How Do Biologists Assemble Genomic Puzzles from Millions of Pieces?

Graph Algorithms

Phillip Compeau and Pavel Pevzner

Bioinformatics Algorithms: an Active Learning Approach

Outline

Who Are These People?

Euler 1707-1783 Hamilton 1805-1865 De Bruijn 1918-2012

The human genome is a three billion nucleotide long "book" written in A, C, G, T alphabet.

Some genomes are 100 X larger than the human genome:

Amoeba dubia

Paris japonica

Why Do We Sequence 1000s of Species?

– Applications in **medicine** (genomes of fungi-producing bacteria), **agriculture** (oil palm genome), **biotechnology** (genomes of energy-producing cyanobacteria), etc., etc., etc.

Brief History of Genome Sequencing

- **1977**: Walter Gilbert and Frederick Sanger develop independent DNA sequencing methods.
- **1980**: They share the Nobel Prize.
- Still, their sequencing methods were too expensive ($3 billion to sequence the human genome).

Walter Gilbert

Frederick Sanger

The Race to Sequence the Human Genome

- **1990**: The public Human Genome Project, headed by Francis Collins, aims to sequence the human genome by 2005.

Francis Collins

The Race to Sequence the Human Genome

- **1990**: The public Human Genome Project, headed by Francis Collins, aims to sequence the human genome by 2005.

Francis Collins

- **1997**: Craig Venter founds Celera Genomics, a private firm, with the same goal.

Craig Venter

The Race to Sequence the Human Genome

- **1990**: The public Human Genome Project, headed by Francis Collins, aims to sequence the human genome by 2005.

Francis Collins

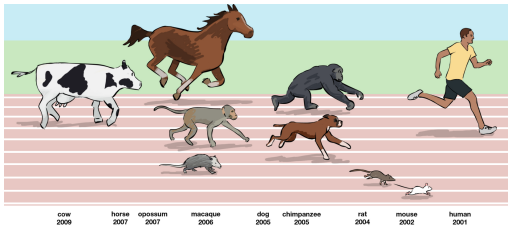
- **1997**: Craig Venter founds Celera Genomics, a private firm, with the same goal.

Craig Venter

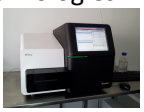
- **2000:**

From Human to Mouse to Rat to …

Early 2000s: Many more mammalian genomes are sequenced using the same Sanger sequencing method, but it is clear that new technology is needed for further progress.



Next Generation Sequencing Technologies



- **Late 2000s**: The market for new sequencing machines takes off.
  - Illumina reduces the cost of sequencing a human genome from $3 billion to $10,000.
  - Complete Genomics builds a genomic factory in Silicon Valley that sequences hundreds of genomes per month.
  - Beijing Genome Institute orders hundreds of sequencing machines, becoming the world's largest sequencing center.

illumina

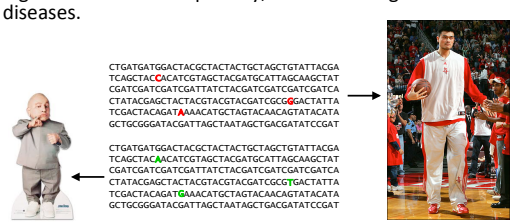
Complete genomics

华大基因 BGI

Personal Genome Sequencing

Few Mutations Can Make a Big Difference…

- Different people have slightly different genomes: on average, roughly 1 mutation in 1000 nucleotides.
- The 1 in 1000 nucleotides difference accounts for height, high cholesterol susceptibility, and 1000s of genetic diseases.



Why Do We Sequence Personal Genomes?

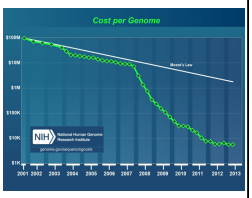
- **2010:** Nicholas Volker became first human being to be saved by genome sequencing.

  - Doctors could not diagnose his condition; he went through dozens of surgeries.
  - Sequencing revealed a rare mutation in a *XIAP* gene linked to a defect in his immune system.
  - This led doctors to use immunotherapy, which saved the child.

10,000 Genomes and Beyond

- **2010**: Scientists launch a project to sequence 10,000 vertebrate genomes.

- **Now**: Human genome sequencing costs just a few thousand dollars and under $1,000 human genomes may arrive any day now.



Outline

- What Is Genome Sequencing?
- **Exploding Newspapers**
- The String Reconstruction Problem
- String Reconstruction as a Hamiltonian Path Problem
- String Reconstruction as an Eulerian Path Problem
- Similar Problems with Different Fates
- De Bruijn Graphs
- Euler's Theorem
- Assembling Read-Pairs
- De Bruijn Graphs Face Harsh Realities of Assembly

The Newspaper Problem

stack of NY Times, June 27, 2000

The Newspaper Problem

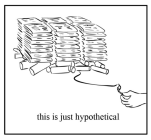
stack of NY Times, June 27, 2000

stack of NY Times, June 27, 2000 on a pile of dynamite

The Newspaper Problem

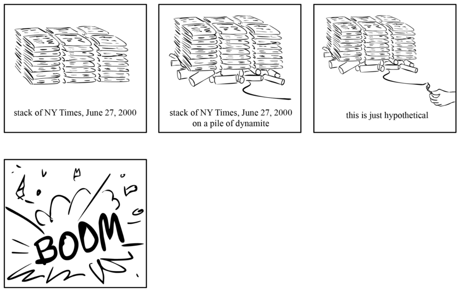
stack of NY Times, June 27, 2000

stack of NY Times, June 27, 2000 on a pile of dynamite

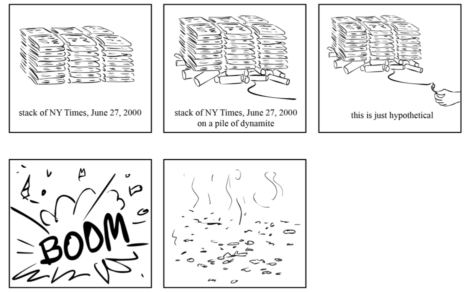


this is just hypothetical

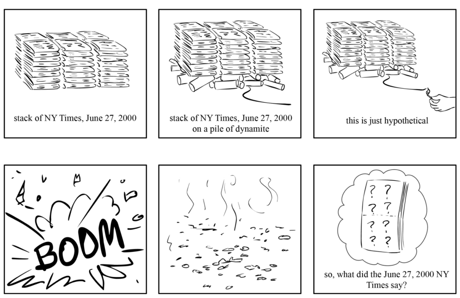
The Newspaper Problem



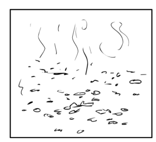
The Newspaper Problem

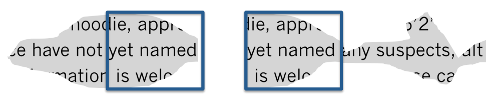


The Newspaper Problem



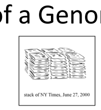
The Newspaper Problem as an Overlapping Puzzle





The Newspaper Problem as an Overlapping Puzzle

Multiple Copies of a Genome (Millions of them)



CTGATGATGGACTACGCTACTACTGCTAGCTGTATTACGATCAGCTACCACATCGTAGCTACGATGCATTAGCAAGCTATCGGATCAGCTACCACATCGTAGC
CTGATGATGGACTACGCTACTACTGCTAGCTGTATTACGATCAGCTACCACATCGTAGCTACGATGCATTAGCAAGCTATCGGATCAGCTACCACATCGTAGC
CTGATGATGGACTACGCTACTACTGCTAGCTGTATTACGATCAGCTACCACATCGTAGCTACGATGCATTAGCAAGCTATCGGATCAGCTACCACATCGTAGC
CTGATGATGGACTACGCTACTACTGCTAGCTGTATTACGATCAGCTACCACATCGTAGCTACGATGCATTAGCAAGCTATCGGATCAGCTACCACATCGTAGC

Breaking the Genomes at Random Positions

CTGATGA TGGACTACGCTAC TACTGCTAG CTGTATTACG ATCAGCTACCACA TCGTAGCTACG ATGCATTAGCAA GCTATCGGA TCAGCTACCA CATCGTAGC
CTGATGATG GACTACGCT ACTACTGCTA GCTGTATTACG ATCAGCTACC ACATCGTAGCT ACGATGCATTA GCAAGCTATC GGATCAGCTAC CACATCGTAGC
CTGATGATGG ACTACGCTAC TACTGCTAGCT GTATTACGATC AGCTACCAC ATCGTAGCTACG ATGCATTAGCA AGCTATCGG A TCAGCTACCA CATCGTAGC
CTGATGATGGACT ACGCTACTACT GCTAGCTGTAT TACGATCAGC TACCACATCGT AGCTACGATGCA TTAGCAAGCT ATCGGATCA GCTACCACATC GTAGC

Generating “Reads”

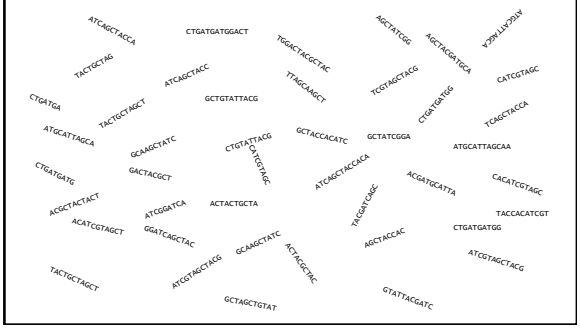
CTGATGA TGGACTACGCTAC TACTGCTAG CTGTATTACG ATCAGCTACCACA TCGTAGCTACG ATGCATTAGCAA GCTATCGGA TCAGCTACCA CATCGTAGC
CTGATGATG GACTACGCT ACTACTGCTA GCTGTATTACG ATCAGCTACC ACATCGTAGCT ACGATGCATTA GCAAGCTATC GGATCAGCTAC CACATCGTAGC
CTGATGATGG ACTACGCTAC TACTGCTAGCT GTATTACGATC AGCTACCAC ATCGTAGCTACG ATGCATTAGCA AGCTATCGG A TCAGCTACCA CATCGTAGC
CTGATGATGGACT ACGCTACTACT GCTAGCTGTAT TACGATCAGC TACCACATCGT AGCTACGATGCA TTAGCAAGCT ATCGGATCA GCTACCACATC GTAGC

“Burning” Some Reads

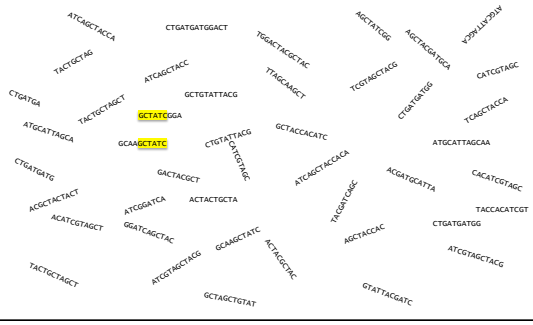
CTGATGA TGGACTACGCTAC TACTGCTAG CTGTATTACG ATCAGCTACCACA TCGTAGCTACG ATGCATTAGCAA GCTATCGGA TCAGCTACCA CATCGTAGC
CTGATGATG GACTACGCT ACTACTGCTA GCTGTATTACG ATCAGCTACC ACATCGTAGCT ACGATGCATTA GCAAGCTATC GGATCAGCTAC CACATCGTAGC
CTGATGATGG ACTACGCTAC TACTGCTAGCT GTATTACGATC AGCTACCAC ATCGTAGCTACG ATGCATTAGCA AGCTATCGG A TCAGCTACCA CATCGTAGC
CTGATGATGGACT ACGCTACTACT GCTAGCTGTAT TACGATCAGC TACCACATCGT AGCTACGATGCA TTAGCAAGCT ATCGGATCA GCTACCACATC GTAGC

No Idea What Position Every Read Comes From

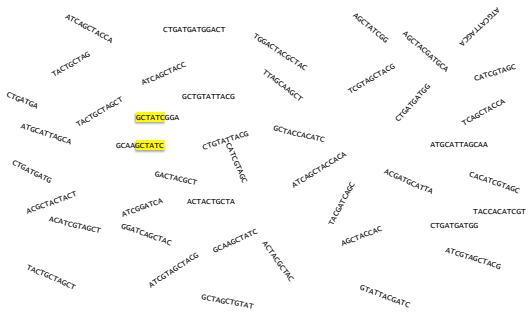
No Idea What Position Every Read Comes From



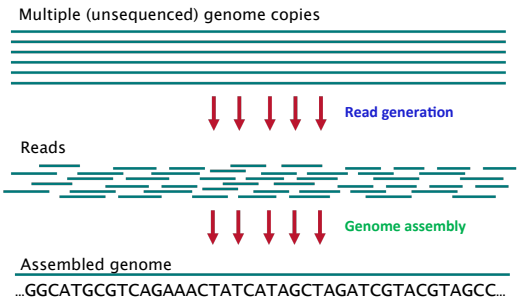
No Idea What Position Every Read Comes From



No Idea What Position Every Read Comes From



From Experimental to Computational Challenges



What Makes Genome Sequencing Difficult?

