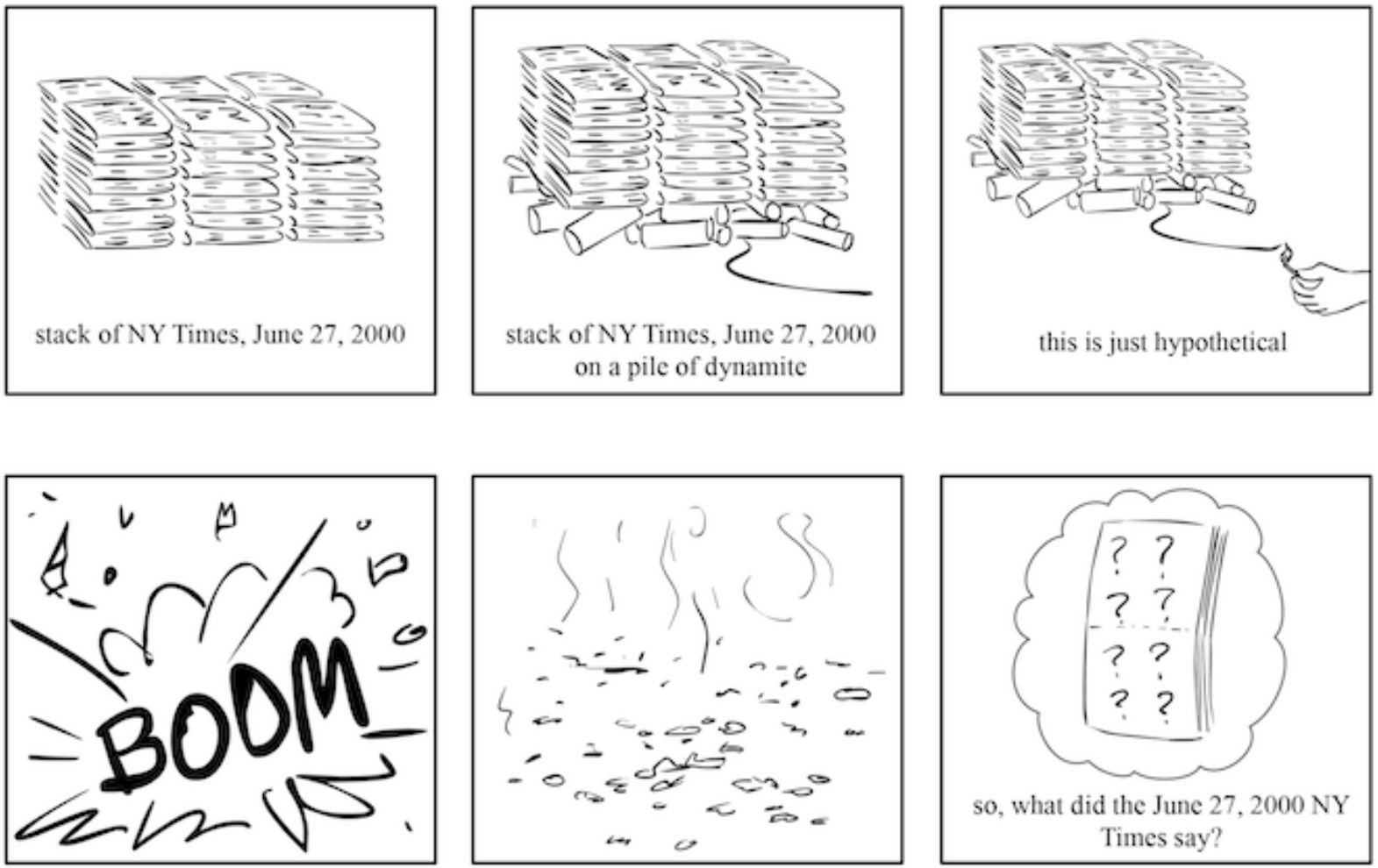
Genome Assembly

Swakkhar Shatabda

CSE 6153: Bioinformatics and Computational Biology, Summer 2022
Department of Computer Science and Engineering
United International University

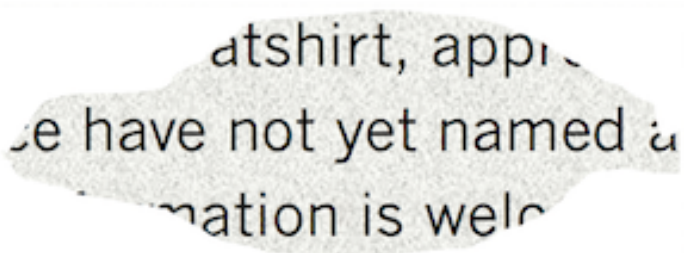
Genome Replication

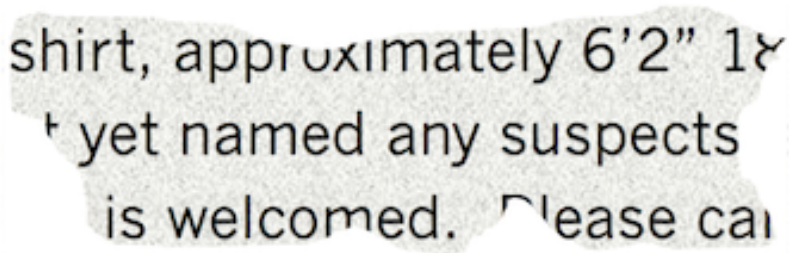
Newspaper Explodes



Newspaper Explodes

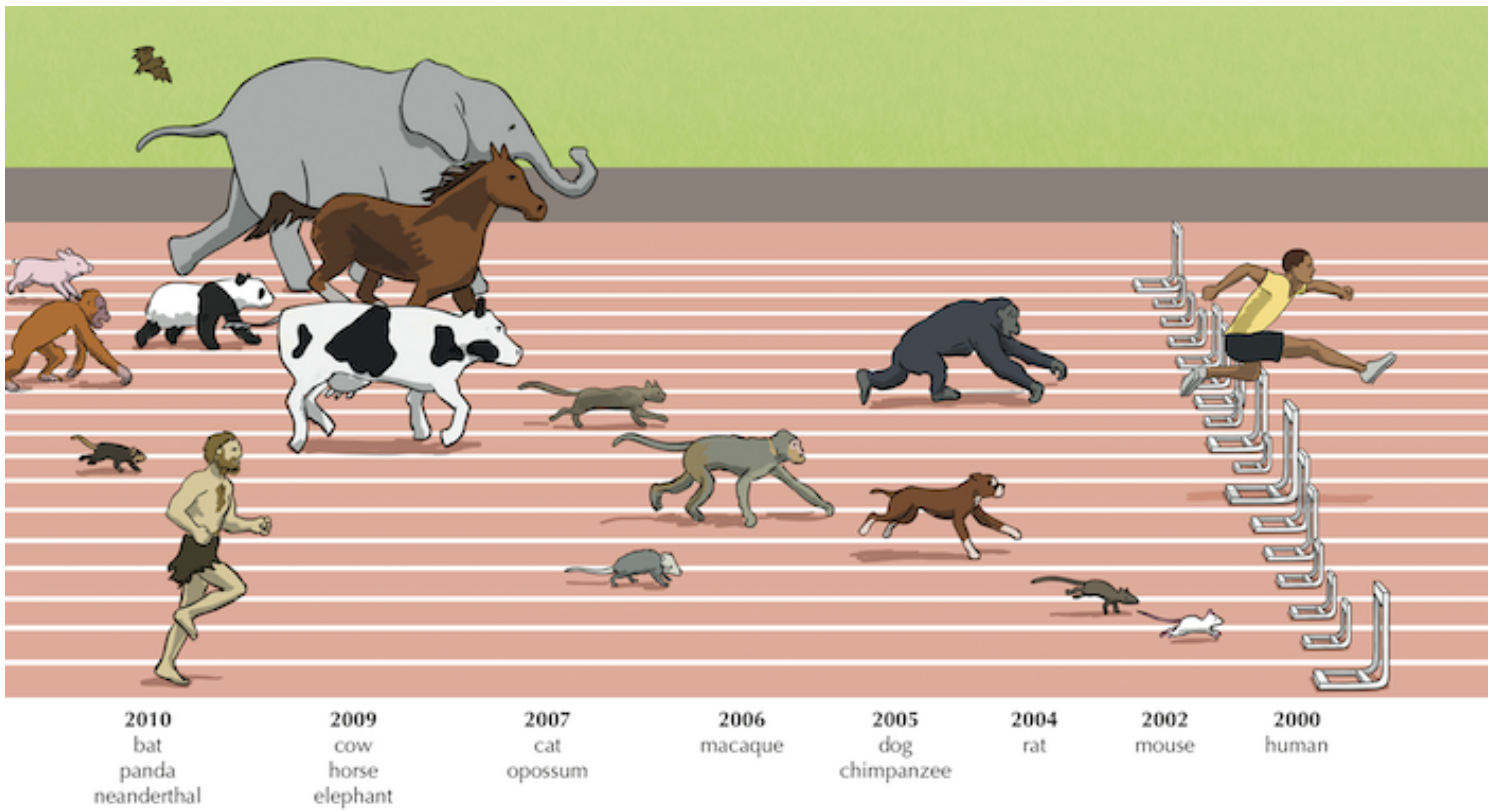
1. Overlapping Information
2. Lost information





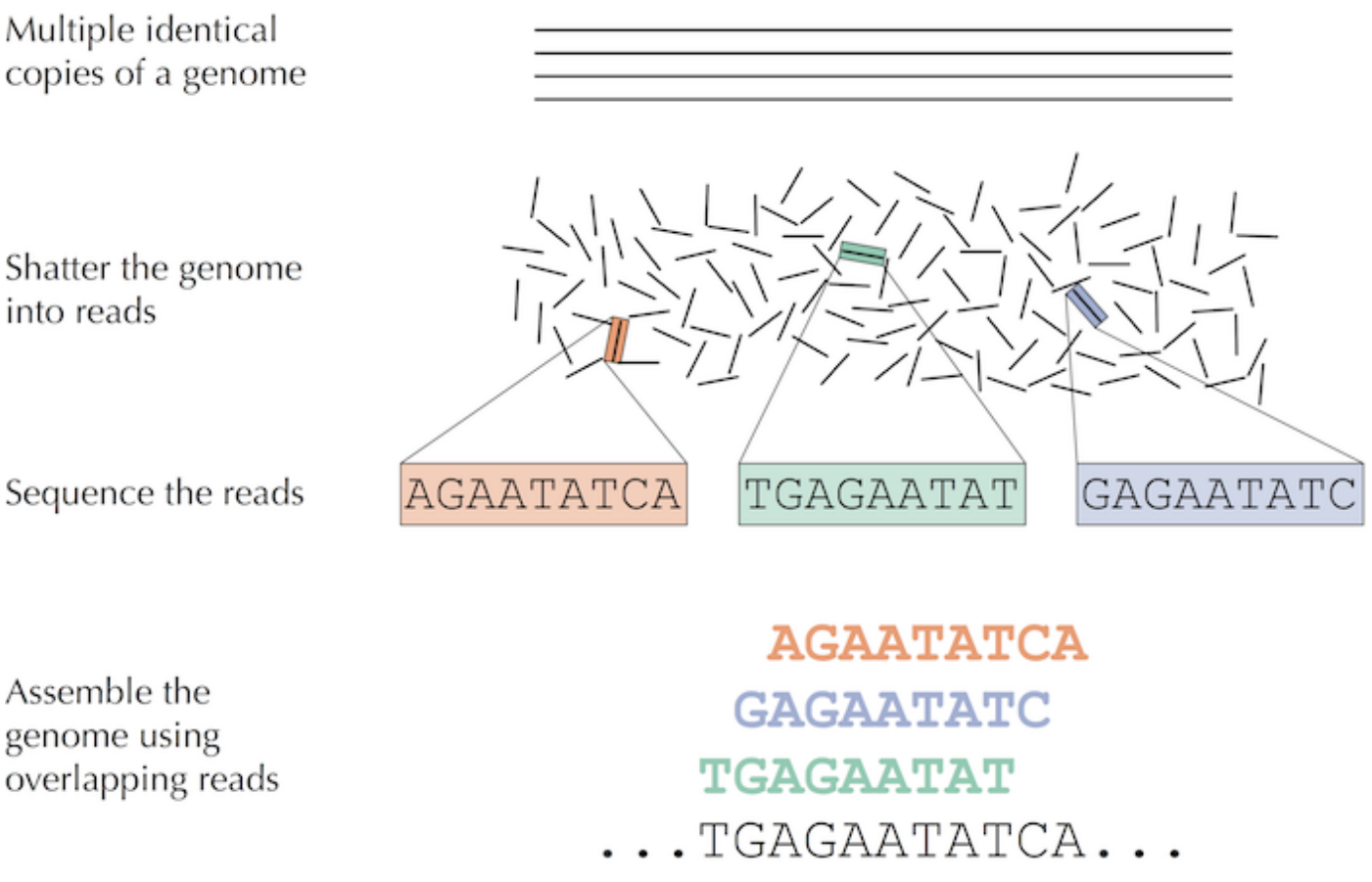
but what do exploding newspapers have to do with biology?

Genome Sequencing



Its possible only to sequence short **reads**.

