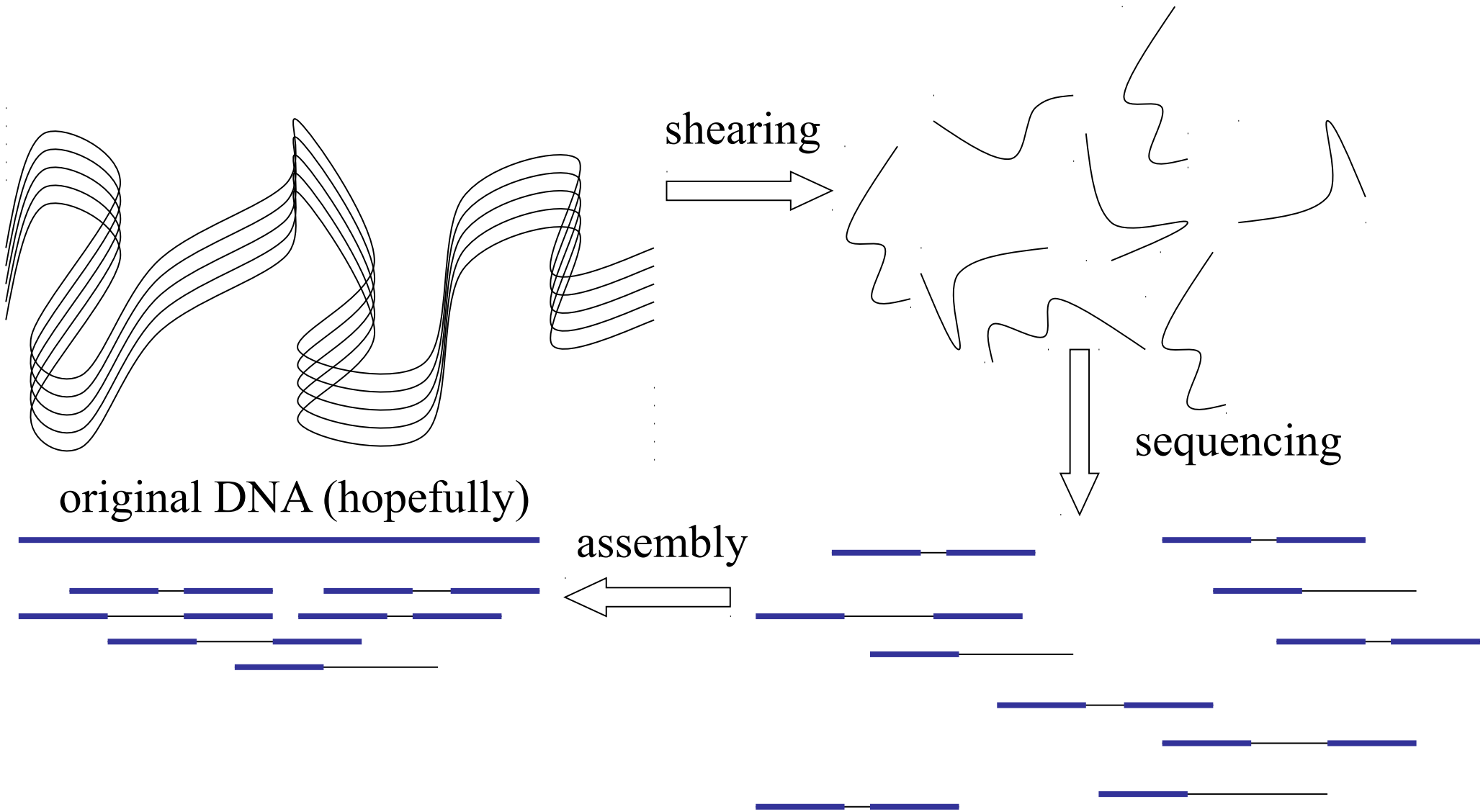
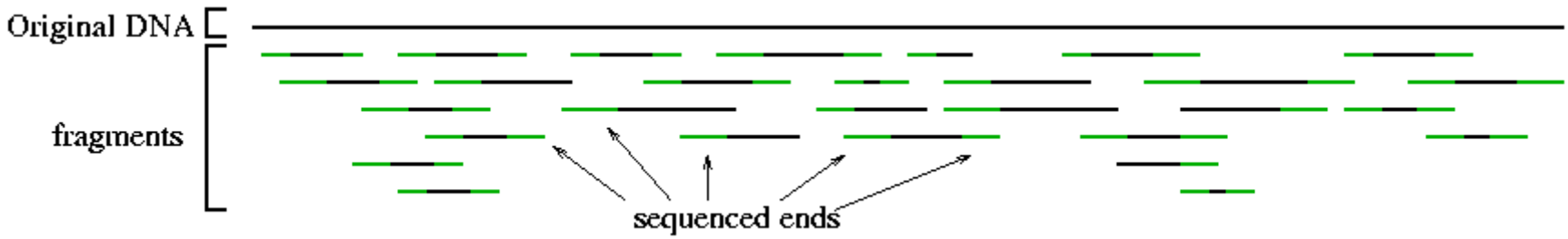
Introduction to Genome Assembly

