Genome assembly

Lars Arvestad

in DD2399♥BB2490

Technology seminar

- ***You*** present omics technologies, Wed., Jan 25
- Everyone, in groups, prepares to present a list of technologies.
- I ask some of you to present a technology
- Prepare to present using words and whiteboard!
  - How does it work?
  - What does the data look like? Features and limitations.
  - What can you do with it? Mention a paper!
  - *No* computer resources!

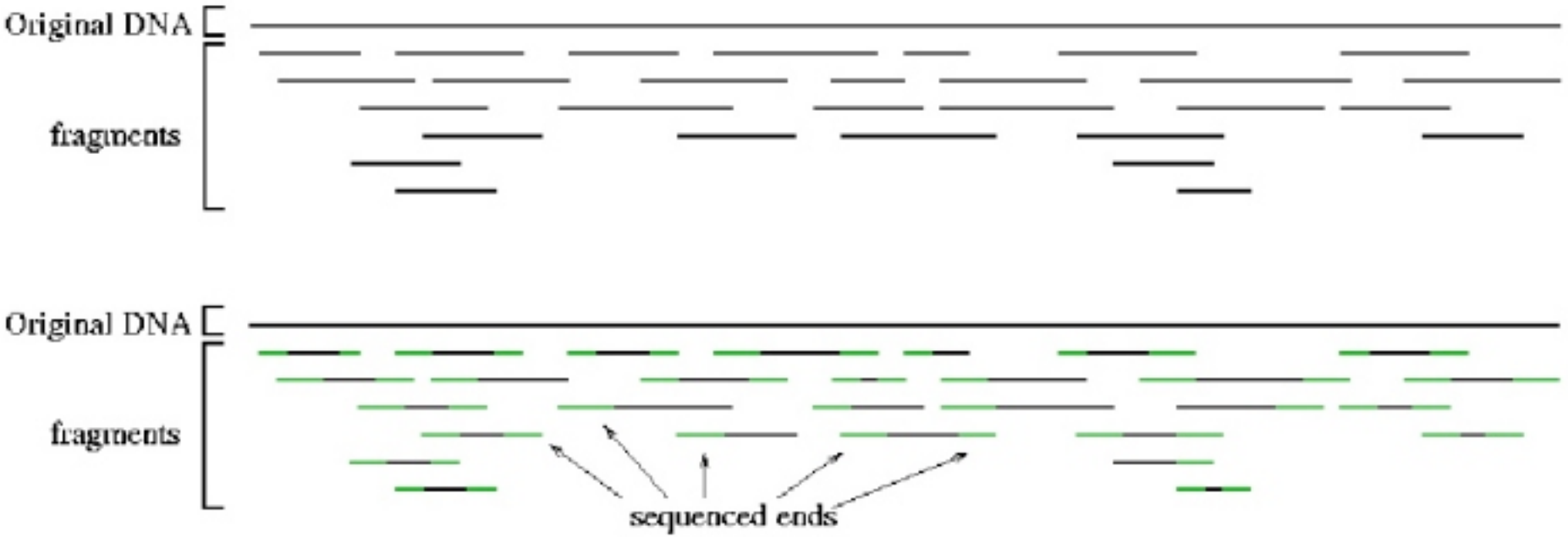
Technology seminar

- DNA sequencing:
  - Sanger sequencing, with PCR
  - “454 sequencing” using the Genome Sequencer from Roche
  - “Illumina sequencing” (previously called “Solexa”)
  - “Solid sequencing” from Applied Biosystems
- Transcriptomics:
  - Affymetrix microarray analysis
  - RNA-seq
- Proteomics:
  - Shotgun proteomics
  - Stable isotope labeling by amino acids in cell culture

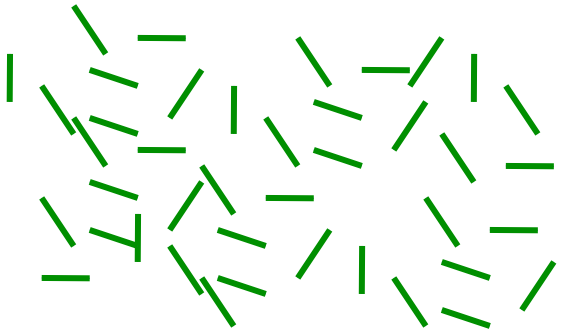
***Objective*: Reconstruct a molecule from parts**