Genome Assembly



Difficulties in Genome Assembly

- Reverse/Forward? which strand it is reading?
- Modern sequencing machines are not perfect, and the reads that they generate often contain errors.
- Some regions of the genome may not be covered by any reads

Assumptions:

1. Reads generated by modern sequencers often have the same length, we may safely assume that reads are all k-mers for some value of k.
2. All reads come from the same strand.
3. Have no errors, and exhibit perfect coverage

String Composition

Given a string *Text*, its *k*-mer composition $Composition_k(Text)$ is the collection of all *k*-mer substrings of *Text* (including repeated *k*-mers). $Composition_3(TATGGGGTGC) = \{ATG, GGG, GGG, GGT, GTG, TAT, TGC, TGG\}$

The problem

Solve the String Composition Problem.

1. **Input:** An integer *k* and a string *Text*.
2. **Output:** $Composition_k(Text)$, where the *k*-mers are written in lexicographic order.

ROSALIND: https://rosalind.info/problems/ba3a/

String Reconstruction Problem

The problem

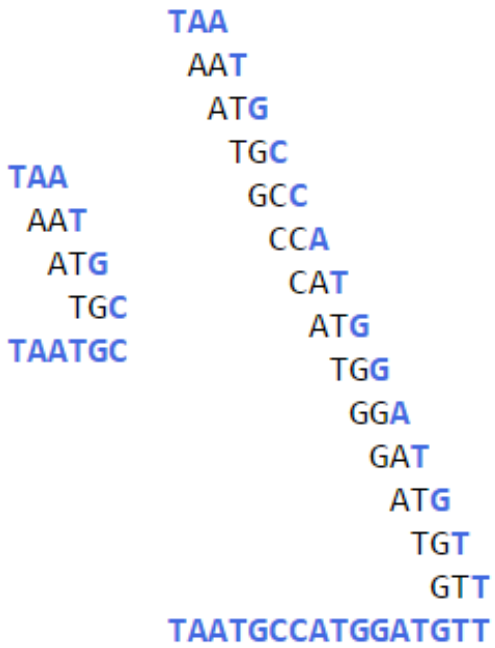
Reconstruct a string from its k-mer composition.

1. **Input:** An integer k and a collection *Patterns* of k-mers.
2. **Output:** A string *Text* with k-mer composition equal to *Patterns* (if such a string exists).

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

Difficulties

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



Difficulties: Repeats

Genome path

String Spelled by a Genome Path Problem

Reconstruct a string from its genome path.

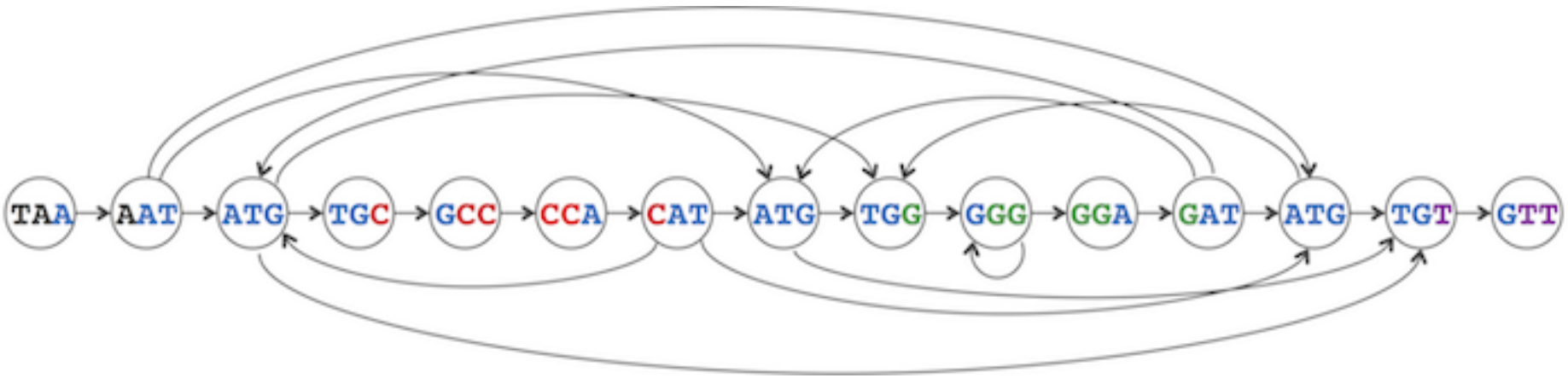
1. **Input:** A sequence of k-mers $Pattern_1, \cdots, Pattern_n$ such that the last $k-1$ symbols of $Pattern_i$ are equal to the first $k-1$ symbols of $Pattern_{i+1}$ for $1 \leq i \leq n-1$.
2. **Output:** A string $Text$ of length $k+n-1$ such that the i-th k-mer in $Text$ is equal to $Pattern_i$ (for $1 \leq i \leq n$).

Prefix: First $k-1$ nucleotides

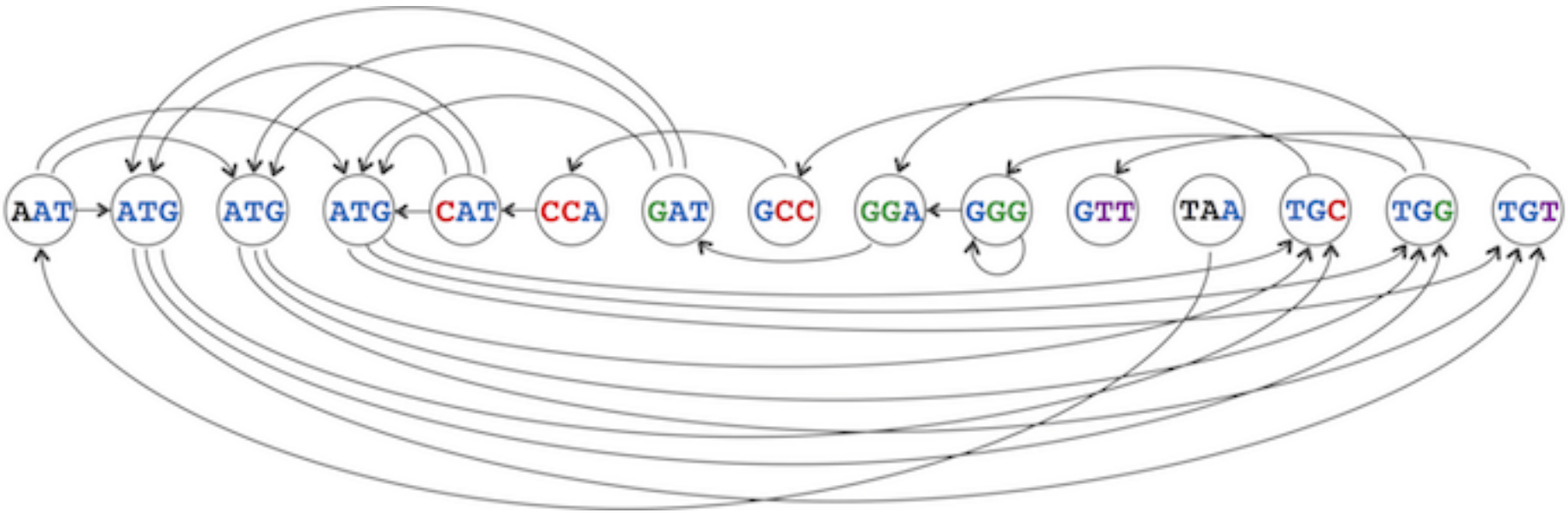
Suffix: Last $k-1$ nucleotides

we will use an arrow to connect any k-mer $Pattern1$ to a k-mer $Pattern2$ if the suffix of $Pattern1$ is equal to the prefix of $Pattern2$.

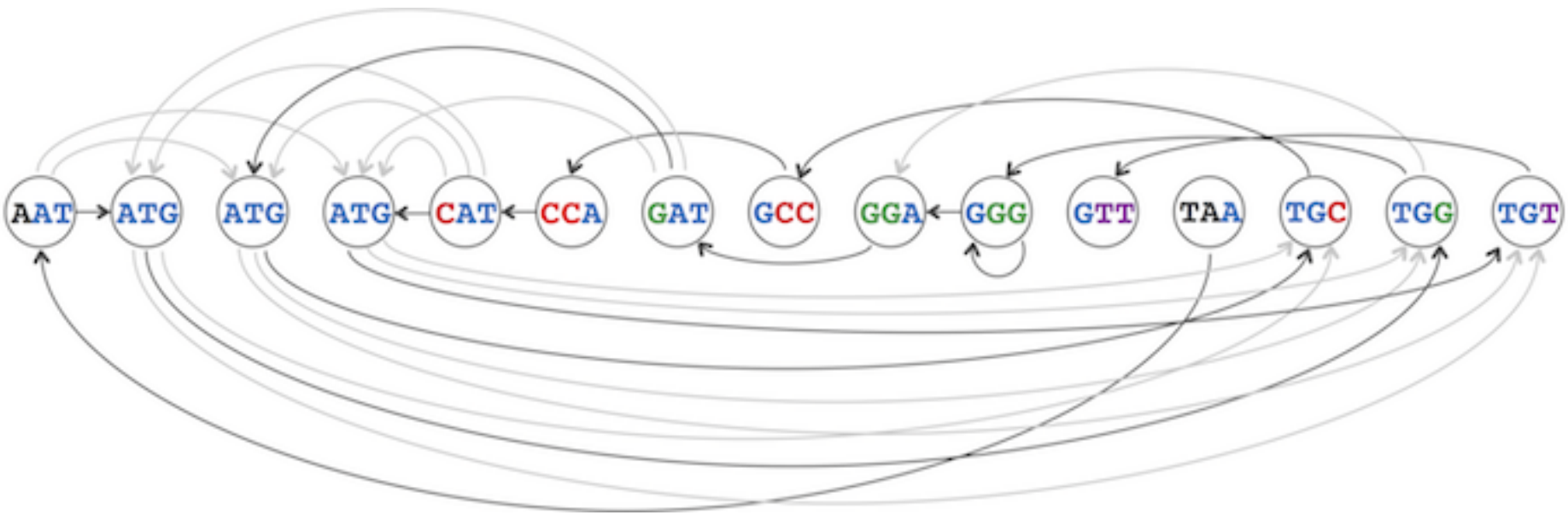
Overlap Graph



Overlap Graph



Overlap Graph



Overlap Graph Problem

The problem

Construct the overlap graph of a collection of k-mers.

1. **Input:** A collection *Patterns* of k-mers.
2. **Output:** The overlap graph *Overlap*(*Patterns*).

ROSALIND: https://rosalind.info/problems/ba3c/

Sample Input:

```
ATGCG
GCATG
CATGC
AGGCA
GGCAT
```

Sample Output:

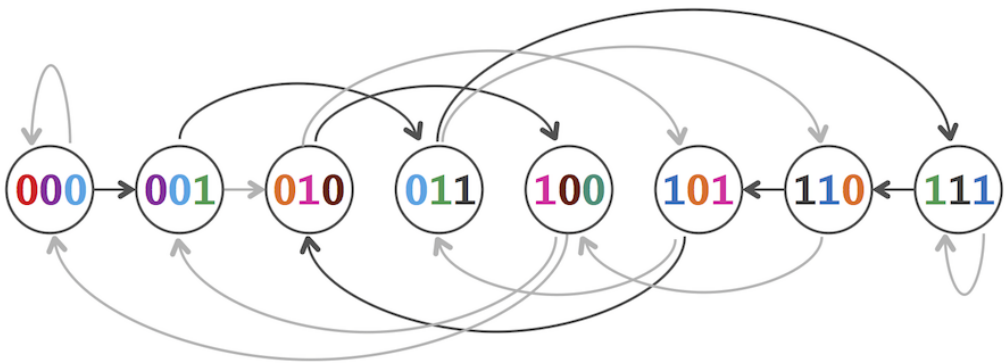
```
AGGCA -> GGCAT
CATGC -> ATGCG
GCATG -> CATGC
GGCAT -> GCATG
```

Hamiltonian Paths

Hamiltonian Path Problem

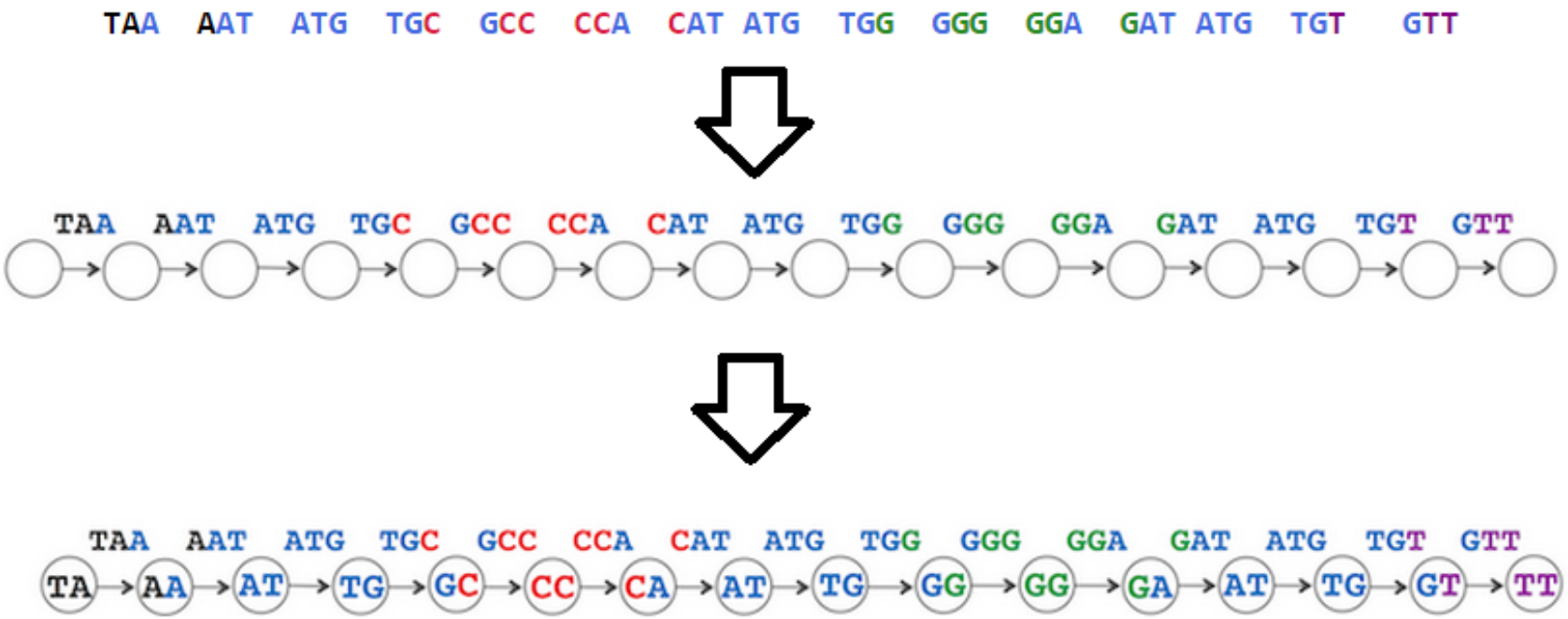
Construct a Hamiltonian path in a graph.

