de Bruijn graphs and DNA fragment assembly

Michael Duff
Dept. of Genetics & Genome Sciences
Graveley Lab, UConn Health

Leonhard Euler
1707-1783

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

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.

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

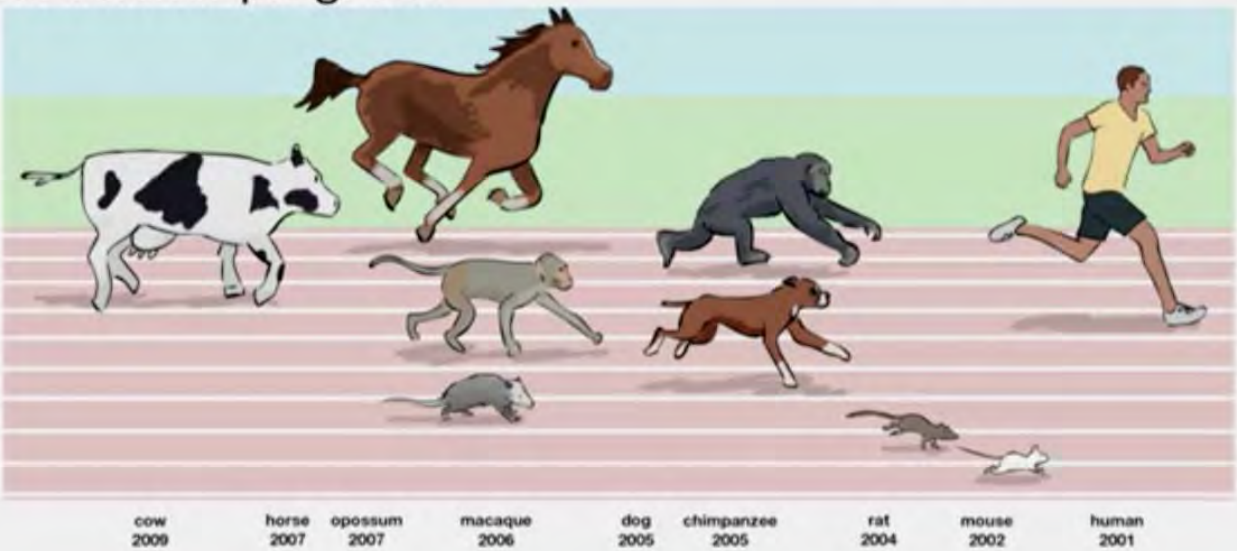
- **1997**: Craig Venter founds Celera Genomics, a private firm, with the same goal.

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.

CTGATGATGGACTACGCTACTACTGCTAGCTGTATTACGA
TCAGCTACCACATCGTAGCTACGATGCATTAGCAAGCTAT
CGATCGATCGATCGATTATCTACGATCGATCGATCGATCA
CTATACGAGCTACTACGTACGTACGATCGCGGGACTATTA
TCGACTACAGATAAAACATGCTAGTACAACAGTATACATA
GCTGCGGGATACGATTAGCTAATAGCTGACGATATCCGAT

CTGATGATGGACTACGCTACTACTGCTAGCTGTATTACGA
TCAGCTACAACATCGTAGCTACGATGCATTAGCAAGCTAT
CGATCGATCGATCGATTATCTACGATCGATCGATCGATCA
CTATACGAGCTACTACGTACGTACGATCGCGTGACTATTA
TCGACTACAGATGAAACATGCTAGTACAACAGTATACATA
GCTGCGGGATACGATTAGCTAATAGCTGACGATATCCGAT

Why Do We Sequence Personal Genomes?

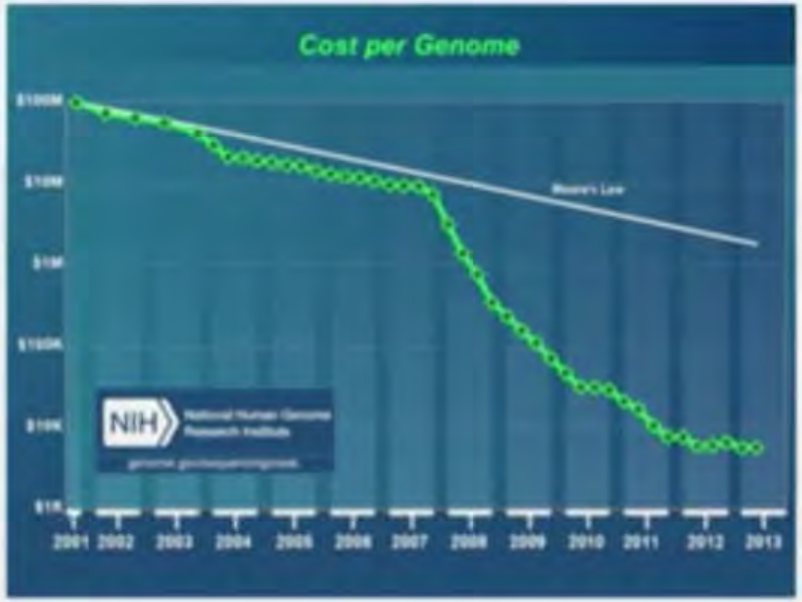
- **2010:** Nicholas Volker became first human being to be saved by genome sequencing.

  - 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.



HUMAN LONGEVITY, INC.

ABOUT
mission & people

SCIENCE & TECHNOLOGY
advanced products

PARTNERS
strategic collaborations

MEDIA
resources & news

CAREERS
we're hiring

Aging is the single biggest risk factor for virtually every significant human disease…

…our goal is to extend and enhance the healthy, high-performance lifespan and change the face of aging. For the first time, the power of human genomics, informatics, next generation DNA sequencing technologies, and stem cell advances are being harnessed in one company, Human Longevity Inc., with the leading pioneers in these fields. Our goal is to solve the diseases of aging by changing the way medicine is practiced.

It's not just a long life we're striving for, but one which is worth living.

Human Genomics

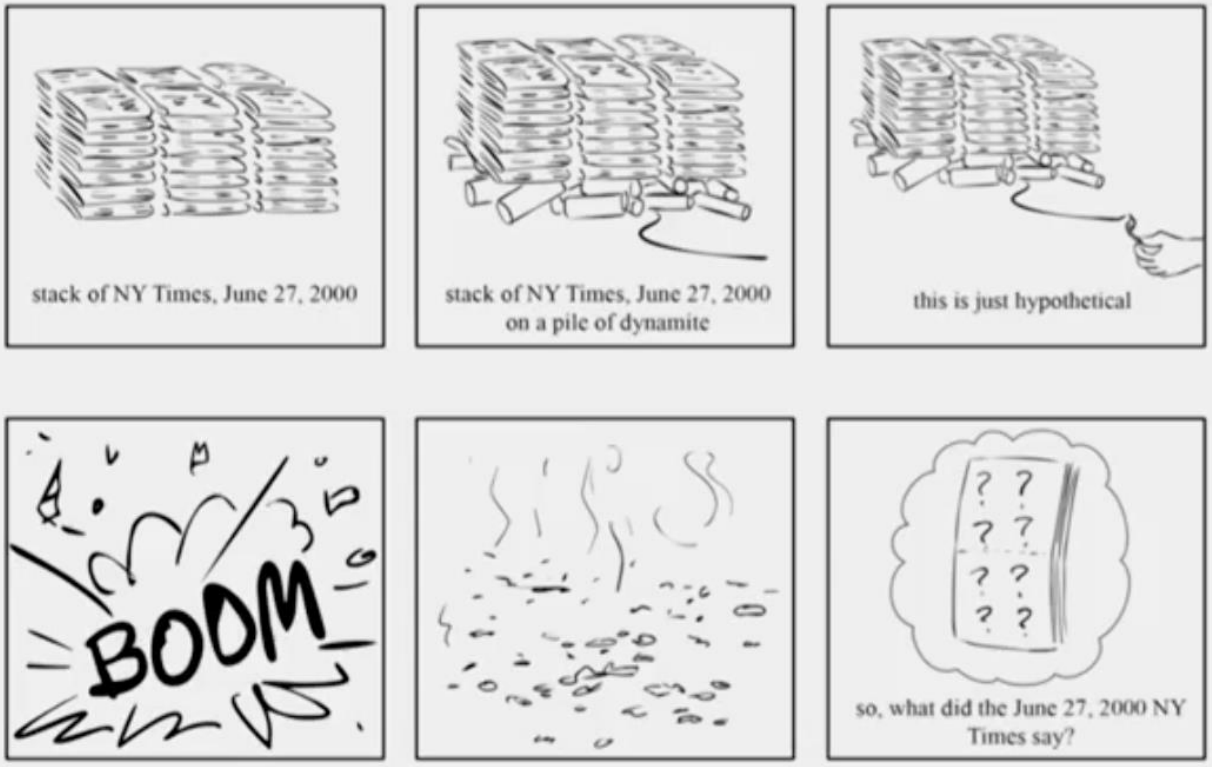
HLI is building the world's largest human genome sequencing center in the world. Along with computing advances, DNA sequencing has seen an explosion of next generation technologies that are enabling faster and better sequencing of human genomes.

HLI has initially purchased two Illumina HiSeq X Ten Sequencing Systems (with the option for an additional three systems). These next generation sequencing machines are clusters of 10 instruments that provide HLI with an annual throughput of tens of thousands of human genomes. HLI plans to sequence up to 40,000 human genomes per year, with plans to rapidly scale to 100,000 human genomes.

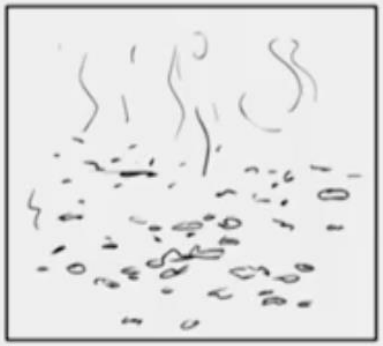
Outline

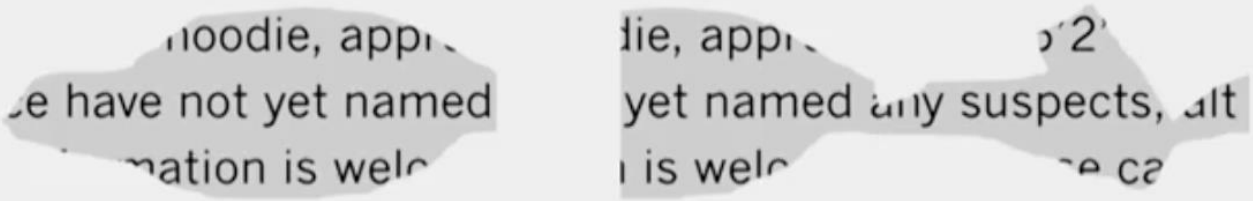
- What Is Genome Sequencing?
- **Exploding Newspapers**
- The String Reconstruction Problem
- String Reconstruction as a Hamiltonian Path Problem
- String Reconstruction as an Eulerian Path Problem
- Similar Problems with Different Fates
- De Bruijn Graphs
- Euler's Theorem
- Assembling Read-Pairs
-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

The Newspaper Problem

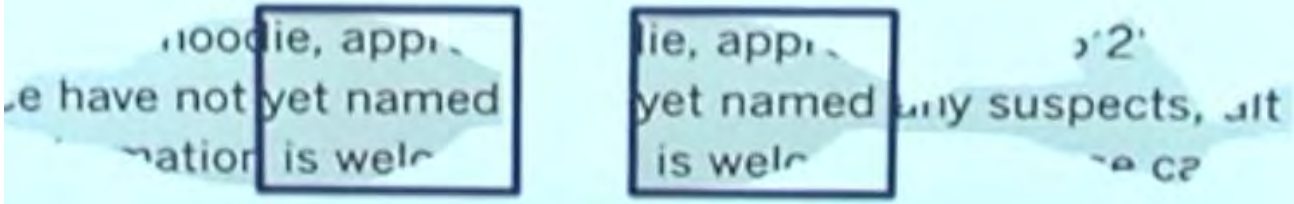


The Newspaper Problem as an Overlapping Puzzle

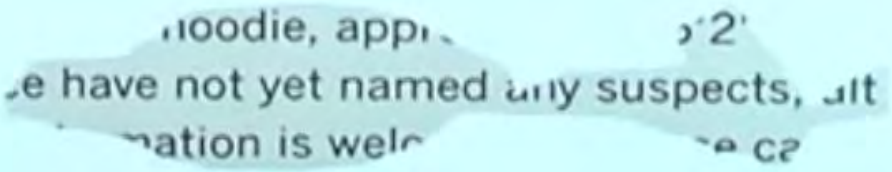




The Newspaper Problem as an Overlapping Puzzle



The Newspaper Problem as an Overlapping Puzzle



The Newspaper Problem as an Overlapping Puzzle


The New York Times

Genetic Code of Human Life Is Cracked by Scientists

JUSTICES REAFFIRM MIRANDA RULE, 7-2; A PART OF 'CULTURE'

A Pearl and a Hindrance in Human DNA

Clinton Rues Extension of Surplus And the Stakes in How to Use It

Sharansky in Eyes of Israelis: A Hero or Betrayer of Peace?

Buildings' Sector Now a Troubled Landlord

Multiple Copies of a Genome (Millions of them)

stack of NY Times, June 27, 2000

CTGATGATGGACTACGCTACTACTGCTAGCTGTATTACGATCAGCTACCACATCGTAGCTACGATGCATTAGCAAGCTATCGGATCAGCTACCACATCGTAGC

CTGATGATGGACTACGCTACTACTGCTAGCTGTATTACGATCAGCTACCACATCGTAGCTACGATGCATTAGCAAGCTATCGGATCAGCTACCACATCGTAGC

CTGATGATGGACTACGCTACTACTGCTAGCTGTATTACGATCAGCTACCACATCGTAGCTACGATGCATTAGCAAGCTATCGGATCAGCTACCACATCGTAGC

CTGATGATGGACTACGCTACTACTGCTAGCTGTATTACGATCAGCTACCACATCGTAGCTACGATGCATTAGCAAGCTATCGGATCAGCTACCACATCGTAGC

Breaking the Genomes at Random Positions

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

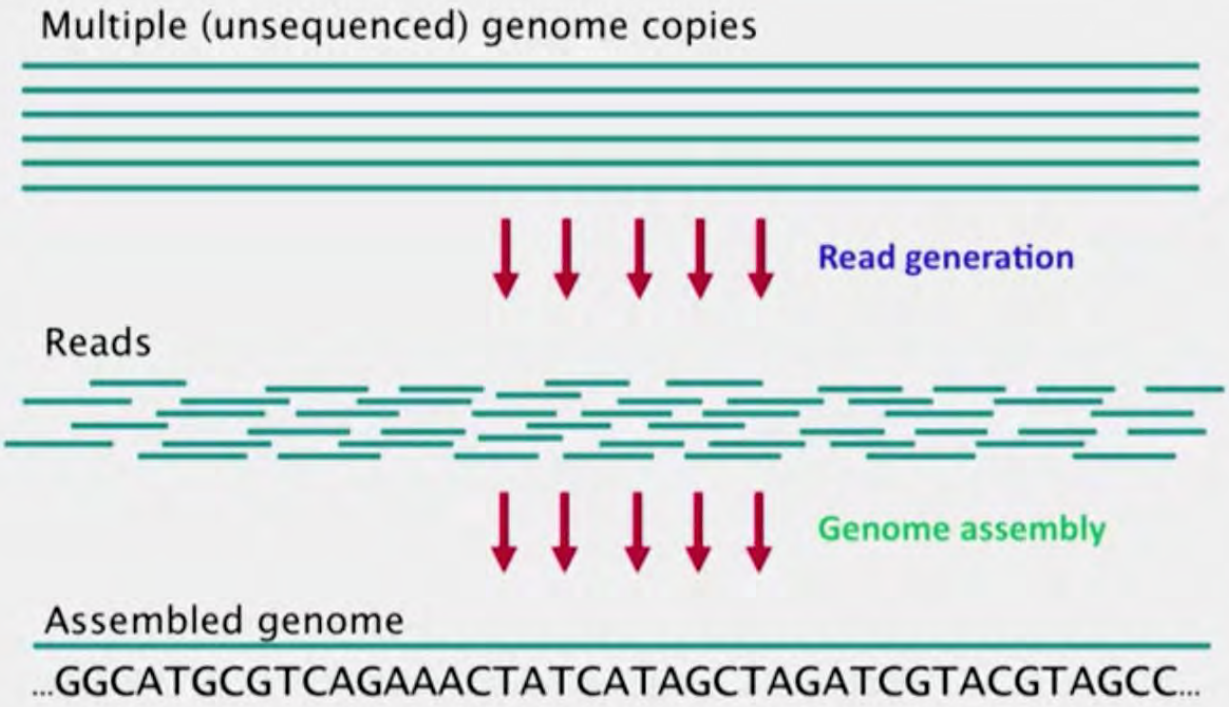
CTGATGATGGACT ACGCTACTACT GCTAGCTGTAT TACGATCAGC TACCACATCGT AGCTACGATGCA TTAGCAAGCT ATCGGATCA GCTACCACATC GTAGC

No Idea What Position Every Read Comes From

No Idea What Position Every Read Comes From

No Idea What Position Every Read Comes From

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**!

	 MiSeq	 NextSeq 500	
Run Mode	N/A	Mid-Output	High-Output
Flow Cells per Run	1	1	1
Output Range	0.3-15 Gb	20-39 Gb	30-120 Gb
Run Time	5-55 hours	15-26 hours	12-30 hours
Reads per Flow Cell†	25 million‡	130 million	400 million
Maximum Read Length	2 x 300 bp	2 x 150 bp	2 x 150 bp
System Overview	Speed and simplicity for targeted and small genome sequencing.	Speed and simplicity for everyday genomics.	

	 HiSeq 2500		 HiSeq 3000	 HiSeq 4000	 HiSeq X Five*	 HiSeq X Ten*
Run Mode	Rapid Run	High-Output	N/A	N/A	N/A	N/A
Flow Cells per Run	1 or 2	1 or 2	1	1 or 2	1 or 2	1 or 2
Output Range	10-300 Gb	50-1000 Gb	125-750 Gb	125-1500 Gb	900-1800 Gb	900-1800 Gb
Run Time	7-60 hours	<1-6 days	<1-3.5 days	<1-3.5 days	<3 days	<3 days
Reads per Flow Cell†	300 million	2 billion	2.5 billion	2.5 billion	3 billion	3 billion
Maximum Read Length	2 x 250 bp	2 x 125 bp	2 x 150 bp	2 x 150 bp	2 x 150 bp	2 x 150 bp
System Overview	Power and efficiency for large-scale genomics.		Maximum throughput and lowest cost for production-scale genomics.	Maximum throughput and lowest cost for production-scale genomics.	Maximum throughput for production-scale human whole-genome sequencing.	Maximum throughput and lowest cost population-scale human whole-genome sequencing.