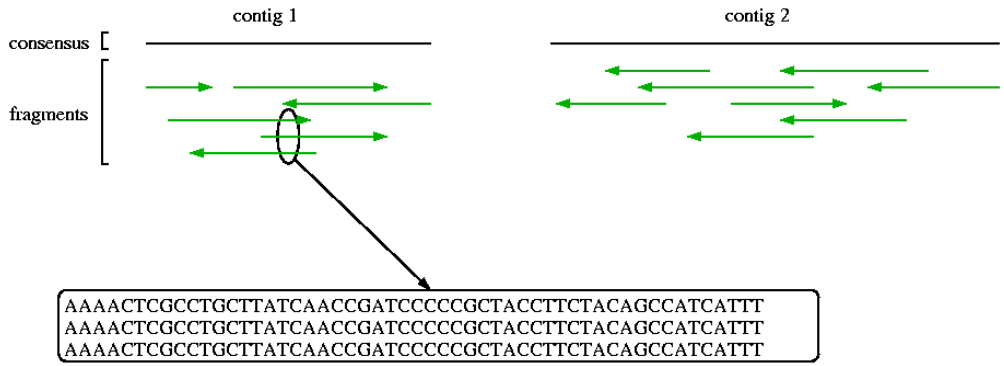
Mihai Pop

Shotgun sequencing



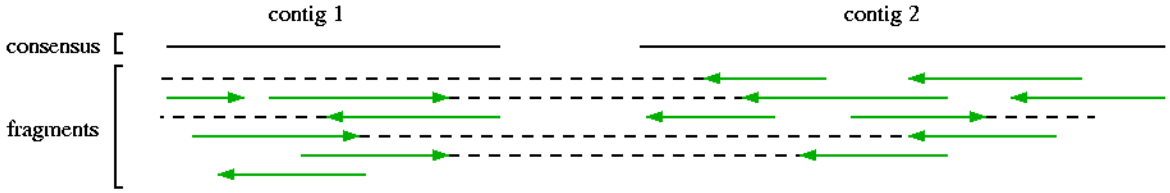
Overview of terms





Assembly

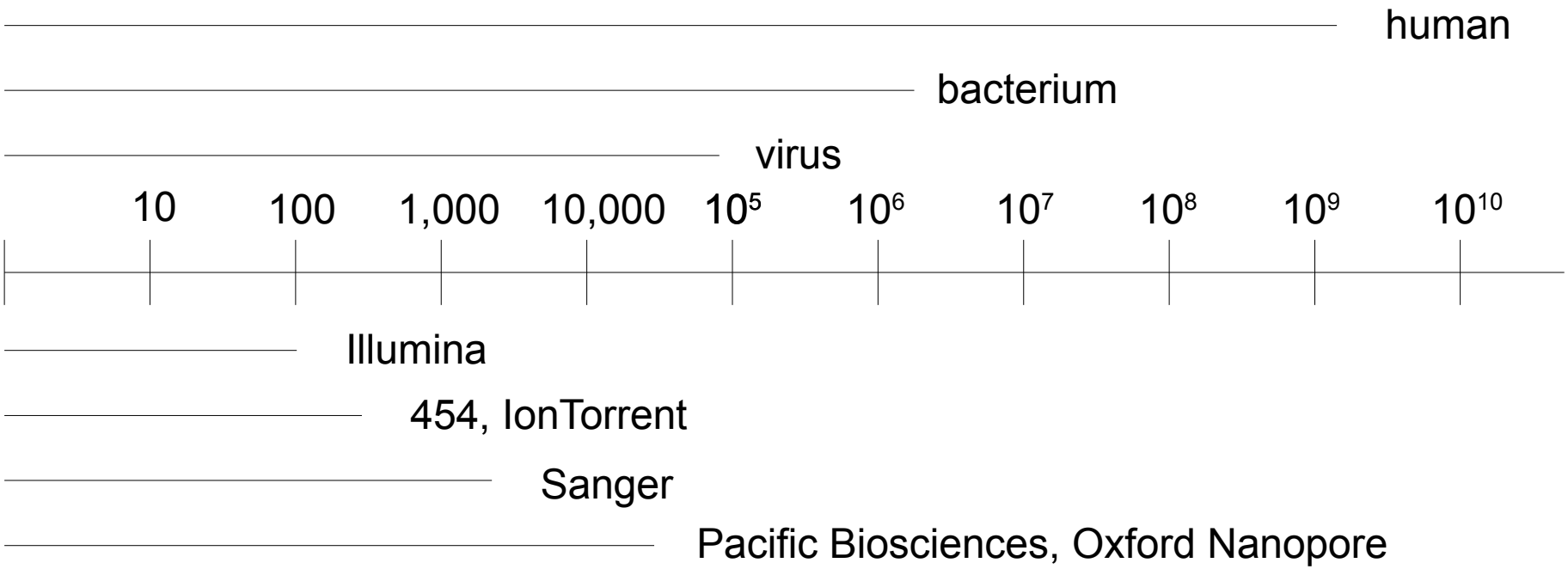
Scaffolding



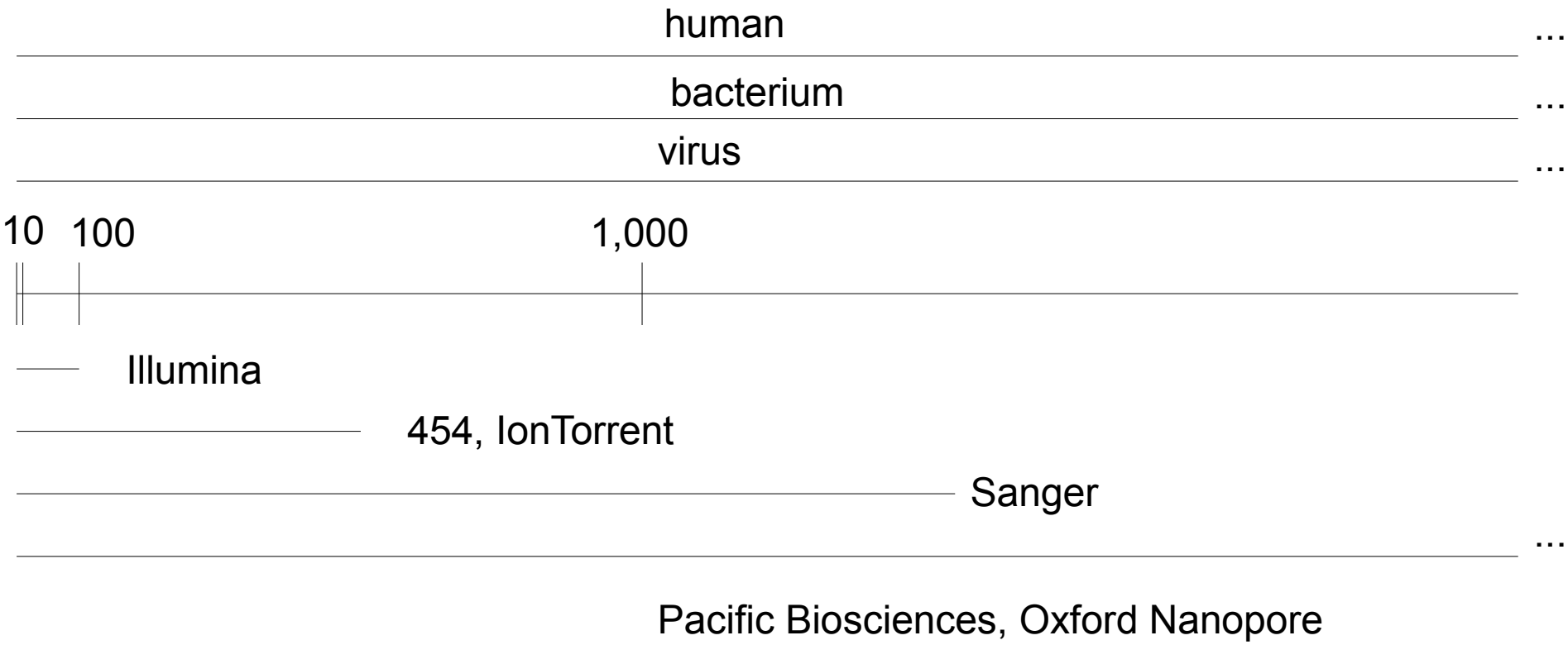
Assembly Glossary

- Read – small (50-2000bp) segment of DNA "read" by a sequencing instrument
- Mate-pair, paired ends – pair of reads whose distance from each other within the genome is approximately known
- Contig – contiguous segment of DNA reconstructed (unambiguously) from a set of reads
- Scaffold – group of contigs that can be ordered and oriented with respect to each other (usually with the help of mate-pair data)

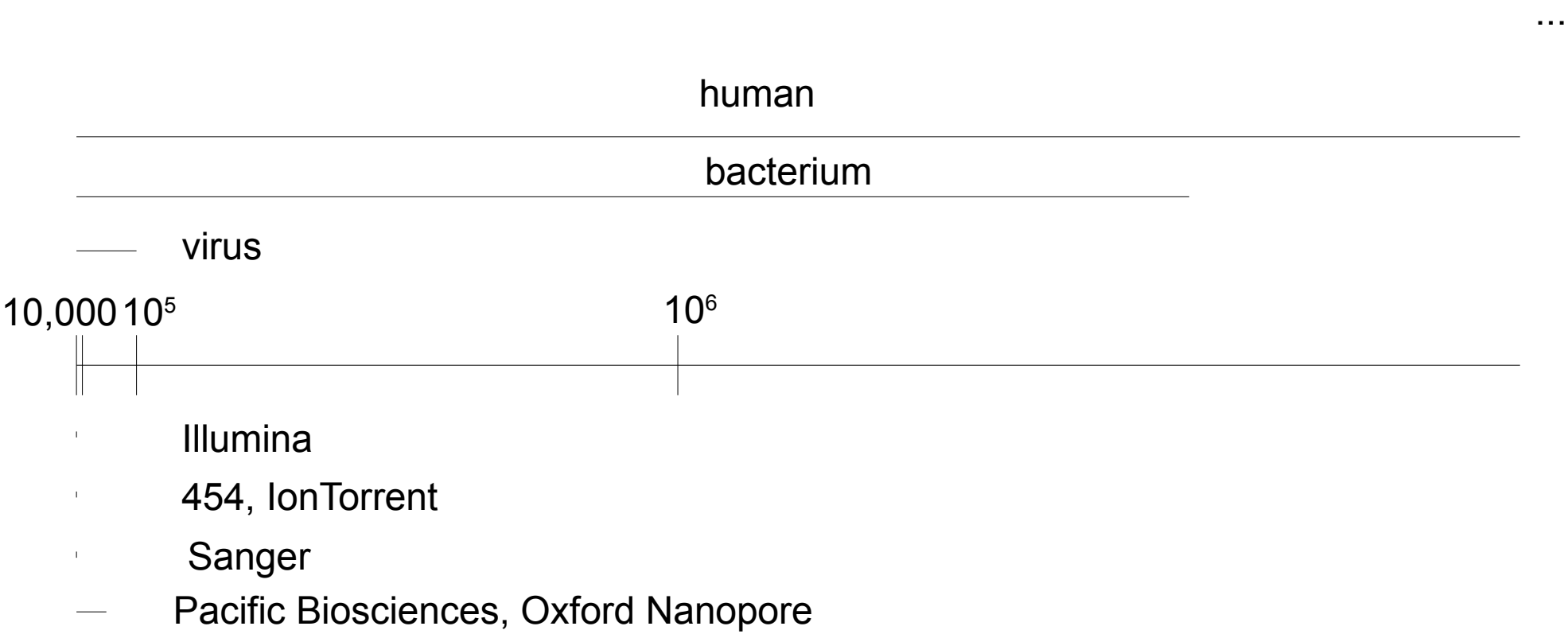
Why assembly?