1. **Input:** A directed graph.
2. **Output:** A path visiting every node in the graph exactly once (if such a path exists).

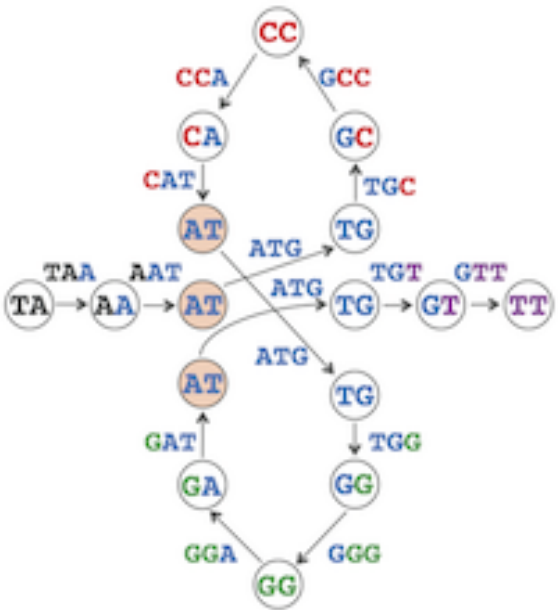


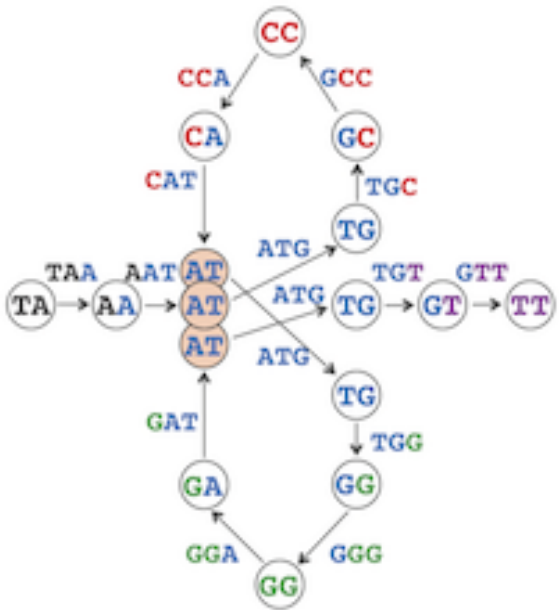
A binary string is a string composed only of 0's and 1's; a binary string is k-universal if it contains every binary k-mer exactly once. For example, 0001110100 is a 3-universal string, as it contains each of the eight binary 3-mers (000, 001, 011, 111, 110, 101, 010, and 100) exactly once.

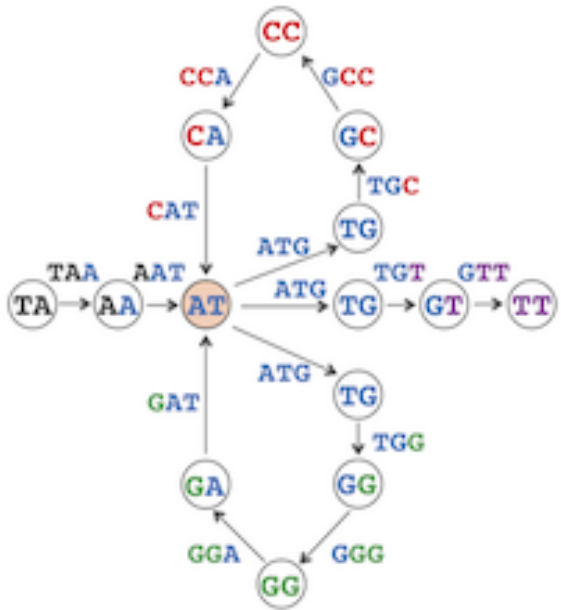
de Bruijn graphs



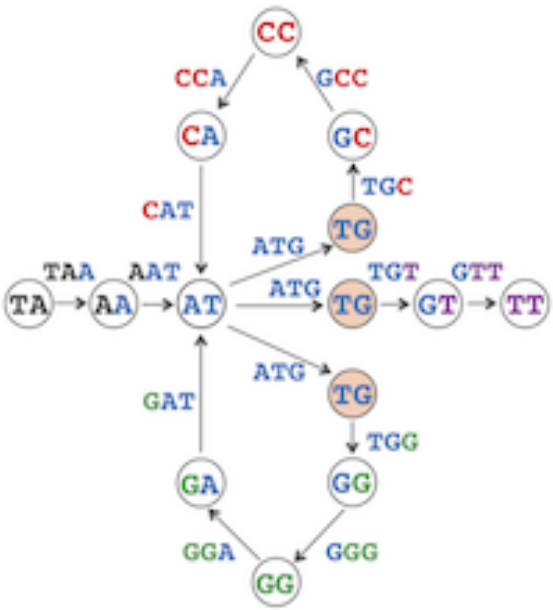
de Bruijn graphs

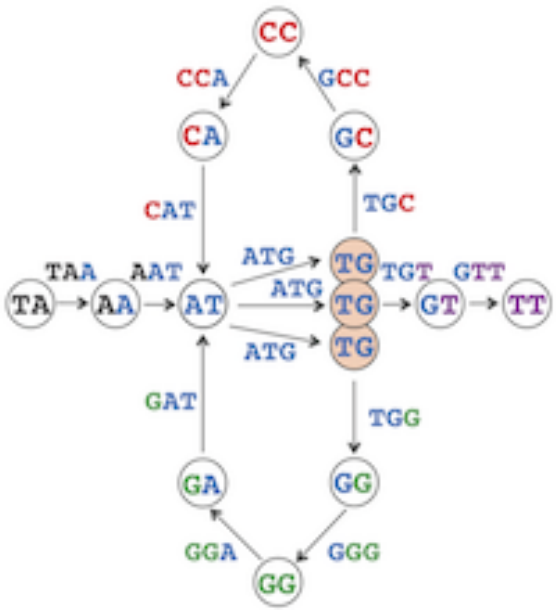


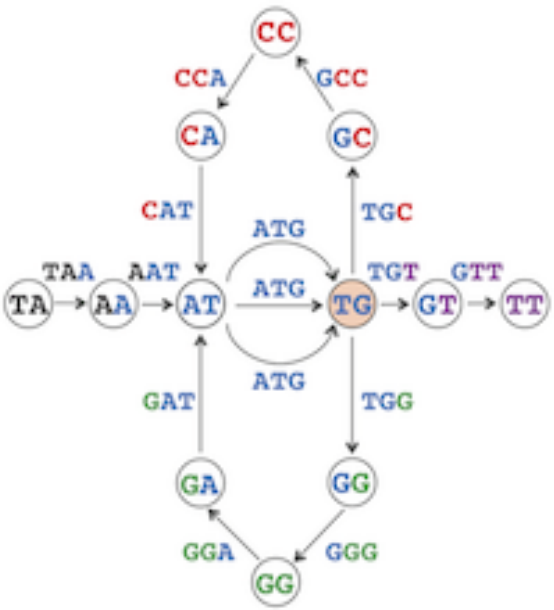




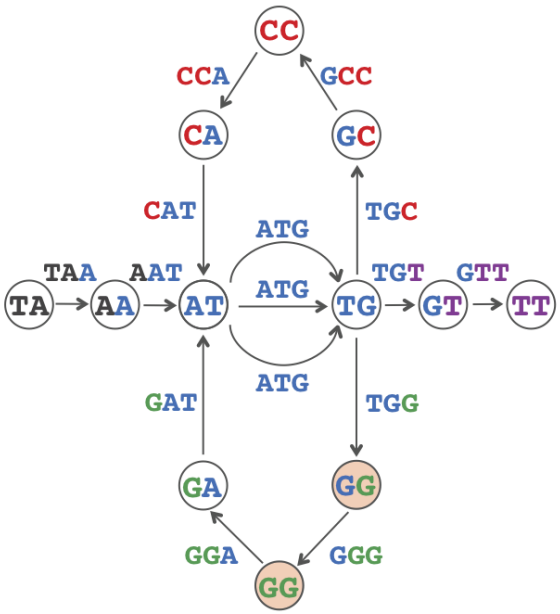
de Bruijn graphs

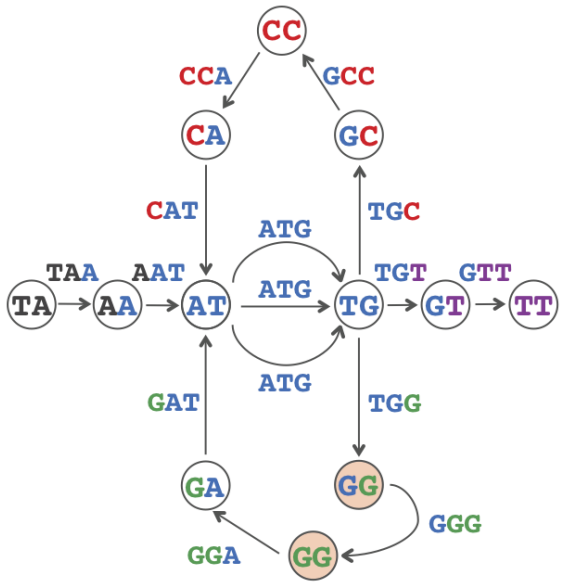


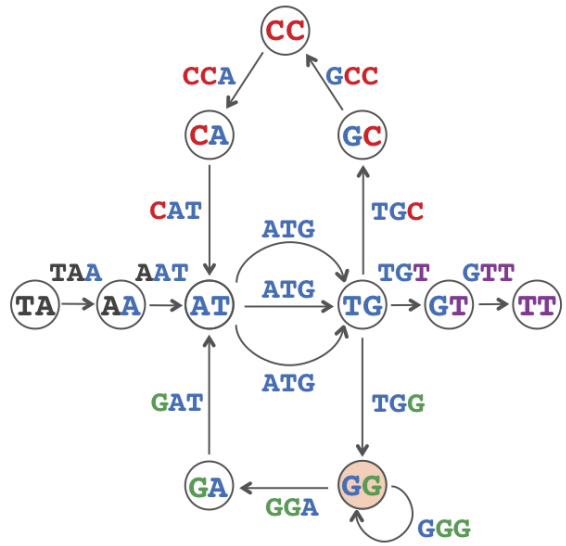




de Bruijn graphs







de Bruijn graphs

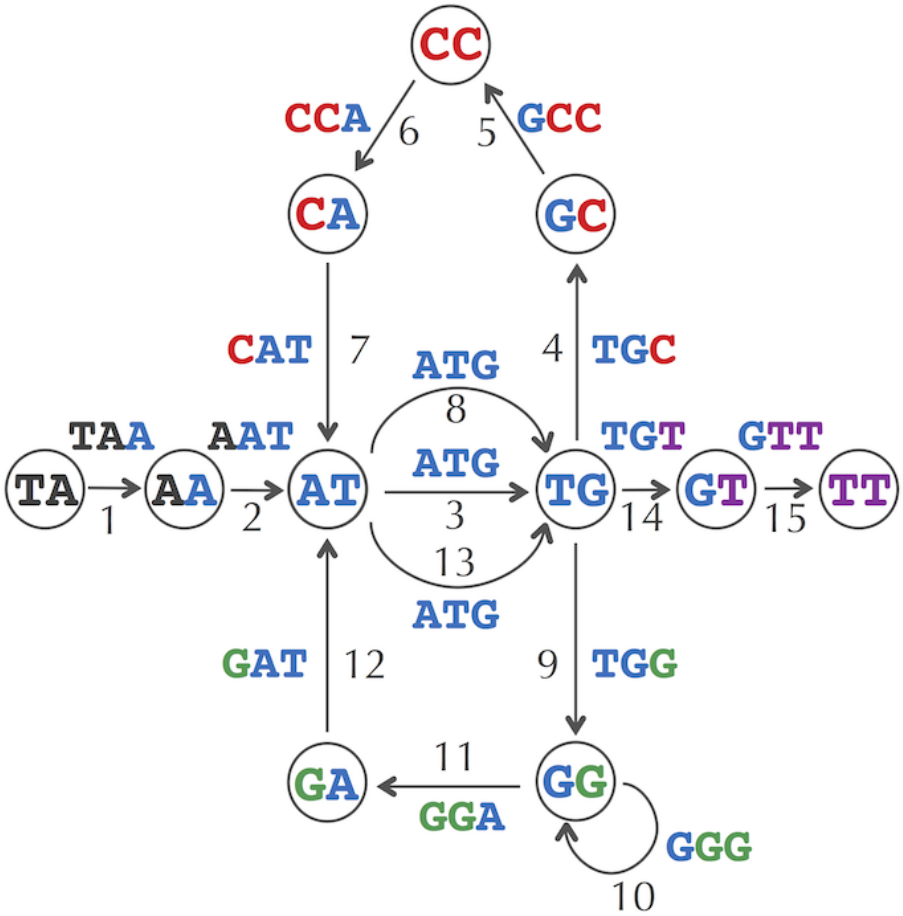
De Bruijn Graph from a String Problem

Construct the de Bruijn graph of a string.