- (Gene)
- Bacterial genome
  Circular
- Eukaryotic genome
  Size?
  Haploid/diploid/polyploidy?
  Complexity?
- Genomes from a sample
  — metagenomics

Shotgun sequencing



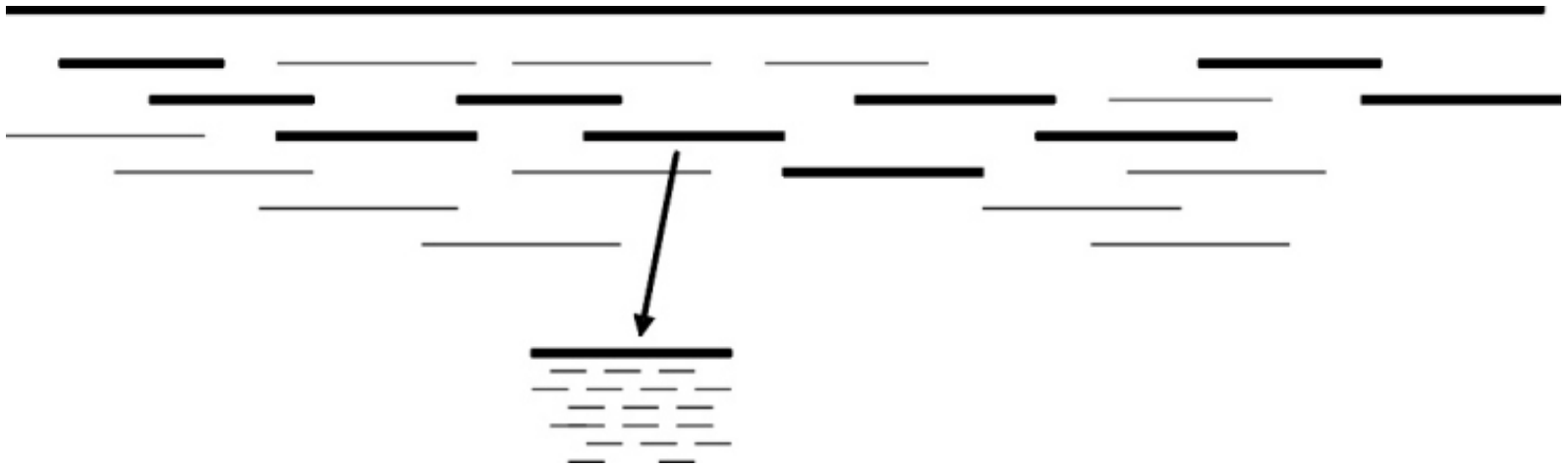
What you see:



BAC-to-BAC sequencing

- … *or* compartmental sequencing
- … *or* hierarchical sequencing

1. Break genome into large fragments, eg Bacterial Artificial Chromosomes (BACs)
2. Order the BACs and choose a ”tiling” of the genome. Requires a *mapping* of the genome!
3. Sequence the BACs



Whole-genome shotgun

- All sequencing directly on whole genomes — avoids BACs and their mapping mapping
- One huge computational problem instead of many small BAC problems

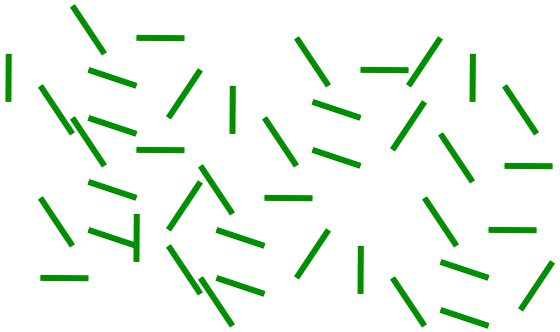
Assembly applications

- **Get models of genomes**
- **Fix problems** with genome models
  - When an assembly is wrong
  - When there is a region missing
- **Get models of genes and their transcripts**
  - From ”fresh” gene sequencing
  - From hits in NCBI’s Trace Archive: Sequencing projects deposit early

Core problem: Assemble the shotgun pieces

In:

A set of reads



Out:

