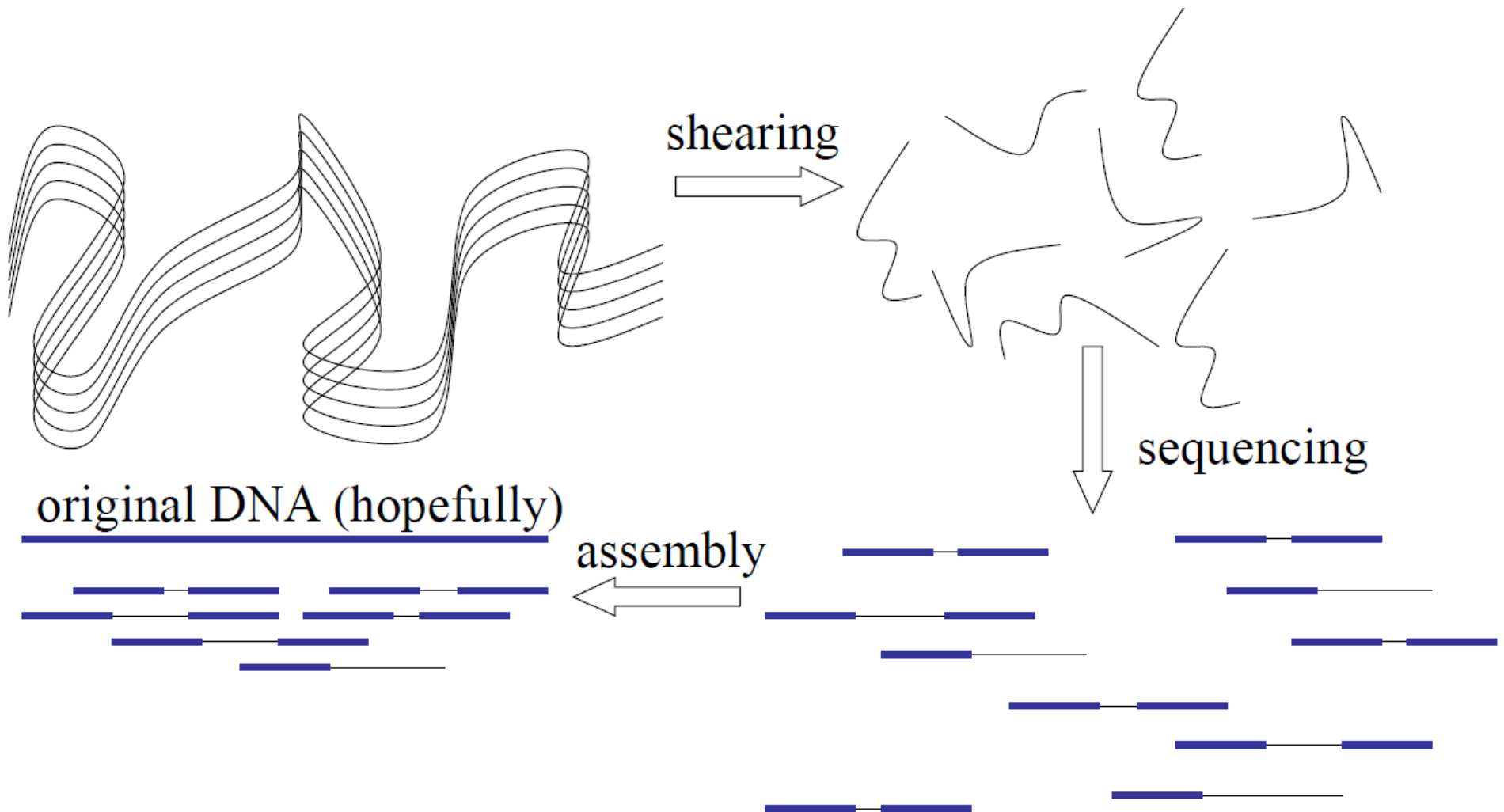