1. **Input:** An integer k and a string Text.
2. **Output:** $DeBruijn_k(Text)$.

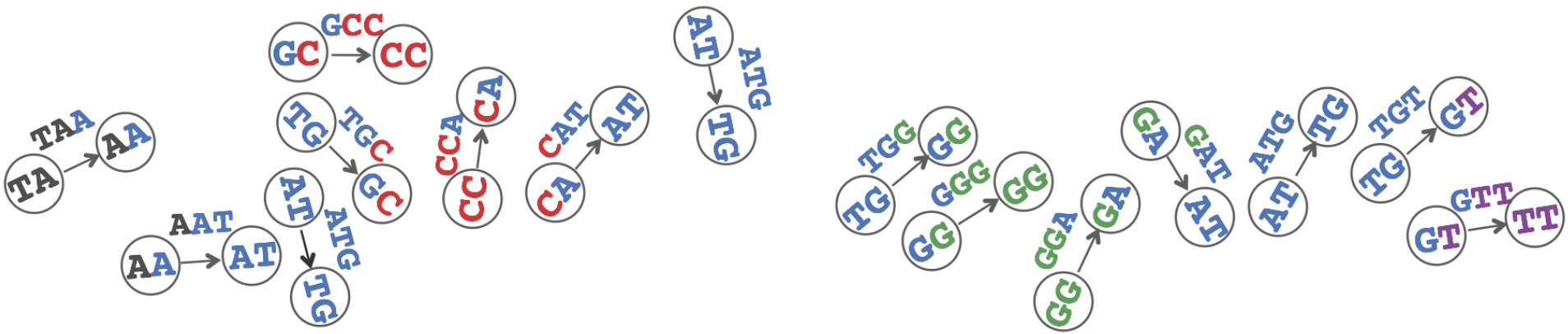
ROSALIND: https://rosalind.info/problems/ba3d/
Try: AAGATTCTCTAAGA for k=4

Euler Path

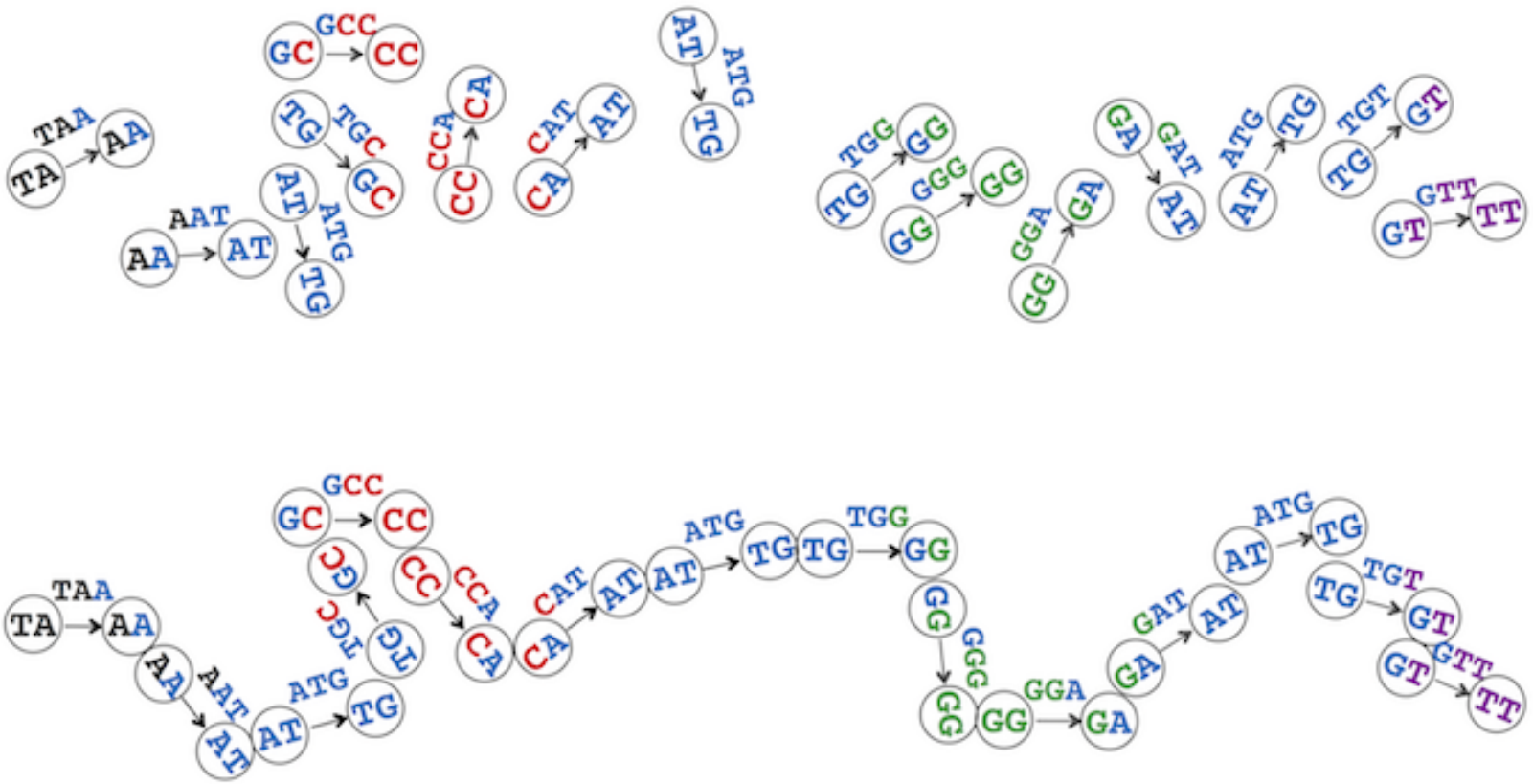


String Reconstruction == Euler Path Problem

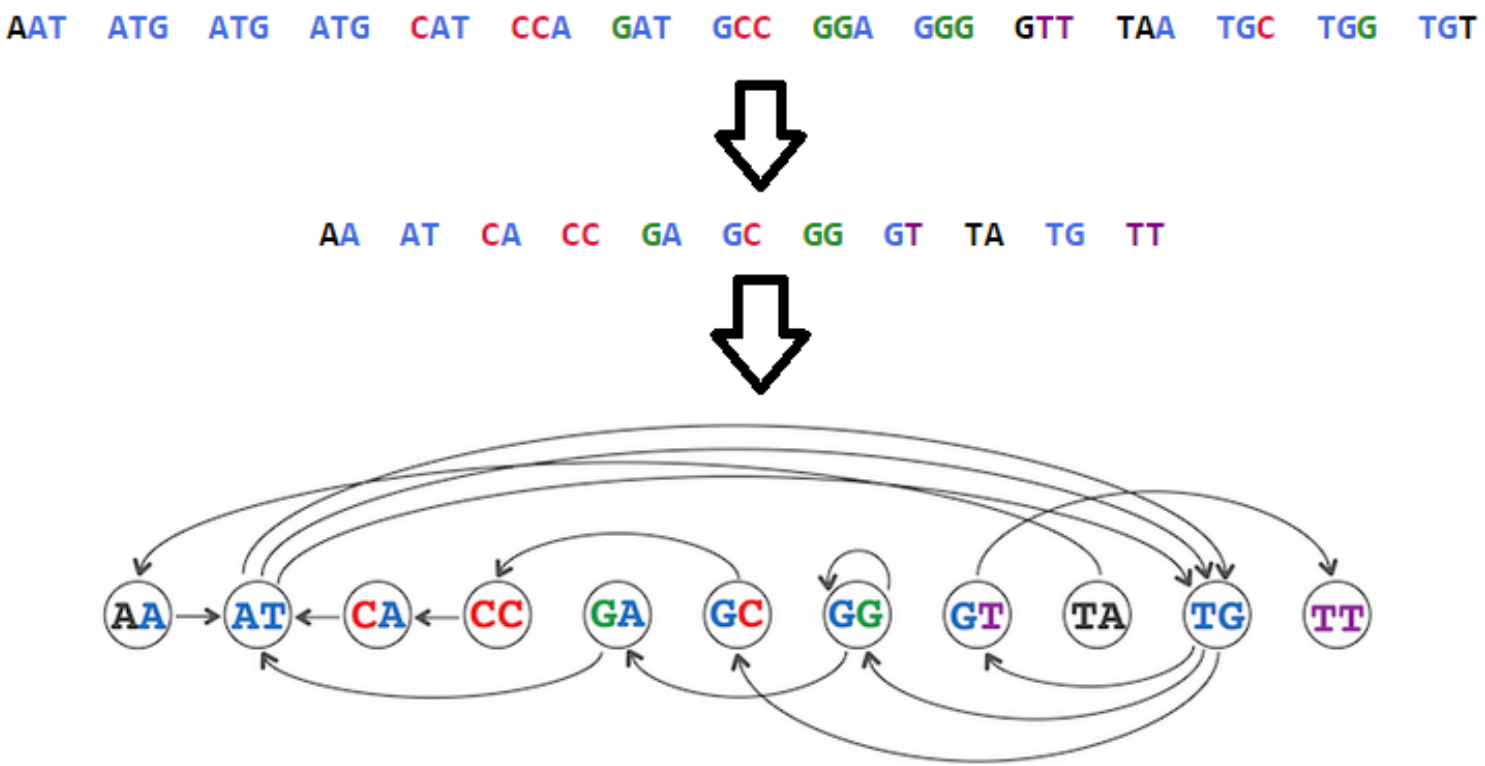
de Bruijn Graphs



Composition$_3$ Graph



Composition$_3$ Graph



de Bruijn Graph

DeBruijn Graph from k-mers Problem

Construct the de Bruijn graph from a set of k-mers.

1. **Input:** A collection of k-mers Patterns.
2. **Output:** The adjacency list of the de Bruijn graph DeBruijn(Patterns).

ROSALIND: https://rosalind.info/problems/ba3e/

Sample Input:

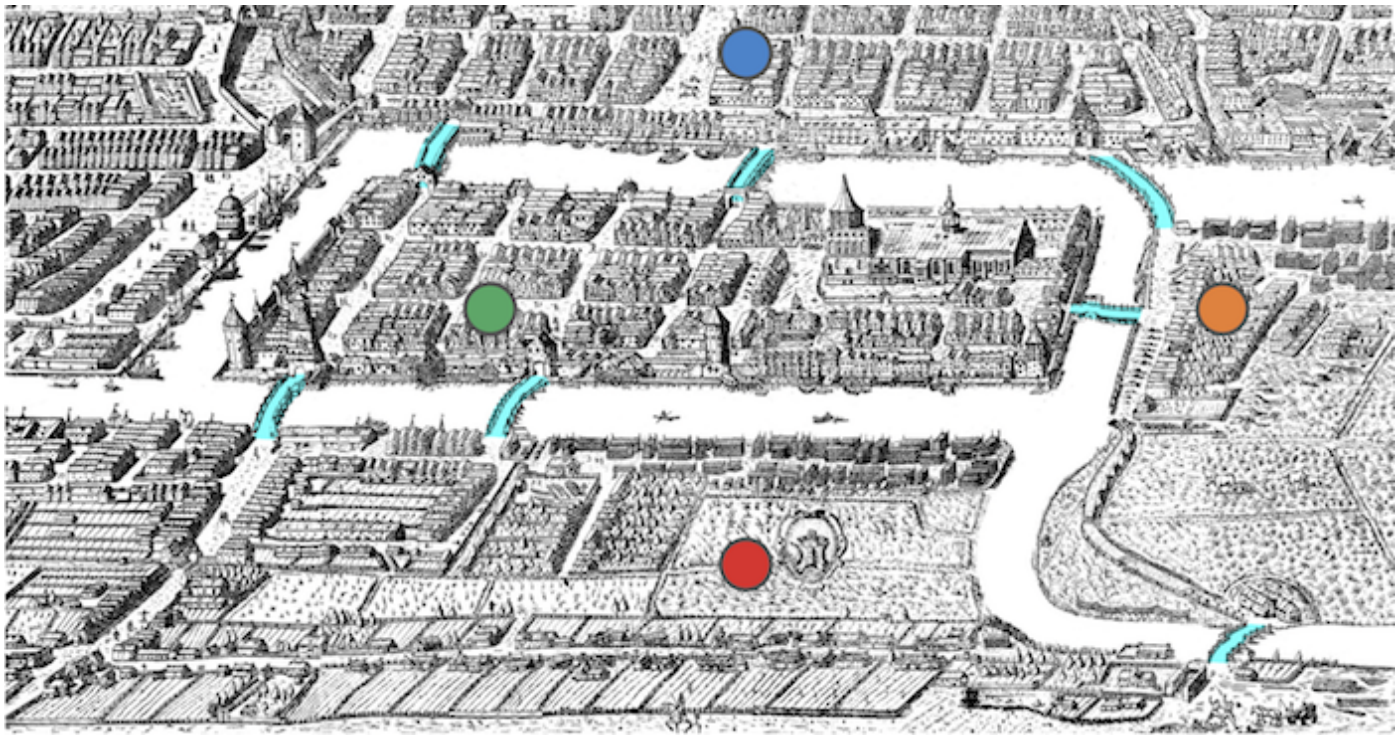
```
GAGG
CAGG
GGGG
GGGA
CAGG
AGGG
GGAG
```

