

Graphs for Genomic Sequences

Raluca Uricaru

LaBRI, Univ. de Bordeaux



© Shannon May

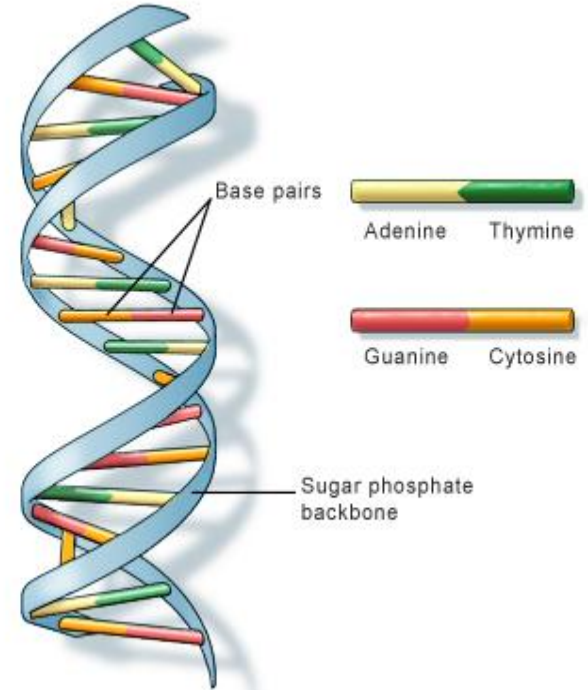
At the beginning ...

... there was the quest for understanding **THE GENOME**:

- **reading** the genome sequence (composed of 4 nucleotides A, T, G, C) followed by
- **understanding** the functions of each piece
- **classifying** the variations among genomes

HOW: *by reading genomes from different species and individuals and comparing them*

PROBLEM: *this is extremely difficult*



U.S. National Library of Medicine

The genome quest



culminated with the **Human Genome Project (HGP)** officially started in 1990

The human genome

“is a book with multiple uses; a history book - a narrative of the journey of our species through time. It's a shop manual, with an incredibly detailed blueprint for building every human cell. And it's a transformative textbook of medicine, with insights that will give health care providers immense new powers to treat, prevent and cure disease”

Francis Collins, director of the “National Human Genome Research Institute”

The human genome



The HGP published the first draft of the human genome in *Nature*, Feb. 2001

- 3 billion base pairs (90% percent complete)
- detailed information about the structure, organization and function of the complete set of human genes
- about 20,500 genes (significantly fewer than the previous estimates ranging from 50,000 to 140,000 genes)
- the full sequence was completed and published in April 2003.

The human genome



The HGP published the first draft of the human genome in February 15, 2001

- 3 billion base pairs (90% percent complete)
- detailed information on the organization and function of the complete genome
- significantly fewer than the previous estimates (30,000 to 140,000 genes)
- the full sequence was completed and published in April 2003.

From 1990 to 2001 : it took 11 years to get a draft

20 years: from HGP to 1000 Genomes Project

A global reference for human genetic variation

The 1000 Genomes Project Consortium

Nature volume 526, pages 68–74 (01 October 2015)

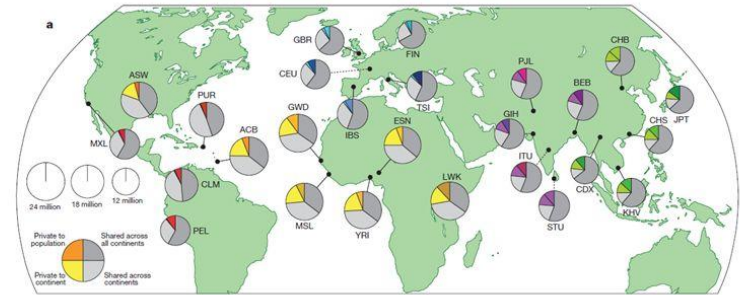
1000 Genomes Project

A global reference for human genetic variation

The 1000 Genomes Project Consortium*

OCTOBER 2015 | VOL 526 | NATURE

Phase III: 2,504 humans : 84.8 million SNPs



THE JACKSON LABORATORY

14

Different people ...



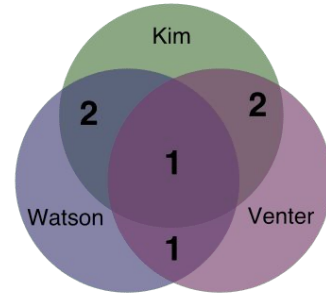
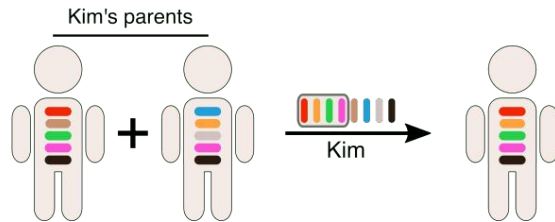
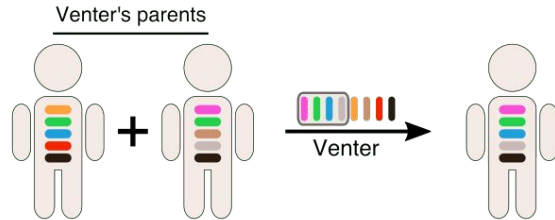
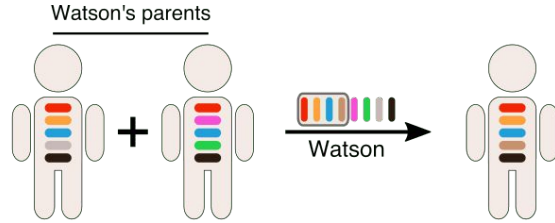
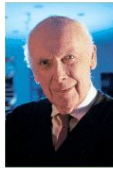
... slightly different genomes



The human genome book is written in a 4 - letter alphabet, books differing slightly between different people :

- roughly 1 difference in 1000 nucleotides in average, accounting from height to genetic diseases
a typical genome differs from the "reference human genome" in ~ 5 million places, ~ 99.9% identity
- lots of common variations, but also very rare ones
- variations ranging from 1 nucleotide (like SNPs) to tens of thousands nucleotides (structural variants)
- entire genes missing in some individuals (non-essential genes)

The “race” debate



Other large scale human sequencing projects



- **UK Biobank** : to decipher the genomes of 500,000 individuals
- **Iceland**'s effort to study the genomes of its entire human population.
- President Barack Obama's **Precision Medicine Initiative**, a genomics study of 1 million americans.

There's more than humans



- **I5K**: 5000 arthropodes
- **Genome 10K**: at least one individual from each vertebrate genus
- **Vertebrate Genomes Project** : find and sequence at least one individual from each of the approximately 66,000 vertebrate species
A Digital Noah's Ark Genome Library of Species
- announcement of the **Earth BioGenome Project (EBP)**
in Feb. 2017 at **BioGenomics- Global Biodiversity Genomics Conference**,
(Smithsonian National Museum of Natural History | Washington, D.C)

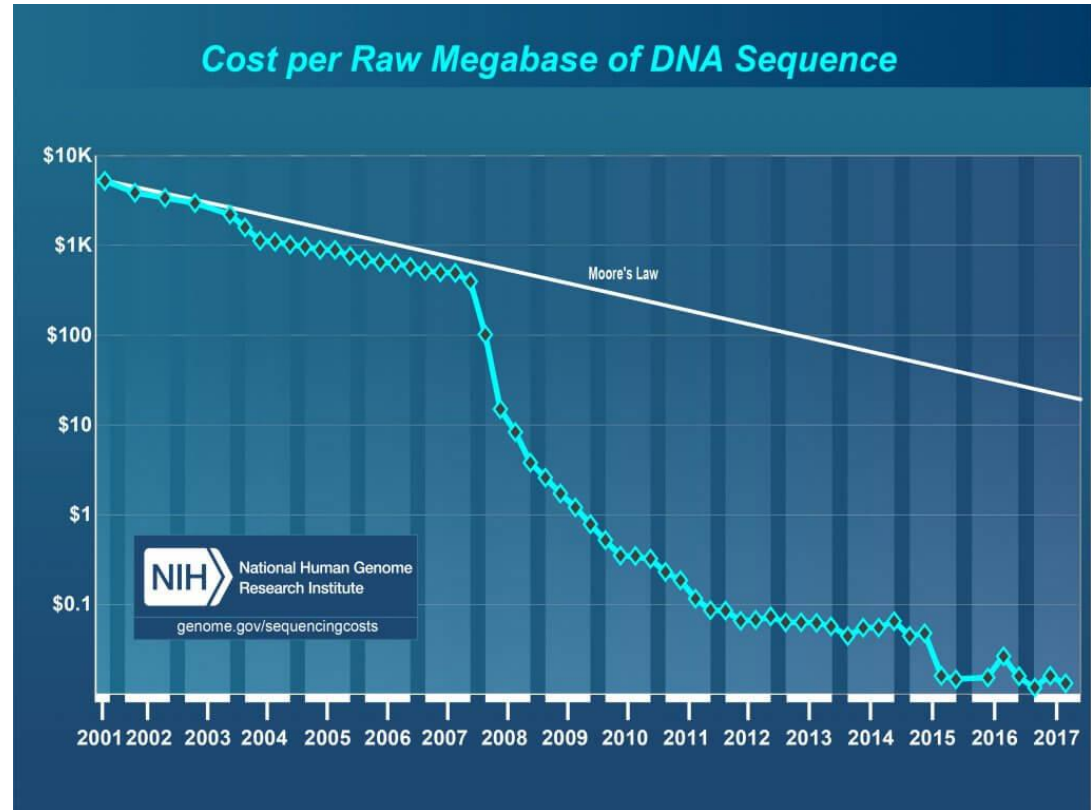
What made this sequencing projects possible

The evolution of sequencing costs

HGP \$3 billion

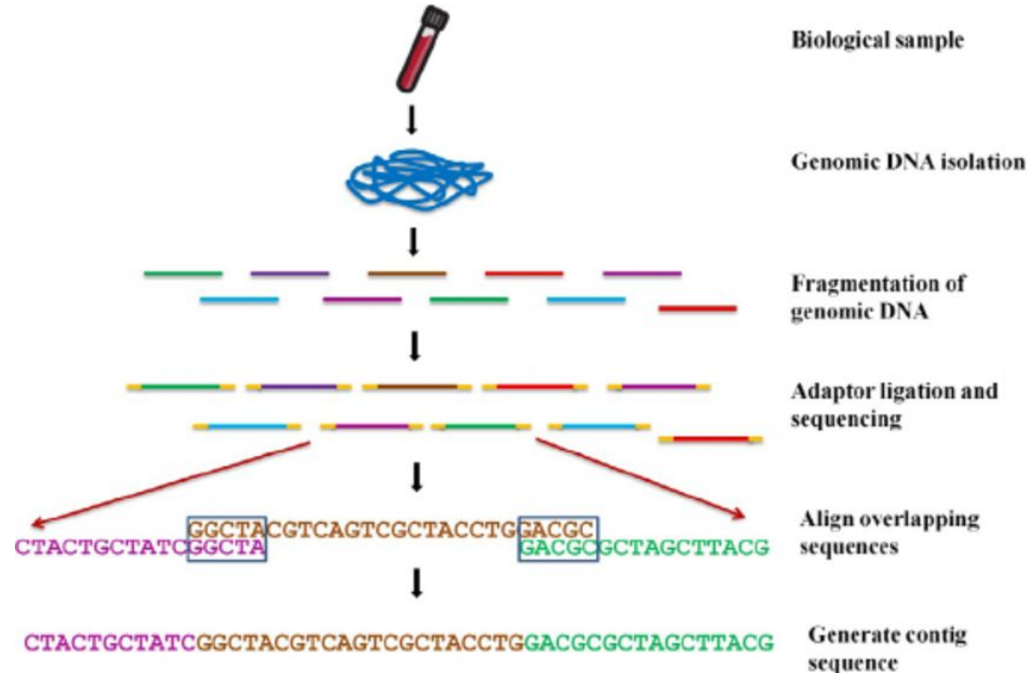
Illumina, early 2000:
\$10,000

Now, below \$1000



What accelerated the trend

The advent of Next Generation Sequencing



Practically, what brought the price of a genome down

- Better and cheaper sequencing machines
- Short genomic sequences (“reads”) are cheaper to produce
- Better computing machines
- **Better algorithms**



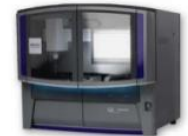
NGS means high sequencing capacity



GS FLX 454
(ROCHE)



HiSeq 2000
(ILLUMINA)



5500xl SOLiD
(ABI)



GS Junior



Ion TORRENT



Sequencing DNA and producing numerous DNA “texts” generated an increasing need for bioinformaticians in order to use: **algorithms**, **statistics**, and other **mathematical techniques**

to decipher the language of DNA



deciphering by comparison

The key is comparison


E.g. 2 insect genomes suspected to be somewhat related, evolutionarily speaking

- a fruit fly (*Drosophila melanogaster*) and
- a malaria mosquito (*Anopheles gambiae*)

We would like to know what parts of the fruit fly genomic sequence are dissimilar and what parts are similar to the mosquito genomic sequence ?

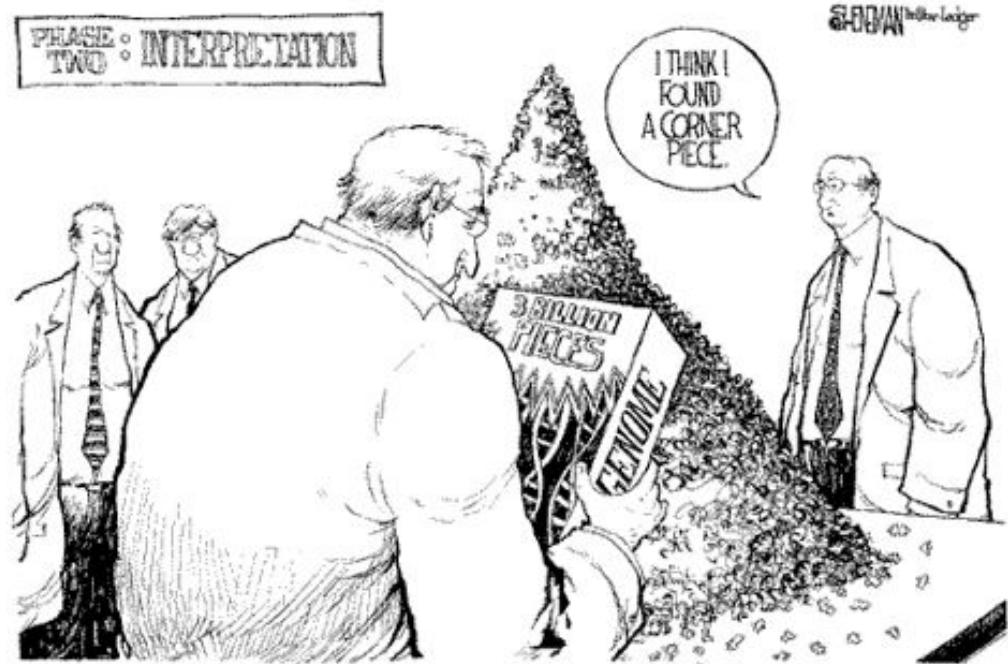


What do we do from here

- 
- **medicine**
 - **personalized medicine**
medical decisions based on a patient's predicted response or risk of disease
 - **therapeutic purposes**
*study of pathogenic genomes like variants of *E. coli*, the antibiotics resistant strains of *S. aureus*, ...*
 - **agriculture**
identification and manipulation of genes linked to specific phenotypic traits, breeding by marker-assisted selection of variants, ...
 - **understanding microbial communities through metagenomics**
with impact on earth sciences, biomedicine, bioenergy, biotechnology, ...
 - ...

How do we get from “reads” to complete sequences ?

Assembly Problem



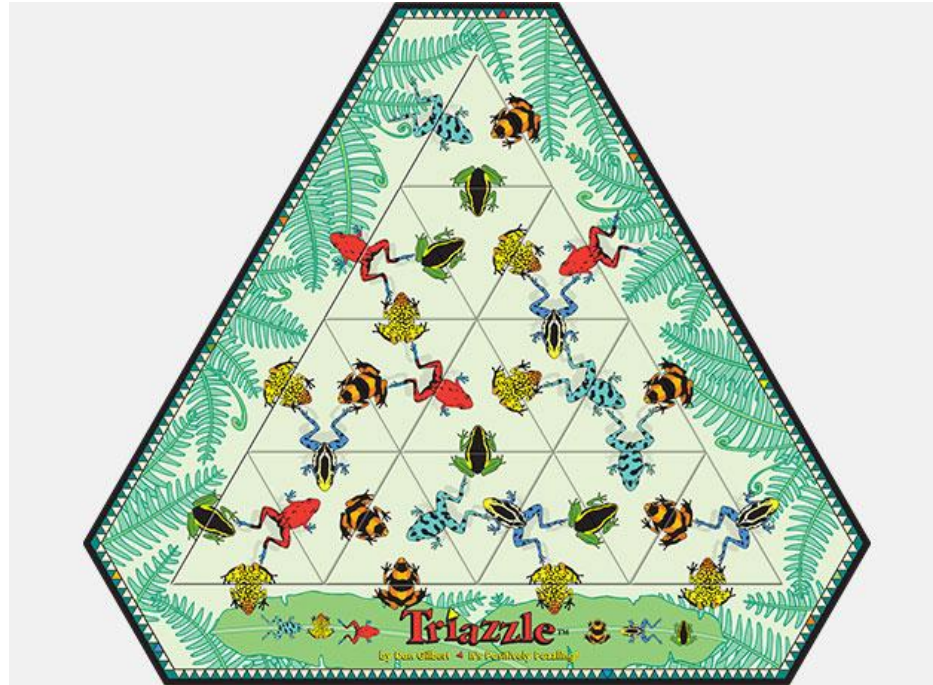
Assembly is an extremely hard problem

Genomes cannot be read as reading a book, they are read by small pieces.

- Though becoming longer nowadays, reads remain quite **short** w.r.t. genome lengths
- There are lots of pieces !!!
- Assembly is not like solving a jigsaw puzzle
 - reads may **have errors** (*up to 3%*)
 - **repeats** are frequent (*more than 50% of the human genome*)
 - **reads overlap**
 - pieces from the genome may be missing
 - the **reference may be different** from the actual sequence
 - or
 - the **reference may not even exist**

Assembly is an extremely hard problem

Not a jigsaw puzzle,
more like a *triazzle*



Assembly problem



Input : a set of reads that are **sub-strings** of the genome

Output : the genome sequence explaining all the reads

Toy example :

Input : {GAAG, AAGT, GTAG, TAGA}

Output : GAAGTAGA

Assembly problem



Toy example :

Input : {GAAG, AAGT, GTAG, TAGA}

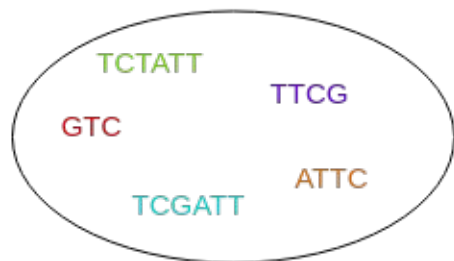
Output : GAAGTAGA

AAGTAGAGTAGAAG is also a solution

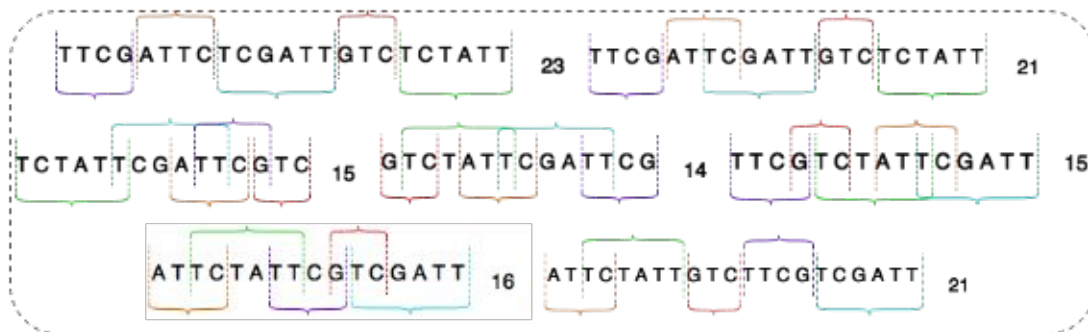
Which one is the best ?

Shortest does not necessarily mean best but fixing another criterion is difficult.

Shortest Common Superstring (SCS) problem



Superstrings



Shortest Superstring



Computing a SCS solution as a Hamiltonian path



Equivalent to finding a Hamiltonian path in an overlap graph :
visit every node in the graph exactly once

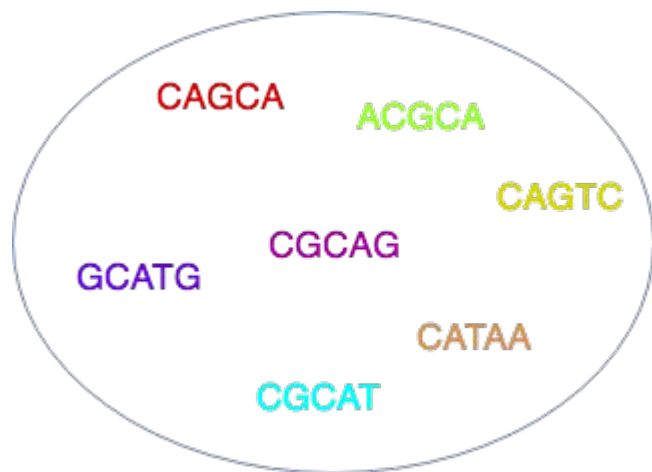
NP-hard problem

Approximation algorithms exist with factors : 4, 3, 2.89, 2.75, 2.5, 2.366, ...

The greedy method (4 approximation) :

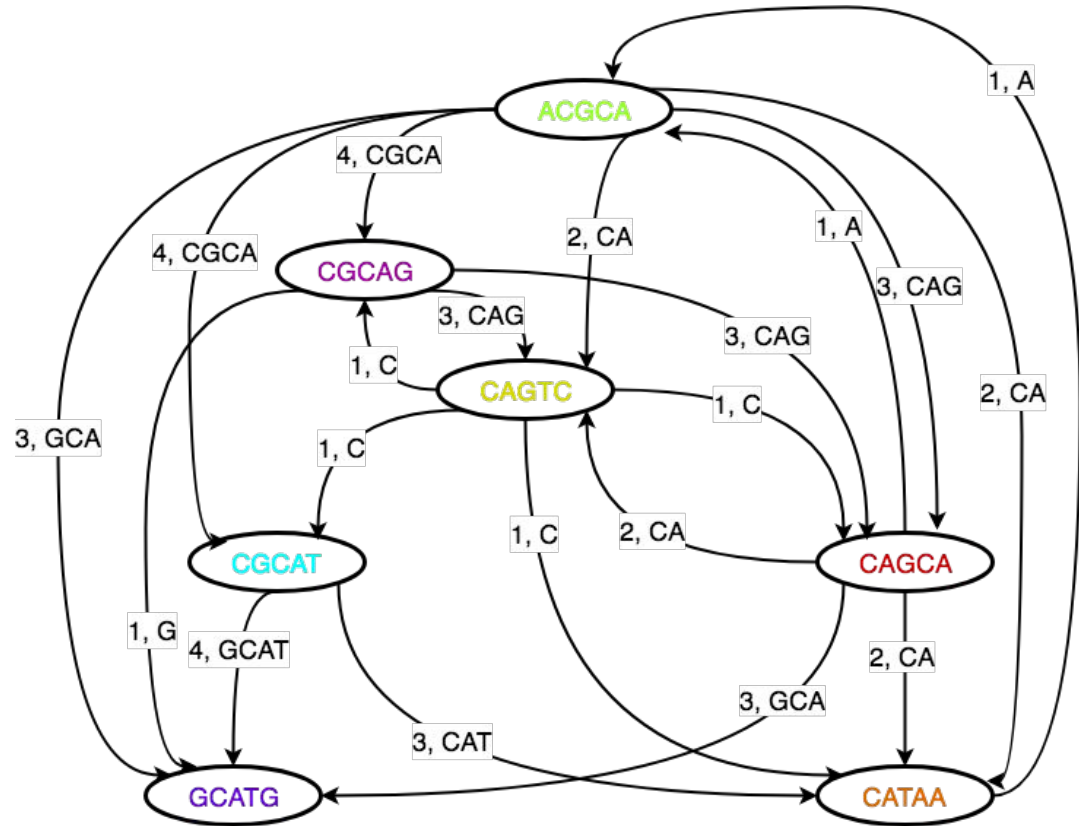
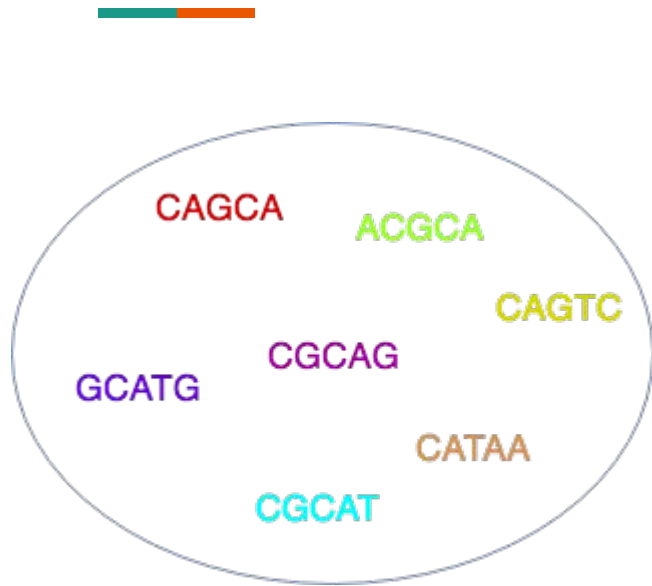
- finds pairs of strings that overlap the best
- merges them
- repeats the operation

A particular case : **r-SCS** problem

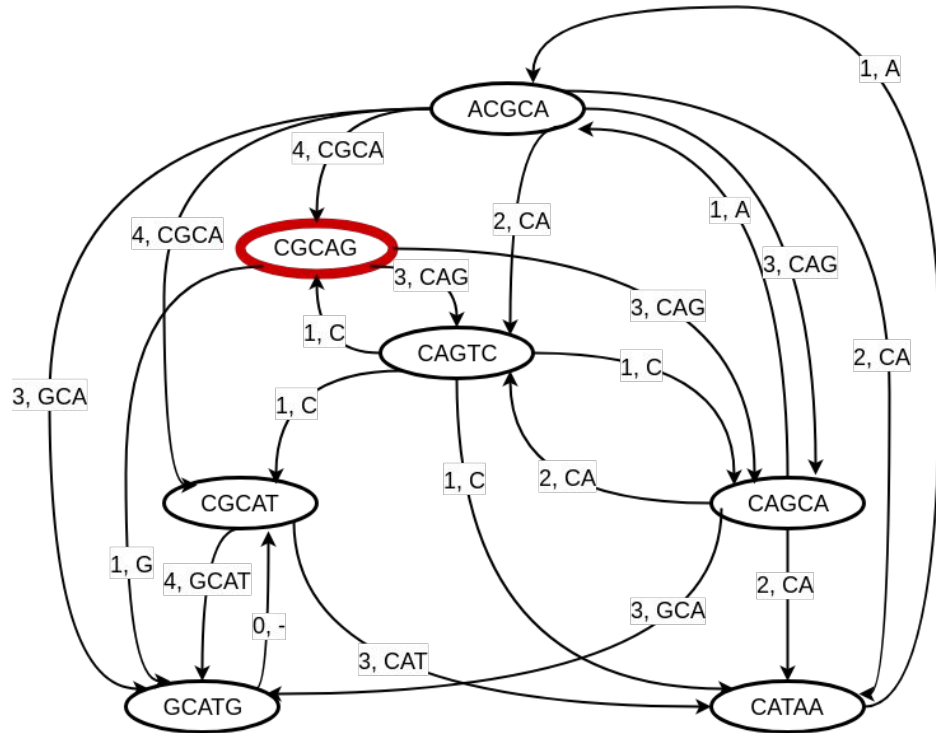


SCS problem applied on a set of strings having the **same length r** .

Overlap Graph built on a set SE



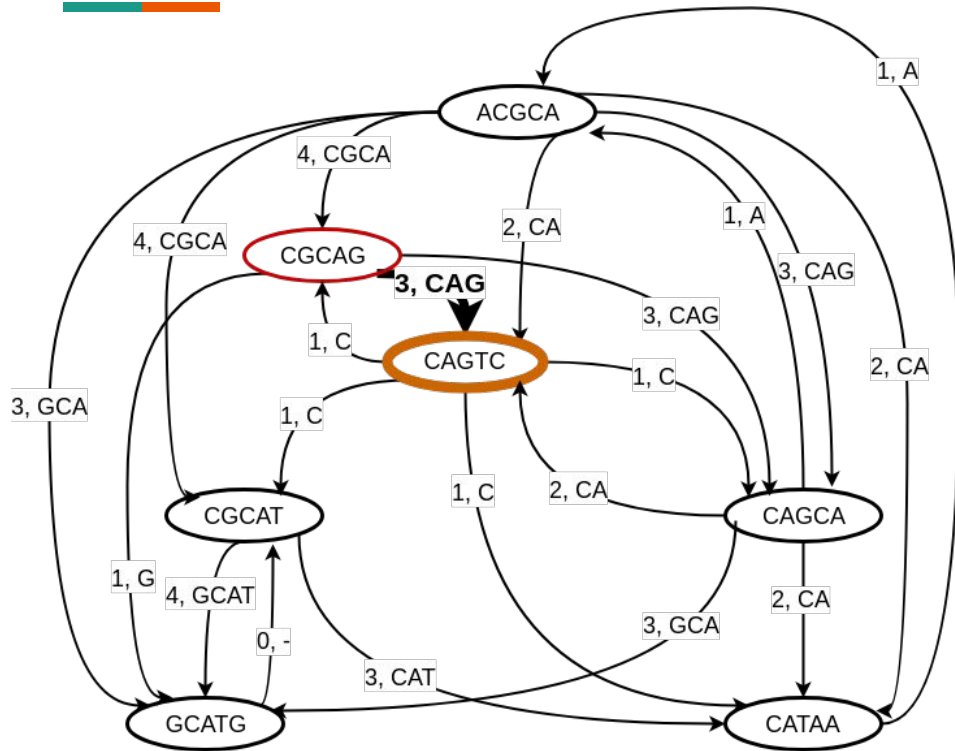
Superstring solution = Hamiltonian path



Superstring solution

$\| \text{CGCAG} \| = 5$

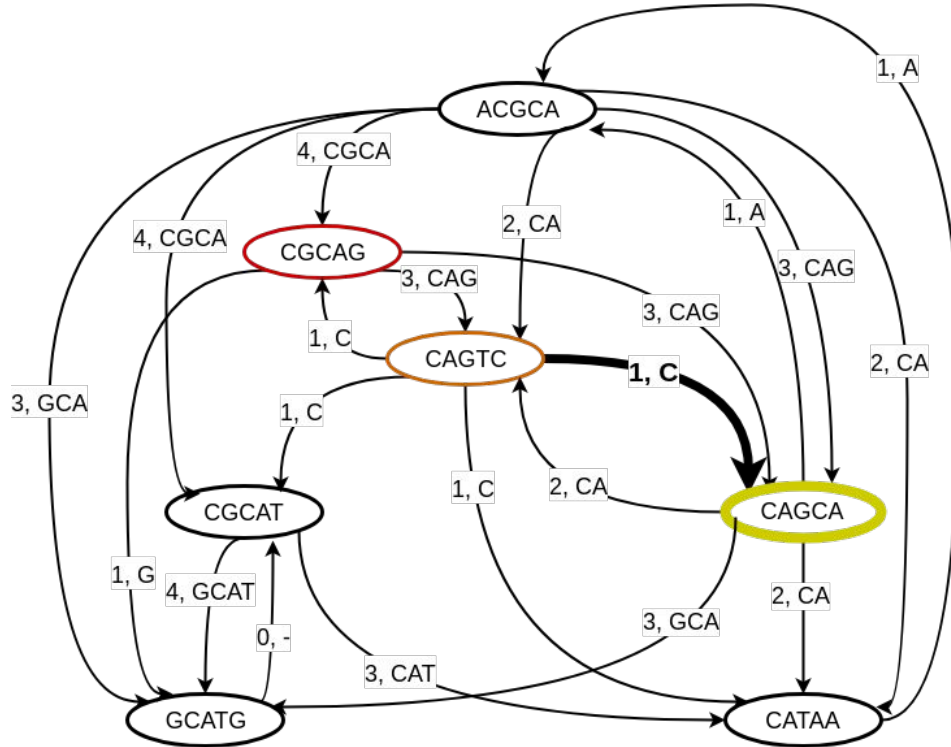
Superstring solution = Hamiltonian path



Superstring solution

$\| \text{CGCAGTC} \| = 7$

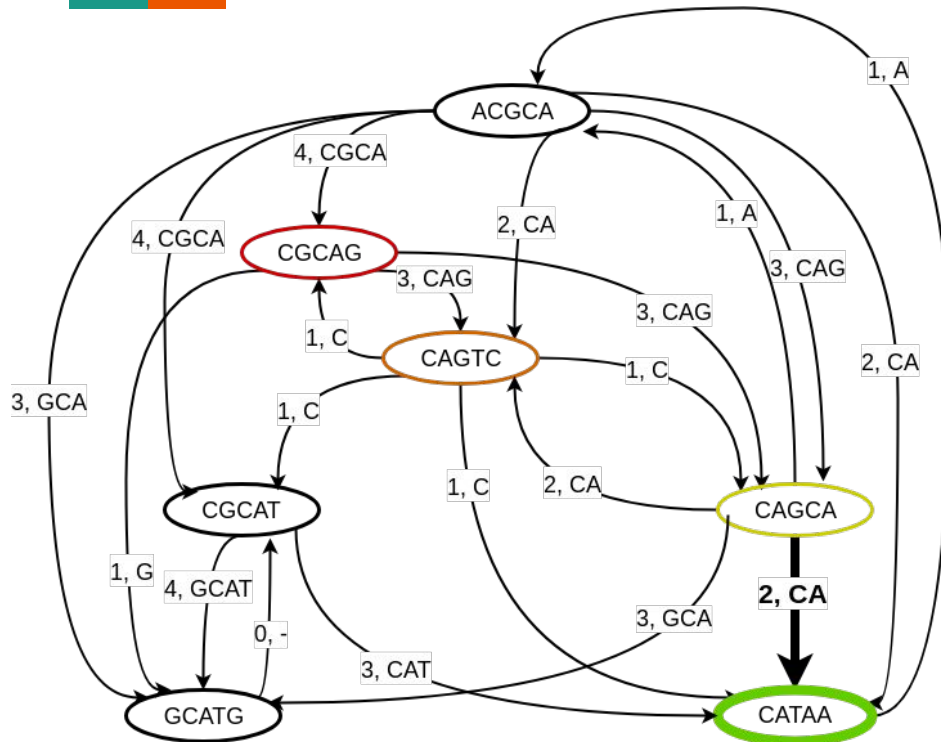
Superstring solution = Hamiltonian path



Superstring solution

|| CGCAGTCAGCA || = 11

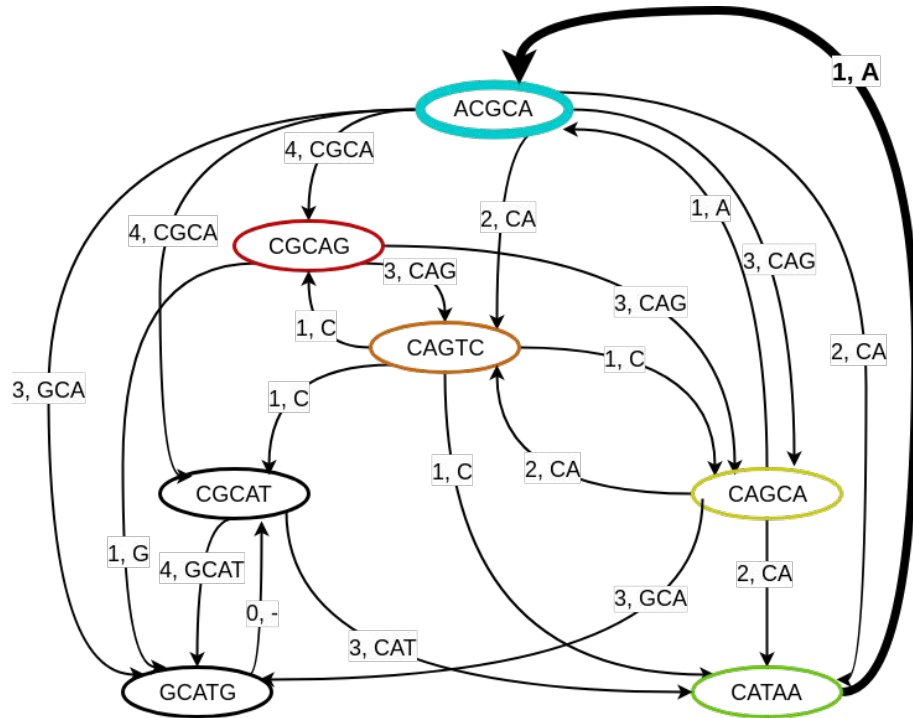
Superstring solution = Hamiltonian path



Superstring solution

|| CGCAGTCAGCATAA || = 14

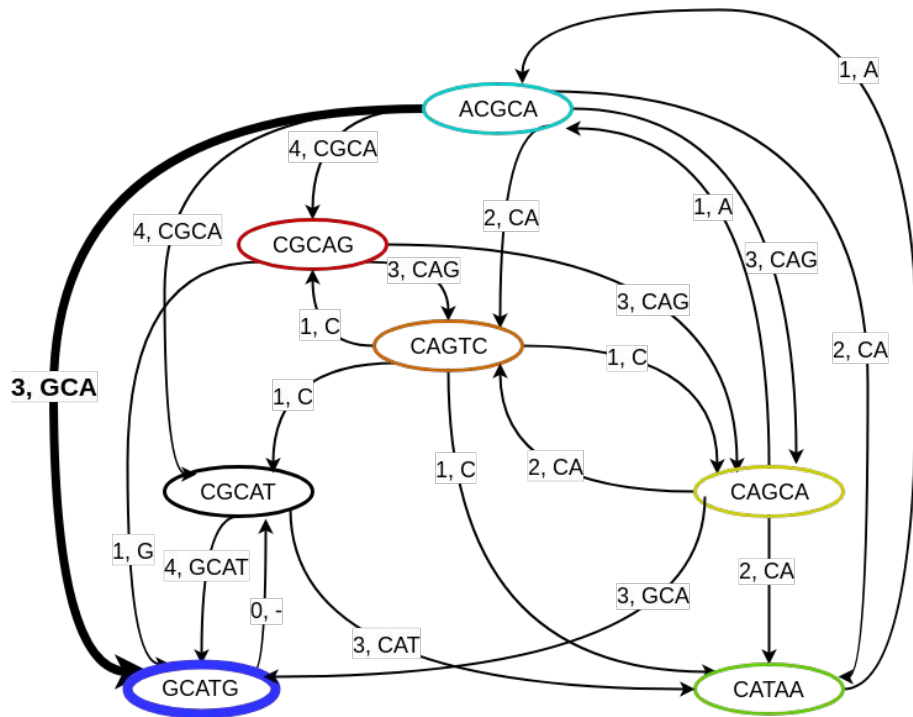
Superstring solution = Hamiltonian path



Superstring solution

|| CGCAGTCAGCATAACGCA || = 18

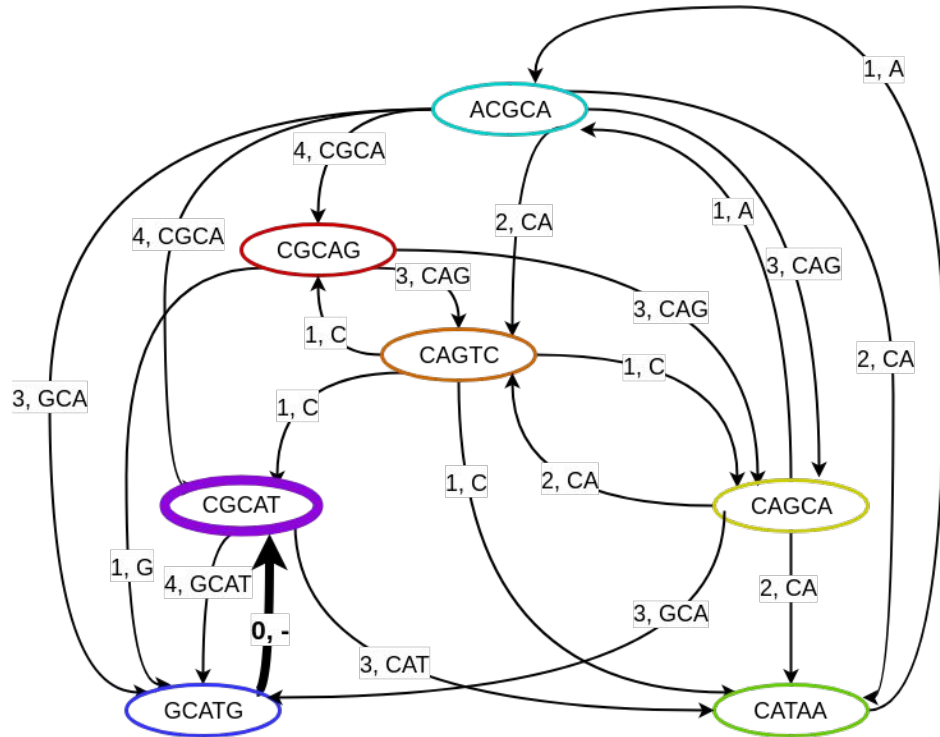
Superstring solution = Hamiltonian path



Superstring solution

||CGCAGTCAGCATAACGCATG|| = 20

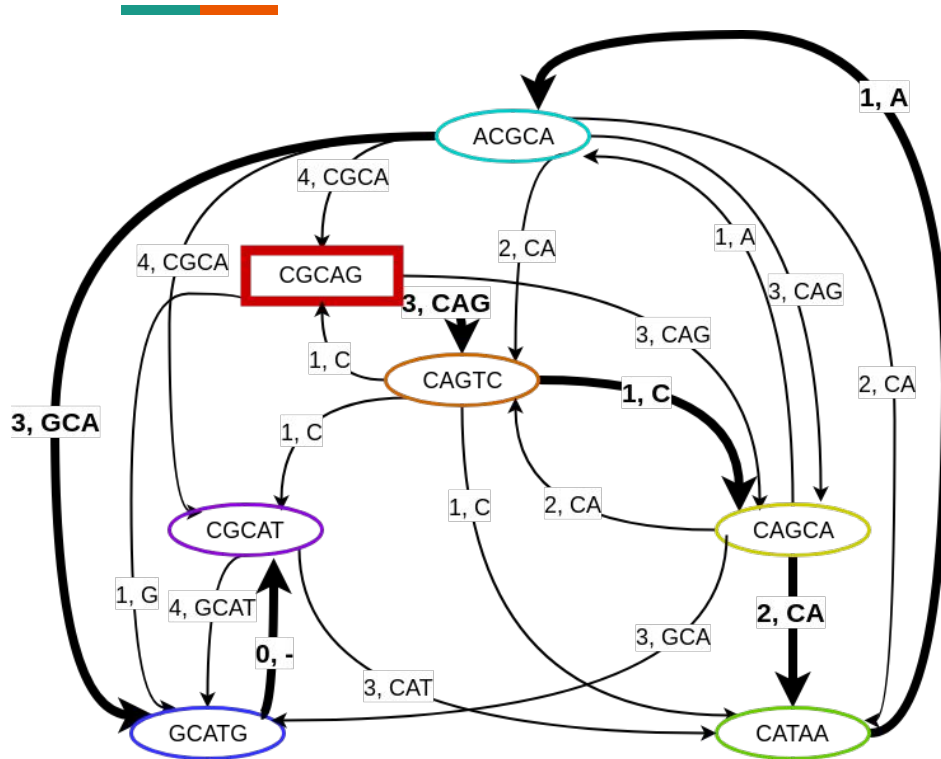
Superstring solution = Hamiltonian path



Superstring solution

`||CGCAGTCAGCATAACGCATGCGCAT|| = 25`

Superstring solution = Hamiltonian path

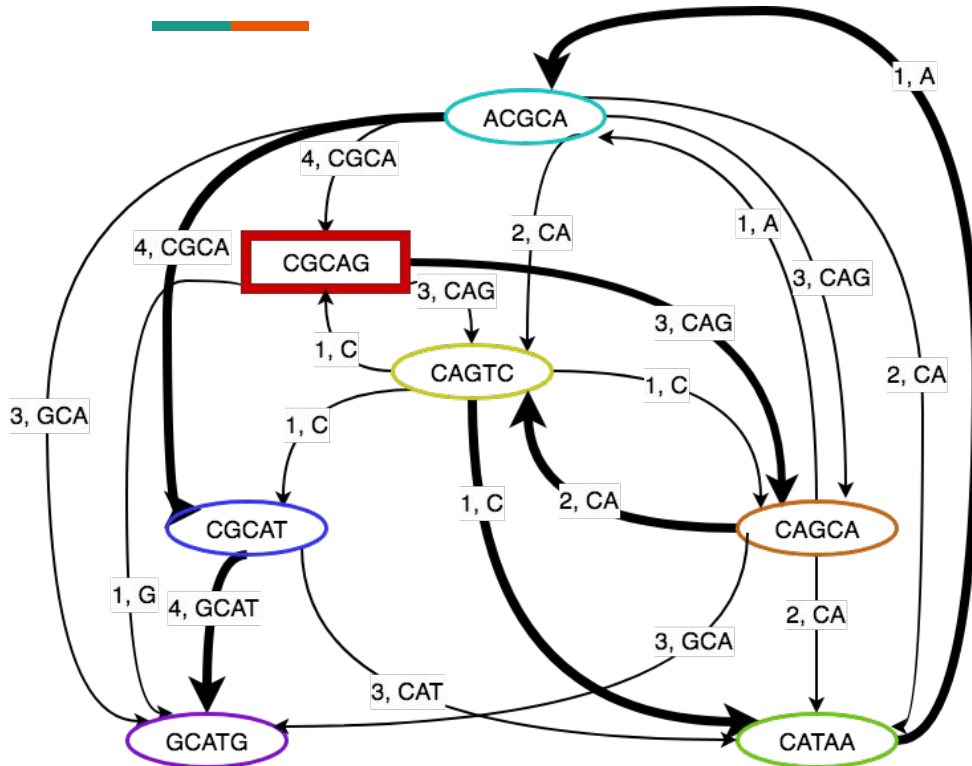


Superstring solution

`||CGCAGTCAGCATAACGCATGCG
CAT|| = 25`

= max-length superstring solution (35)
- hamiltonian path length (10)

SCS solution = heaviest Hamiltonian path



Shortest superstring solution

$||\text{CGCAGCAGTCATAACGCATG}|| = 20$

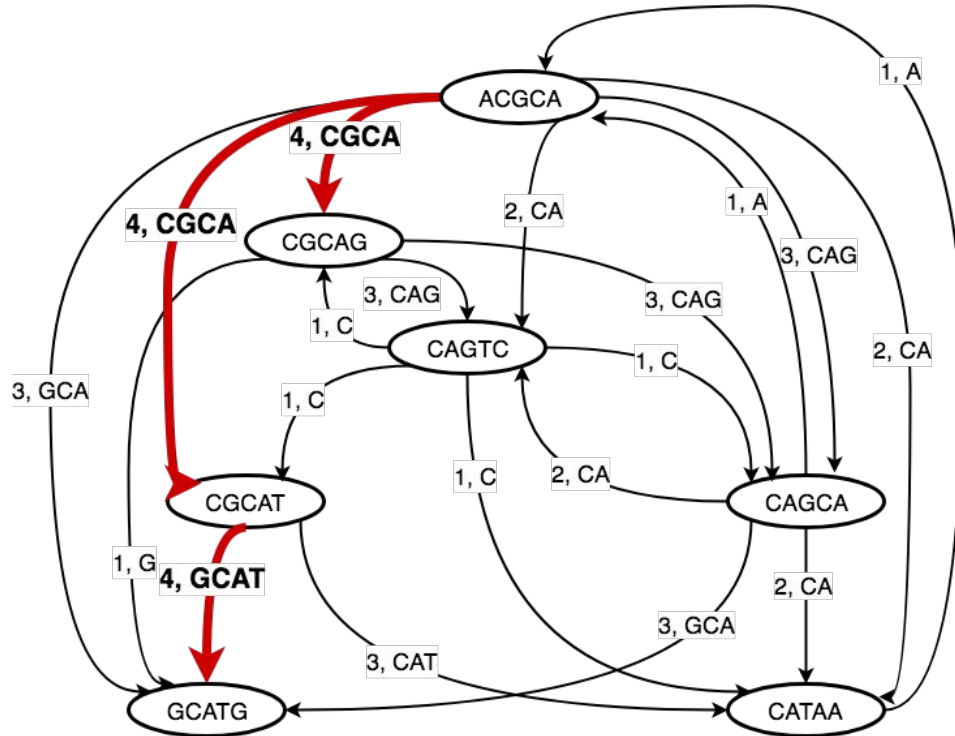
= max-length superstring solution (35) -
hamiltonian path length (15)

MAX - ATSP problem




On the approximation ratio of the r-SCS problem

r-SCS problem : Golovnev et al. solution

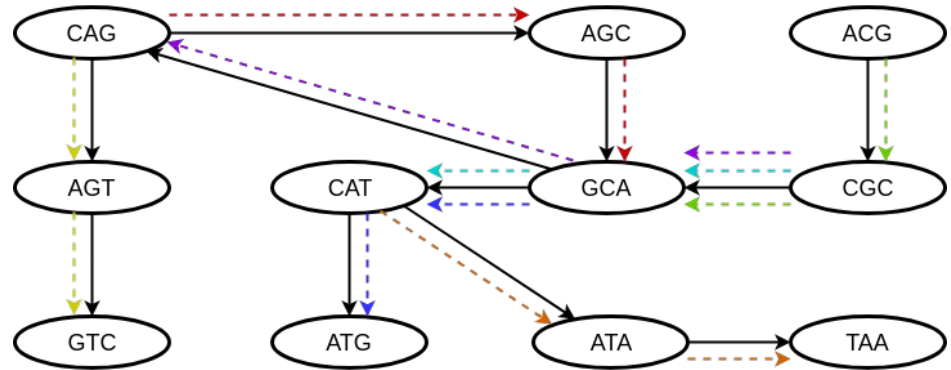
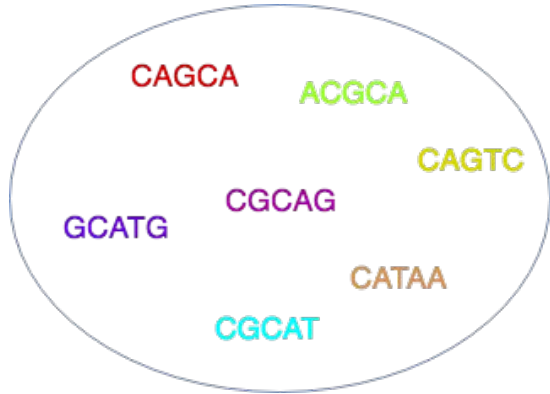


Simple idea : focus on the best overlaps ($r-1$ overlaps)

r-SCS problem : Golovnev et al.

- 
1. Build the de Bruijn graph $\text{dB}(r - 1)$ on S and transpose the r -SCS problem into a 2-SCS instance.
 2. Solve the 2-SCS instance with the algorithm from *Crochemore et al.*
 3. Output the corresponding superstring solution for the original r -SCS problem.


de Bruijn Graph for $k = 3$



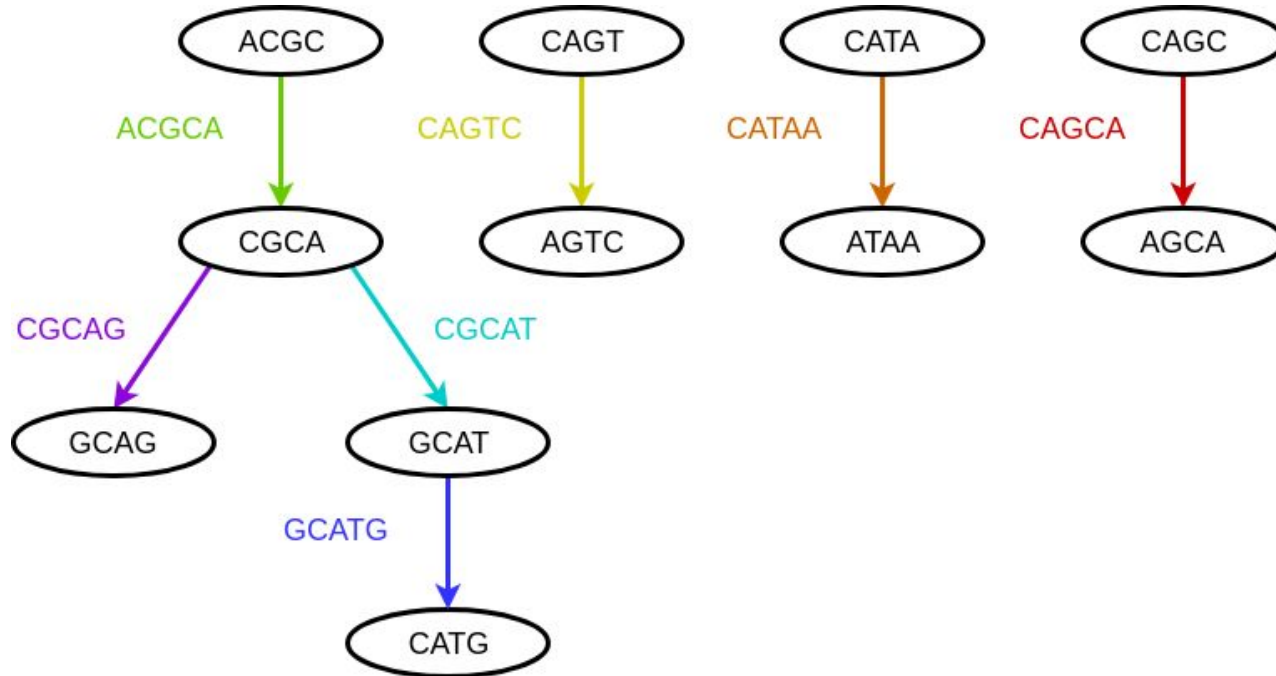
de Bruijn Graph

- overlap graph built on all the k -length strings present in the reads
- Nodes = k -mers and edges = $(k-1)$ -overlaps

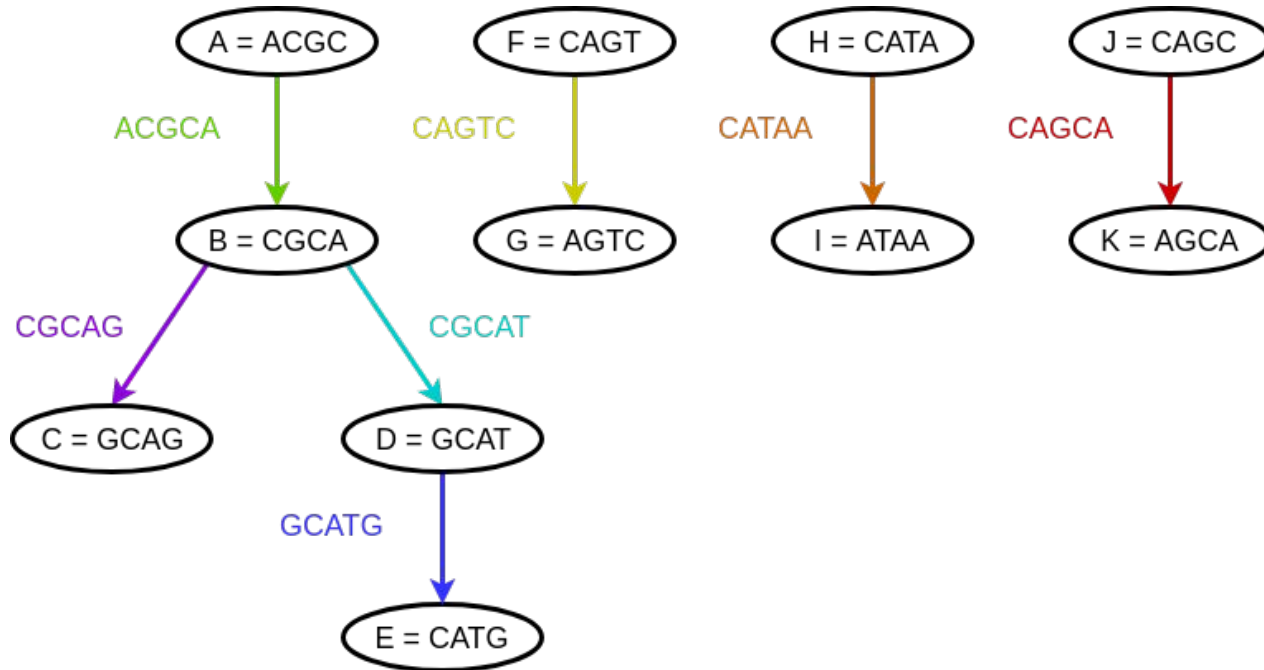
r-SCS problem : Golovnev et al.

- 
1. Build the de Bruijn graph $dB(r - 1)$ on S and transpose the r-SCS problem into a 2-SCS instance.
 2. Solve the 2-SCS instance with the algorithm from Crochemore *et al.*
 3. Output the corresponding superstring solution for the original r-SCS problem.

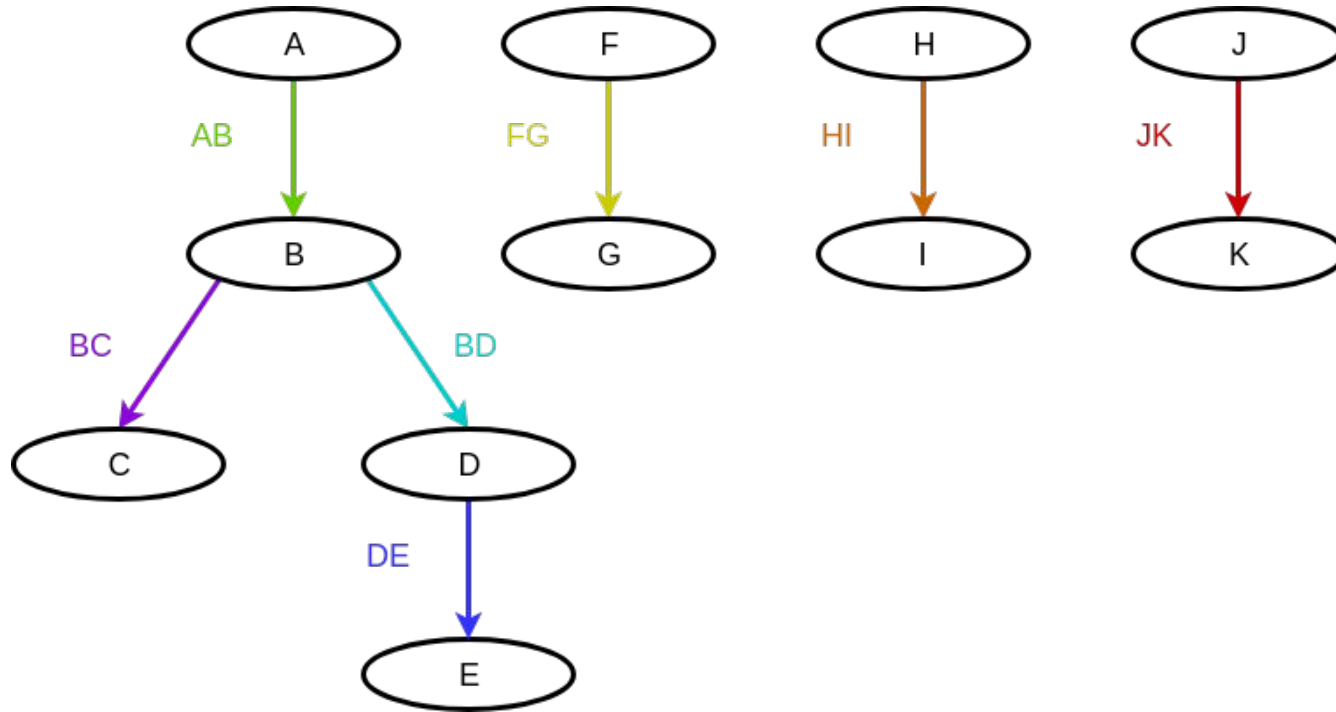
Build a de Bruijn Graph




r-SCS to 2-SCS



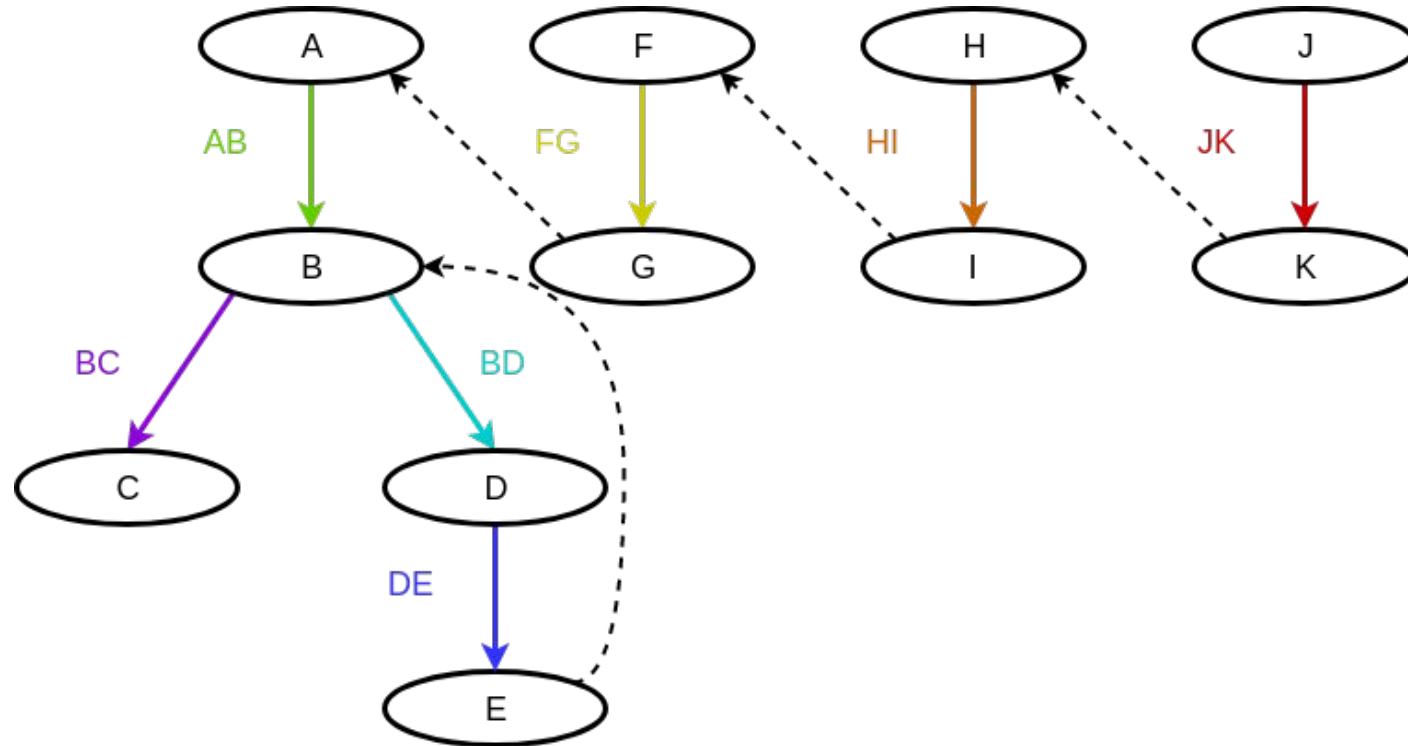
From r-SCS to 2-SCS




r-SCS problem on S : Golovnev et al.

- 
1. Build the de Bruijn graph $dB(r - 1)$ on S and transpose the r-SCS problem into a 2-SCS instance.
 2. Solve the 2-SCS instance with the algorithm from Crochemore *et al.*
 3. Output the corresponding superstring solution for the original r-SCS problem.

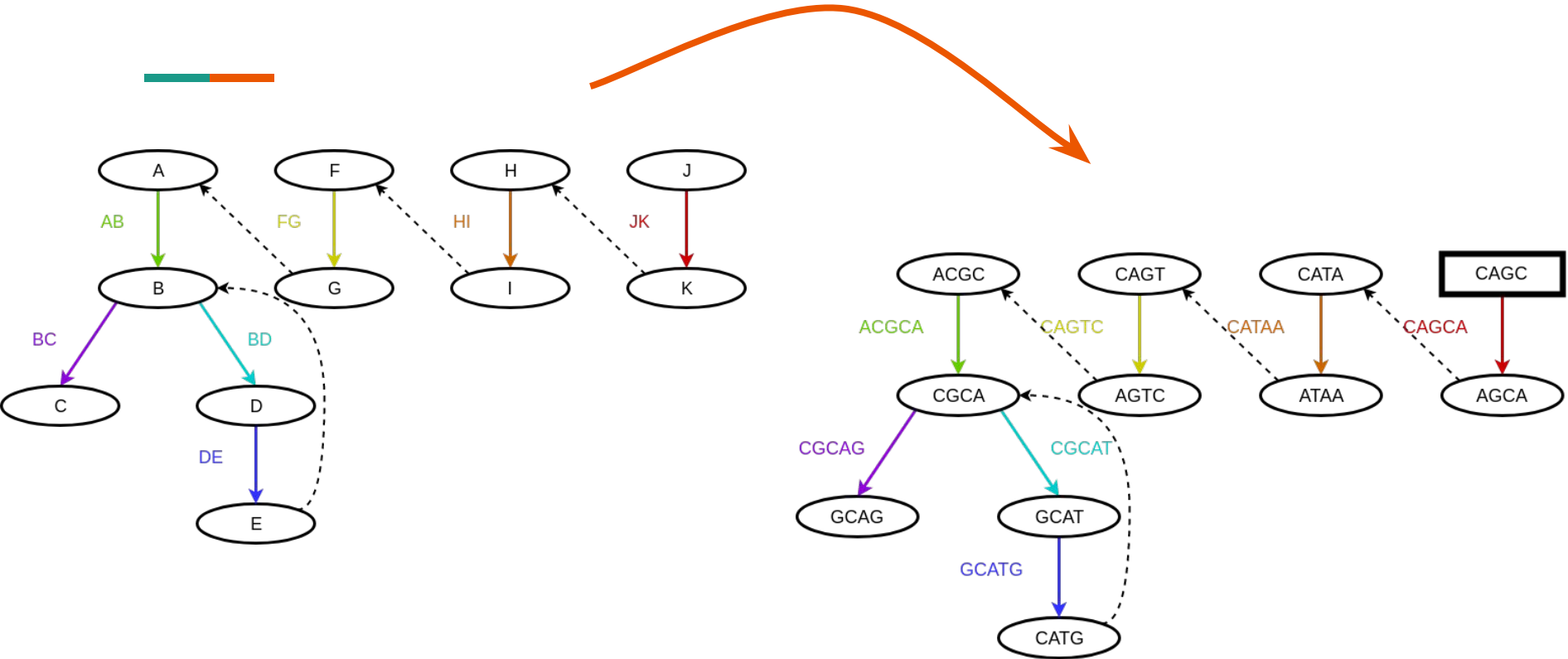
Solve 2-SCS : eulerian path with minimal additional edges



r-SCS problem on S : Golovnev et al.

- 
1. Build the de Bruijn graph $dB(r - 1)$ on S and transpose the r-SCS problem into a 2-SCS instance.
 2. Solve the 2-SCS instance with the algorithm from Crochemore *et al.*
 3. Output the corresponding superstring solution for the original r-SCS problem.

From 2-SCS back to r-SCS



CAGCACATAACAGTCACGCATGCGCAG

Golovnev et al. - approximation ratio

With $x = \frac{w(H)}{n}$ we get the following ratios :

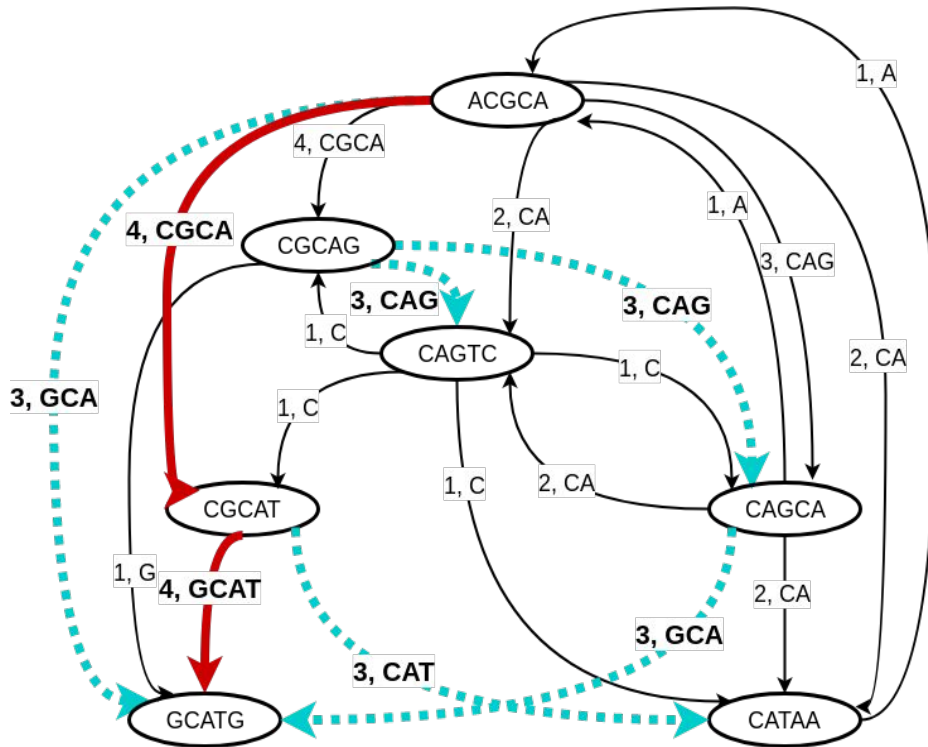
2-SCS based method : $\frac{(r^2 - 2r + 2) - (r - 1)x}{r - x}$

MAX-ATSP

global ratio : $\alpha(r) = \max_{0 \leq x \leq r-1} \left\{ \min \left\{ \frac{(r^2 - 2r + 2) - (r - 1)x}{r - x}, \frac{r - \frac{2}{3}x}{r - x} \right\} \right\}$.

Better than the general best SCS ratio (2.366) for $r < 7$

r-SCS problem : a hierarchical solution



Extend the idea of Golovnev et al. :

the best overlaps (r-1 overlaps)

+

the second best (r-2 overlaps)

r-SCS problem : a hierarchical solution

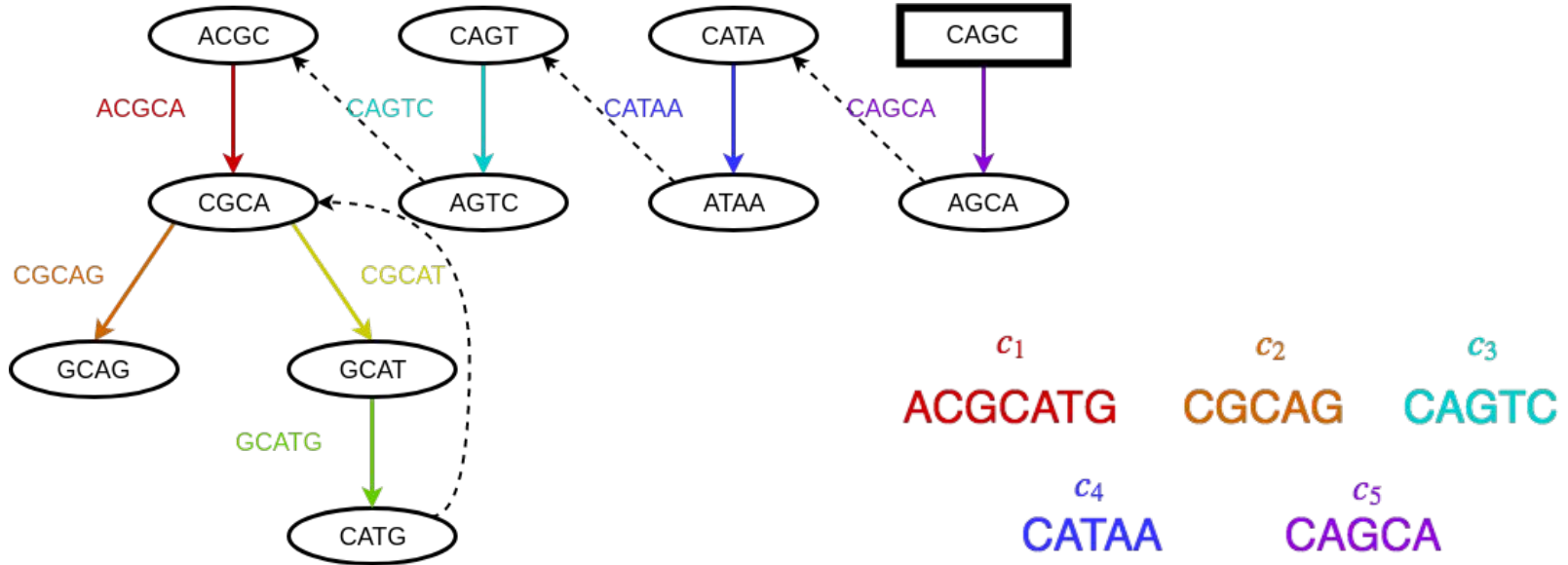
Step 1

1. Build the de Bruijn graph $dB(r - 1)$ on S and transpose the r -SCS problem into a 2-SCS instance.
2. Solve the 2-SCS instance with the algorithm from Crochemore *et al.*


Step 2

3. Build a set of contigs S' by removing the edges added by the eulerian procedure; build a de Bruijn graph $dB(r - 2)$ on the $(r-2)$ -affixes of S' and transpose it into a 2-SCS instance.
4. Solve the novel 2-SCS instance and output the corresponding superstring solution (named γ) for the original r -SCS problem.

r-SCS hierarchical solution : Step 1 (Golovnev et al.)

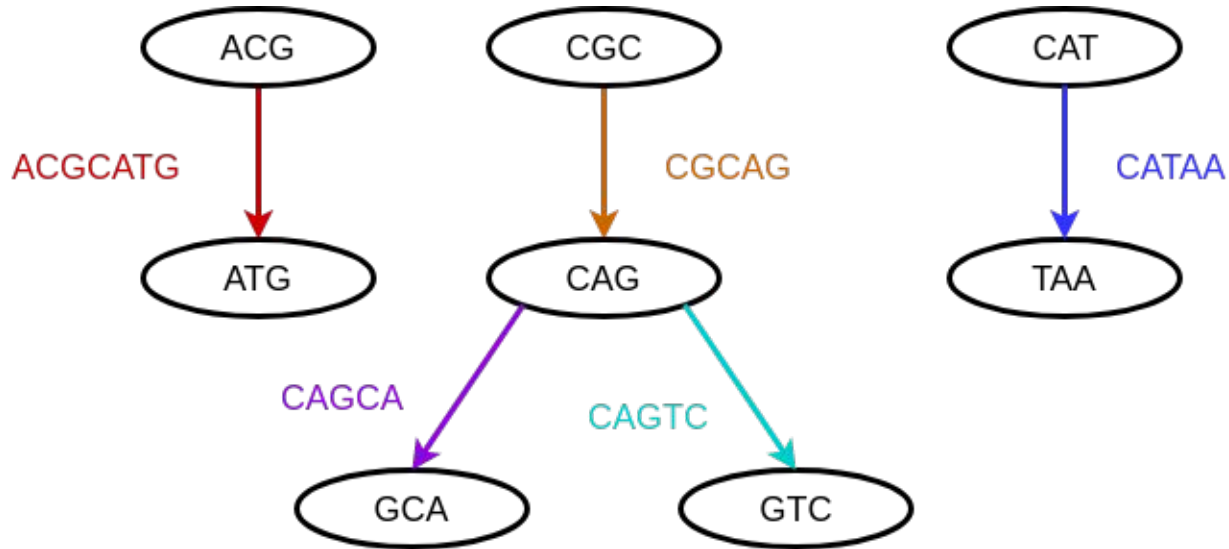


r-SCS problem : a hierarchical solution


- 
1. Build the de Bruijn graph $dB(r - 1)$ on S and transpose the r-SCS problem into a 2-SCS instance.
 2. Solve the 2-SCS instance with the algorithm from Crochemore *et al.*
 3. Build a set of contigs S' by removing the edges added by the eulerian procedure; build a de Bruijn graph $dB(r - 2)$ on the $(r-2)$ -affixes of S' and transpose it into a 2-SCS instance.
 4. Solve the novel 2-SCS instance and output the corresponding superstring solution (named γ) for the original r-SCS problem.

Step 2

r-SCS hierarchical solution : Step 2 - build dB (r-2)

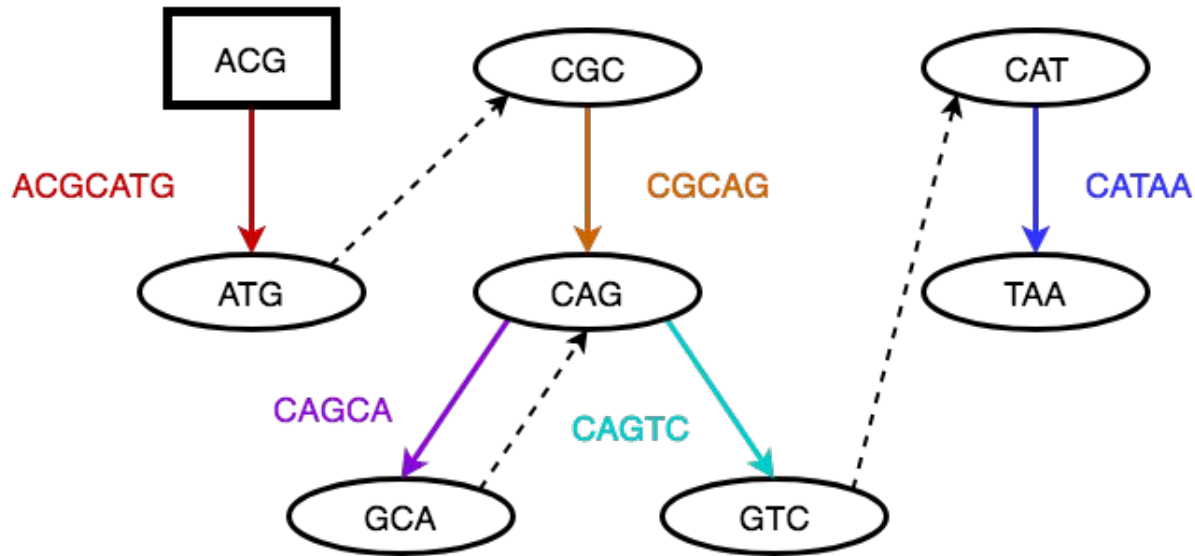


r-SCS problem : a hierarchical solution

- 
1. Build the de Bruijn graph $dB(r - 1)$ on S and transpose the r-SCS problem into a 2-SCS instance.
 2. Solve the 2-SCS instance with the algorithm from Crochemore *et al.*
 3. Build a set of contigs S' by removing the edges added by the eulerian procedure; build a de Bruijn graph $dB(r - 2)$ on the $(r-2)$ -affixes of S' and transpose it into a 2-SCS instance.
 4. Solve the novel 2-SCS instance and output the corresponding superstring solution (named γ) for the original r-SCS problem.

Step 2

r-SCS hierarchical solution : Step 2 - solve 2-SCS and compute γ



ACGCATGCGCAGCACAGTCCATAA

2-steps hierarchical solution - approximation ratio

With $x = \frac{w(H)}{n}$ we get the following ratios :

2-SCS based method

2-steps hierarchical method

$$\frac{(r^2 - 2r + 2) - (r - 1)x}{r - x}$$

MAX-ATSP

Same global
ratio :

$$\alpha(r) = \max_{0 \leq x \leq r-1} \left\{ \min \left\{ \frac{(r^2 - 2r + 2) - (r - 1)x}{r - x}, \frac{r - \frac{2}{3}x}{r - x} \right\} \right\} .$$

Better than the general best SCS ratio (2.366) for $r < 7$

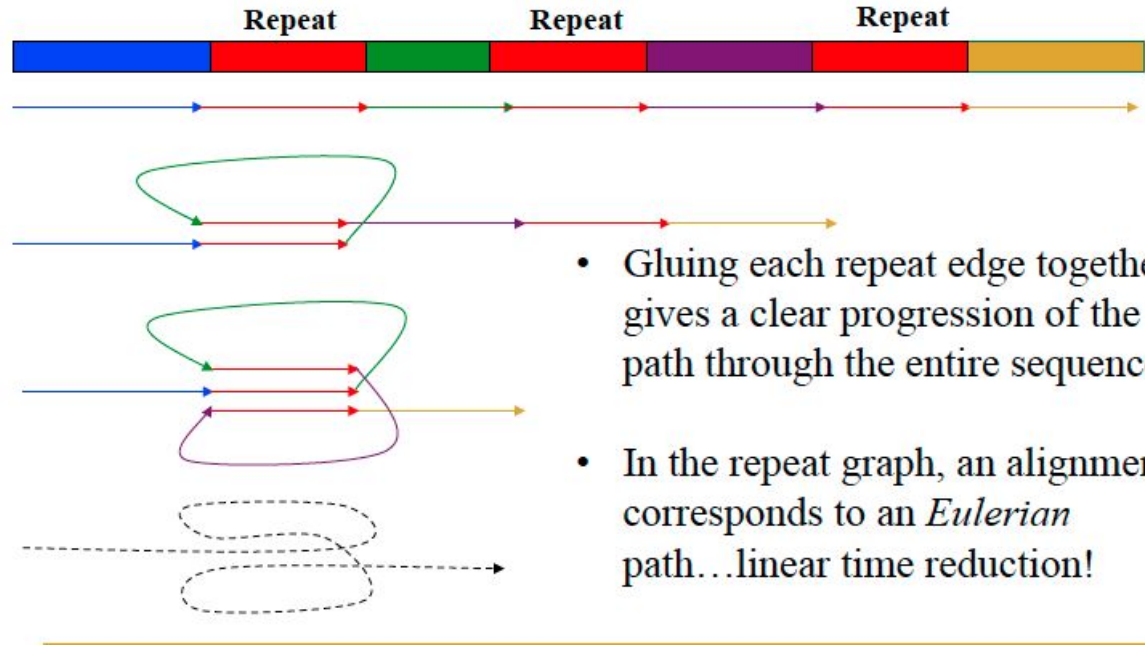
SCS solution as an Eulerian path



Hamiltonian path approach - HARD

Alternative :
Eulerian cycle formulation
on de Bruijn Graphs

easier to compute - linear time on perfect data



No speech without the buzzword **Big Data**



Eulerian path is easier to compute but **extremely large number of solutions** and either way :

Graphs are long to build and extremely large to store
hundreds of GB of data (hundreds of millions to billions of reads)

*It's not **BIG DATA**, it's **VERY BIG DATA***



Clever solution for space-efficient de Bruijn graph representation

© Rizk & Chikhi 2013

Bloom Filter



Memory usage for $k = 32$, $N = 2 * 10^9$, $E = 4 * 10^9$

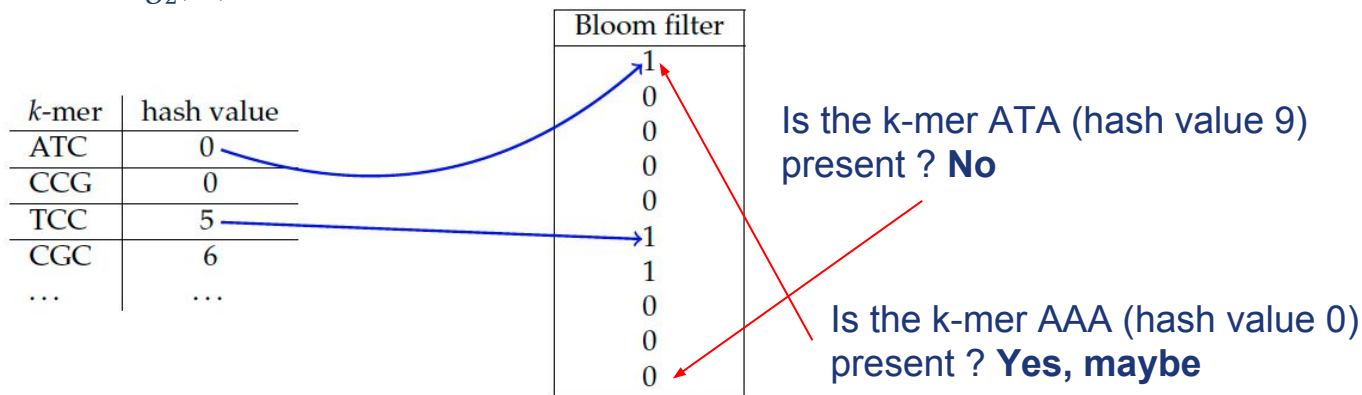
- Ascii sequence : 32B / node + 8B / edge (C pointer) **80 GB**
- 2 bit / nt, 8 possible edges : (8 + 1)B / node [Z. Iqbal, 2012] **18 GB**
- **Edges can be inferred !**
- Nodes only : 8B / node **16 GB**
- **Self information of n nodes** [Conway, Bromage, 2011]
• 20 bits per node **5 GB**
lower limit ???

Encoding de Bruijn graphs

Bloom filter

Bit array representing a set with a “precision“ of ϵ .
a proportion ϵ of elements will be wrongly included
false positives

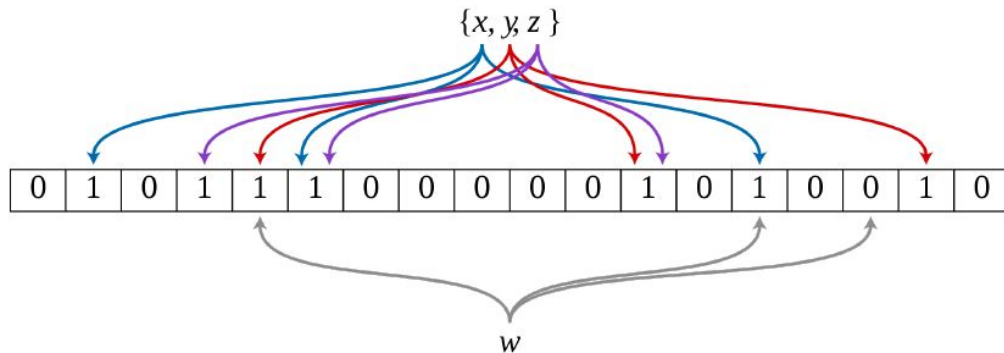
n elements : $1.44 \log_2(1/\epsilon) * n$ bits



Bloom filter

Hash function

- should have good “repartition” properties
- use of several functions to reduce false positive rate
 - insert n elements in a m bit array : ratio $r = m/n$
 - the FPR may be computed with respect to r

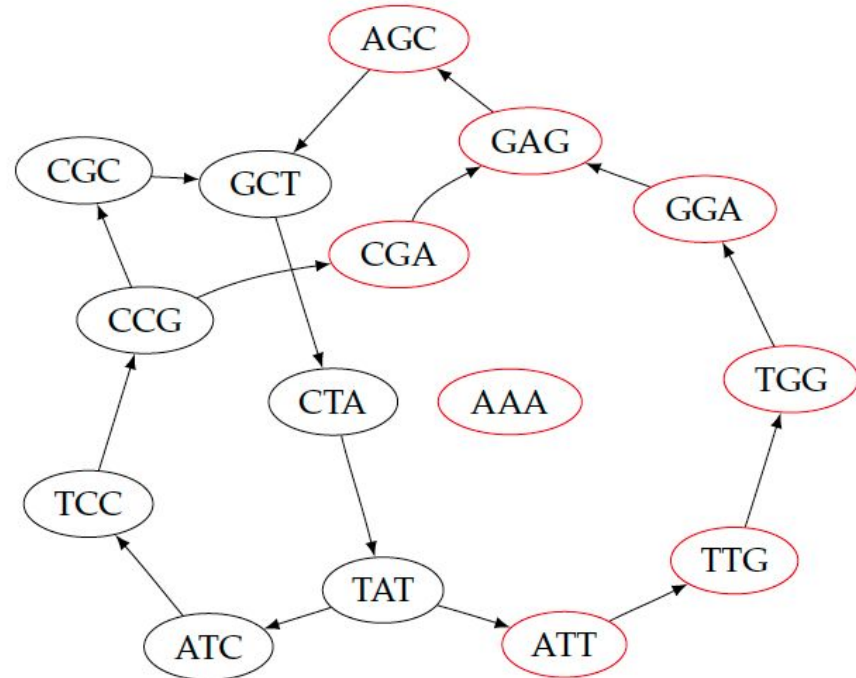


Bloom Filter



Set of nodes : {TAT, ATC, CGC, CTA, CCG, TCG, GCT}

de Bruijn graph as stored in a Bloom filter [Pell et. al. 2012]

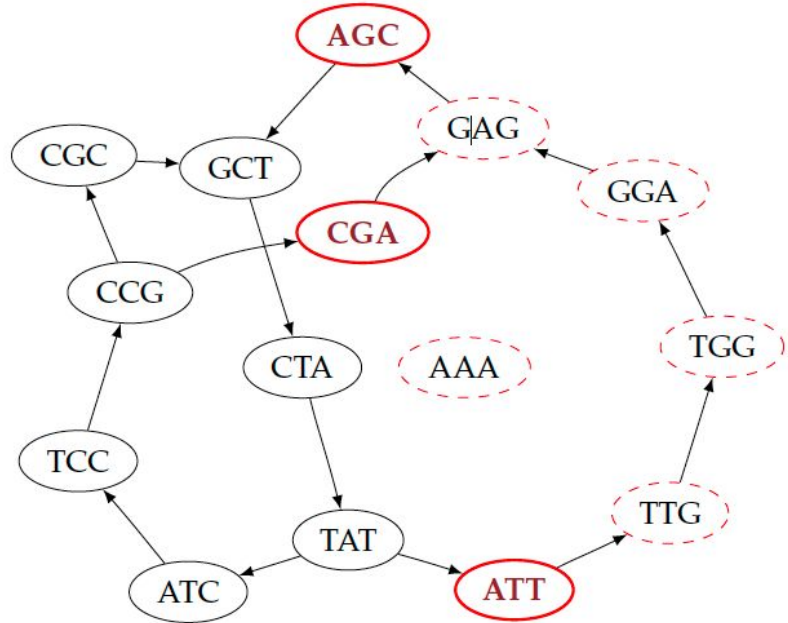


Black nodes : true positives ; **Red nodes :** false positives

Bloom Filter



To traverse the graph from true positive nodes, only a small fraction of the false positives need to be avoided (**critical FP**).



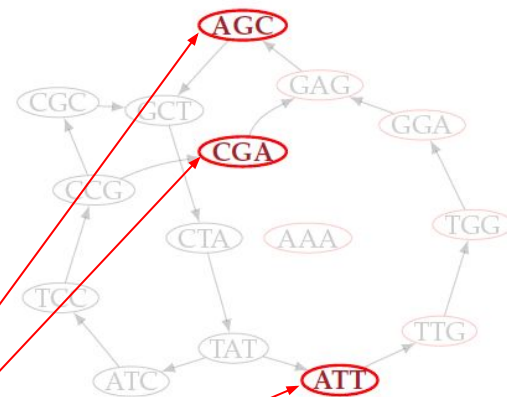
Minia : De Bruijn representation by Chikhi & Rizk, 2013



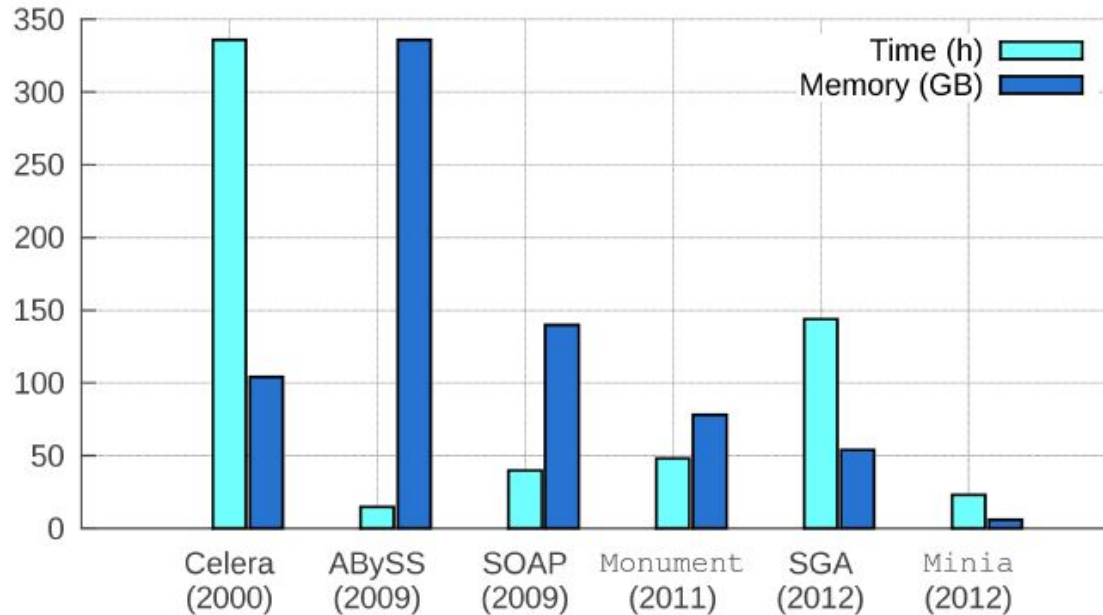
Store the **Bloom Filter** + **critical FPs** explicitly in memory

- both efficient and exact
 - self - information **30 bits**
- vs
Minia $10 + 3 * 6 = 28$ bits

Bloom filter
1
0
0
0
0
1
1
0
0
0
0



Minia : rough performance comparison on human data



Minia : Besides genomic sequences assembly

Numerous applications and tools based on Minia :

- Metagenomic and transcriptome assembly
- Mutation detection
- Structural variant detection
- Targeted assembly
- K-mer counting
- Read error correction
- Read compression

Commet, Simka

KisSplice, DiscoSNP

TakeABreak, MindTheGap

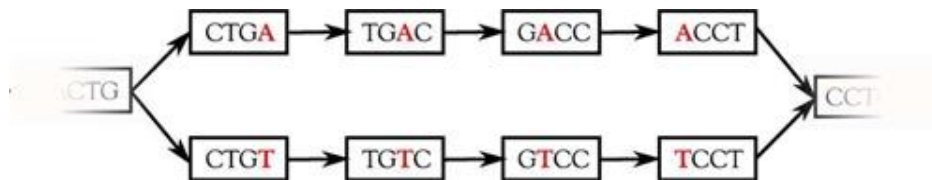
Mapsembler

DSK

Lordec, Bloocoo, LoRMA

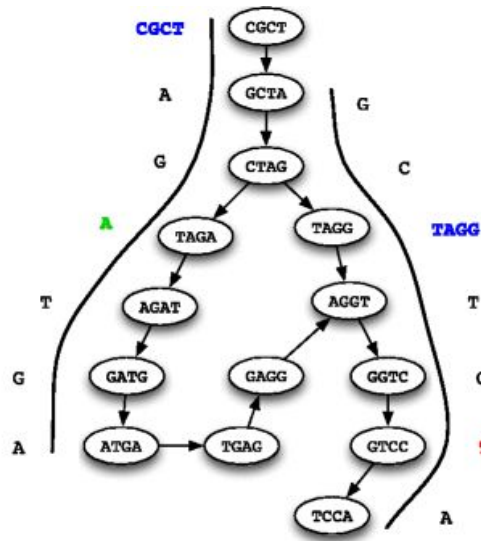
Leon

Besides genomic sequences assembly



DiscoSNP

Leon



read	encoding		anchor dictionary
	anchor	left path	
CGCTAGATGA	1	0	6, A
GCTAGGTCTA	3	2	4, (T, 3)
			0: AAAA
			1: CGCT
			2: CGAA
			3: TAGG
			· · ·
			· · ·

Minia : Besides genomic sequences assembly

Numerous applications and tools based on Minia :

- Metagenomic and transcriptome assembly
- Mutation detection
- Structural variant detection
- Taxonomic classification
- Read error correction
- Read compression

GATB : the Genome Analysis Toolbox with de Bruijn graph

Splice, DiscoSNP

TakeABreak, MindTheGap

Mapsembler

DSK

Lordec, Blooco, LoRMA

Leon

Despite these clever ideas, things are still not simple



SCS - **overly simplified abstraction** of the assembly problem

Remember the puzzle comparison

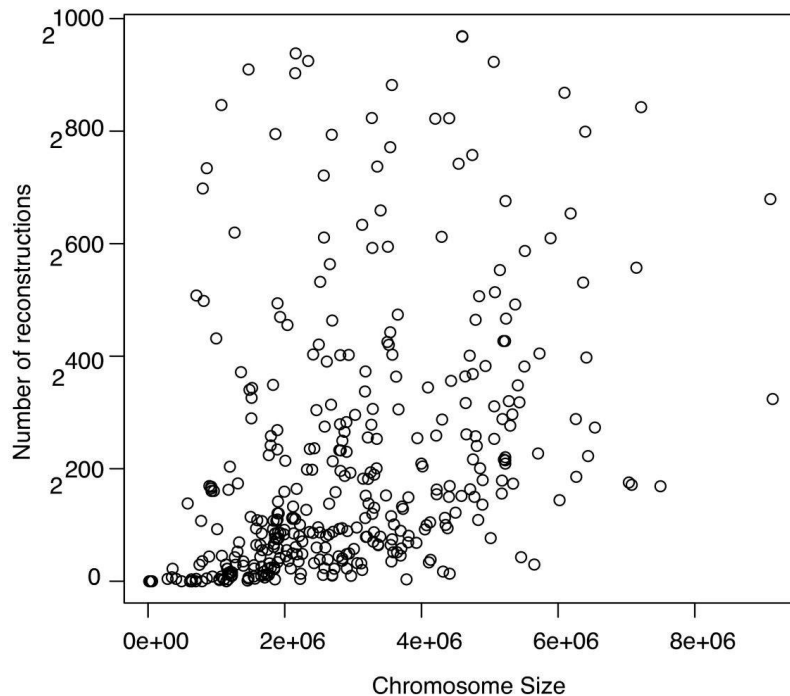
- *reads may have errors thus overlaps may not be exact*
- *repeats are frequent (**more than 50% of the human genome**)*
- *reads may come from one strain or another*

Things are not so simple

“Number of words consistent with genome graphs.


The size of the solution space for each chromosome using reads of length 50 nt. Only the 365 chromosomes with fewer than 2^{900} possible reconstructions are shown.”

375 organisms (408 chromosomes in total)



© Kingsford, Schatz, Pop, 2010

Take home message

- 
- Bioinformatics is undergoing a complete revolution
 - Biological data and bioinformatics with it, are constantly evolving
Tools and methods must be frequently updated
 - Bioinformatics **does not mean** taking computer science methods and applying them blindly to biological data
It means having truly understanding of biological data and producing methods that are perfectly adapted
 - Large scale sequencing projects will shape the future of life related sciences



Thank you !



and

Happy Journées Montoises !