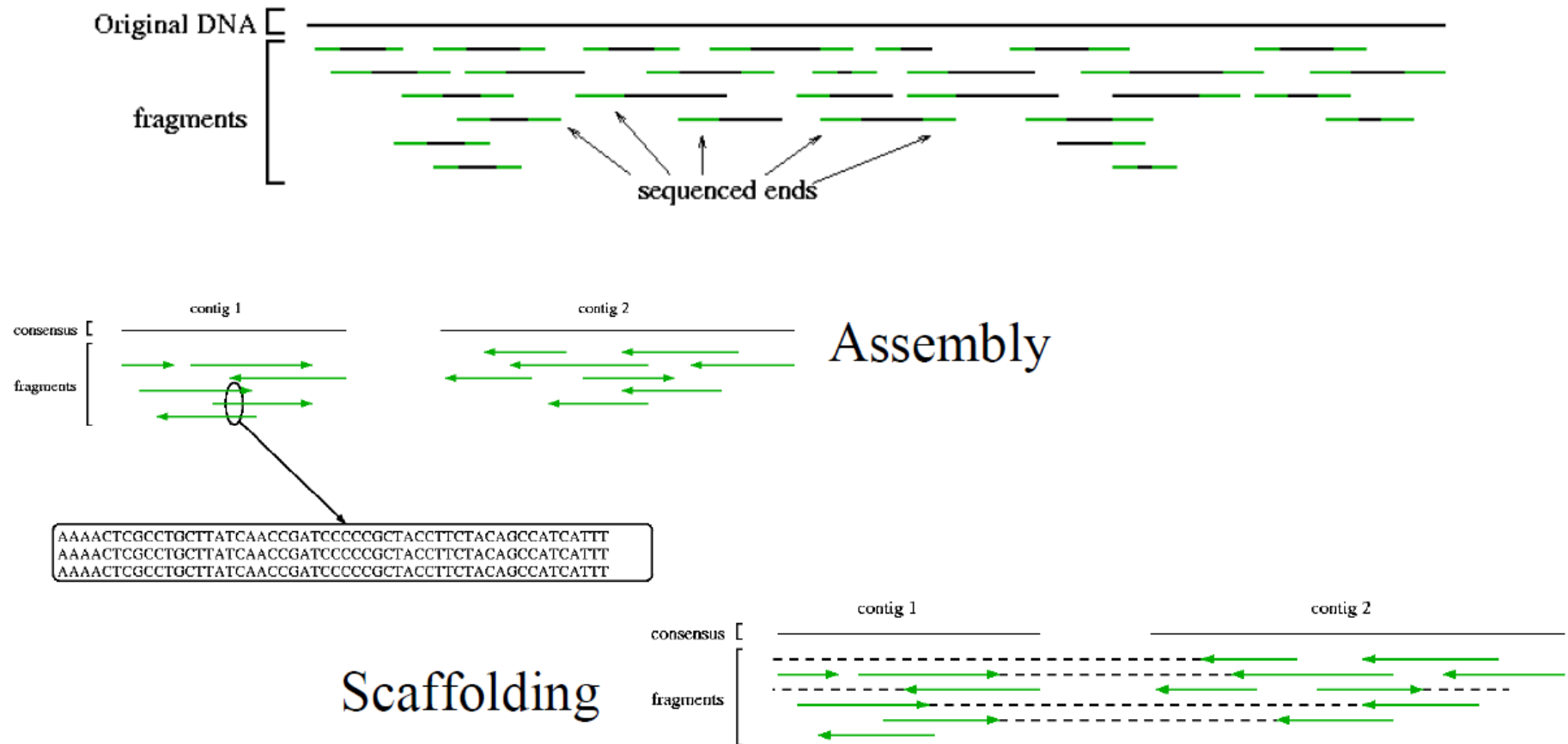