Sample Output:

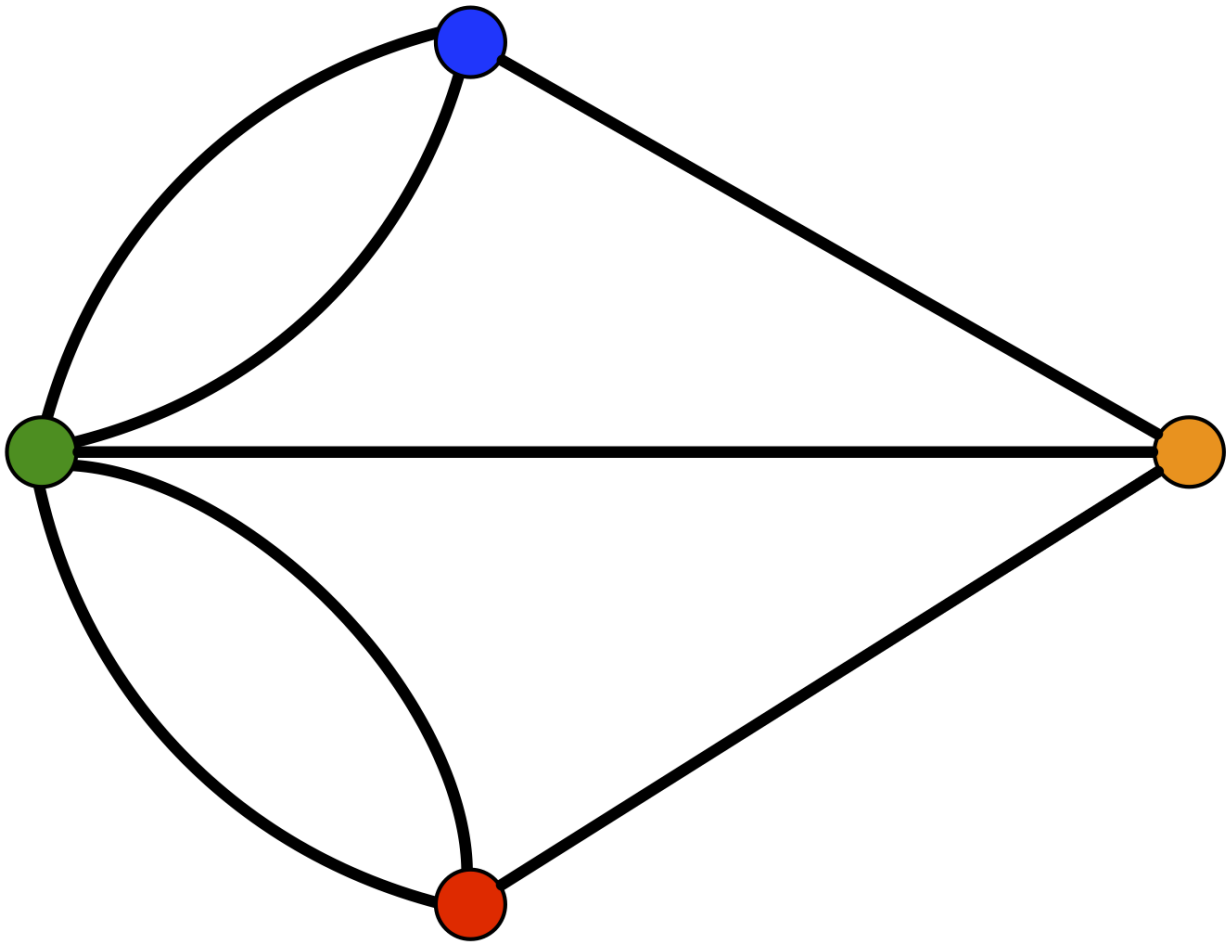
```
AGG -> GGG
CAG -> AGG,AGG
GAG -> AGG
GGA -> GAG
GGG -> GGA,GGG
```

Euler Path vs Hamiltonian Path

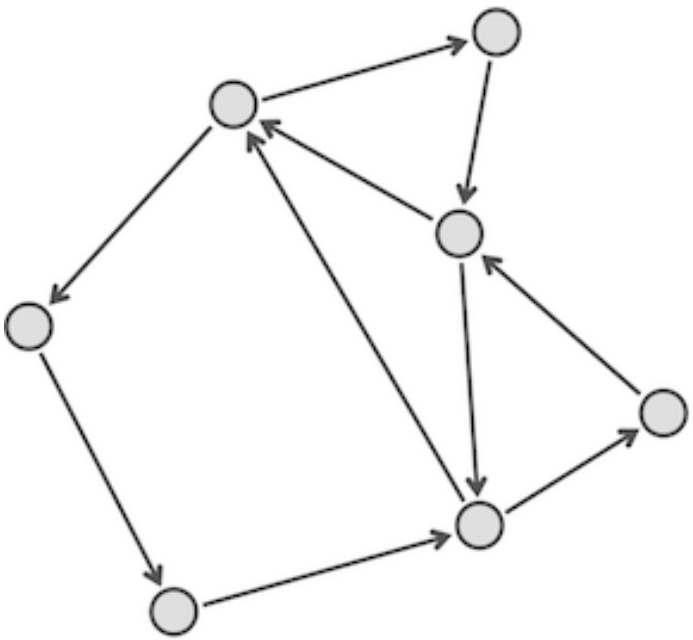
Bridges of Königsberg Problem.

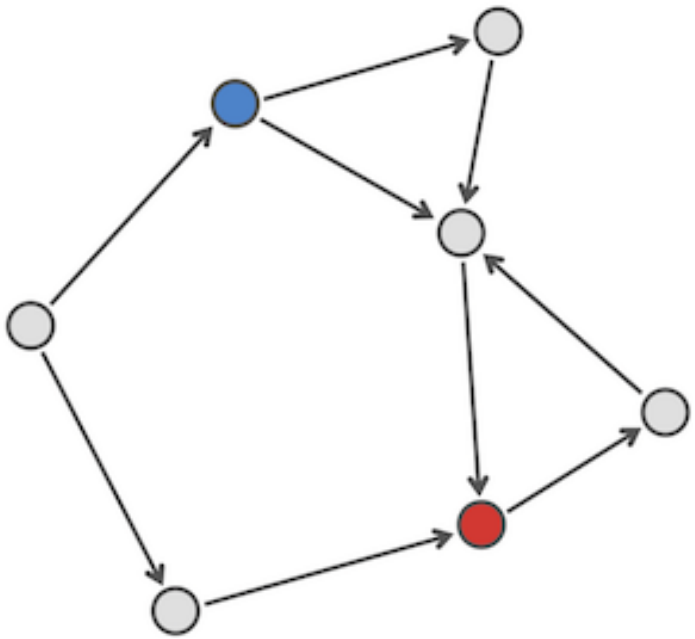


The graph Königsberg.



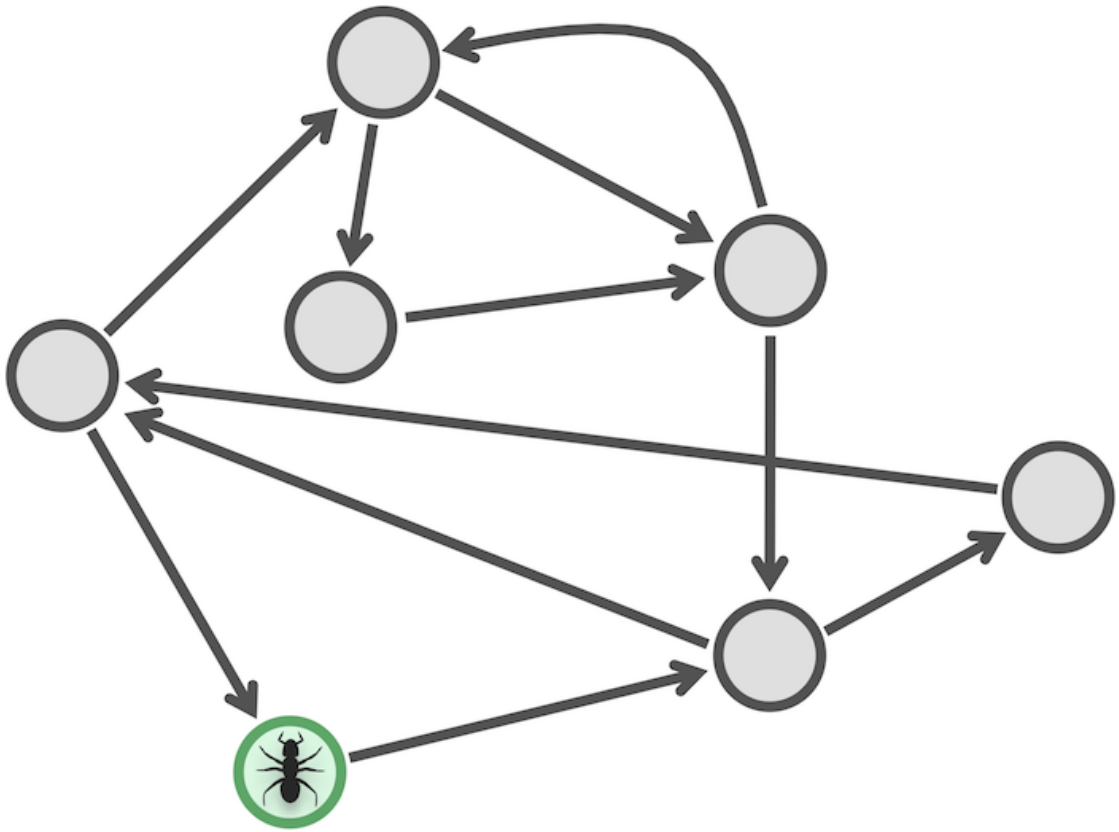
The Euler's Theorem





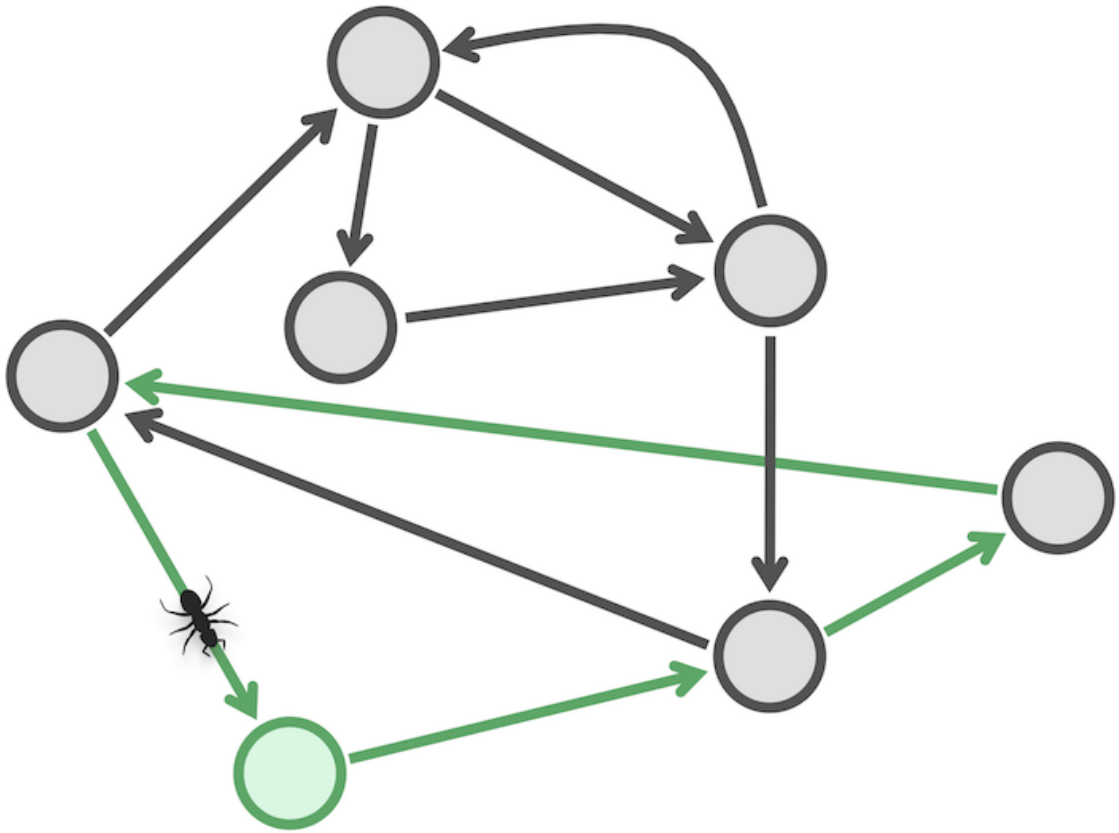
Euler's Theorem: Every balanced, strongly connected directed graph is Eulerian.

