Bioinformatics: Genome Assembly

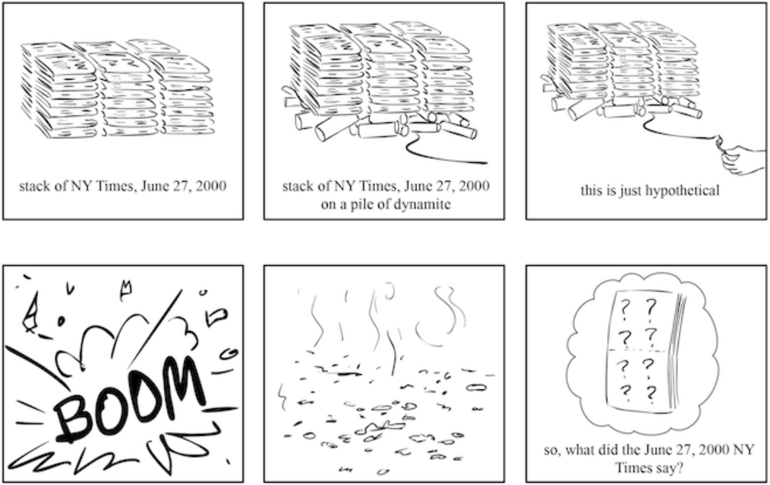
Swakkhar Shatabda

Department of Computer Science and Engineering
BRAC University

Book Reference

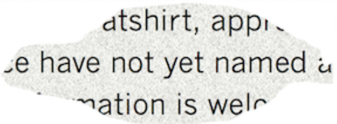
Chapter 3, Bioinformatics Algorithms: An Active Learning Approach - I

Newspaper Explodes

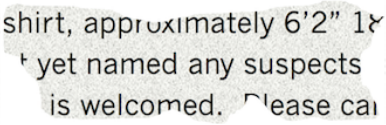

stack of NY Times, June 27, 2000
stack of NY Times, June 27, 2000 on a pile of dynamite
this is just hypothetical
BOOM
? ? ? ? ? ? ? ?
so, what did the June 27, 2000 NY Times say?

Newspaper Explodes

1. Overlapping Information
2. Lost information



atshirt, appr
e have not yet named a
mation is welc

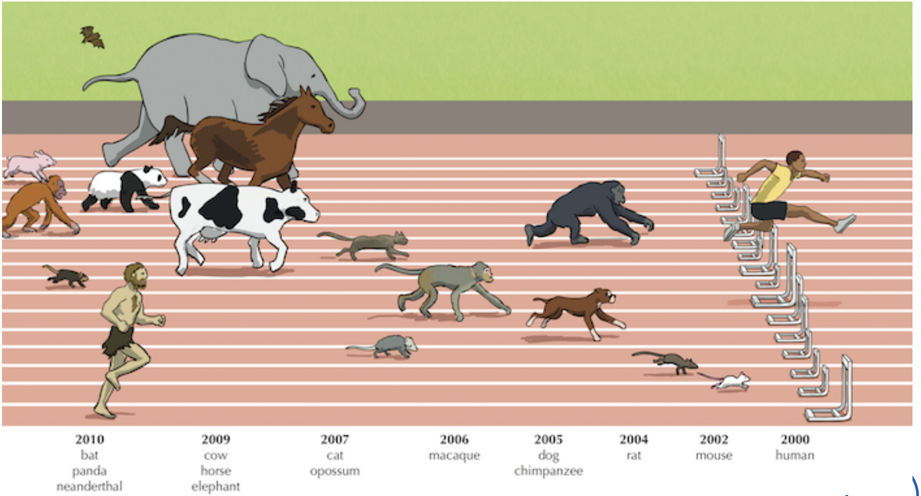


shirt, approximately 6’2” 1
yet named any suspects
is welcomed. lease ca

but what do exploding newspapers have to do with biology?

- Determining the order of nucleotides in a genome, or genome sequencing, presents a fundamental task in bioinformatics.
  - Human Genome - 3 billion bp
  - Amoeba dubia - 200 times longer!
- The first sequenced genome, belonging to a fX174 bacterial phage (i.e., a virus that preys on bacteria), had only 5,386 nucleotides and was completed in 1977 by Frederick Sanger.