# Genome Assembly

# Shotgun Sequencing



# Overview of terms



# Shortest common superstring problem

*Given a set of strings,  $\Sigma=(s_1, \dots, s_n)$ , determine the shortest string  $S$  such that every  $s_i$  is a sub-string of  $S$ .*

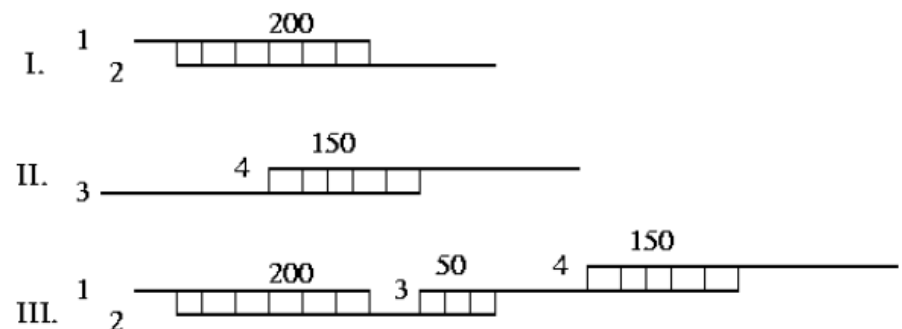
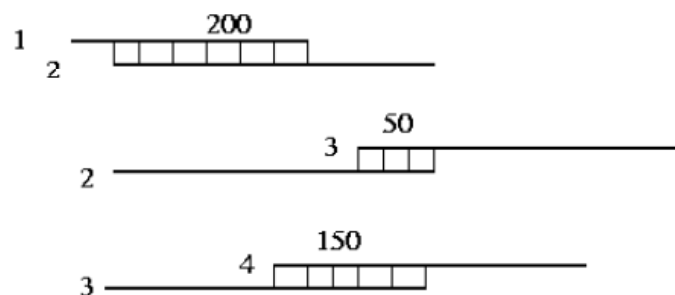
NP-hard

approximations: 4, 3, 2.89, ...

...ACAGGACTGCACAGATTGATAG

ACTGCACAGATTGATAGCTGA...

## Greedy algorithm (4-approximation)



# Greedy algorithm details

Compute all pairwise overlaps

\*Pick best (e.g. in terms of alignment score) overlap

Join corresponding reads

Repeat from \* until no more joins possible

- How do you compute an overlap alignment?
- Hint: modify Smith-Waterman dynamic programming algorithm

