CMSC423: Bioinformatic Algorithms, Databases and Tools
Lecture 15

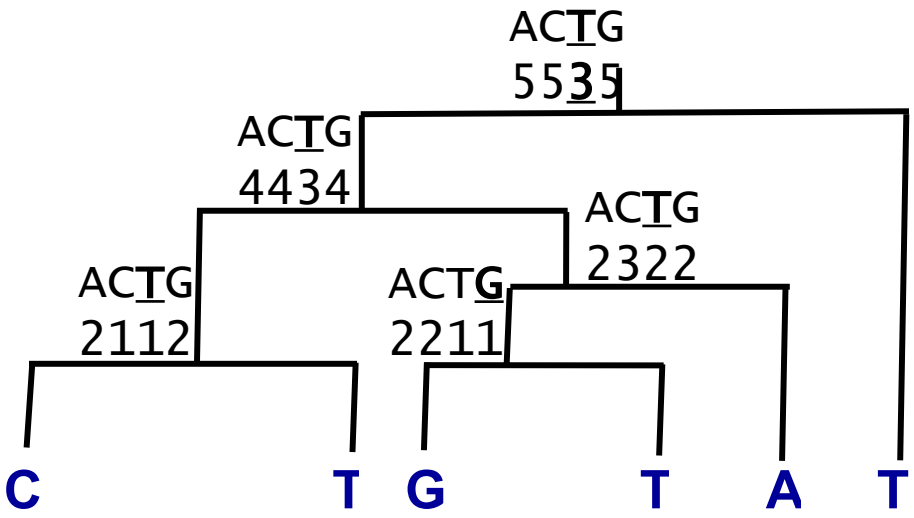
Genome assembly

Admin

- Project questions?

Questions/answers

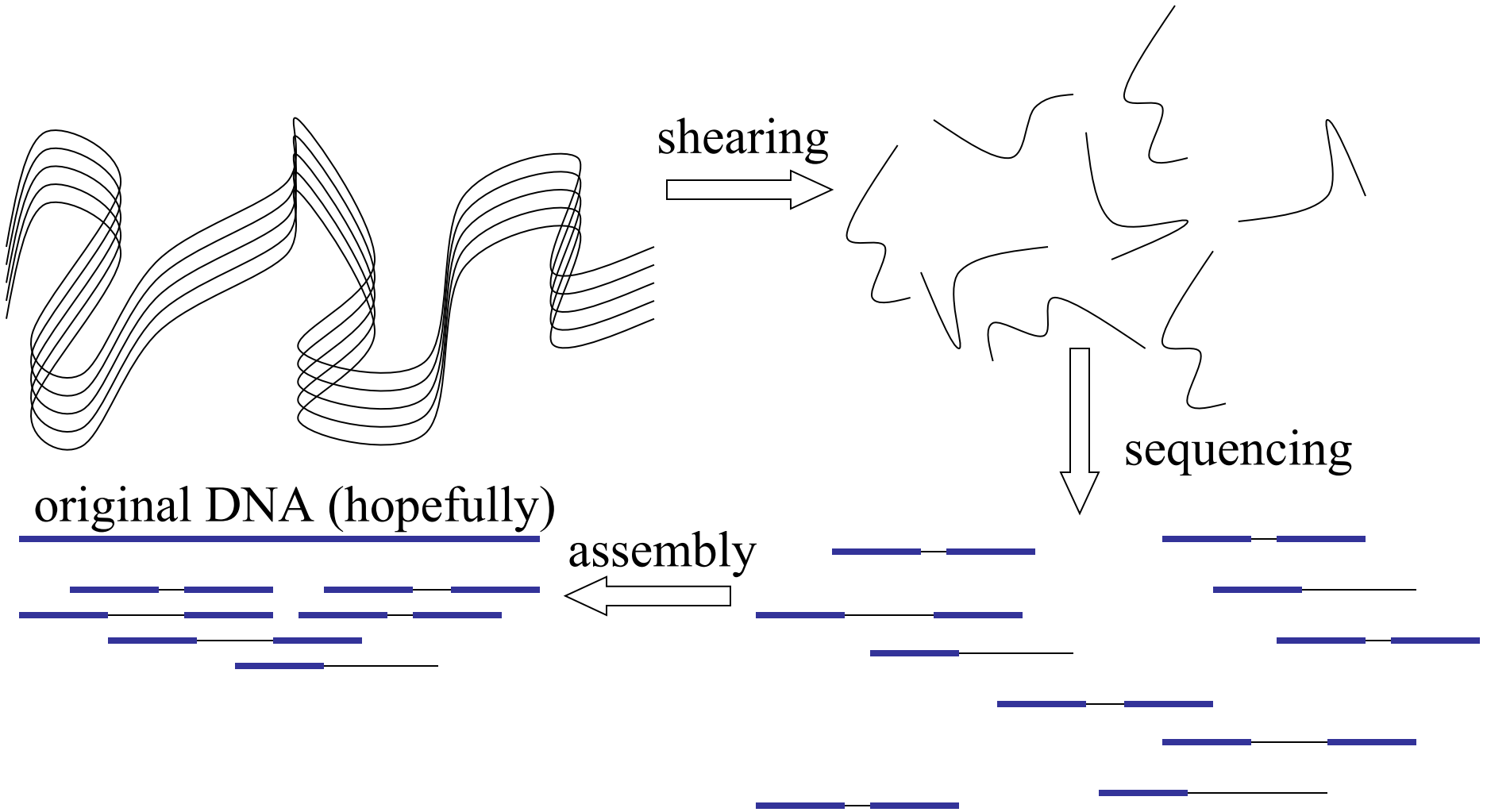
- Why do you need a multiple alignment for phylogeny?
- What is the running time of the neighbor-joining algorithm, given k sequences of length L?
- What is the parsimony score of the following tree, and what are the labels at internal nodes?



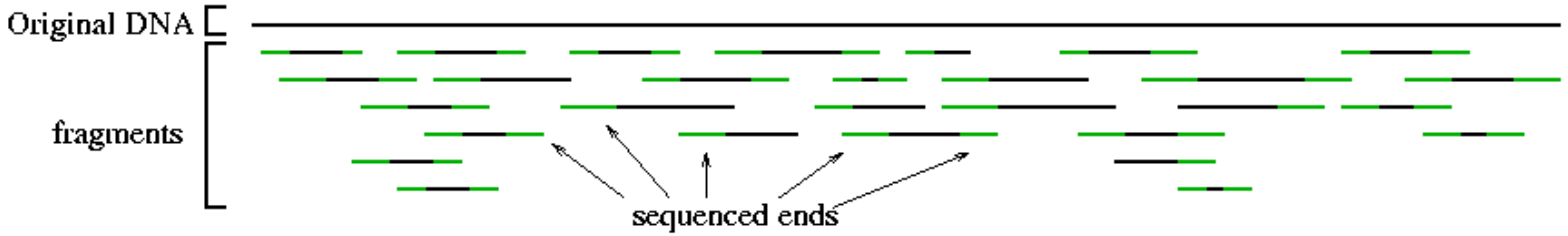
Reading assignment

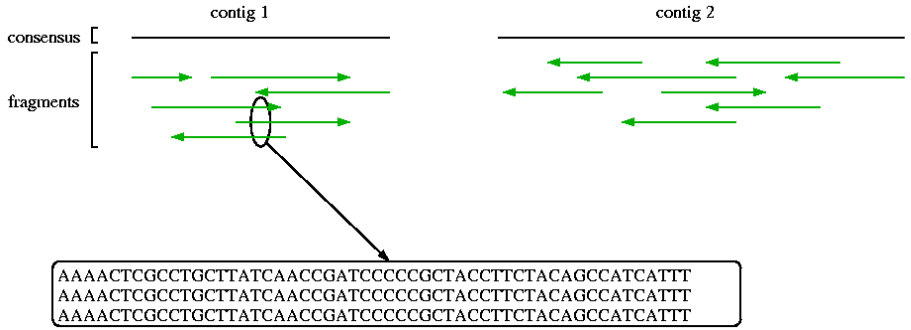
- http://www.cbcb.umd.edu/research/assembly_primer.shtml
- Chapter 4.5 – coverage statistics
- Chapter 8 – genome assembly

Shotgun sequencing



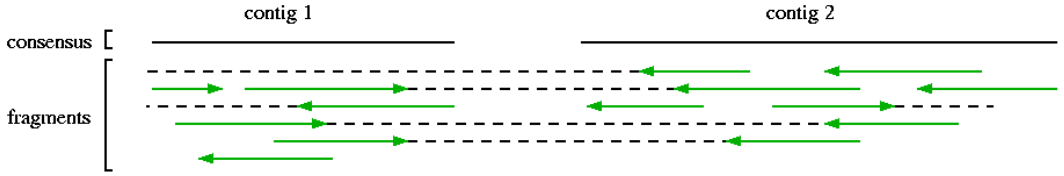
Overview of terms





Assembly

Scaffolding



Shortest common superstring problem

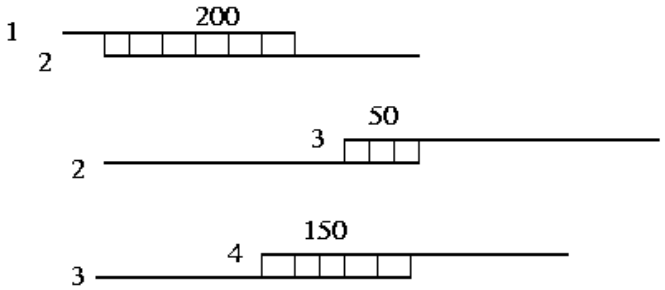
Given a set of strings, $\Sigma=(s_1, ..., s_n)$, determine the shortest string S such that every s_i is a sub-string of S.

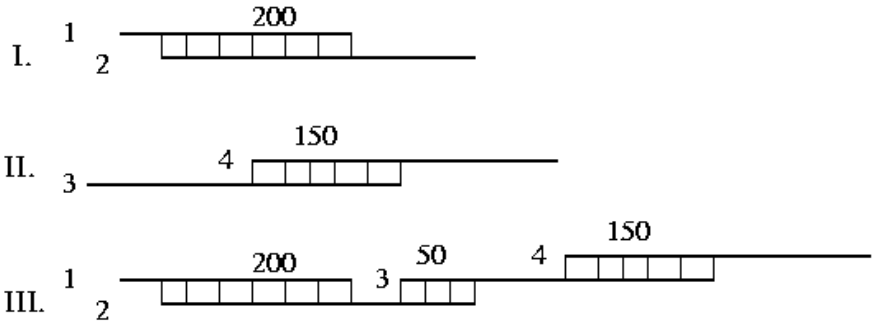
NP-hard

approximations: 4, 3, 2.89, ...

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

Greedy algorithm (4-approximation)





phrap, TIGR Assembler, CAP

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

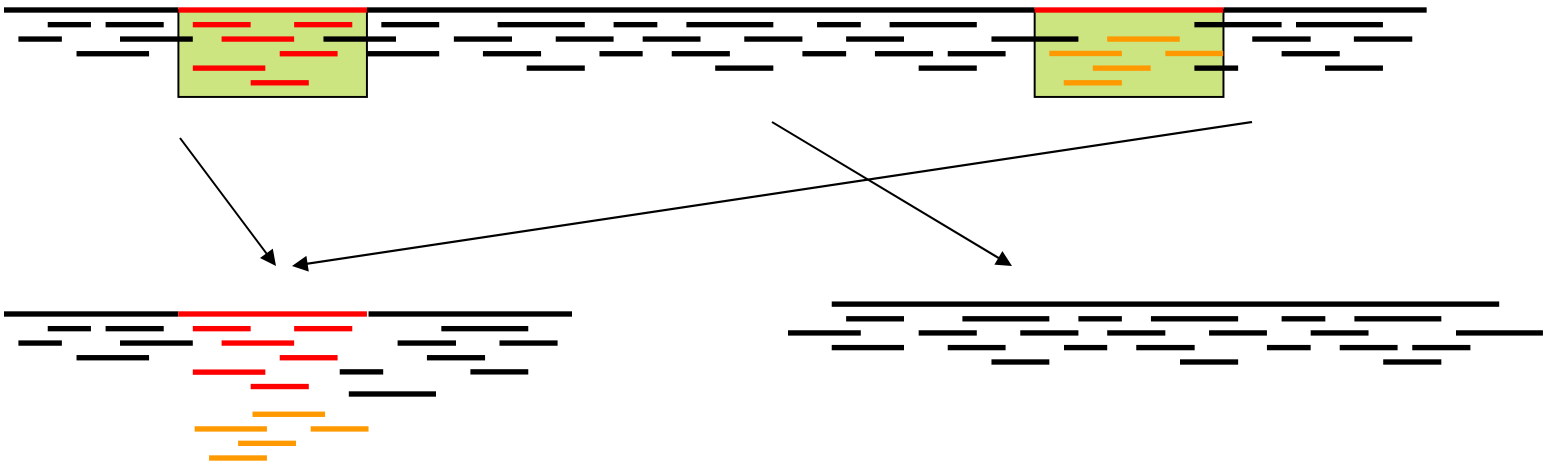
- How do you compute an overlap alignment?
- Hint: modify Smith-Waterman dynamic programming algorithm

Repeats (where greedy fails)

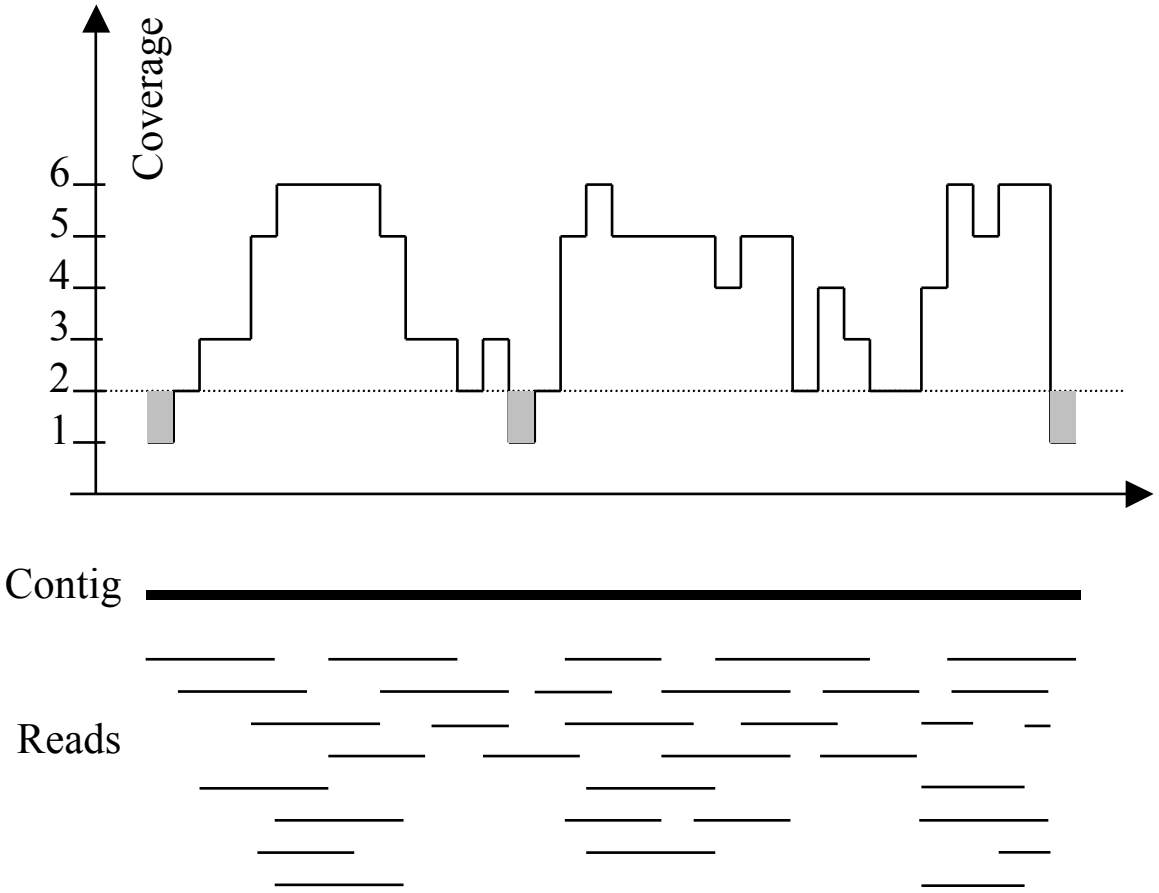
AAAAAAAAAAAAAAAAAAAA

AAAAAA AAAAAA AAAAAA
AAAAAA AAAAAA
AAAAAA AAAAAA

AAAAAA
AAAAAA
AAAAAA
AAAAAA
AAAAAA
AAAAAA
AAAAAA
AAAAAA



Impact of randomness – non-uniform coverage



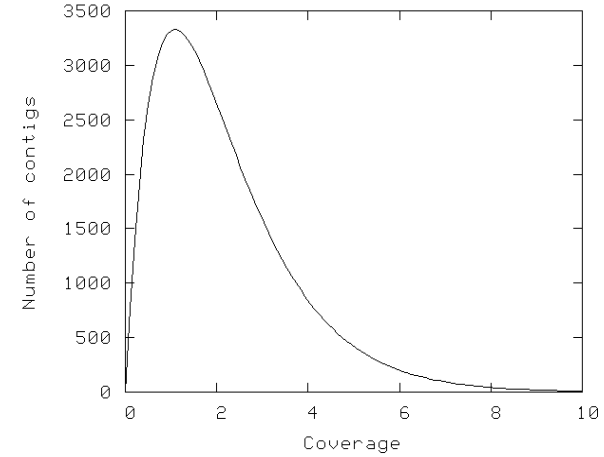
Imagine raindrops on a sidewalk

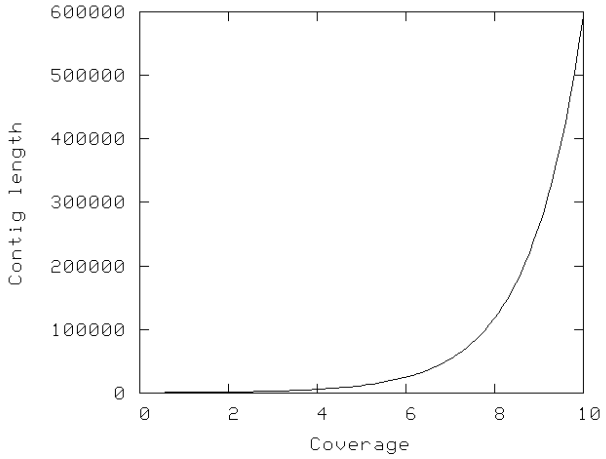
Lander-Waterman statistics

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

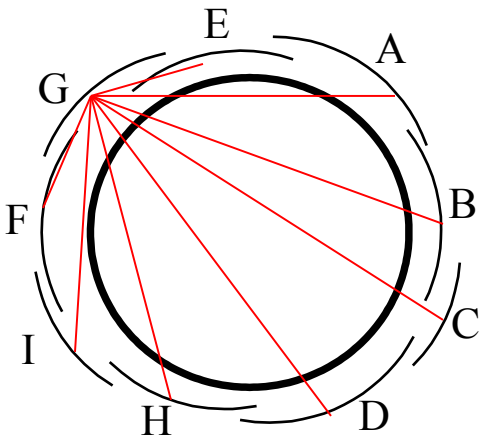
See chapter 4.5

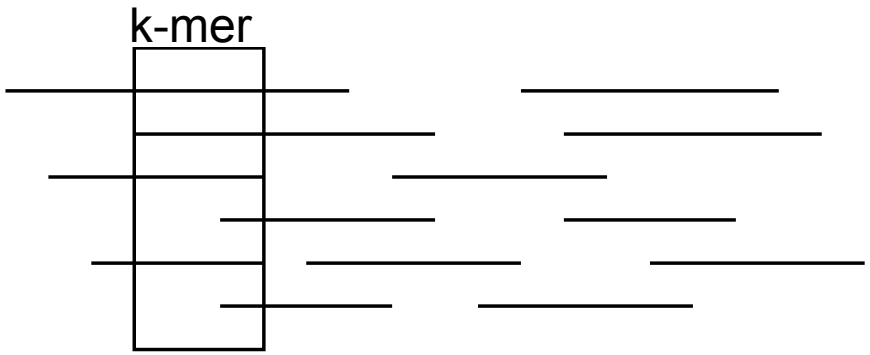




All pairs alignment

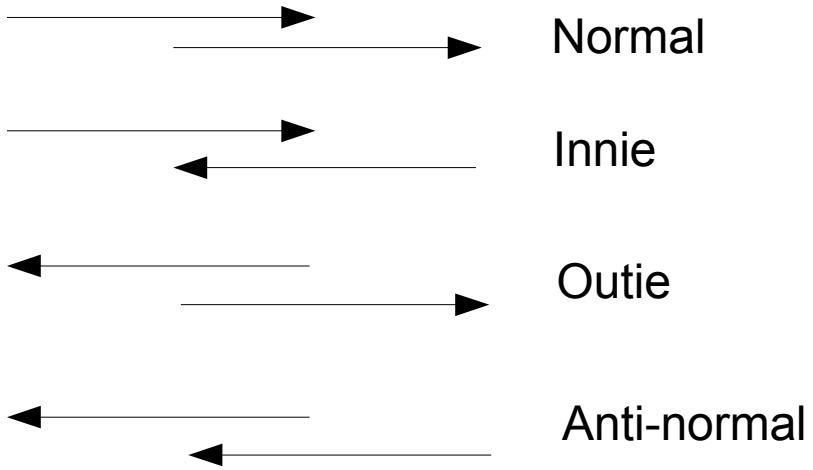
- Needed by the assembler
- Try all pairs – must consider ~ n^2 pairs
- Smarter solution: only n x coverage (e.g. 8) pairs are possible
  - Build a table of k-mers contained in sequences (single pass through the genome)
  - Generate the pairs from k-mer table (single pass through k-mer table)



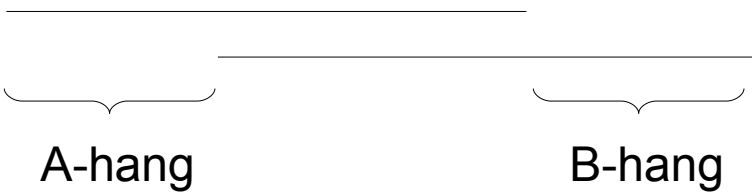


Additional pairwise-alignment details

- 4 types of overlaps
- Often – assume first read is “forward”



- Representing the alignment

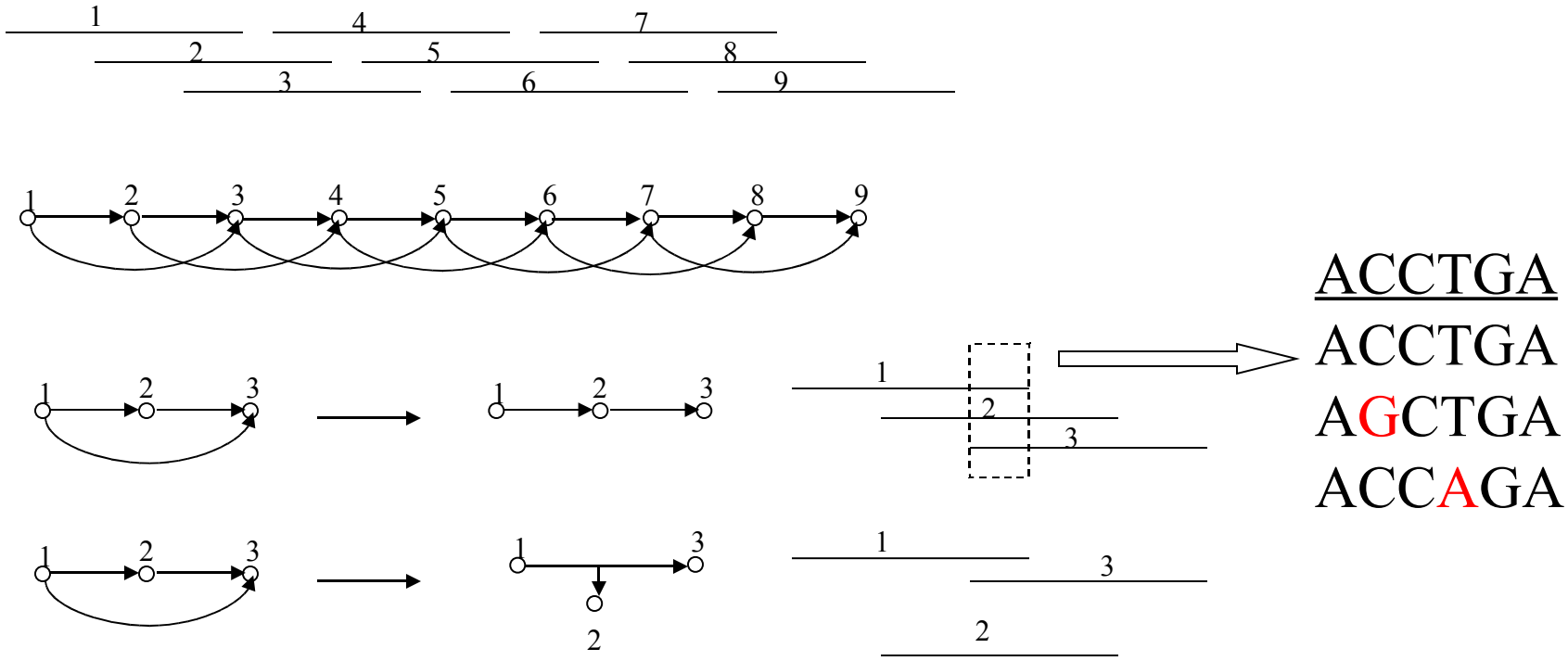


- Why not store length of overlap?

Overlap-layout-consensus

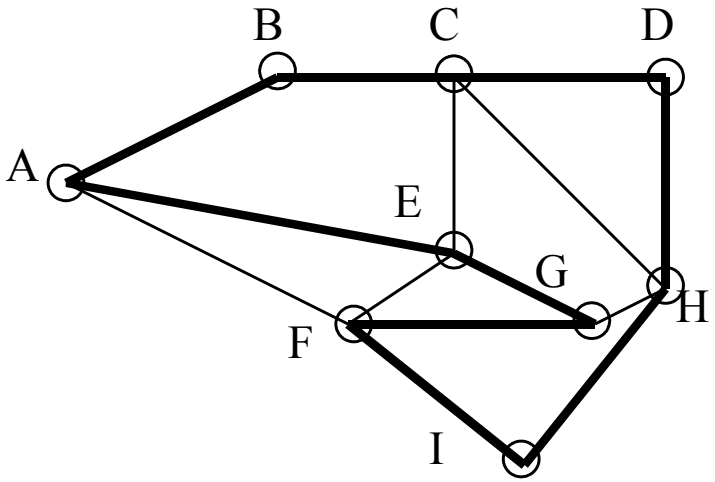
Main entity: read

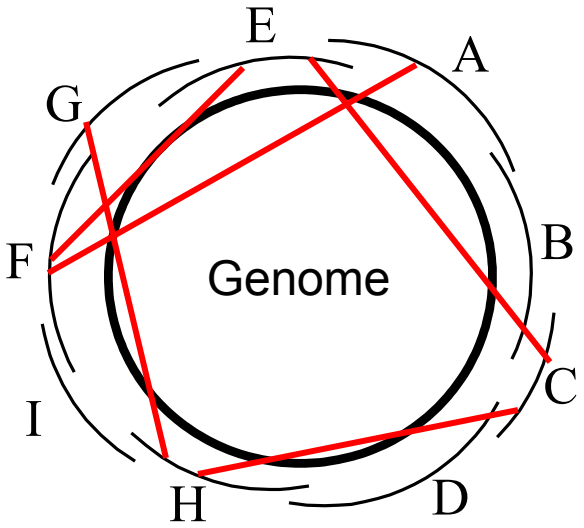
Relationship between reads: overlap



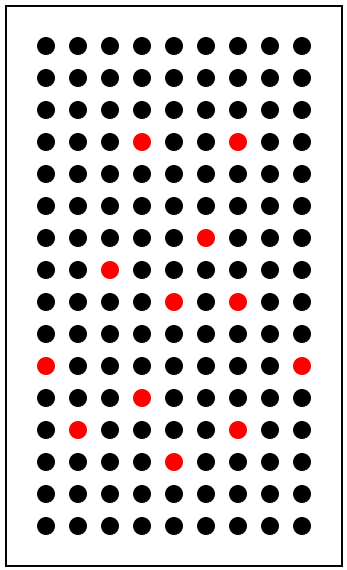
Paths through graphs and assembly

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





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

Assembling SBH data

Main entity: oligomer (overlap)
Relationship between oligomers: adjacency

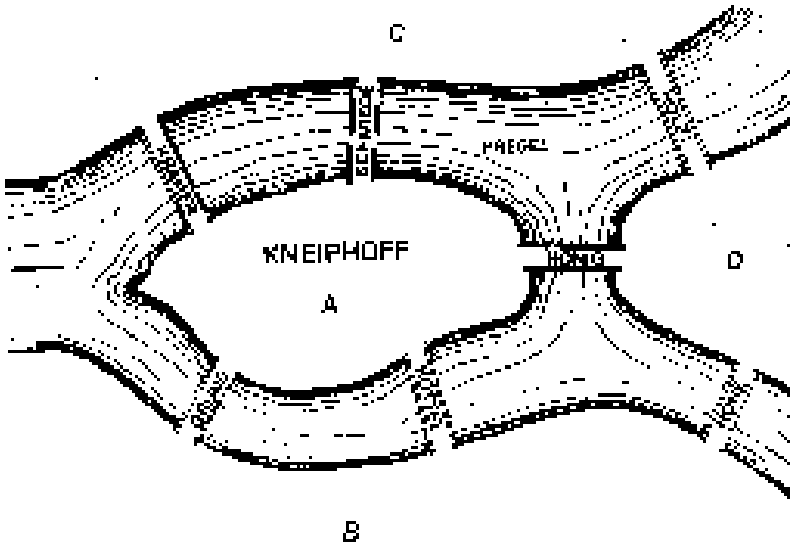
ACCTGATGCCAATTGCACT...

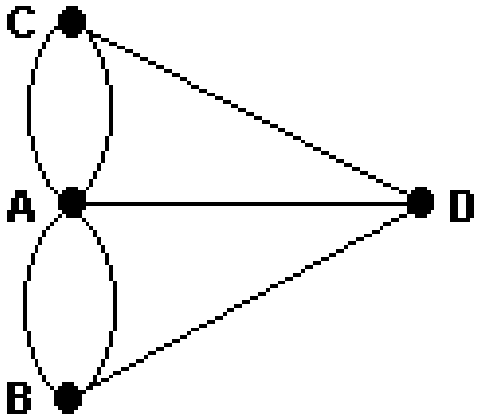
CTGAT follows CCTGA (they share 4 nucleotides: CTGA)

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

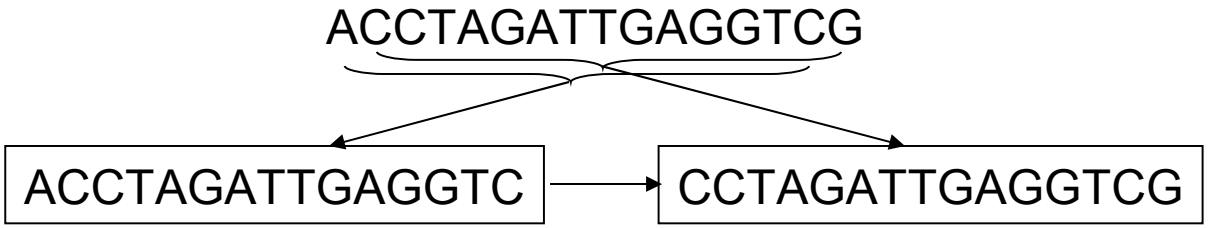
In assembly: fake the SBH experiment - break the reads into k-mers

Eulerian circuit





- Eulerian circuit: visit each edge (bridge) exactly once and come back to the start



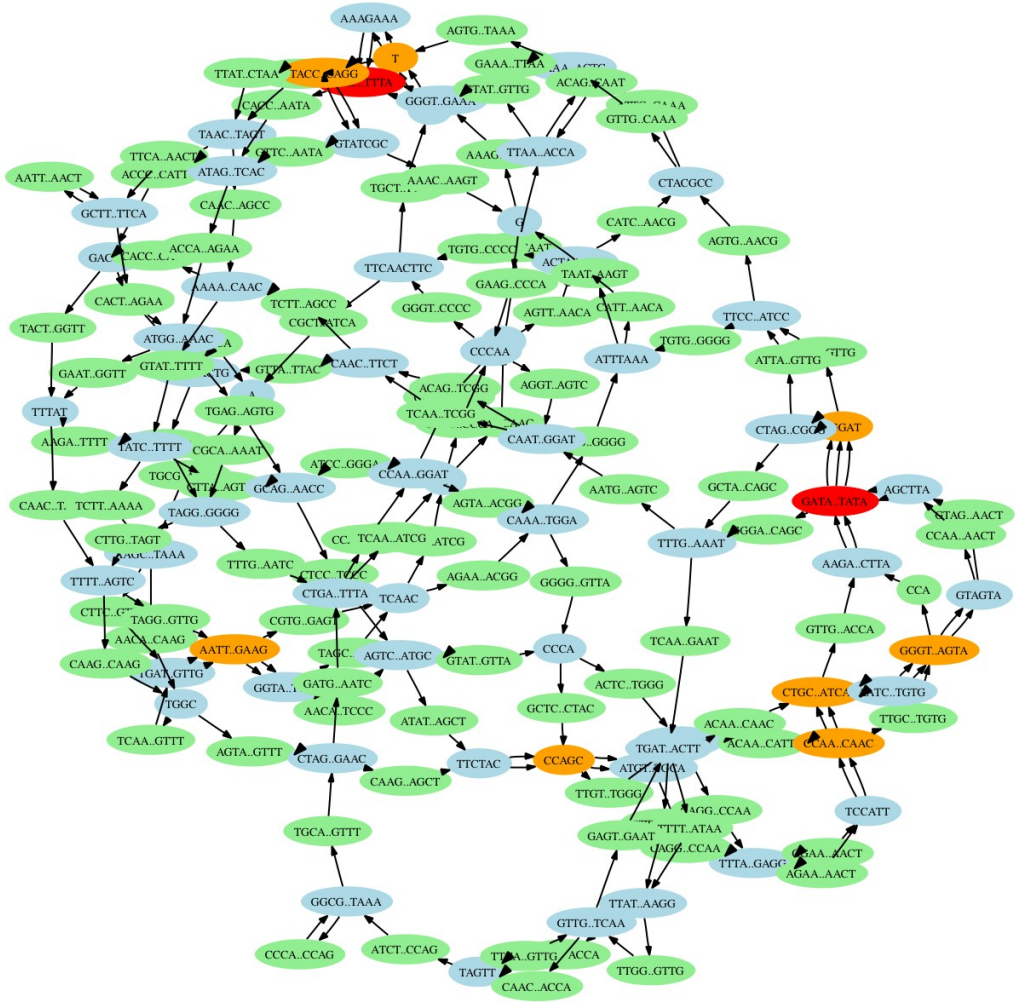
deBruijn graph

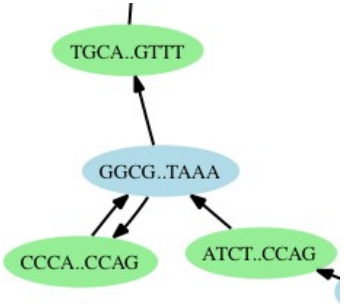
- Nodes – set of k-mers obtained from the reads
- Edges – link k-mers that overlap by k-1 letters

  ACCAGTGCA

  CCAGTGCAT
- 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*





Read-length vs. genome complexity