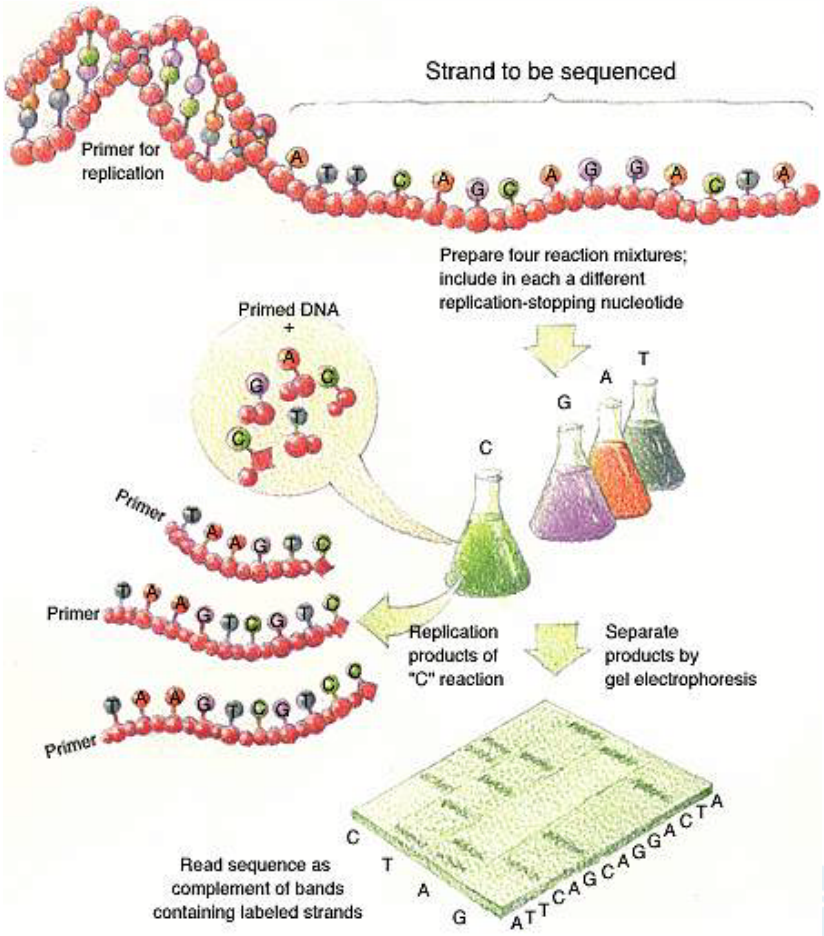
Lecture 14: DNA Sequencing and Assembly

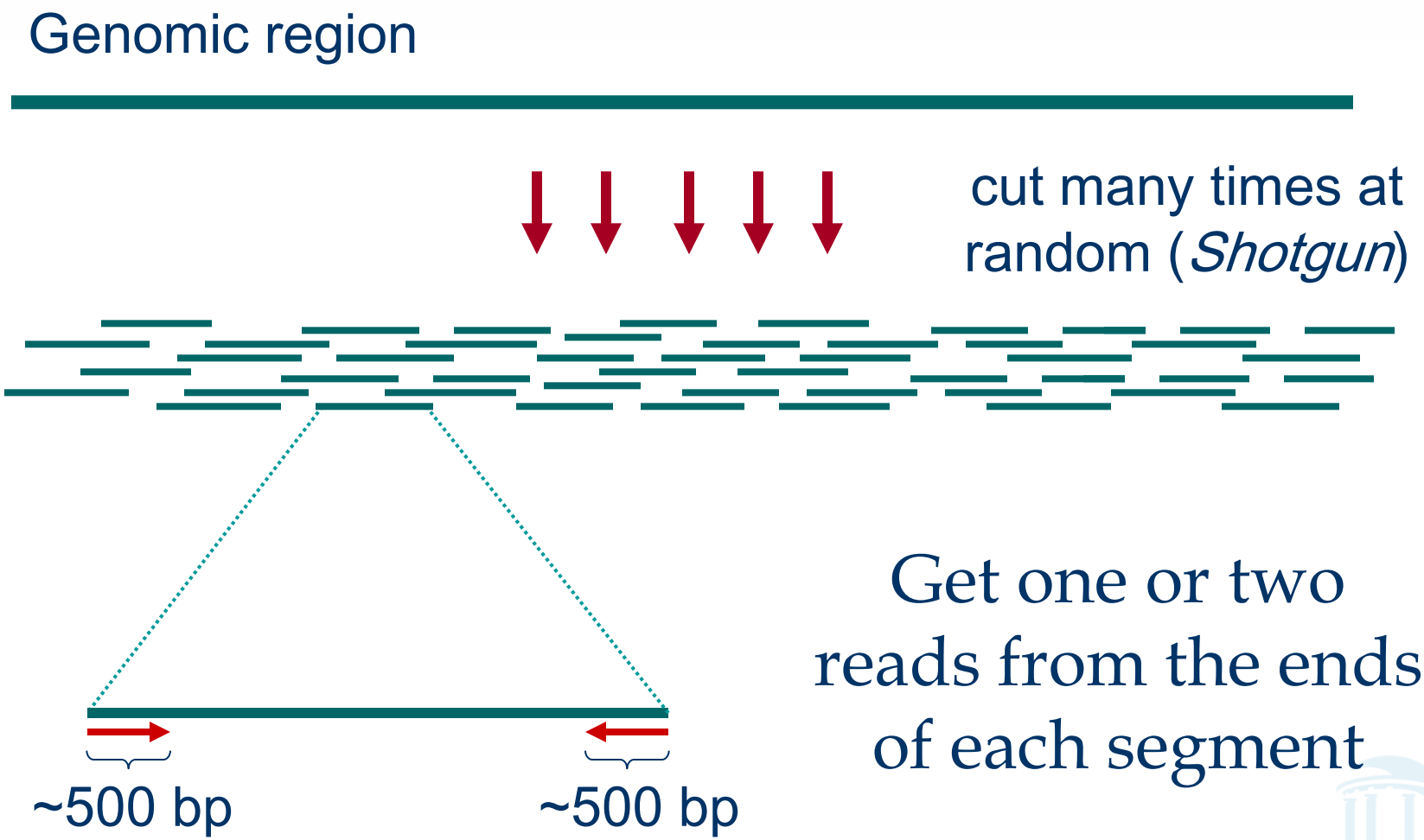
Study Chapter 8.9

DNA Sequencing

- Shear DNA into millions of small fragments
- Read 500 – 700 nucleotides at a time from the small fragments (Sanger method)



Shotgun Sequencing



Get one or two reads from the ends of each segment

Current sequencing technologies

High throughput

Illumina HiSeq 2500 (8 @ UNC)
2 x 100 bp reads
11 days for 16 samples
~35 GB per sample (12x coverage)

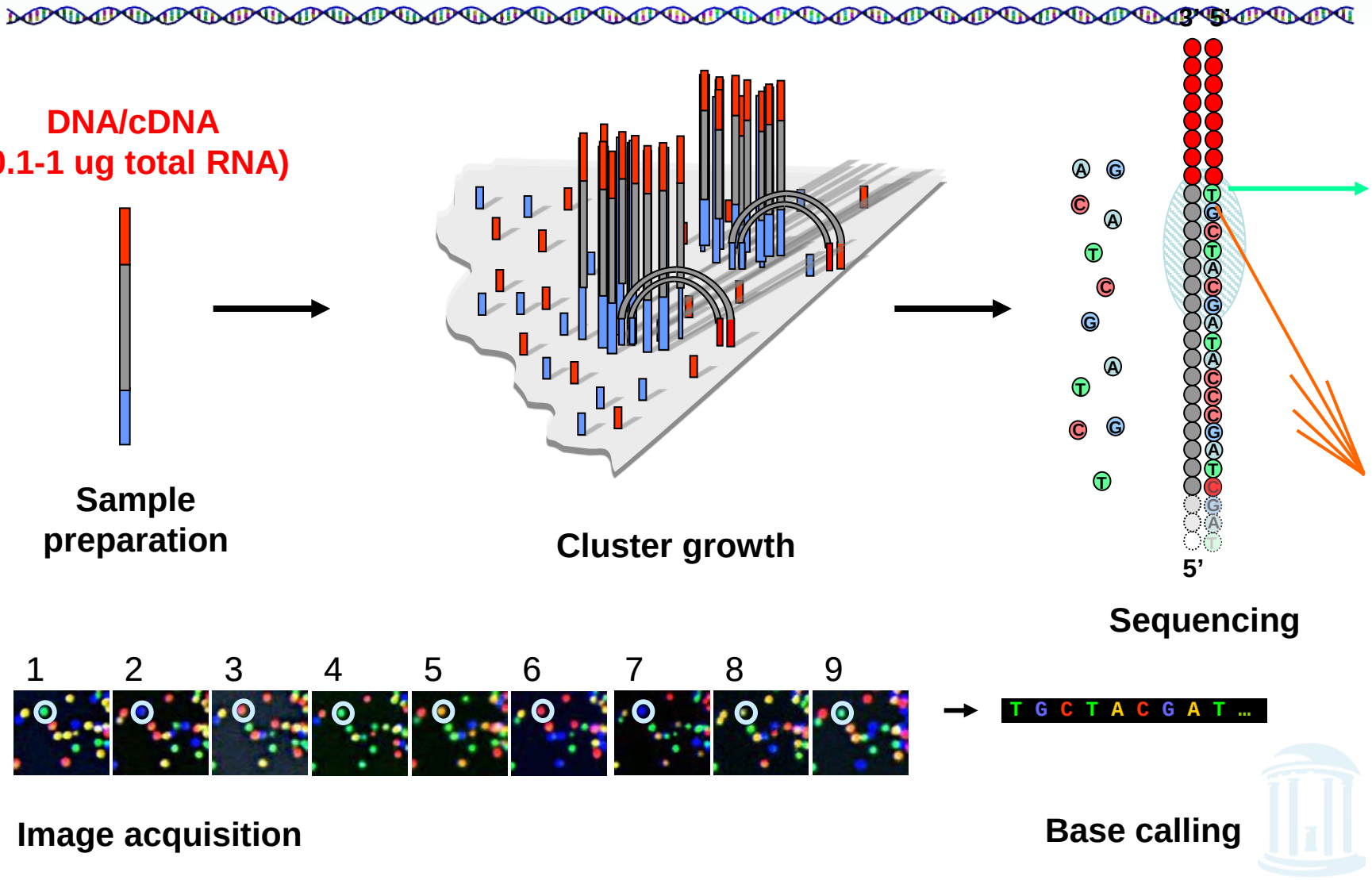
Pacific Biosciences (1@UNC)
1000-10,000 bp reads
20 min, 200 MB

Individual labs

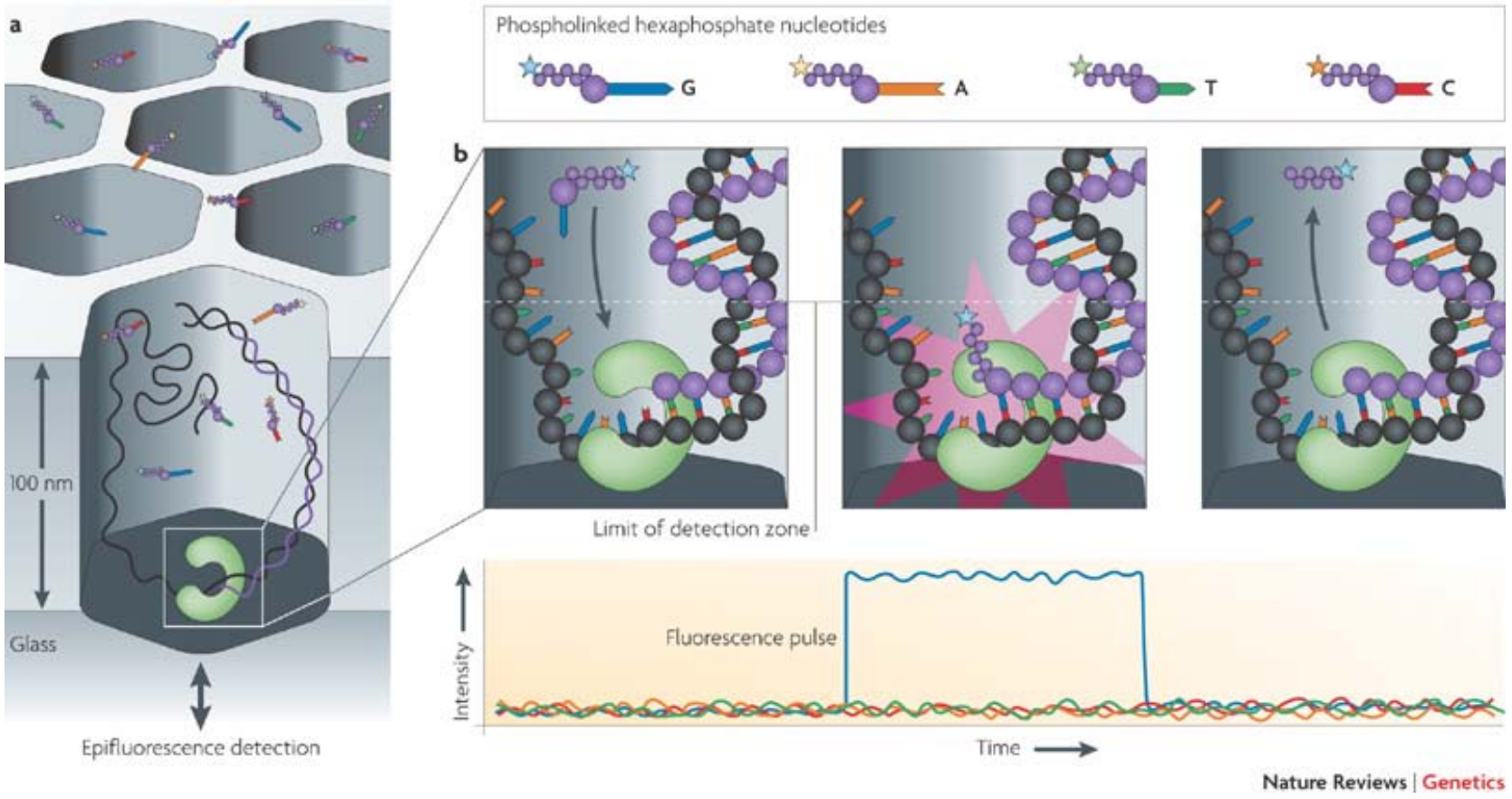
Illumina MiSeq
2x250 bp reads
20 hours, 1 GB per day

Life Technologies Ion Torrent
2 hours
~100 MB to 3 GB

Illumina reversible dye terminator chemistry



Pacific Biosciences Single-Molecule Real-Time sequencing



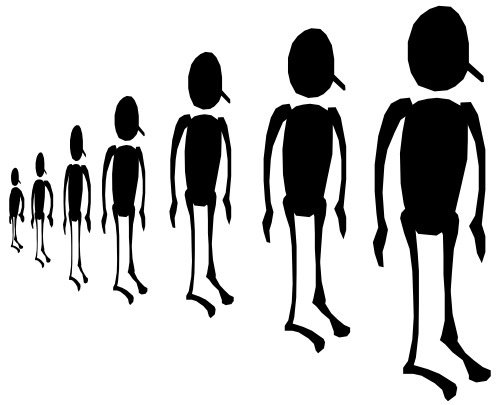
- No PCR steps are required
- Mutated polymerase has slower base incorporation (1-3 bp per second)
- Read lengths > 1 kb, but a high error rate (~15%)

Metzker ML (2010) *Nat Rev Genet*

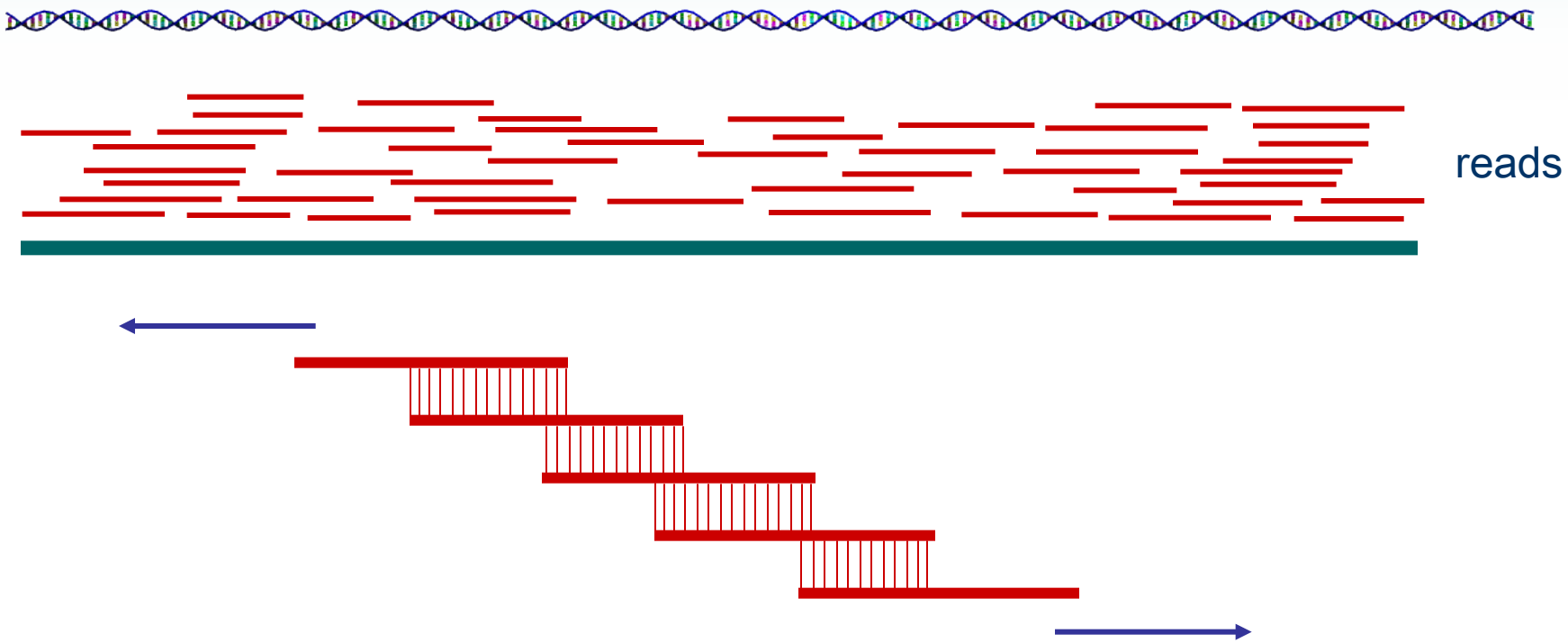
What do we do with these reads?

Fragment Assembly

- Assembles the individual overlapping short reads (fragments) into a genomic sequence
- Shortest Superstring problem is an overly simplified abstraction
- Problems:
  - DNA read error rate of 1% to 3%
  - Can't separate *coding* and *template* strands
  - DNA is **full** of repeats
- Let's take a closer look



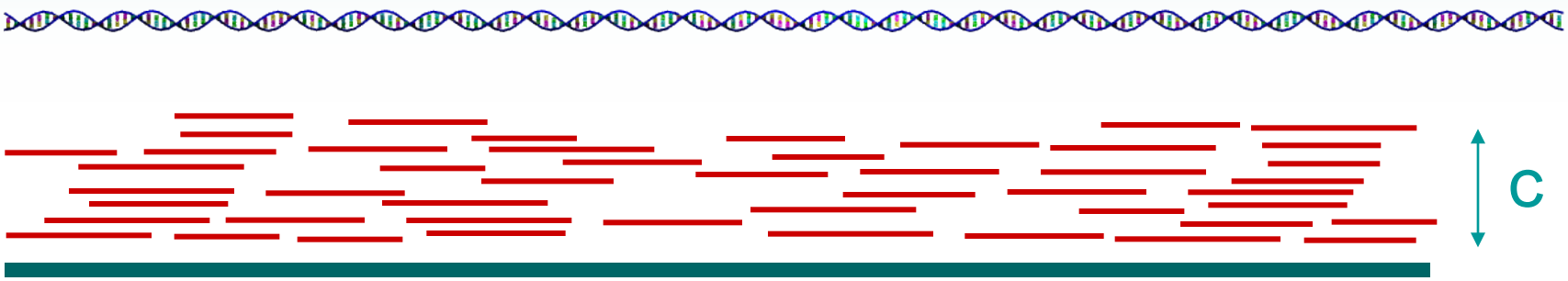
Fragment Assembly



Cover region with ~7-fold redundancy

Overlap reads and extend to reconstruct the original genomic region

Read Coverage



Length of genomic segment: L

Number of reads: n

Length of each read: l

Coverage $C = n\,l / L$

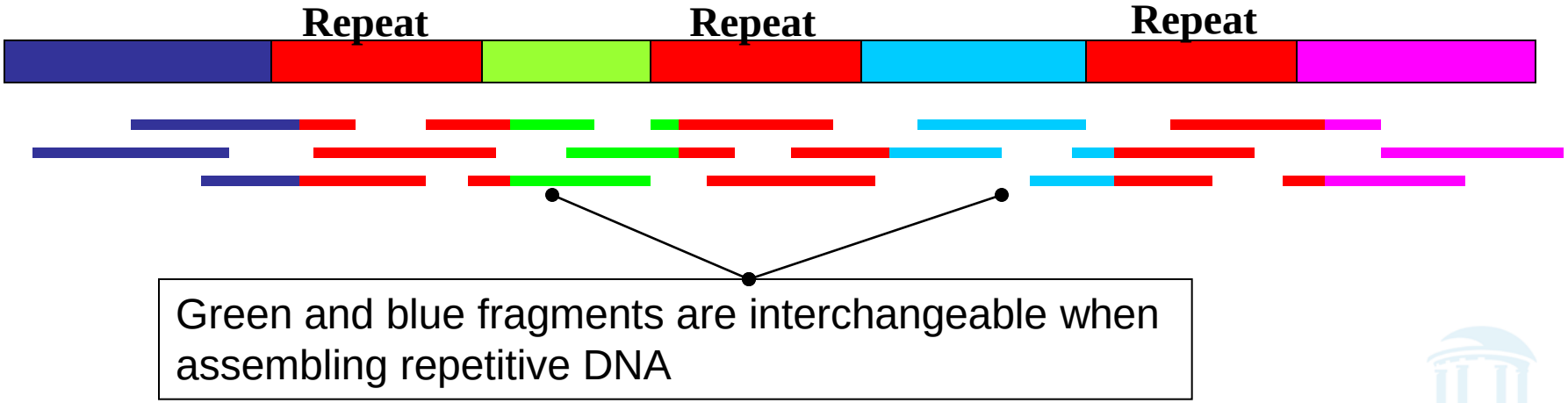
How much coverage is enough?

Lander-Waterman model:

Assuming uniform distribution of reads, C=10 results in 1 gapped region per 1,000,000 nucleotides

Challenges in Fragment Assembly

- > 50% of human genome is **repeats**:
  - over 1 million *Alu* repeats (about 300 bp)
  - about 200,000 LINE repeats (1000 bp and longer)
- Repeats are a **major** problem for fragment assembly
  - assume reads are 100bp and we have 300bp repeats



Types of Genome Assemblies

- De Novo –
  An assembly based entirely on self-consistency or self-similarity of short reads (contigs).

- Comparative –
  An assembly of a genome using the sequence of a close relative as a reference. Sometimes called a "template assembly" or "resequencing"

- Confounding problem for both types: Repeats

Repeat Types

- **Low-Complexity DNA** (e.g. ATATATATACATA…)
- **Microsatellite repeats** $(a_1 \ldots a_k)^N$ where k ~ 3-6
  (e.g. CAGCAGTAGCAGCACCAG)
- **Transposons/retrotransposons**
  - **SINE** Short Interspersed Nuclear Elements
    (e.g., *Alu*: ~300 bp long, $>10^6$ in human)
  - **LINE** Long Interspersed Nuclear Elements
    ~500 - 5,000 bp long, > 200,000 in human
  - **LTR retrotransposons** Long Terminal Repeats (~700 bp) at each end
- **Gene Families** genes duplicate & then diverge
- **Segmental duplications** ~very long, very similar copies

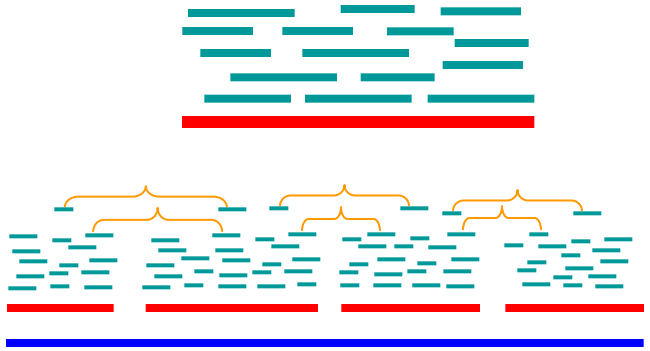
Overlap-Layout-Consensus Assembly

Assembler programs ARACHNE, PHRAP, CAP, TIGR, CELERA

Common Approach:

Overlap: find potentially overlapping reads

Layout: merge reads into **contigs** and then combine contigs into **supercontigs**



Consensus: requires many overlapping reads to derive the DNA sequence and to correct for read errors

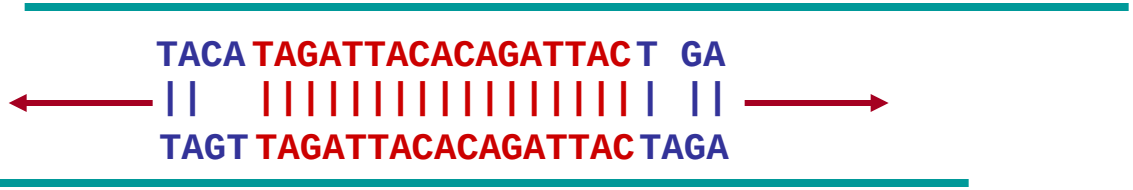
..ACGATTACAATAGGTT..

Overlap

- Find the best match between the suffix of one read and the prefix of another (shortest superstring)
- Due to sequencing errors, most algorithms use dynamic programming to find the optimal *overlap alignment*
- Filter out fragment pairs that do not share a significantly long common substring

Overlapping Reads

- Make an index of all *k*-mers of all reads
  (*k* ~ 20-24)
- Find read-pairs sharing a k-mer
- Extend alignment –
  throw away if not >95% similar



Histogram Similarity

v = tagattacacagattattga

- Histogram of 3-mers (18 total)

	A_2	C_2	G_2	T_2
	A_3:C_3:G_3:T_3	A_3:C_3:G_3:T_3	A_3:C_3:G_3:T_3	A_3:C_3:G_3:T_3
A_1	0:0:0:0	2:0:0:0	2:0:0:0	0:0:0:3
C_1	0:1:1:0	0:0:0:0	0:0:0:0	0:0:0:0
G_1	0:0:0:2	0:0:0:0	0:0:0:0	0:0:0:0
T_1	0:1:1:1	0:0:0:0	1:0:0:0	2:0:1:0

Overlapping Reads and Repeats

- Does this really speed up the process?
- A *k*-mer that appears N times, initiates N^2 comparisons (you consider all pairs of reads that share the k-mer substring)
- For an *Alu* that appears 10^6 times → 10^{12} comparisons – too much
- **How to avoid repeats:**

  Discard all *k*-mers that appear more than

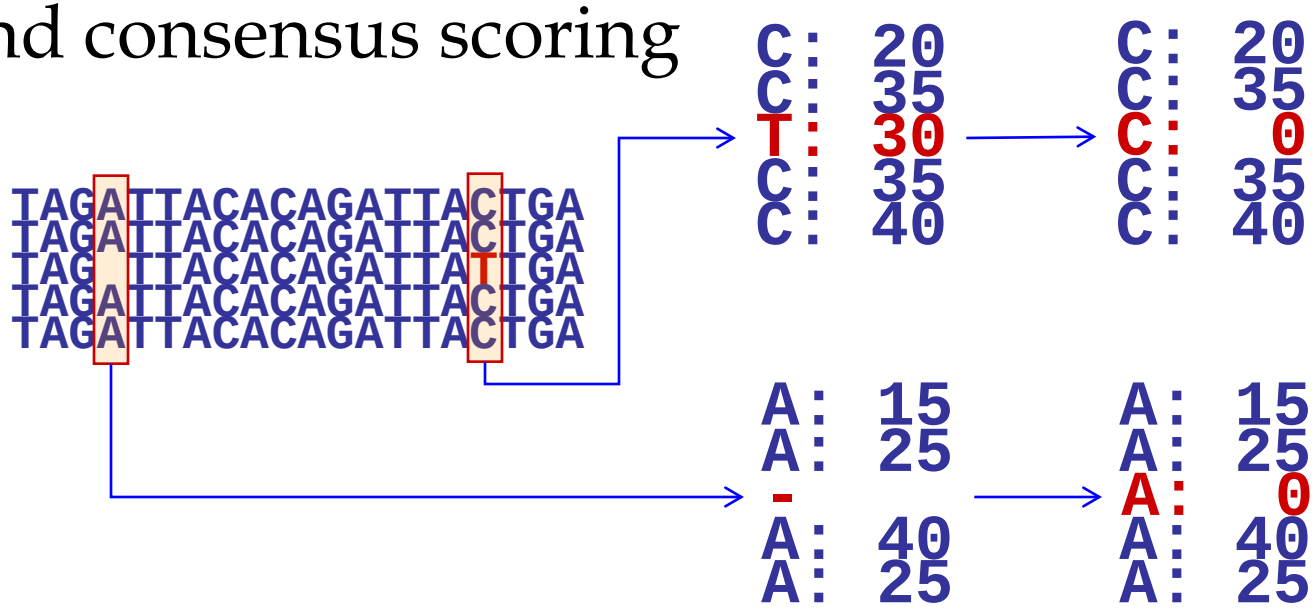
  $t \times$ Coverage, ($t \sim 10$)

Finding Overlapping Reads

k-mer table makes it easy to create local multiple alignments from the overlapping reads

Finding Overlapping Reads (cont'd)

- Correct errors using multiple alignment and consensus scoring



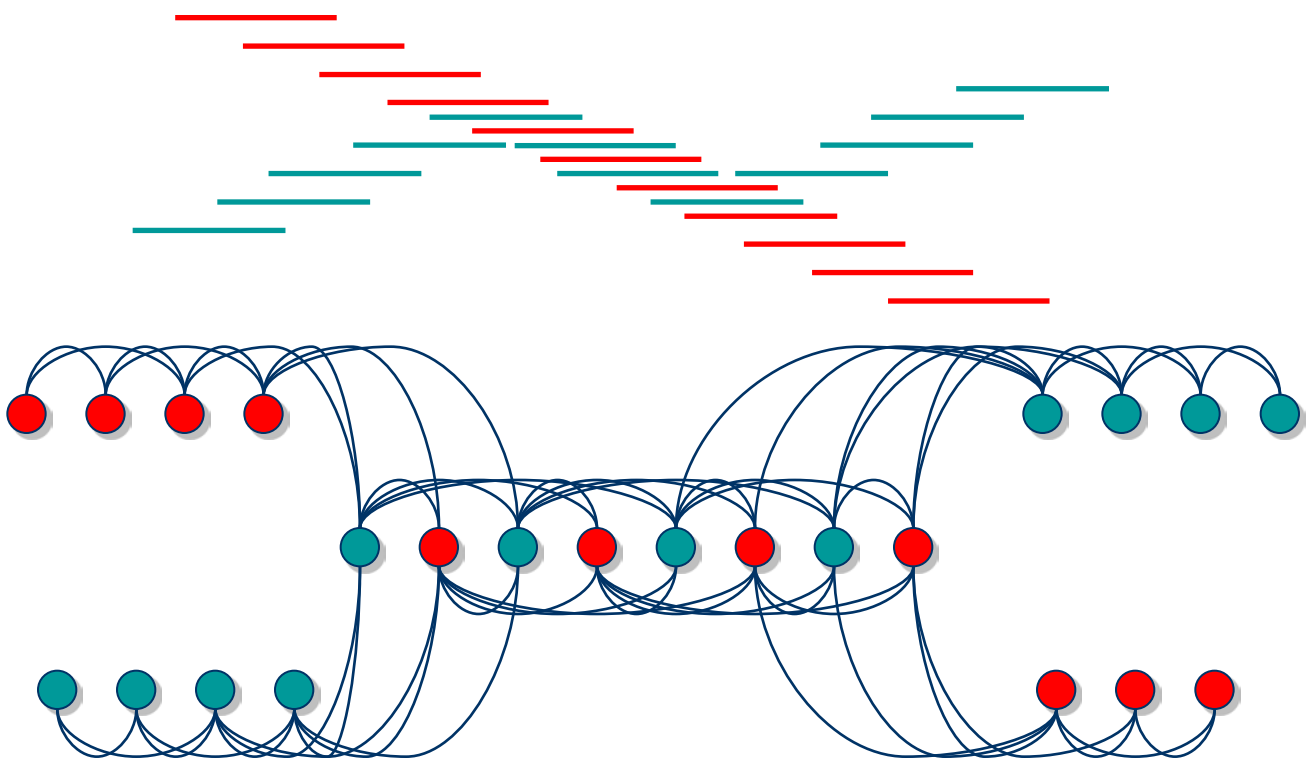
- Score alignments
- Accept alignments with good scores

Layout

- Repeats are still a major challenge
- Do two aligned fragments really overlap, or are they from two copies of a repeat?
- Solution: repeat masking – hide the repeats?
  - Masking results in high rate of misassembly (up to 20%)
  - Misassembly means alot more work at the finishing step

2. Merge Reads into Contigs

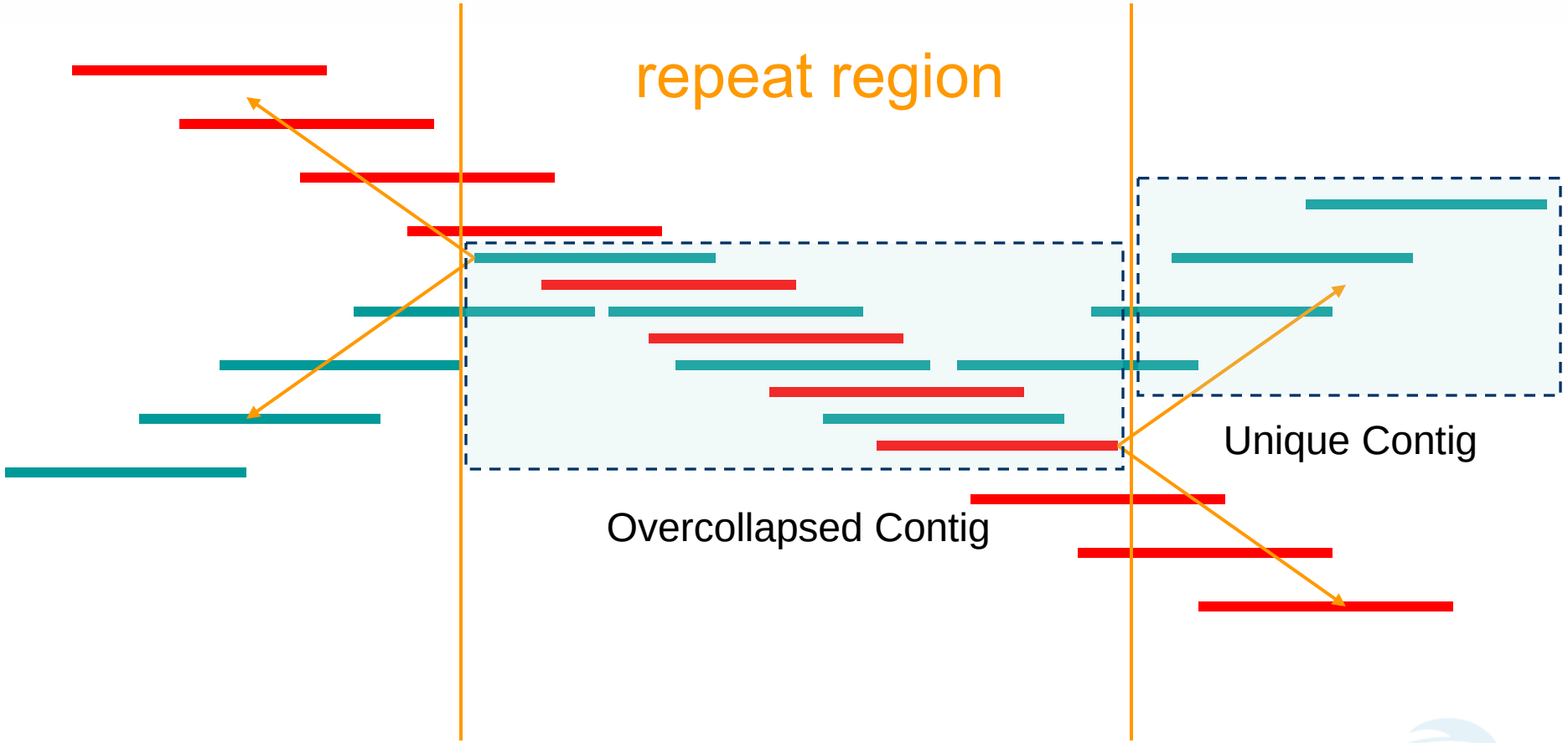
- Overlap graph:
  - Nodes: reads r_1…..r_n
  - Edges: overlaps (r_i, r_j, shift, orientation, score)



Reads that come from two regions of the genome (blue and red) that contain the same repeat

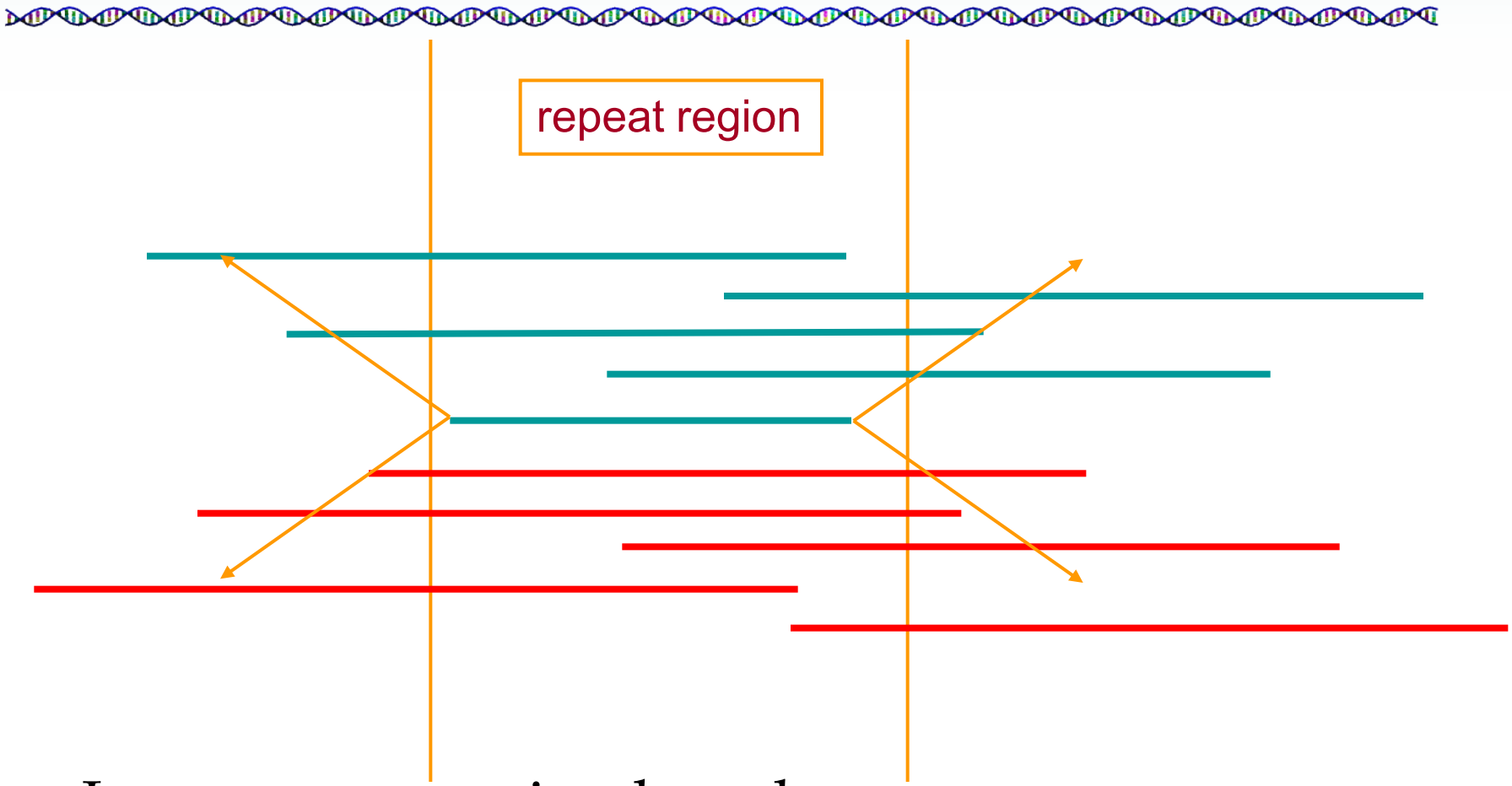
Note:
of course, we don't know the "color" of these nodes

2. Merge Reads into Contigs



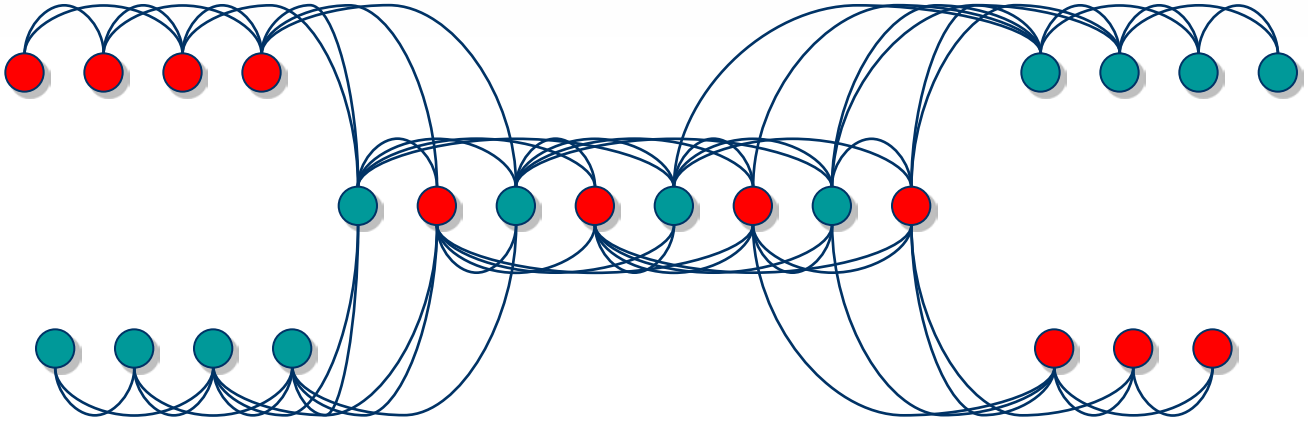
We want to merge reads up to potential repeat boundaries

2. Merge Reads into Contigs

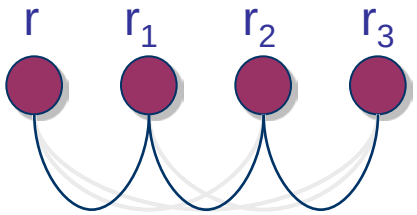


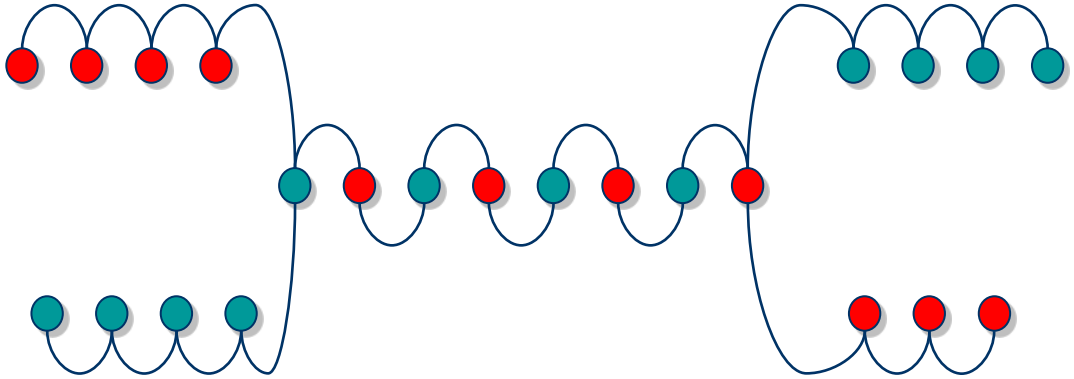
- Ignore non-maximal reads
- Merge only maximal reads into contigs

2. Merge Reads into Contigs

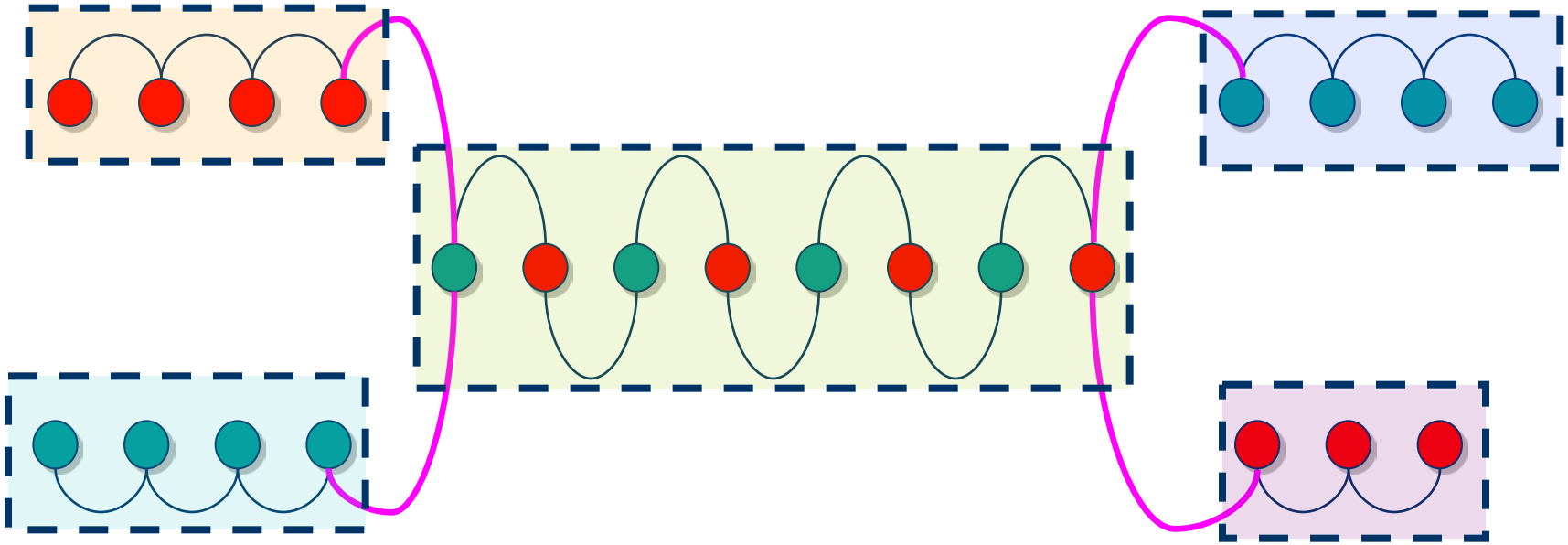


- Remove transitively inferable overlaps
  - If read r overlaps to the right reads r_1, r_2, and r_1 overlaps r_2, then (r, r_2) can be inferred by (r, r_1) and (r_1, r_2)

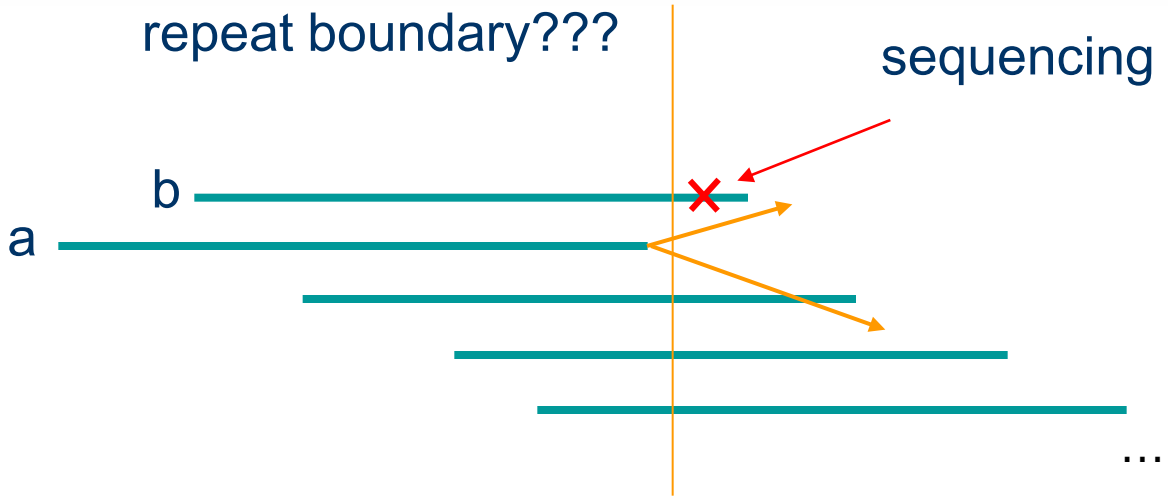


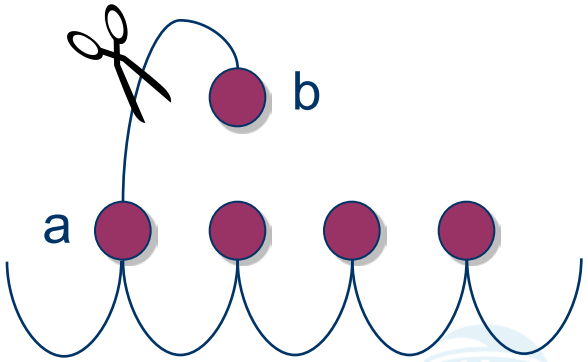


2. Merge Reads into Contigs



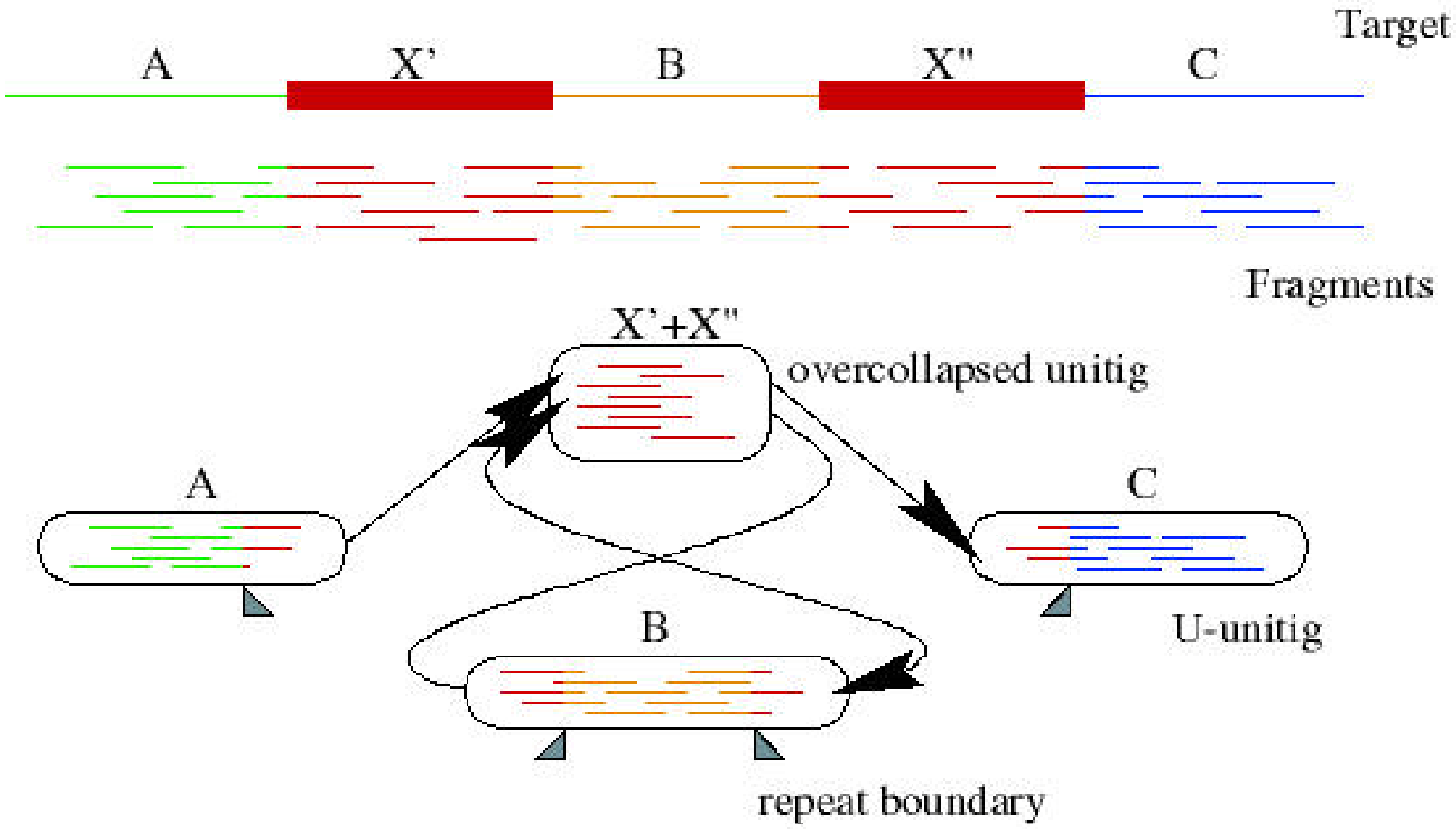
2. Merge Reads into Contigs





- Ignore "hanging" reads, when detecting repeat boundaries

Overlap graph after forming contigs



Repeats, errors, and contig lengths

- Repeats shorter than read length are easily resolved
  - Read that spans across a repeat disambiguates order of flanking regions
- Repeats with more base pair diffs than sequencing error rate are OK
  - We throw overlaps between two reads in different copies of the repeat
- To make the genome **appear** less repetitive, try to:
  - Increase read length
  - Decrease sequencing error rate

Role of error correction:

Discards up to 98% of single-letter sequencing errors

decreases error rate

$\Rightarrow$ decreases effective repeat content

$\Rightarrow$ increases contig length

Consensus

- A consensus sequence is derived from a profile of the assembled fragments
- A sufficient number of reads is required to ensure a statistically significant consensus
- Reading errors are corrected

Derive Consensus Sequence

Derive multiple alignment from pairwise read alignments

Derive each consensus base by weighted voting

Some Assemblers

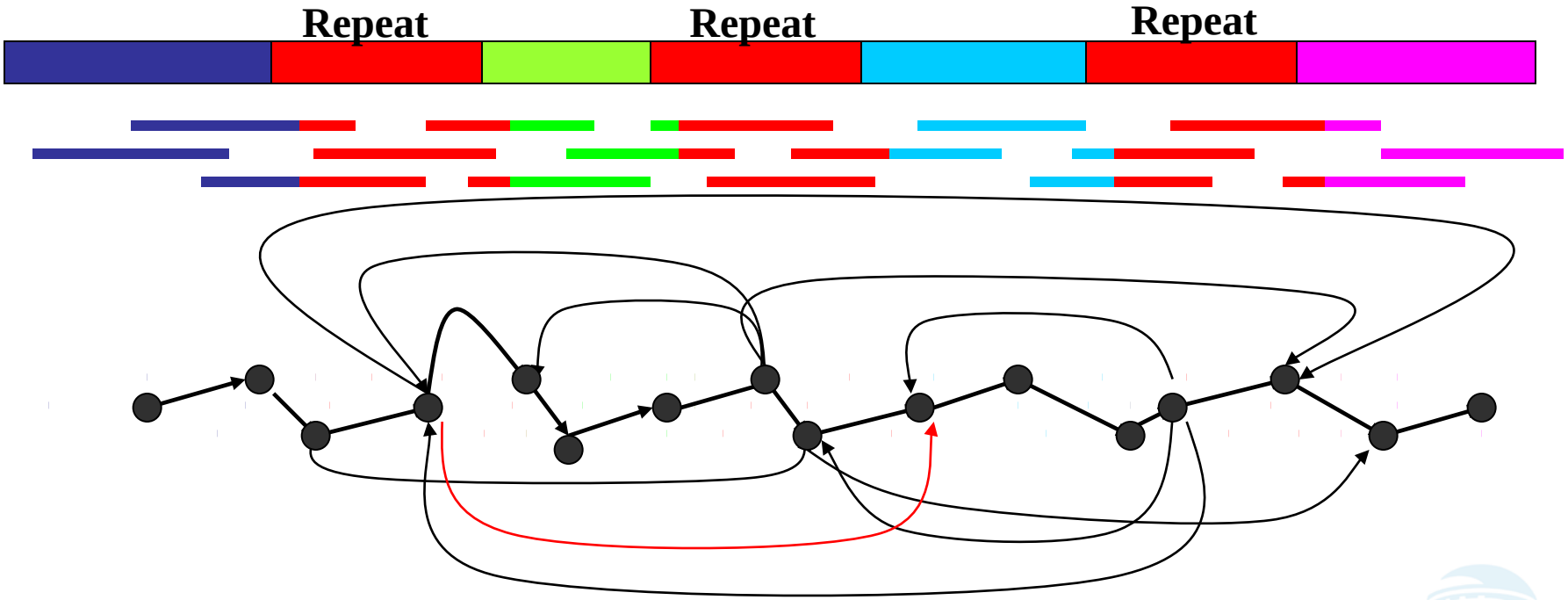
- PHRAP
  - Early assembler, widely used, good model of read errors
  - Overlap $O(n^2)$ → layout (no mate pairs) → consensus
- Celera
  - First assembler to handle large genomes (fly, human, mouse)
  - Overlap → layout → consensus
- Arachne
  - Public assembler (mouse, several fungi)
  - Overlap → layout → consensus
- Phusion
  - Overlap → clustering → PHRAP → assemblage → consensus
- Euler
  - Indexing → Euler graph → layout by picking paths → consensus

EULER Fragment Assembly

- Traditional "overlap-layout-consensus" technique has a high rate of mis-assembly
- EULER uses the Eulerian Path approach borrowed from the SBH problem
- Fragment assembly without repeat masking can be done in linear time with greater accuracy

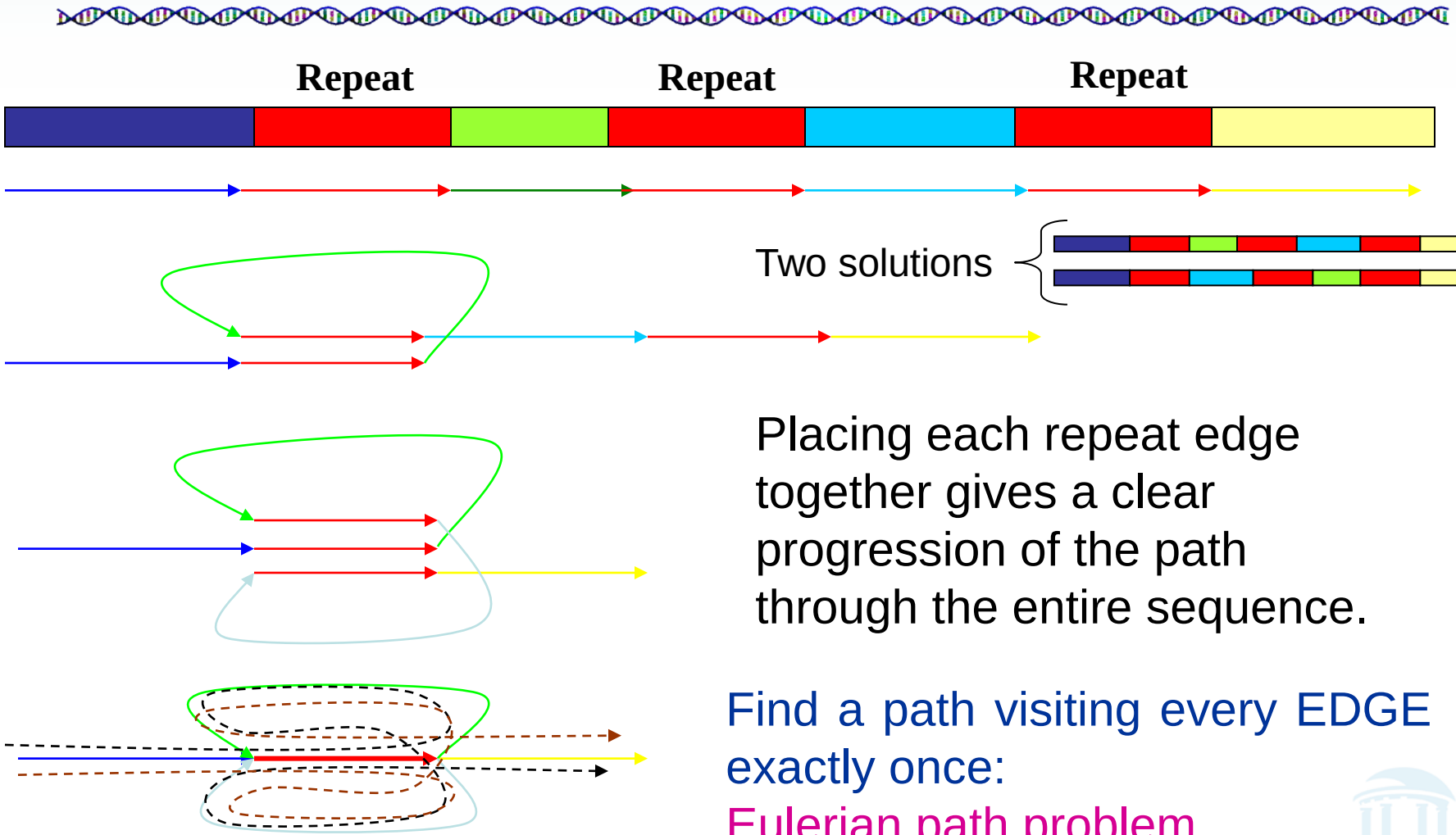
Overlap Graph: Hamiltonian Approach

Each vertex represents a read from the original sequence.
Vertices from repeats are connected to many others.



Find a path visiting every VERTEX exactly once: Hamiltonian path problem

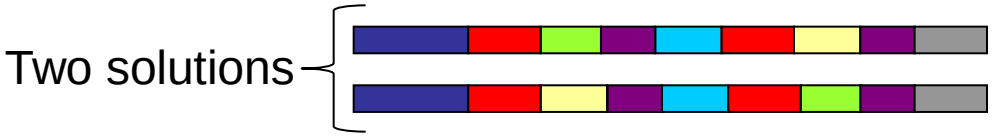
Overlap Graph: Eulerian Approach

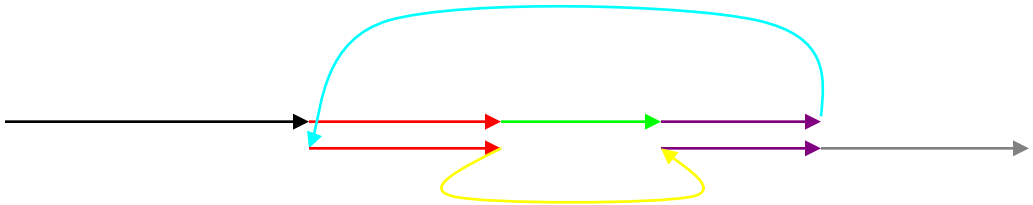


Placing each repeat edge together gives a clear progression of the path through the entire sequence.

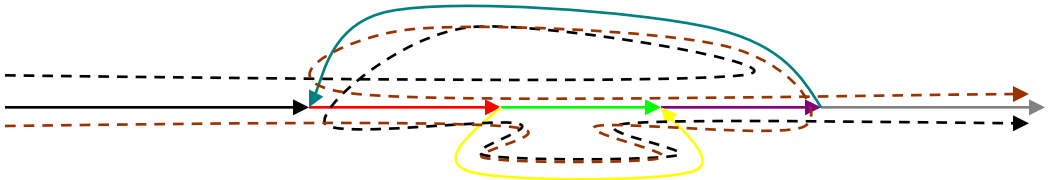
Find a path visiting every EDGE exactly once:
Eulerian path problem

Multiple Repeats





Can be easily constructed with any number of repeats



Construction of Repeat Graph

- Construction of repeat graph from *k* – mers: emulates an SBH experiment with a huge (virtual) DNA chip.

- Breaking reads into *k* – mers: Transform sequencing data into virtual DNA chip data.

Construction of Repeat Graph (cont'd)

- Error correction in reads: "consensus first" approach to fragment assembly. Makes reads (almost) error-free BEFORE the assembly even starts.

- Using reads and mate-pairs to simplify the repeat graph (Eulerian Superpath Problem).

Hybrid Sequencing

- Use short read sequencing to create accurate overlap graphs

- Align noisy long reads to overlap graphs to link contigs
  - How to align a noisy read to a graph?

Conclusions

- Graph theory is a vital tool for solving biological problems

- Wide range of applications, including sequencing, motif finding, protein networks, and many more

References

- Simons, Robert W. *Advanced Molecular Genetics Course*, UCLA (2002). http://www.mimg.ucla.edu/bobs/C159/Presentations/Benzer.pdf

- Batzoglou, S. *Computational Genomics Course*, Stanford University (2006). http://ai.stanford.edu/~serafim/CS262_2006/