- Modern sequencing machines cannot read an entire genome one nucleotide at a time from beginning to end (like we read a book)
- They can only shred the genome and generate short **reads**.
- The genome assembly is not the same as a jigsaw puzzle: we must use *overlapping* reads to reconstruct the genome, a giant **overlap puzzle**!

Outline

- What Is Genome Sequencing?
- Exploding Newspapers
- **The String Reconstruction Problem**
- String Reconstruction as a Hamiltonian Path Problem
- String Reconstruction as an Eulerian Path Problem
- Similar Problems with Different Fates
- De Bruijn Graphs
- Euler's Theorem
- Assembling Read-Pairs
- De Bruijn Graphs Face Harsh Realities of Assembly

The Genome Sequencing Problem

Genome Sequencing Problem. Reconstruct a genome from reads.
- **Input**. A collection of strings *Reads*.
- **Output**. A string *Genome* reconstructed from *Reads*.

What Is *k*-mer Composition?

$Composition_3$(TAATGCCATGGGATGTT) =

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

k-mer Composition

$Composition_3$(TAATGCCATGGGATGTT) =
TAA AAT ATG TGC GCC CCA CAT ATG TGG GGG GGA GAT ATG TGT GTT
=
AAT ATG ATG ATG CAT CCA GAT GCC GGA GGG GTT TAA TGC TGG TGT

e.g., lexicographic order (like in a dictionary)

Reconstructing a String from its Composition

String Reconstruction Problem. Reconstruct a string from its *k*-mer composition.

- **Input**. A collection of *k*-mers.
- **Output**. A *Genome* such that $Composition_k(Genome)$ is equal to the collection of *k*-mers.

A Naive String Reconstruction Approach

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

A Naive String Reconstruction Approach

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

TAA

A Naive String Reconstruction Approach

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

TAA

A Naive String Reconstruction Approach

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

TAA
AAT

A Naive String Reconstruction Approach

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

TAA
AAT

A Naive String Reconstruction Approach

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

TAA
AAT
ATG

A Naive String Reconstruction Approach

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

TAA
AAT
ATG

A Naive String Reconstruction Approach

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

TAA
AAT
ATG

A Naive String Reconstruction Approach

ATG ATG CAT CCA GAT GCC GGA GGG GTT TGC TGG

TAA
AAT
ATG
TGT

A Naive String Reconstruction Approach

ATG ATG CAT CCA GAT GCC GGA GGG GTT TGC TGG

TAA
AAT
ATG
TGT

What's Next?

ATG ATG CAT CCA GAT GCC GGA GGG TGC TGG

TAA
AAT
ATG
TGT
GTT

Outline

- What Is Genome Sequencing?
- Exploding Newspapers
- The String Reconstruction Problem
- **String Reconstruction as a Hamiltonian Path Problem**
- String Reconstruction as an Eulerian Path Problem
- Similar Problems with Different Fates
- De Bruijn Graphs
- Euler's Theorem
- Assembling Read-Pairs
- De Bruijn Graphs Face Harsh Realities of Assembly

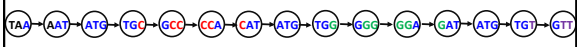
Representing a Genome as a Path

$Composition_3$(TAATGCCATGGGATGTT) =

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

Representing a Genome as a Path

$Composition_3$(TAATGCCATGGGATGTT) =



Representing a Genome as a Path

$Composition_3$(TAATGCCATGGGATGTT) =

Can we construct this genome path without knowing the genome **TAATGCCATGGGATGTT**, only from its composition?

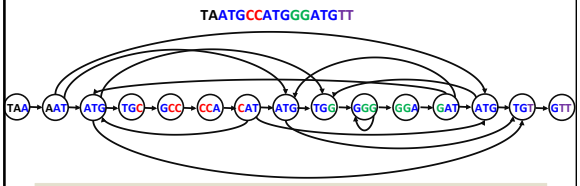
Representing a Genome as a Path

$Composition_3$(TAATGCCATGGGATGTT) =

Can we construct this genome path without knowing the genome **TAATGCCATGGGATGTT**, only from its composition?

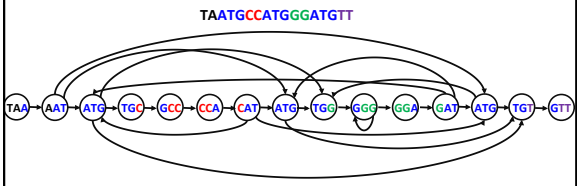
Yes. We simply need to connect k-mer$_1$ with k-mer$_2$ if ***suffix***(k-mer$_1$)=***prefix***(k-mer$_2$).
E.g. **TAA** → **AAT**

A Path Turns into a Graph



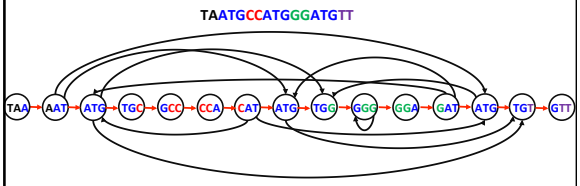
Yes. We simply need to connect k-mer$_1$ with k-mer$_2$ if ***suffix***(k-mer$_1$)=***prefix***(k-mer$_2$).
E.g. **TAA** → **AAT**

A Path Turns into a Graph



Can we still find the genome path in this graph?

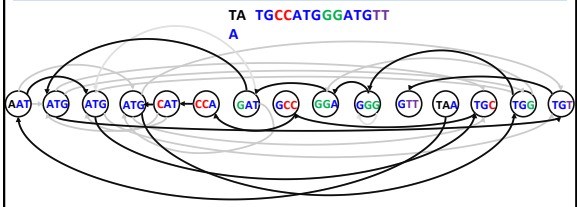
A Path Turns into a Graph



Can we still find the genome path in this graph?

Where Is the Genomic Path?

A **Hamiltonian path:** a path that visits each node in a graph exactly once.



What are we trying to find in this graph?

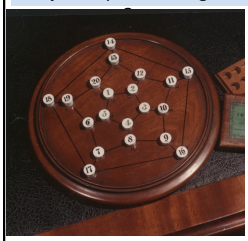
Hamiltonian Path Problem

Hamiltonian Path Problem. Find a Hamiltonian path in a graph.

- **Input.** A graph.
- **Output.** A path visiting every **node** in the graph exactly once.

Does This Graph Have a Hamiltonian Path?

Hamiltonian Path Problem. Find a Hamiltonian path in a graph.
Input. A graph.
Output. A path visiting every **node** in the graph exactly once.

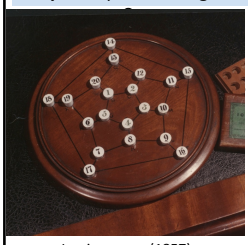


Icosian game (1857)

William Hamilton

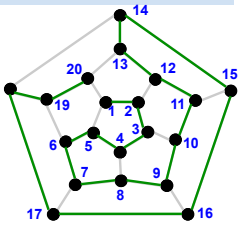
Does This Graph Have a Hamiltonian Path?

Hamiltonian Path Problem. Find a Hamiltonian path in a graph.
Input. A graph.
Output. A path visiting every **node** in the graph exactly once.



Icosian game (1857)

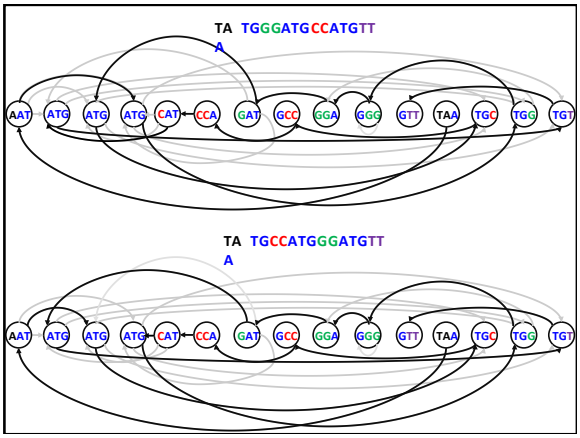
William Hamilton



Undirected graph

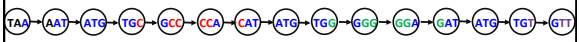
Outline

- What Is Genome Sequencing?
- Exploding Newspapers
- The String Reconstruction Problem
- String Reconstruction as a Hamiltonian Path Problem
- **String Reconstruction as an Eulerian Path Problem**
- Similar Problems with Different Fates
- De Bruijn Graphs
- Euler's Theorem
- Assembling Read-Pairs
- De Bruijn Graphs Face Harsh Realities of Assembly

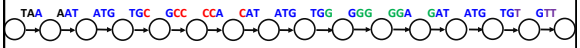


A Slightly Different Path

TAATGCCATGGGATGTT



3-mers as **nodes**



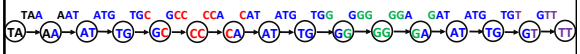
3-mers as **edges**

How do we label the starting and ending nodes of an edge?