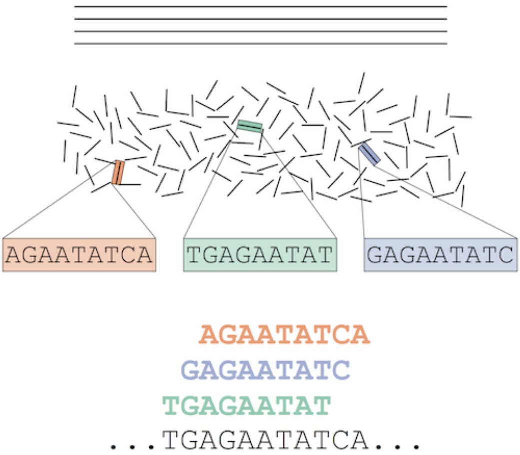
Genome Sequencing



Genome Sequencing

- Can not read the nucleotides of a genome from beginning to end in the same way that you would read a book.
- Technology can read sequences in much shorter DNA fragments called **reads**.
- Traditional Approach
  - Researchers take a small tissue or blood sample containing millions of cells with identical DNA
  - Use biochemical methods to break the DNA into fragments
  - sequence these fragments to produce reads.
  - The difficulty is that researchers do not know where in the genome these reads came from, and so they must use overlapping reads to reconstruct the genome.
  - Similar to the newspaper problem.

Genome Assembly



Multiple identical copies of a genome

Shatter the genome into reads

Sequence the reads

AGAATATCA TGAGAATAT GAGAATATC

Assemble the genome using overlapping reads

```
      AGAATATCA
     GAGAATATC
    TGAGAATAT
 ...TGAGAATATCA...
```

Difficulties in Genome Assembly

- The barrier to sequence such a genome is not always experimental; biologists can easily generate enough reads to analyze a large genome, but assembling these reads still presents a major computational challenge.
- 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

Initial 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).
$Composition_3(TATGGGTGC) = \{ATG, GGG, GGG, GGT, GTG, TAT, TGC, TGG\}$

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:3A https://rosalind.info/problems/ba3a/

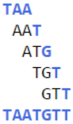
String Reconstruction Problem

The problem

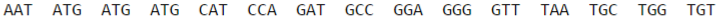
Reconstruct a string from its *k*-mer composition.

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 GTT TAA TGT

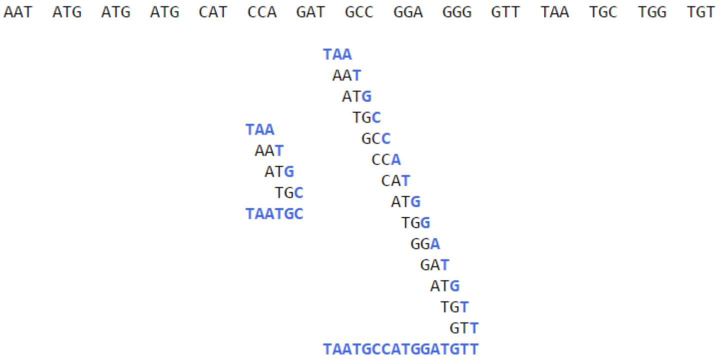


```
TAA
 AAT
  ATG
   TGT
    GTT
TAATGTT
```



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

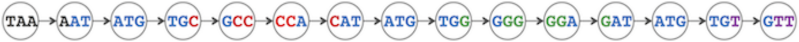
TAA
AAT
ATG
TGC
TAATGC

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

Difficulties: Repeats

- Approximately 50% of the human genome is made up of repeats, e.g., the approximately 300 nucleotide-long Alu sequence is repeated over a million times, with only a few nucleotides inserted/deleted/substituted each time

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

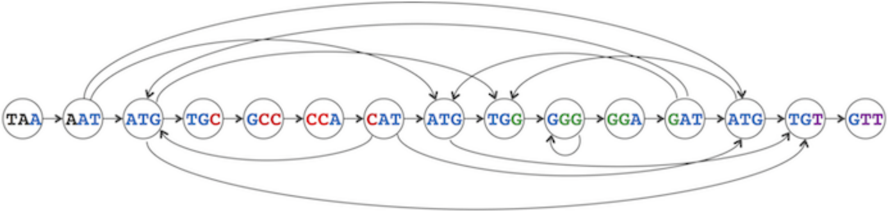
Suffix: Last $k-1$ nucleotides

we will use an arrow to connect any k-mer *Pattern*1 to a k-mer *Pattern*2 if the suffix of *Pattern*1 is equal to the prefix of *Pattern*2.

Overlap Graph

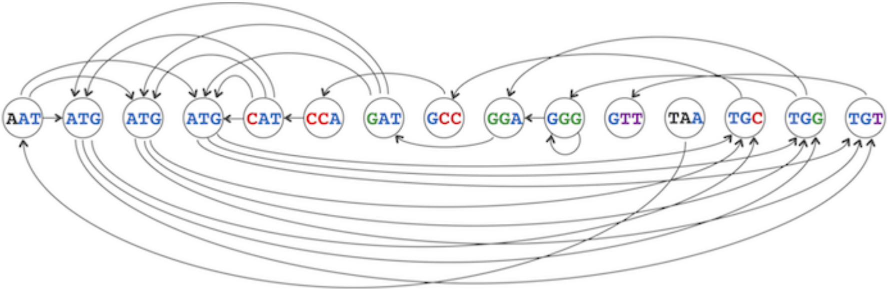
Overlap Graph

From an arbitrary collection of k-mers Patterns, we form a node for each *k*-mer in Patterns and connect *k*-mers Pattern and Pattern' by a directed edge if SUFFIX(Pattern) = PREFIX(Pattern'). The resulting graph is called the overlap graph on these *k*-mers.