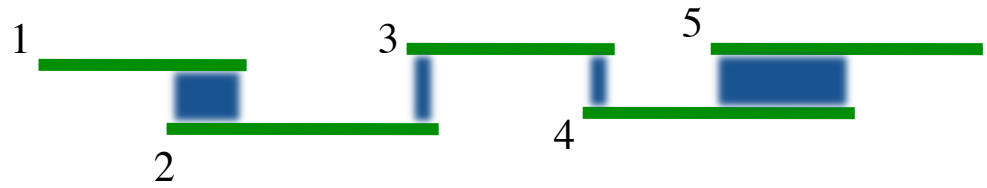
Ideally: a genome model

In practice:

A set of *contigs*

Greedy assembly

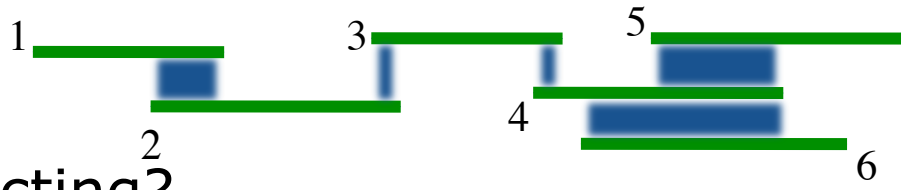
- While there are sequences with overlap:
  - Find sequences with largest overlap
  - Merge those sequences



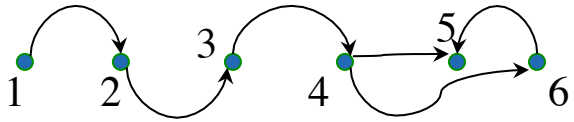
- **Advantage**:
  - Simple
- **Disadvantage**:
  - Early mistakes create bad assemblies
  - A lot of comparisons

Overlap-Layout-Consensus

- **Clean your input**
  Remove ”vector sequence”, low quality, etc



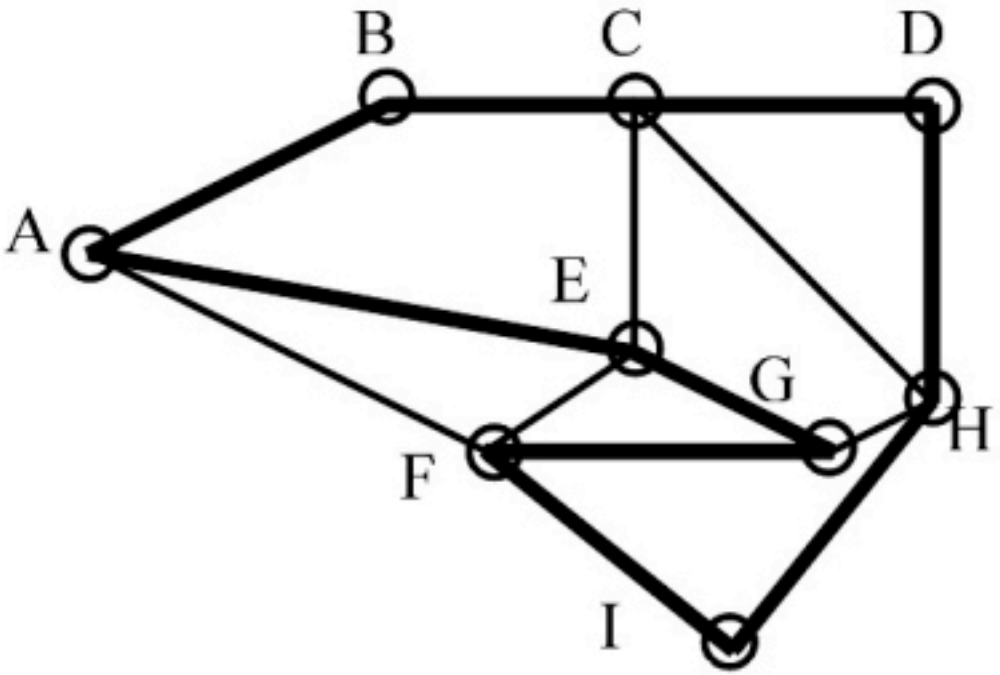
- **Overlap**: What reads are intersecting?
  - Create a node for each read
  - Create directed edge for each overlap