The Euler's Theorem



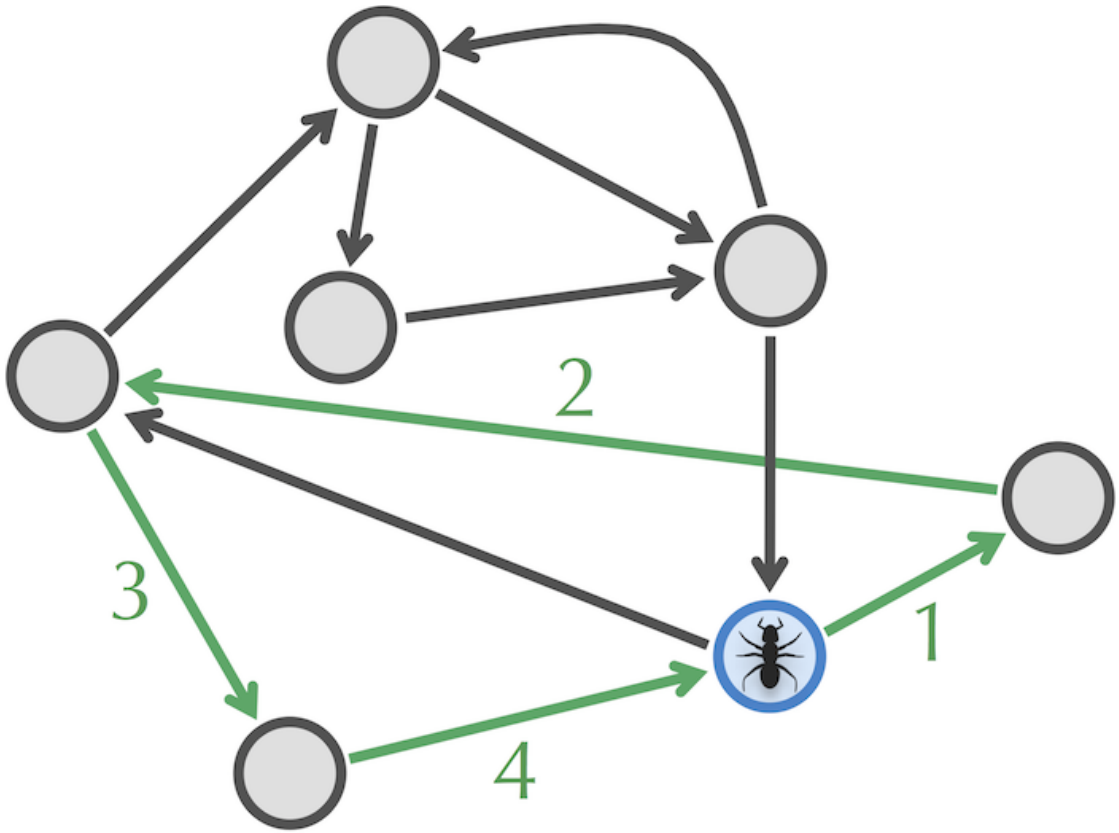
Euler's Theorem: Every balanced, strongly connected directed graph is Eulerian.

The Euler's Theorem



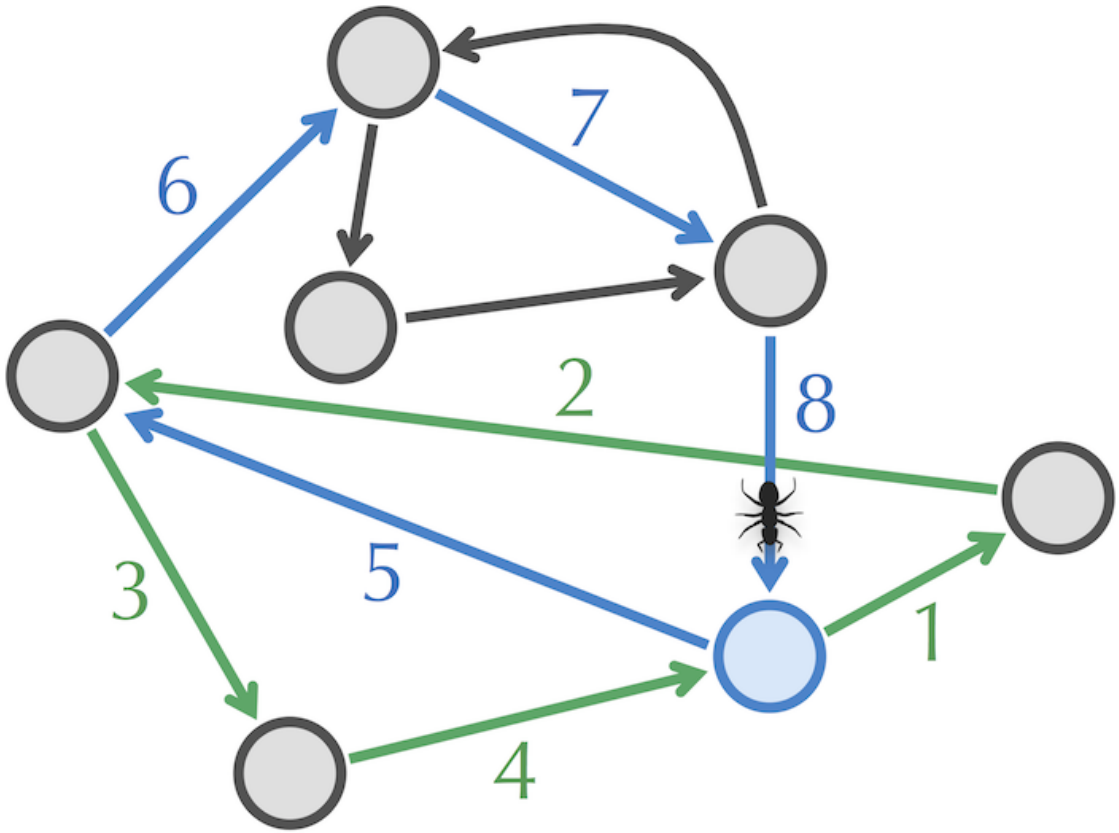
Euler's Theorem: Every balanced, strongly connected directed graph is Eulerian.

The Euler's Theorem



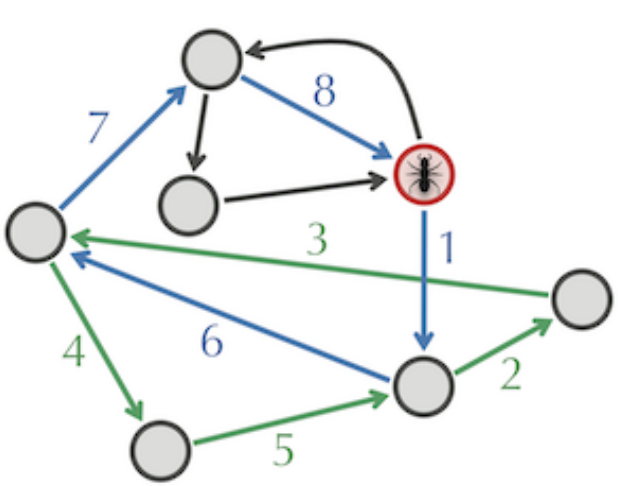
Euler's Theorem: Every balanced, strongly connected directed graph is Eulerian.

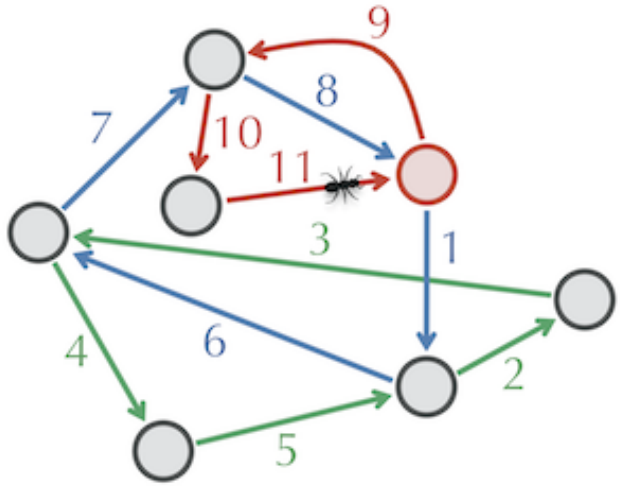
The Euler's Theorem



Euler's Theorem: Every balanced, strongly connected directed graph is Eulerian.

The Euler's Theorem





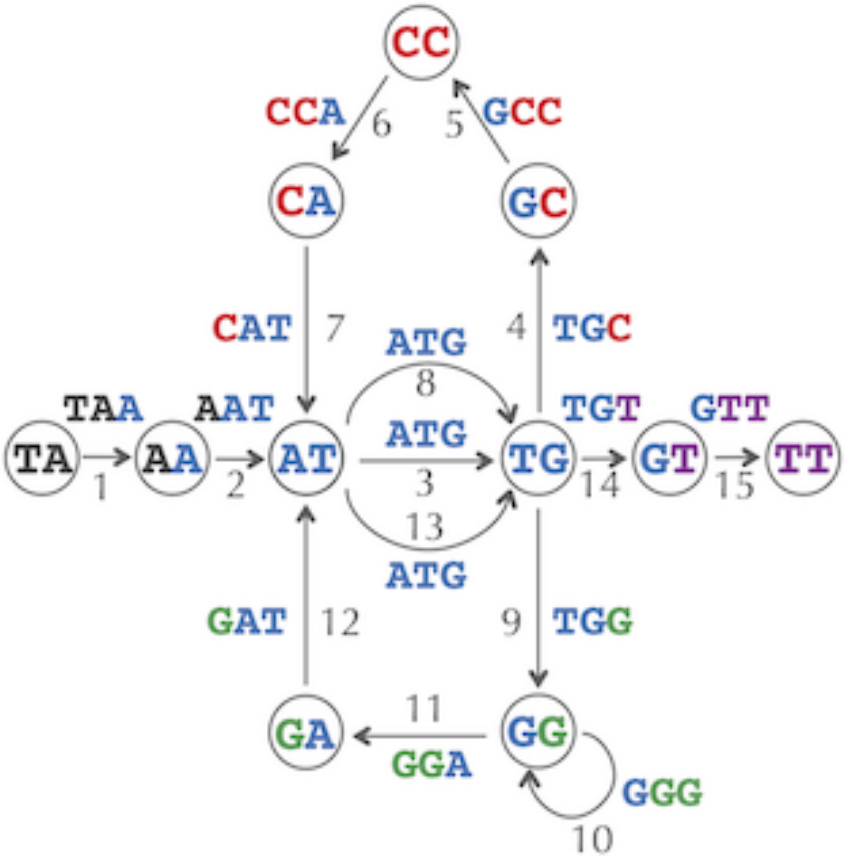
Euler's Theorem: Every balanced, strongly connected directed graph is Eulerian.