# Repeats (where greedy fails)

**AAAAAAAAAAAAAAAAAAAAAAAA**

AAAAAA AAAAAA AAAAAA

AAAAAA AAAAAA

AAAAAA AAAAAA

**AAAAAA**

AAAAAA

AAAAAA

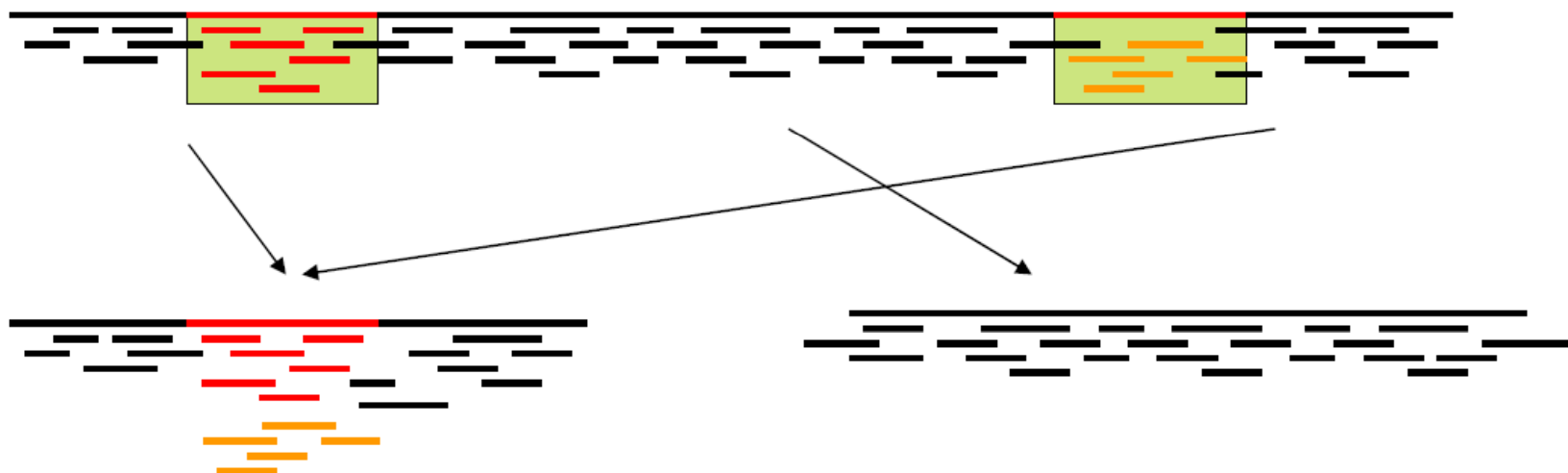
AAAAAA

AAAAAA

AAAAAA

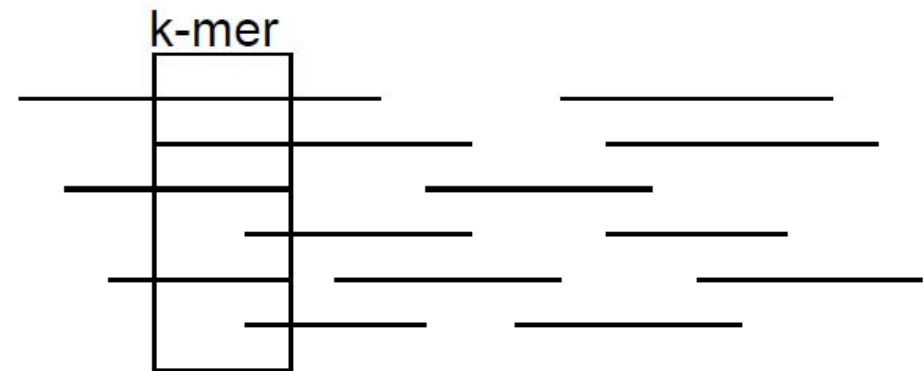
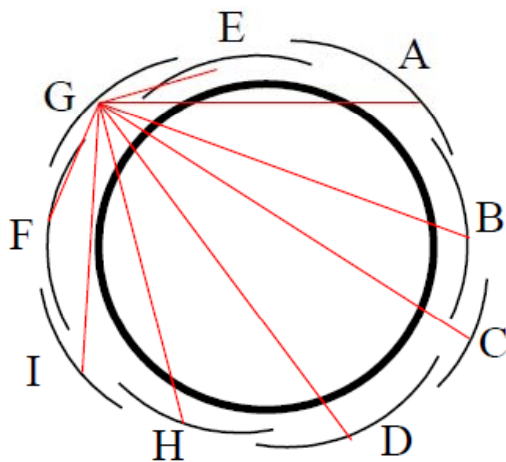
AAAAAA

AAAAAA



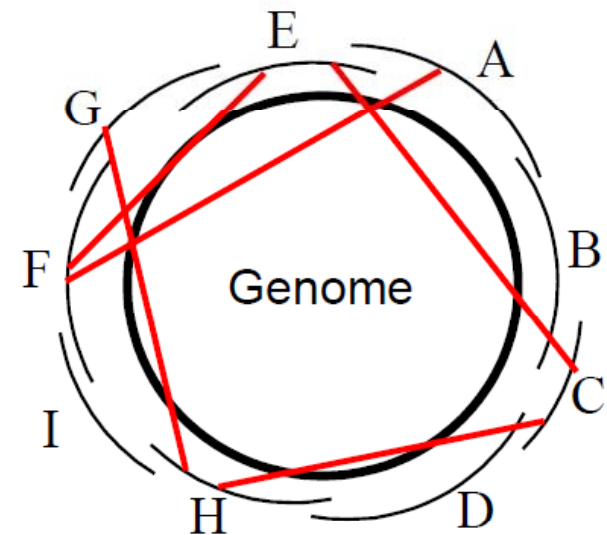
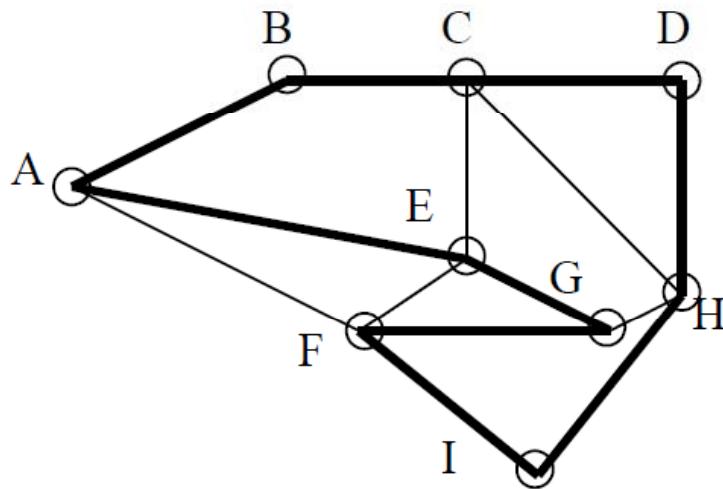
# All pairs alignment