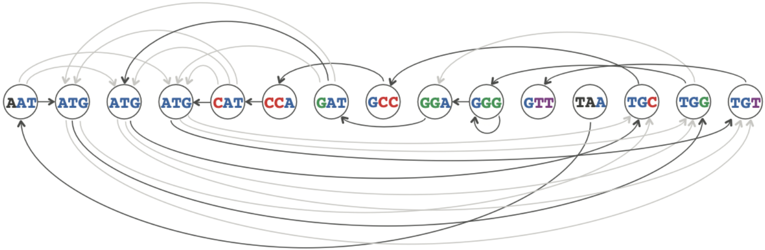


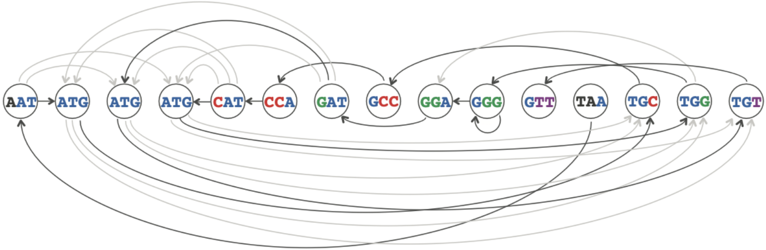
Overlap Graph

In genome sequencing, we do not know in advance how to correctly order reads.



Overlap Graph





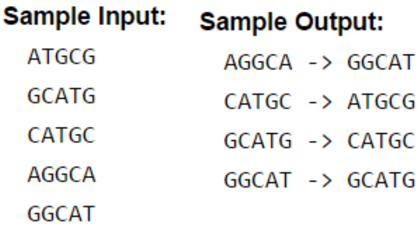
Overlap Graph Problem

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:3C https://rosalind.info/problems/ba3c/



Sample Input:

```
ATGCG
GCATG
CATGC
AGGCA
GGCAT
```

Sample Output:

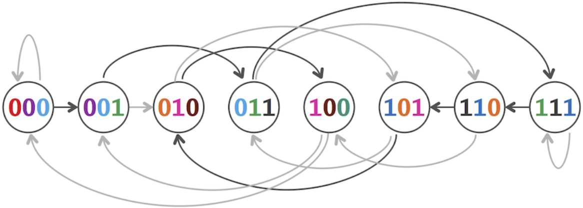
```
AGGCA -> GGCAT
CATGC -> ATGCG
GCATG -> CATGC
GGCAT -> GCATG
```

Hamiltonian Paths

Hamiltonian Path Problem

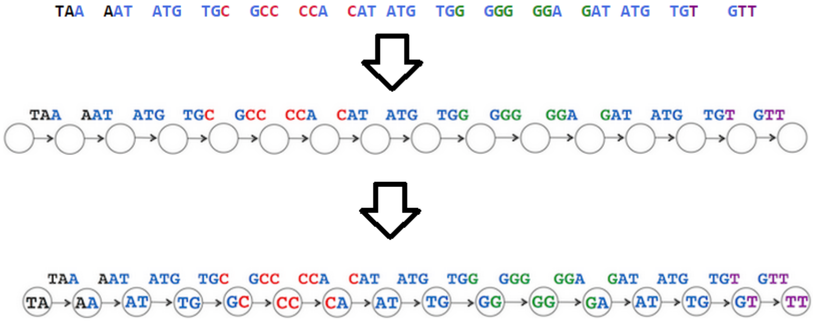
Construct a Hamiltonian path in a graph.

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

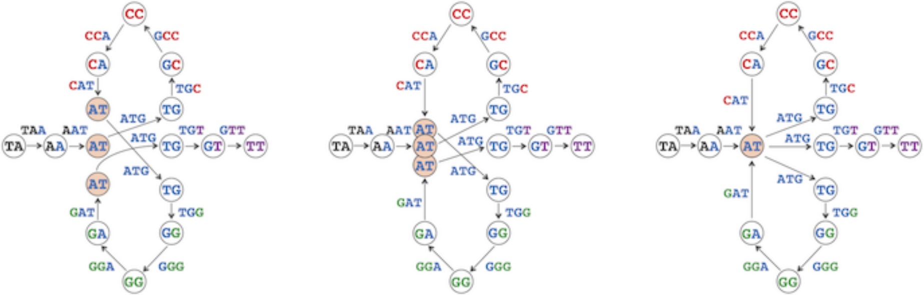


A binary string is a string composed only of 0's and 1's; a binary string is k-universal if it contains every binary k-mer exactly once. For example, 0001110100 is a 3-universal string, as it contains each of the eight binary 3-mers (000, 001, 011, 111, 110, 101, 010, and 100) exactly once.