Why assembly?



Why assembly?



So...

- Sequencing technologies only "read" small chunks of DNA, yet genomes are substantially larger
- The shotgun sequencing approach generates many random fragments from the original DNA
- The task of the assembly program is to stitch together the many small pieces into a reconstruction of the genome

- Essentially..... a huge jigsaw puzzle

- Think: shred a collection of Harry Potter books at random then try to rebuild the original without any additional information.

Assembling two cities

it was the best

was the *age of*

best *of times* **it**

it was the age

of times **it was**

wisdom **it was the**

it was the best

was the best *of*

the worst *of times*

was the worst *of*

was the best *of*

times **it was the**

it was the *age*

times **it was the**

was the *age of*

the best *of times*

worst *of times* **it**

age of wisdom **it**

it was the *age*

of wisdom **it was**

it was the worst

the *age of* wisdom

of times **it was**

the *age of* foolishness

Shortest common superstring problem

What are we looking for? (mathematically)

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

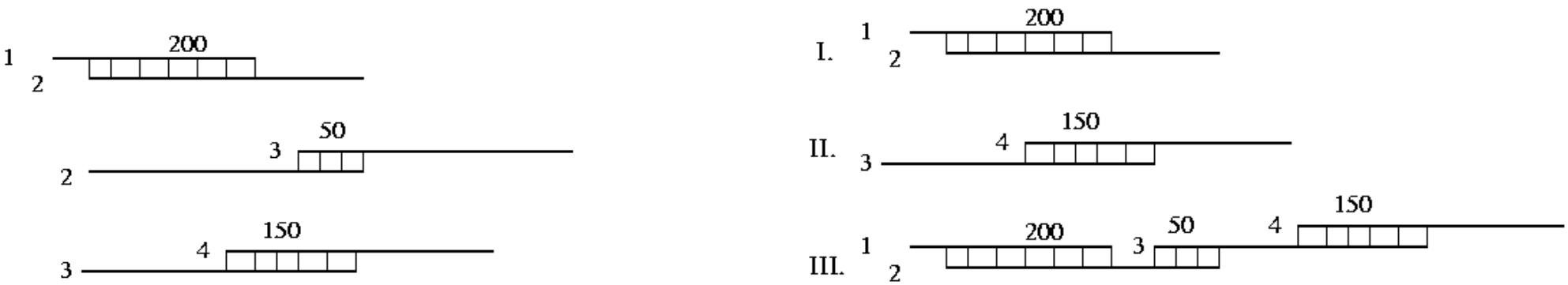
NP-hard
approximations: 4, 3, 2.89, ...

```
...ACAGGACTGCACAGATTGATAG
          ACTGCACAGATTGATAGCTGA...
```

Greedy algorithm 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

Greedy algorithm (4-approximation)

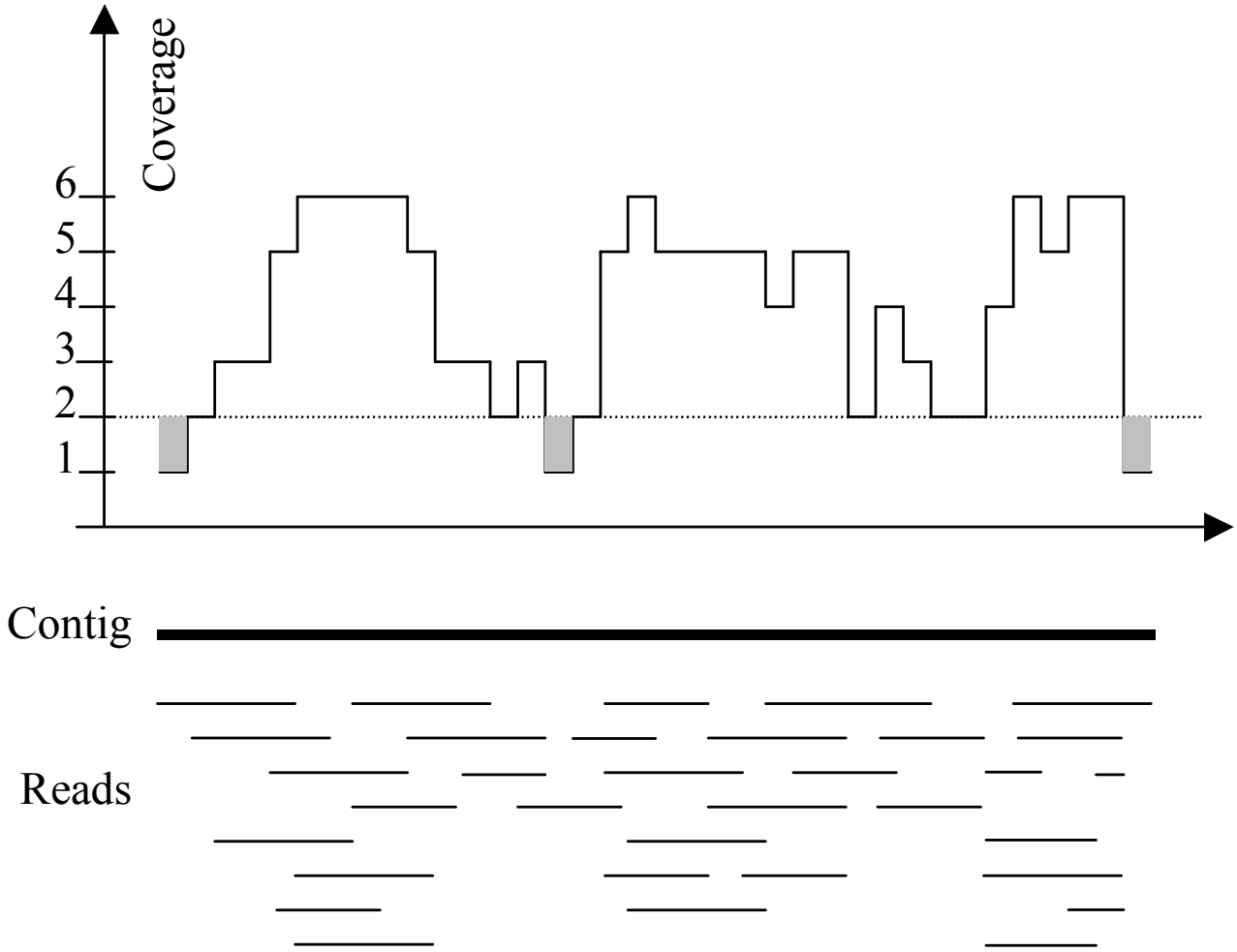


Is assembly even possible?

- If we randomly sequence will we ever cover every base in the genome?

- How much DNA do we need to sequence to cover every base in the genome?