- Needed by the assembler
- Try all pairs – must consider  $\sim n^2$  pairs
- Smarter solution: only  $n \times$  coverage (e.g. 8) pairs are possible
  - Build a table of k-mers contained in sequences (single pass through the genome)
  - Generate the pairs from k-mer table (single pass through k-mer table)



# Paths through graphs and assembly

- Hamiltonian circuit: visit each node (city) exactly once, returning to the start



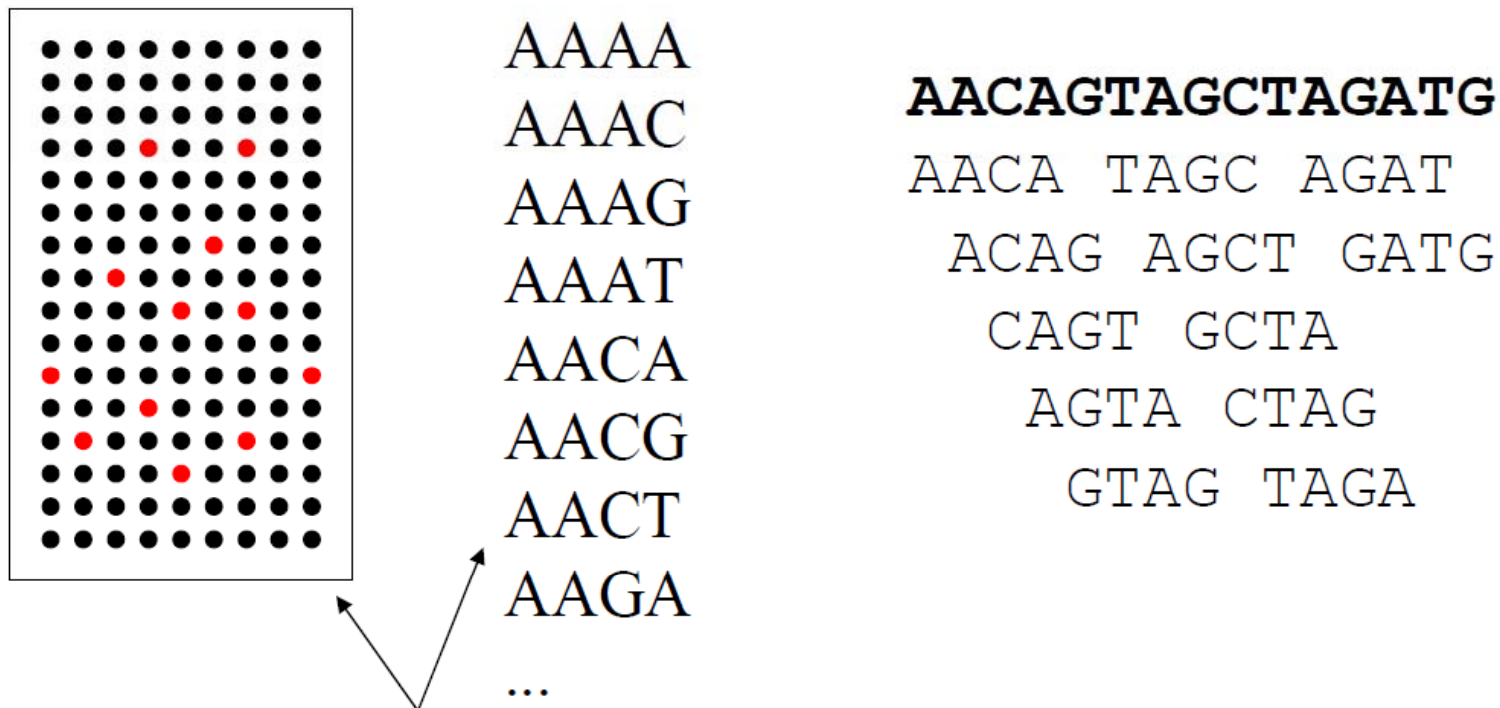
NP-complete: no efficient solution exists