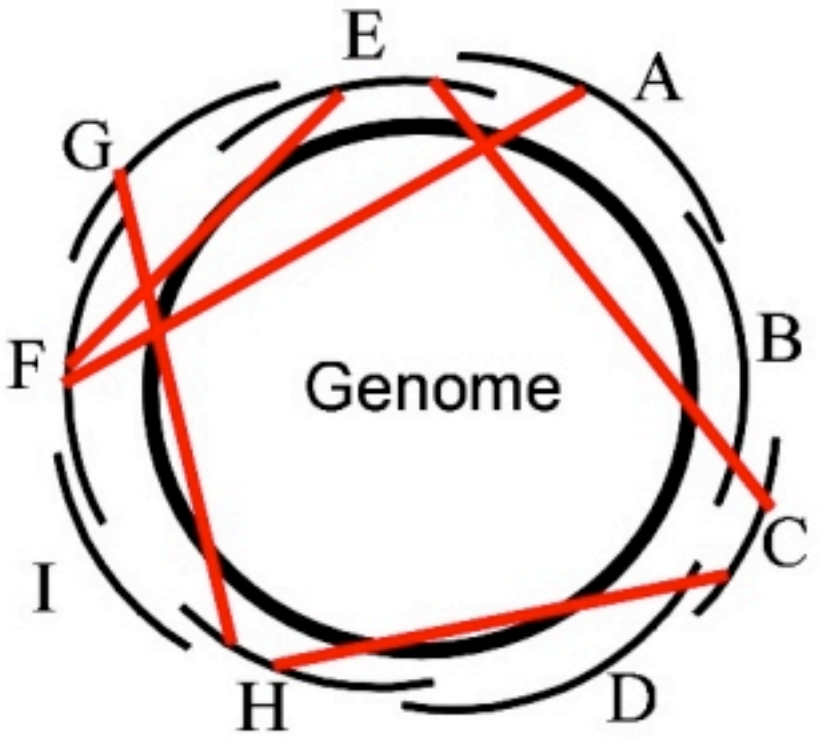


- **Layout**: How combine the reads?
  - Simplify graph
  - Find suitable paths in the graph
    - Hamiltonian

- **Consensus:** Derive contigs from layout

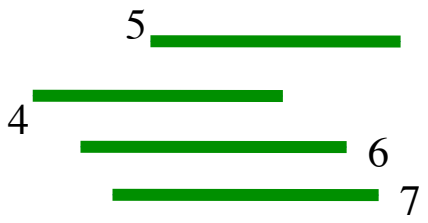
The layout stage





From http://www.cbcb.umd.edu/research/assembly_primer.shtml

Consensus stage

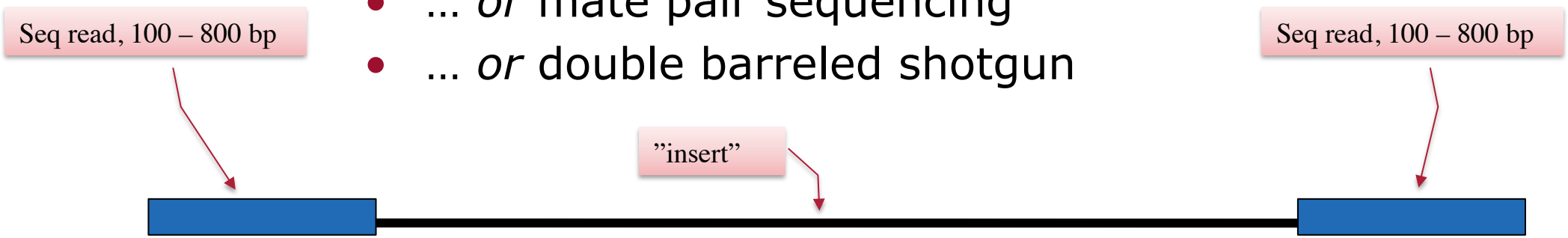



```
Seq4    TTCACACACCCTATACCAATAGTTTTCTGGCTCCTGACCATCAAACTG
Seq5                          TTTTCTGGCTCCTGACCTTCAAACTGCCTCCATATGACTGTGCTCT
Seq6                  TACCAATAGTTTACTGGCTCCTGACCCTCAAACTGCCTCC
Seq7                      ATAGTTTTCTGGCTCCTGACCGTCAAACTGCCTCCATATGA
```

```
Cons    TTCACACACCCTATACCAATAGTTTTCTGGCTCCTGACCNTCAAACTGCCTCCATATGACTGTGCTCT
```