Impact of randomness – non-uniform coverage

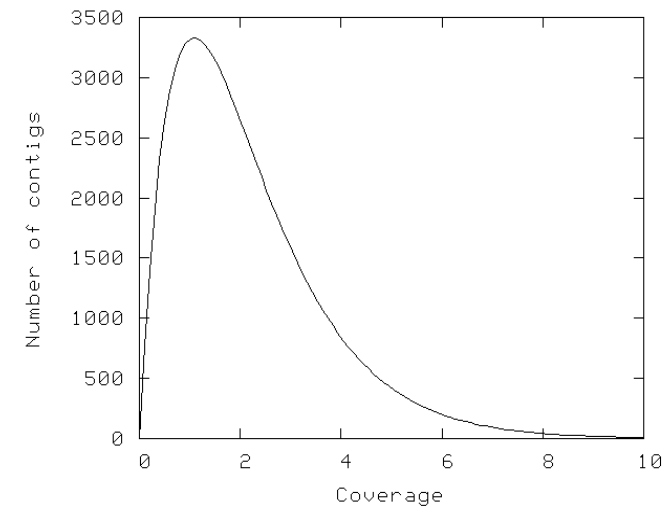


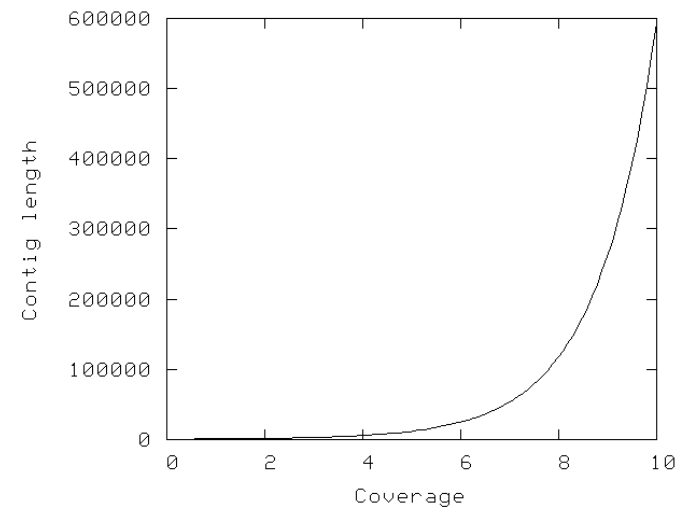
Imagine raindrops on a sidewalk

Lander-Waterman statistics

L = read length
T = minimum overlap
G = genome size
N = number of reads
c = coverage (NL / G)
$\sigma = 1 - T/L$

$E(\#islands) = Ne^{-c\sigma}$
$E(island\ size) = L(e^{c\sigma} - 1) / c + 1 - \sigma$
contig = island with 2 or more reads





Greedy approach gets 'stuck'

it was the best
was the best *of*
the best *of times*
best *of times* **it**
of times **it was**
times **it was the**
it was the age
it was the best
it was the worst

wisdom **it was the**

the worst *of times*
was the best *of*
it was the *age*
was the *age of*
of times **it was**
worst *of times* **it**
age of wisdom **it**
it was the *age*
of wisdom **it was**

the *age of* wisdom *of times* **it was**

the *age of* foolishness

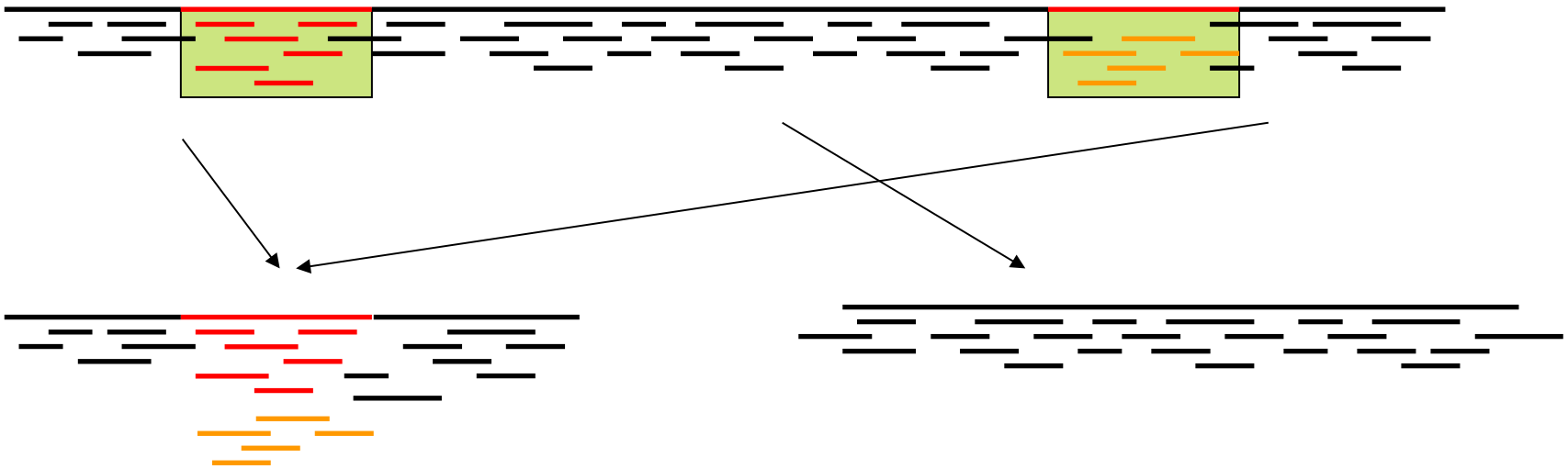
Repeats (where greedy fails)

