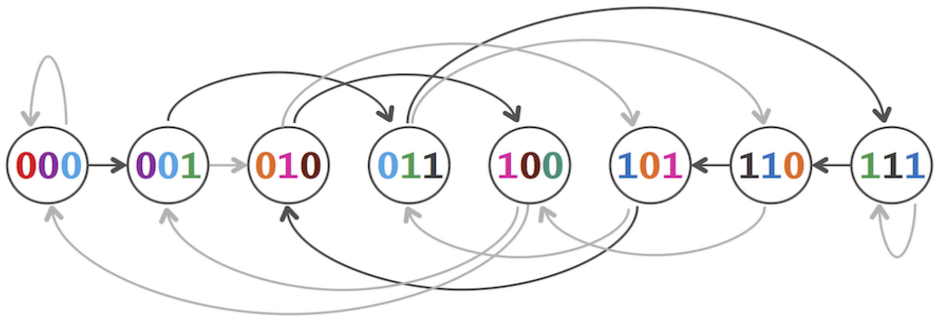
Hamiltonian Path and k-Universal Strings

- A path in a graph visiting every node once is called a **Hamiltonian path**
- Two Hamiltonian paths constructing universal strings
  - "**TAATGGGATGCCATGTT**"
  - "**TAATGCCATGGGATGTT**"



de Bruijn's approach to k-universal

- Nicolaas de Bruijn, a Dutch mathematician (1946)
- A **binary string** is a string composed only of 0's and 1's
- A binary string is ***k*-universal** if it contains every binary *k*-mer exactly once.
- "0001110100" is a 3-universal string, as it contains each of the eight binary 3-mers ("000", "001", "011", "111", "110", "101", "010", and "100") exactly once.



- Finding a *k*-universal string is equivalent to solving the String Reconstruction Problem
- Finding a *k*-universal string can be reduced to finding a Hamiltonian path in the overlap graph formed on all binary *k*-mers

Gluing nodes and de Bruijn graphs

- Representing genome "**TAATGCCATGGGATGTT**" as a sequence of 3-mers.

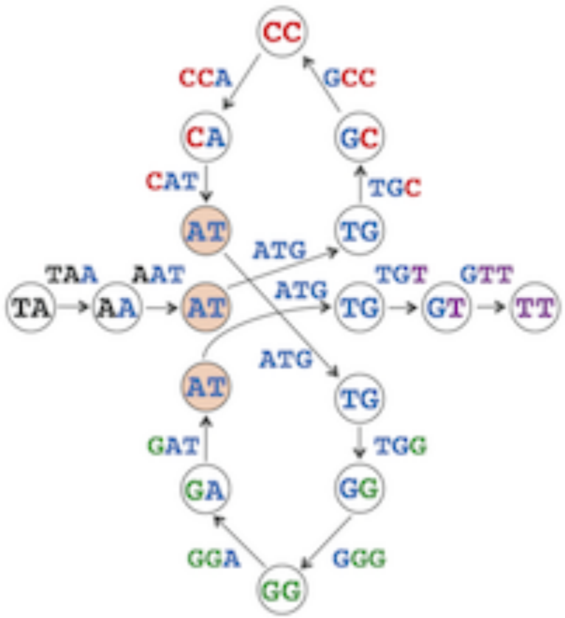
"TAA" "AAT" "ATG" "TGC" "GCC" "CCA" "CAT" "ATG" "TGG" "GGG" "GGA" "GAT" "ATG" "TGT" "GTT"

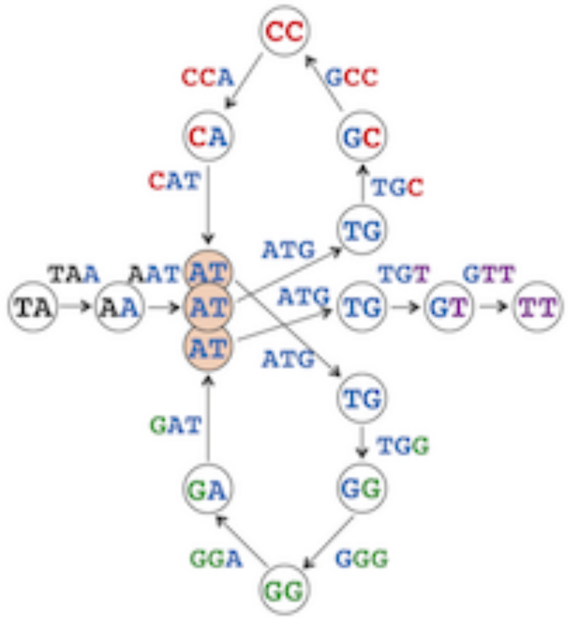
- Instead of assigning 3-mers to nodes, we assign them to edges.

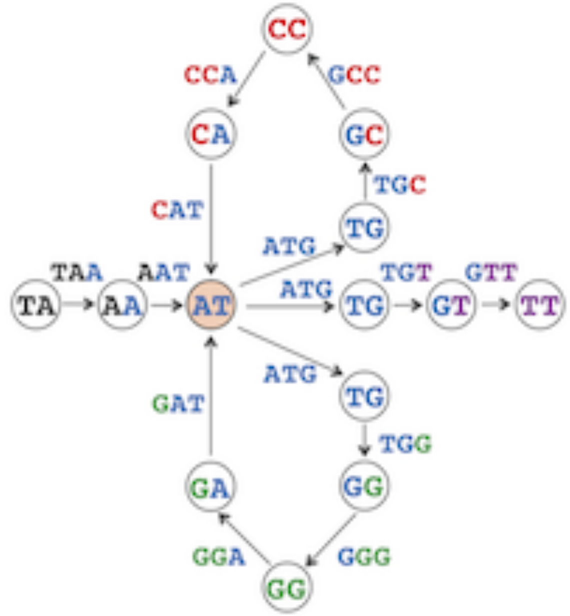
- **Graph Representation:**
  - Nodes are labeled by 2-mers representing overlaps between consecutive 3-mers.
  - Example: The node between "CAT" and "ATG" is labeled "AT".
  - Genome can be reconstructed by following the path from left to right.

de Bruijn Graph for String Reconstruction

- We start **gluing** identically labeled nodes
- We bring the three "**AT**" nodes closer and closer to each other until they have been glued into a single node.

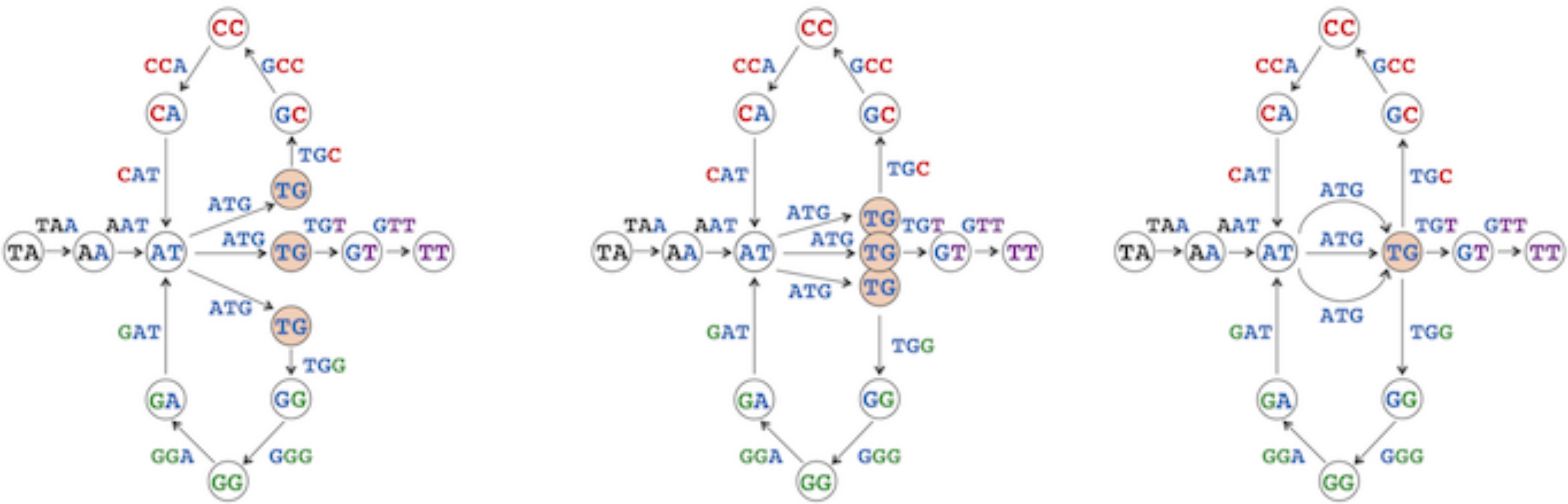






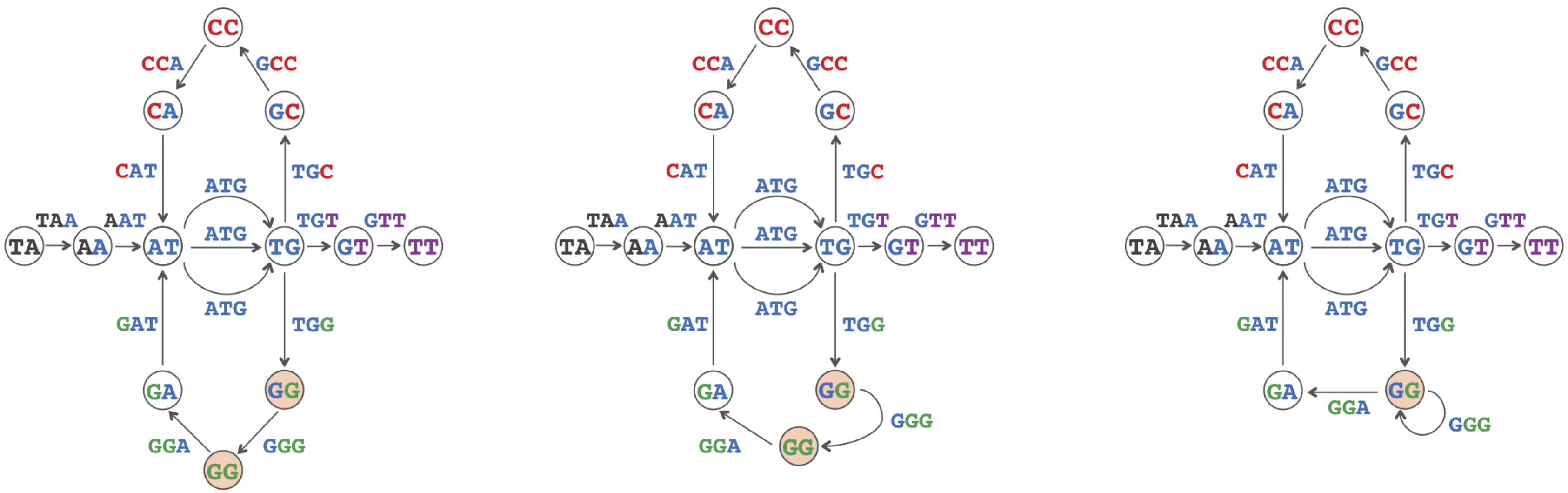
de Bruijn Graph for String Reconstruction

- There are also three nodes labeled by "**TG**", which we bring closer and closer to each other in the figure below until they are glued into a single node.



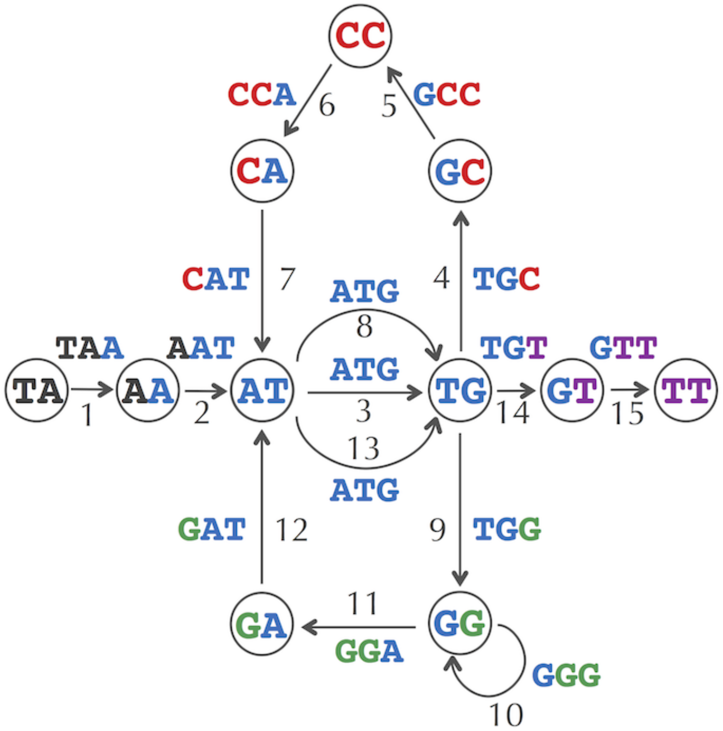
de Bruijn Graph for String Reconstruction

- Finally, we glue together the two nodes labeled "GG” which produces a special type of edge called a **loop** connecting "GG" to itself.
- The number of nodes in the resulting graph has reduced from sixteen to eleven, while the number of edges stayed the same.
- This graph is called the **de Bruijn graph** of **"TAATGCCATGGGATGTT”** denoted as *DeBruijn*$_3$("**TAATGCCATGGGATGTT**").



Eulerian paths

- Solving the String Reconstruction Problem reduces to finding a path in the de Bruijn graph that visits every ***edge*** exactly once. Such a path is called an **Eulerian Path** in honor of the great mathematician Leonhard Euler (pronounced “oiler").



TAATGCCATGGGATGTT

Constructing de Bruijn graphs from *k*-mer composition

- Given a collection of *k*-mers *Patterns*, the nodes of ***DeBruijn$_k$*(*Patterns*)** are simply all unique (*k*−1)-mers occurring as a **prefix** or **suffix** in ***Patterns***.
- We are given the following collection of 3-mers:

"TAA" "AAT" "ATG" "TGC" "GCC" "CCA" "CAT" "ATG"
"TGG" "GGG" "GGA" "GAT" "ATG" "TGT" "GTT"

- The set of eleven *unique* 2-mers occurring as a prefix or suffix of 3-mers in this collection is as follows:

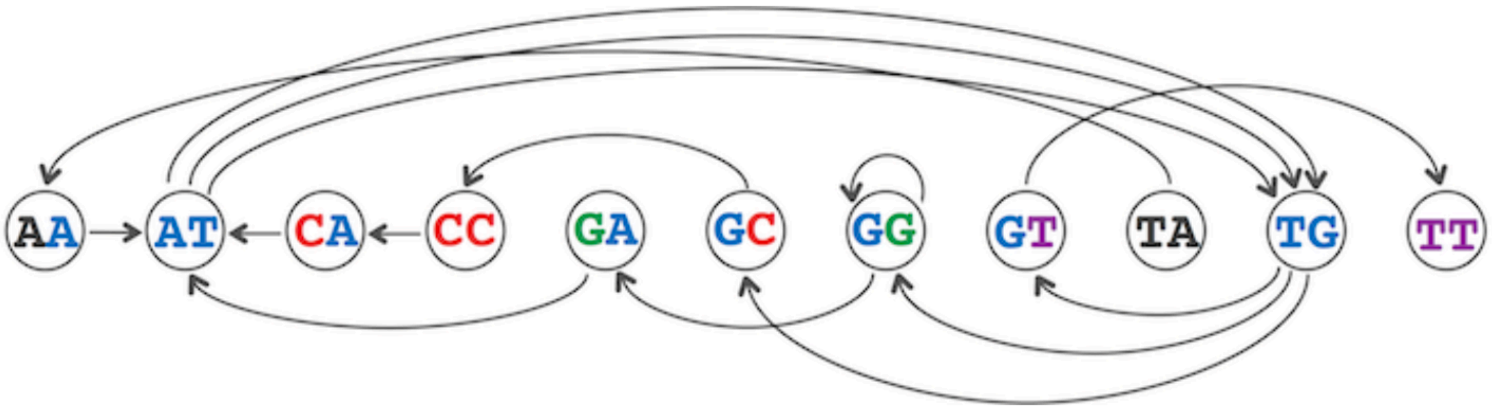
"AA" "AT" "CA" "CC" "GA" "GC" "GG" "GT" "TA" "TG" "TT"

Constructing de Bruijn graphs from *k*-mer composition

- From the given 2-mers:

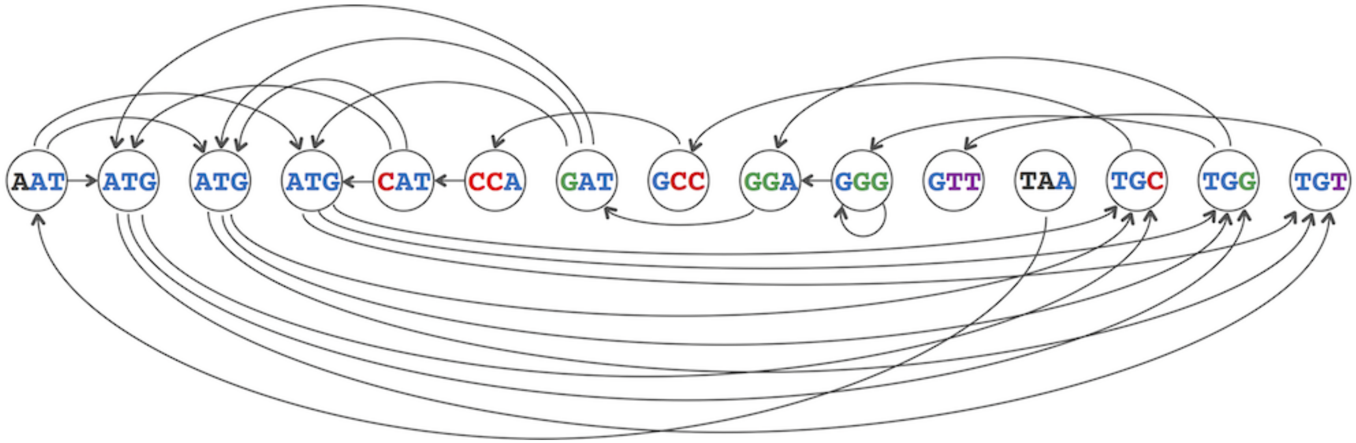
"AA" "AT" "CA" "CC" "GA" "GC" "GG" "GT" "TA" "TG" "TT"

- For every *k*-mer in *Patterns*, we connect its **prefix node** to **its suffix node** by a **directed edge** in order to produce *DeBruijn*(*Patterns*).



De Bruijn graphs versus Overlap Graphs

- We now have two ways of solving the String Reconstruction Problem.
- Find a Hamiltonian path in the overlap graph



- Find an Eulerian path in the de Bruijn graph