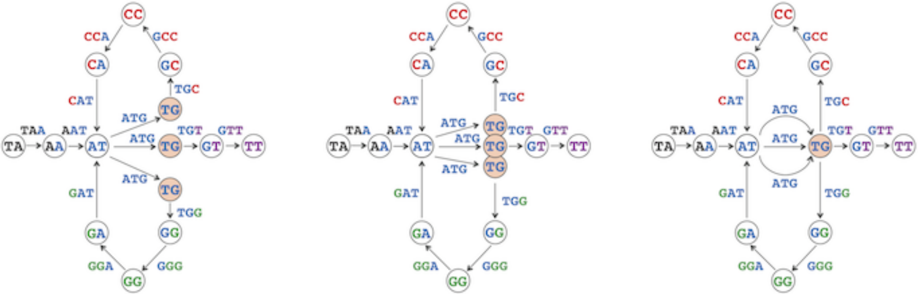
de Bruijn graphs


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

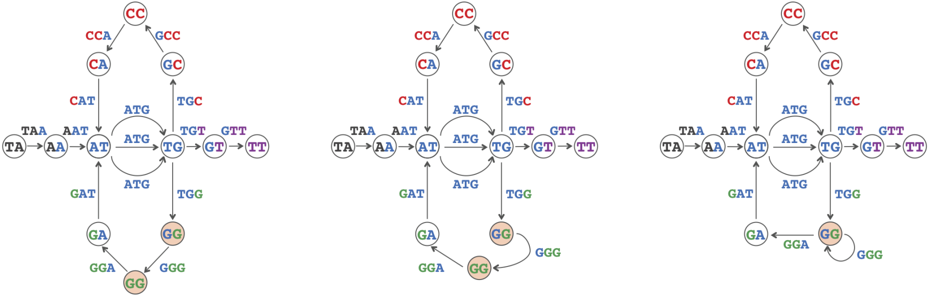
de Bruijn graphs



de Bruijn graphs



de Bruijn graphs



de Bruijn graphs

De Bruijn Graph from a String Problem

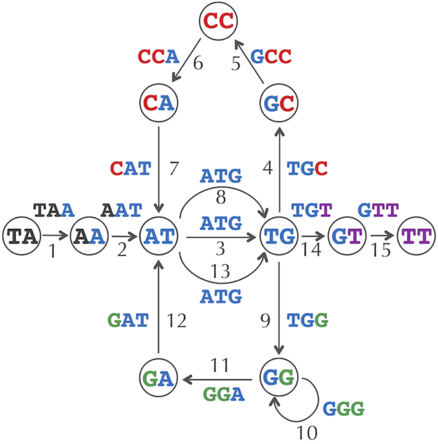
Construct the de Bruijn graph of a string.