3. Competition Outline

The purpose of this Competition is to encourage the development of privately funded and commercially viable technologies for sequencing whole human genomes in a manner described in these Guidelines. A $10 million prize purse will be awarded to the first Team(s) to build a Device or develop a Method (see Section 11 for definition) and then utilize that Device or Method to satisfy these Guidelines within the Competition Period. In the event that more than one Team satisfies these Guidelines, up to three separate prize purses may be awarded (see Sections 3.2 and 3.3).

3.1. **A $10 million USD Grand Prize will be awarded to the first Team (or split by the first Teams) that achieve ALL Best-In-Class Requirements:** (see Table 1 below) submit 100 human genome sequences in 30 days or less at a maximum cost of $1,000 USD per genome sequence, attain an accuracy score of no more than one error per 1,000,000 bases, present each genome as 98% complete, and provide accurate haplotype phasing as defined in these Guidelines. NOTE: In the scenario where at least one team achieves all Best-In-Class requirements, no Category Prizes shall be awarded. The score for each of the 100 genome sequences submitted by a Team to the Judging Panel will be impacted by the following (see section 8 for Scoring details):

- All insertions and deletions
- All rearrangements
- All copy number polymorphisms
- All sequences that are private to an individual genome (i.e., not part of any known reference genome)

A LIMIT THEOREM FOR RANDOM COVERINGS OF A CIRCLE

BY

LEOPOLD FLATTO

ABSTRACT

Let $N_{\alpha,m}$ equal the number of randomly placed arcs of length α $(0<\alpha<1)$ required to cover a circle C of unit circumference m times. We prove that $\lim_{\alpha\to 0} P(N_{\alpha,m} \leqq (1/\alpha)(\log(1/\alpha) + m\log\log(1/\alpha) + x) = \exp((-1/(m-1)!)\exp(-x))$. Using this result for $m=1$, we obtain another derivation of Steutel's result $E(N_{\alpha,1}) = (1/\alpha)(\log(1/\alpha) + \log\log(1/\alpha) + \gamma + o(1))$ as $\alpha\to 0$, γ denoting Euler's constant.

1.

Let C be a circle of unit circumference. Suppose that arcs of given length α $(0<\alpha<1)$ are thrown independently and uniformly on C. The distribution function of the number N_α of these randomly placed arcs needed to cover the circle C has been calculated by Stevens [9] who has shown that

$$P(N_\alpha \leqq n) = \sum_{0\leqq k\leqq 1/\alpha} (-1)^k \binom{n}{k} (1-k\alpha)^{n-1} \tag{1.1}$$

for any positive integer n.

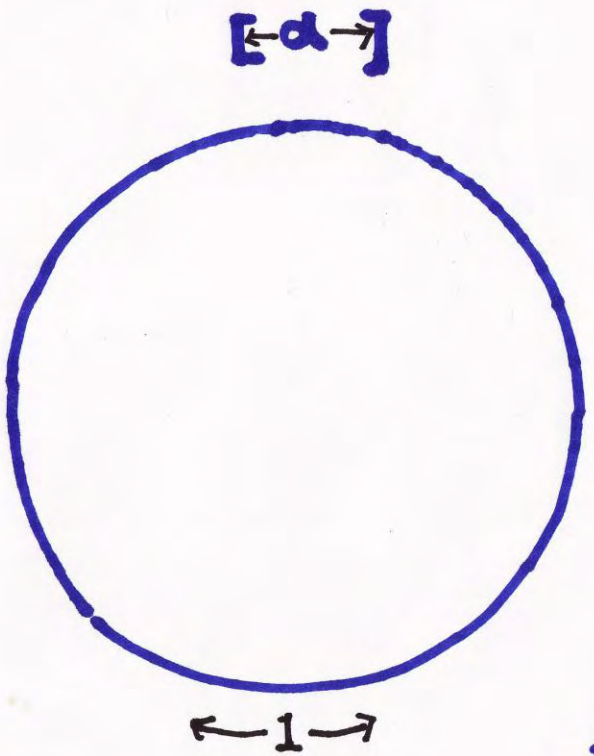
Using (1.1), one may readily compute the expectation $E(N_\alpha)$ as

$$E(N_\alpha) = 1 - \sum_{1\leqq k\leqq 1/\alpha} (-1)^k \frac{(1-k\alpha)^{k-1}}{(k\alpha)^{k+1}} \tag{1.2}$$

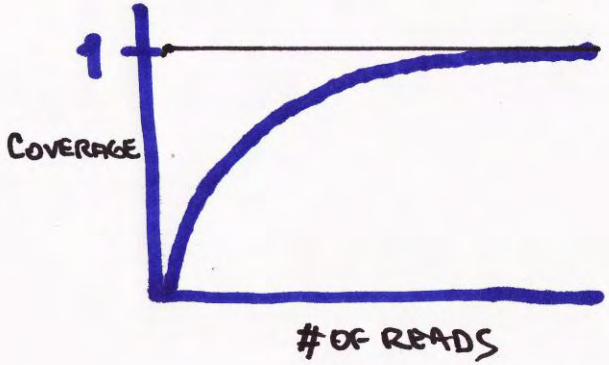
(a derivation of (1.2) is given in [5]).

Unfortunately, neither (1.1) nor (1.2) is very illuminating, since the summands undergo violent oscillations; therefore, it becomes of interest to study the asymptotic behavior of $P(N_\alpha \leqq n)$, $E(N_\alpha)$ as $\alpha\to 0$. Using (1.2), Flatto and Konheim [5] have shown that

Received October 8, 1972



$$\frac{1}{\alpha}\left[\log\frac{1}{\alpha} + \log\log\frac{1}{\alpha} + \gamma\right]$$



DEPTH 5

$$\Pr\{\text{correct}\} = \left\{\binom{5}{0}\rho^5(1-\rho)^0 + \binom{5}{1}\rho^4(1-\rho) + \binom{5}{2}\rho^3(1-\rho)^2\right\} * 1$$

$$+ \binom{5}{3}\rho^2(1-\rho)^3 \begin{cases} \xrightarrow{1/9} \text{ALL SAME} & * 0 \\ \xrightarrow{2/9} \text{ALL DIFFERENT} & * 1 \\ \xrightarrow{2/3} 2 \& 1 & * \frac{1}{2} \end{cases} \Rightarrow * \frac{5}{9}$$

$$+ \left\{\binom{5}{4}\rho^1(1-\rho)^4 + \binom{5}{5}\rho^0(1-\rho)^5\right\} * 0$$

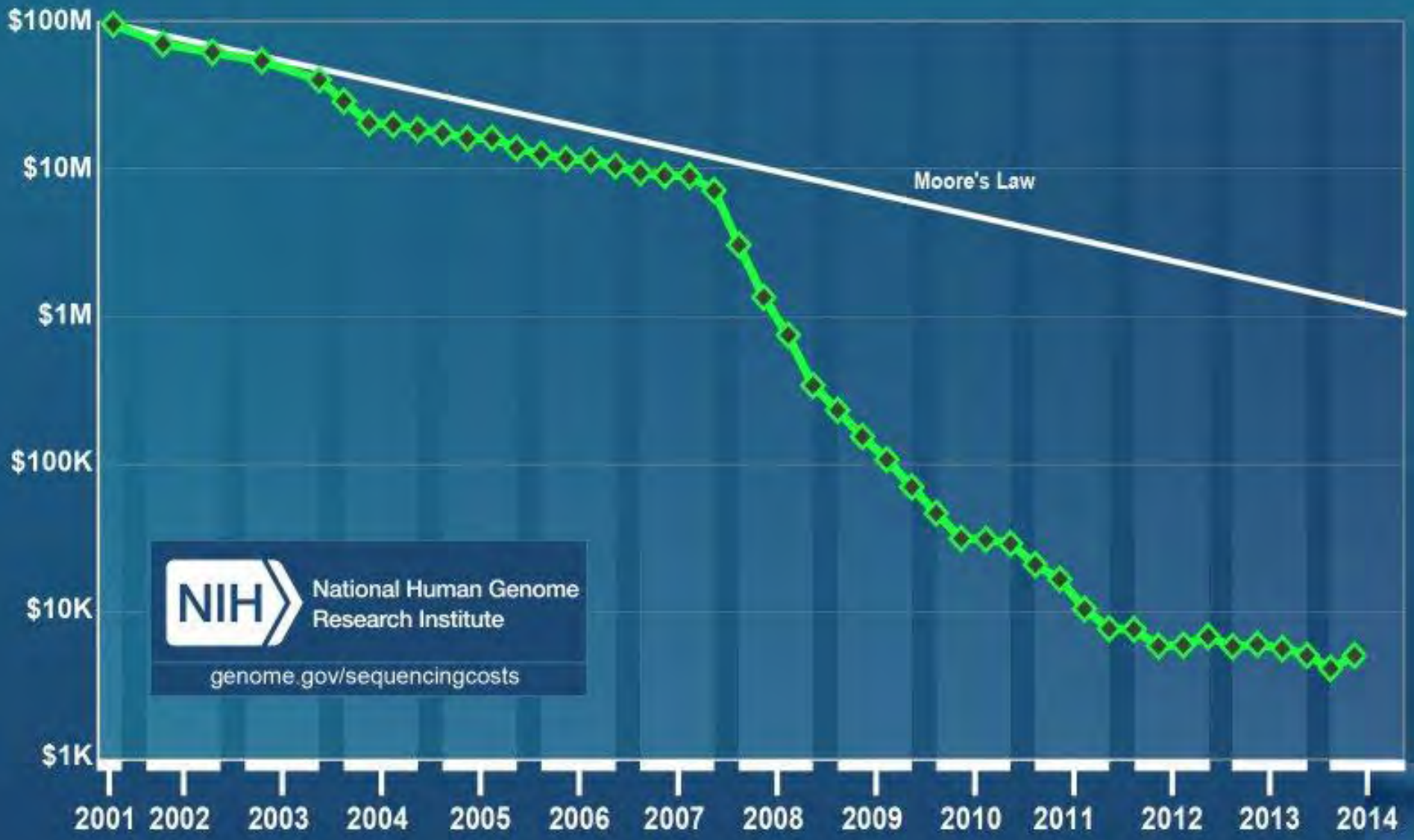
$$.\overline{44}\,\rho^5 + 1.\overline{66}\,\rho^4 - 6.\overline{66}\,\rho^3 + 5.\overline{55}\,\rho^2 - .99999 = 0$$

$$\boxed{\rho = .986046} \quad 1 \text{ IN } 76$$

DEPTH 6 :

$$\boxed{\rho = .98} \quad 1 \text{ IN } 50$$

Cost per Genome



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*.

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

TAA
AAT
ATG
TGT

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 TGC TGG

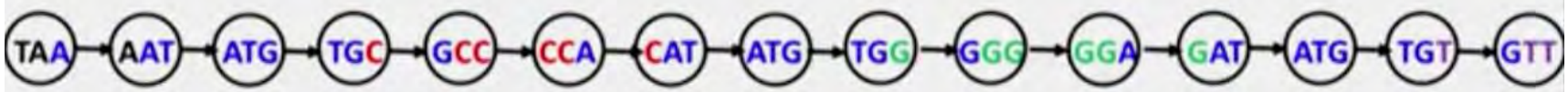
TAA
AAT
ATG
TGT
GTT

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) =



Representing a Genome as a Path

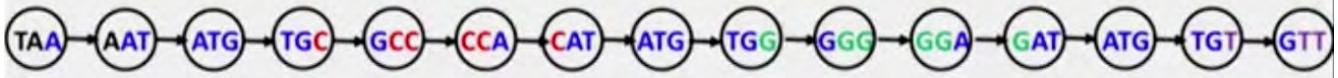
$Composition_3$(TAATGCCATGGGATGTT) =

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

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

Representing a Genome as a Path

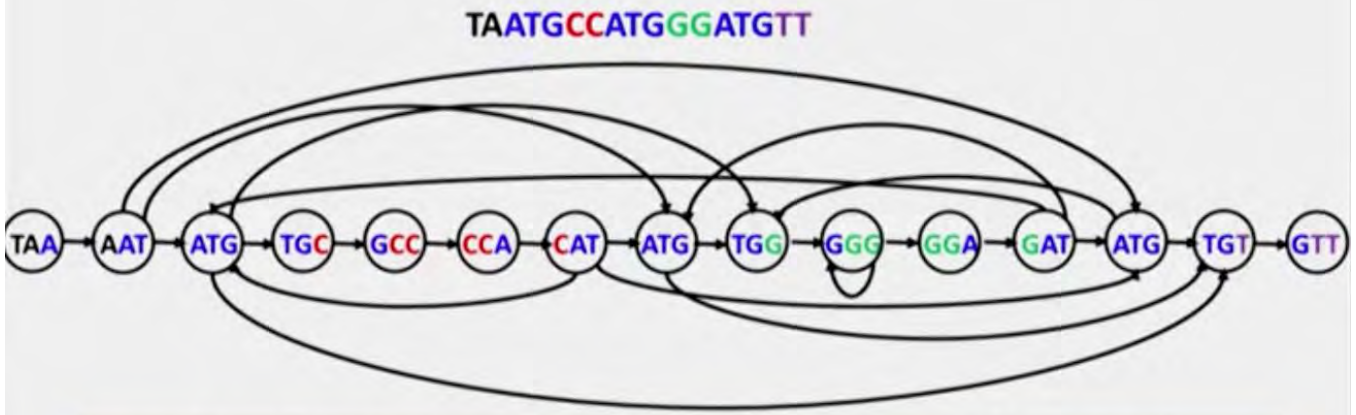
$Composition_3$(TAATGCCATGGGATGTT) =



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**

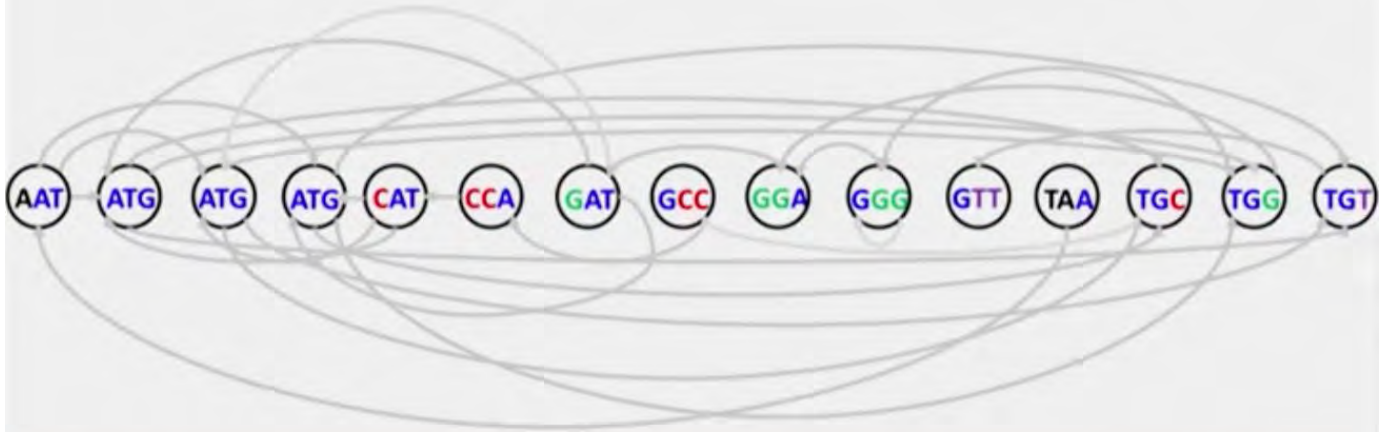
A Path Turns into a Graph



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

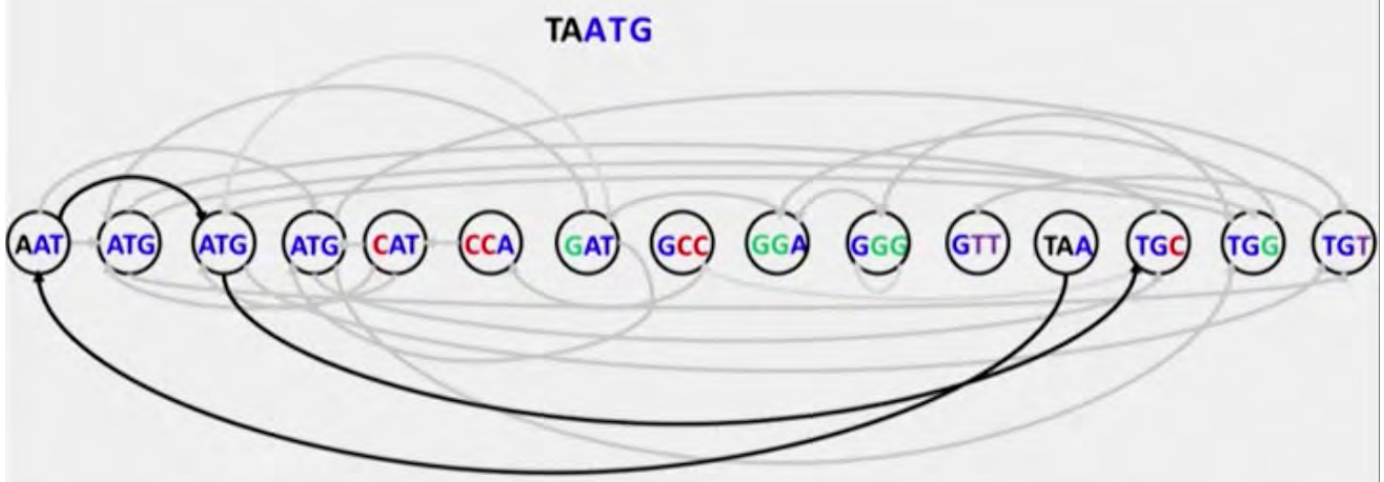
E.g. **TAA** → **AAT**

Where Is the Genomic Path?



