CMSC 829C: Algorithms, AI, and Hardware for Biological Data

Lecture 5: Genome Assembly

Fall 2026

September 15, 2026

Can Firtina | canfirtina@gmail.com | cs.umd.edu/~firtina

Assistant Professor of Computer Science at UMD

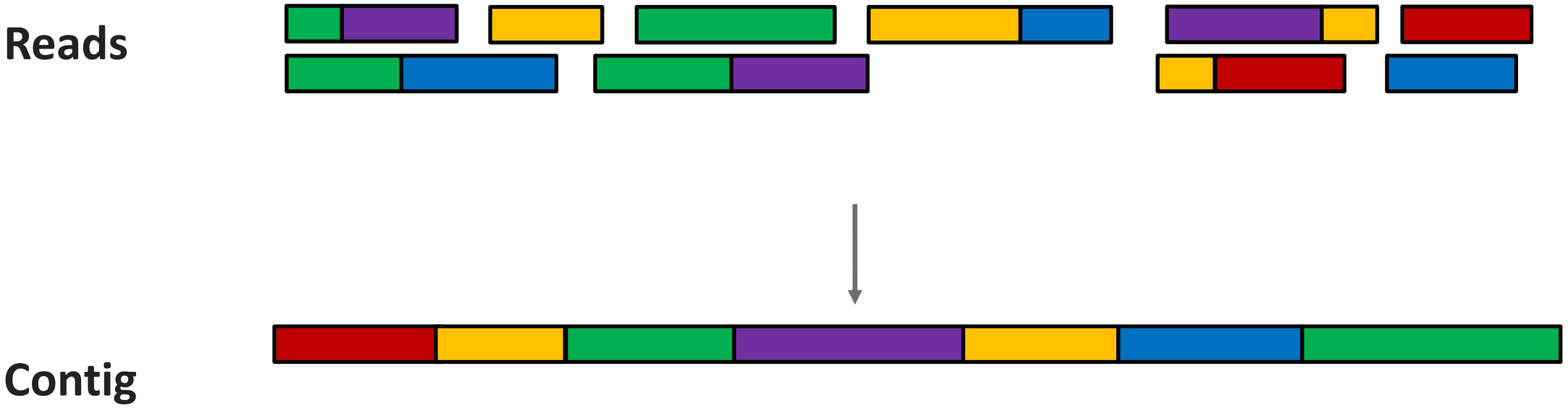
UNIVERSITY OF MARYLAND

Agenda

- From overlapping reads to contigs
  - Overlap graphs, cleaning, consensus and scaffolding
- Assembly with de Bruijn graphs
- Modern assembly and how to evaluate it

Genome Assembly

- Reconstruct genomic sequence from overlapping reads
- A contig is one continuous assembled sequence



Read Length, Accuracy and Coverage

Small puzzle pieces

Short reads

Large puzzle pieces

Long reads

Either way, every position must be read, ideally several times

Accurate local sequence

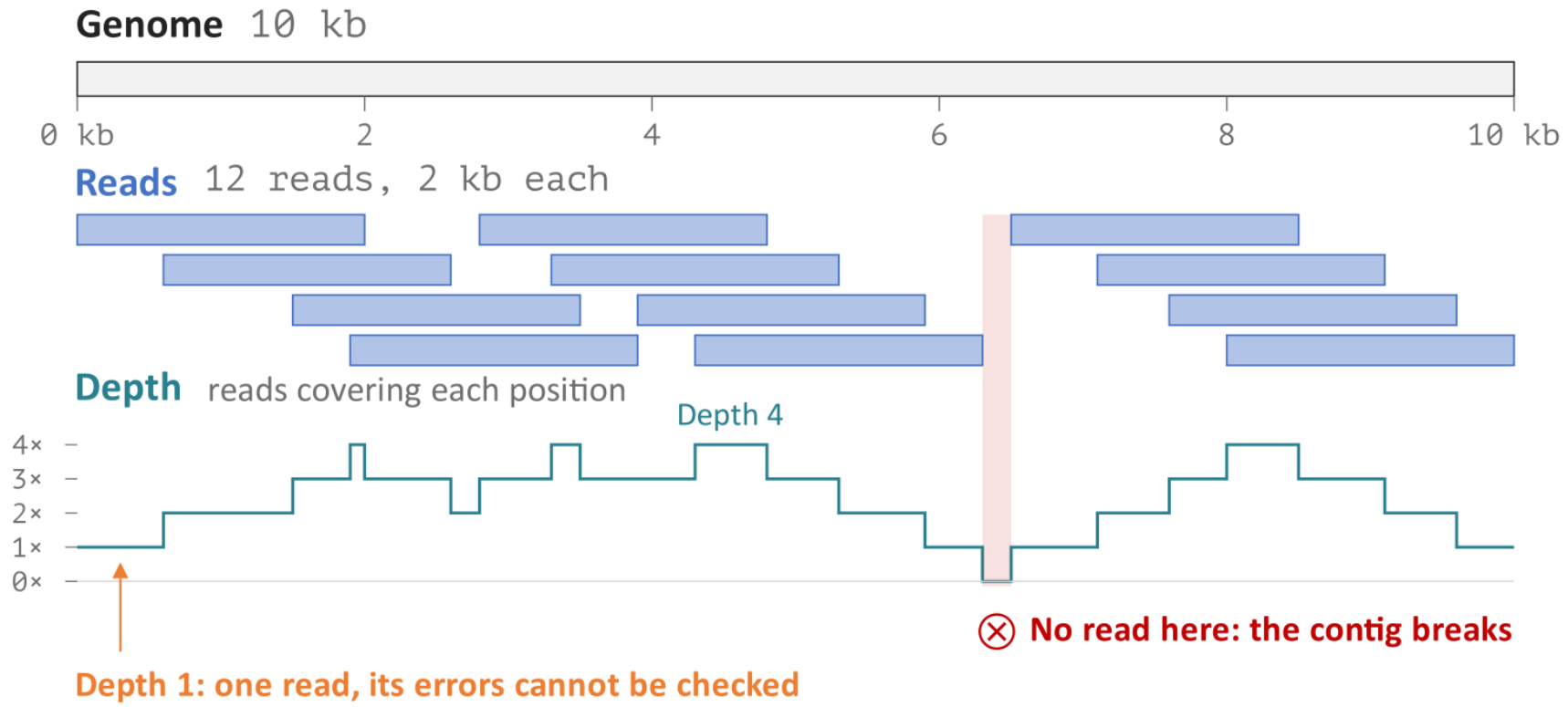
Repeats can exceed read length

Longer reads can span repeats

Accuracy depends on the technology

Coverage: How Many Reads Cover a Position

- Average coverage is total read bases divided by genome length
- Local depth varies: gaps break contigs, single reads go unchecked



Average coverage

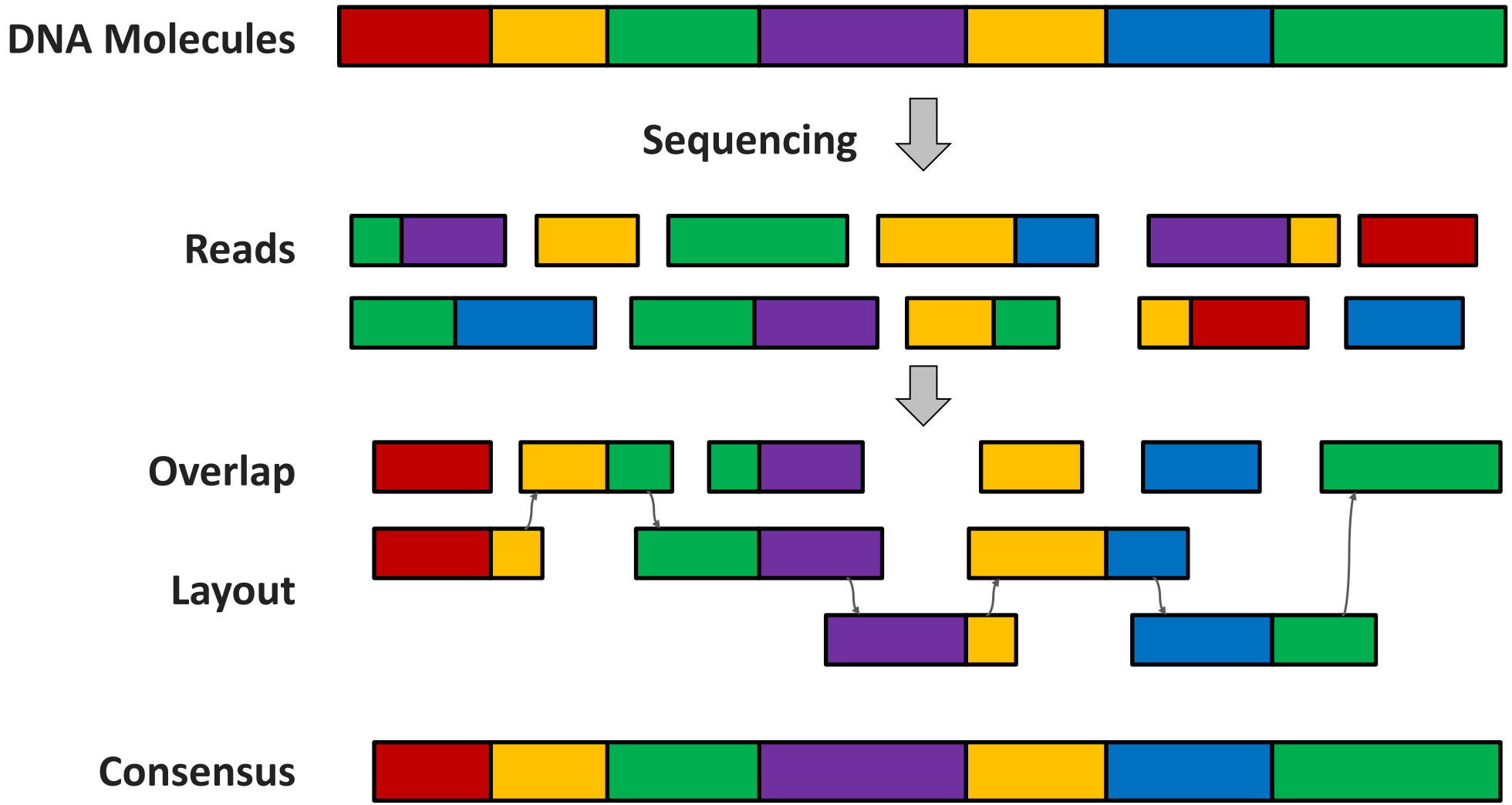
$$\frac{12 \times 2\ \text{kb}}{10\ \text{kb}} = 2.4\times$$

Average, not a promise

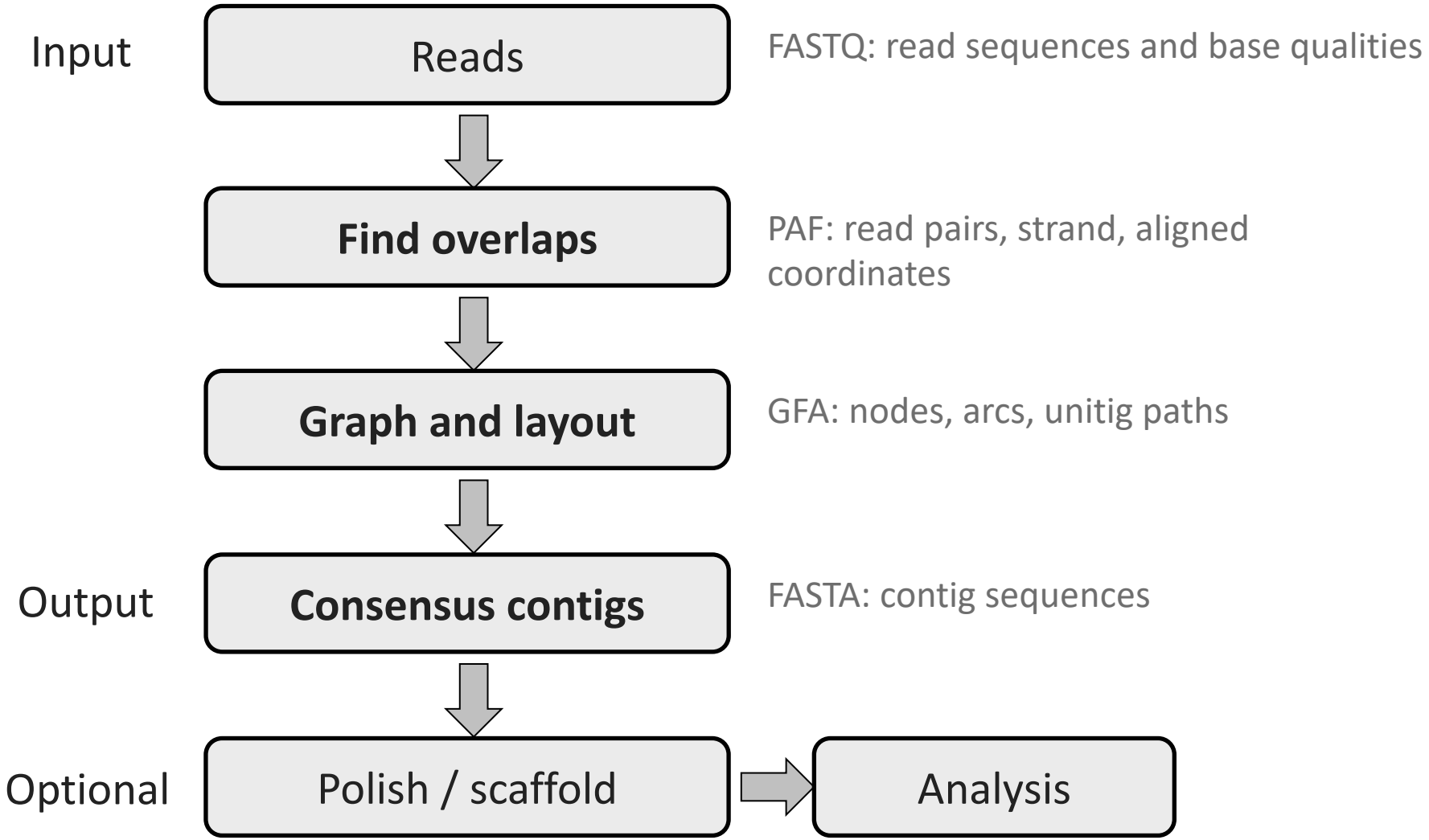
Local depth here: 0× to 4×

More depth: more chances to bridge gaps and outvote errors

From Reads to an Assembly



Overlap, Layout and Consensus



Two Ways to Build an Assembly Graph

- **Overlap graphs: connect whole reads through overlaps**
- De Bruijn graphs: connect k-mers (DNA words of length k)