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

ROSALIND:3D https://rosalind.info/problems/ba3d/

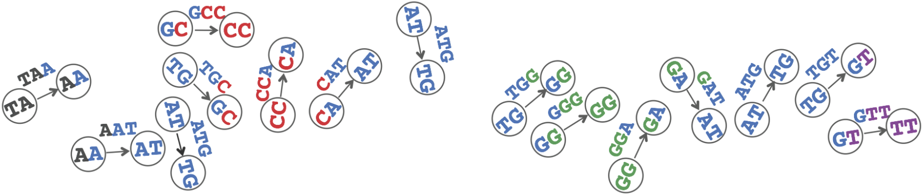
Try: AAGATTCTCTAAGA for k=4

Euler Path

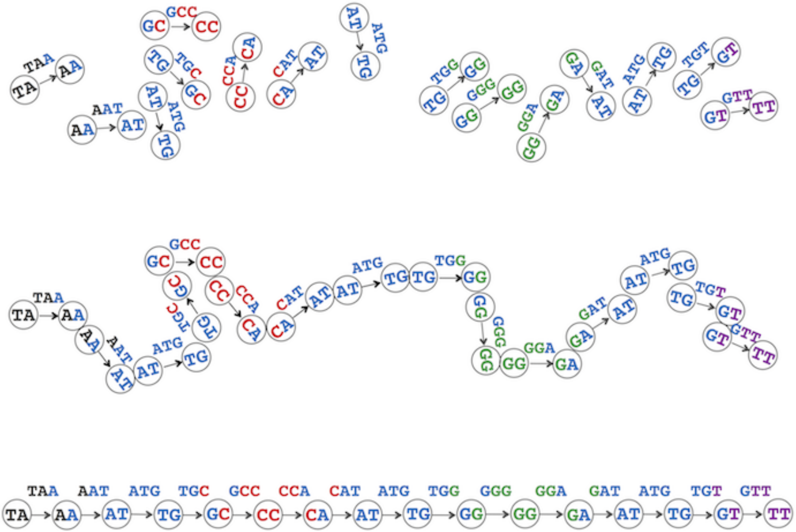


String Reconstruction == Euler Path Problem

de Bruijn Graphs



Composition$_3$ Graph


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

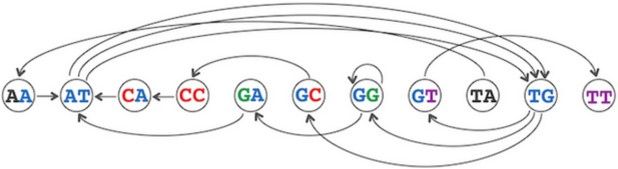
Composition$_3$ Graph

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

⇩

AA AT CA CC GA GC GG GT TA TG TT

⇩



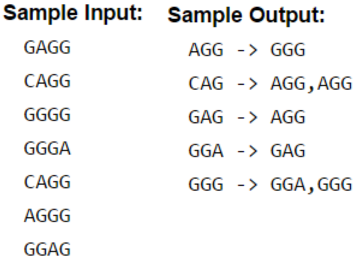
de Bruijn Graph

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:3E https://rosalind.info/problems/ba3e/



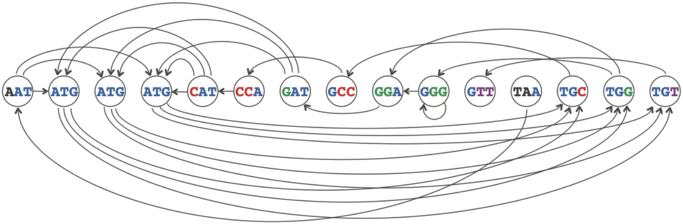
Sample Input:

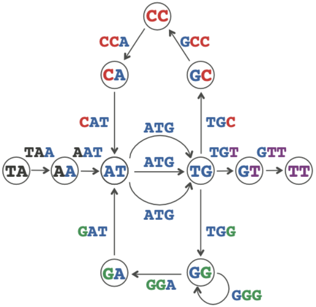
```
GAGG
CAGG
GGGG
GGGA
CAGG
AGGG
GGAG
```

Sample Output:

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

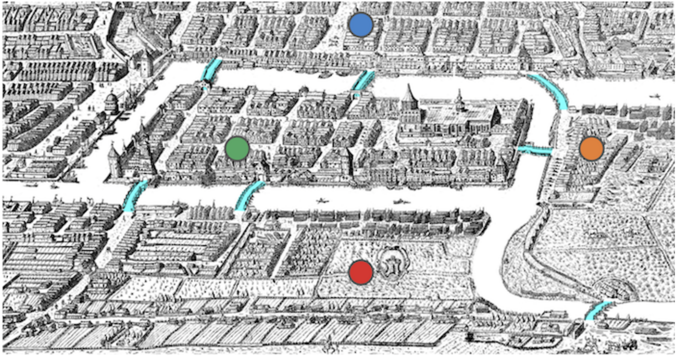
Euler Path vs Hamiltonian Path





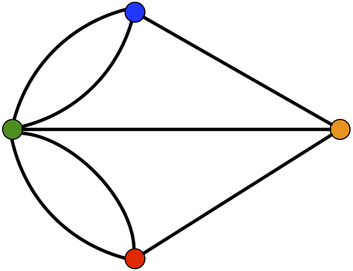
Bridges of Königsberg Problem.

The Prussian city of Königsberg in 1735 (Kaliningrad, Russia) comprised both banks of the Pregel River as well as two river islands; seven bridges connected these four different parts of the city. Königsberg's residents enjoyed taking walks, and they asked a simple question: Is it possible to set out from my house, cross each bridge exactly once, and return home?



The graph Königsberg.

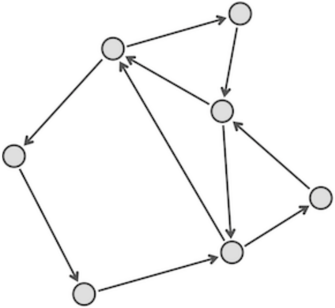
In 1735, Leonhard Euler drew the following graph, which we call Königsberg; this graph's nodes represent the four sectors of the city, and its edges represent the seven bridges connecting different sectors.

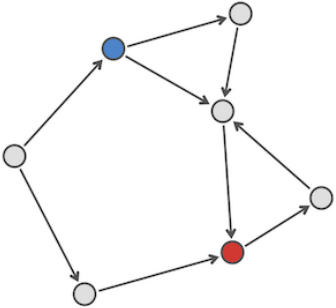


A cycle that traverses each edge of a graph exactly once is called an Eulerian cycle, and we say that a graph containing such a cycle is Eulerian.