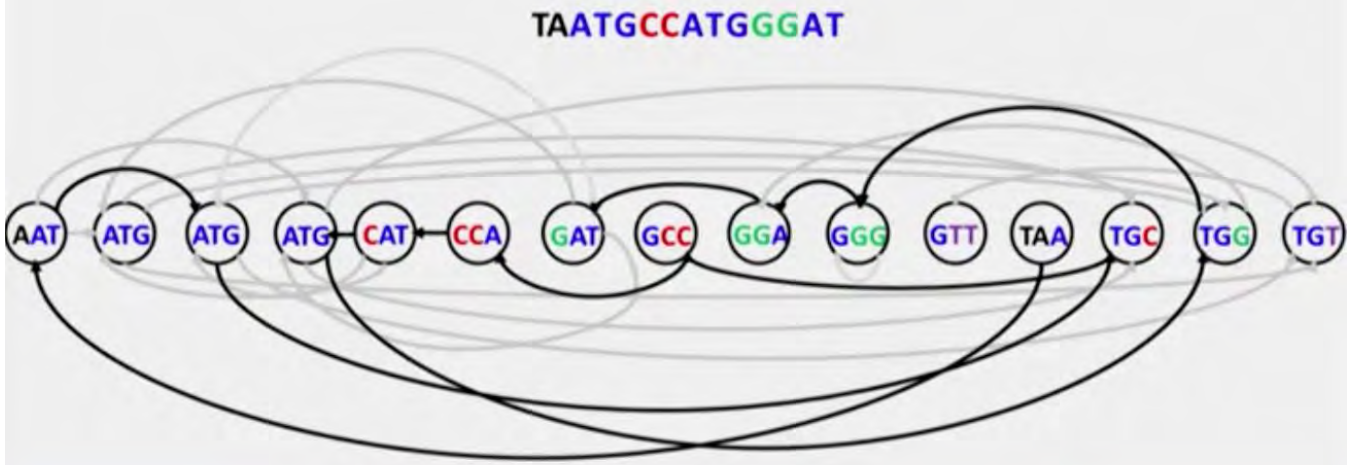
Nodes are arranged from left to right in lexicographic order.

Where Is the Genomic Path?



Nodes are arranged from left to right in lexicographic order.

Where Is the Genomic Path?

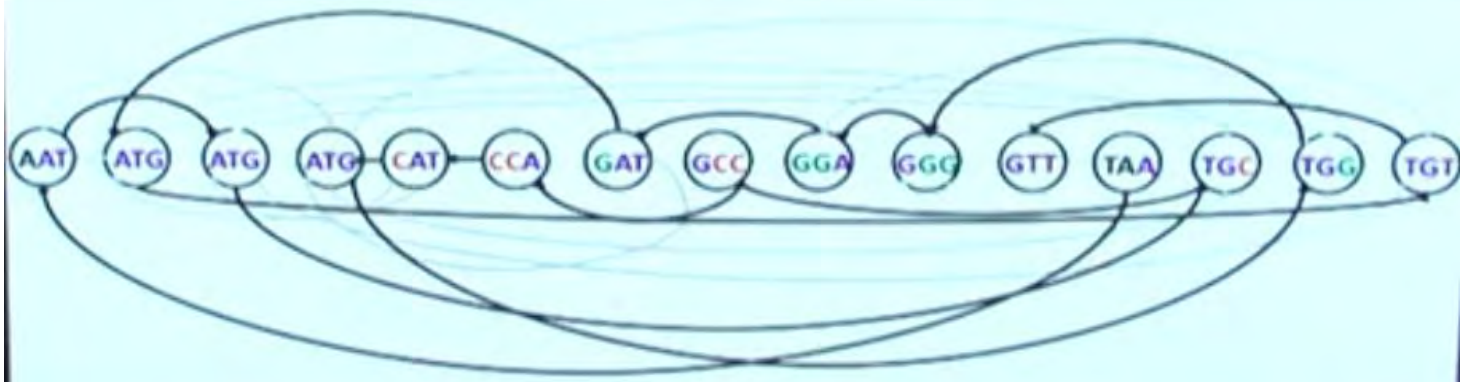


What are we trying to find in this graph?

Where Is the Genomic Path?

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

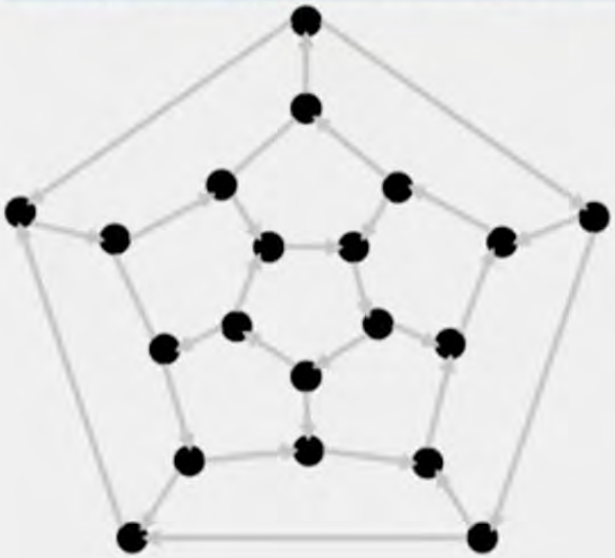
TAATGCCATGGGATGTT



What are we trying to find in this graph?

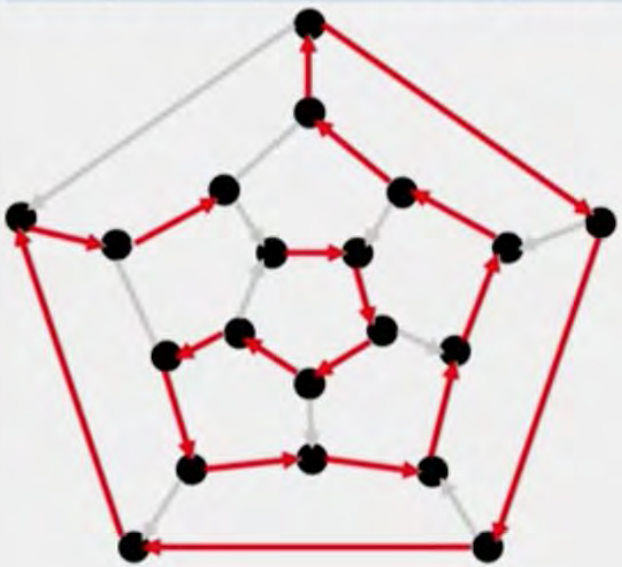
Does This Graph Have a Hamiltonian Path?

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



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.

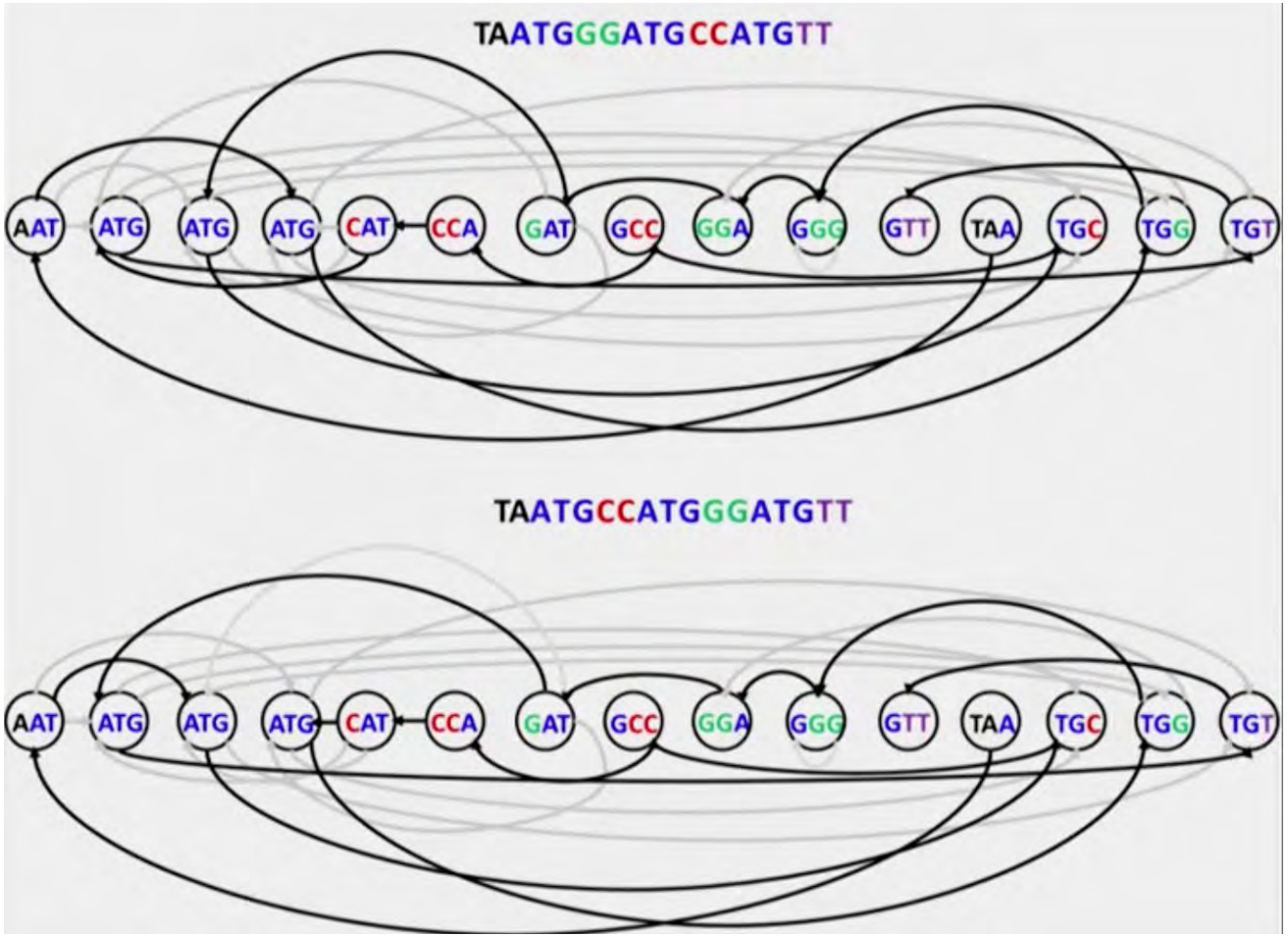


Sir William Rowan Hamilton (1805-1865)

Icosian game

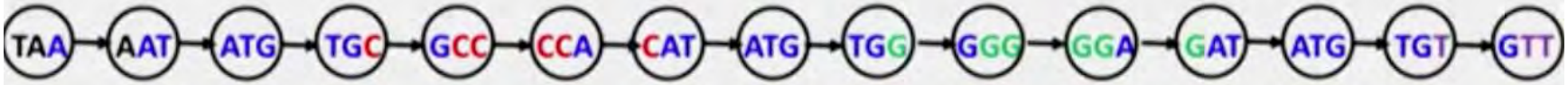
Outline

- 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

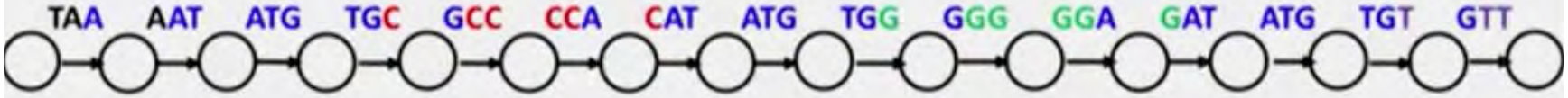

TAATGGGATGCCATGTT
AAT
ATG
ATG
ATG
CAT
CCA
GAT
GCC
GGA
GGG
GTT
TAA
TGC
TGG
TGT
TAATGCCATGGGATGTT
AAT
ATG
ATG
ATG
CAT
CCA
GAT
GCC
GGA
GGG
GTT
TAA
TGC
TGG
TGT