Labeling Nodes in the New Path

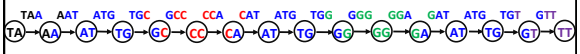
TAATGCCATGGGATGTT

3-mers as **nodes**



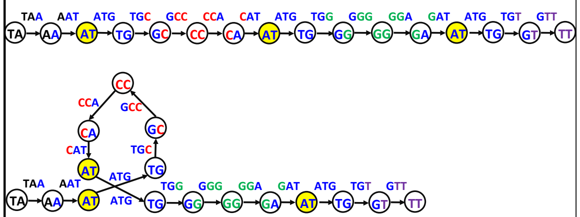
3-mers as **edges** and 2-mers as **nodes**

Labeling Nodes in the New Path

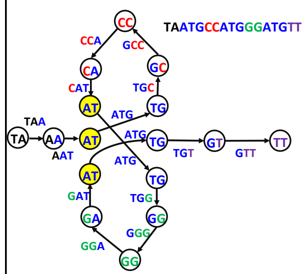


3-mers as **edges** and 2-mers as **nodes**

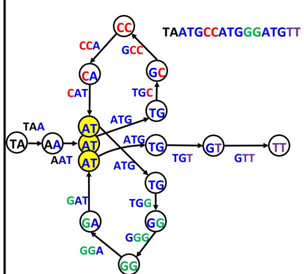
Gluing Identically Labeled Nodes



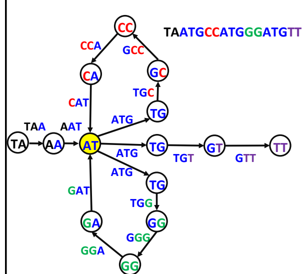
Gluing Identically Labeled Nodes



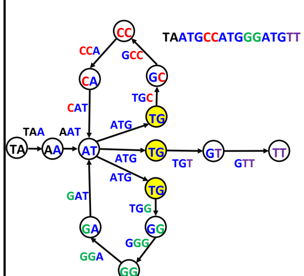
Gluing Identically Labeled Nodes



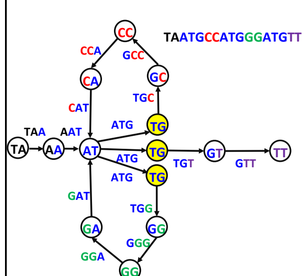
Gluing Identically Labeled Nodes



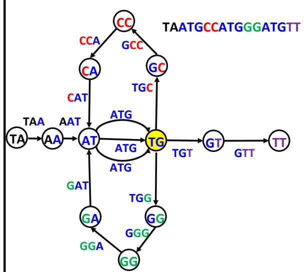
Gluing Identically Labeled Nodes



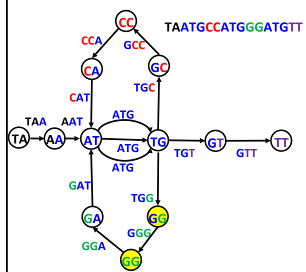
Gluing Identically Labeled Nodes



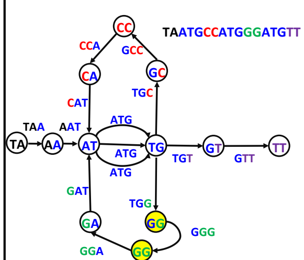
Gluing Identically Labeled Nodes



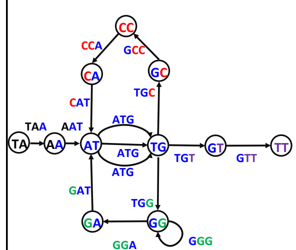
Gluing Identically Labeled Nodes



Gluing Identically Labeled Nodes



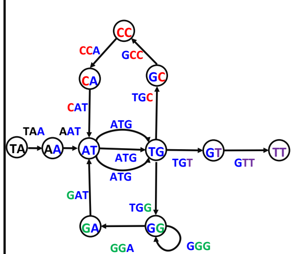
De Bruijn Graph of **TAATGCCATGGGATGTT**



Where is the *Genome* hiding in this graph?

It Was Always There!

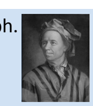
TA TGCCATGGGATGTT
A



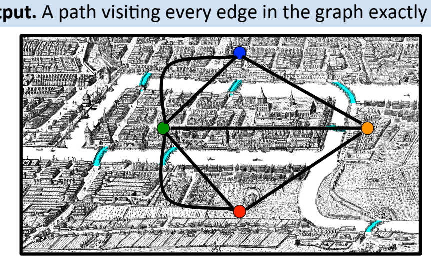
An **Eulerian path** in a graph is a path that visits each **edge** exactly once.

Eulerian Path Problem

Eulerian Path Problem. Find an Eulerian path in a graph.



- **Input.** A graph.
- **Output.** A path visiting every edge in the graph exactly once.



Eulerian Versus Hamiltonian Paths

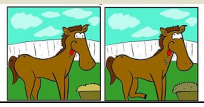
Eulerian Path Problem. Find an Eulerian path in a graph.

- **Input.** A graph.
- **Output.** A path visiting every edge in the graph exactly once.

Hamiltonian Path Problem. Find a Hamiltonian path in a graph.

- **Input.** A graph.
- **Output.** A path visiting every node in the graph exactly once.

Find a difference!



Outline

- What Is Genome Sequencing?
- Exploding Newspapers
- The String Reconstruction Problem
- String Reconstruction as a Hamiltonian Path Problem
- String Reconstruction as an Eulerian Path Problem
- **Similar Problems with Different Fates**
- De Bruijn Graphs
- Euler's Theorem
- Assembling Read-Pairs
- De Bruijn Graphs Face Harsh Realities of Assembly

Eulerian Versus Hamiltonian Paths

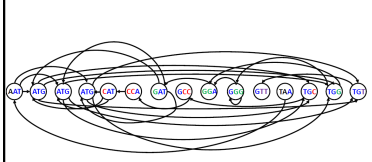
Eulerian Path Problem. Find an **Eulerian** path in a graph.

- **Input.** A graph.
- **Output.** A path visiting every **edge** in the graph exactly once.

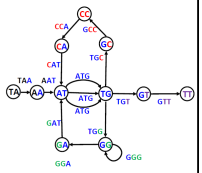
Hamiltonian Path Problem. Find a **Hamiltonian** path in a graph.

- **Input.** A graph.
- **Output.** A path visiting every **node** in the graph exactly once.

What Problem Would You Prefer to Solve?



Hamiltonian Path Problem



Eulerian Path Problem

While Euler solved the Eulerian Path Problem (even for a city with a million bridges), nobody has developed a fast algorithm for the Hamiltonian Path Problem yet.

NP-Complete Problems

- The Hamiltonian Path Problem belongs to a collection containing thousands of computational problems for which no fast algorithms are known.

"I can't find an efficient algorithm, I guess I'm just too dumb."

From Garey and Johnson. Computers and Intractability. 1979

Change of Attitude

That would be an excellent argument, but the question of whether or not NP-Complete problems can be solved efficiently is one of seven **Millennium Problems** in mathematics.

"I can't find an efficient algorithm, because no such algorithm is possible."

From Garey and Johnson. Computers and Intractability. 1979

The Modern State of Affairs

NP-Complete problems are all equivalent: find an efficient solution to one, and you have an efficient solution to them all.

"I can't find an efficient algorithm, but neither can all these famous people."

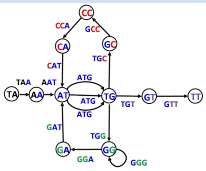
Outline

- What Is Genome Sequencing?
- Exploding Newspapers
- The String Reconstruction Problem
- String Reconstruction as a Hamiltonian Path Problem
- String Reconstruction as an Eulerian Path Problem
- Similar Problems with Different Fates
- **De Bruijn Graphs**
- Euler's Theorem
- Assembling Read-Pairs
- De Bruijn Graphs Face Harsh Realities of Assembly

Eulerian Path Problem

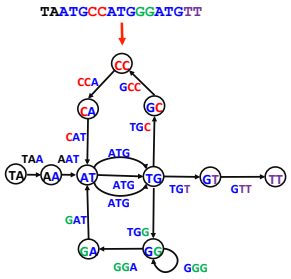
Eulerian Path Problem. Find an **Eulerian** path in a graph.

- **Input.** A graph.
- **Output.** A path visiting every **edge** in the graph exactly once.

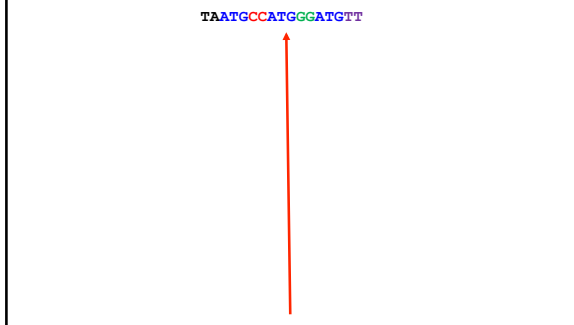


We constructed the de Bruijn graph from *Genome*, but in reality, *Genome* is unknown!

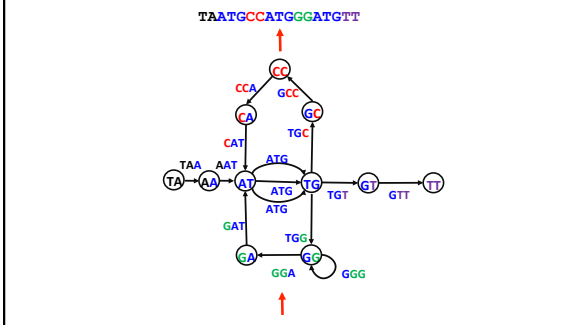
What We Have Done: From *Genome* to de Bruijn Graph



What We Want: From Reads (*k*-mers) to *Genome*



What We will Show: From Reads to de Bruijn Graph to *Genome*



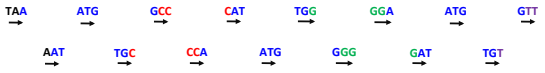
Constructing de Bruijn Graph when *Genome* Is Known

TAATGCCATGGGATGTT

Constructing de Bruijn when *Genome* Is Unknown

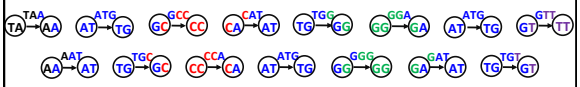
Composition$_3$(**TAATGCCATGGGATGTT**)

Representing Composition as a Graph Consisting of Isolated Edges



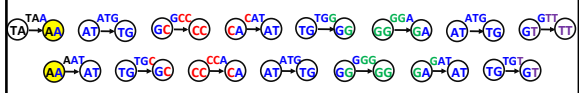
Composition$_3$(**TAATGCCATGGGATGTT**)

Constructing de Bruijn Graph from *k*-mer Composition

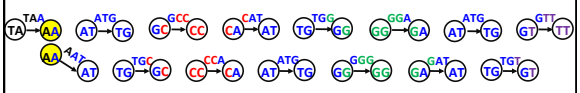


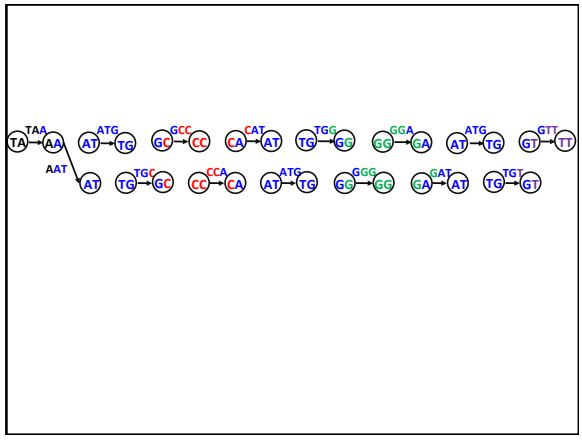
Composition$_3$(**TAATGCCATGGGATGTT**)