Paired ends sequencing

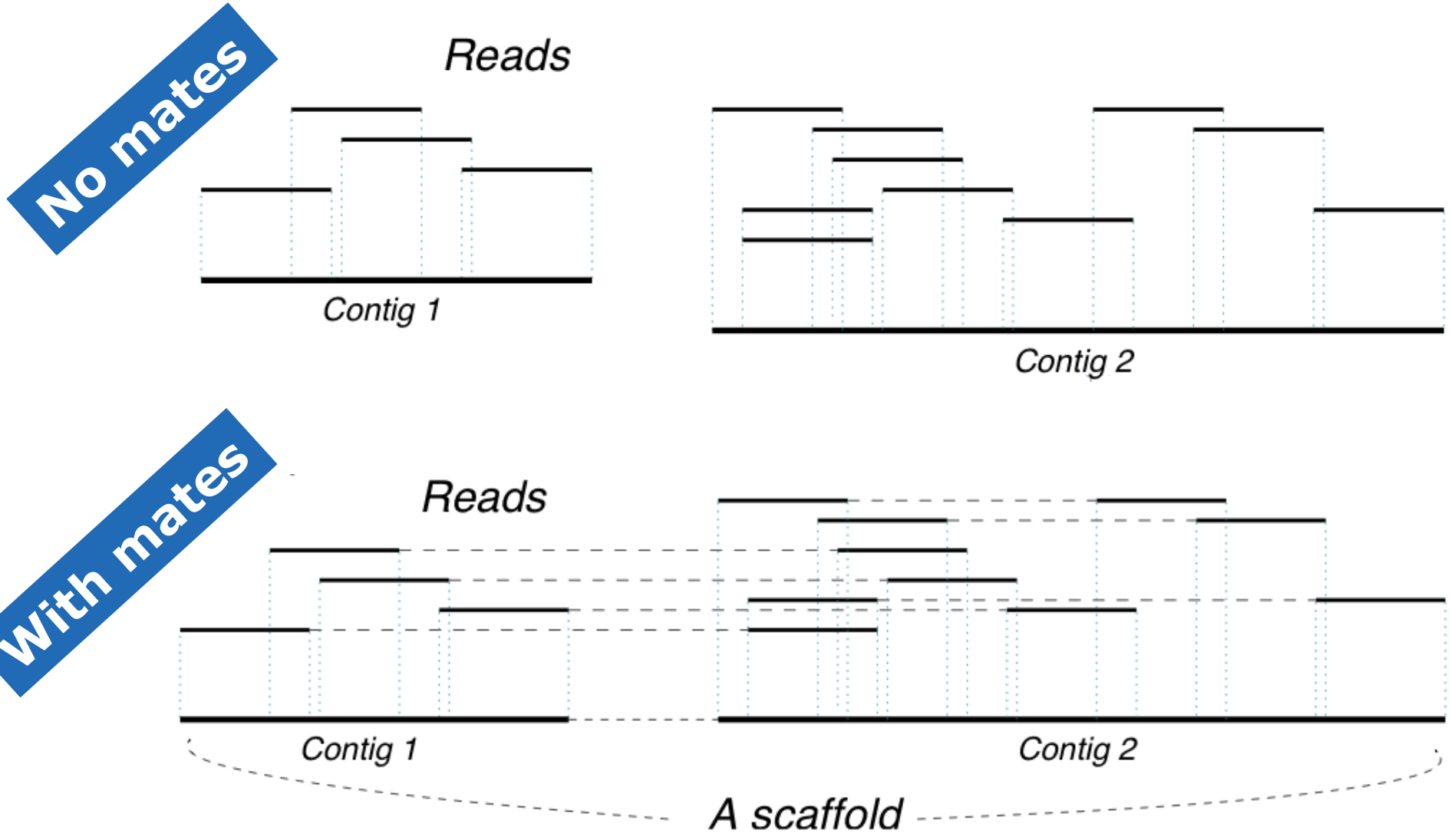
- … *or* mate pair sequencing
- … *or* double barreled shotgun



- **Advantage**: adds constraints

- Paired ends: 200 – 500 bp
- Mate pairs: 2 kbp – 10 kbp

Value of mate pairs?