A Slightly Different Path

TAATGCCATGGGATGTT



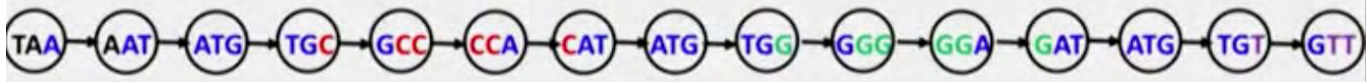
3-mers as **nodes**



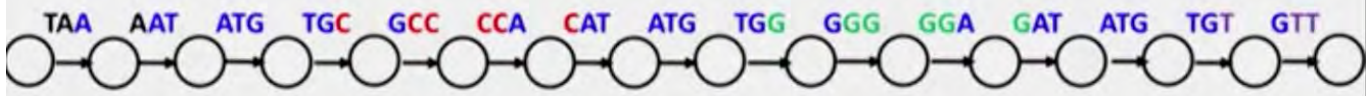
3-mers as **edges**

A Slightly Different Path

TAATGCCATGGGATGTT

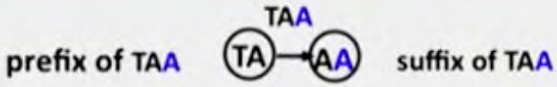


3-mers as **nodes**

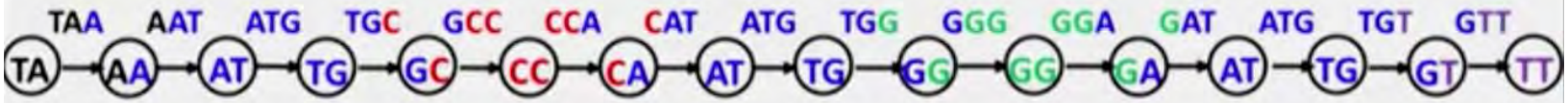


3-mers as **edges**

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

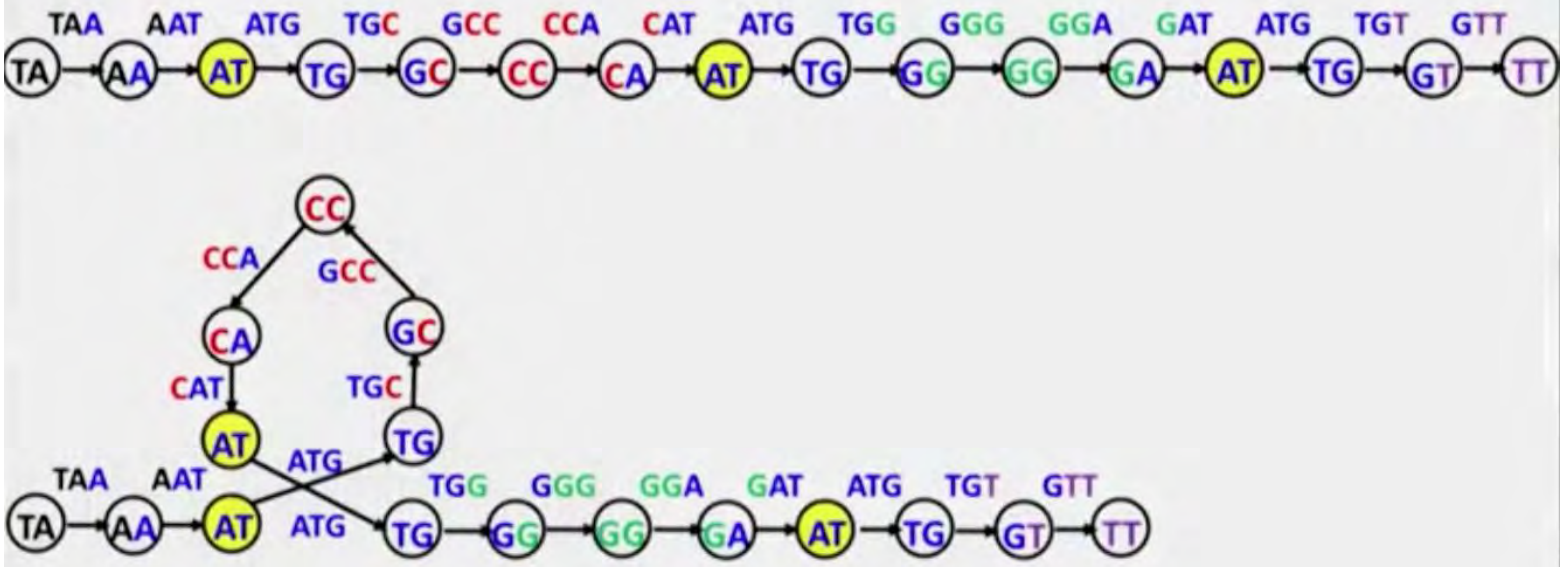


Labeling Nodes in the New Path

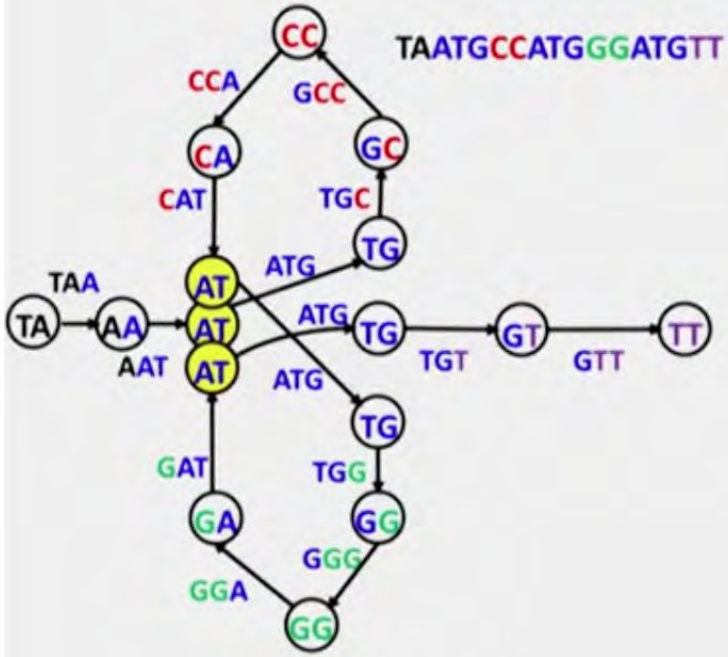


3-mers as **edges** and 2-mers as **nodes**

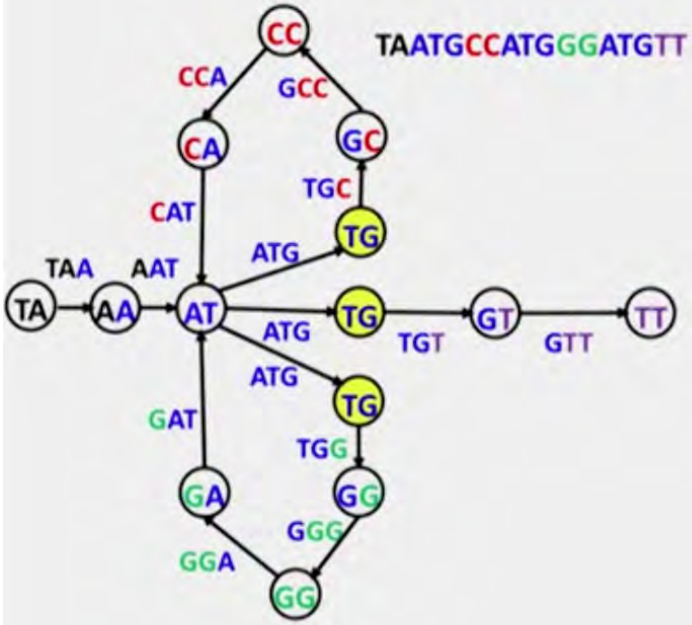
Gluing Identically Labeled Nodes



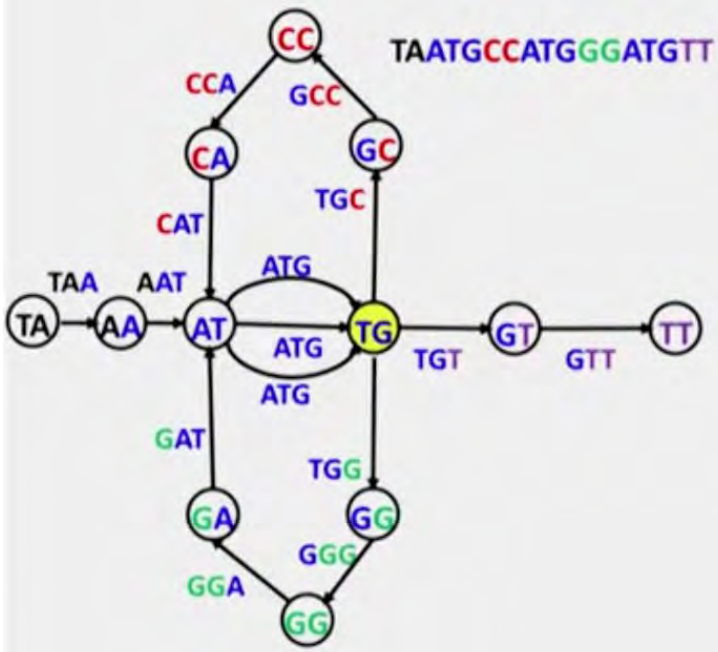
Gluing Identically Labeled Nodes



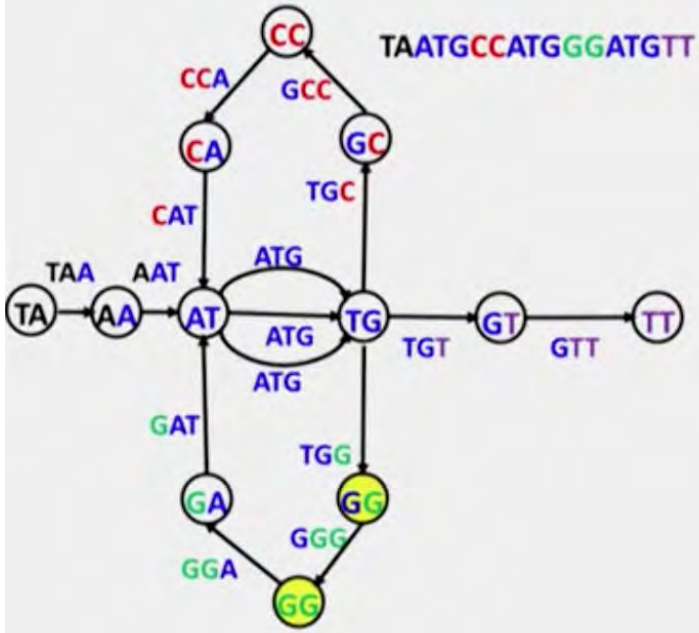
Gluing Identically Labeled Nodes



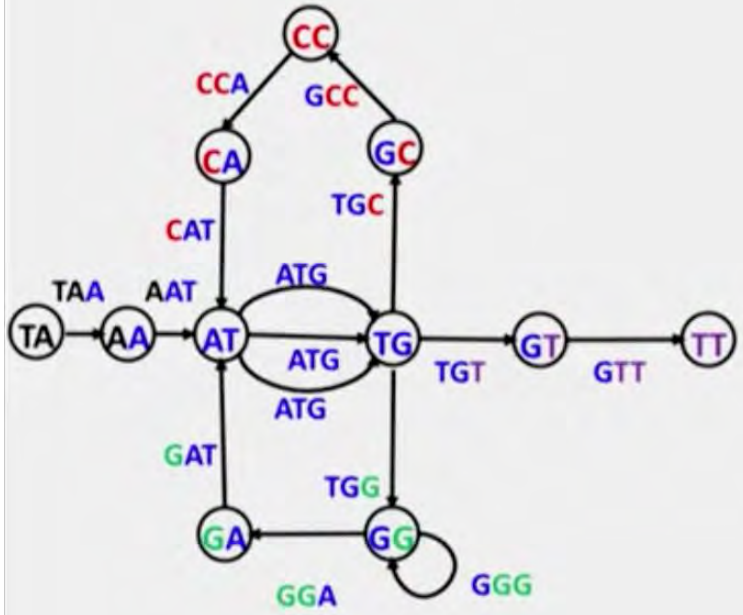
Gluing Identically Labeled Nodes



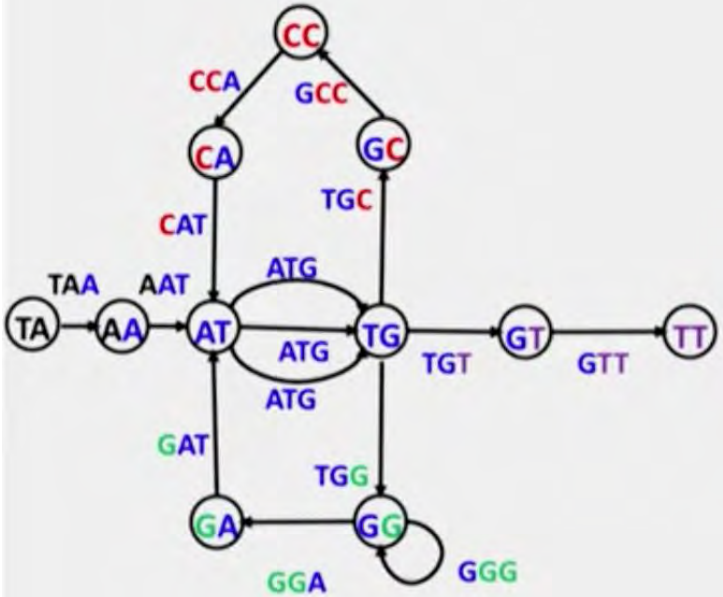
Gluing Identically Labeled Nodes



De Bruijn Graph of TAATGCCATGGGATGTT



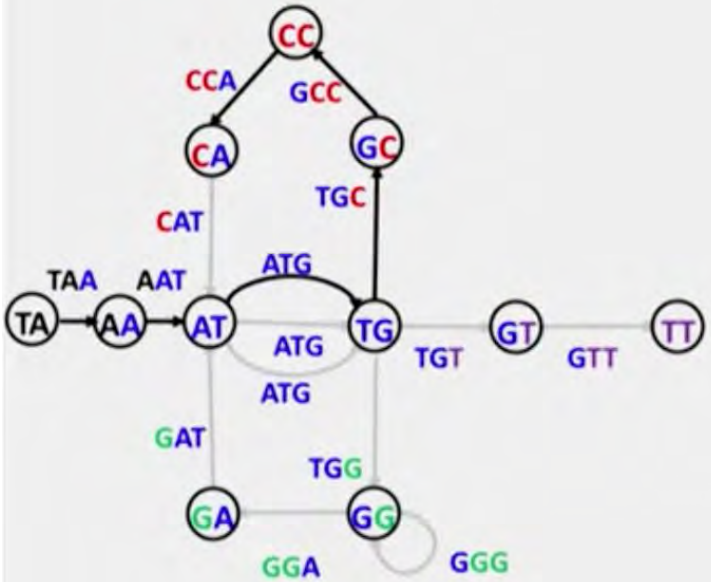
De Bruijn Graph of TAATGCCATGGGATGTT



Where is the *Genome* hiding in this graph?

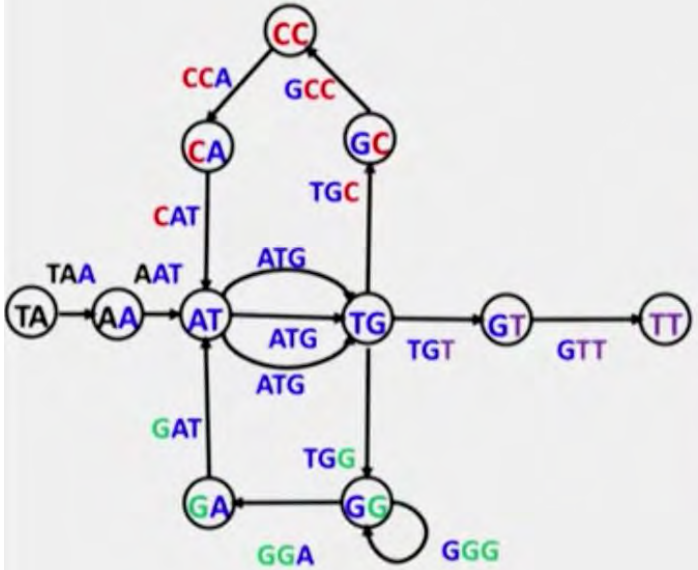
It Was Always There!

TAATGCCA



It Was Always There!

TAATGCCATGGGATGTT

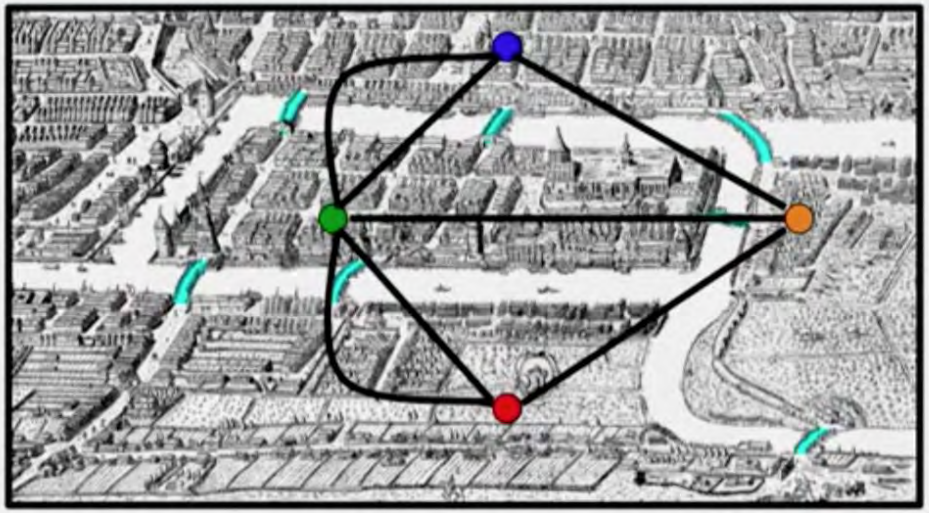


An **Eulerian path** in a graph is a path that visits each **edge** exactly once.

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.

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

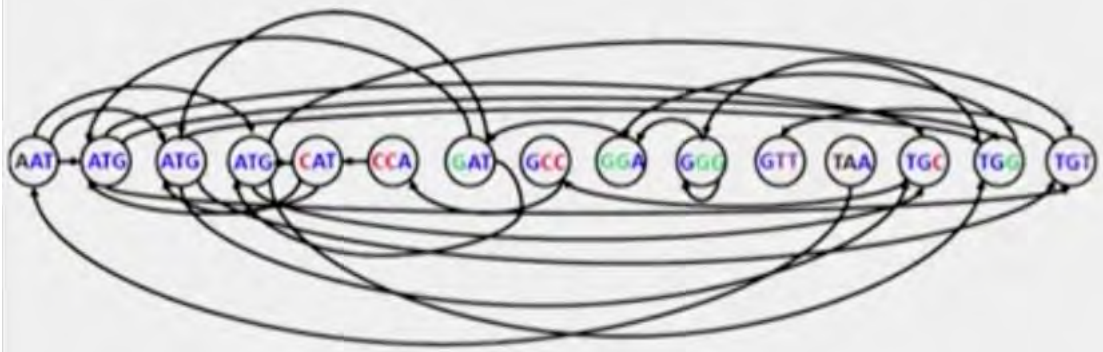
Eulerian Path Problem. Find an **Eulerian** path in a graph.

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

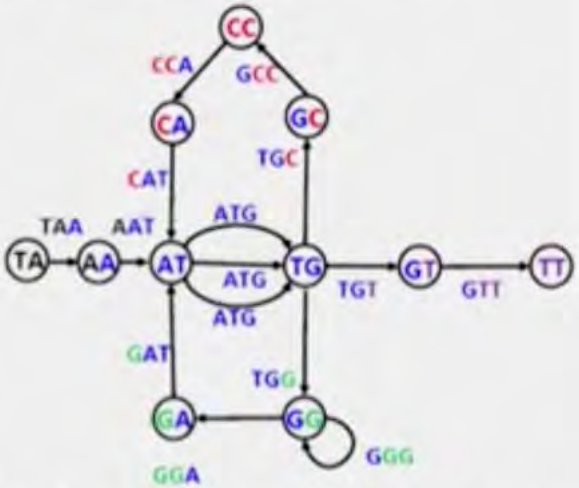
Hamiltonian Path Problem. Find a **Hamiltonian** path in a graph.

- **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

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."

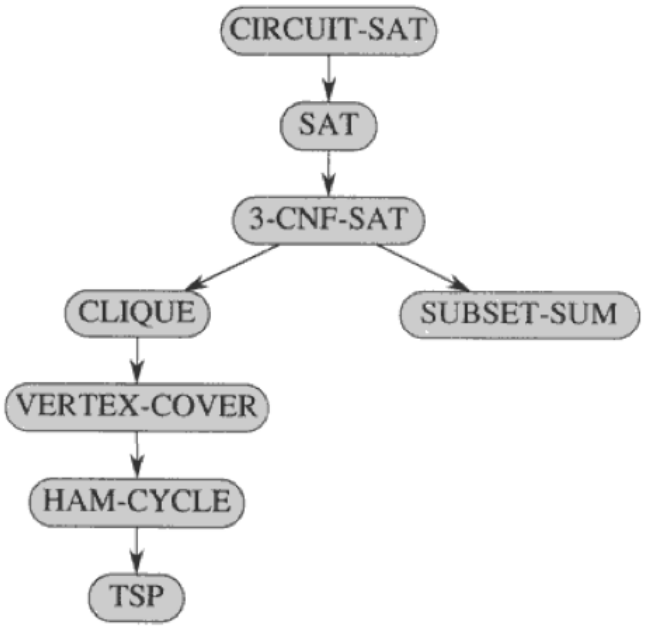
From Garey and Johnson. Computers and Intractability. 1979

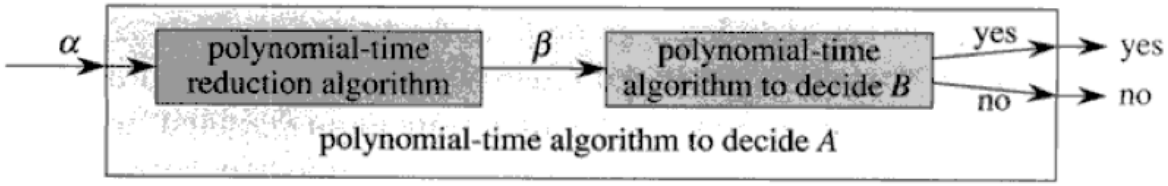
The Modern State of Affairs

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."

The Hamitonian path problem is NP-complete.