# Travelling salesman problem (TSP)

Given a list of cities and their pairwise distances, the task is to find a shortest possible tour that visits each city exactly once.

# Sequencing by hybridization



probes - all possible k-mers

# Assembling SBH data

Main entity: oligomer (overlap)

Relationship between oligomers: adjacency

ACCTGATGCCAATTGCACT...

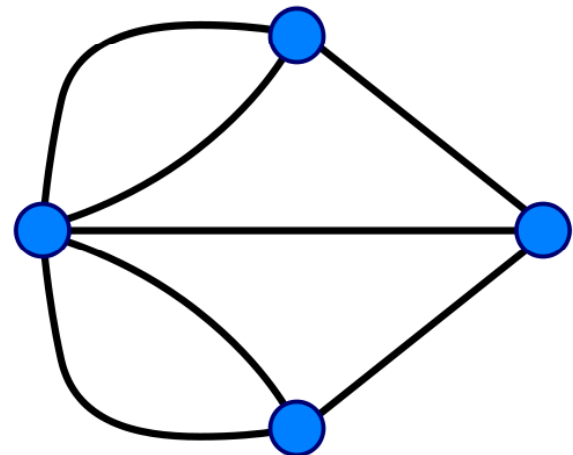
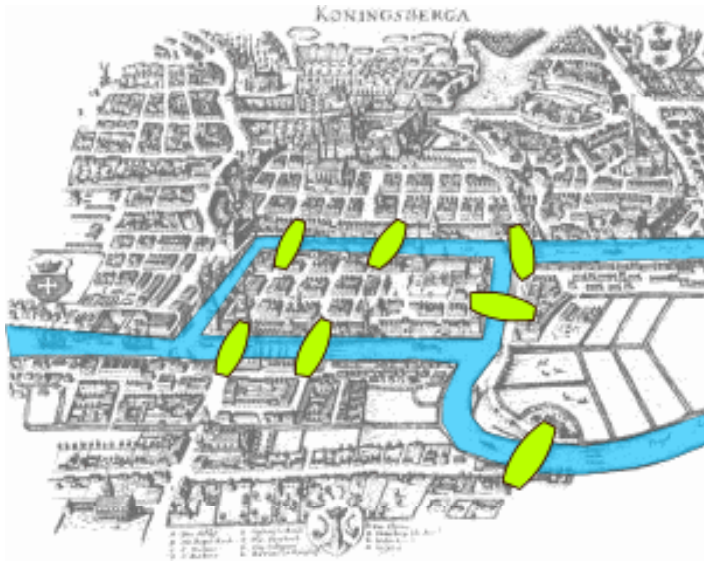
CTGAT follows CCTGA (they share 4 nucleotides: CTGA)

Problem: given all the k-mers, find the original string

In assembly: fake the SBH experiment - break the reads into k-mers

# The seven bridges of Königsberg

Find a walk through the city that would cross each bridge once and only once.



# Eulerian path

- An **Eulerian path** (Eulerian walk) in an undirected graph is a **path** that uses each edge exactly once. If such a path exists, the graph is called traversable.
- An **Eulerian cycle** (Eulerian circuit or Eulerian tour) in an undirected graph is a **cycle** that uses each edge exactly once. If such a cycle exists, the graph is called Eulerian.
- For directed graphs, path has to be replaced to directed path and cycle with directed cycle.