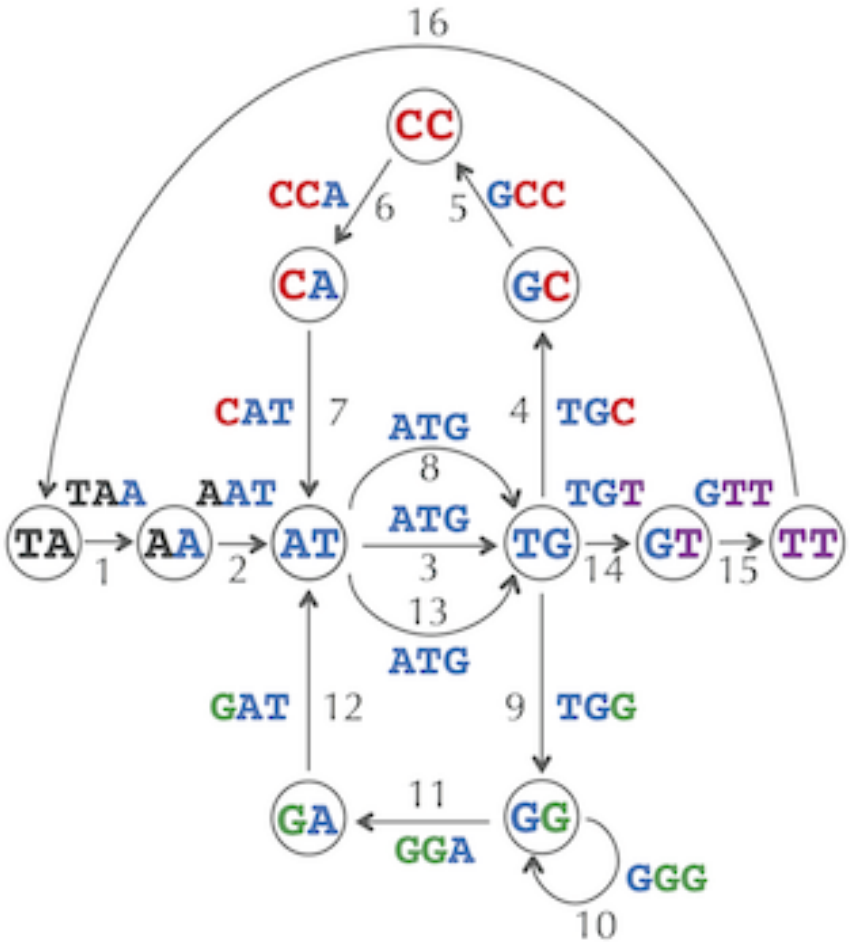
The Euler's Theorem

```
EULERIANCYCLE(Graph)
    form a cycle Cycle by randomly walking in Graph (don't visit the same edge twice!)
    while there are unexplored edges in Graph
        select a node newStart in Cycle with still unexplored edges
        form Cycle' by traversing Cycle (starting at newStart) and then randomly walking
        Cycle ← Cycle'
    return Cycle
```

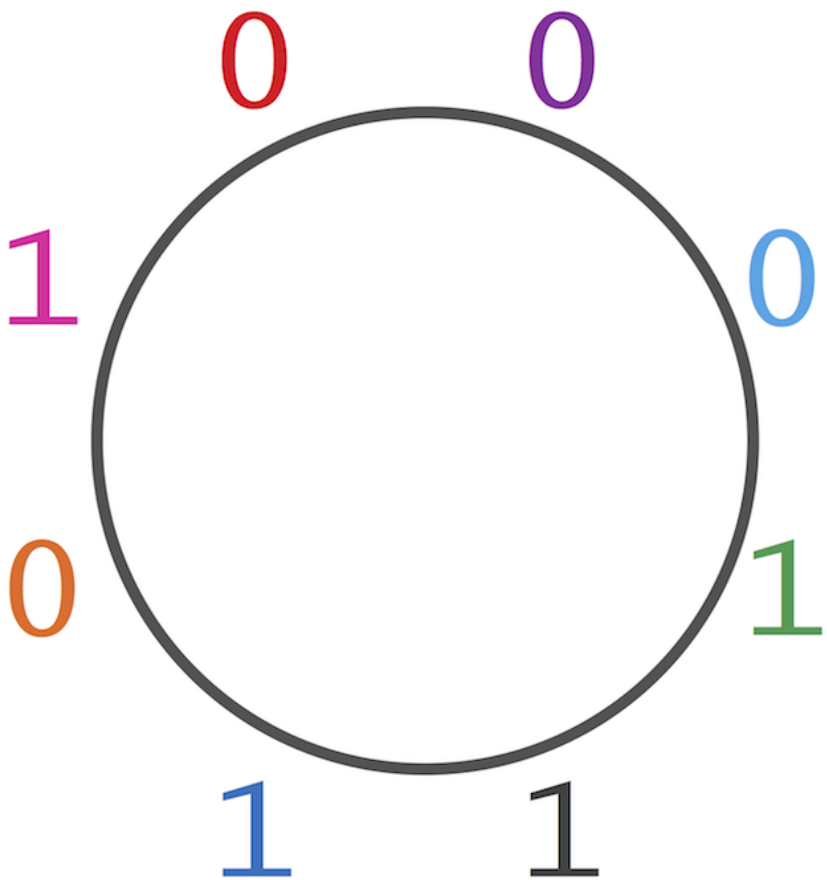
Euler's Theorem: Every balanced, strongly connected directed graph is Eulerian.

The Euler's Path to Cycle

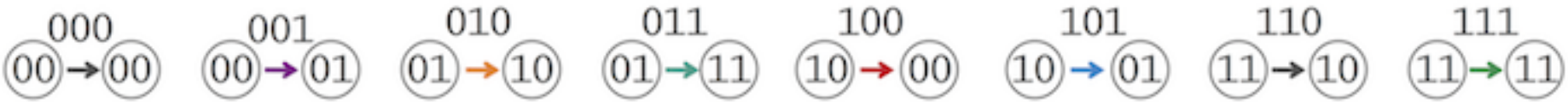


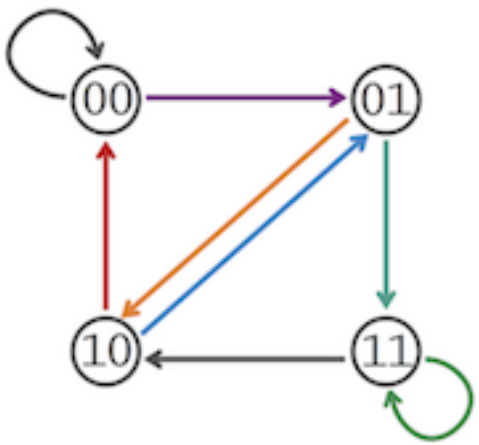


k-universal circular string

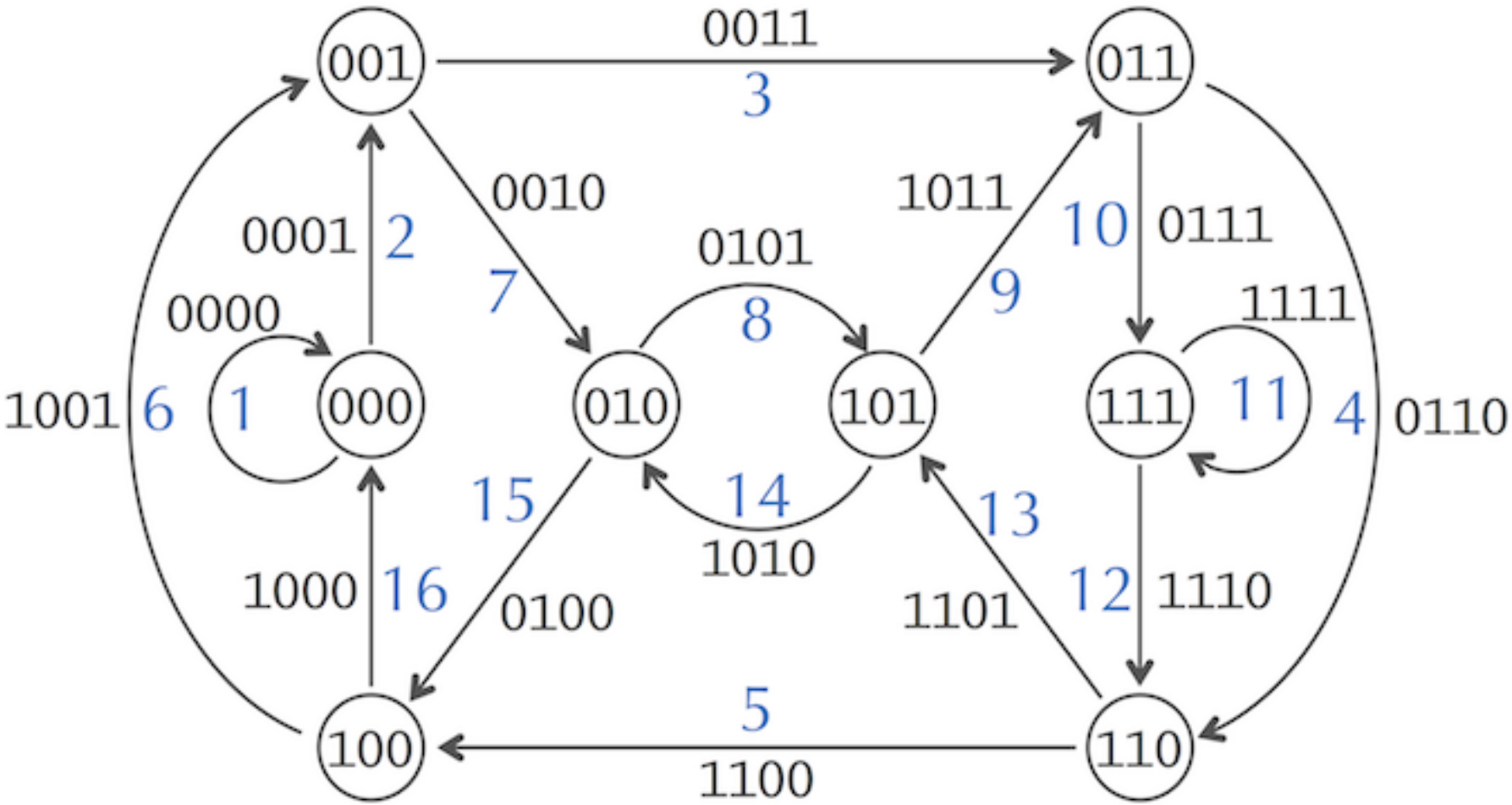


k-universal circular string





k-universal circular string

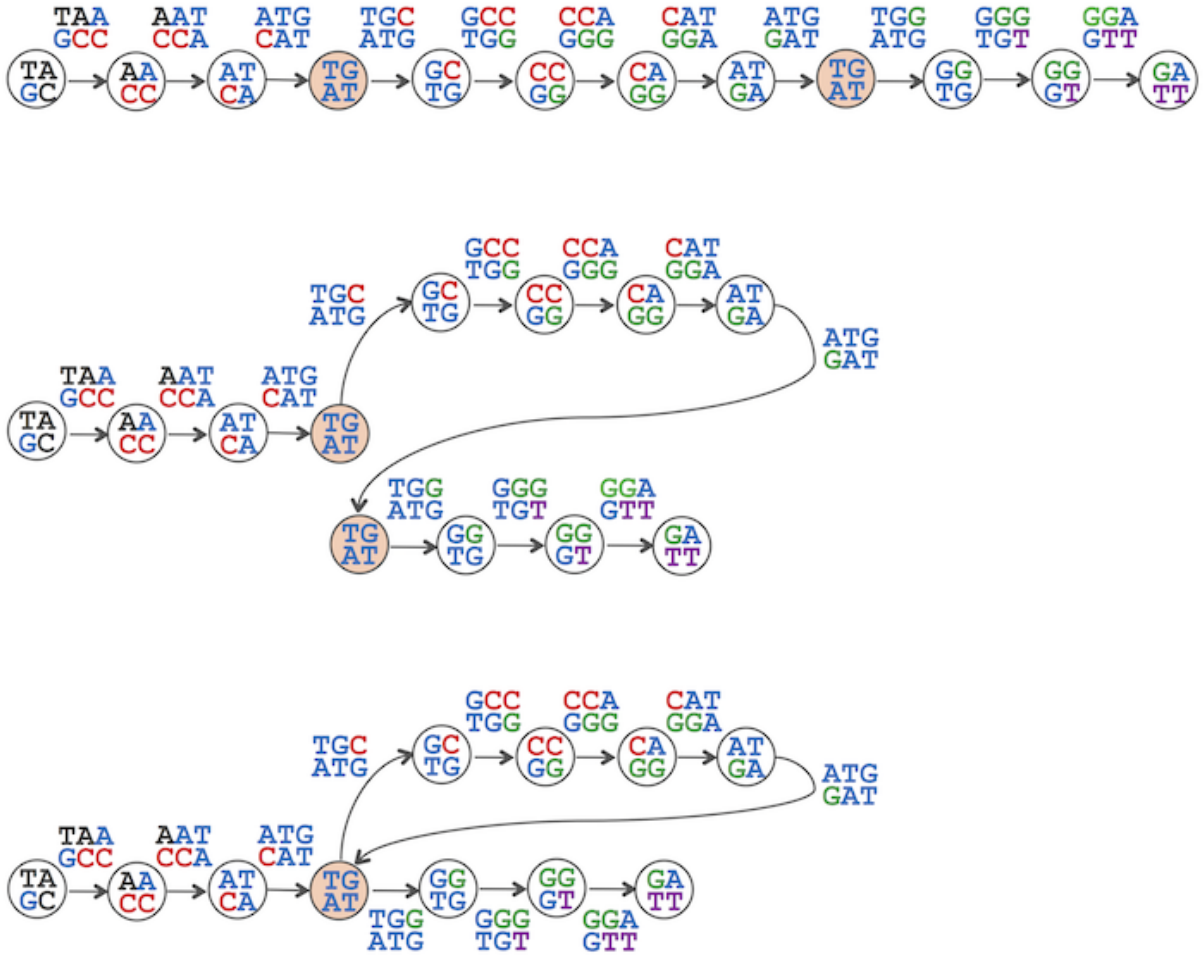


Read Pairs

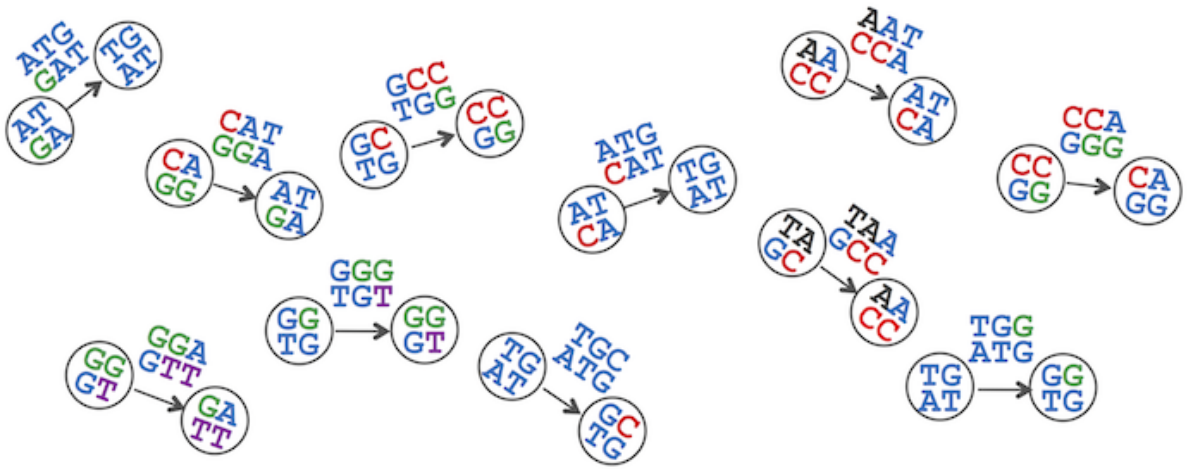
long "gapped" read of length k + d + k whose first and last k-mers are known but whose middle segment of length d is unknown.
(3,2) mers of TAATGCCATGGGATGTT