The Euler's Theorem

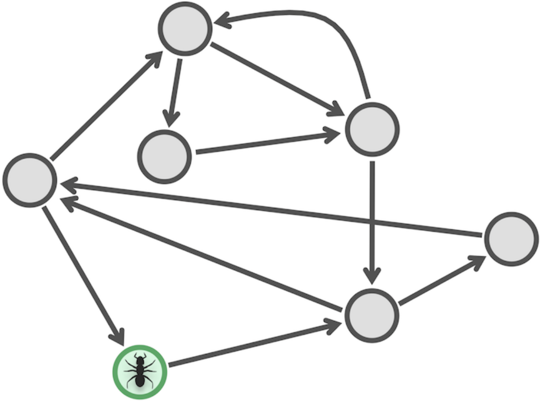
A node v is balanced if $\mathrm{IN}(v) = \mathrm{OUT}(v)$, and a graph is balanced if all its nodes are balanced.





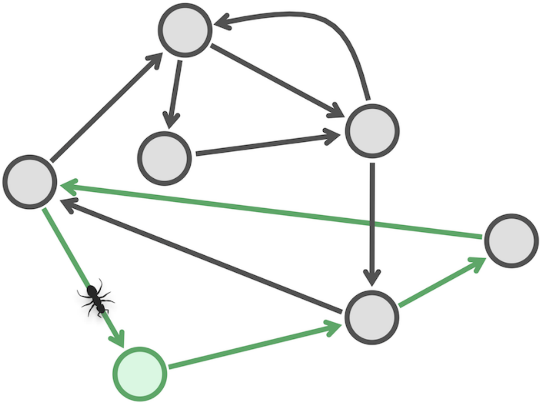
Euler's Theorem: Every balanced, strongly connected directed graph is Eulerian.

The Euler's Theorem



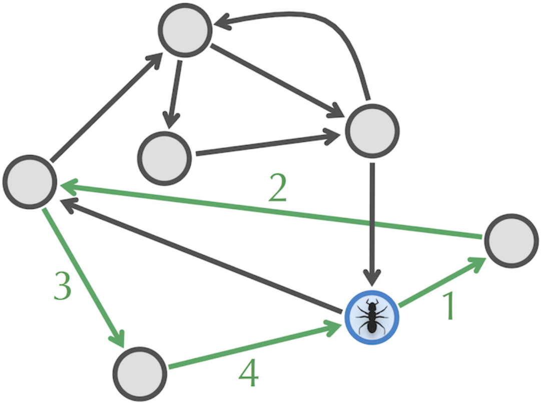
Euler's Theorem: Every balanced, strongly connected directed graph is Eulerian.

The Euler's Theorem



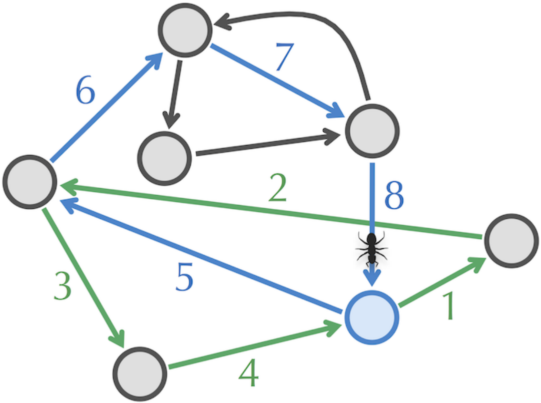
Euler's Theorem: Every balanced, strongly connected directed graph is Eulerian.

The Euler's Theorem



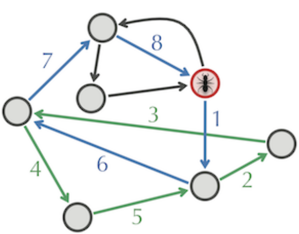
Euler's Theorem: Every balanced, strongly connected directed graph is Eulerian.