# Eulerian path

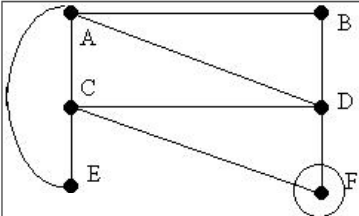
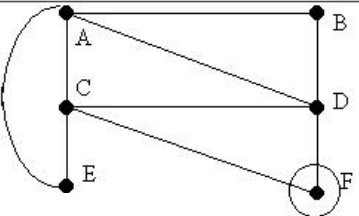
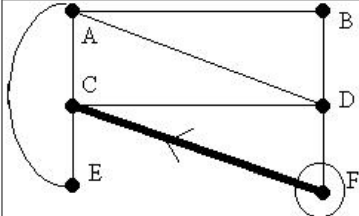
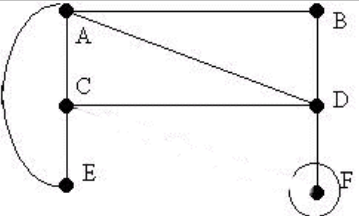
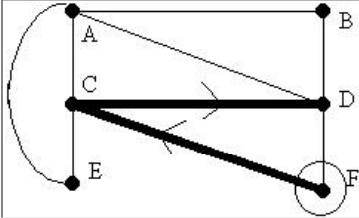
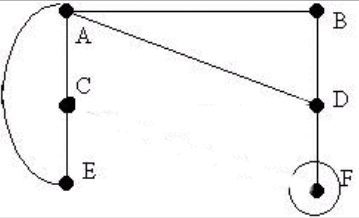
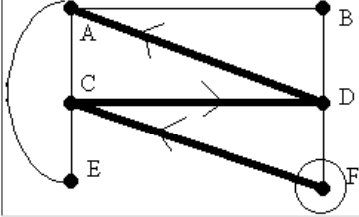
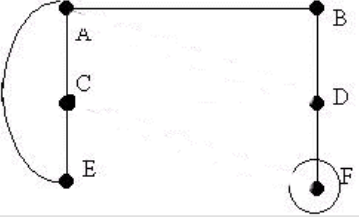
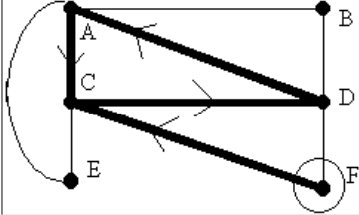
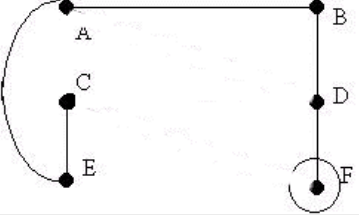
A undirected graph is Eulerian if and only if it is connected and every graph vertex has an even degree.

A directed graph is Eulerian if and only if it is connected and every vertex has equal in-degree and out-degree.

# Fleury's algorithm for finding Eulerian cycles

1. Pick any vertex to start
2. From that vertex pick an edge to traverse (never cross a bridge of the reduced graph unless there is no other choice)
3. Travel that edge and delete it from the graph
4. Repeat 2-4 until all edges have been traversed, and you are back at the starting vertex.
5. Delete the edge