Paired de Bruijn Graphs

Paired de Bruijn Graphs

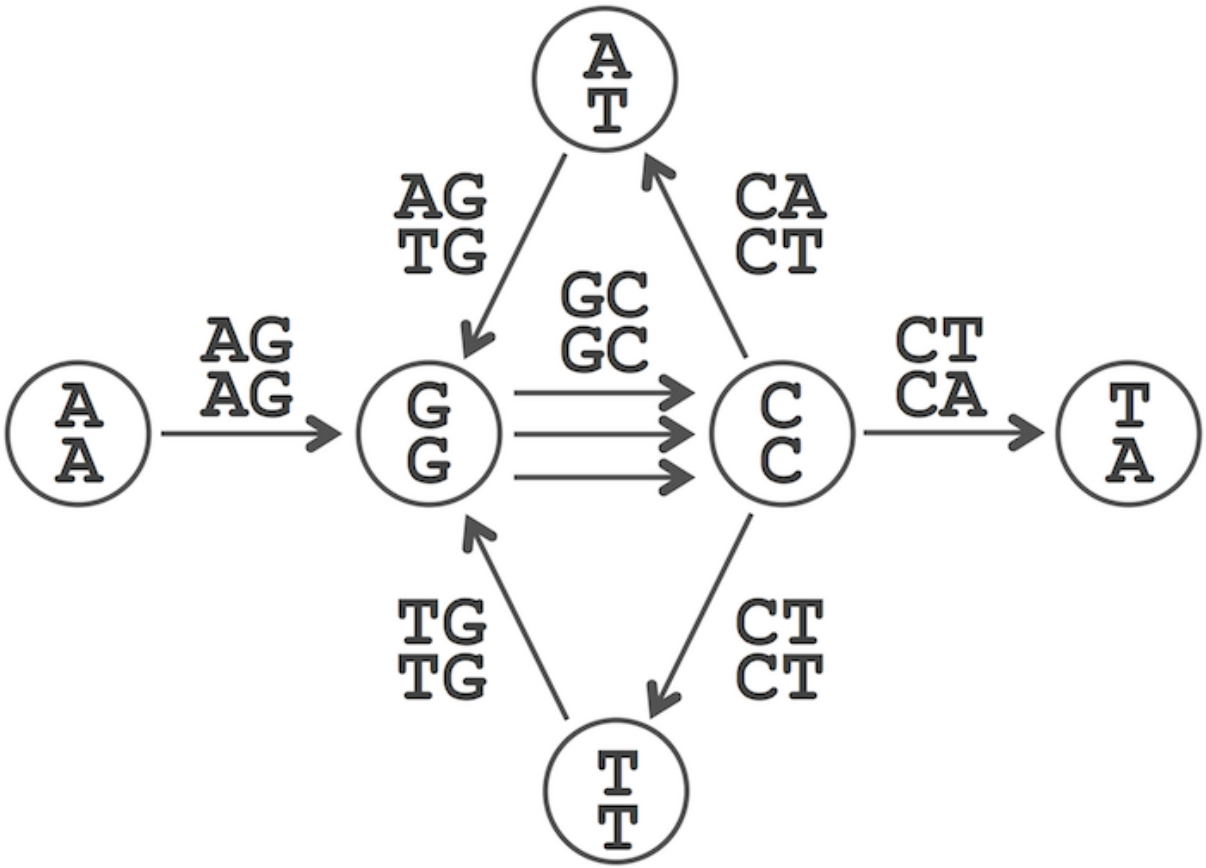


Paired de Bruijn Graphs



Paired de Bruijn Graphs

(AG—AG) →(GC—GC) → (CA—CT) → (AG—TG) → (GC—GC) →
(CT—CT) → (TG—TG) → (GC—GC) → (CT—CA)



Coverage

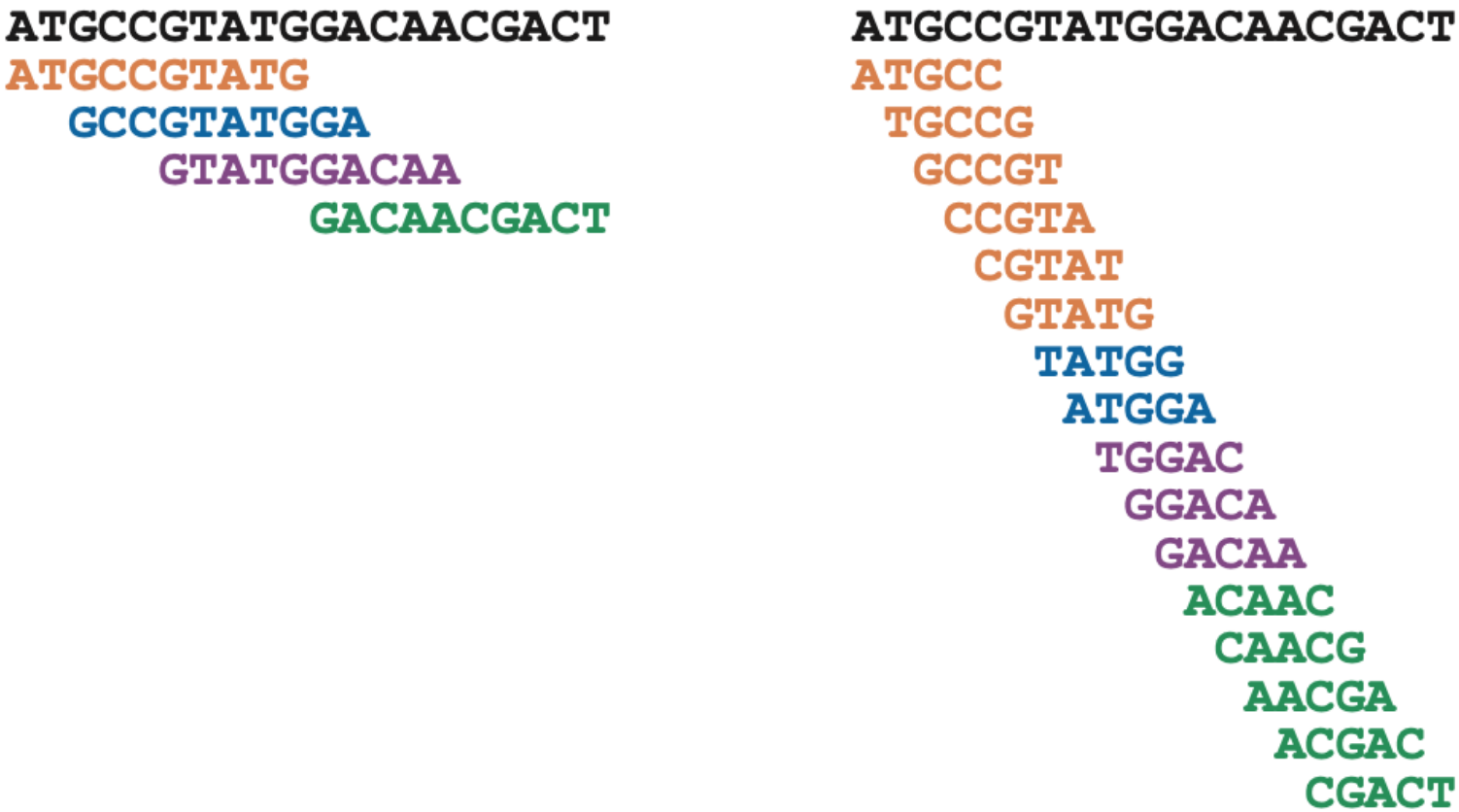
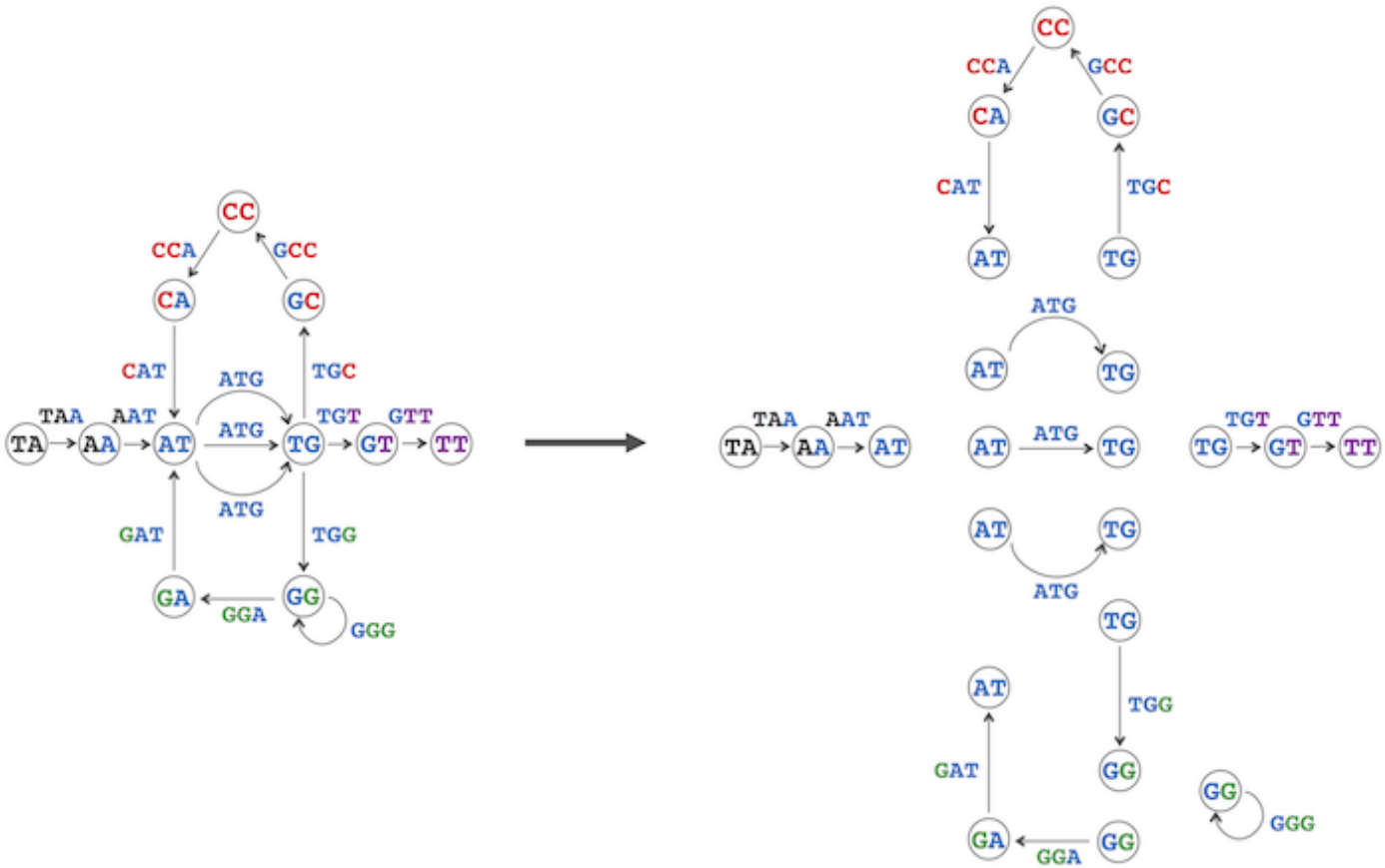


FIGURE 3.37 Breaking 10-mer reads (left) into 5-mers results in perfect coverage of a genome by 5-mers (right).

Contigs


CC
CCA
GCC
CA
GC
CAT
ATG
TGC
TAA
AAT
TGT
GTT
TA
AA
AT
ATG
TG
GT
TT
ATG
GAT
TGG
GA
GGA
GG
GGG
CC
CCA
GCC
CA
GC
CAT
TGC
AT
TG
ATG
AT
TG
TAA
AAT
TA
AA
AT
AT
ATG
TG
TGT
GTT
TG
GT
TT
AT
TG
ATG
TG
AT
TGG
GAT
GG
GA
GG
GGA
GG
GGG