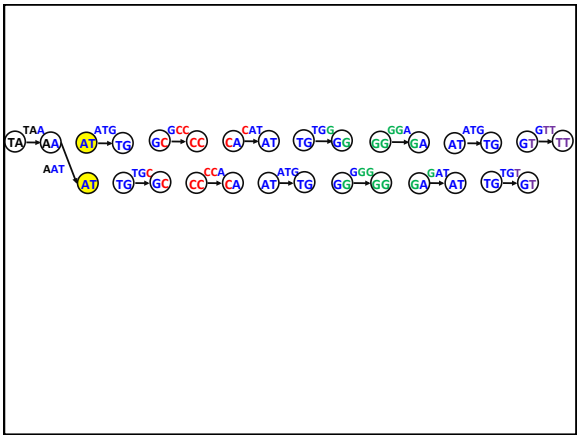
Gluing Identically Labeled Nodes

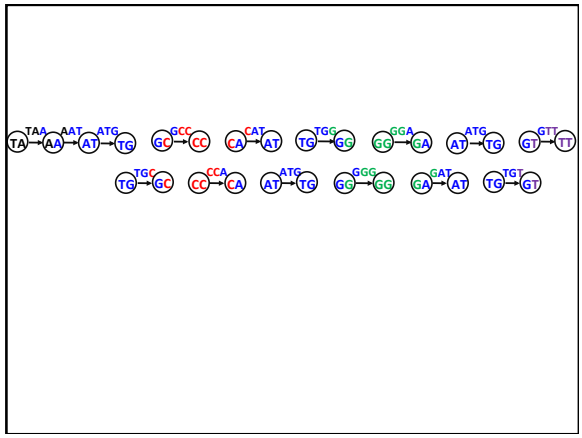


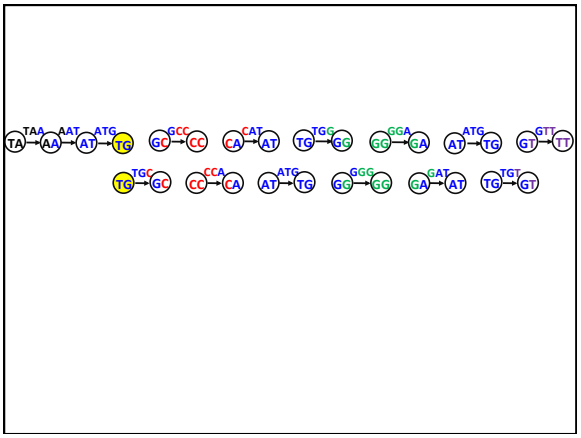
Gluing Identically Labeled Nodes

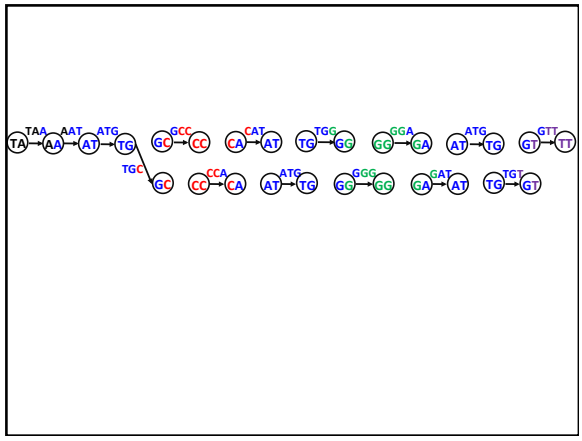


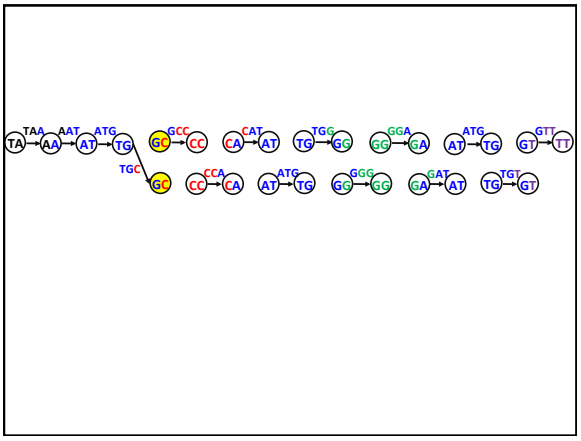

TAA
AAT
ATG
GCC
CAT
TGG
GGA
ATG
GTT
TGC
CCA
ATG
GGG
GAT
TGT

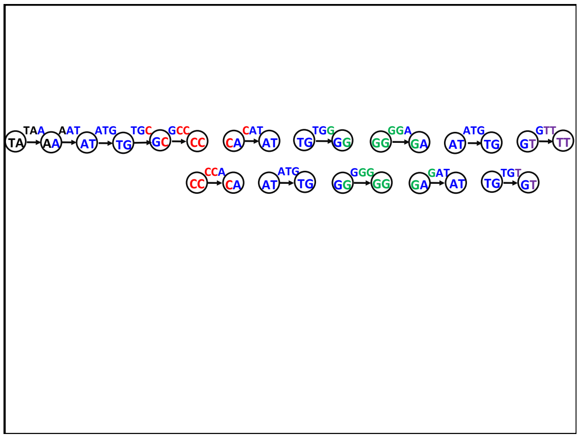

TAA
AAT
ATG
GCC
CAT
TGG
GGA
ATG
GTT
TGC
CCA
ATG
GGG
GAT
TGT

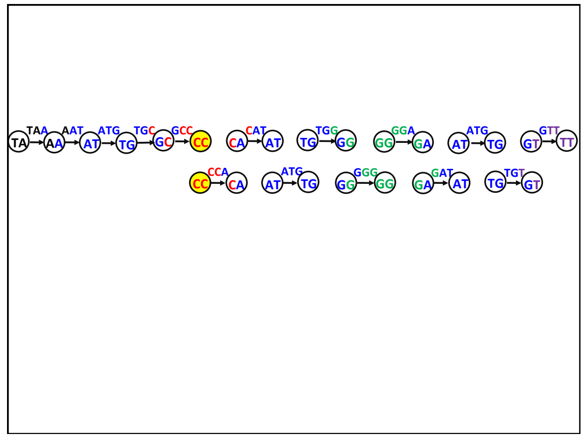

TAA
AAT
ATG
GCC
CAT
TGG
GGA
ATG
GTT
TGC
CCA
ATG
GGG
GAT
TGT

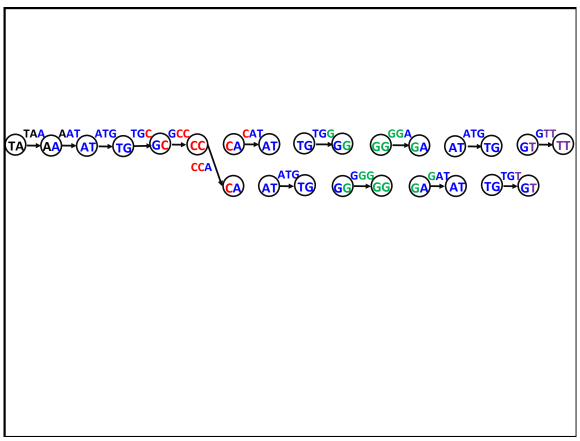

TAA
AAT
ATG
GCC
CAT
TGG
GGA
ATG
GTT
TGC
CCA
ATG
GGG
GAT
TGT

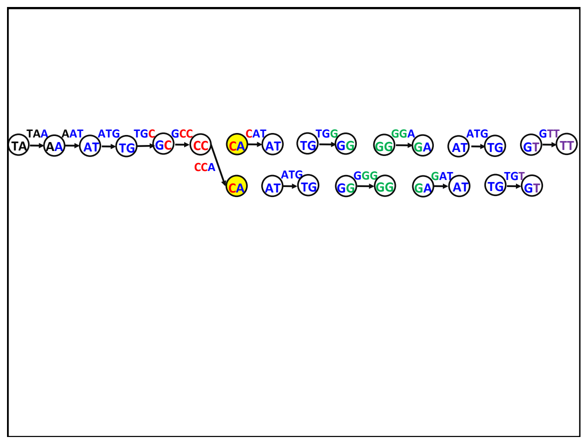

TAA
AAT
ATG
GCC
CAT
TGG
GGA
ATG
GTT
TGC
CCA
ATG
GGG
GAT
TGT

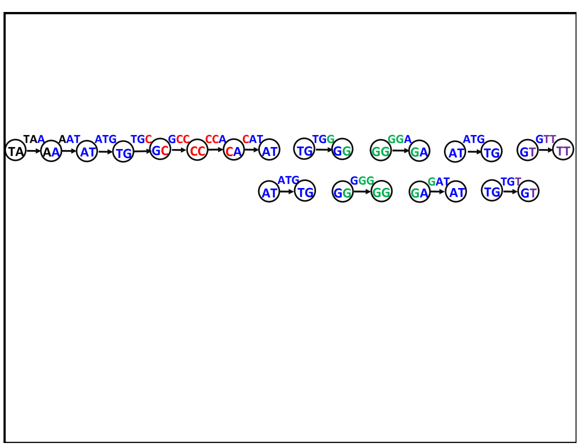

TAA
AAT
ATG
GCC
CAT
TGG
GGA
ATG
GTT
TGC
CCA
ATG
GGG
GAT
TGT

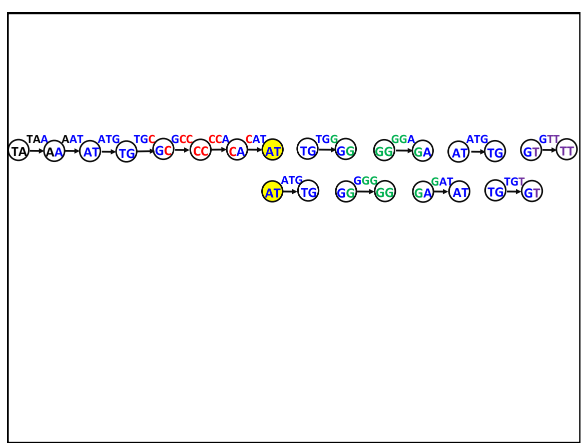

TA TAA AA AAT AT ATG TG TGC GC GCC CC CA CAT AT TG TGG GG GG GGA GA AT ATG TG GT GTT TT
CC CCA CA AT ATG TG GG GGG GG GA GAT AT TG TGT GT

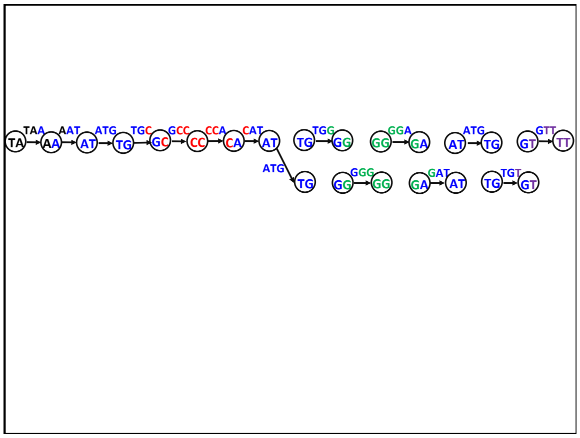

TA TAA AA AAT AT ATG TG TGC GC GCC CC CA CAT AT TG TGG GG GG GGA GA AT ATG TG GT GTT TT
CC CCA CA AT ATG TG GG GGG GG GA GAT AT TG TGT GT

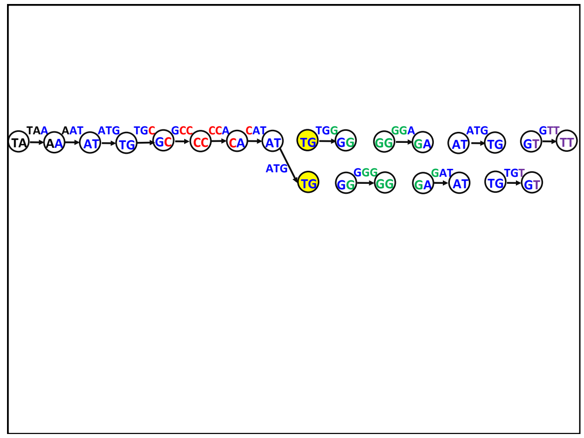

TA TAA AA AAT AT ATG TG TGC GC GCC CC CA CAT AT TG TGG GG GG GGA GA AT ATG TG GT GTT TT
CCA
CA AT ATG TG GG GGG GG GA GAT AT TG TGT GT