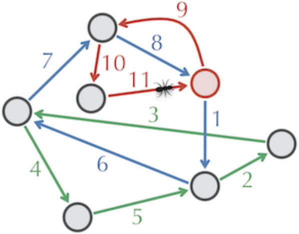
The Euler's Theorem



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

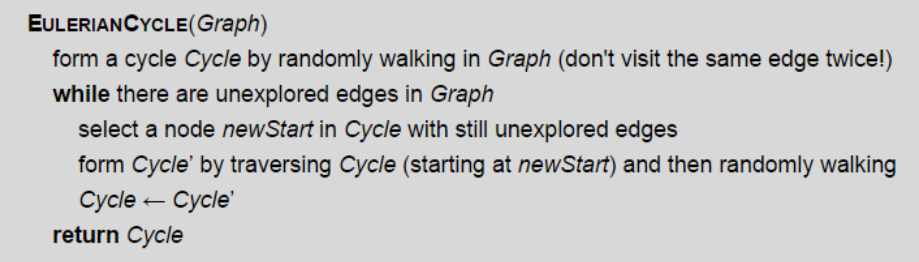
The Euler's Theorem





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

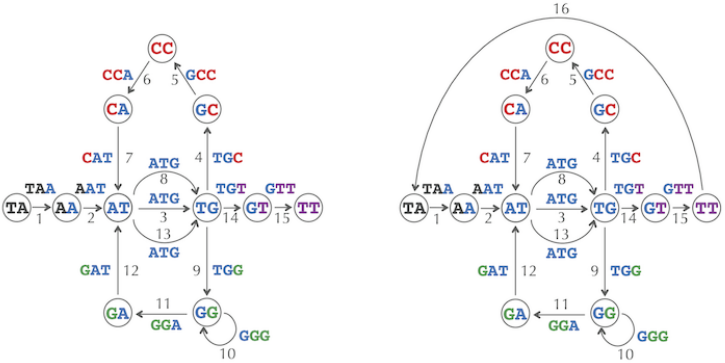
The Euler's Theorem



```
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.

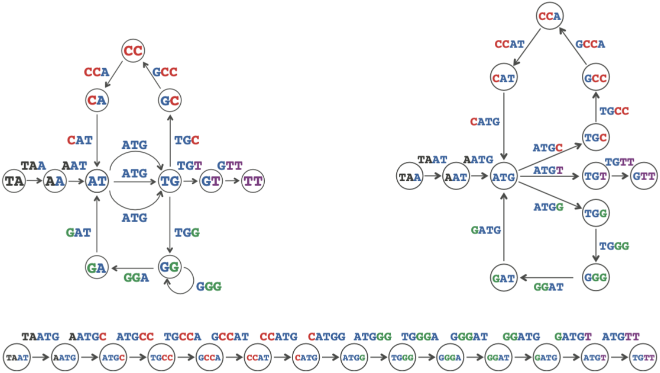
The Euler's Path to Cycle



A nearly balanced graph has an Eulerian path if and only if adding an edge between its unbalanced nodes makes the graph balanced and strongly connected.

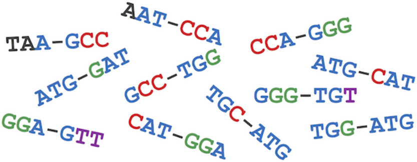
Some Issues

- Repeats - no unique paths!
- Longer reads - more errors



Read Pairs

- Biologists have suggested an indirect way of increasing read length by generating read-pairs, which are pairs of reads separated by a fixed distance *d* in the genome
- 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 TAATGCCATGGATGTT

Paired de Bruijn Graphs

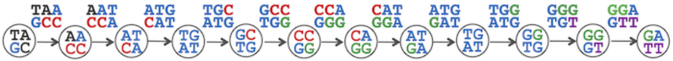
Given a (k, d)-mer $(a_1 \dots a_k \,|\, b_1, \dots b_k)$, we define its **prefix** and **suffix** as the following $(k-1, d+1)$-mers:

$$\textsc{Prefix}((a_1 \dots a_k \,|\, b_1, \dots b_k)) = (a_1 \dots a_{k-1} \,|\, b_1 \dots b_{k-1})$$
$$\textsc{Suffix}((a_1 \dots a_k \,|\, b_1, \dots b_k)) = (a_2 \dots a_k \,|\, b_2 \dots b_k)$$

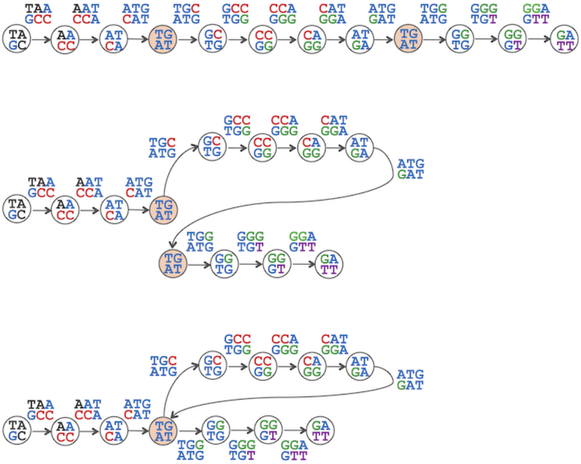
For example, $\textsc{Prefix}((\texttt{GAC} \,|\, \texttt{TCA})) = (\texttt{GA} \,|\, \texttt{TC})$ and $\textsc{Suffix}((\texttt{GAC} \,|\, \texttt{TCA})) = (\texttt{AC} \,|\, \texttt{CA})$.

Note that for consecutive (k, d)-mers appearing in *Text*, the suffix of the first (k, d)-mer is equal to the prefix of the second (k, d)-mer. For example, for the consecutive (k, d)-mers (**TAA** | **GCC**) and (**AAT** | **CCA**) in **TAATGCCATGGGATGTT**,

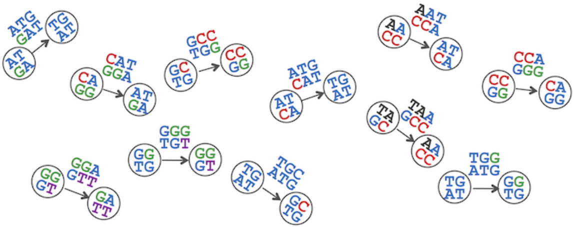
$$\textsc{Suffix}((\texttt{TAA} \,|\, \texttt{GCC})) = \textsc{Prefix}((\texttt{AAT} \,|\, \texttt{CCA})) = (\texttt{AA} \,|\, \texttt{CC})\,.$$



Paired de Bruijn Graphs

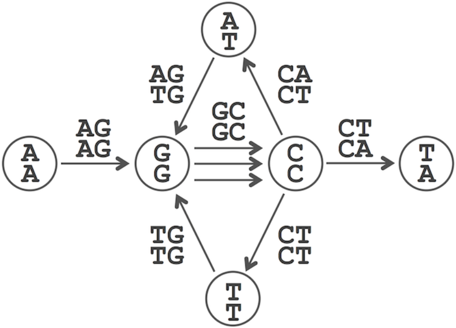


Paired de Bruijn Graphs



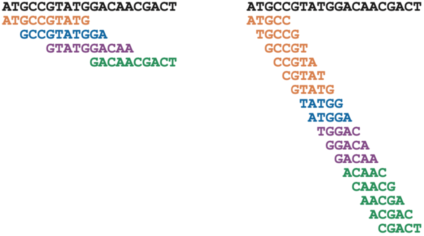
Paired de Bruijn Graphs

Every Eulerian path in the de Bruijn graph constructed from a k-mer composition spells out a solution of the String Reconstruction Problem. But is this the case for the paired de Bruijn graph?



Imperfect Coverage - Read Breaking

Four 10-mer reads that capture some but not all of the 10-mers in an example genome. Breaking these reads into shorter 5-mers, then these 5-mers exhibit perfect coverage.



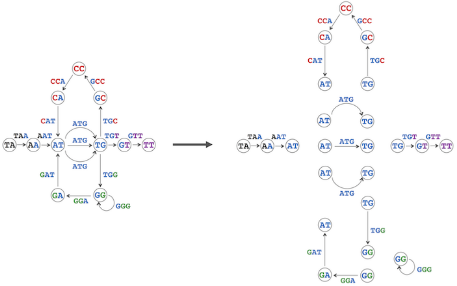
```
ATGCCGTATGGACAACGACT
ATGCCGTATG
  GCCGTATGGA
     GTATGGACAA
          GACAACGACT
```