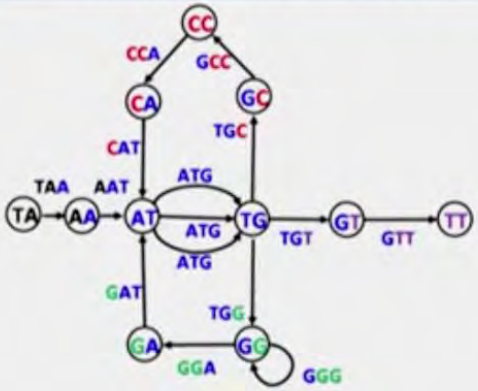
Outline

- 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

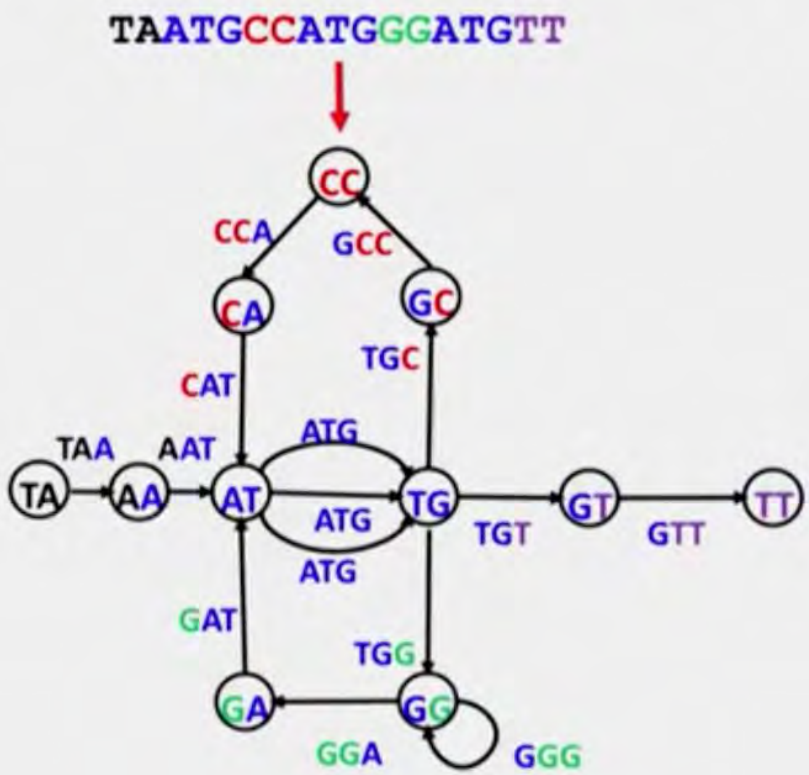
Eulerian Path Problem

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

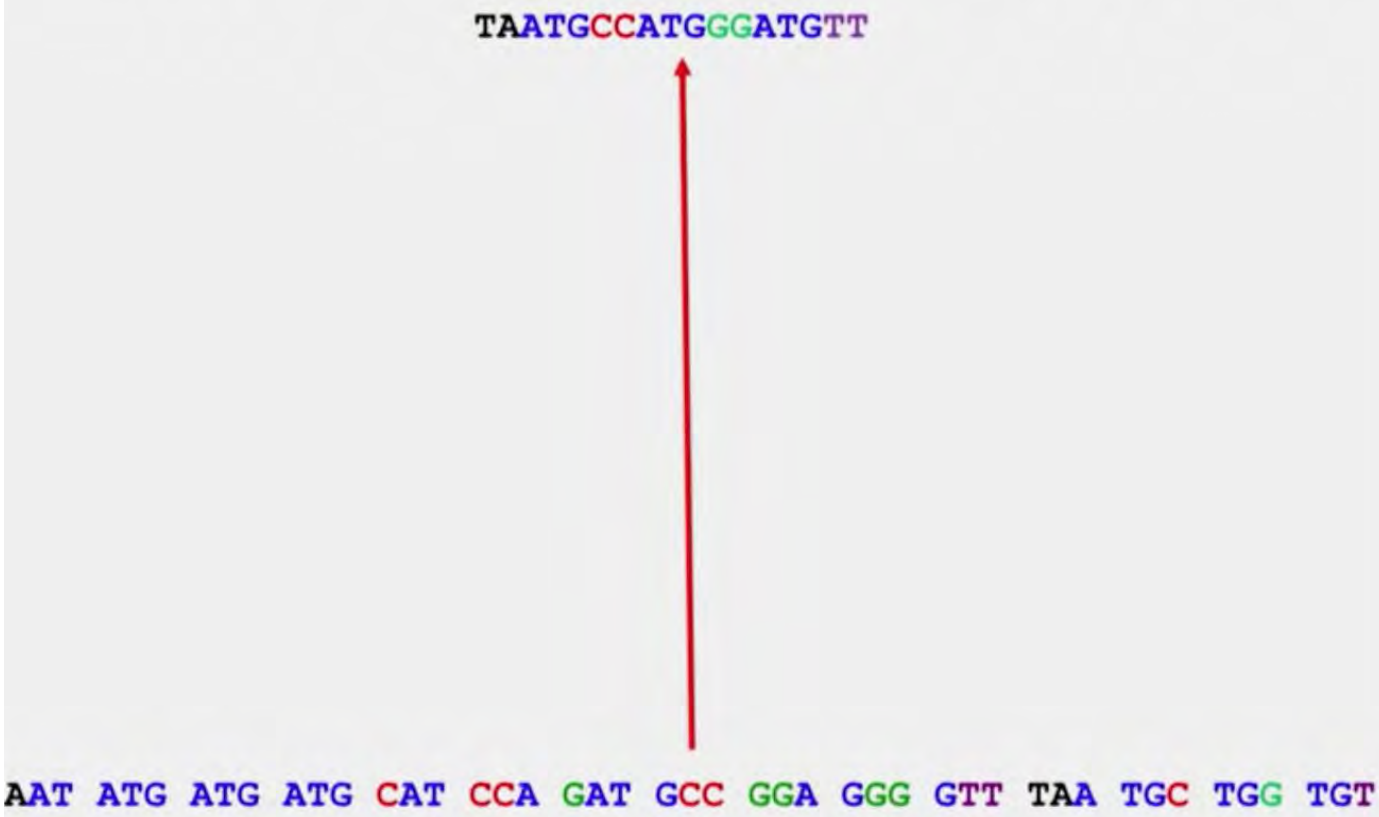
- **Input.** A graph.
- **Output.** A path visiting every **edge** in the graph exactly once.



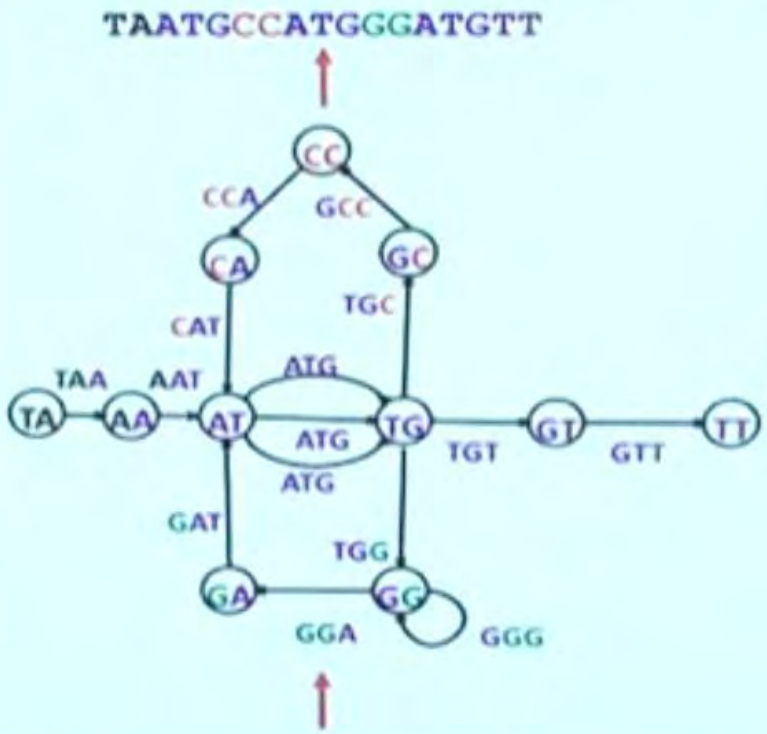
What We Have Done: From *Genome* to de Bruijn Graph



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



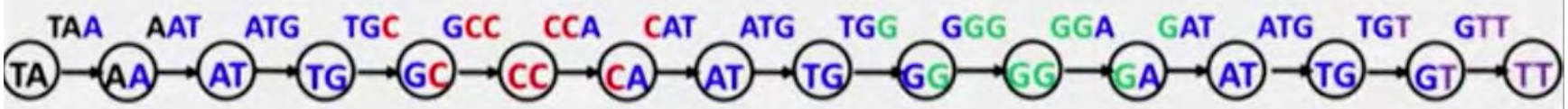
What We will Show: From Reads to de Bruijn Graph to *Genome*



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

Constructing de Bruijn Graph when *Genome* Is Known

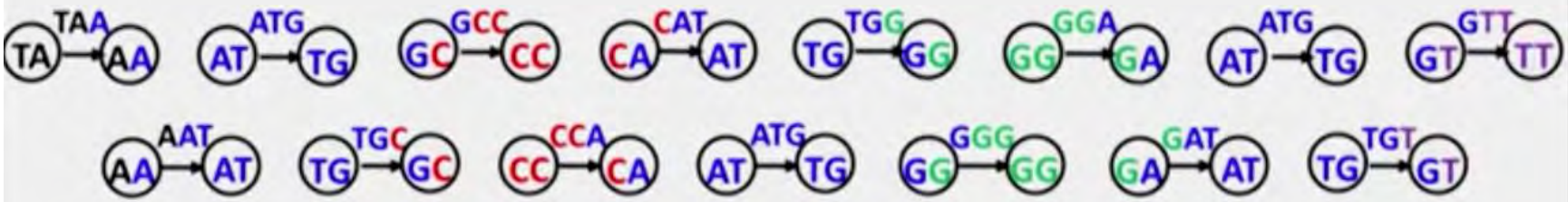
TAATGCCATGGGATGTT



Representing Composition as a Graph Consisting of Isolated Edges

Composition$_3$(**TAATGCCATGGGATGTT**)

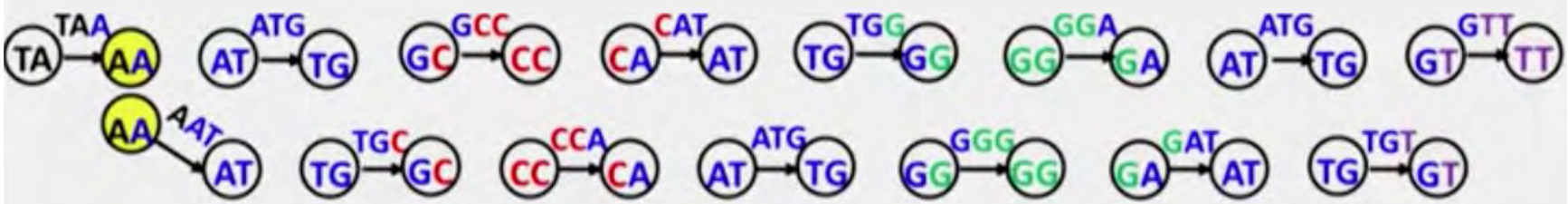
Constructing de Bruijn Graph from *k*-mer Composition



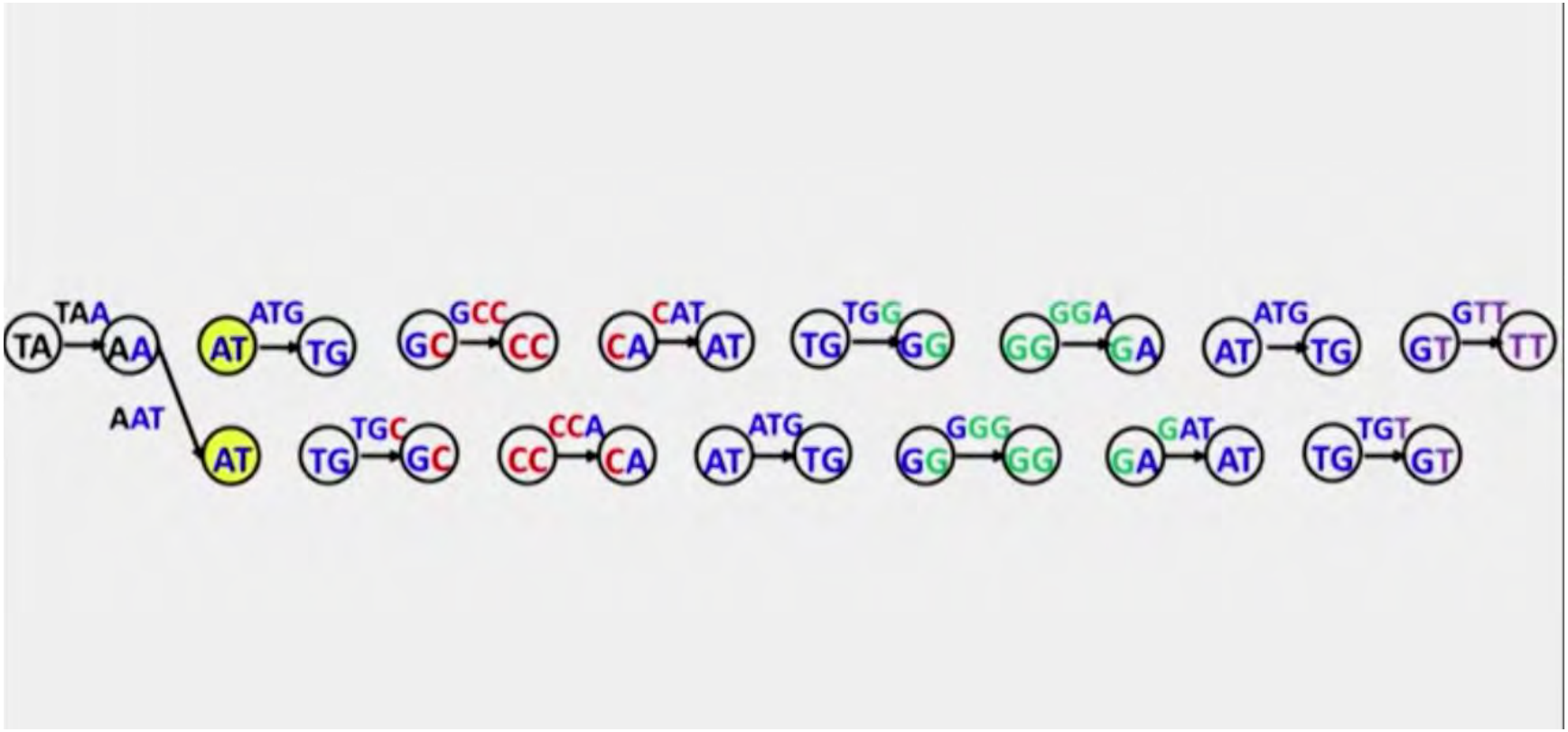
Composition$_3$(**TAATGCCATGGGATGTT**)

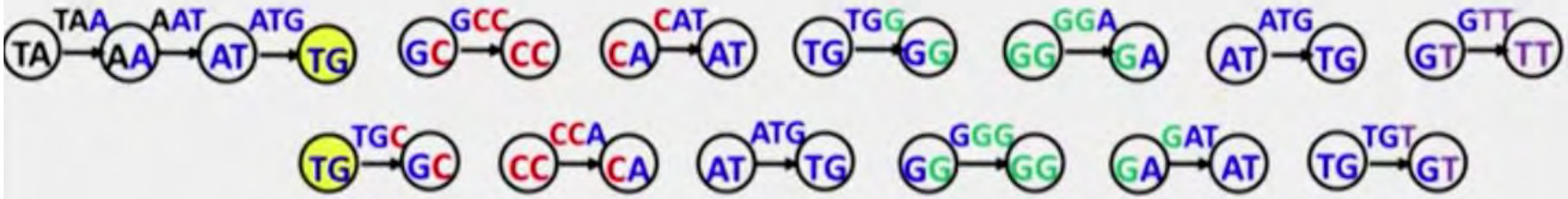
Gluing Identically Labeled Nodes

Gluing Identically Labeled Nodes

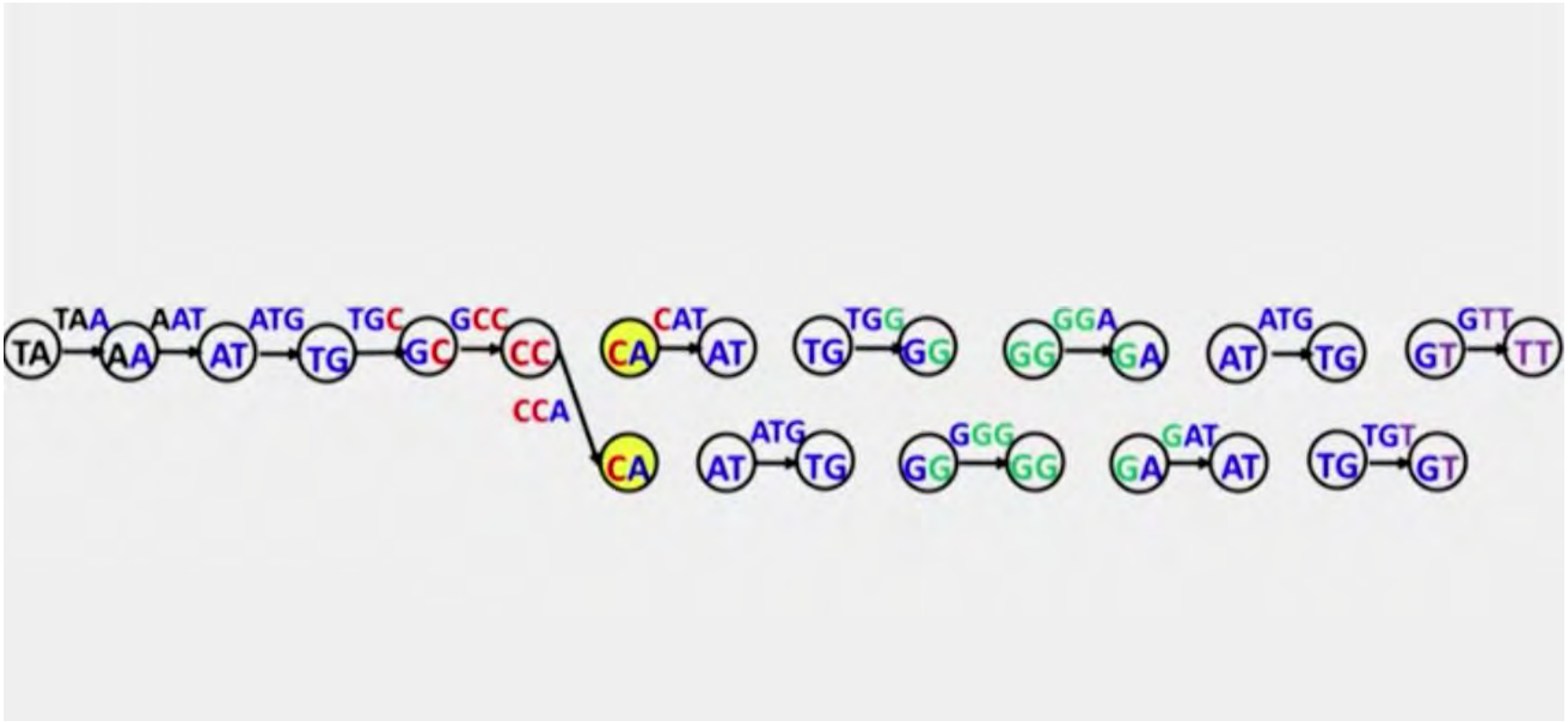



TA
TAA
AA
AAT
AT
ATG
TG
GC
GCC
CC
CA
CAT
AT
TG
TGG
GG
GG
GGA
GA
AT
ATG
TG
GT
GTT
TT
AT
TG
TGC
GC
CC
CCA
CA
AT
ATG
TG
GG
GGG
GG
GA
GAT
AT
TG
TGT
GT

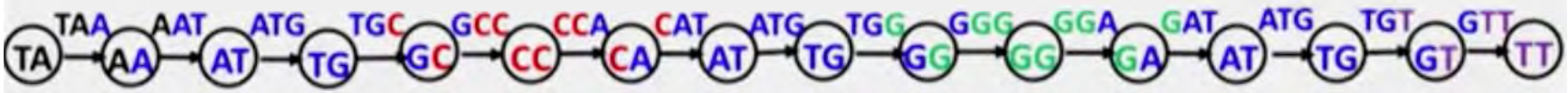

TA
TAA
AA
AAT
AT
ATG
TG
GC
GCC
CC
CA
CAT
AT
TG
TGG
GG
GG
GGA
GA
AT
ATG
TG
GT
GTT
TT
AT
TG
TGC
GC
CC
CCA
CA
AT
ATG
TG
GG
GGG
GG
GA
GAT
AT
TG
TGT
GT