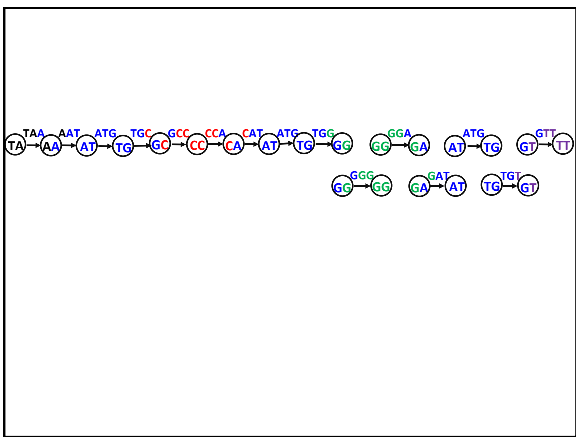

TA TAA AA AAT AT ATG TG TGC GC GCC CC CA CAT AT TG TGG GG GG GGA GA AT ATG TG GT GTT TT
CCA
CA AT ATG TG GG GGG GG GA GAT AT TG TGT GT

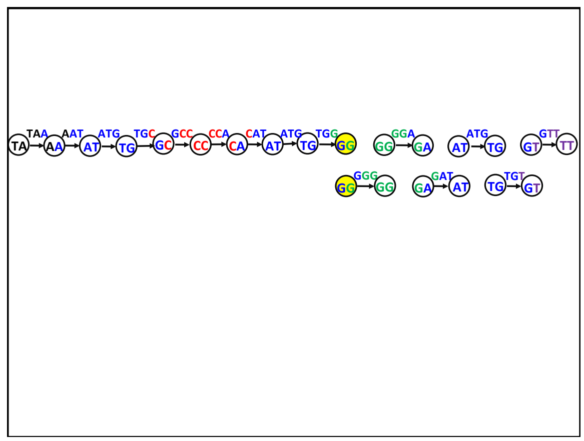

TA TAA AA AAT AT ATG TG TGC GC GCC CC CCA CA CAT AT TG TGG GG GG GGA GA AT ATG TG GT GTT TT
AT ATG TG GG GGG GG GA GAT AT TG TGT GT

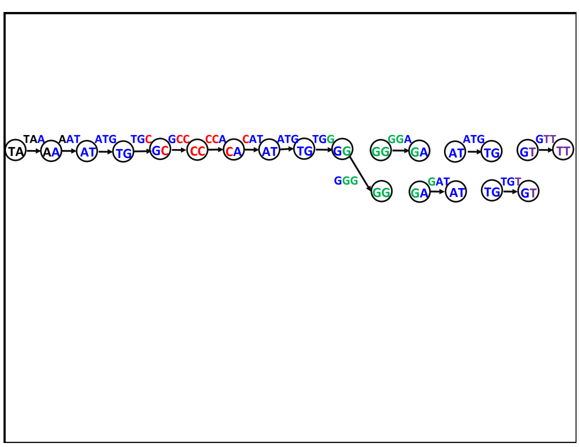

TA TAA AA AAT AT ATG TG TGC GC GCC CC CCA CA CAT AT TG TGG GG GG GGA GA AT ATG TG GT GTT TT
AT ATG TG GG GGG GG GA GAT AT TG TGT GT

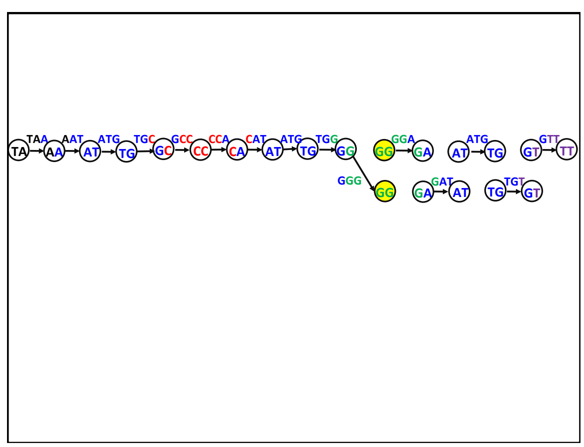

TA TAA AA AAT AT ATG TG TGC GC GCC CC CCA CA CAT AT TG TGG GG GG GGA GA AT ATG TG GT GTT TT
ATG TG GG GGG GG GA GAT AT TG TGT GT

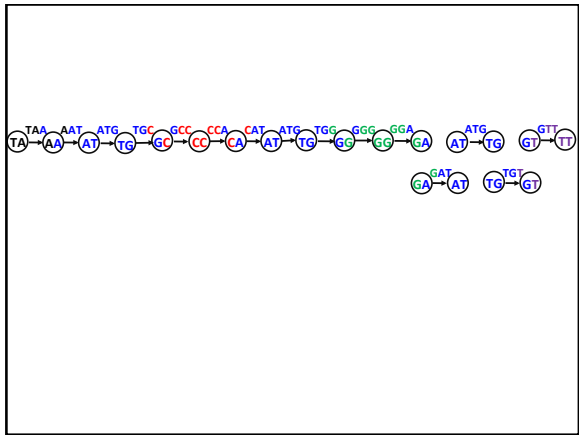

TA TAA AA AAT AT ATG TG TGC GC GCC CC CCA CA CAT AT TG TGG GG GG GGA GA AT ATG TG GT GTT TT
ATG TG GG GGG GG GA GAT AT TG TGT GT

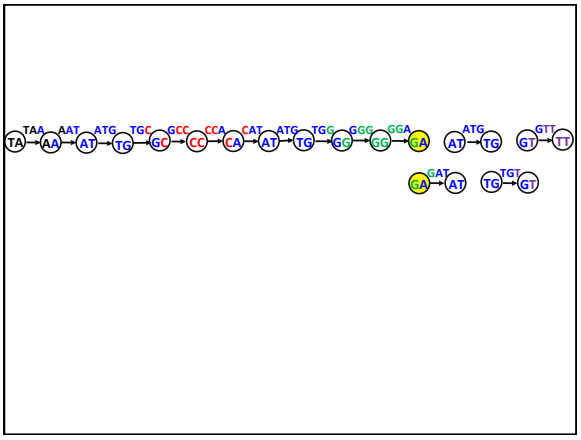

TA TAA AA AAT AT ATG TG TGC GC GCC CC CCA CA CAT AT ATG TG TGG GG GG GGA GA AT ATG TG GT GTT TT
GG GGG GG GA GAT AT TG TGT GT

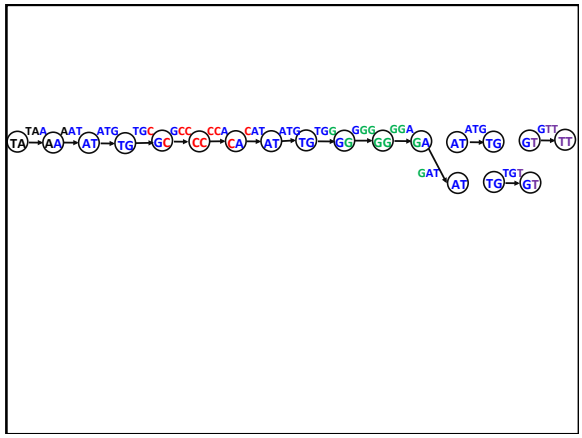

TA TAA AA AAT AT ATG TG TGC GC GCC CC CCA CA CAT AT ATG TG TGG GG GG GGA GA AT ATG TG GT GTT TT
GG GGG GG GA GAT AT TG TGT GT

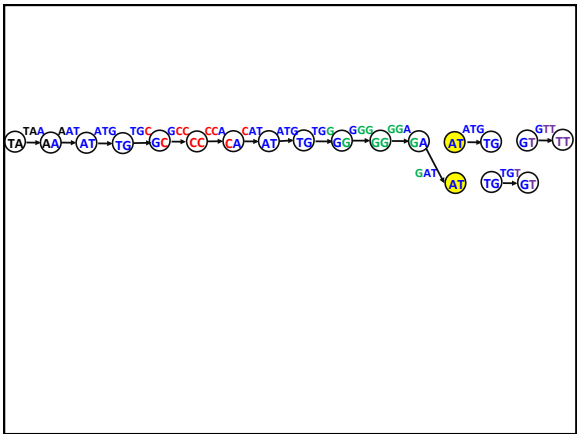

TA TAA AA AAT AT ATG TG TGC GC GCC CC CCA CA CAT AT ATG TG TGG GG GG GGA GA AT ATG TG GT GTT TT
GGG GG GA GAT AT TG TGT GT

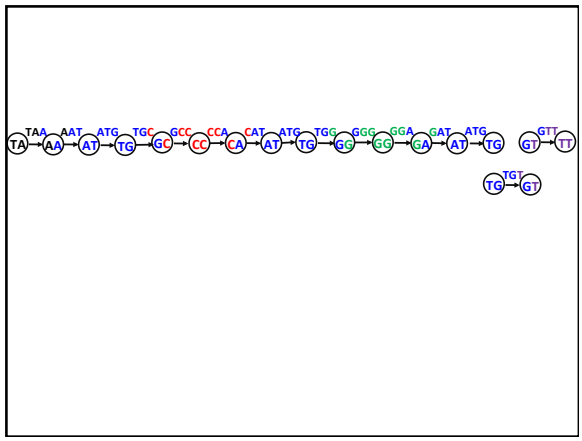

TA TAA AA AAT AT ATG TG TGC GC GCC CC CCA CA CAT AT ATG TG TGG GG GG GGA GA AT ATG TG GT GTT TT
GGG GG GA GAT AT TG TGT GT