Overlap graph

Nodes: whole reads

Edges: overlaps found by comparing read pairs

Seeds pick candidates, alignment confirms

Cost grows with the number of read pairs

Repeats stay inside reads, so long reads separate copies

De Bruijn graph, k = 4

Nodes: distinct k-mers from a hash table

Edges: consecutive windows, k – 1 shared bases

One pass over the reads, no pairwise comparison

Every repeated k-mer collapses into one node

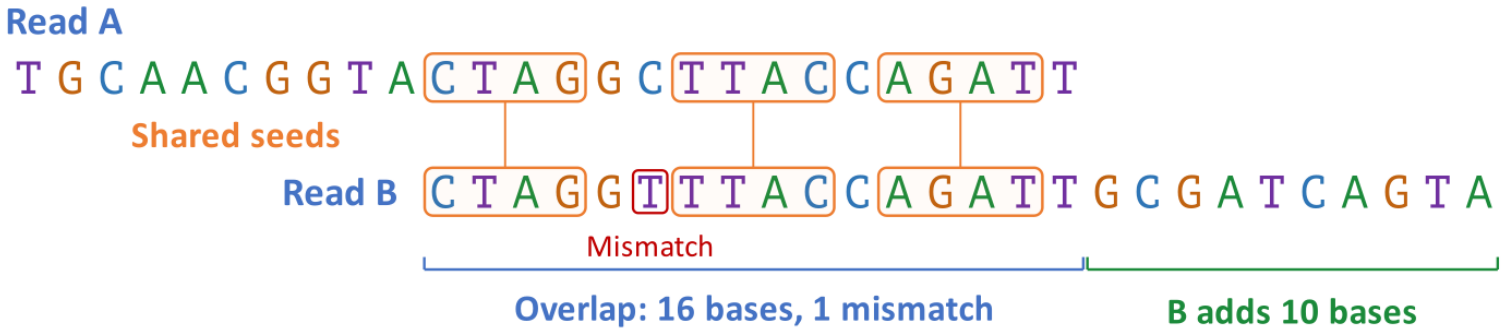
Read identity is lost unless reads are threaded back

Same three reads, same contig

A C G T T A G C C A

Finding Overlapping Reads

- Find candidate read pairs using shared seeds
- Chain matching seeds and estimate the overlap
- Account for the reverse-complement orientation



Seed index seed to read and position

C T A G	A 11	B 1
T T A C	A 17	B 7
A G A T	A 22	B 12

Same offset for all three: 11 − 1 = 17 − 7 = 22 − 12 = 10

One chain, so A and B are a candidate pair worth aligning

Other strand

Read C

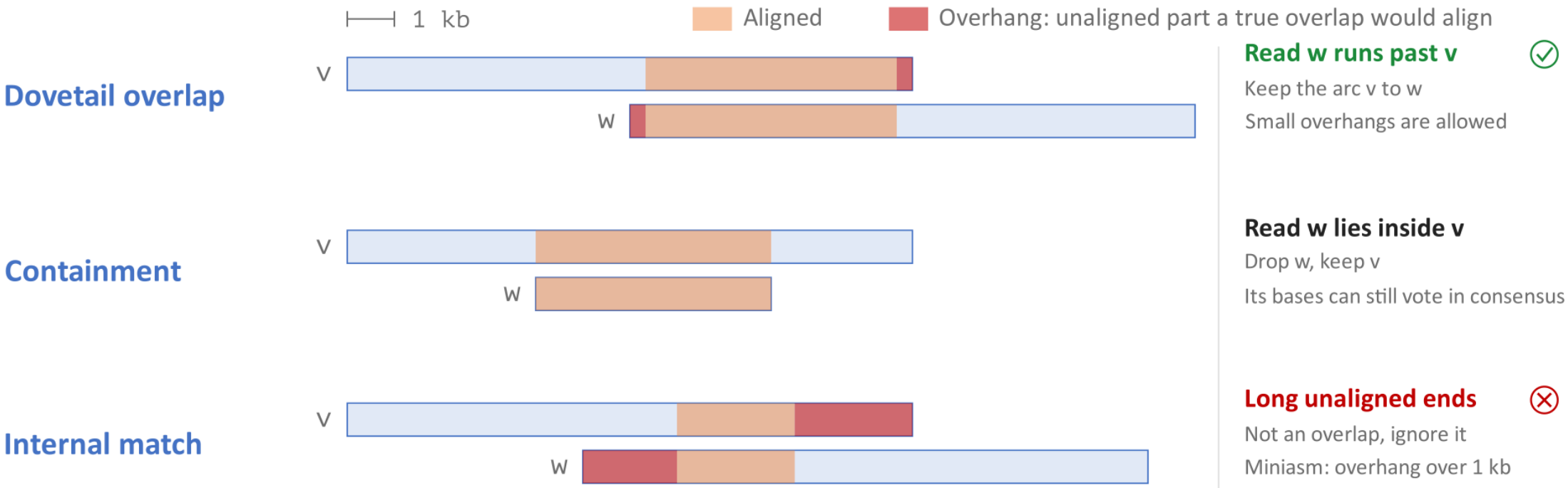
Reverse complement it, then query the same index

Candidate pair A, B
3 seeds, offset 10

Dovetail overlap
16 bases, then 10 new

Which Matches Extend a Read?

- Compare the aligned region with what lies beyond it on each read
- Only *informative* (dovetail) overlaps become arcs of the graph



Dovetail is the string-graph term of Myers, Celera and Canu, the miniasm paper just says overlap

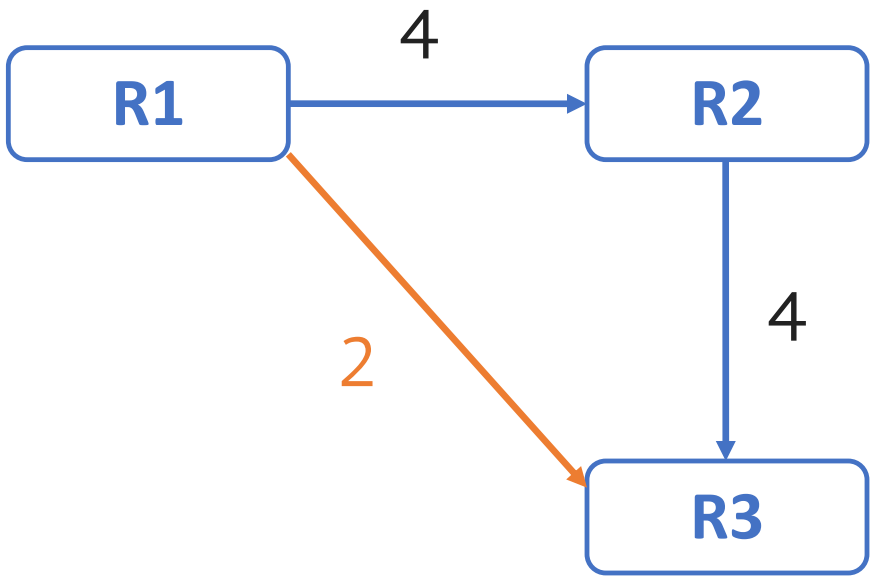
Reads as an Overlap Graph

- Nodes represent oriented reads
- Edges represent informative overlaps

R1 A C G T T A

R2 G T T A G C

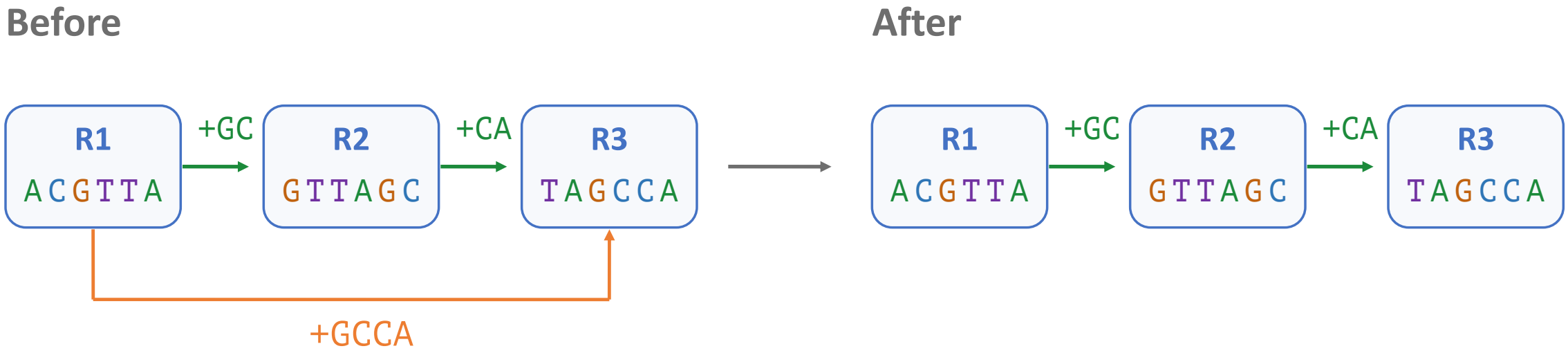
R3 T A G C C A



Minimum overlap: 2 bases

Removing a Transitive Overlap

- Both routes add the same sequence after R1



Direct GCCA

Via R2 GC + CA = GCCA

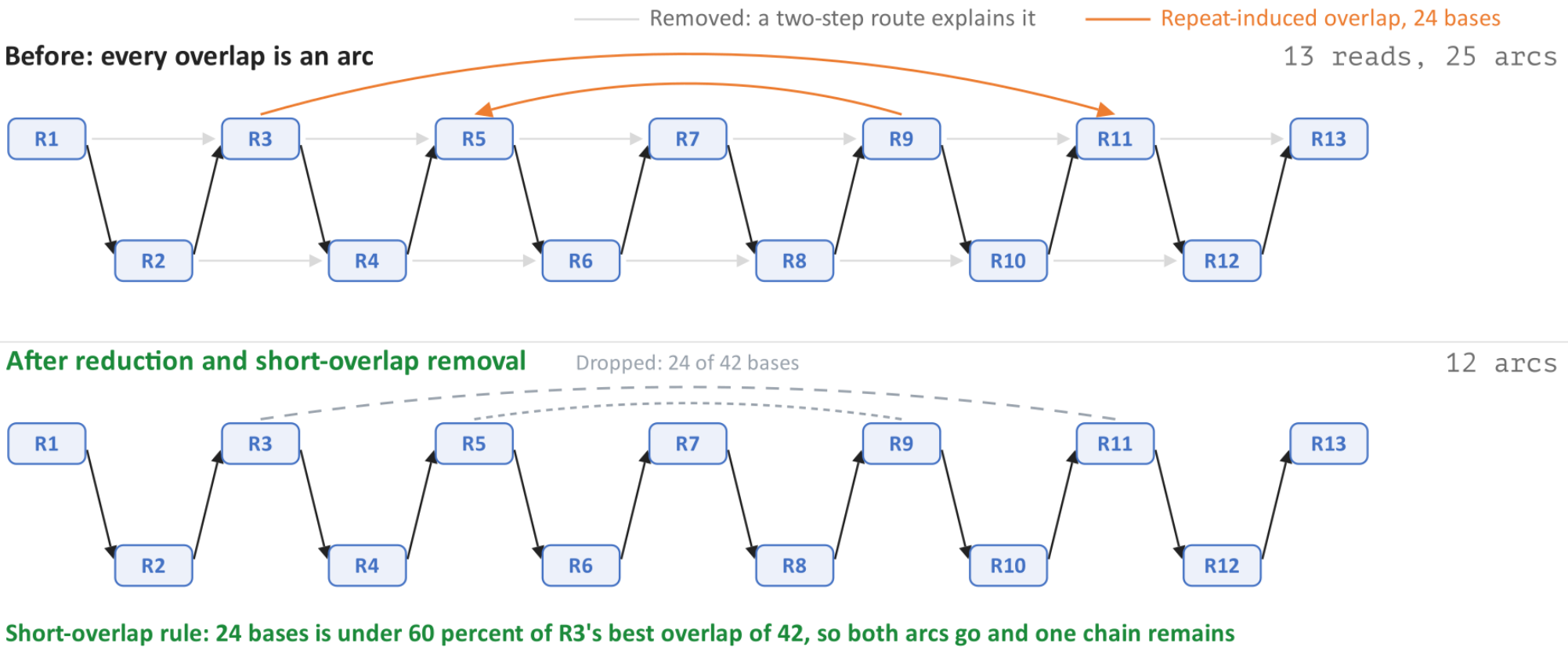
Contig

A C G T T A G C C A

The reduced graph is Myers' string graph: each arc spells only the bases it adds

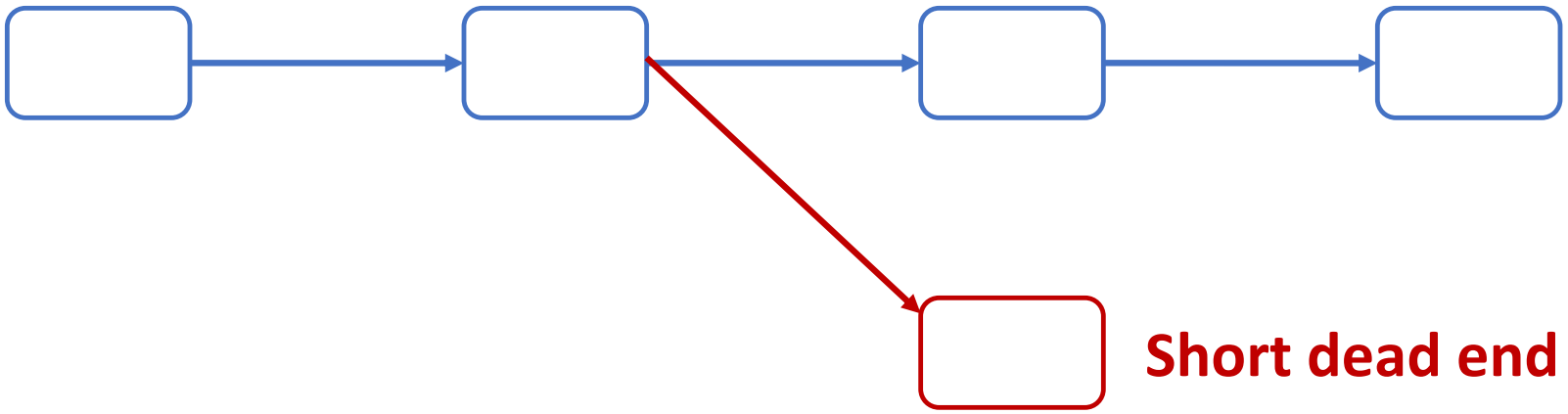
Graph Simplification

- Remove redundant edges while preserving sequence relationships
- Repeat-induced overlaps survive: no two-step route explains them
- Miniasm then drops arcs much shorter than a read's best overlap



Short Dead Ends in the Graph

- A tip is a short branch that stops early
- Trimming suspicious tips is a heuristic



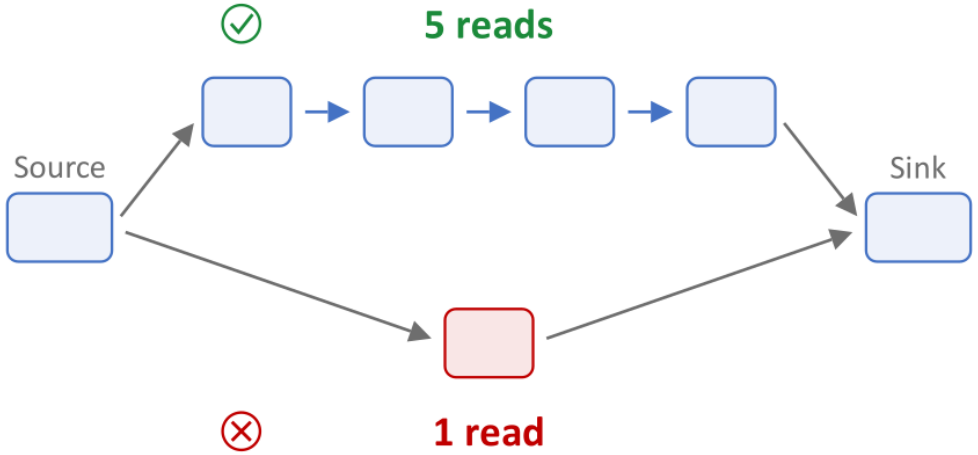
Miniasm cuts a dead end made of fewer than 4 reads (-e 4)

A short end can also be real sequence

Bubbles: Errors or Real Variants

- Alternative paths can share a source and a sink
- Sequencing errors and real alleles look alike, yet need different treatment

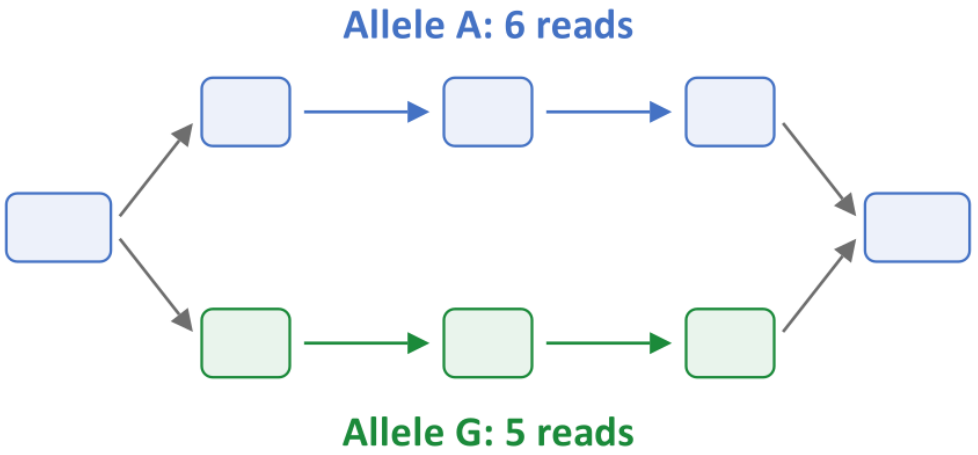
Likely a sequencing error



Keep the path with more reads

Miniasm: read count, then path length

Two alleles of one locus

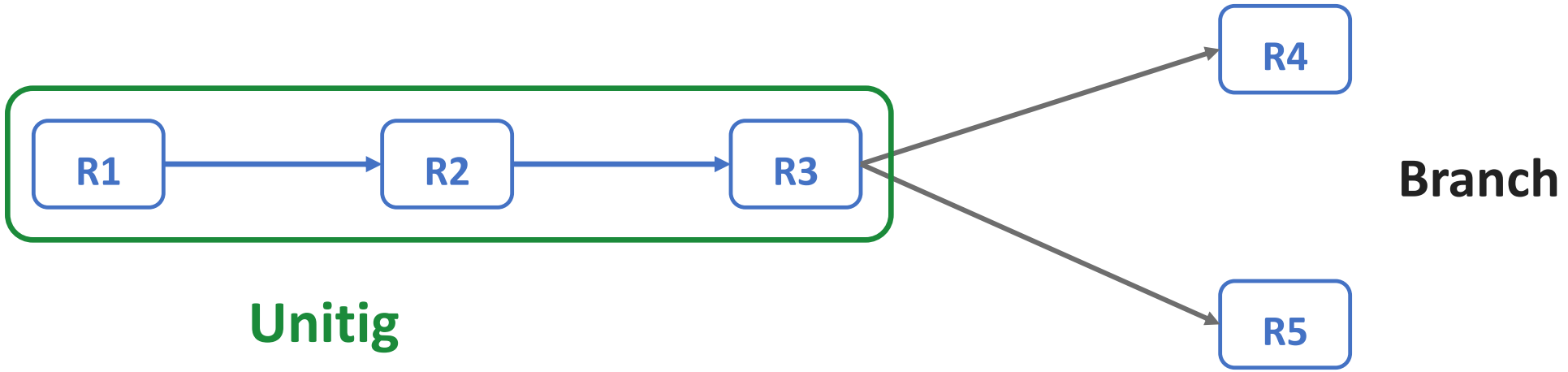


Popping keeps one path and loses a haplotype

Haplotype-aware assemblers keep both

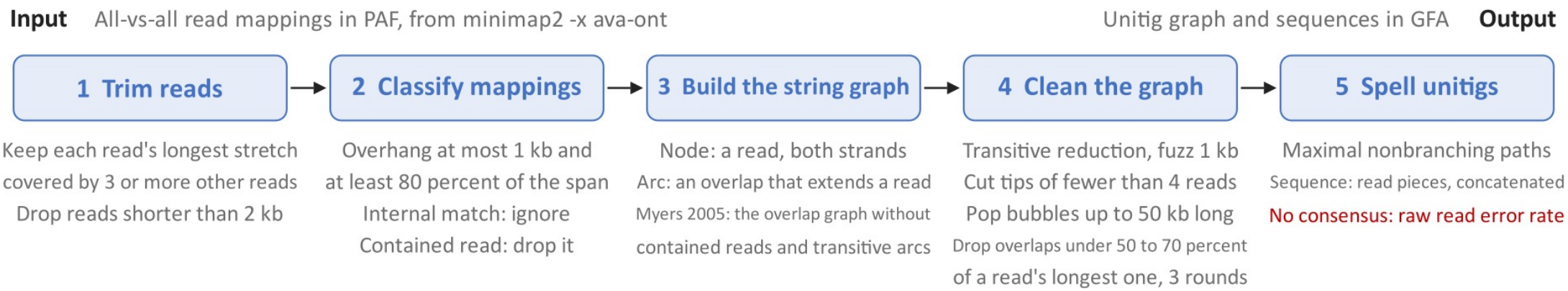
From Nonbranching Paths to Unitigs

- Continue while each internal step has one way in and one way out
- Compact the maximal unambiguous path



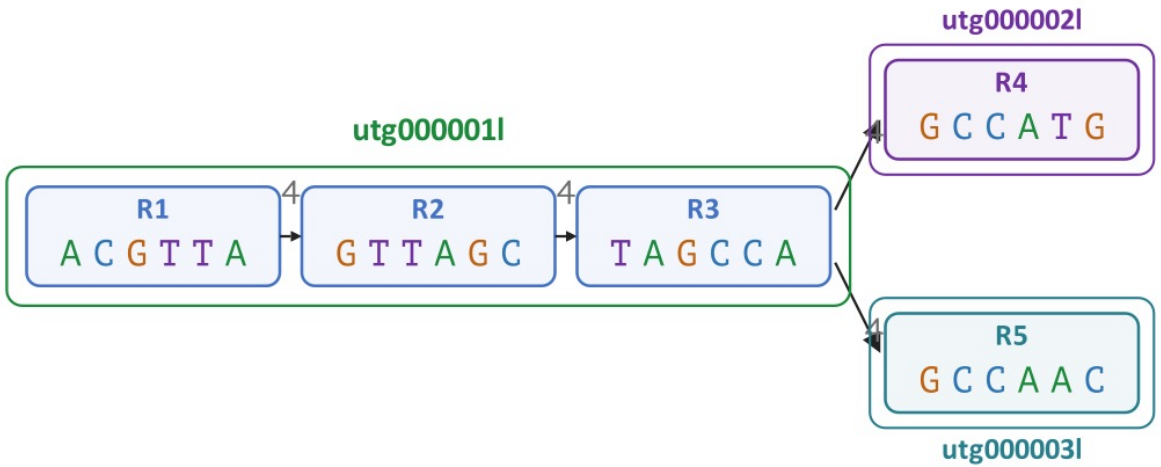
Miniasm: From Mappings to Unitigs

- Every step is a filter or a graph rule with a threshold
- No base is corrected: a consensus tool must follow



Unitigs and Links in a GFA File

- One S line per unitig, a lines give the read pieces behind it
- L lines keep the branch: three unitigs, two links, one component



Spelling stops at R3, the branching read

R4 and R5 each start a new unitig, the 4 shared bases appear at both ends

utg000001l A C G T T A G C C A

utg000002l G C C A T G

utg000003l G C C A A C

One connected component: three S lines joined by two L lines

Miniasm GFA output

```
S utg000001l  ACGTTAGCCA  LN:i:10
a utg000001l  0  R1:1-6  +  2
a utg000001l  2  R2:1-6  +  2
a utg000001l  4  R3:1-6  +  6
S utg000002l  GCCATG      LN:i:6
a utg000002l  0  R4:1-6  +  6
S utg000003l  GCCAAC      LN:i:6
a utg000003l  0  R5:1-6  +  6
L utg000001l  +  utg000002l  +  4M  SD:i:6
L utg000001l  +  utg000003l  +  4M  SD:i:6
```

S unitig name and sequence

a read piece: offset, read and range, strand, bases it supplies

L link: unitig ends, 4M overlap, SD extension

Further Reading: Assembly Graphs

- Read the following paper if you are curious about
  - How the transitive reduction works:

BIOINFORMATICS

Vol. 21 Suppl. 2 2005, pages ii79–ii85
doi:10.1093/bioinformatics/bti1114

Genes and Genomes

The fragment assembly string graph

Eugene W. Myers
Department of Computer Science, University of California, Berkeley, CA, USA

  - How to collapse bubbles in overlap graphs:

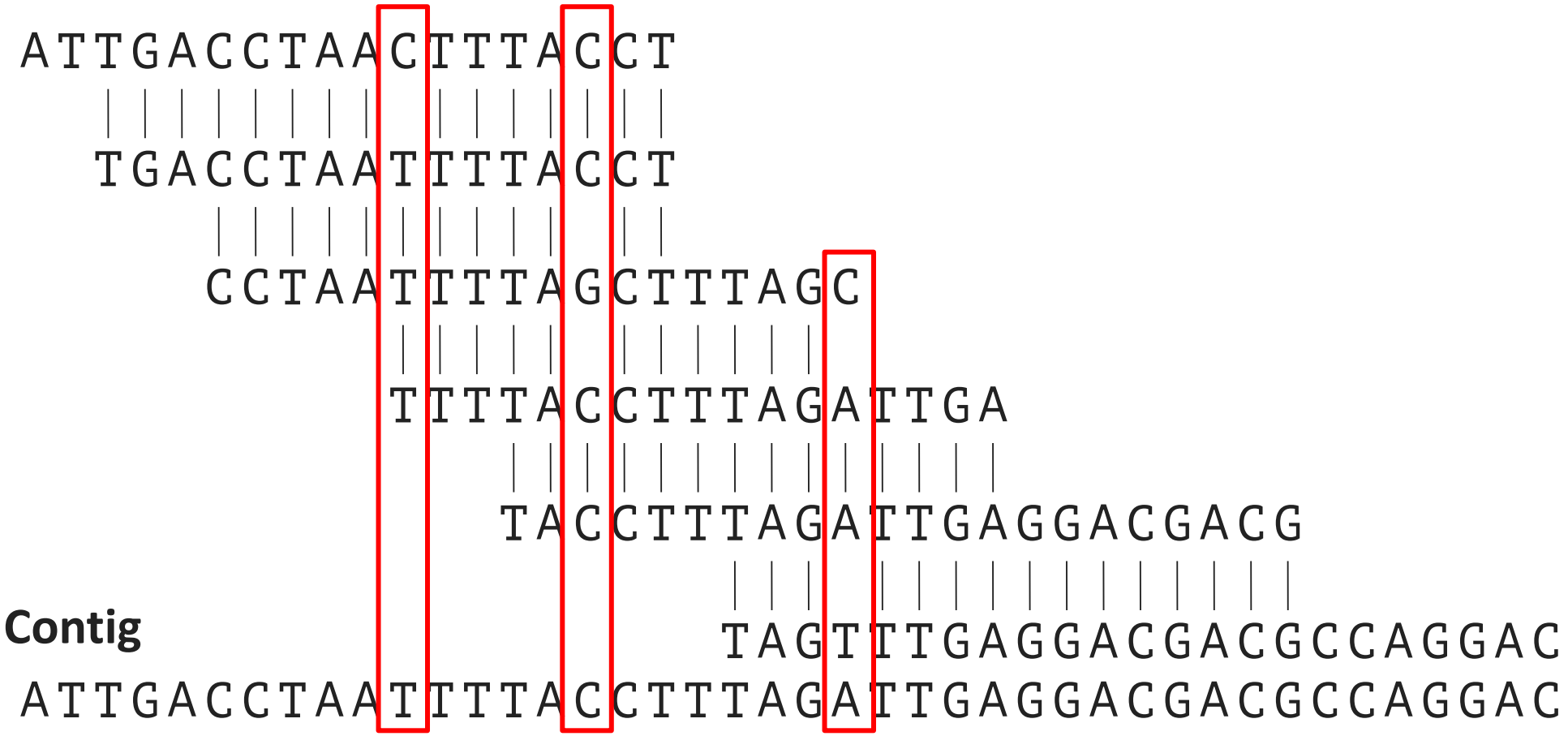
Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences FREE

Heng Li Author Notes

Bioinformatics, Volume 32, Issue 14, 15 July 2016, Pages 2103–2110,
https://doi.org/10.1093/bioinformatics/btw152
Published: 19 March 2016 **Article history** ▾

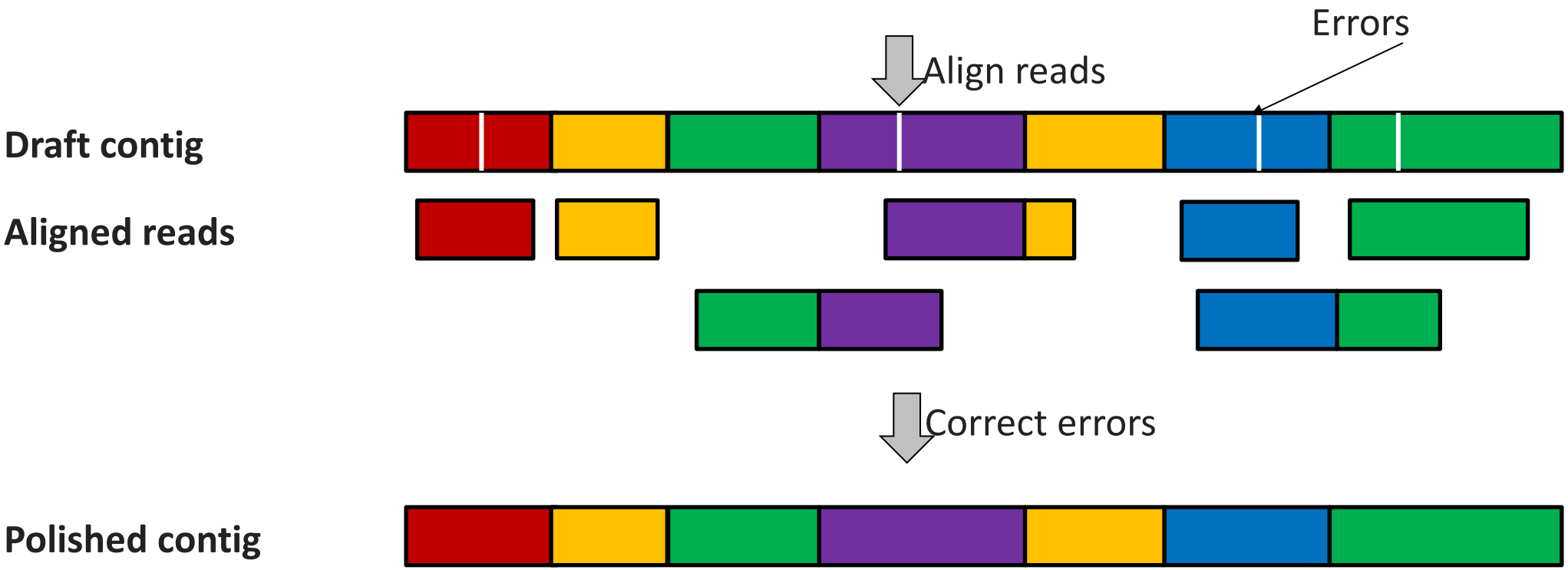
Consensus from Aligned Reads

- Align the reads consistently with the layout
- Combine their evidence to infer the assembled bases



Assembly Polishing

- Align reads to the draft and revise unsupported bases
  - By combining evidence from different read technologies



Error Correction using ML [Bioinformatics '20]

- **Apollo: a sequencing-technology-independent, scalable and accurate assembly polishing algorithm**
  Can Firtina, Jeremie S. Kim, Mohammed Alser, Damla Senol Cali, A Ercument Cicek, Can Alkan, Onur Mutlu
  Bioinformatics, June 2020.
  [PDF] [Code] [Link]

Apollo: a sequencing-technology-independent, scalable and accurate assembly polishing algorithm

FREE

Can Firtina, Jeremie S Kim, Mohammed Alser, Damla Senol Cali, A Ercument Cicek,
Can Alkan ✉, Onur Mutlu ✉

Bioinformatics, Volume 36, Issue 12, 15 June 2020, Pages 3669–3679,
https://doi.org/10.1093/bioinformatics/btaa179
Published: 13 March 2020 **Article history** ▾

Accelerating ML & Genome Graphs [ACM TACO '24]

- **ApHMM: Accelerating Profile Hidden Markov Models for Fast and Energy-efficient Genome Analysis**
  Can Firtina, Kamlesh Pillai, Gurpreet S. Kalsi, Bharathwaj Suresh, Damla Senol Cali, Jeremie S. Kim, Taha Shahroodi, Meryem Banu Cavlak, Joël Lindegger, Mohammed Alser, Juan Gómez Luna, Sreenivas Subramoney, Onur Mutlu
  ACM Transactions on Architecture and Code Optimization (**TACO**), February 2024.
  [PDF] [Code] [DOI]
  HiPEAC 2024 Talk [Video] [Slides (pptx)] [Slides (pdf)]

RESEARCH-ARTICLE | OPEN ACCESS |

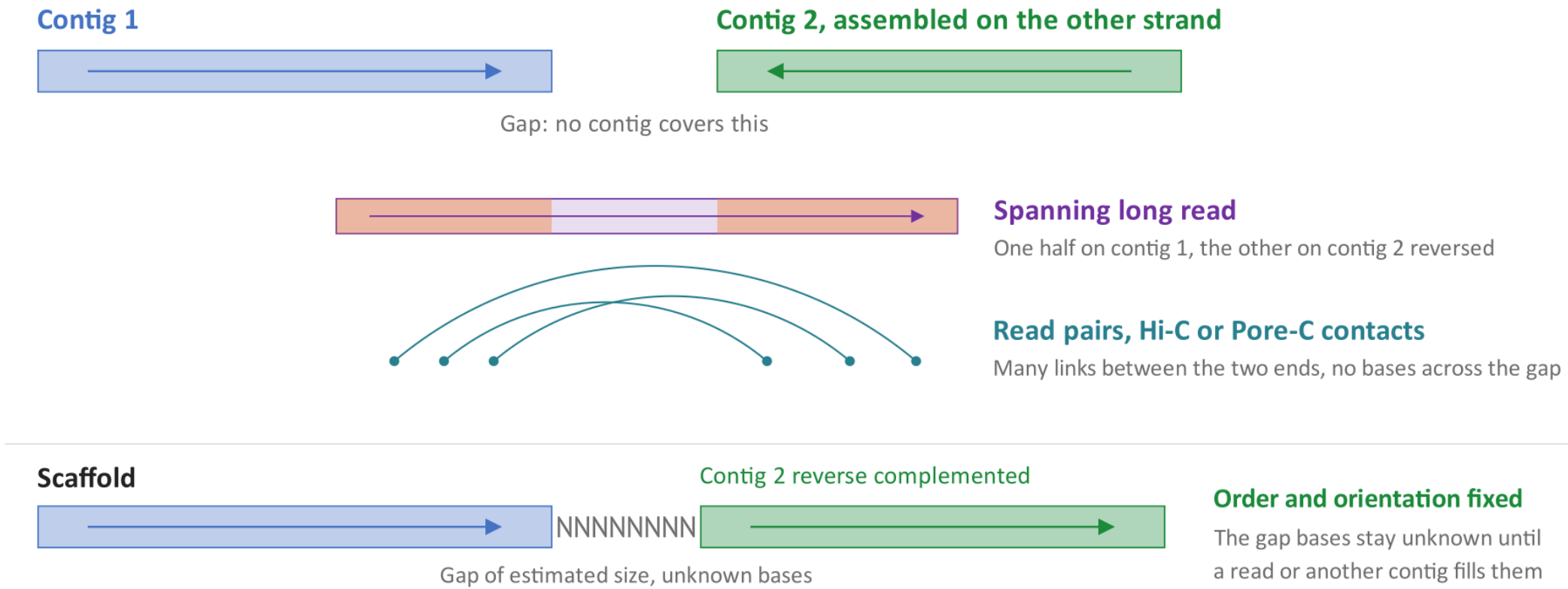
ApHMM: Accelerating Profile Hidden Markov Models for Fast and Energy-efficient Genome Analysis

Authors: Can Firtina, Kamlesh Pillai, Gurpreet S. Kalsi, Bharathwaj Suresh, Damla Senol Cali, Jeremie S. Kim, Taha Shahroodi, Meryem Banu Cavlak, Joël Lindegger, Mohammed Alser, Juan Gómez Luna, Sreenivas Subramoney, Onur Mutlu (Less) | Authors Info & Claims

ACM Transactions on Architecture and Code Optimization, Volume 21, Issue 1 • Article No.: 19, Pages 1 - 29
https://doi.org/10.1145/3632950

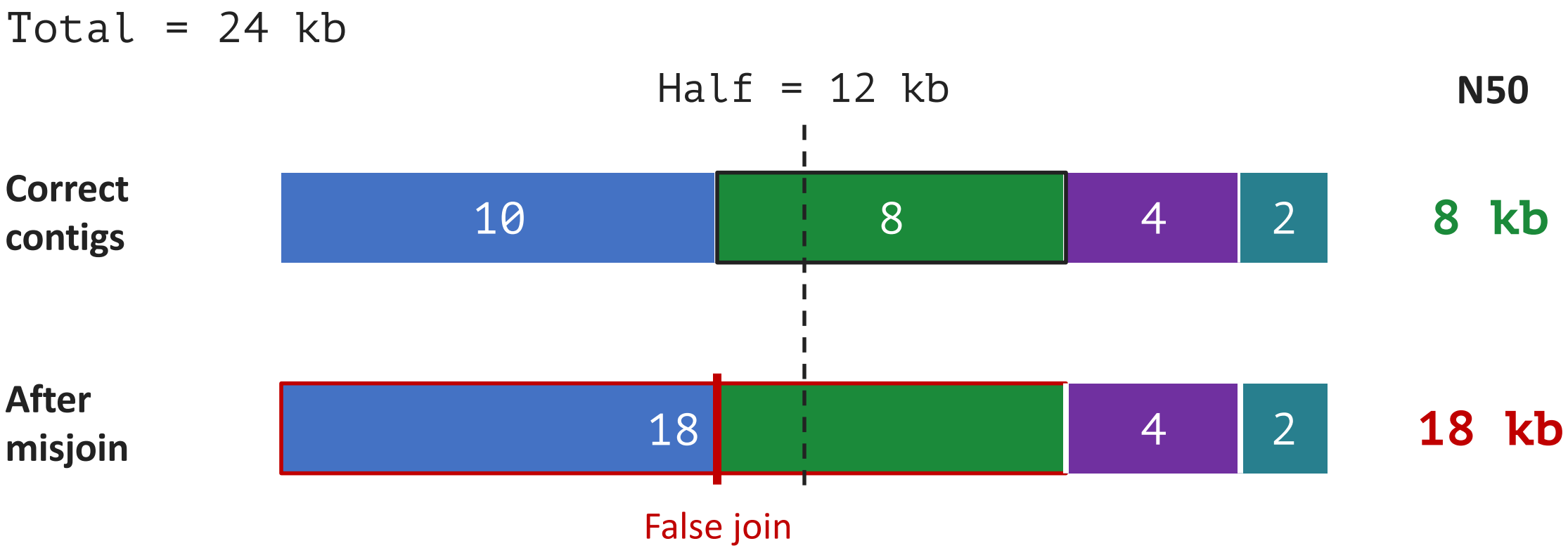
Scaffolds: Order and Orientation

- Use long-range evidence to order and orient contigs
- Unknown sequence between contigs remains a gap of estimated size



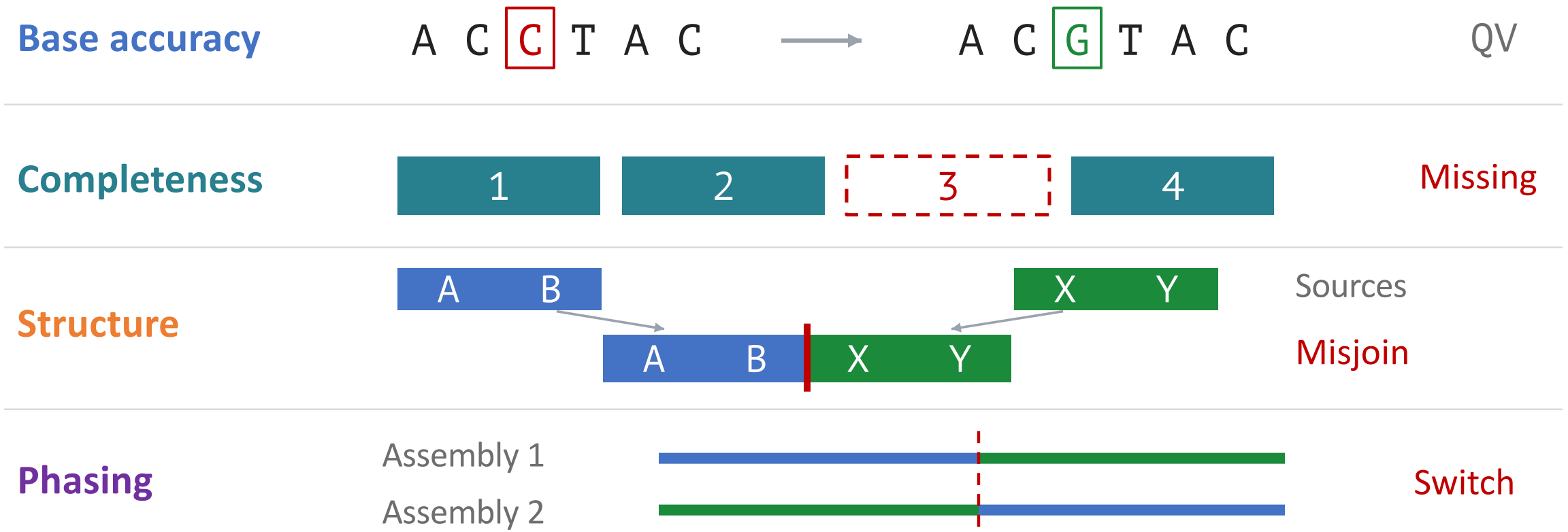
Contiguity: What N50 Measures

- Sort contigs by length and accumulate half the total assembly
- N50 alone does not establish a correct assembly



Assembly Quality Has Several Dimensions

- Check base accuracy, completeness, structure and phase
- Different checks reveal different kinds of error



New Directions: Overlapping Raw Signals

- **Rawsamble: Overlapping Raw Nanopore Signals using a Hash-based Seeding Mechanism**
  Can Firtina, Maximilian Mordig, Harun Mustafa, Sayan Goswami, Nika Mansouri Ghiasi, Stefano Mercogliano, Furkan Eris, Joel Lindegger, Andre Kahles, and Onur Mutlu
  Bioinformatics, February 2026
  [online link] [pdf] [code]
  ISMB 2024 Talk [video] [pptx] [pdf]
  CSHL 2024 (Biological Data Science) Talk [video] [pptx] [pdf]
  Social Media Thread [Twitter (X)] [LinkedIn]

Bioinformatics, 2026, **42**(3), btag087
https://doi.org/10.1093/bioinformatics/btag087
Published: 26 February 2026
Original Paper | Sequence Analysis

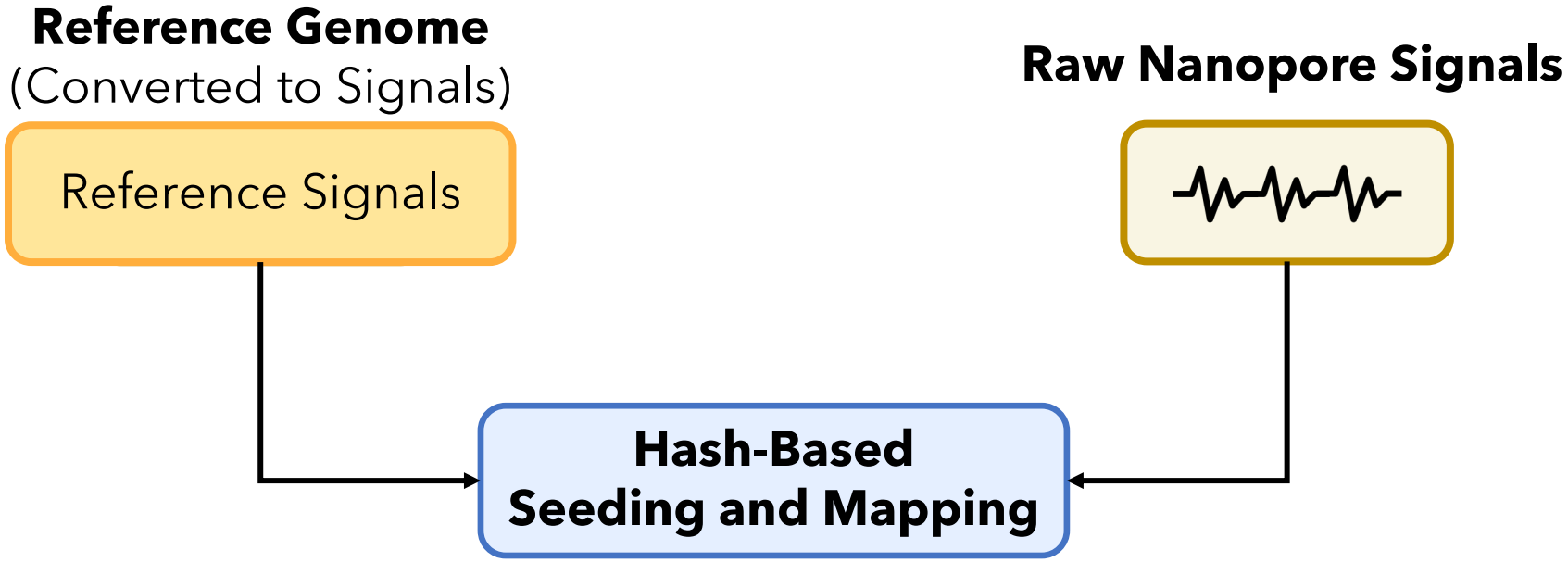
OXFORD

Rawsamble: overlapping raw nanopore signals using a hash-based seeding mechanism

Can Firtina[1,2,*], Maximilian Mordig[3,4], Harun Mustafa[3,5,6], Sayan Goswami[3], Nika Mansouri Ghiasi[1], Stefano Mercogliano[1], Furkan Eris[1], Joel Lindegger[1], André Kahles[3,5,6,*], Onur Mutlu[1,*]

[1]Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich, 8092, Switzerland
[2]Department of Computer Science, University of Maryland, MD 20742, United States
[3]Department of Computer Science, ETH Zurich, Zurich, 8092, Switzerland
[4]Max Planck Institute for Intelligent Systems, Tubingen, 72076, Germany
[5]University Hospital Zurich, Zurich, 8091, Switzerland
[6]Swiss Institute of Bioinformatics, Zurich, 8057, Switzerland

Beyond Reference Mapping: Overlapping

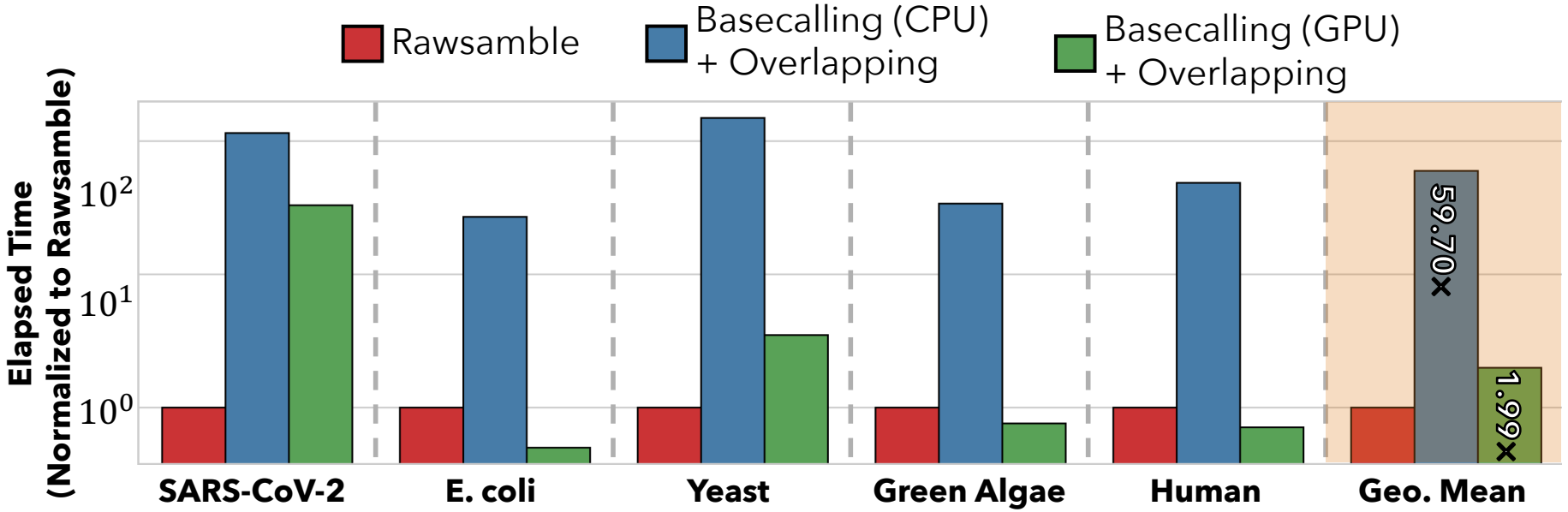


Challenge: Reference genome is not always available

Assembly: Constructing genome from **overlapping reads**

Existing solutions cannot find overlapping reads **without basecalling**

Eliminating Basecalling from the Pipeline



Average speedup of **60**× than CPU-based basecalling:
Eliminating basecalling leaves a significant room for performance improvements

Average speedup of **2**× than the **GPU-based basecalling**:
GPU vs. GPU comparison is likely to provide even better results

New Directions: Overlapping Raw Signals

- **Rawsamble: Overlapping Raw Nanopore Signals using a Hash-based Seeding Mechanism**
  Can Firtina, Maximilian Mordig, Harun Mustafa, Sayan Goswami, Nika Mansouri Ghiasi, Stefano Mercogliano, Furkan Eris, Joel Lindegger, Andre Kahles, and Onur Mutlu
  Bioinformatics, February 2026
  [online link] [pdf] [code]
  ISMB 2024 Talk [video] [pptx] [pdf]
  CSHL 2024 (Biological Data Science) Talk [video] [pptx] [pdf]
  Social Media Thread [Twitter (X)] [LinkedIn]

Bioinformatics, 2026, **42**(3), btag087
https://doi.org/10.1093/bioinformatics/btag087
Published: 26 February 2026
Original Paper | Sequence Analysis

OXFORD

Rawsamble: overlapping raw nanopore signals using a hash-based seeding mechanism

Can Firtina[1,2,*], Maximilian Mordig[3,4], Harun Mustafa[3,5,6], Sayan Goswami[3], Nika Mansouri Ghiasi[1], Stefano Mercogliano[1], Furkan Eris[1], Joel Lindegger[1], André Kahles[3,5,6,*], Onur Mutlu[1,*]

[1]Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich, 8092, Switzerland
[2]Department of Computer Science, University of Maryland, MD 20742, United States
[3]Department of Computer Science, ETH Zurich, Zurich, 8092, Switzerland
[4]Max Planck Institute for Intelligent Systems, Tubingen, 72076, Germany
[5]University Hospital Zurich, Zurich, 8091, Switzerland
[6]Swiss Institute of Bioinformatics, Zurich, 8057, Switzerland

Two Ways to Build an Assembly Graph

- Overlap graphs: connect whole reads through overlaps
- **De Bruijn graphs: connect k-mers (DNA words of length k)**

Overlap graph

Nodes: whole reads

Edges: overlaps found by comparing read pairs

Seeds pick candidates, alignment confirms

Cost grows with the number of read pairs

Repeats stay inside reads, so long reads separate copies

De Bruijn graph, k = 4

Nodes: distinct k-mers from a hash table

Edges: consecutive windows, k – 1 shared bases

One pass over the reads, no pairwise comparison

Every repeated k-mer collapses into one node

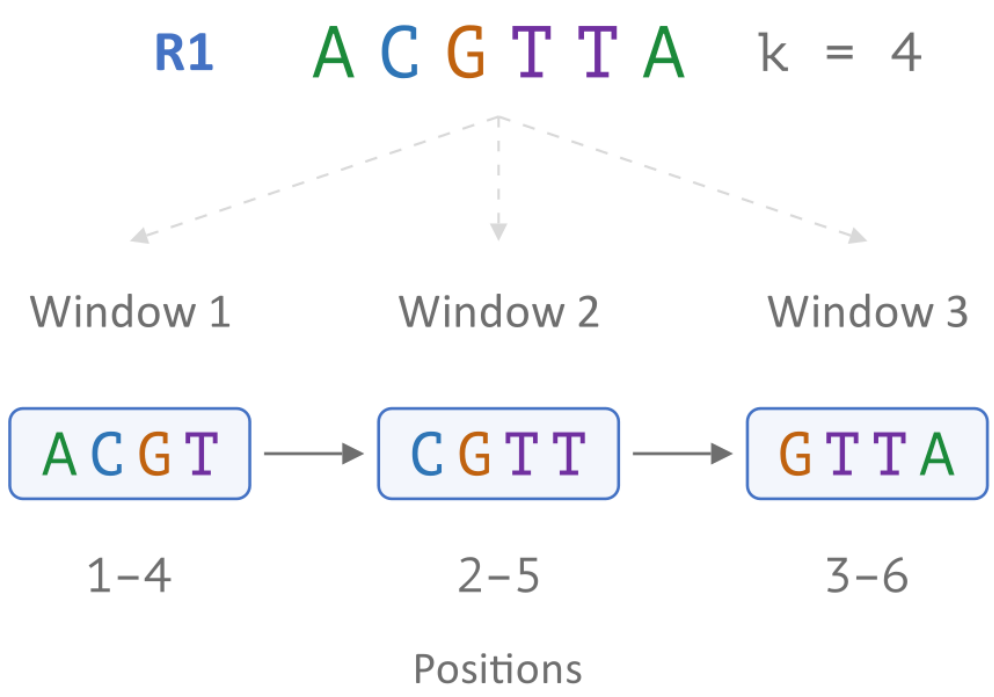
Read identity is lost unless reads are threaded back

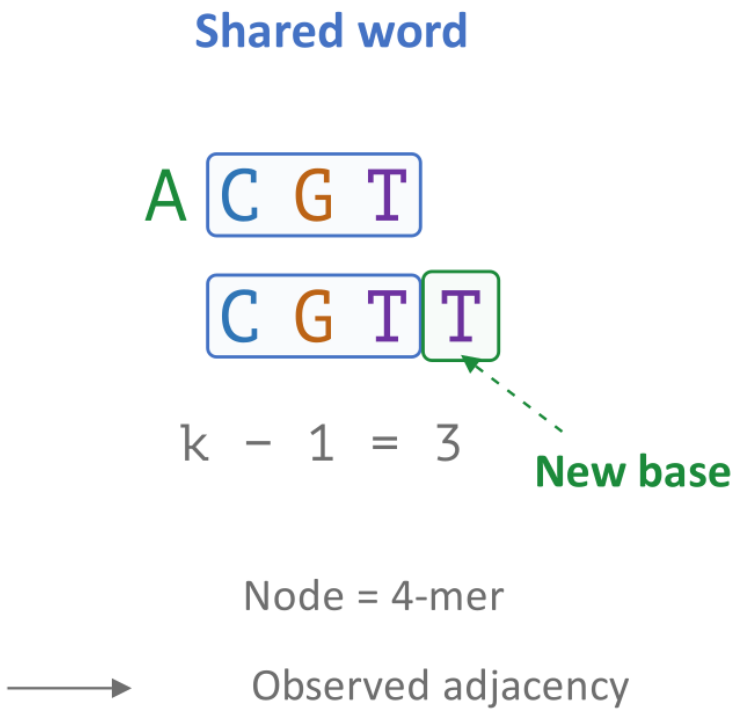
Same three reads, same contig

A C G T T A G C C A

From a Read to K-mer Nodes

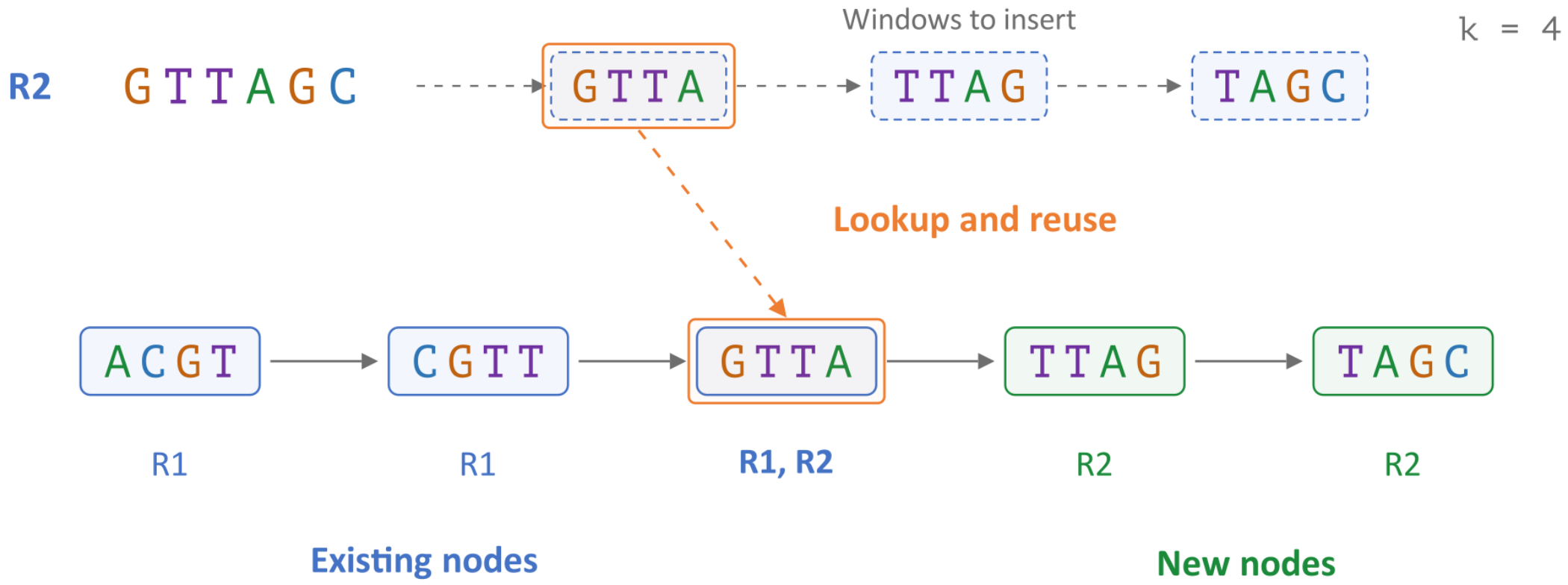
- Create one node for each distinct k-mer
- Consecutive windows share k − 1 bases





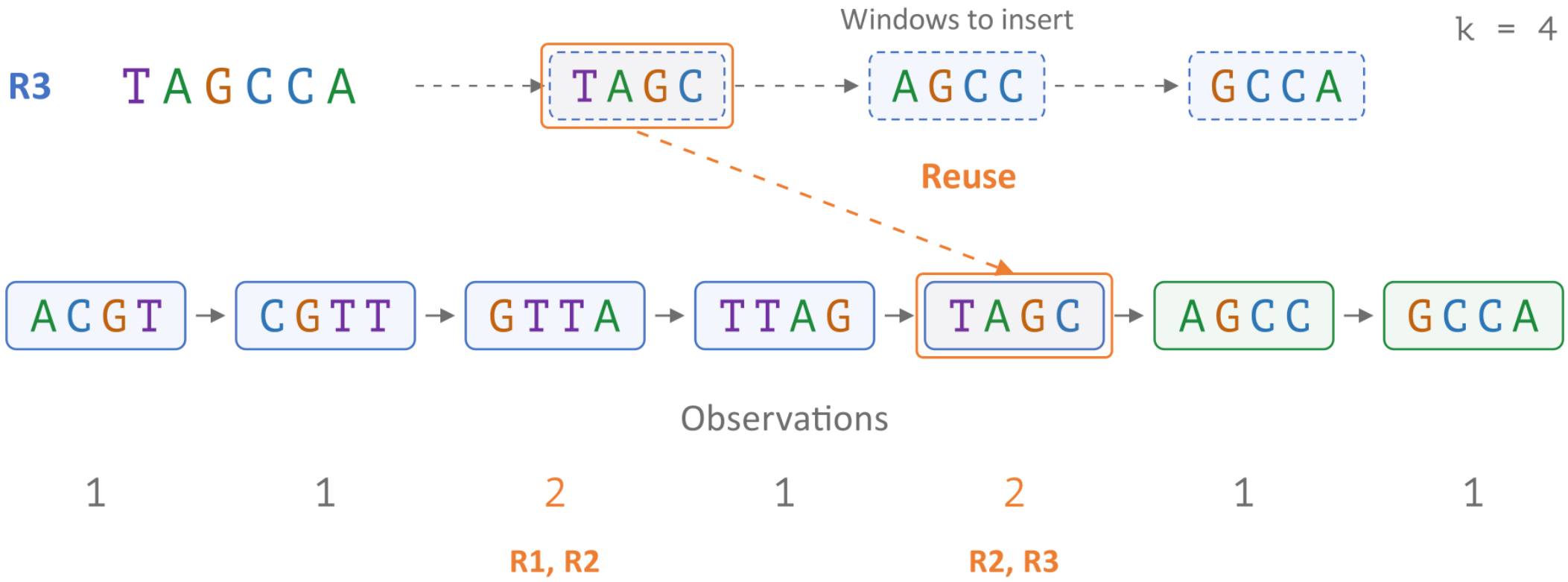
An Identical K-mer Reuses Its Node

- Look up the full k-mer sequence
- Reuse its node and update the observed support



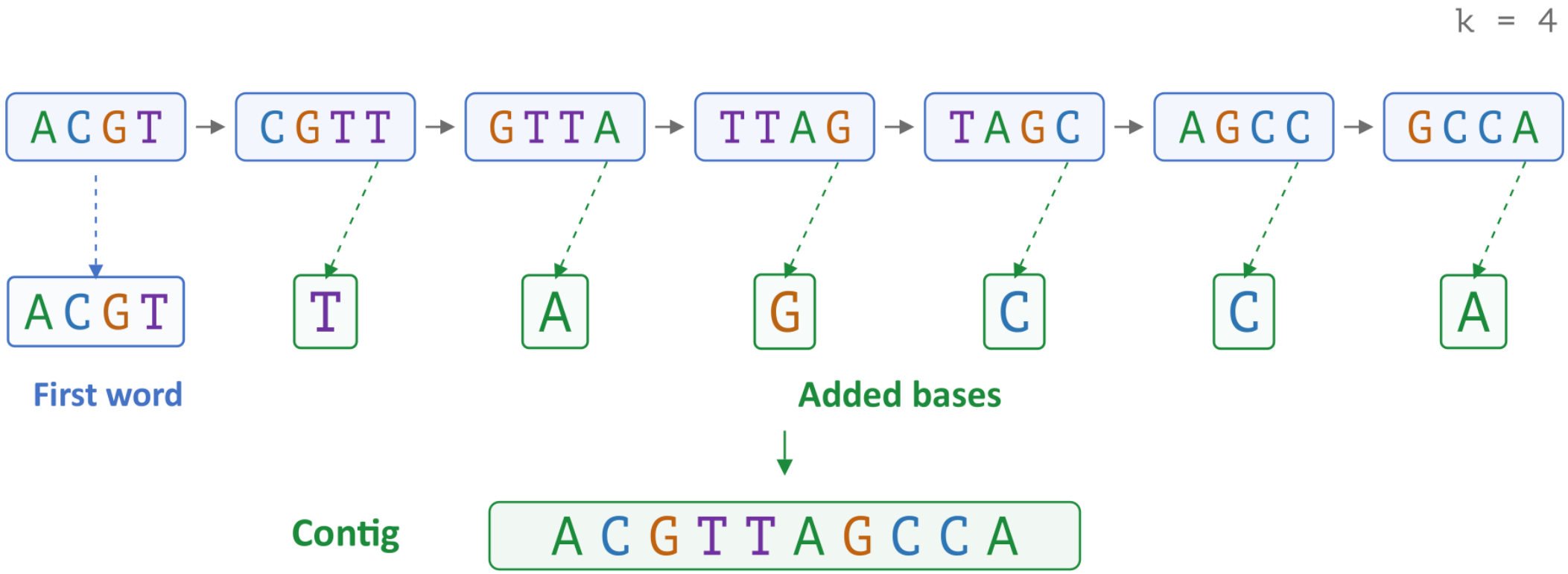
Shared Words Connect the Read Paths

- The third read reuses TAGC and adds two new words
- Counts come from observations in the input reads



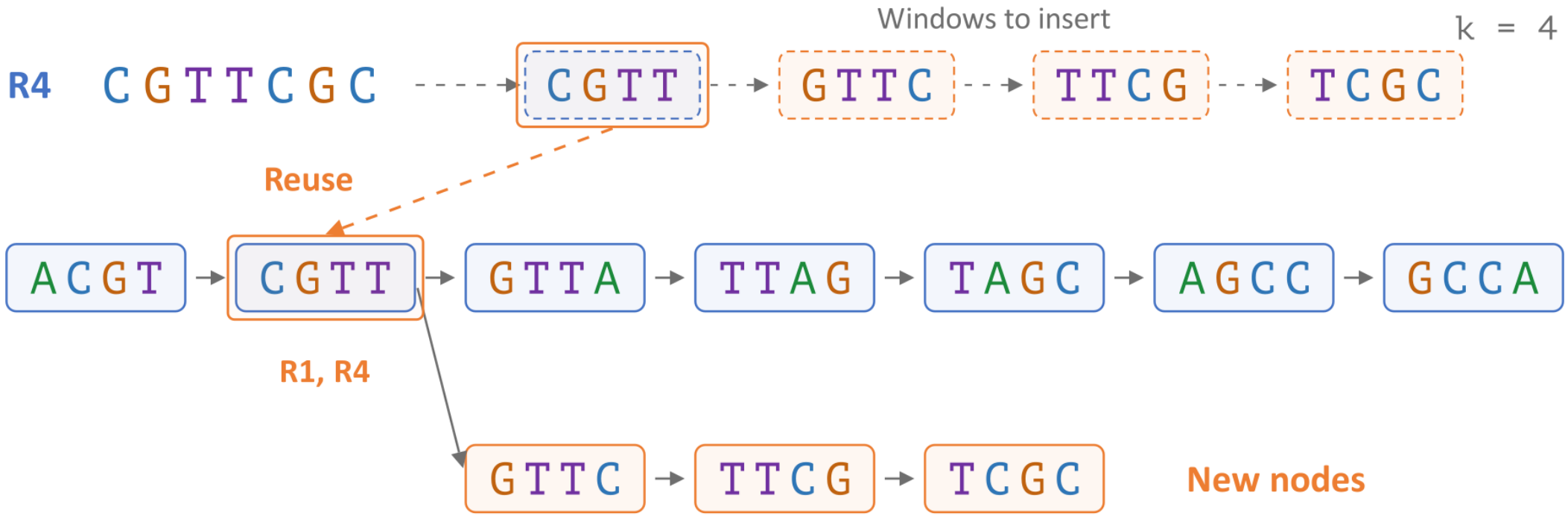
Spelling a Path

- Write the first k-mer, then append one new base per step



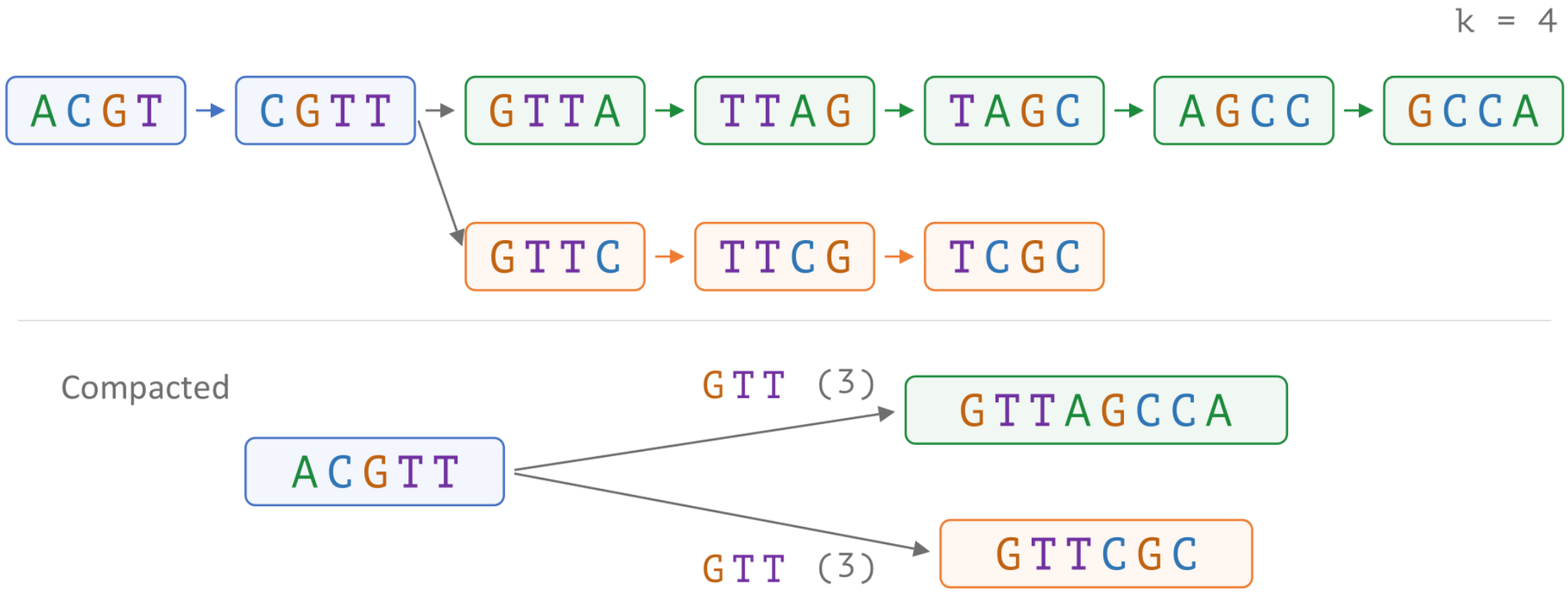
Another Read Creates a Branch

- A new read supports a second continuation after CGTT
- Keep both connections until there is enough evidence



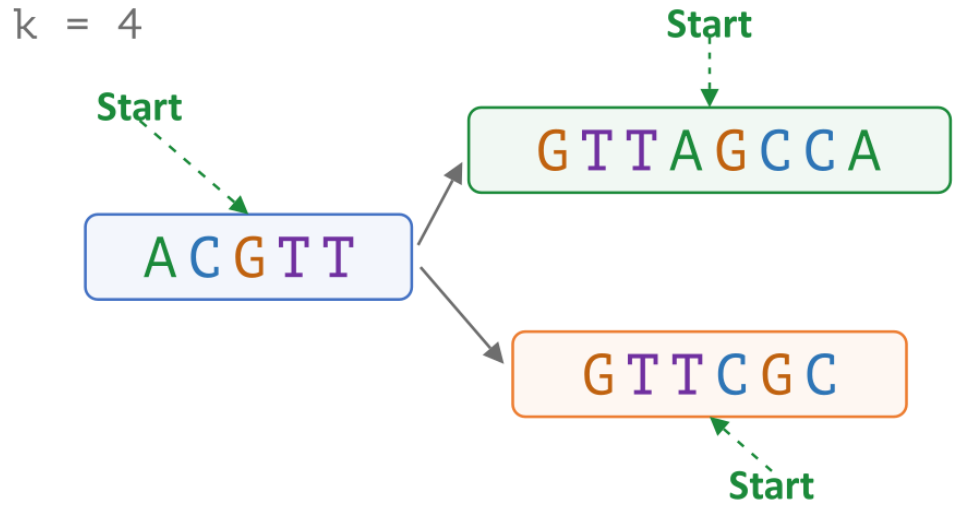
A Unitig Stops at an Unresolved Branch

- Merge u and v only when u has one exit and v has one entrance
- Compaction preserves the unresolved alternatives



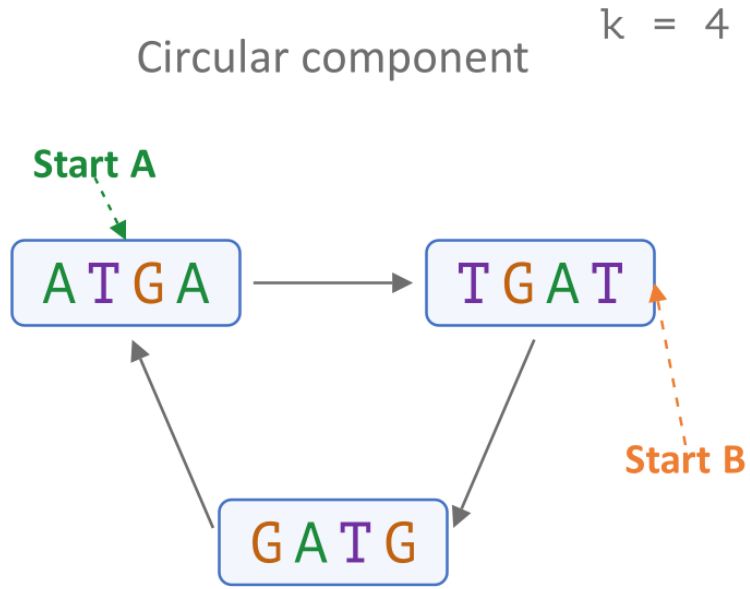
Finding Every Unitig

- Find maximal unambiguous paths throughout every component
- Start where the rule fails, or anywhere and then extend both ways
- An isolated cycle has no start: any node gives a rotation of it



A path starts where the merge rule fails:
no single predecessor, or a predecessor with several exits

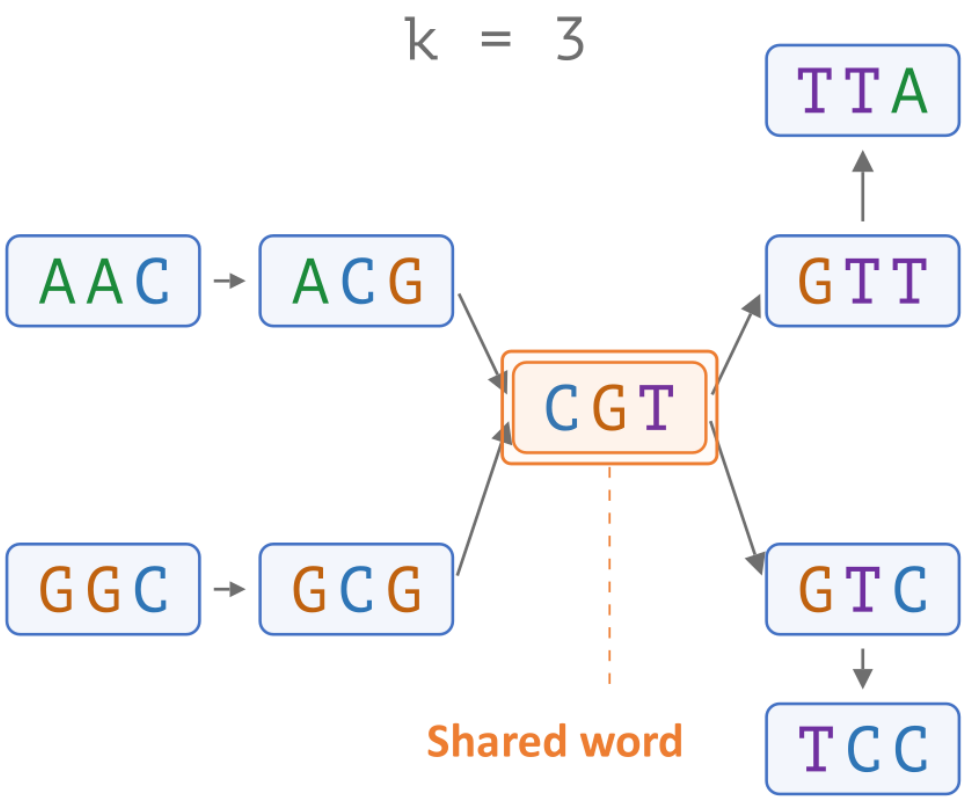
Or begin anywhere and extend both ways until the rule fails

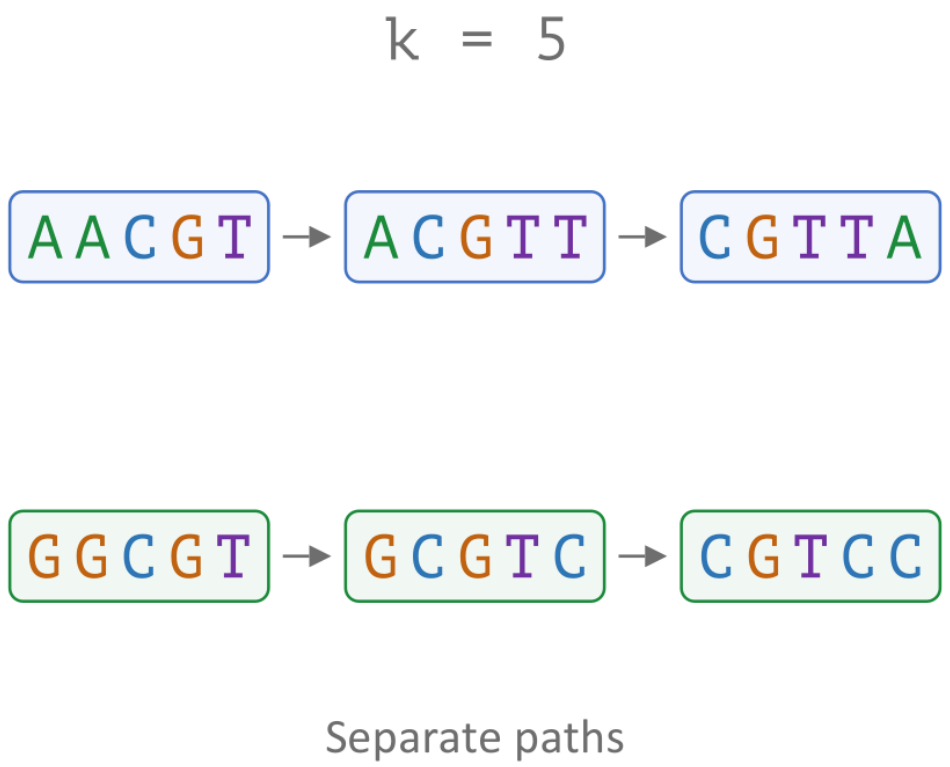


Every node passes the rule, so any node is a start
Each start spells a rotation of the same circle
A graph cycle alone does not prove a circular chromosome

Choosing k

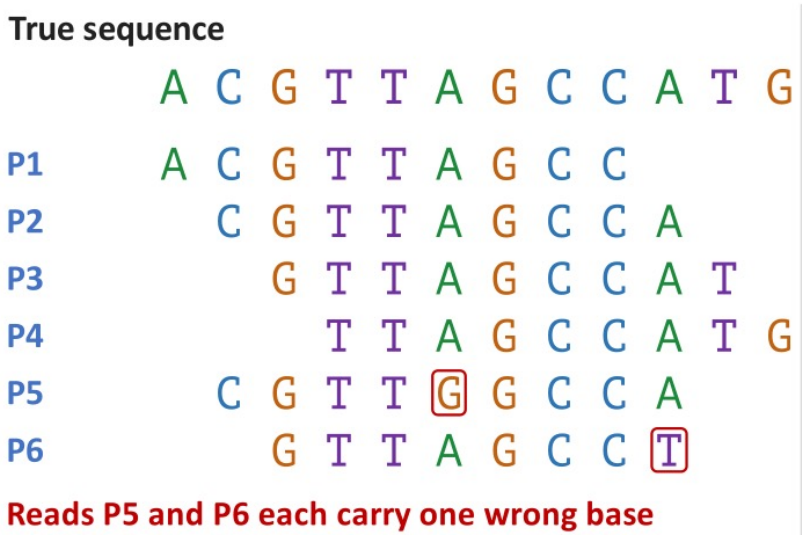
- Smaller k can merge distinct genomic contexts
- Larger k requires longer exact, supported words