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

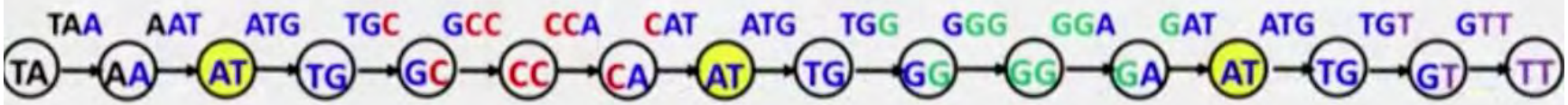

TA
TAA
AA
AAT
AT
ATG
TG
TGC
GC
GCC
CC
CA
CAT
AT
TG
TGG
GG
GG
GGA
GA
AT
ATG
TG
GT
GTT
TT
CC
CCA
CA
AT
ATG
TG
GG
GGG
GG
GA
GAT
AT
TG
TGT
GT

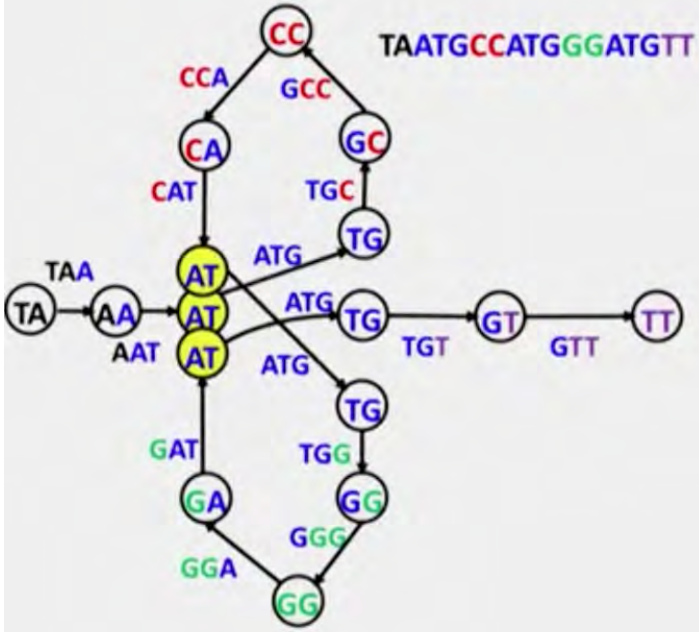

TA
TAA
AA
AAT
AT
ATG
TG
TGC
GC
GCC
CC
CCA
CA
CAT
AT
TG
TGG
GG
GG
GGA
GA
AT
ATG
TG
GT
GTT
TT
CA
AT
ATG
TG
GG
GGG
GG
GA
GAT
AT
TG
TGT
GT

We Are Not Done with Gluing Yet

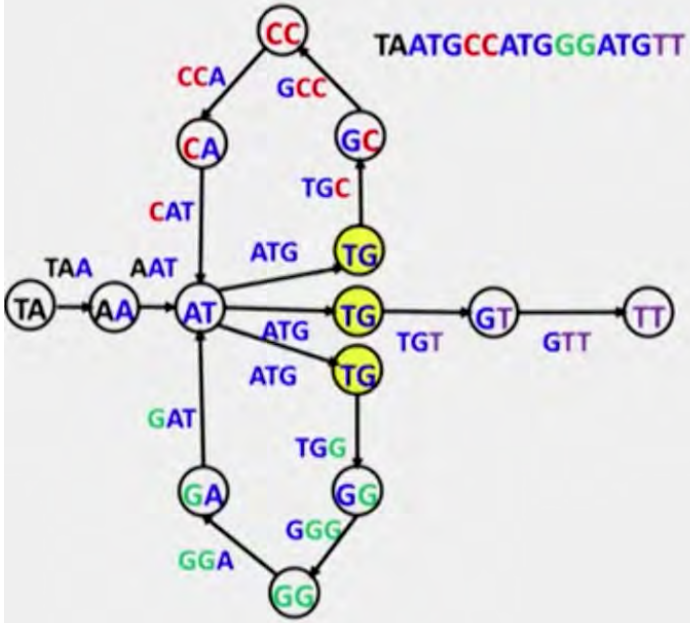


Gluing Identically Labeled Nodes

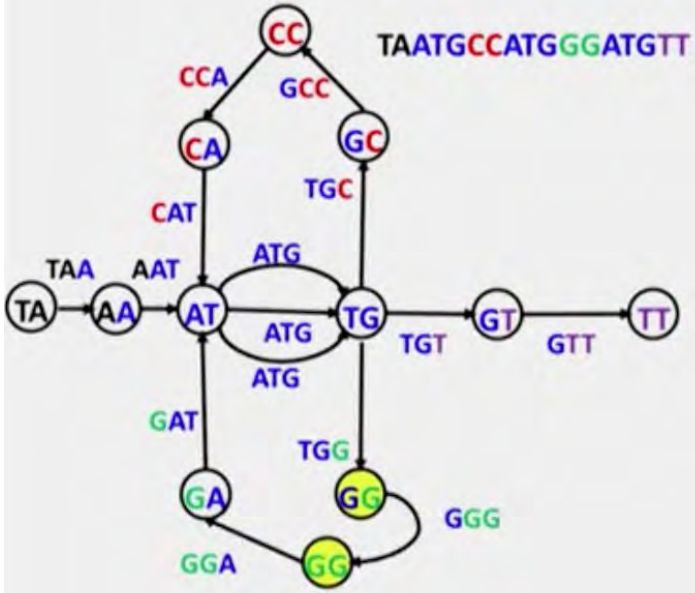



TAATGCCATGGGATGTT
CC
CCA
GCC
CA
GC
CAT
TGC
TG
ATG
TAA
AT
TA
AA
AT
ATG
TG
GT
TT
AAT
AT
TGT
GTT
ATG
TG
GAT
TGG
GA
GG
GGG
GGA
GG

Gluing Identically Labeled Nodes

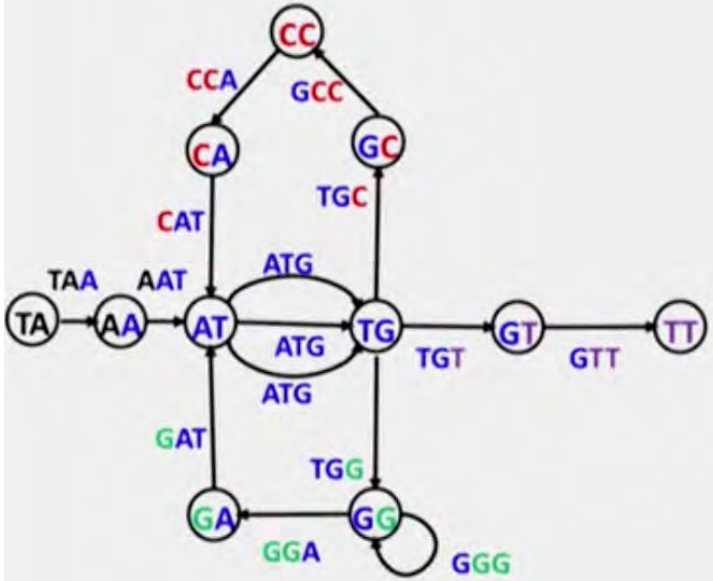


Gluing Identically Labeled Nodes



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.

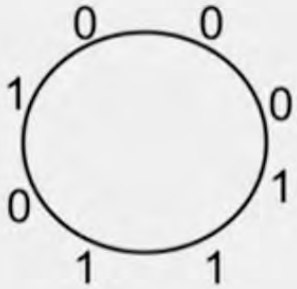
DeBruijn(k-mers)
form a node for each (*k-1*)-mer from ***k-mers***
for each *k*-mer in ***k-mers***
connect its prefix node with its suffix node by an edge

Nicolaas Govert "Dick" de Bruijn (9 July 1918 – 17 February 2012)

From Hamilton  to Euler  to de Bruijn 

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

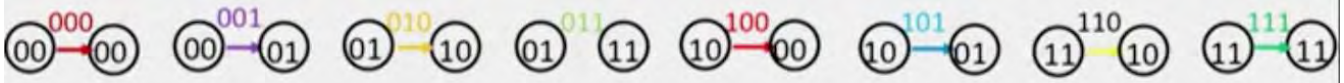
000 001 010 011 100 101 110 111

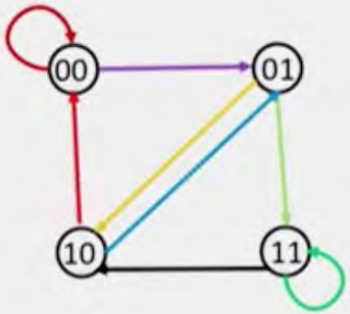


From Hamilton  to Euler  to de Bruijn 

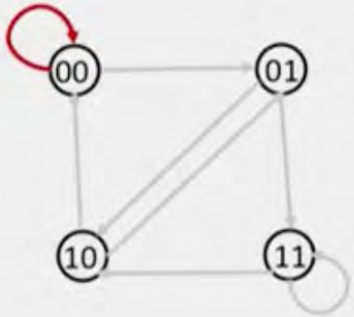
Universal String Problem (Nicolaas de Bruijn, 1946). Find a circular string containing each binary *k*-mer exactly once.

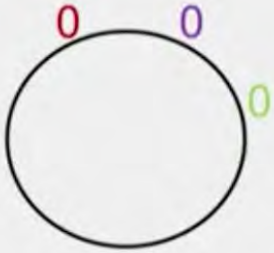
000 001 010 011 100 101 110 111



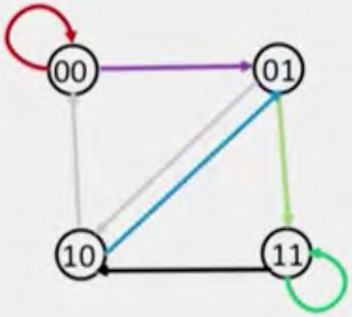


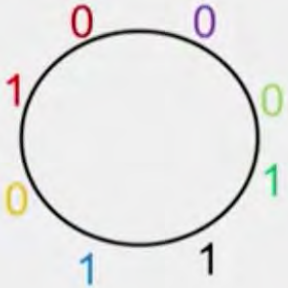
From Hamilton  to Euler  to de Bruijn 



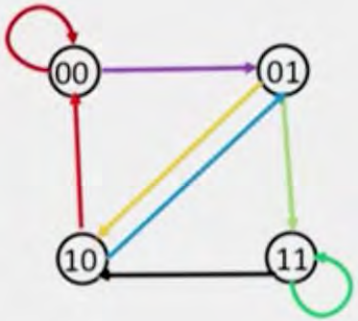


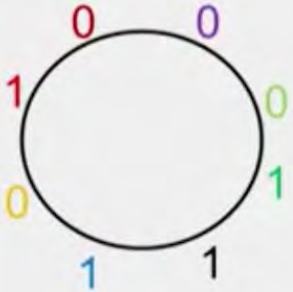
From Hamilton  to Euler  to de Bruijn 



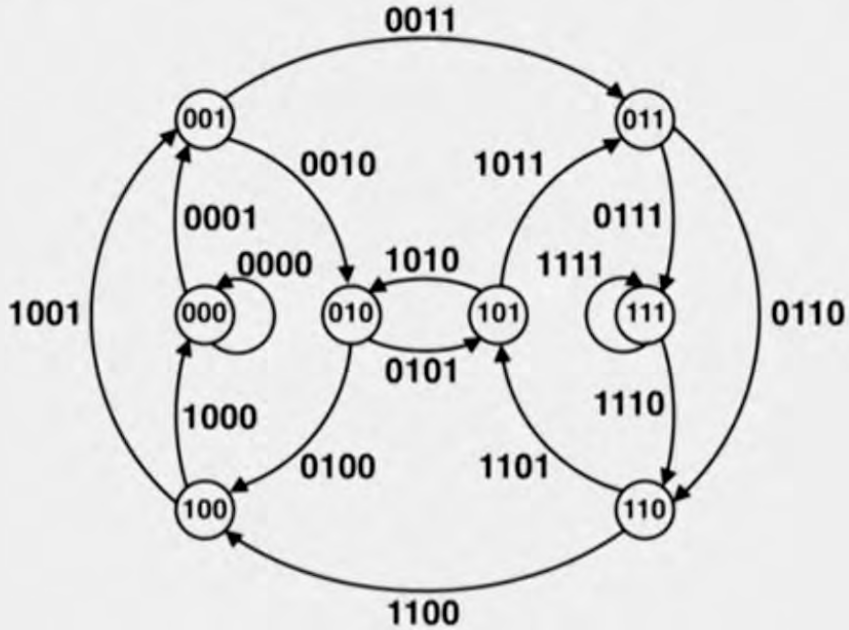


From Hamilton  to Euler  to de Bruijn 





De Bruijn Graph for 4-Universal String



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

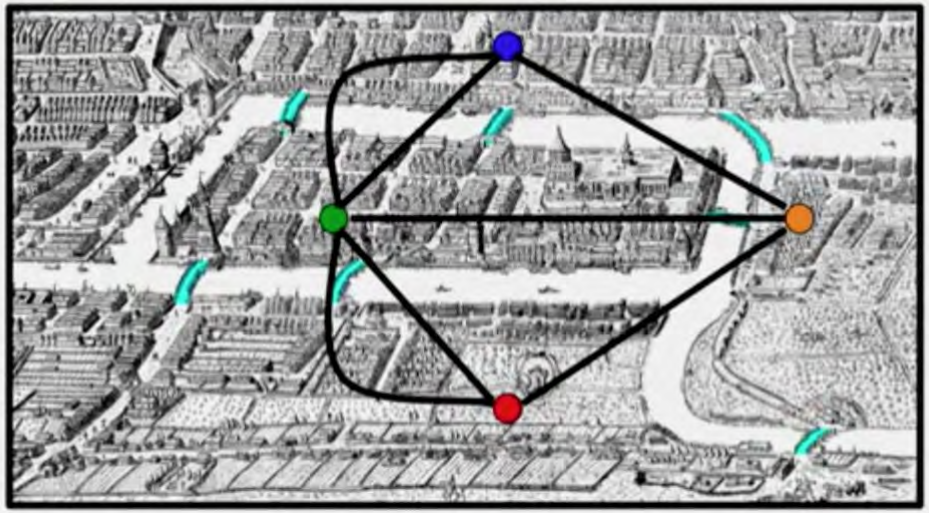
Outline

- 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.

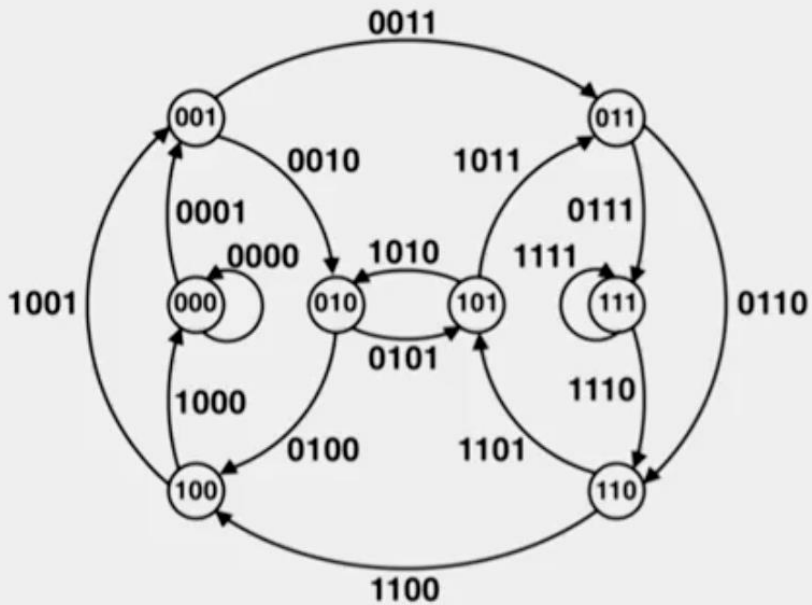


A Graph is **Eulerian** if It Contains an Eulerian Cycle.

Is this graph Eulerian?

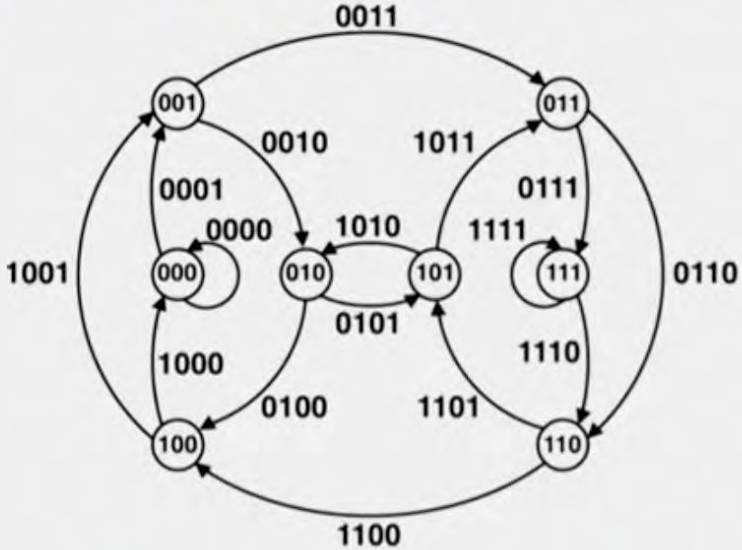
A graph is **balanced** if ***indegree*** = ***outdegree*** for each node

Is the Graph for 4-Universal String Balanced?



Euler's Theorem

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

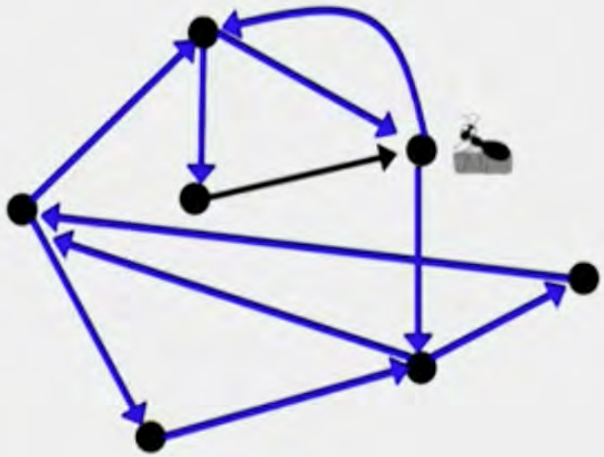


(*) and strongly connected, of course!