```
ATGCCGTATGGACAACGACT
ATGCC
 TGCCG
  GCCGT
   CCGTA
    CGTAT
     GTATG
      TATGG
       ATGGA
        TGGAC
         GGACA
          GACAA
           ACAAC
            CAACG
             AACGA
              ACGAC
               CGACT
```

Read breaking must deal with a practical trade-off. On the one hand, the smaller the value of k, the larger the chance that the k-mer coverage is perfect. On the other hand, smaller values of k result in a more tangled de Bruijn graph, making it difficult to infer the genome from this graph.

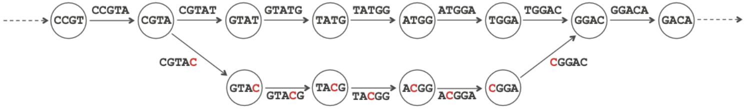
Contigs

- Even after read breaking, most assemblies still have gaps in k-mer coverage, causing the de Bruijn graph to have missing edges, and so the search for an Eulerian path fails.
- In this case, biologists often settle on assembling contigs (long, contiguous segments of the genome) rather than entire chromosomes.
- In practice, biologists have no choice but to break genomes into contigs, even in the case of perfect coverage, since repeats prevent them from being able to infer a unique Eulerian path.



Other issues

- Error prone reads may lead to bubbles.
- CGTACGGACA vs CGTATGGACA


CCGT
CCGTA
CGTA
CGTAT
GTAT
GTATG
TATG
TATGG
ATGG
ATGGA
TGGA
TGGAC
GGAC
GGACA
GACA
CGTAC
GTAC
GTACG
TACG
TACGG
ACGG
ACGGA
CGGA
CGGAC