# Fleury's algorithm

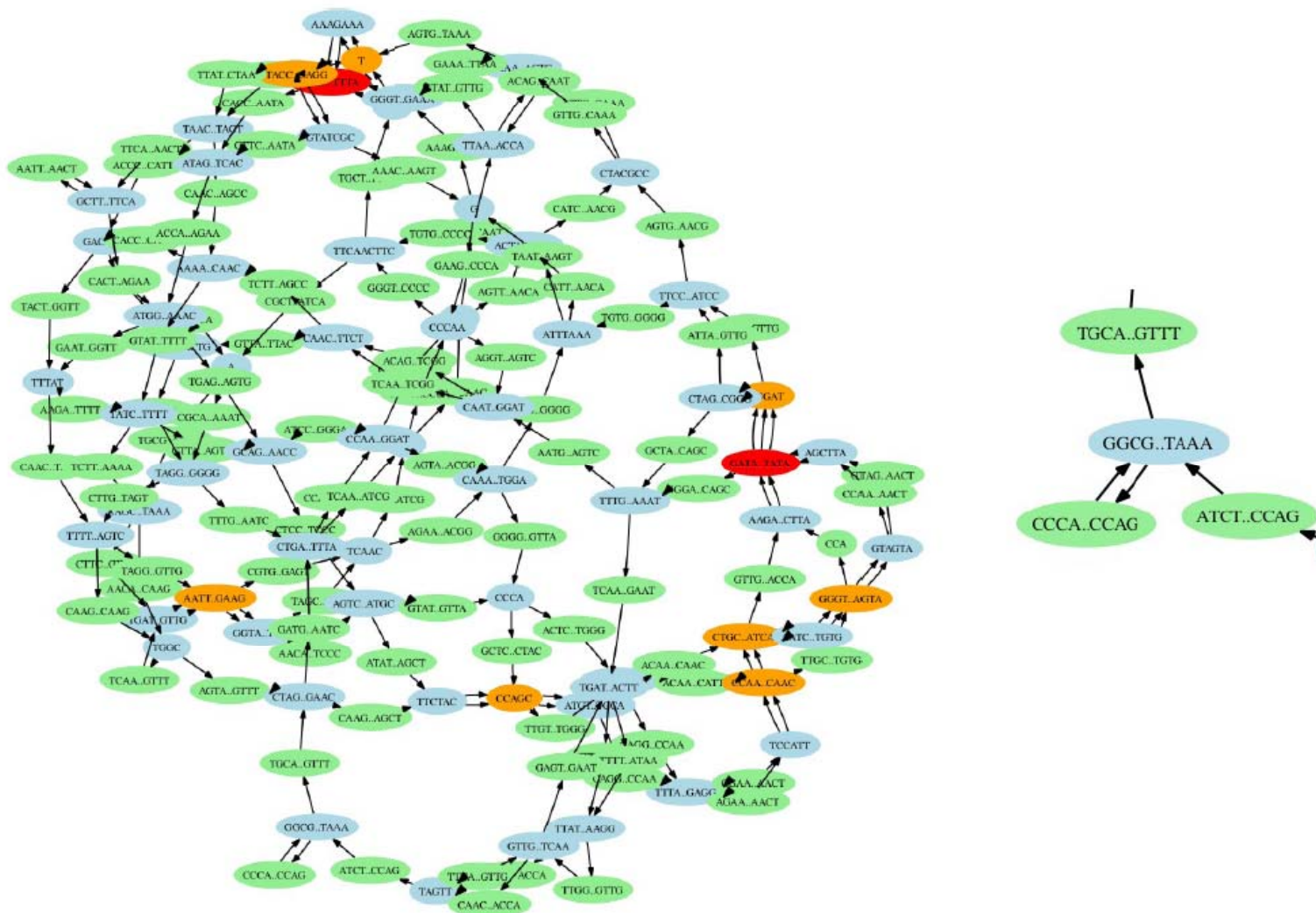
Step in recipe	Marked graph	Reduced graph
Pick any vertex (e.g. F)		
Travel from F to C (arbitrary choice)		
Travel from C to D (arbitrary)		
Travel from D to A (arbitrary)		
Travel from A to C (can't go to B: that edge is a <i>bridge</i> of the reduced graph, and there <i>are</i> two other choices, we chose one of them)		



# deBruijn graph

- Nodes – set of k-mers obtained from the reads
- Edges – link k-mers that overlap by k-1 letters  
ACCAAGTGCA  
  CCAGTGCAAT
- This formulation particularly useful for very short reads
- Solution – Eulerian path through the graph
- Note – multiple Eulerian paths possible (exponential number) due to repeats

# deBruijn graph of *Mycoplasma genitalium*





# Acknowledgement

Mihai Pop for slides.

More information on genome assembly:

[http://www.cbcb.umd.edu/research/assembly\\_primer.shtml](http://www.cbcb.umd.edu/research/assembly_primer.shtml)