Recruiting an Ant to Prove Euler's Theorem

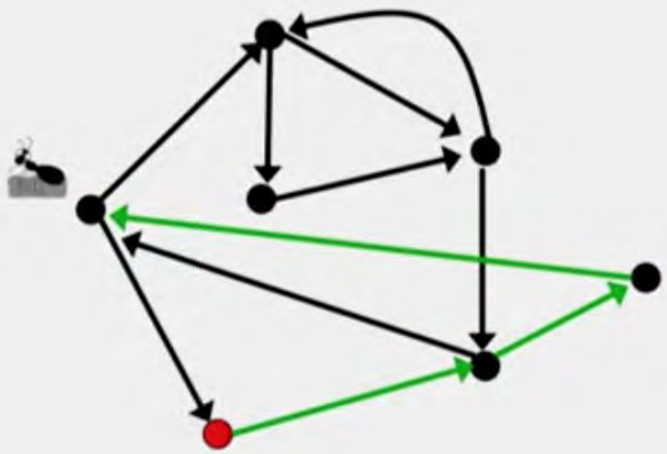
Let an ant randomly walk through the graph.
The ant cannot use the same edge twice!

If Ant Was a Genius…

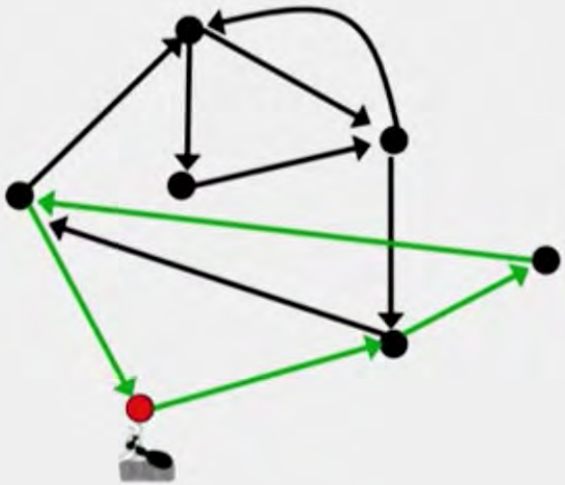


Walking… and Walking… and Walking…

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

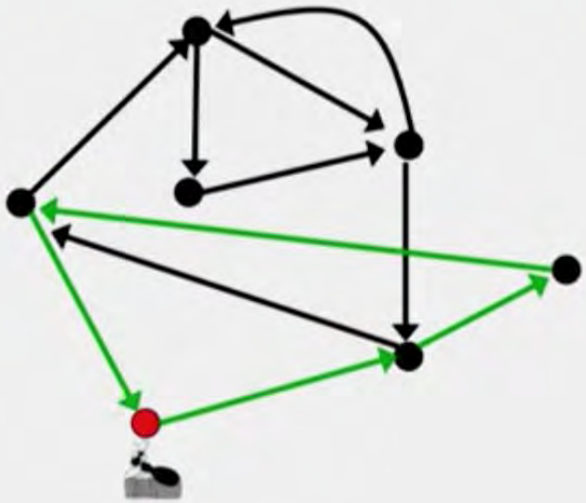


The Ant Can Only Get Stuck at the Starting Node



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

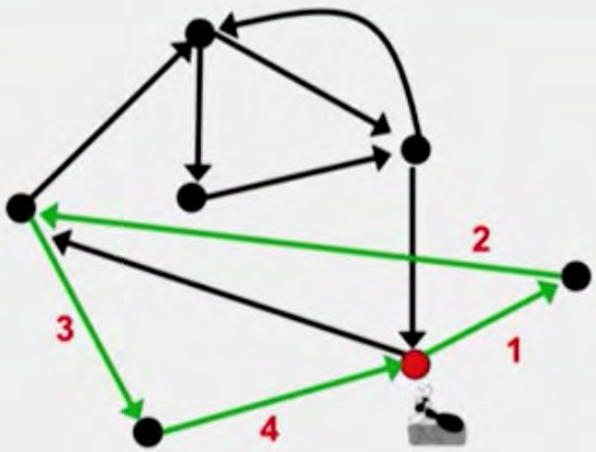
The constructed cycle is not Eulerian. **Can we enlarge it?**



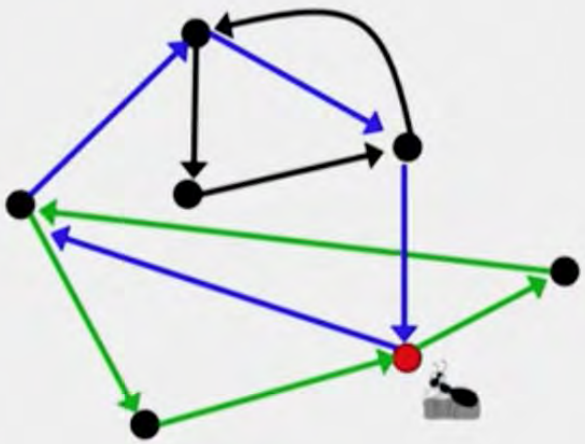
I Returned Back BUT… I Can Continue Walking!

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

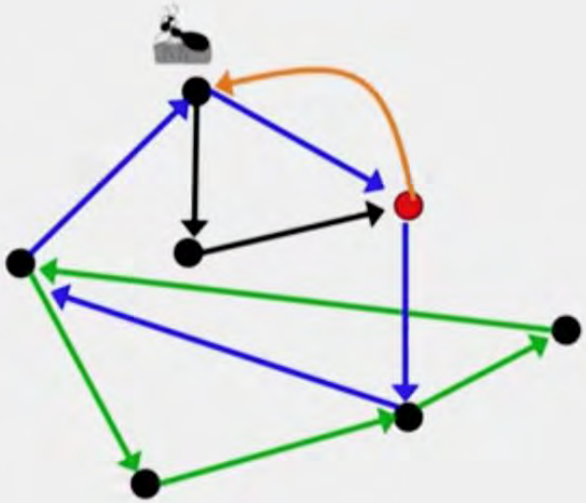
After completing the cycle, start random exploration of still untraversed edges in the graph.



Enlarging the Previously Constructed Cycle

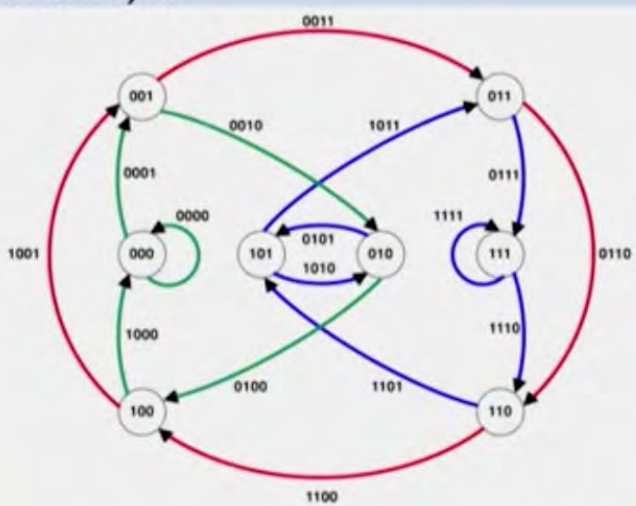


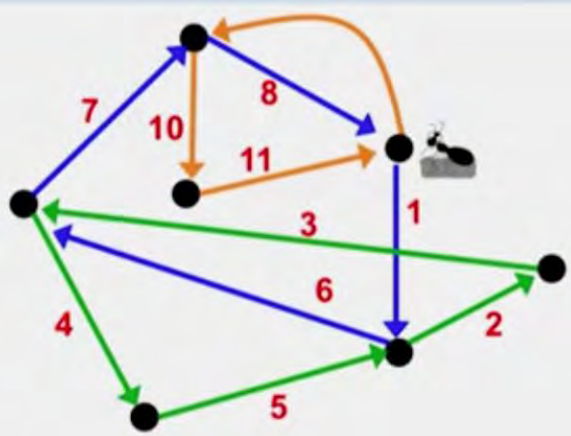
Enlarging the Green-Blue Cycle



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*



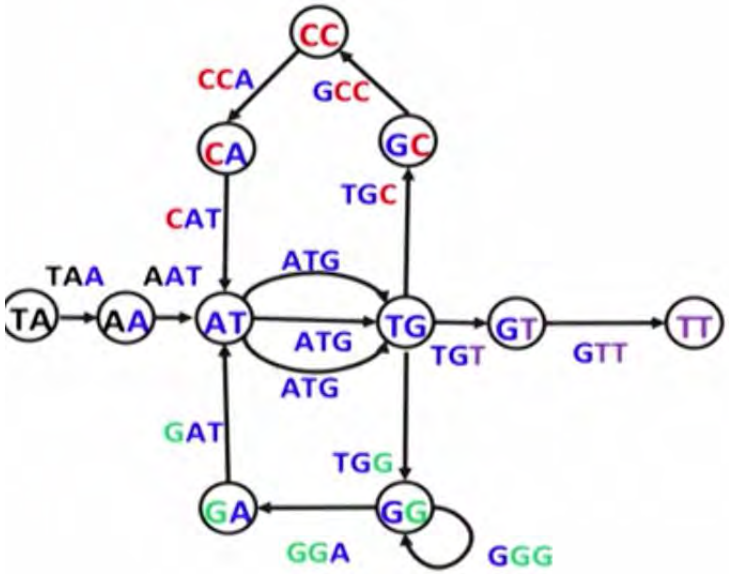


Outline

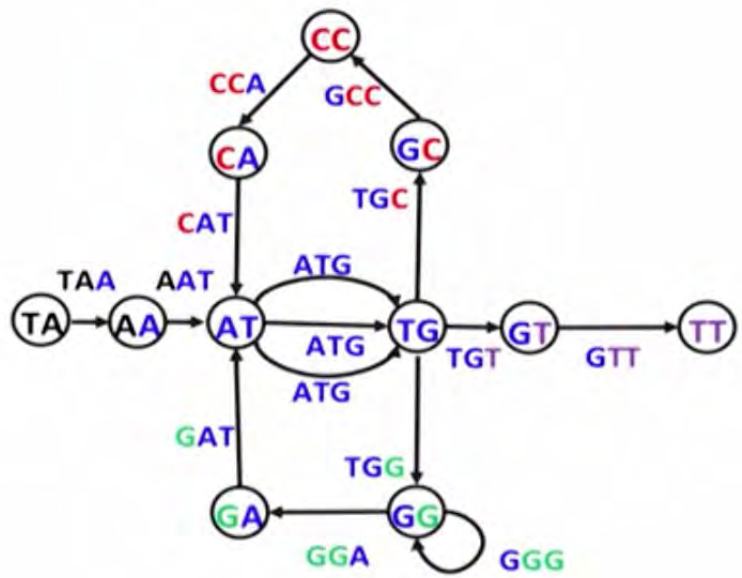
- What Is Genome Sequencing?
- Exploding Newspapers
- The String Reconstruction Problem
- String Reconstruction as a Hamiltonian Path Problem
- String Reconstruction as an Eulerian Path Problem
- Similar Problems with Different Fates
- De Bruijn Graphs
- Euler's Theorem
- **Assembling Read-Pairs**
-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

Multiple Eulerian Paths

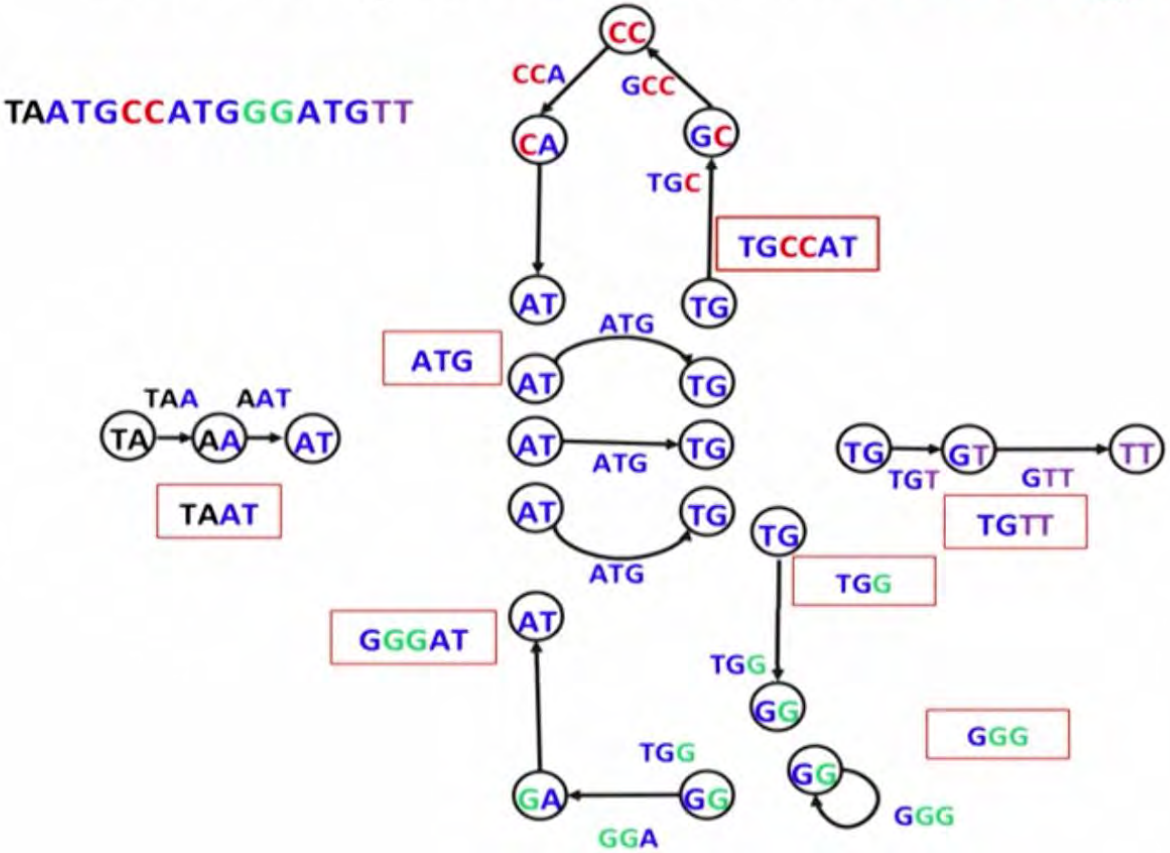
TAATGCCATGGGATGTT



TAATGGGATGCCATGTT



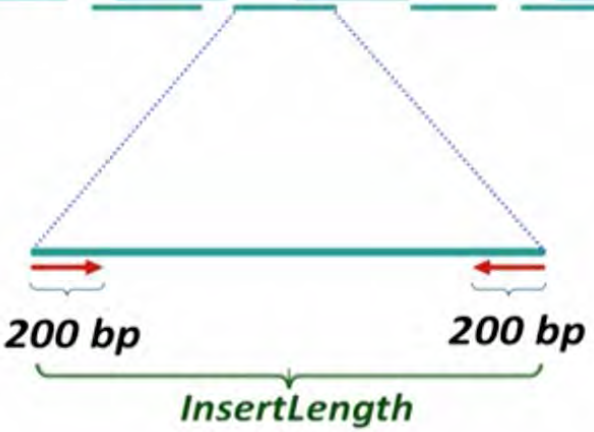
Breaking Genome into Contigs



DNA Sequencing with Read-pairs

Multiple identical copies of genome

Randomly cut genomes into large equally sized fragments of size ***InsertLength***



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

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

Genome

Read 1 → *Read 2* →

...A T C A G A T T A C G T T C C G A G ...

Distance d=11

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
 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

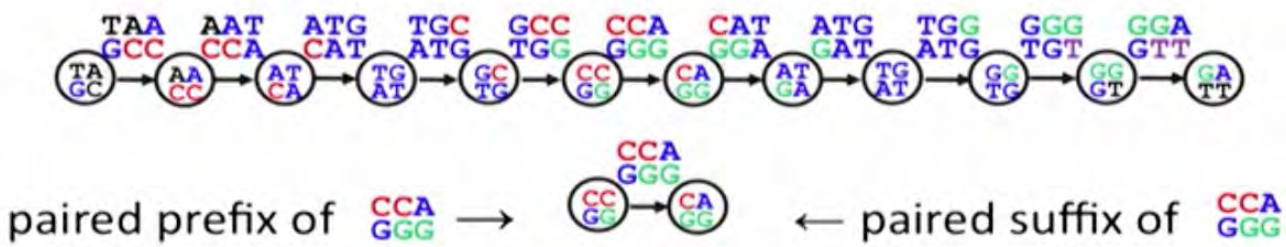
Representing *PairedComposition* in lexicographic order

String Reconstruction from Read-Pairs Problem

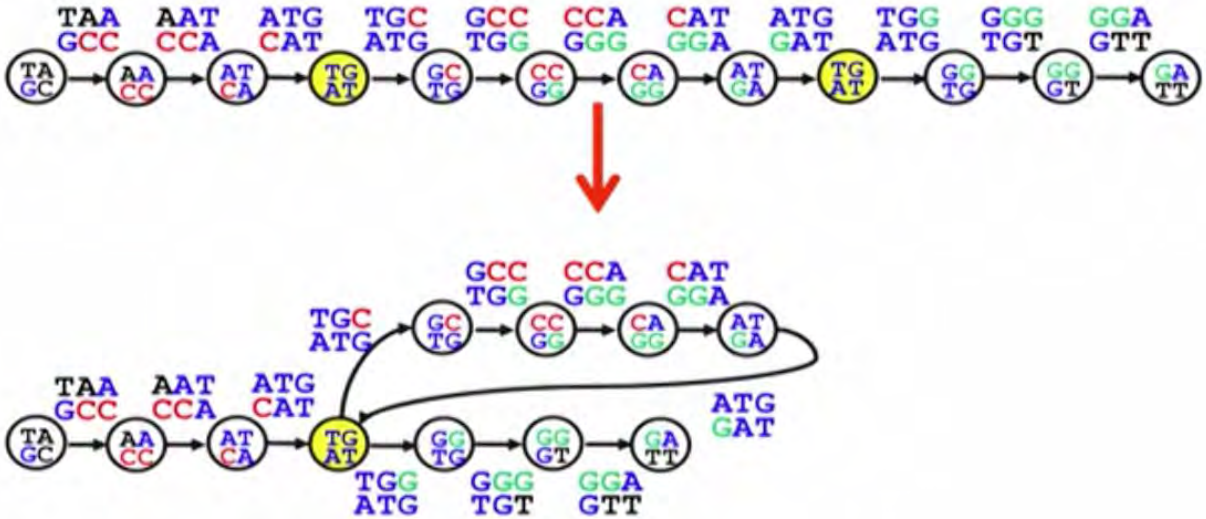
String Reconstruction from Read-Pairs Problem. Reconstruct a string from its paired *k*-mers.