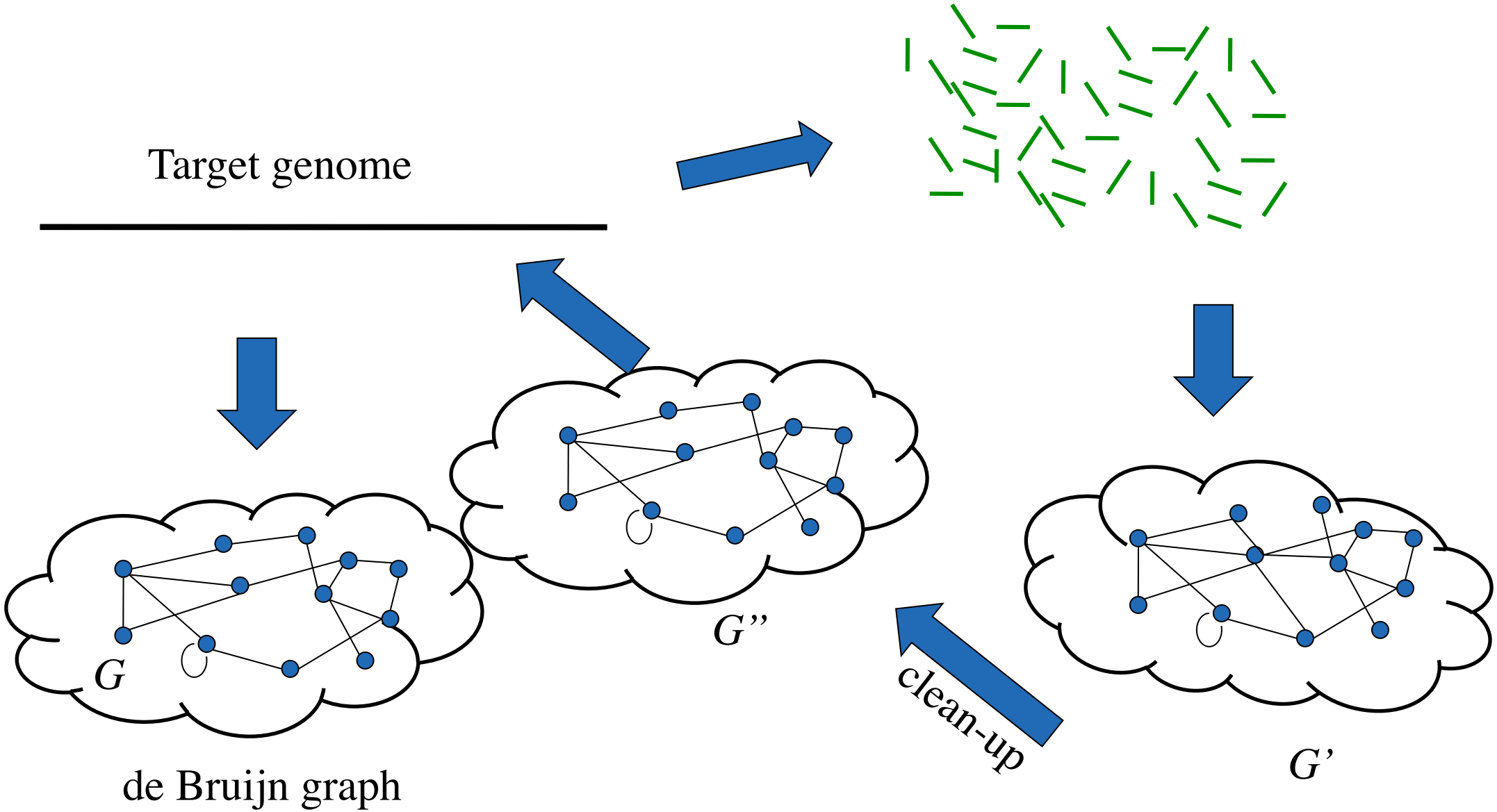


Scaffolds

- … *or* super-contigs

CCCAGTCCATTCTCCCACTGATGTCTGTAACTATATTCATCAATCTCTTT
GATTTCCAAAGCGCATACCATGGCCTCTGAATGTACTTTGCAGCTTGCCC
TTCACACACCCTATACCAATAGTTTTCTGGCTCCTGACCATCAAACTGCC
TCCATATGACTGTGCTCTTGTCTTTCCTTAGTTGCATGGGTGTCATCTTA
TGGGTCACGACCTTCTTAACTGGAACTTTCTTCTTATGGGAGCGATCCCA
TTTCTTCCAACCTCTCAAAAATTCACCCTCTTCAACAATTGACGCCTCCT
CCTTAAGATGCTCAAAATCAAAATAAAACCTAAAATCCTTCCCCTCCTGA
TCCTTCCCATCCGGATTATAATTACCTGCCAAGCATATACTCAAGTCCAT
GACAAATGTCCTGTTCAAATACATAGCCTCCCCCAACCCACACAAGAAAC
TCCACATGTAATGATTCATTCCCTTGCAATAATCCCCTCCTCTTGAATAA
TACAAGTACTTCCCTTTTCTAAAGTTCGTTTCTGATCNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNCCTCTCAAACTATCGTGAGTACGGGATTATATTTCTAAAGTAAGTAAA
TTATCCAAGATGGGCCGATCCATTACGAAAAGGAAAGATAATCCACTATT
ATTTCTTAATATAAGGGCAAATTTAAATATATAAAGATGTAATTGTTGTT
GGCGAGTGCCTCTCTTGGTTGTTGAGAGTGAAATTGACAGCAAAGTTGTA
GATTGTGACAGCCAATGTAACTTATTCACAAATTGGCCTGCCAATGGTAC
ATCATGAATCGCTATGCCACATACTTGATTATACCTTCTAAAGTACCTGT
GAATTTTATTTATTTTTTCCATTTTAAAAGTATGTTTTATTTGGAAAAAA