AAAAAAAAAAAAAAAAAAAAA

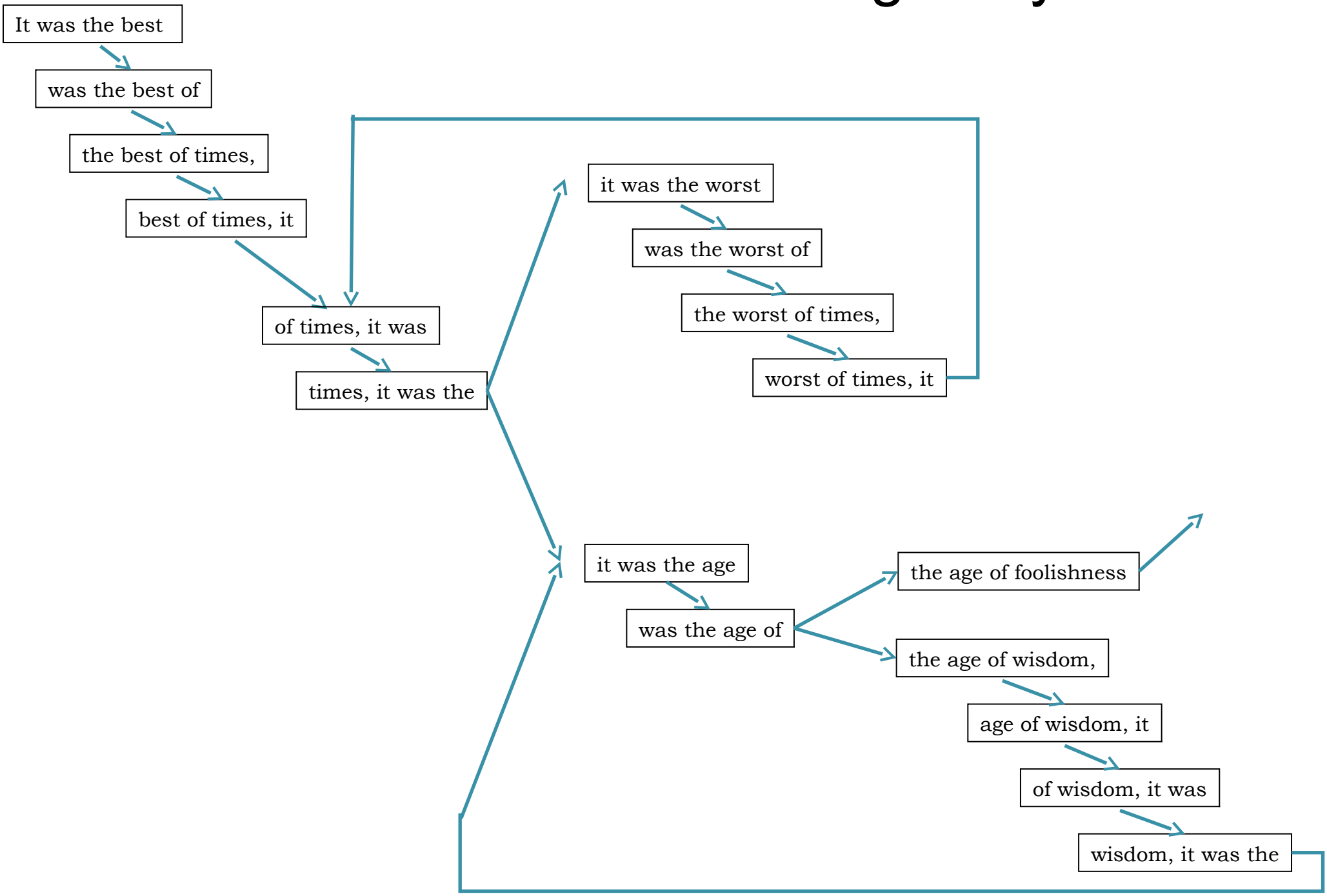
```
AAAAAA    AAAAAA AAAAAA
     AAAAAA   AAAAAA
   AAAAAA  AAAAAA
```

AAAAAA

```
AAAAAA
AAAAAA
AAAAAA
AAAAAA
AAAAAA
AAAAAA
AAAAAA
```



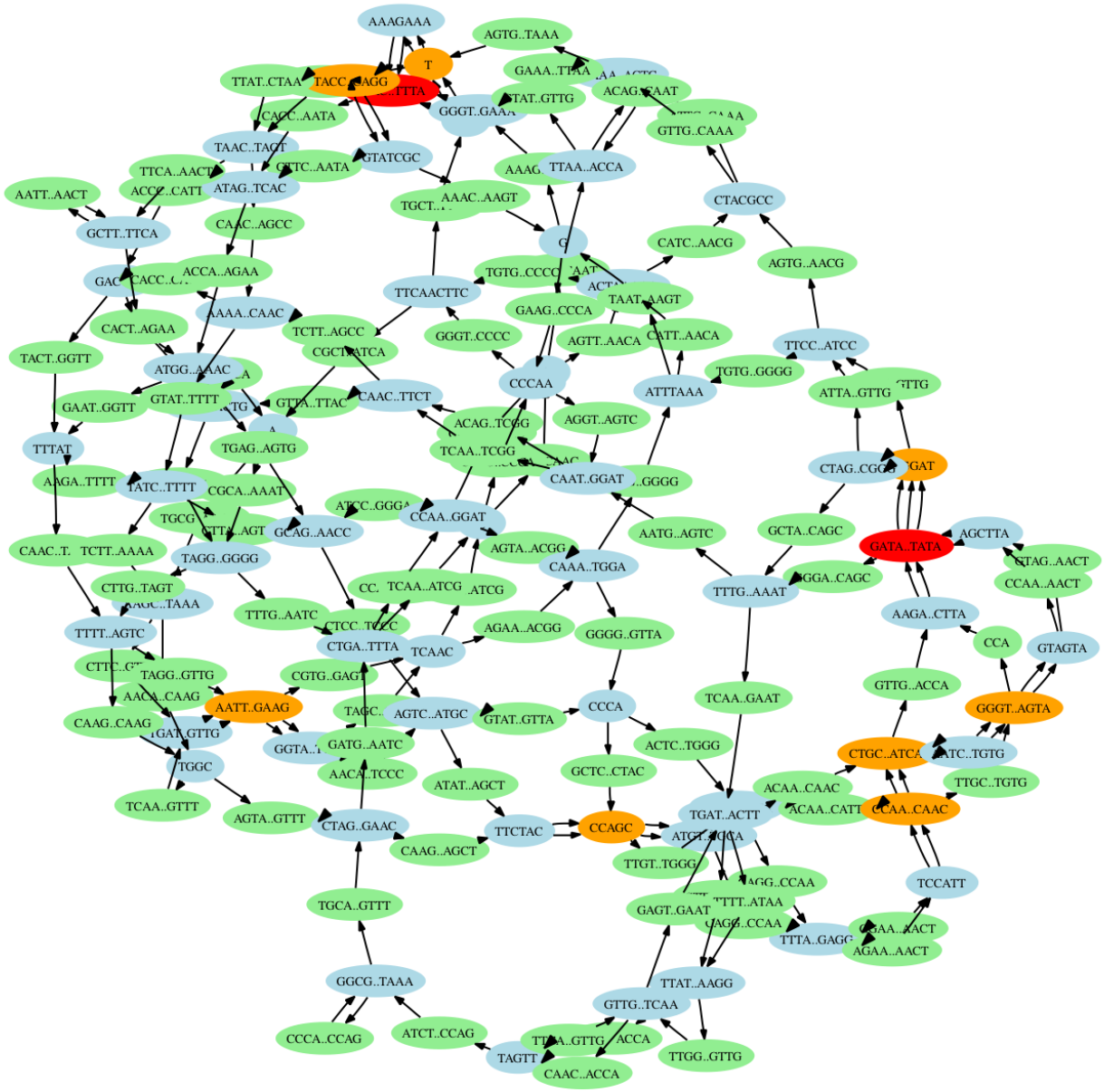
Can we do better than greedy?



Graph-based assembly

- Better than Greedy (can see better what is going on)

- Repeats still a problem



Introduction to genome assembly

Assembly is really hard!

Brief aside (assembly paradigms)

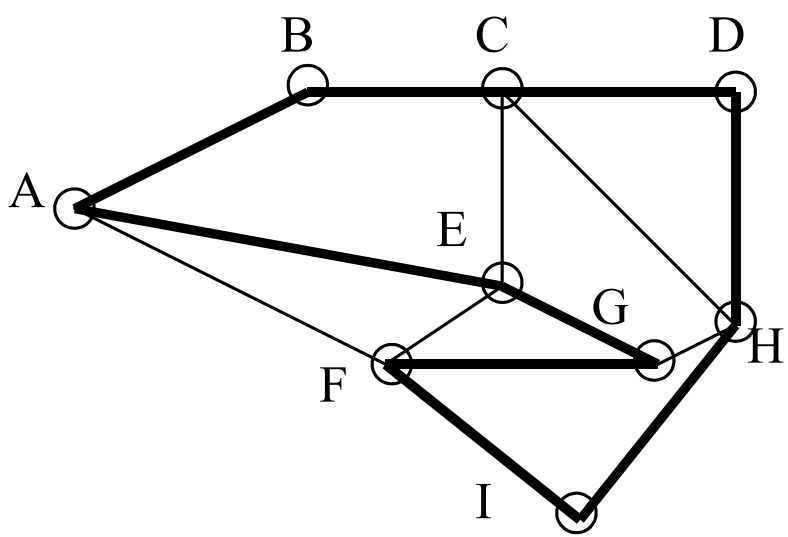
- Greedy algorithm
  - easy to implement
  - relatively efficient
  - but... can make mistakes because it is greedy (only takes into account local information)
- How can you "reason" about repeats?
- Graph theory can help: 2 paradigms
  - Overlap-Layout-Consenusus: nodes=reads, edges= reads overlap
  - deBruijn/repeat graph: nodes = k-mers, edges = k+1-mers (extracted from the reads).
- Both translate into: find a constrained path within a graph

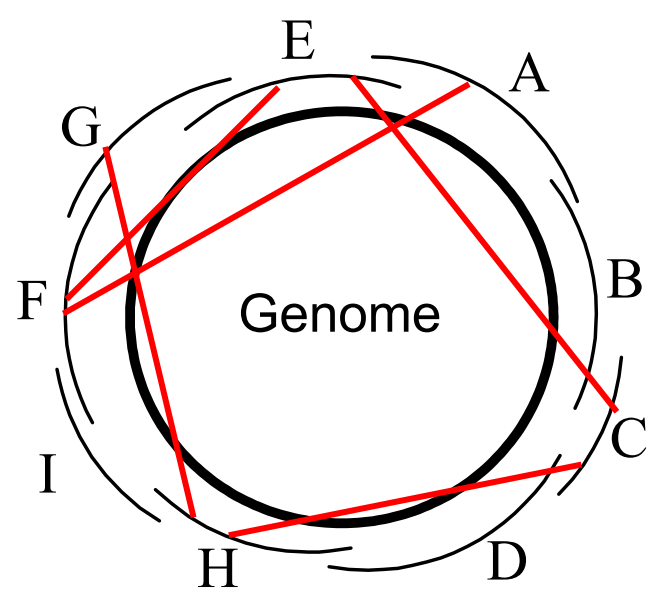
Overlap layout consensus

- Reads are represented as nodes in a graph
- Edges indicate that reads overlap
- A correct assembly must use every read hence:

  Hamiltonian traversal of graph: visit every node in a graph exactly once

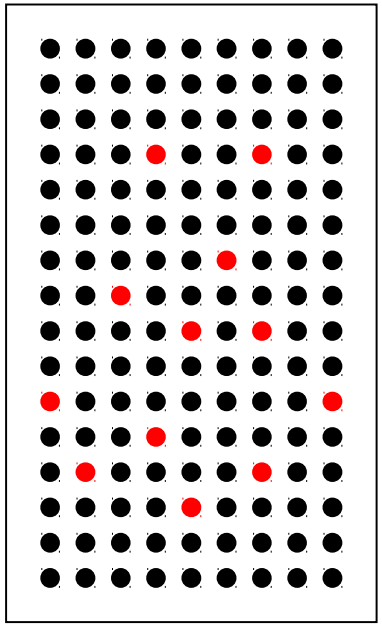
  Bad news: NP-hard





De Bruijn graph (Eulerian) formulation

Inspiration: sequencing by hybridization



AAAA
AAAC
AAAG
AAAT
AACA
AACG
AACT
AAGA
...

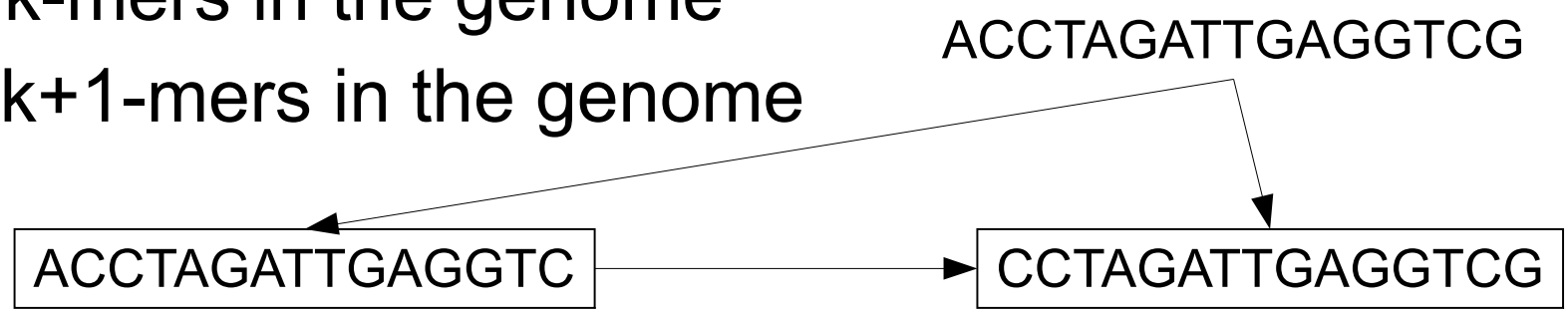
probes - all possible k-mers

AACAGTAGCTAGATG

```
AACA TAGC AGAT
 ACAG AGCT GATG
  CAGT GCTA
   AGTA CTAG
    GTAG TAGA
```

De Bruijn graph formulation

- Nodes are k-mers in the genome
- Edges are k+1-mers in the genome



- Need to use all k+1-mers in the genome (we know they are there), hence

Eulerian Path: visit every edge in the graph exactly once
Chinese Postman Path: visit every edge in the graph at least once

Good news: EP, CPP are both easy to solve

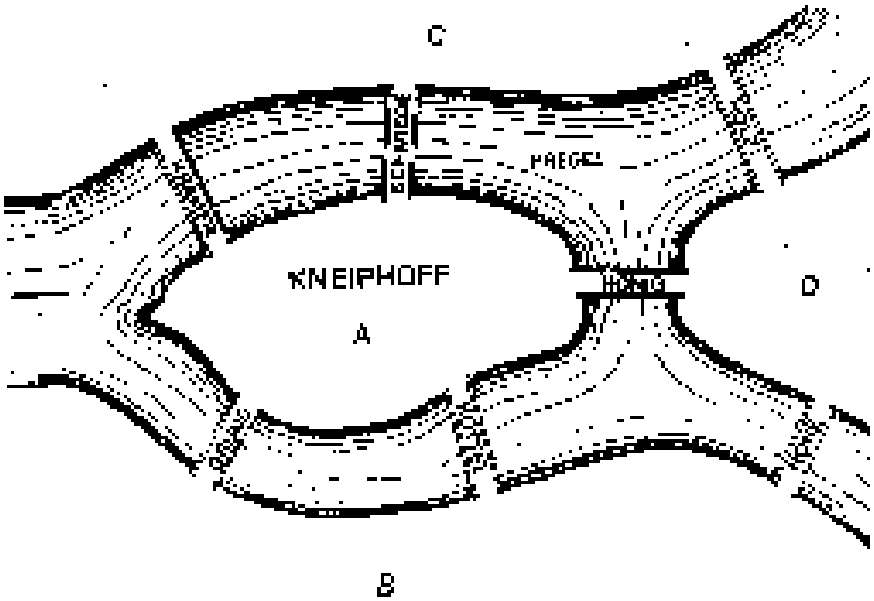
Aside: graph traversals

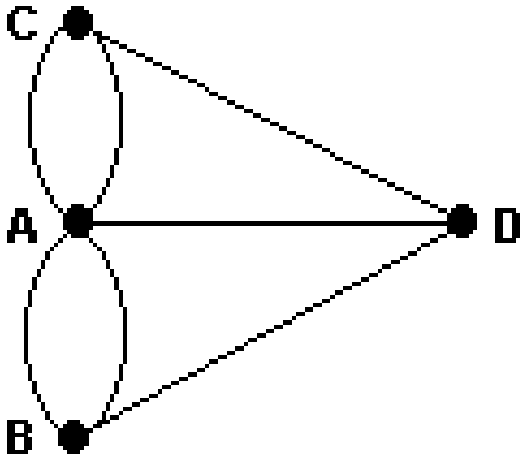
- Hamiltonian path: visit every single node of a graph EXACTLY once (NP-hard)
- Eulerian path: visit every edge of a graph EXACTLY once (polynomial time)
- Chinese Postman: find the shortest path in a graph that visits all the edges (i.e. Eulerian path where you allow a minimum number of edges to be reused)

- Note: a Hamiltonian path or an Eulerian path are not guaranteed to exist. A Chinese postman path can always be constructed

Eulerian circuit

The 7 bridges of Koenigsburg





Problem solved?

- Number of possible Eulerian/Chinese Postman tours in a graph

Generally an exponential number of compatible sequences

- Value computed by application of the BEST theorem (Hutchinson, 1975)

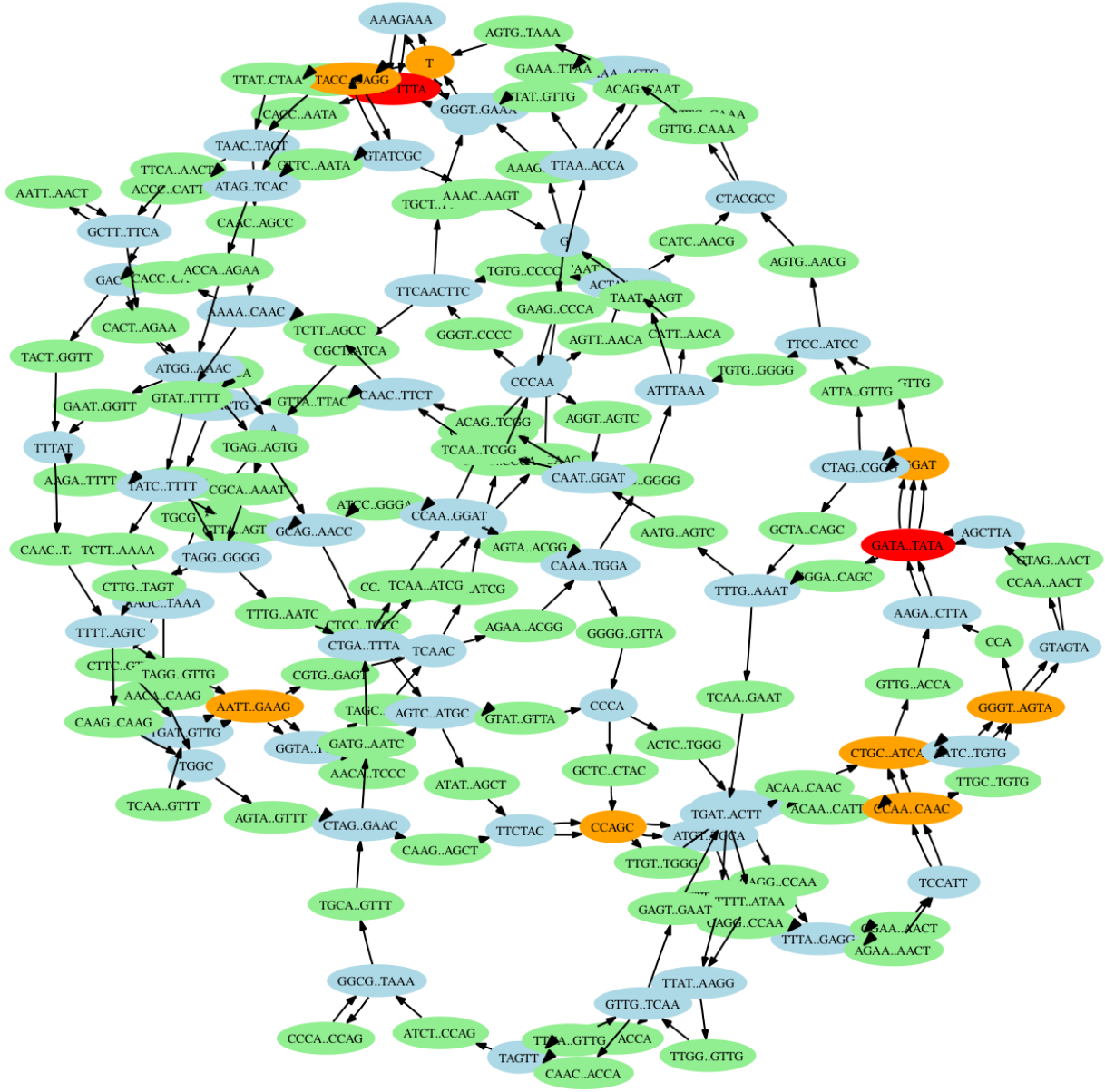
$$\mathcal{W}(G,t) = (\det L)\Big\{\prod_{u\in V}(r_u - 1)!\Big\}\Big\{\prod_{(u,v)\in E} a_{uv}!\Big\}^{-1}$$

L = n x n matrix with r_u-a_{uu} along the diagonal and -a_{uv} in entry uv

$r_u = d^+(u)+1$ if $u=t$, or $d^+(u)$ otherwise

a_{uv} = multiplicity of edge from u to v

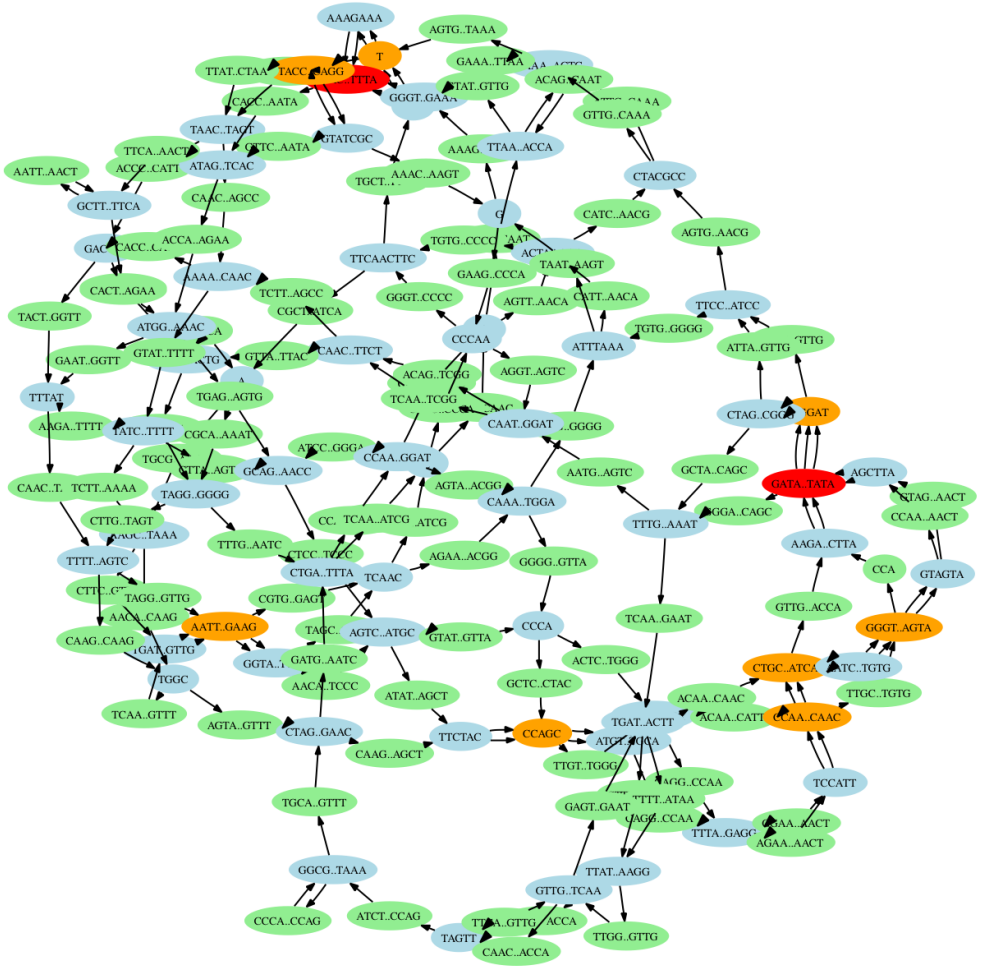
Genome assembly is impossible

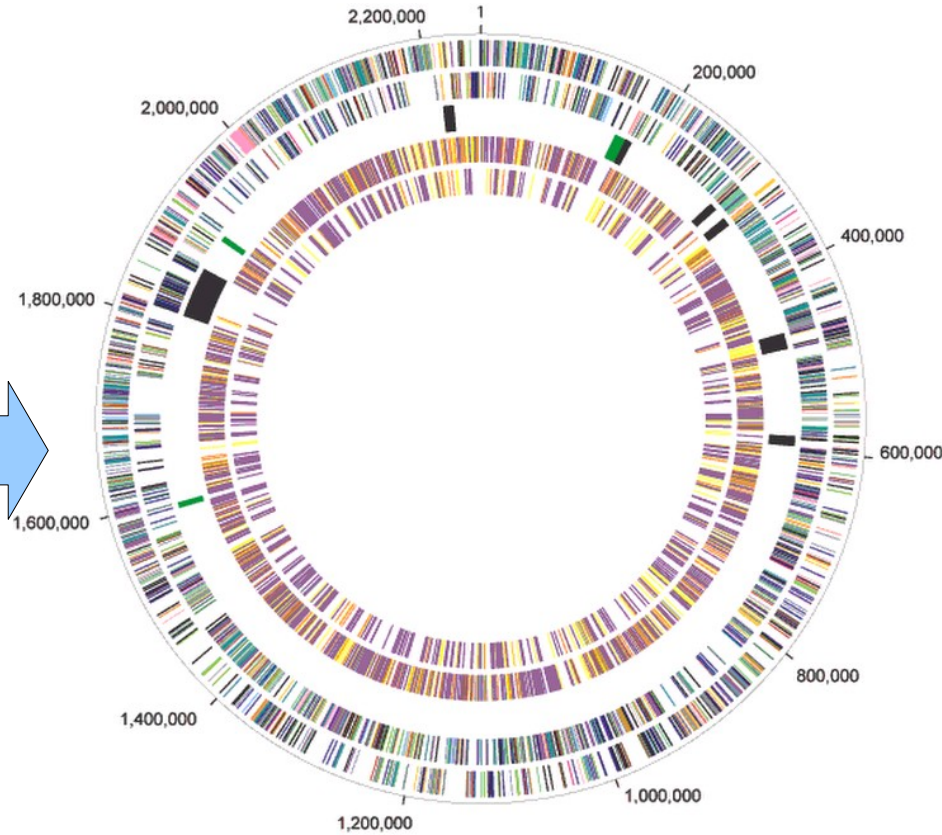


25-mer *M. genitalium* graph

Puzzle images from Puzzle Guy
http://www.amazon.com/gp/customer-media/customer gallery/A1G5WDK65OOCU5

There are no shortcuts in assembly





<u>Theorem:</u> Must try all possible assemblies before finding the correct one

Peltola et al. 1970s
Myers et al. 1990s
Medvedev et al. 2000s
Nagarajan et al. 2000s

One solution...

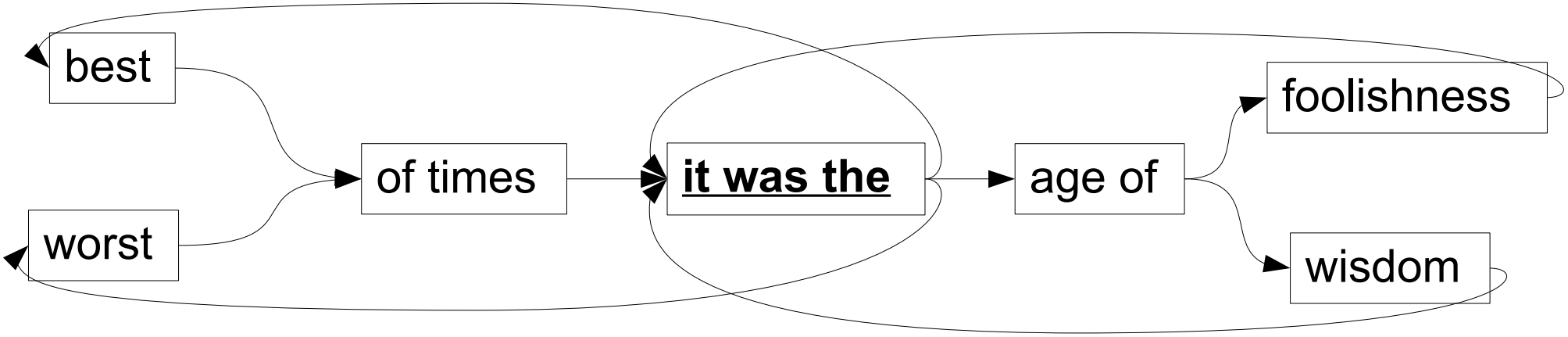
- Input: 100bp reads
- Output: 36bp contigs

True for most de Bruijn assemblers

(meta)genome assembly is impossible

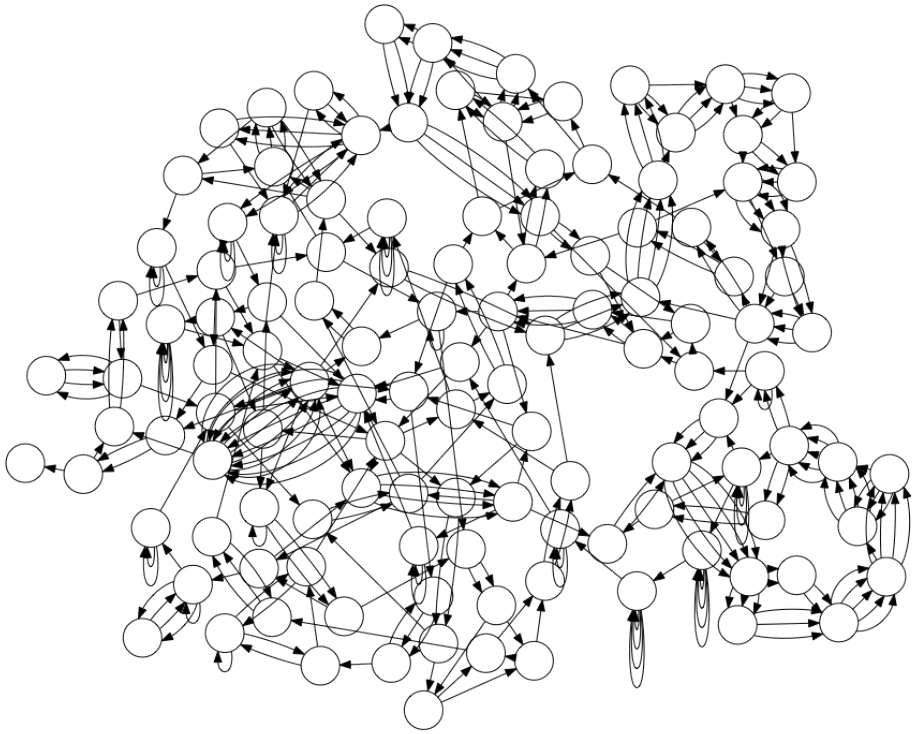
it was the best *of times* **it was the** *age of* wisdom
it was the *age of* foolishness **it was the** worst *of times*

it was the best *of times* **it was the** worst *of times*
it was the *age of* wisdom **it was the** *age of* foolishness

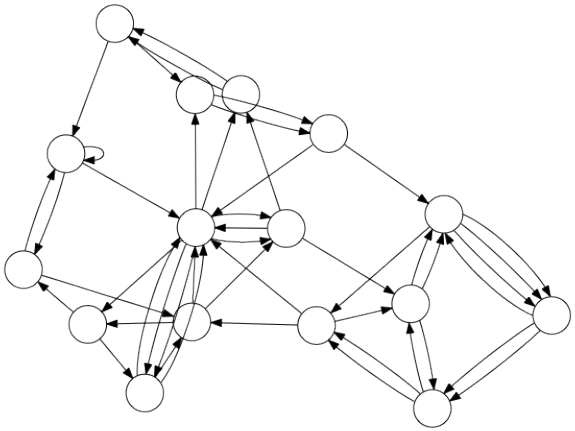


Read length matters

k = 50



k = 1,000

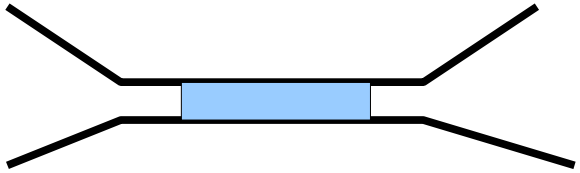


k = 5,000

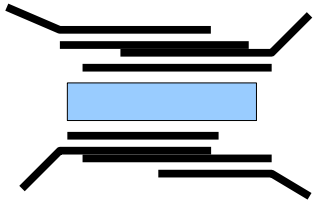
Does anyone see the mistake in the picture?

Read length matters...

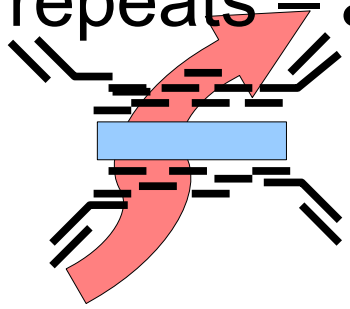
- Reads (much) longer than repeats – assembly trivial



- Reads roughly equal to repeats – assembly computationally difficult (NP-hard)



- Reads shorter than repeats – assembly undetermined



Number of possible reconstructions exponential in # of repeats

Nagarajan, Pop. J. Comp. Biol. 2009, Kingsford et al., BMC Bioinformatics 2010

Assembly...summary

- The basic idea of both OLC and deBruijn approaches: identify sections of DNA that MUST be present in the actual genome:
  - OLC – each read must be used because it is a piece of the original genome
  - deBruijn – each edge must be used because the DNA string corresponding to it is a piece of the original genome

- Both approaches provably "impossible" (NP-hard)