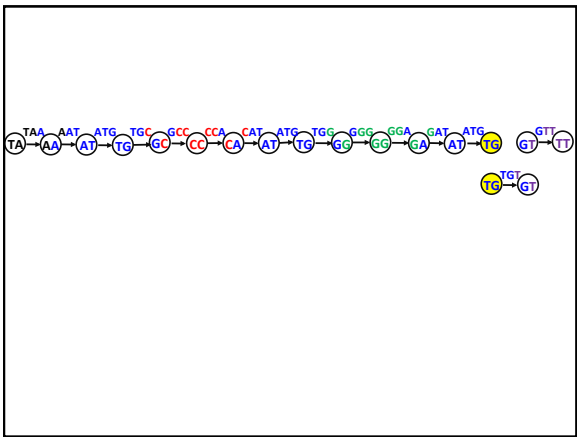

TAA AAT ATG TGC GCC CCA CAT ATG TGG GGG GGA ATG GTT
TA AA AT TG GC CC CA AT TG GG GG GA AT TG GT TT
GAT TGT
GA AT TG GT

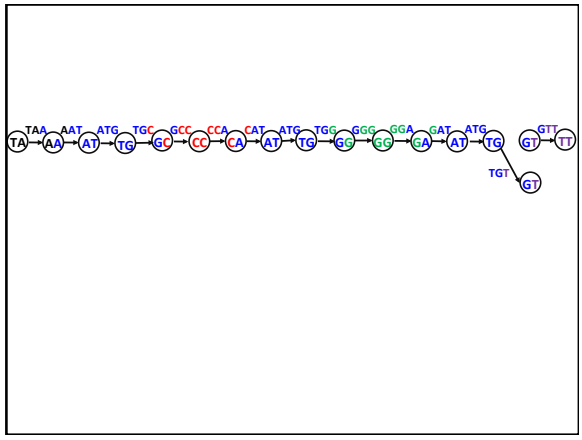

TAA AAT ATG TGC GCC CCA CAT ATG TGG GGG GGA ATG GTT
TA AA AT TG GC CC CA AT TG GG GG GA AT TG GT TT
GAT TGT
GA AT TG GT

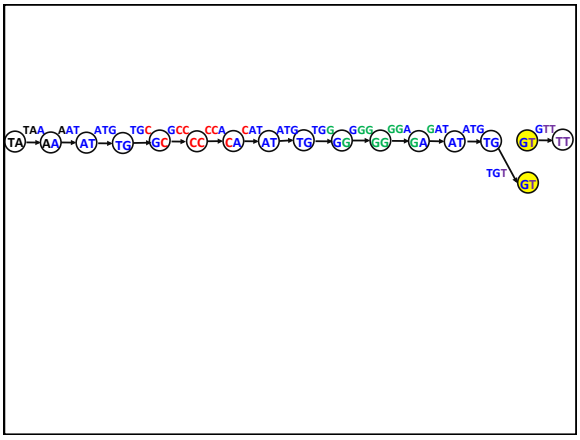

TAA AAT ATG TGC GCC CCA CAT ATG TGG GGG GGA ATG GTT
TA AA AT TG GC CC CA AT TG GG GG GA AT TG GT TT
GAT TGT
AT TG GT


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

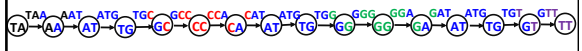

TAA AAT ATG TGC GCC CCA CAT ATG TGG GGG GGA GAT ATG GTT
TA AA AT TG GC CC CA AT TG GG GG GA AT TG GT TT
TGT
TG GT


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

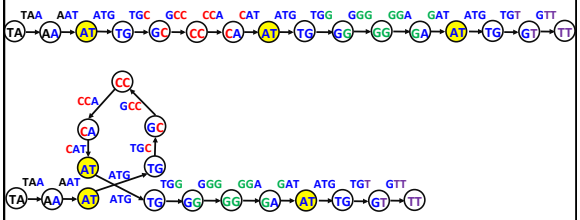




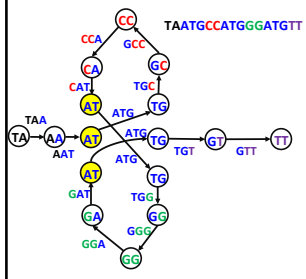
We Are Not Done with Gluing Yet

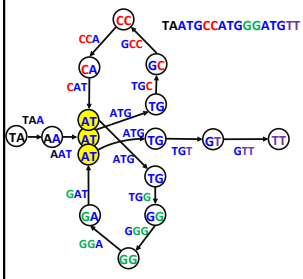


Gluing Identically Labeled Nodes

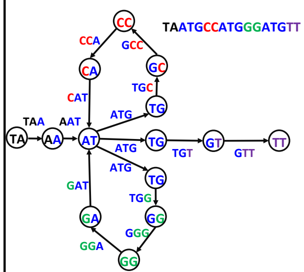


Gluing Identically Labeled Nodes

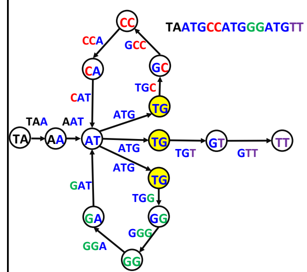




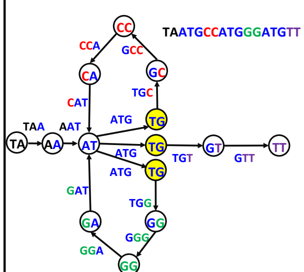
Gluing Identically Labeled Nodes



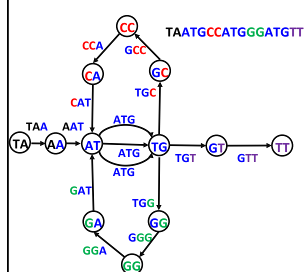
Gluing Identically Labeled Nodes



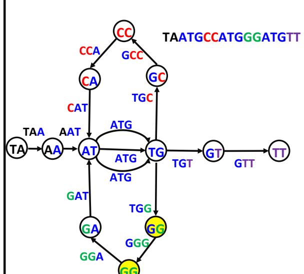
Gluing Identically Labeled Nodes



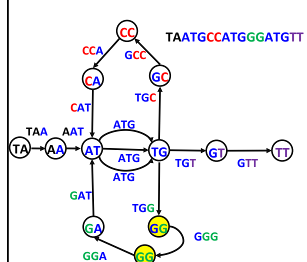
Gluing Identically Labeled Nodes



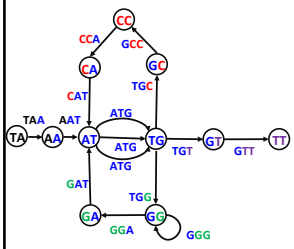
Gluing Identically Labeled Nodes



Gluing Identically Labeled Nodes



The Same de Bruijn Graph:
DeBruin(Genome)=DeBruin(Genome Composition)



Constructing de Bruijn Graph

De Bruijn graph of a collection of *k*-mers:
- Represent every *k*-mer as an edge between its prefix and suffix
- Glue **ALL** nodes with identical labels.

***DeBruijn*(*k-mers*)**
form a node for each (*k-1*)-mer from ***k-mers***
for each *k*-mer in ***k-mers***
connect its prefix node with its suffix node by an edge

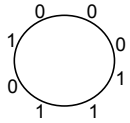
From Hamilton  to Euler  to de Bruijn 

Universal String Problem (De Bruijn, 1946). Find a circular string containing each binary *k*-mer exactly once.

From Hamilton  to Euler  to de Bruijn 

Universal String Problem (De Bruijn, 1946). Find a circular string containing each binary *k*-mer exactly once.

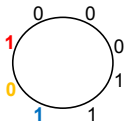
000 001 010 011 100 101 110 111



From Hamilton  to Euler  to de Bruijn 

Universal String Problem (Nicolaas de Bruijn, 1946). Find a circular string containing each binary *k*-mer exactly once.

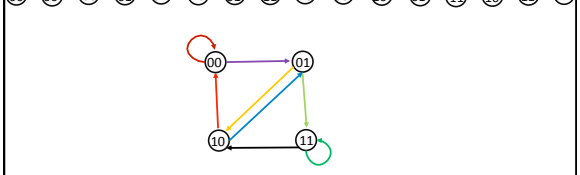
000 001 010 011 100 101 110 111



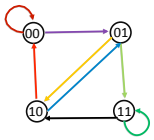
From Hamilton  to Euler  to de Bruijn 

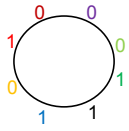
Universal String Problem (Nicolaas de Bruijn, 1946). Find a circular string containing each binary *k*-mer exactly once.

000 001 010 011 100 101 110 111

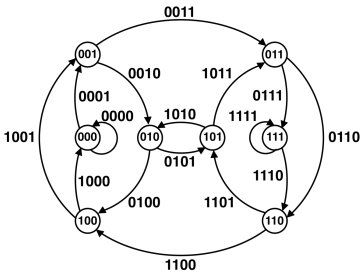


From Hamilton  to Euler  to de Bruijn 





De Bruijn Graph for 4-Universal String



Does it have an Eulerian cycle? If yes, how can we find it?

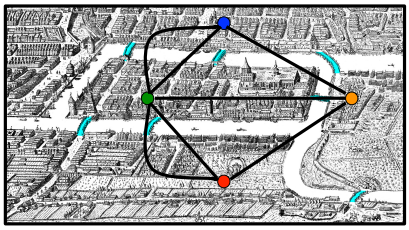
Outline

- What Is Genome Sequencing?
- Exploding Newspapers
- The String Reconstruction Problem
- String Reconstruction as a Hamiltonian Path Problem
- String Reconstruction as an Eulerian Path Problem
- Similar Problems with Different Fates
- De Bruijn Graphs
- **Euler's Theorem**
- Assembling Read-Pairs
- De Bruijn Graphs Face Harsh Realities of Assembly

Eulerian CYCLE Problem

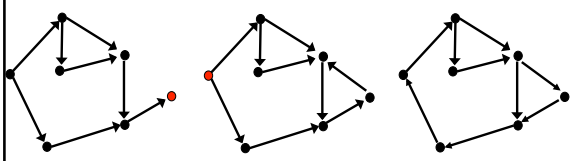
Eulerian CYCLE Problem. Find an Eulerian cycle in a graph.

- **Input.** A graph.
- **Output.** A cycle visiting every edge in the graph exactly once.



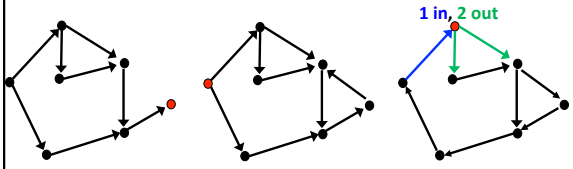
A Graph is **Eulerian** if It Contains an Eulerian Cycle.

Is this graph Eulerian?



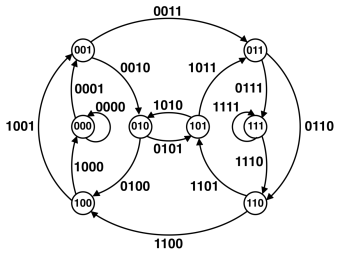
A Graph is **Eulerian** if It Contains an Eulerian Cycle.

Is this graph Eulerian?



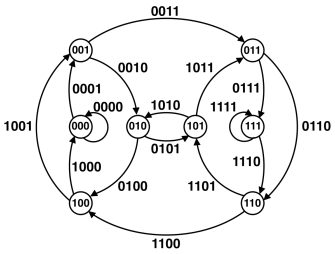
A graph is **balanced** if ***indegree*** = ***outdegree*** for each node

Is the Graph for 4-Universal String Balanced?



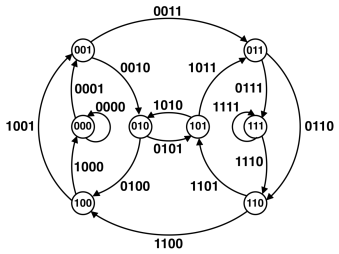
Euler's Theorem

- Every Eulerian graph is balanced



Euler's Theorem

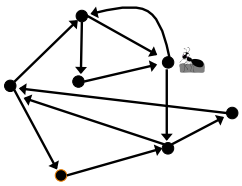
- Every Eulerian graph is balanced
- **Every balanced* graph is Eulerian**



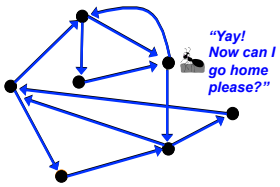
(*) and strongly connected, of course!

Recruiting an Ant to Prove Euler's Theorem

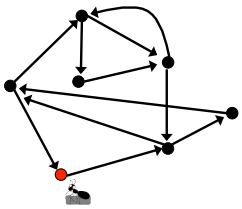
Let an ant randomly walk through the graph.
The ant cannot use the same edge twice!



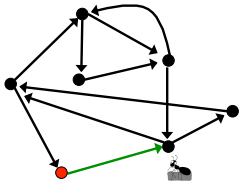
If Ant Was a Genius…



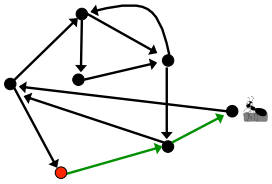
A Less Intelligent Ant Would Randomly Choose a Node and Start Walking…



Walking…

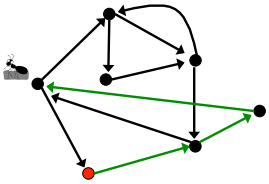


Walking… and Walking…

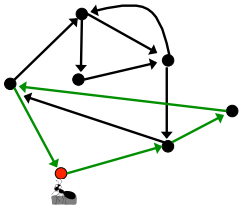


Walking… and Walking… and Walking…

Can it get stuck? **In what node?**

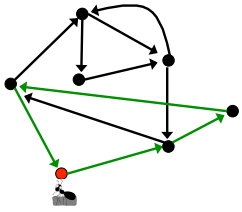


The Ant Can Only Get Stuck at the Starting Node



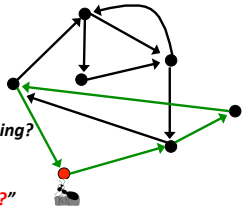
The Ant Has Completed a Cycle BUT has not Proven Euler's theorem yet…

The constructed cycle is not Eulerian. **Can we enlarge it?**



Let's Start at a Different Node in the Green Cycle

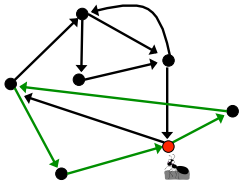
Let's start at a node with still unexplored edges.