TATCAAATTATTATTTTACTATTATATTTTAATATATTCTTAAATAAAAA
ACACTATTAAAAAATATTATGCACCGCAATAATAAACACATATTAAACAA
ACAGATAATTTTTATATGGCATTTCACTATTGTTGTGGAATAATATCTTT

The Euler approach



Genome coverage

- … or read depth
- … or coverage depth
- … or redundancy

How many times is a position sequenced, on average?

- Drosophila: C=14x
- Human: >7x
- Dolphin: 2.59x
- Mouse lemur: 1.93x

Lately

- Korean: 28x
- Panda: > 50x

How good coverage do you need?

- High coverage good, but expensive
- What if I want at least 99 % of the genome?

Lander-Waterman model

- **Assumption**: Reads are uniformly distributed
- Coverage C
- #times position i sequence: X_i
- X_i is Poission distributed

$$\Pr(X_i = k) = C^k e^{-C}/k!$$

More Lander-Waterman

Require $0 < \theta < 1$ overlap to join reads into a contig.

- Expected number of contigs if N reads: $Ne^{-C(1-\theta)}$

  Dog: 8x, require e.g. 10% overlap, 32×10^6 *reads:*

  24 000 contigs

- Expected contig size: $L\frac{e^{C(1-\theta)}-1}{C} + \theta$.

  Dog, assume $L = 500$*: contigs are* $\sim$ *83 700 bp*

Lander-Waterman and reality

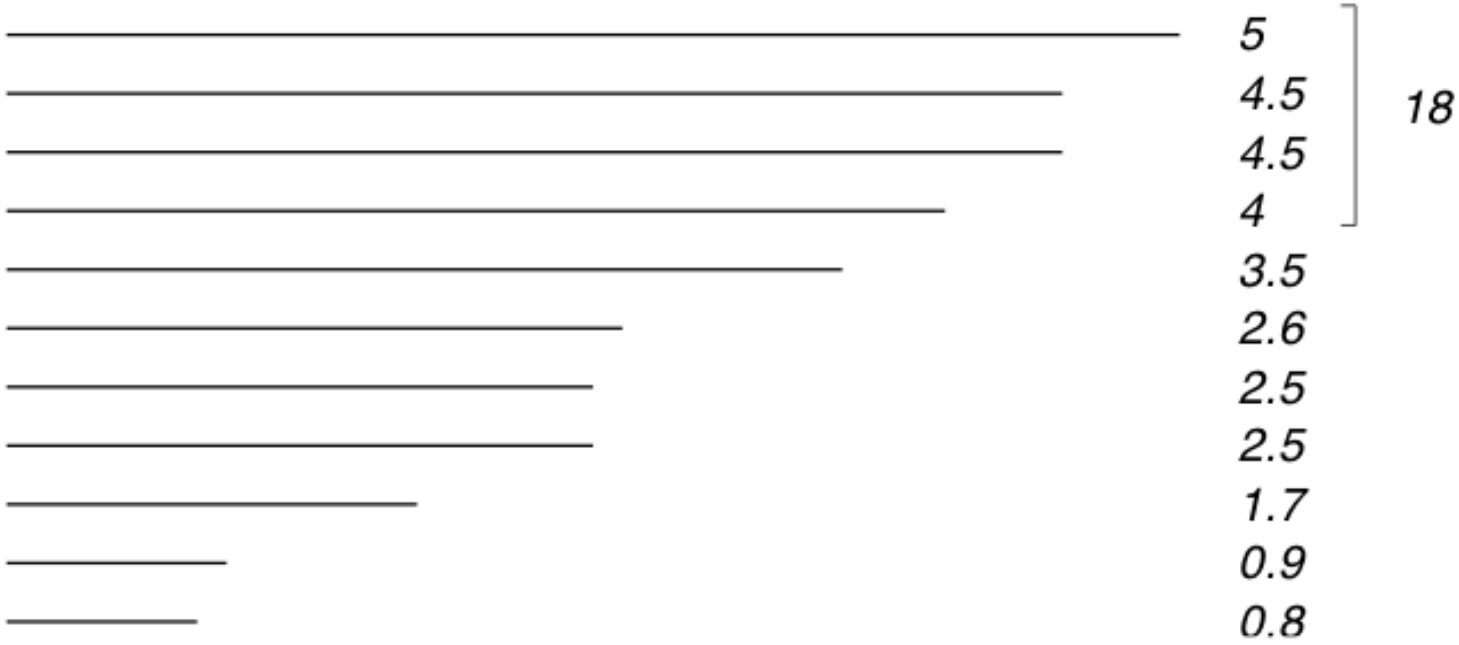
> ”For both a simulated unassisted 2x mouse genome assembly (Margulies et al. 2005) and the assisted 1.9x cat genome assembly of Pontius et al. (2007) euchromatic genome coverage by assembled contigs was only 65%, significantly less than the theoretical Poisson expectation (Lander and Waterman 1988) of 85%.”
>
> *Green, 2007*

- Why this discrepancy?

Quality: N50

N50: "covering half the assembly"
NG50: "covering half the actual genome"
Scaffold N50: Looking scaffolds, not contigs

- Operational definition:
  1. Sort all contigs by size
  2. Add contig sizes, one by one, towards the smallest
  3. Stop when you have contigs covering half the genome
  4. The length of the last contig is the N50
- Given contigs from a 30 Mbp genome:



N50 is 4 Mbp, because 5+4.5+4.5+ 4 > 30

N50 characteristics

- High N50 ⇒ Good contigs ⇒ Good assembly
- Low N50 ⇒ Many small contigs ⇒ Genome badly sequenced ⇒ Bad assembly

- Bad assembly could have a high N50:

> ”The standard of judging assembly quality by size of contigs is questionable. Large contigs may simply reflect overly aggressive joining of contigs, thereby creating larger contigs with mis-assemblies. As a consequence, genome scientists who are not experts at assembly can be completely misled by statistics about contig sizes, and as a result might prefer the ’larger’ but incorrect assembly when given a choice.”
>
> Salzberg & Yorke, 2005

Assembly resources

- NCBI's Trace Archive
- Lots of assemblers
    - Cap3
    - Phrap
    - Minimus (paper)
    - Velvet
    - AllPaths
    - Abyss
    - … and many more

Student presentations

Half of next week's Tuesday lecture!

1. de Bruijn graph based assembly
   - Based on Pop's review paper (and references therein if needed!)
   - What is the basic graph construction?
   - How do you find an assembly in a de Bruijn graph?
2. Problems and challenges in assembly
   - Based on Salzberg and Yorke's "Beware of mis-assembled genomes"