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

Labeling Nodes by Paired Prefixes and Suffixes

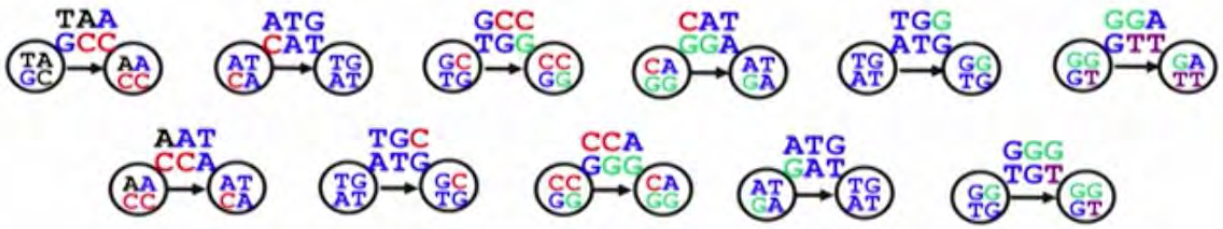


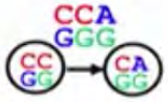
Glue nodes with identical labels



Paired de Bruijn Graph from the Genome

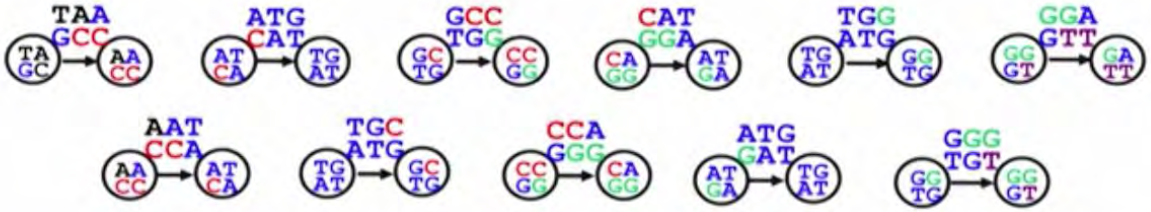
Constructing Paired de Bruijn Graph



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

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?

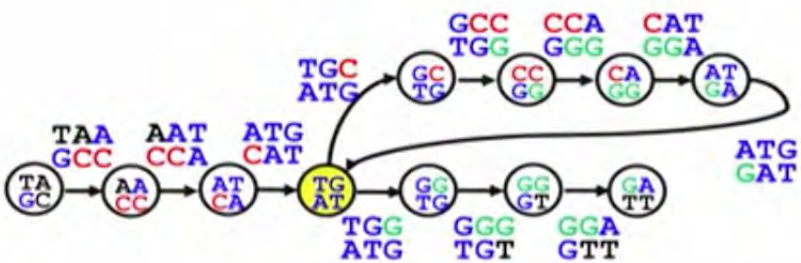
Unique genome reconstruction

TAATGCCATGGGATGTT

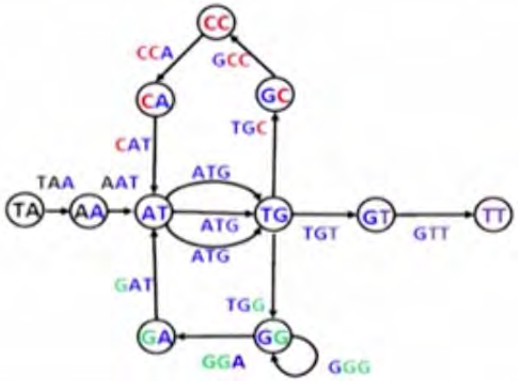
Multiple genome reconstructions

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.**

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
```

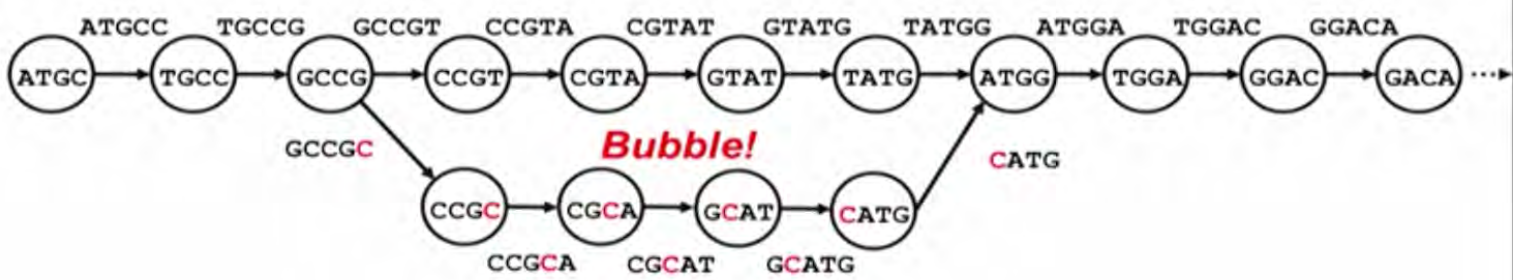
2nd Unrealistic Assumption: Error-free Reads

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

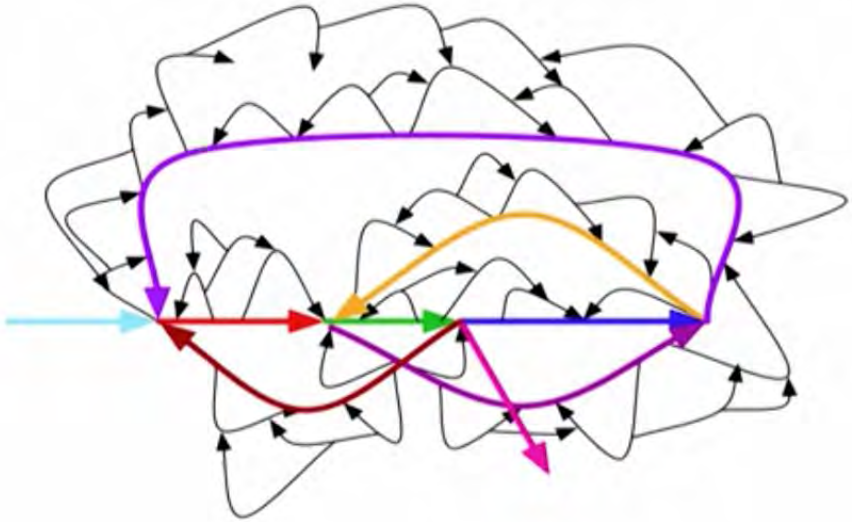
Erroneous read
(change of t into C)

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

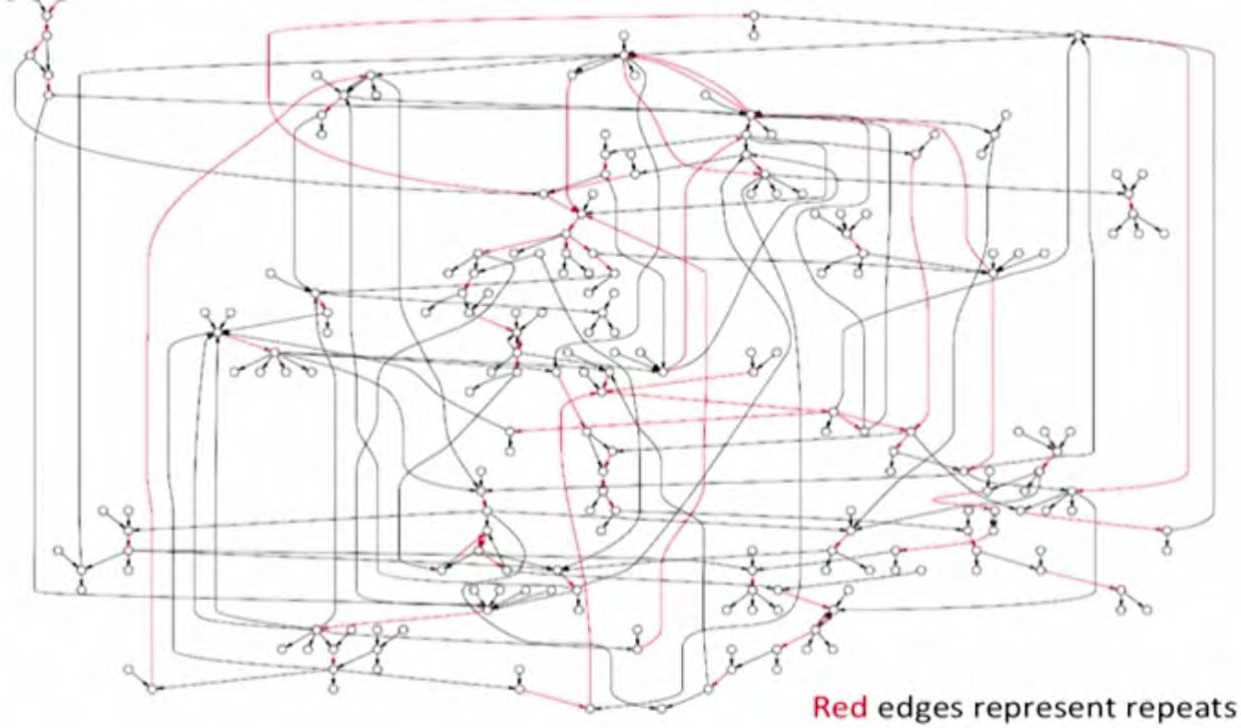
Errors in Reads Lead to **Bubbles** in the De Bruijn Graph



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



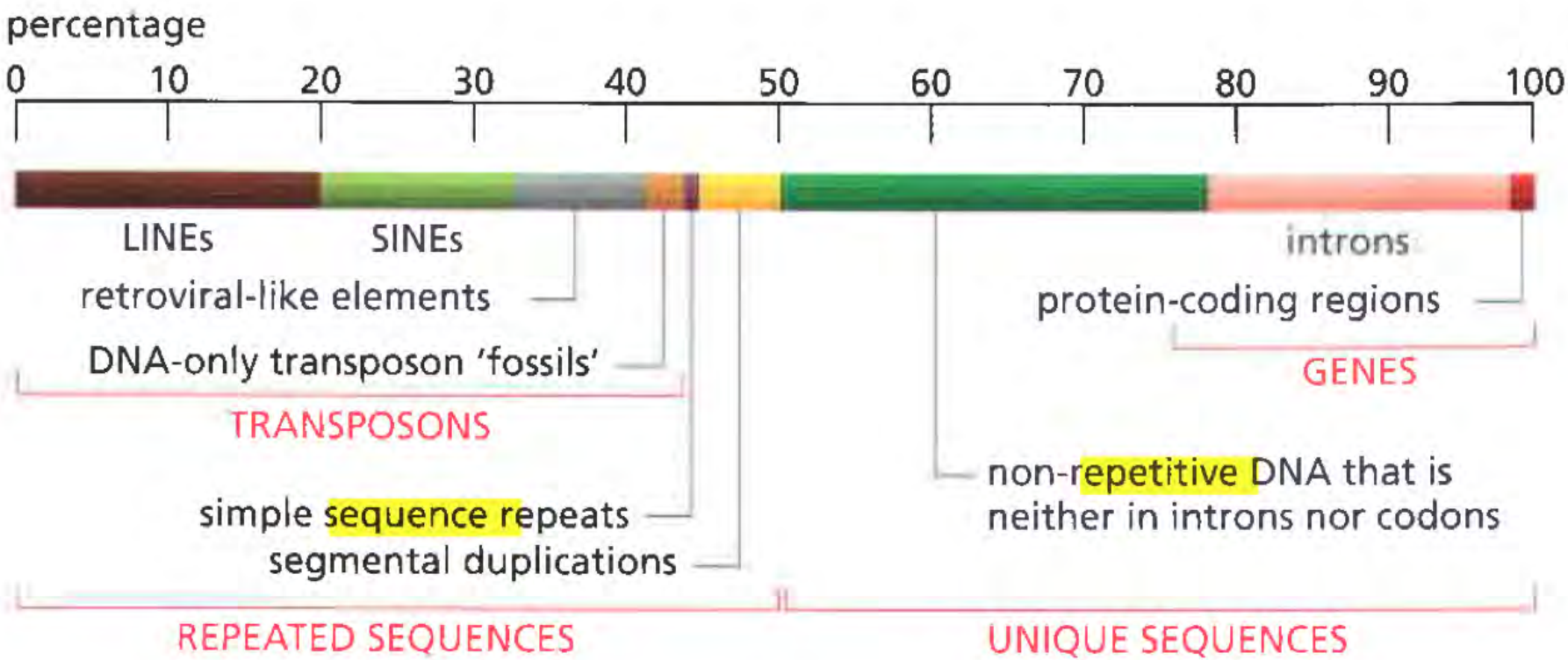
De Bruin Graph of *N. meningitidis* Genome AFTER Removing Bubbles



Repeats in the human genome:

-Alu sequencs: ~300 bp long, repeated approx 1 million times in the genome
-LINE repeats: ~1000bp long, repeated approx 200,000 times
-approximately 25% of human genes are duplicated

=> Repeats and duplicates make up approx half the human genome



Velvet: Algorithms for de novo short read assembly using de Bruijn graphs

Daniel R. Zerbino and Ewan Birney[1]

EMBL-European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, United Kingdom

We have developed a new set of algorithms, collectively called "Velvet," to manipulate de Bruijn graphs for genomic sequence assembly. A de Bruijn graph is a compact representation based on short words (*k*-mers) that is ideal for high coverage, very short read (25–50 bp) data sets. Applying Velvet to very short reads and paired-ends information only, one can produce contigs of significant length, up to 50-kb N50 length in simulations of prokaryotic data and 3-kb N50 on simulated mammalian BACs. When applied to real Solexa data sets without read pairs, Velvet generated contigs of ~8 kb in a prokaryote and 2 kb in a mammalian BAC, in close agreement with our simulated results without read-pair information. Velvet represents a new approach to assembly that can leverage very short reads in combination with read pairs to produce useful assemblies.

Genome Science (2008)

An Eulerian path approach to DNA fragment assembly

Pavel A. Pevzner*, Haixu Tang†, and Michael S. Waterman†‡§

*Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA; and Departments of †Mathematics and ‡Biological Sciences, University of Southern California, Los Angeles, CA

Contributed by Michael S. Waterman, June 7, 2001

For the last 20 years, fragment assembly in DNA sequencing followed the "overlap–layout–consensus" paradigm that is used in all currently available assembly tools. Although this approach proved useful in assembling clones, it faces difficulties in genomic shotgun assembly. We abandon the classical "overlap–layout–consensus" approach in favor of a new EULER algorithm that, for the first time, resolves the 20-year-old "repeat problem" in fragment assembly. Our main result is the reduction of the fragment assembly to a variation of the classical Eulerian path problem that allows one to generate accurate solutions of large-scale sequencing problems. EULER, in contrast to the CELERA assembler, does not mask such repeats but uses them instead as a powerful fragment assembly tool.

Because the Eulerian path approach transforms a once difficult layout problem into a simple one, a natural question is: "Could the Eulerian path approach be applied to fragment assembly?" Idury and Waterman, mimicked fragment assembly as an SBH problem (11) by representing every read of length n as a collection of $n - l + 1$ overlapping l-tuples (continuous short strings of fixed length l). At first glance, this transformation of every read into a collection of l-tuples (breaking the puzzle into smaller pieces) is a very short-sighted procedure, because information about the sequencing reads is lost. However, the loss of information is minimal for large l and is well paid for by the computational advantages of the Eulerian path approach. In addition, lost information can be restored at later stages.

How to apply de Bruijn graphs to genome assembly

Phillip E C Compeau, Pavel A Pevzner & Glenn Tesler

A mathematical concept known as a de Bruijn graph turns the formidable challenge of assembling a contiguous genome from billions of short sequencing reads into a tractable computational problem.

Nature Biotech, Nov 2011

De Bruijn Sequences—A Model Example of the Interaction of Discrete Mathematics and Computer Science

Combinatorics, graph theory, and abstract algebra can all be applied to the same algorithmic problem.

ANTHONY RALSTON
SUNY at Buffalo
Amherst, NY 14226

Mathematics Magazine, 1982.

Customers DevNet SampleNet Search

PACIFIC BIOSCIENCES®

APPLICATIONS PRODUCTS COMMUNITY SERVICES & SUPPORT NEWS & EVENTS ABOUT US BLOG

PACBIO RS II

Single Molecule, Real-Time DNA sequencing provides the industry's highest consensus accuracy and longest read lengths.

Applications and research areas include:

- DE NOVO ASSEMBLY
- TARGETED SEQUENCING
- BASE MODIFICATIONS
- ISOFORM SEQUENCING DETECTION
- MICROBIOLOGY
- PLANT & ANIMAL
- HUMAN

LEARN MORE

UPCOMING WEBINARS

Feb. 11 & 12: Iso-Seq Sample Preparation & Bioinformatics Tips and Tricks

REGISTER NOW

Oxford NANOPORE technologies

Home Technology About us News Careers

Technology

- Introduction to nanopore sensing
  - Biological nanopores
  - Solid-state nanopores
- Electronics for nanopore sensing
- The MinION™ device: a miniaturised sensing system
- The PromethION™ system
- The GridION™ system
- Workflow versatility: no fixed run time
- Nanopore sensing: informatics
- Automatic optimisation of system performance
- Analytes and applications: DNA, RNA, proteins
- Fields of use
- Publications

Introduction to nanopore sensing

The concept of using a nanopore as a biosensor was first proposed in the mid 1990s when research into nanopores was beginning at academic institutions such as Oxford, Harvard and UCSC. Oxford Nanopore was founded in 2005 to translate academic nanopore research into a commercial, electronics-based sensing technology. The comprehensive end-to-end system includes sample preparation, molecular analysis and informatics, and is designed to provide novel benefits to a range of users for a broad number of applications.

Oxford Nanopore has a broad intellectual property portfolio that includes internal innovation and collaborations with world-leading nanopore researchers. This IP includes fundamental nanopore sensing techniques through to solid-state nanopore sensing technology, including graphene.

Nanopore fabrication
A nanopore is, essentially, a nano-scale hole. This hole may be:

- biological: formed by a pore-forming protein in a membrane such as a lipid bilayer;
- solid-state: formed in synthetic materials such as silicon nitride or graphene; or
- hybrid: formed by a pore-forming protein set in synthetic material.

Nanopore sensing
A nanopore may be used to identify a target analyte as follows: