AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

22BIO211: Intelligence of Biological Systems - 2

CONSTRUCTING DE BRUIJN GRAPHS

Dr. Manjusha Nair M
Amrita School of Computing, Amritapuri

Email : manjushanair@am.amrita.edu
Contact No: 9447745519

Two ways of solving the String Reconstruction Problem

Hamiltonian Path Problem: Use Overlap Graph
Finding a path visiting every node exactly once

Eulerian Path Problem : Use De Bruijn Graph
Finding a path visiting every edge exactly once

Which graph would you rather work with, the overlap graph or the de Bruijn graph???

Why Eulerian path and why not Hamiltonian path?

Hamiltonian path Problem is NP hard. Eulerian Path problem is Polynomial time

Why de Bruijn ?
More compact representation of the graph. Can walk using Eulerian path.

de Bruijn Graphs

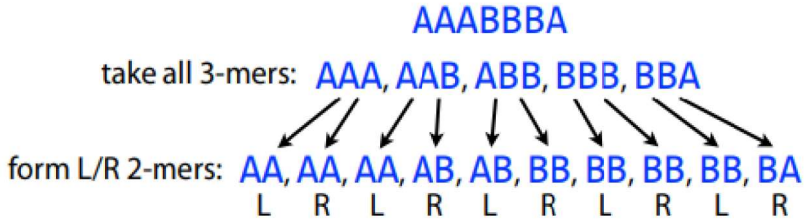
Nicolaas de Bruijn

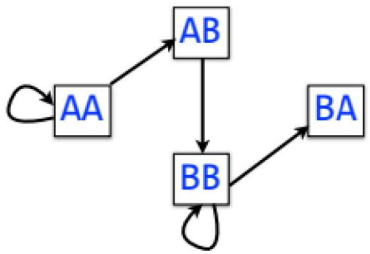
- Directed graph representing overlaps between sequences of symbols.
- Vertices/nodes in the graph are k-1 mers.
- Edges represent consecutive k-mers (which overlap by k-1 symbols).
- **de Bruijn Graph** is a convenient means of reconstructing the original sequence.

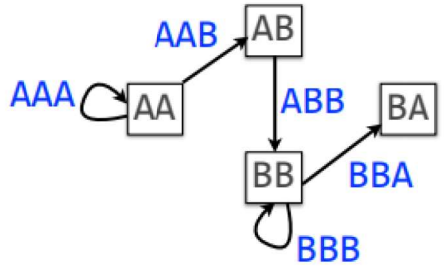
Constructing de Bruijn Graphs

- *Step1*: Construct all the k-mers
- Step 2 - take all (k-1)-mers (Suffix and Prefix) from the set of k-mers.

  We should have double the size of k-mer reads.
- Step 3: construct a graph with nodes being unique k-1-mers;
- Step 4: draw an edge between two k-1 mers only if the two k-1 mers are taken from the same read.
- Step 5- Represent the K- mer as the edge label







Let 2-mers be nodes in a new graph. Draw a directed edge from each left 2-mer to corresponding right 2-mer:

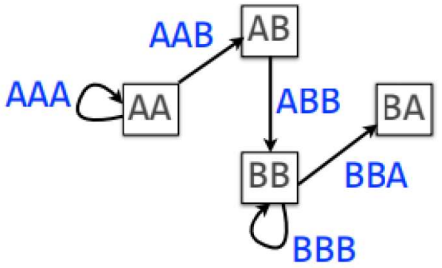
Each edge in this graph corresponds to a length-3 input string

Constructing de Bruijn Graphs

AAABBBA

K-mer	Prefix	Suffix
AAA	AA	AA
AAB	AA	AB
ABB	AB	BB
BBB	BB	BB
BBA	BB	BA

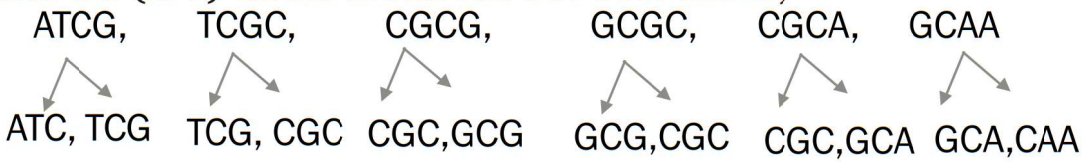
Nodes	Connected Nodes	Edge weight
AA	AA,AB	AAA,AAB
AB	BB	ABB
BB	BB,BA	BBB,BBA
BA		




```
graph = { "AA" : [("AA","AAA),("AB","AAB")],
          "AB" : [("BB", "ABB")],
          "BB" : [("BB", "BBB"), ("BA", "BBA")],
          "BA" : []
        }
```

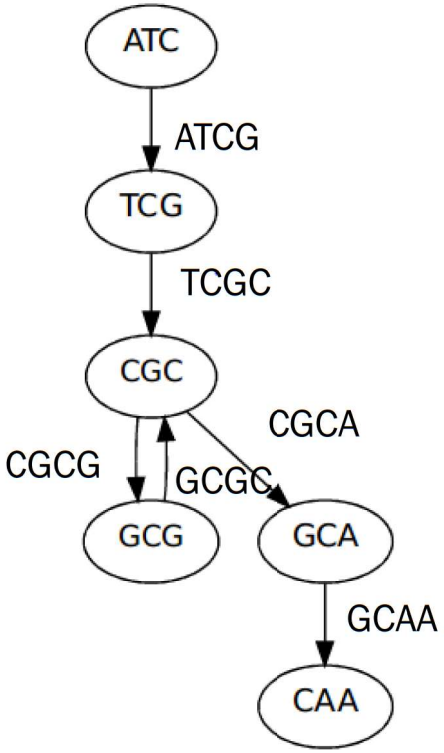
Constructing de Bruijn Graphs : Another example

- Text : "ATCGCGCAA"
- Set of k- mers : ATCG, TCGC, CGCG, GCGC, CGCA, GCAA.
- 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.

Hamiltonian path does not exist: We need to trace an Eulerian trial



de Bruijn Graph

Construct de Bruijn graphs

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

Given a collection of k-mers Patterns

AA AT

TG GT

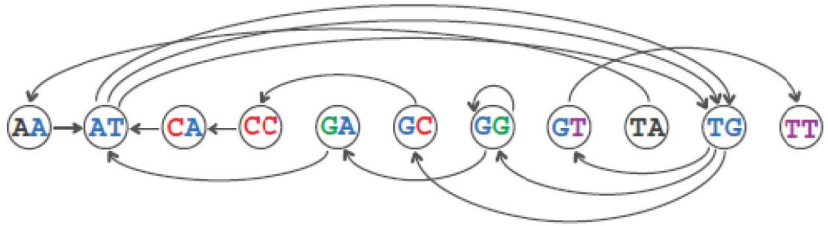
Nodes of DeBruijn$_k$(Patterns)

All unique (k -1) mers occurring as a prefix or suffix of 3-mers in Patterns

AA AT CA CC GA GC GG GT TA TG TT

Set of eleven unique 2-mers

Connect its prefix node to its suffix node by a directed edge in order to produce DeBruijn(Patterns).



de Bruijn Graphs – Construction using Gluing nodes

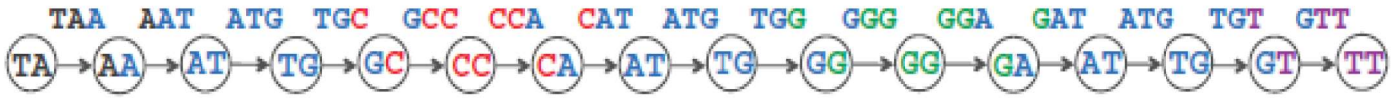
Genome TAATGCCATGGGATGTT

Sequence of 3-mers TAA AAT ATG TGC GCC CCA CAT ATG TGG GGG GGA GAT ATG TGT GTT

Edges Instead of assigning these 3-mers to nodes, we will assign them to edges

Vertices Label each node of this graph with a 2-mer representing the overlapping nucleotides shared by the edges on either side of the node.

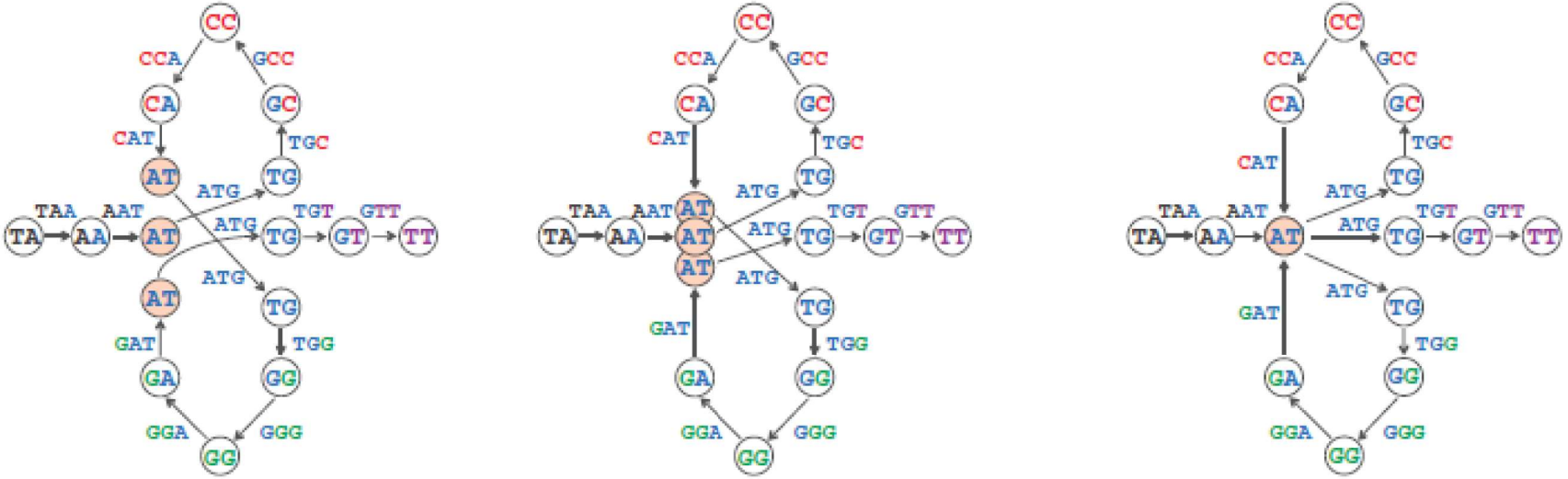
PathGraph$_3$(Genome)



For example, the node with incoming edge **CAT** and outgoing edge **ATG** is labeled **AT**.

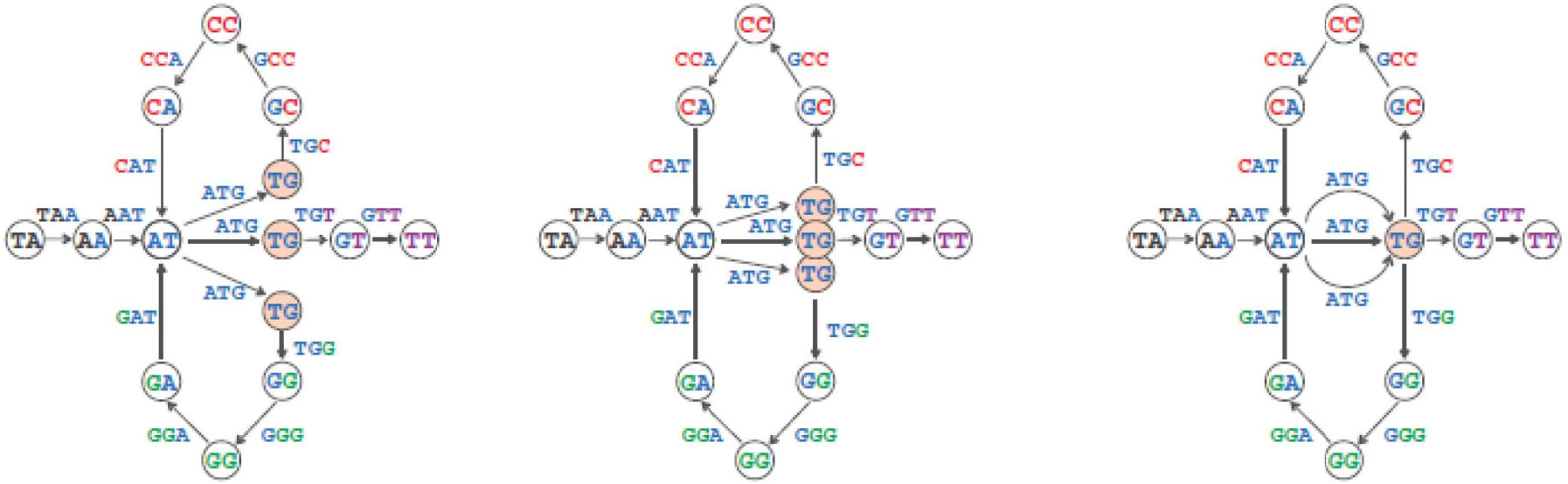
Gluing nodes and de Bruijn Graphs

Start **gluing** identically labeled nodes



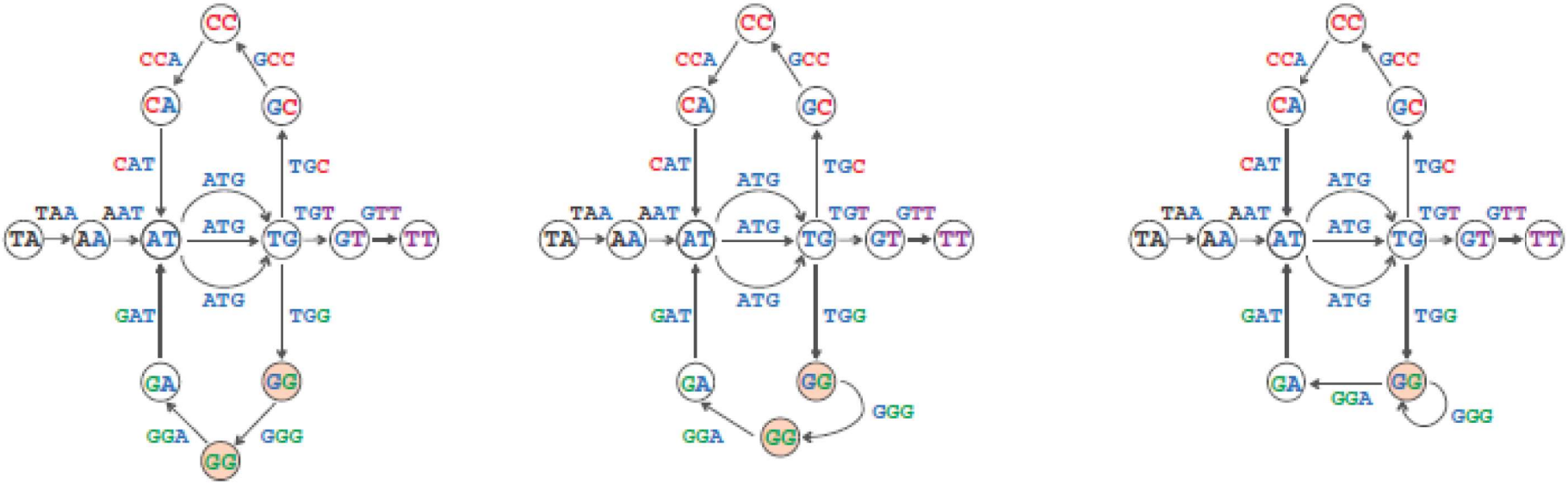
Bring the three **AT** nodes closer and closer to each other until they have been glued into a single node.

Gluing nodes and de Bruijn Graphs



Bringing the three nodes labeled **TG** closer (left) and closer (middle) to each other to eventually glue them into a single node (right).

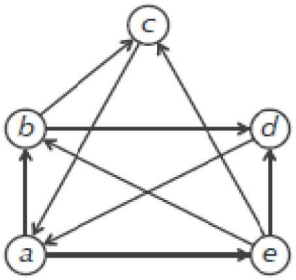
Gluing nodes and de Bruijn Graphs



Bringing the two nodes labeled GG closer (left) and closer (middle) to each other to eventually glue them into a single node (right).

Gluing nodes affect Graph Representations

Graph



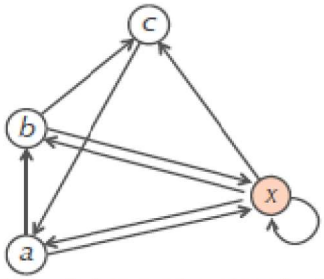
Adjacency Matrix

	a	*b*	*c*	*d*	*e*
a	0	1	0	0	1
b	0	0	1	1	0
c	1	0	0	0	0
d	1	0	0	0	0
e	0	1	1	1	0

Adjacency List

a is adjacent to *b* and *e*
b is adjacent to *c* and *d*
c is adjacent to *a*
d is adjacent to *a*
e is adjacent to *b*, *c*, and *d*

Graph



Adjacency Matrix

	a	*b*	*c*	*x*
a	0	1	0	1
b	0	0	1	1
c	1	0	0	0
x	1	1	1	1

Adjacency List

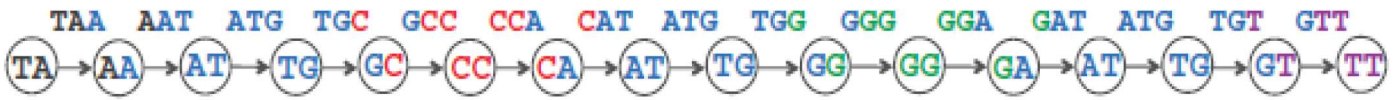
a is adjacent to *b* and *x*
b is adjacent to *c* and *x*
c is adjacent to *a*
x is adjacent to *a*, *b*, *c*, and *x*

The graph produced by gluing nodes *d* and *e* into a single node *x*, along with the new graph's adjacency matrix and adjacency list.

The effect of gluing on the adjacency matrix

Genome TAATGCCATGGGATGTT

PathGraph$_3$(Genome)



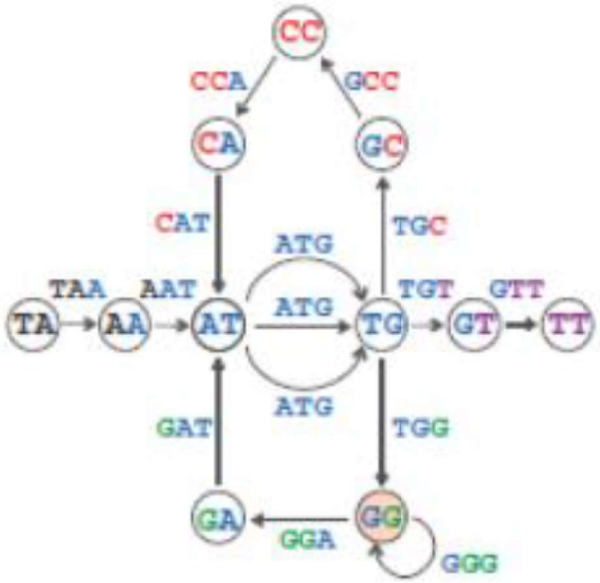
Adjacency Matrix

	TA	AA	AT_1	TG_1	GC	CC	CA	AT_2	TG_2	GG_1	GG_2	GA	AT_3	TG_3	GT	TT
TA	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AA	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
AT_1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
TG_1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
GC	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
CC	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
CA	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
AT_2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
TG_2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
GG_1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
GG_2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
GA	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
AT_3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
TG_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
GT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
TT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16 X 16 adjacency matrix

The effect of gluing on the adjacency matrix

deBrnijn$_3$(Genome)

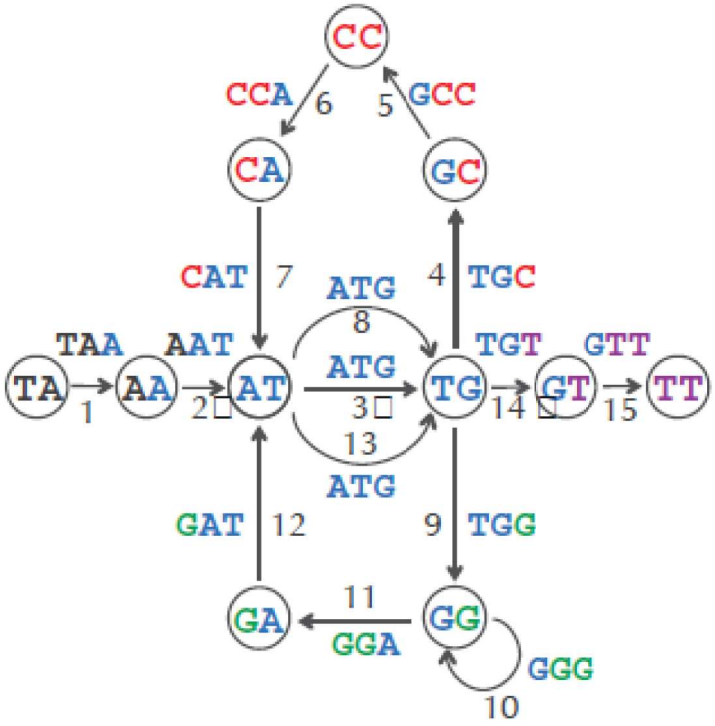


Adjacency Matrix

	TA	AA	AT	TG	GC	CC	CA	GG	GA	GT	TT
TA	0	1	0	0	0	0	0	0	0	0	0
AA	0	0	1	0	0	0	0	0	0	0	0
AT	0	0	0	3	0	0	0	0	0	0	0
TG	0	0	0	0	1	0	0	1	0	1	0
GC	0	0	0	0	0	1	0	0	0	0	0
CC	0	0	0	0	0	0	1	0	0	0	0
CA	0	0	1	0	0	0	0	0	0	0	0
GG	0	0	0	0	0	0	0	1	1	0	0
GA	0	0	1	0	0	0	0	0	0	0	0
GT	0	0	0	0	0	0	0	0	0	0	1
TT	0	0	0	0	0	0	0	0	0	0	0

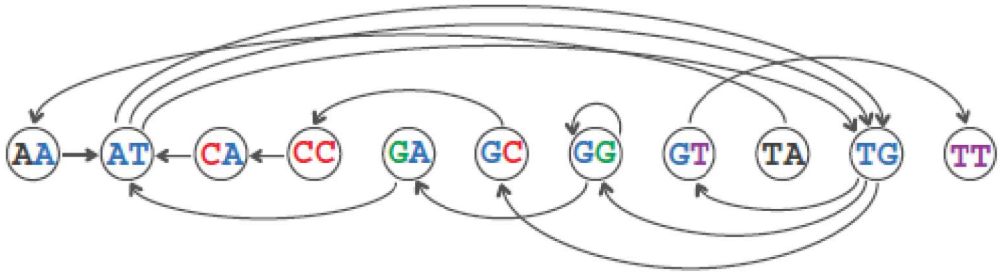
11 X 11 adjacency matrix

Two de Bruijn graphs created



DEBRUIJN$_3$(TAATGCCATGGGATGTT)
With gluing

Both the de Bruijn graphs are the same, although it has been drawn Differently.



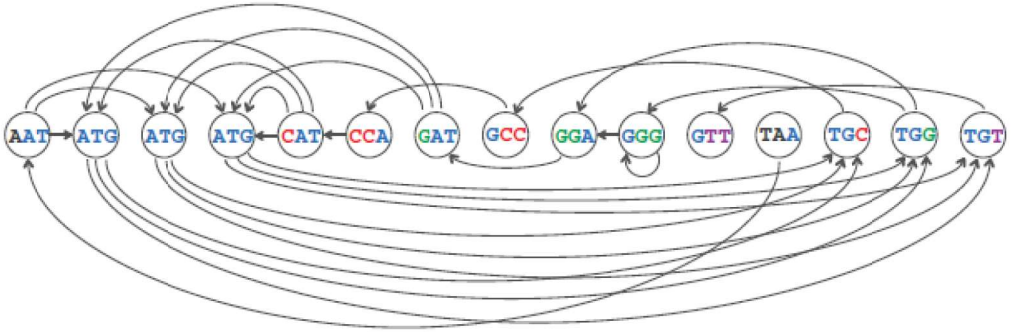
DEBRUIJN$_3$(TAATGCCATGGGATGTT)
Without gluing

Two ways of solving the String Reconstruction Problem

String Reconstruction Problem

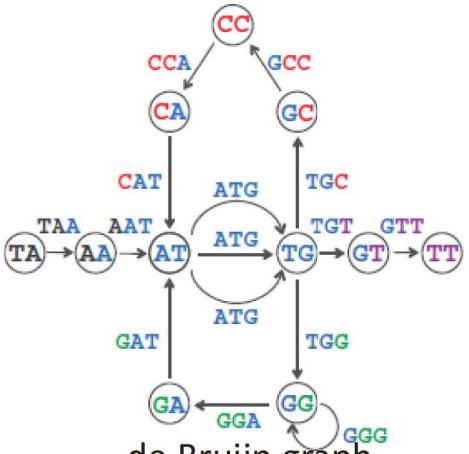
Find a Hamiltonian path in the overlap graph

Find an Eulerian path in the de Bruijn graph



The overlap graph

Text = TAATGCCATGGGATGTT



de Bruijn graph

Summary

- Two ways of solving the String Reconstruction Problem
- de Bruijn Graphs
- Constructing de Bruijn Graph
- de Bruijn Graphs – Construction using Gluing nodes
- Gluing nodes affect Graph Representations