"Why should I start at a different node? Backtracking? I'm not evolved to walk backwards! And what difference does it make???"

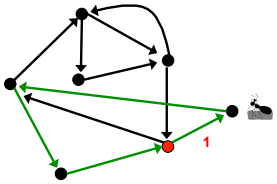
New Instructions for the Ant:

Starting at a node that has an unused edge, traverse the already constructed (green cycle) and return back to the starting node.



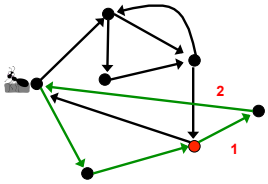
An Ant Traversing Previously Constructed Cycle

Starting at a node that has an unused edge, traverse the already constructed (green cycle) and return back to the starting node.



An Ant Traversing Previously Constructed Cycle

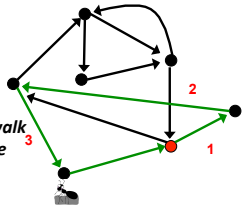
Starting at a node that has an unused edge, traverse the already constructed (green cycle) and return back to the starting node.



An Ant Traversing Previously Constructed Cycle

Starting at a node that has an unused edge, traverse the already constructed (green cycle) and return back to the starting node.

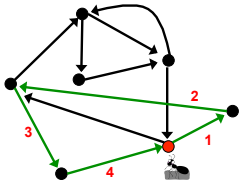
"Why do I have to walk along the same cycle again??? Can I see something new?"



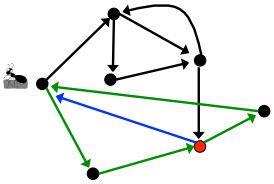
I Returned Back BUT… I Can Continue Walking!

Starting at a node that has an unused edge, traverse the already constructed (green cycle) and return back to the starting node.

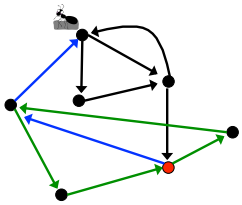
After completing the cycle, start random exploration of still untraversed edges in the graph.



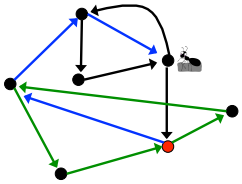
Enlarging the Previously Constructed Cycle



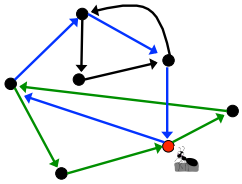
Enlarging the Previously Constructed Cycle



Enlarging the Previously Constructed Cycle



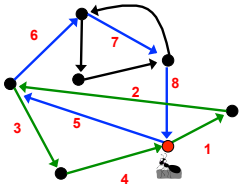
Enlarging the Previously Constructed Cycle



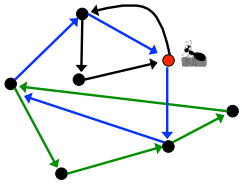
Stuck Again!

No Eulerian cycle yet… can we enlarge the green-blue cycle?

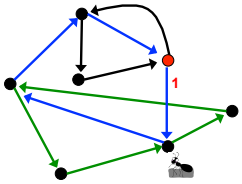
The ant should walk along the constructed cycle starting at yet another node. **Which one?**



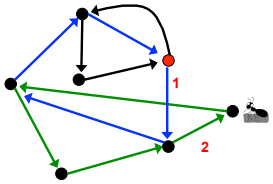
Starting at a New Node, Again…



Traversing the Previously Constructed Green-Blue Cycle

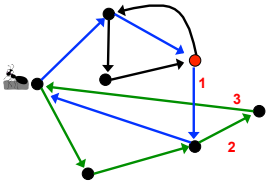


Traversing the Previously Constructed Green-Blue Cycle

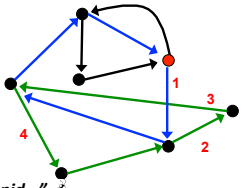


Traversing the Previously Constructed Green-Blue Cycle

"I hate to traverse the same cycle! What difference does it make where I start my walk???

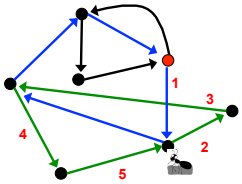


Traversing the Previously Constructed Green-Blue Cycle

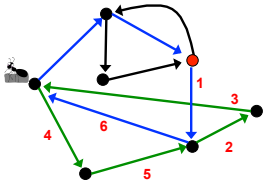


"These instructions are stupid…"

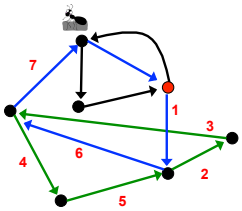
Traversing the Previously Constructed Green-Blue Cycle



Traversing the Previously Constructed Green-Blue Cycle

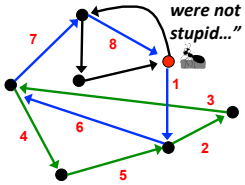


Traversing the Previously Constructed Green-Blue Cycle

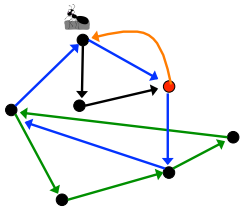


I Returned Back BUT… I Can Continue Walking!

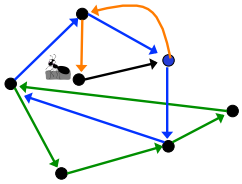
"Hmm, maybe these instructions were not that stupid…"



Enlarging the Green-Blue Cycle

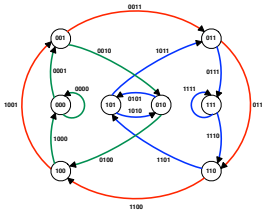


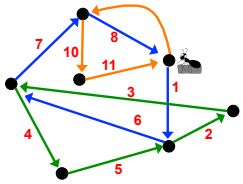
Enlarging the Green-Blue Cycle



I Proved Euler's Theorem! Can I Go Home Please?

EulerianCycle(*BalancedGraph*)
  form a *Cycle* by randomly walking in *BalancedGraph* (avoiding already visited edges)
  while *Cycle* is not Eulerian
    select a node *newStart* in *Cycle* with still unexplored outgoing edges
    form a *Cycle'* by traversing *Cycle* from *newStart* and randomly walking afterwards
    Cycle ← *Cycle'*
  return *Cycle*

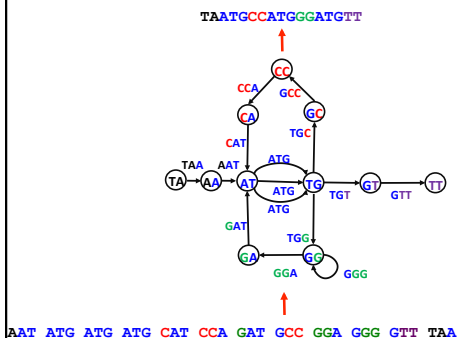




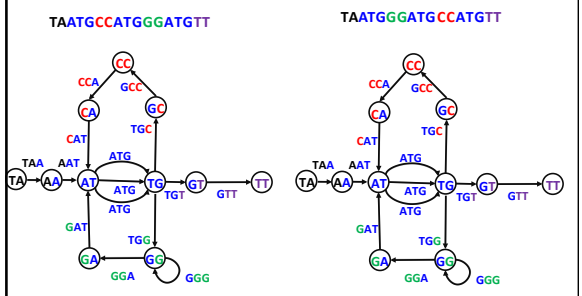
Outline

- What Is Genome Sequencing?
- Exploding Newspapers
- The String Reconstruction Problem
- String Reconstruction as a Hamiltonian Path Problem
- String Reconstruction as an Eulerian Path Problem
- Similar Problems with Different Fates
- De Bruijn Graphs
- Euler's Theorem
- **Assembling Read-Pairs**
- De Bruijn Graphs Face Harsh Realities of Assembly

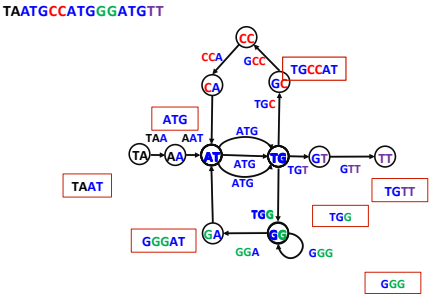
From Reads to de Bruijn Graph to *Genome*



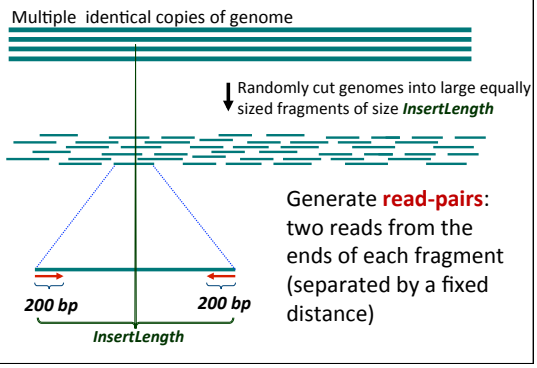
Multiple Eulerian Paths



Breaking Genome into Contigs



DNA Sequencing with Read-pairs

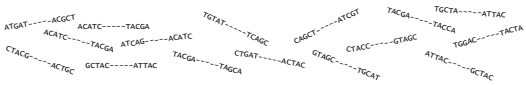


Generate **read-pairs**: two reads from the ends of each fragment (separated by a fixed distance)

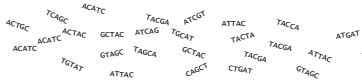
Generating long segments from multiple copies of a genome

Generating read-pairs

Assemble read-pairs using "overlaps" ?



Does assembly of read-pairs improve on assembly of individual reads?



From *k*-mers to Paired *k*-mers

A **paired *k*-mer** is a pair of *k*-mers at a fixed distance *d* apart in *Genome*.
E.g. TCA and TCC are at distance *d*=11 apart.

Disclaimers:
1. In reality, *Read1* and *Read2* are typically sampled from different strands:
(→ ..... ← rather than → ..... →)
2. In reality, the distance *d* between reads is measured with errors.

What is *PairedComposition*(**TAATGCCATGGGATGTT**)?

TAA GCC

paired 3-mer

What is *PairedComposition*(**TAATGCCATGGGATGTT**)?

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

Representing a **paired 3-mer** **TAA GCC** as a 2-line expression: **TAA / GCC**

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

PairedComposition(**TAATGCCATGGGATGTT**)

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

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

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

Representing *PairedComposition* in lexicographic order

String Reconstruction from Read-Pairs Problem

String Reconstruction from Read-Pairs Problem. Reconstruct a string from its paired *k*-mers.

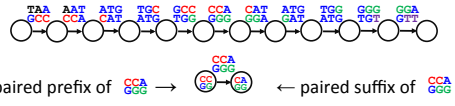
- **Input.** A collection of paired *k*-mers.
- **Output.** A string *Text* such that *PairedComposition*(*Text*) is equal to the collection of paired *k*-mers.

How Would de Bruijn Assemble Paired *k*-mers?

Paired de Bruijn Graphs

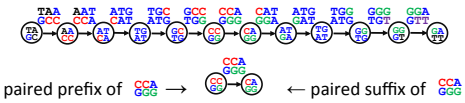
Representing Genome **TAATGCCATGGGATGTT** as a Path

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

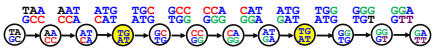


paired prefix of CCA / GGG → ← paired suffix of CCA / GGG

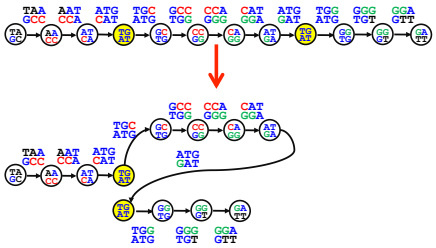
Labeling Nodes by Paired Prefixes and Suffixes



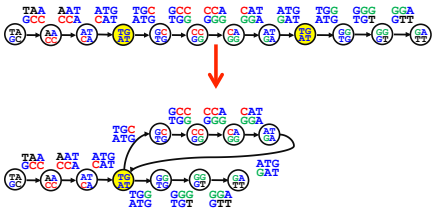
Glue nodes with identical labels



Glue nodes with identical labels



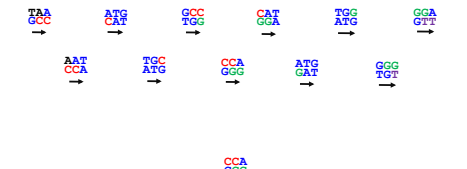
Glue nodes with identical labels



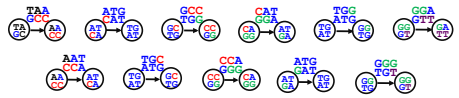
Paired de Bruijn Graph from the Genome

Constructing Paired de Bruijn Graph from paired k-mers

Constructing Paired de Bruijn Graph from paired k-mers

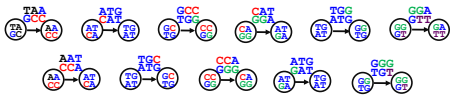


Constructing Paired de Bruijn Graph



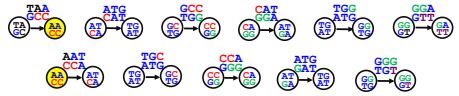
paired prefix of CCA GGG →  ← paired suffix of CCA GGG

Constructing Paired de Bruijn Graph

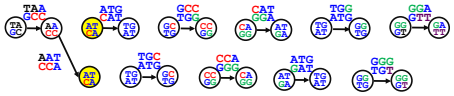


- **Paired de Bruijn graph for a collection of paired *k*-mers:**
  - Represent every paired *k*-mer as an edge between its paired prefix and paired suffix.
  - Glue **ALL** nodes with identical labels.

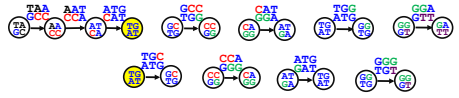
Constructing Paired de Bruijn Graph



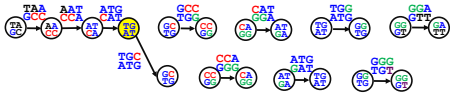
Constructing Paired de Bruijn Graph



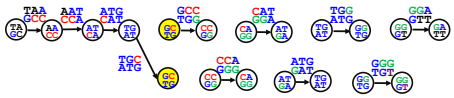
Constructing Paired de Bruijn Graph



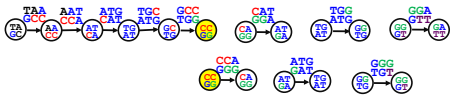
Constructing Paired de Bruijn Graph



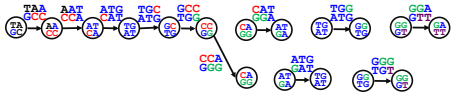
Constructing Paired de Bruijn Graph



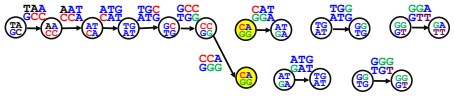
Constructing Paired de Bruijn Graph



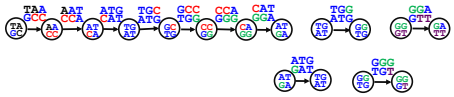
Constructing Paired de Bruijn Graph



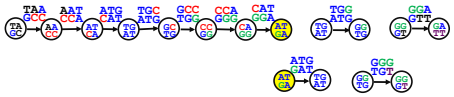
Constructing Paired de Bruijn Graph



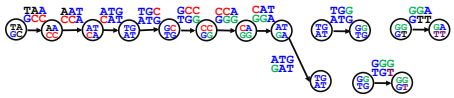
Constructing Paired de Bruijn Graph



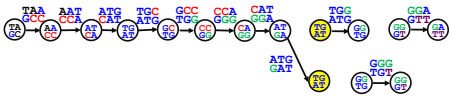
Constructing Paired de Bruijn Graph



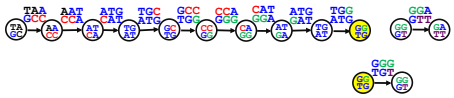
Constructing Paired de Bruijn Graph



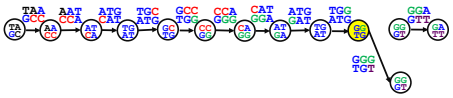
Constructing Paired de Bruijn Graph



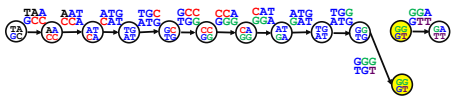
Constructing Paired de Bruijn Graph



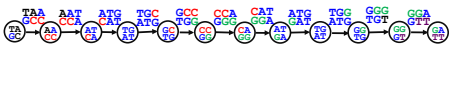
Constructing Paired de Bruijn Graph



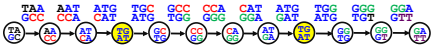
Constructing Paired de Bruijn Graph



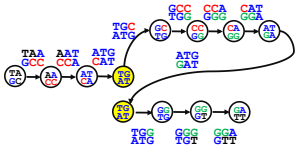
Constructing Paired de Bruijn Graph



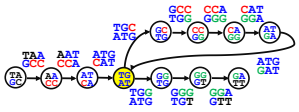
We Are Not Done with Gluing Yet



Constructing Paired de Bruijn Graph



Constructing Paired de Bruijn Graph



Paired de Bruijn Graph from read-pairs

Paired de Bruijn Graphs

- **Paired de Bruijn graph for a collection of paired *k*-mers:**
  - Represent every paired *k*-mer as an edge between its paired prefix and paired suffix.
  - Glue **ALL** nodes with identical labels.

Which Graph Represents a Better Assembly?

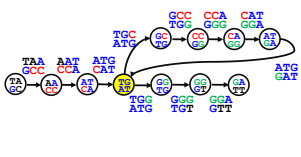
Unique genome reconstruction

TAATGCCATGGGATGTT

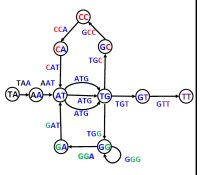
Multiple genome reconstructions

TAATGCCATGGGATGTT

TAATGGGATGCCATGTT



Paired de Bruijn Graph



De Bruijn Graph

Outline

- What Is Genome Sequencing?
- Exploding Newspapers
- The String Reconstruction Problem
- String Reconstruction as a Hamiltonian Path Problem
- String Reconstruction as an Eulerian Path Problem
- Similar Problems with Different Fates
- De Bruijn Graphs
- Euler's Theorem
- Assembling Read-Pairs
- **De Bruijn Graphs Face Harsh Realities of Assembly**

Some Ridiculously Unrealistic Assumptions

- Perfect coverage of genome by reads (every *k*-mer from the genome is represented by a read)
- Reads are error-free.
- Multiplicities of *k*-mers are known
- Distances between reads within read-pairs are exact.

Some Ridiculously Unrealistic Assumptions

- **Imperfect** coverage of genome by reads (every *k*-mer from the genome is represented by a read)
- Reads are **error-prone**.
- Multiplicities of *k*-mers are **unknown.**
- Distances between reads within read-pairs are **inexact.**
- **Etc., etc., etc.**

1st Unrealistic Assumption: Perfect Coverage

```
atgccgtatggacaacgact
atgccgtatg
  gccgtatgga
    gtatggacaa
       gacaacgact
```

250-nucleotide reads generated by Illumina technology capture only a small fraction of 250-mers from the genome, thus violating the key assumption of the de Bruijn graphs.

Breaking Reads into Shorter *k*-mers

```
atgccgtatggacaacgact
atgccgtatg
  gccgtatgga
    gtatggacaa
       gacaacgact
```

```
atgccgtatggacaacgact
atgcc
 tgccg
  gccgt
   ccgta
    cgtat
     gtatg
      tatgg
       atgga
        tggac
         ggaca
          gacaa
           acaac
            caacg
             aacga
              acgac
               cgact
```

2nd Unrealistic Assumption: Error-free Reads

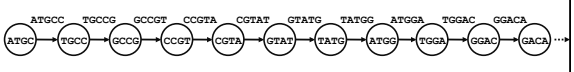
```
atgccgtatggacaacgact
atgccgtatg
  gccgtatgga
    gtatggacaa
       gacaacgact
   cgtaCggaca
```

Erroneous read (change of **t** into C)

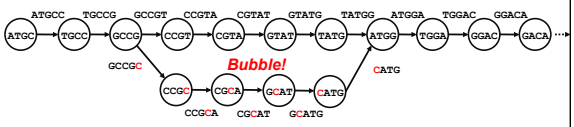
```
atgccgtatggacaacgact
atgcc
 tgccg
  gccgt
   ccgta
    cgtat
     gtatg
      tatgg
       atgga
        tggac
         ggaca
          gacaa
           acaac
            caacg
             aacga
              acgac
               cgact
cgtaC
 gtaCg
  taCgg
   aCgga
    Cggac
```

De Bruijn Graph of ATGGCGTGCAATG… Constructed from Error-Free Reads

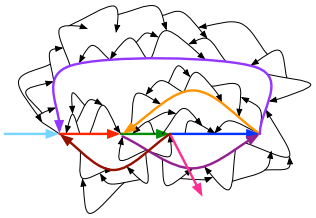
.



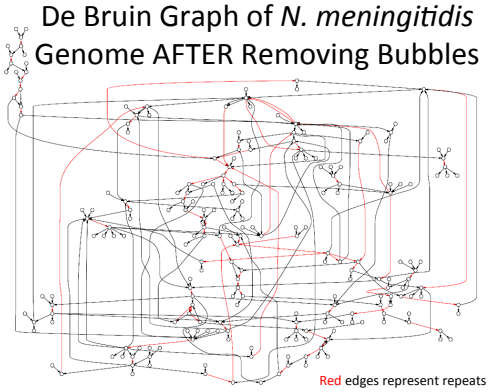
Errors in Reads Lead to **Bubbles** in the De Bruijn Graph



Bubble Explosion…Where Are the Correct Edges of the de Bruijn Graph?



De Bruin Graph of *N. meningitidis* Genome AFTER Removing Bubbles



Red edges represent repeats

Who Are These People?

10 scientists and entrepreneurs who made their genomes available in 2009

- Personal genome sequencing will soon expand to millions of individuals.

Who Are These People?

10 scientists and entrepreneurs who made their genomes available in 2009

- Personal genome sequencing will soon expand to millions of individuals.
- We will soon sequence genomes of tens of thousands of species.

Who Are These People?

10 scientists and entrepreneurs who made their genomes available in 2009

- Personal genome sequencing will soon expand to millions of individuals.
- We will soon sequence genomes of tens of thousands of species.

- Thousands of cancer genomes have already been sequenced, and genome sequencing will soon become a routine technique in medicine.