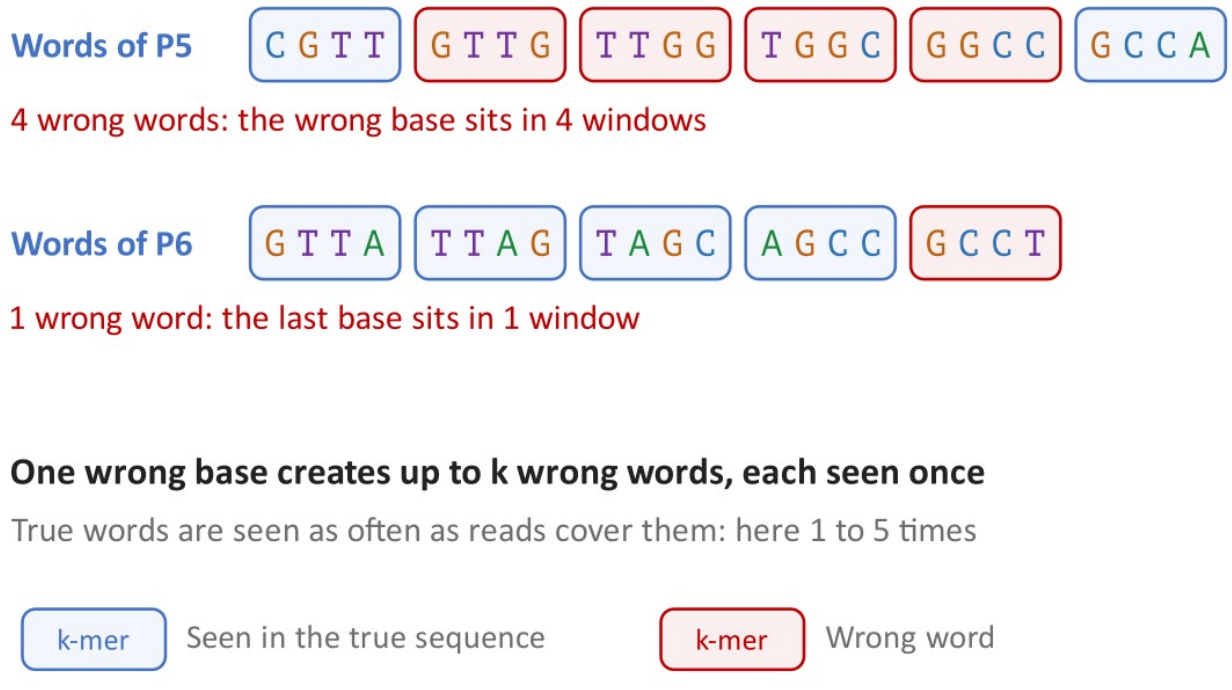




Sequencing Errors Create Rare Words

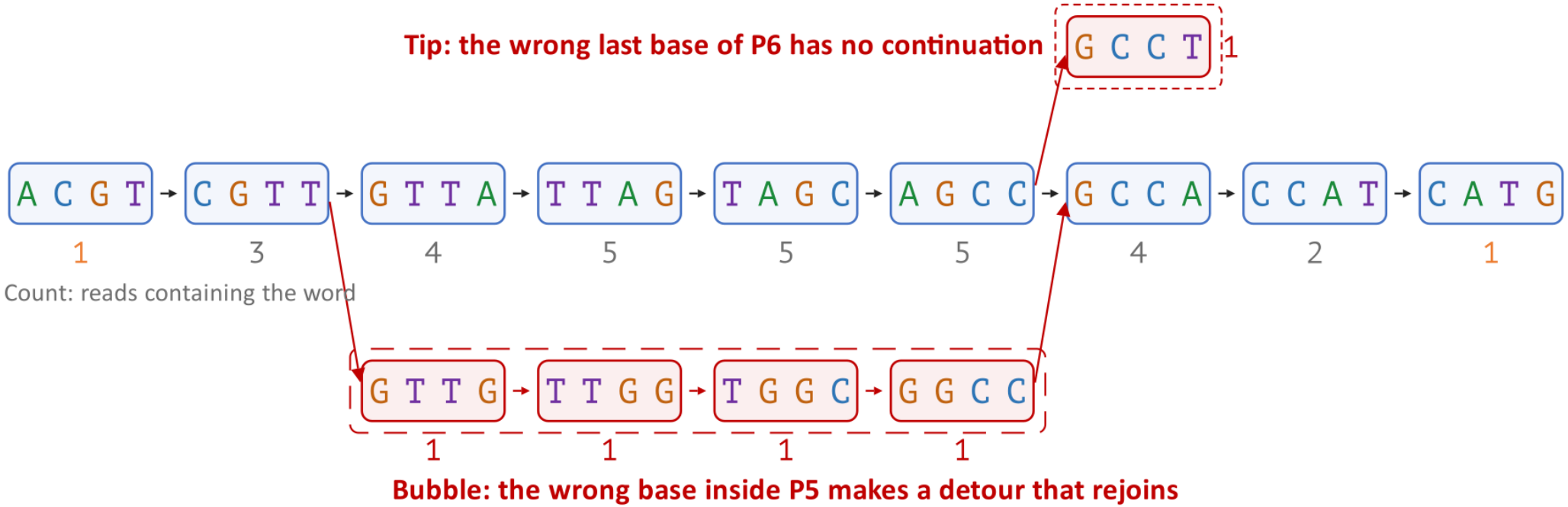
- A wrong base changes every window that contains it: up to k wrong k-mers
- Wrong words are seen once, true words as often as reads cover them





Errors in the Graph: Bubbles and Tips

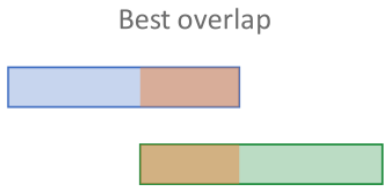
- An error inside a read makes a bubble, an error near a read end makes a tip
- Drop rare words and short tips, then collapse near-identical bubbles



Seen once: drop by a count cutoff, by tip length, or by bubble similarity Read ends are rare too: a plain cutoff would also drop ACGT and CATG

Assembler Families at a Glance

- Every family builds a graph, then walks its unambiguous paths
- They differ in what a node is and in how much read context survives



Greedy extension

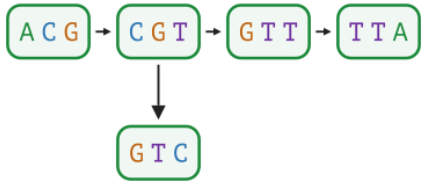
Take the longest overlap first, then extend and repeat

Phrap, TIGR Assembler
early short-read tools

Overlap graph

Reads are nodes, overlaps are edges

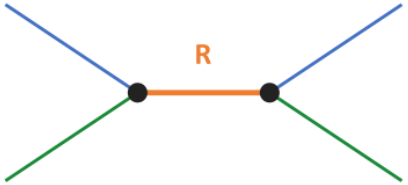
Celera, Canu,
miniasm, hifiasm



De Bruijn graph

K-mers are nodes, exact k − 1 overlaps

Velvet, SPAdes,
ABySS, MEGAHIT



Repeat graph

Edges carry sequence, each repeat appears once

Flye

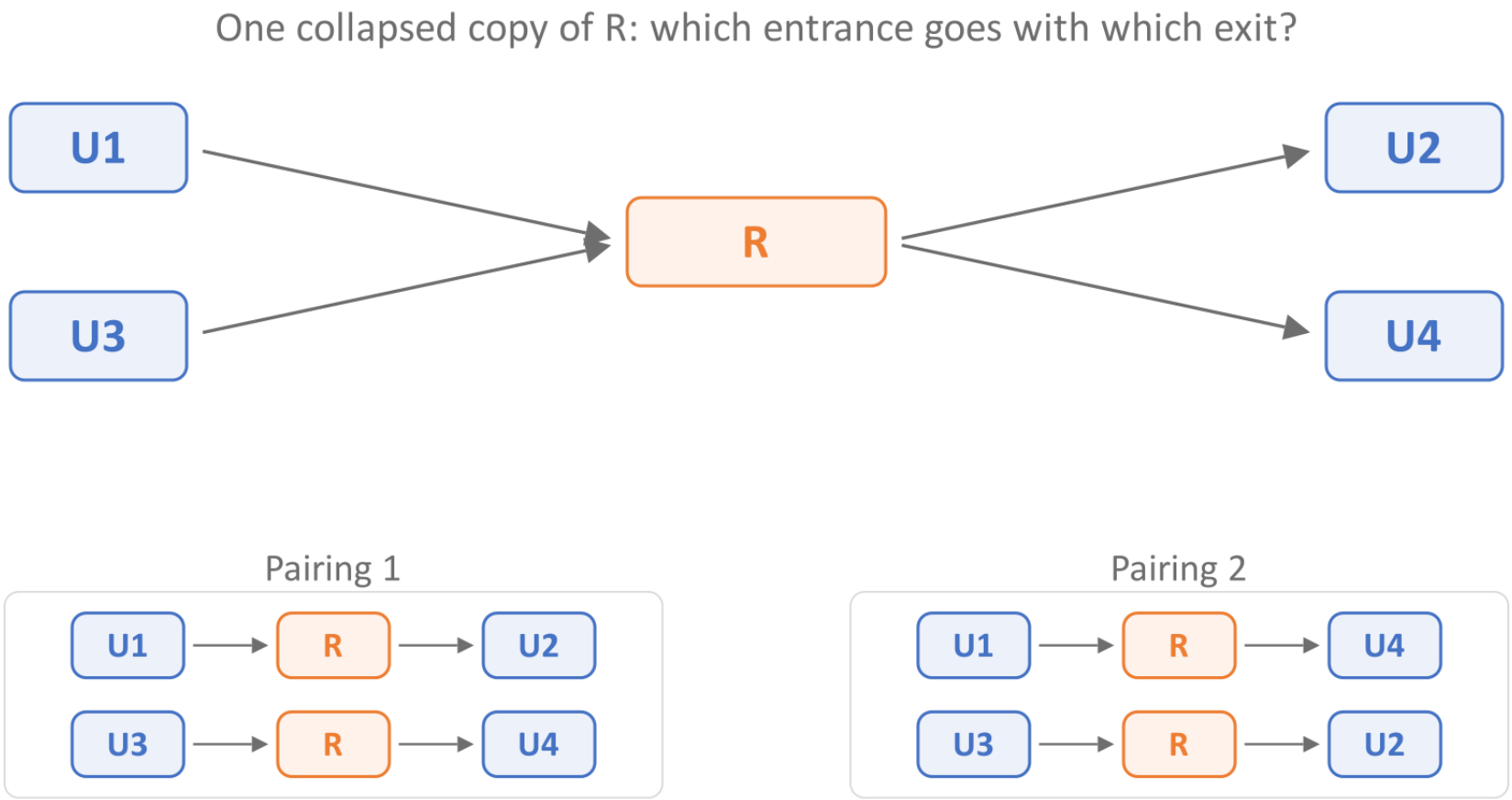
Multiplex de Bruijn

Local k grows where long accurate reads allow

MBG in Verkko, LJA,
metaMDBG in minimizer space

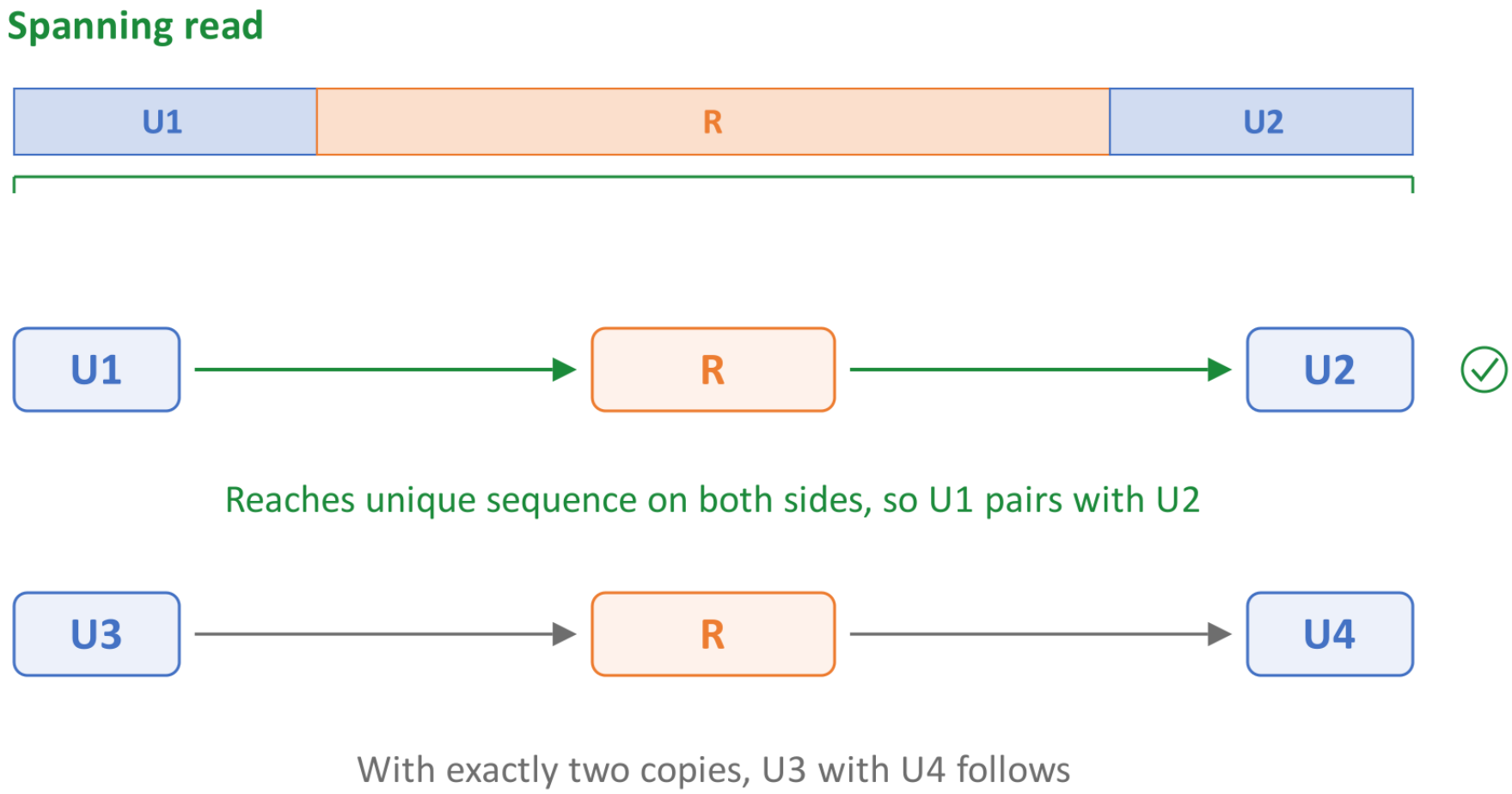
Repeats Can Hide the Correct Connections

- Two repeat copies allow more than one pairing



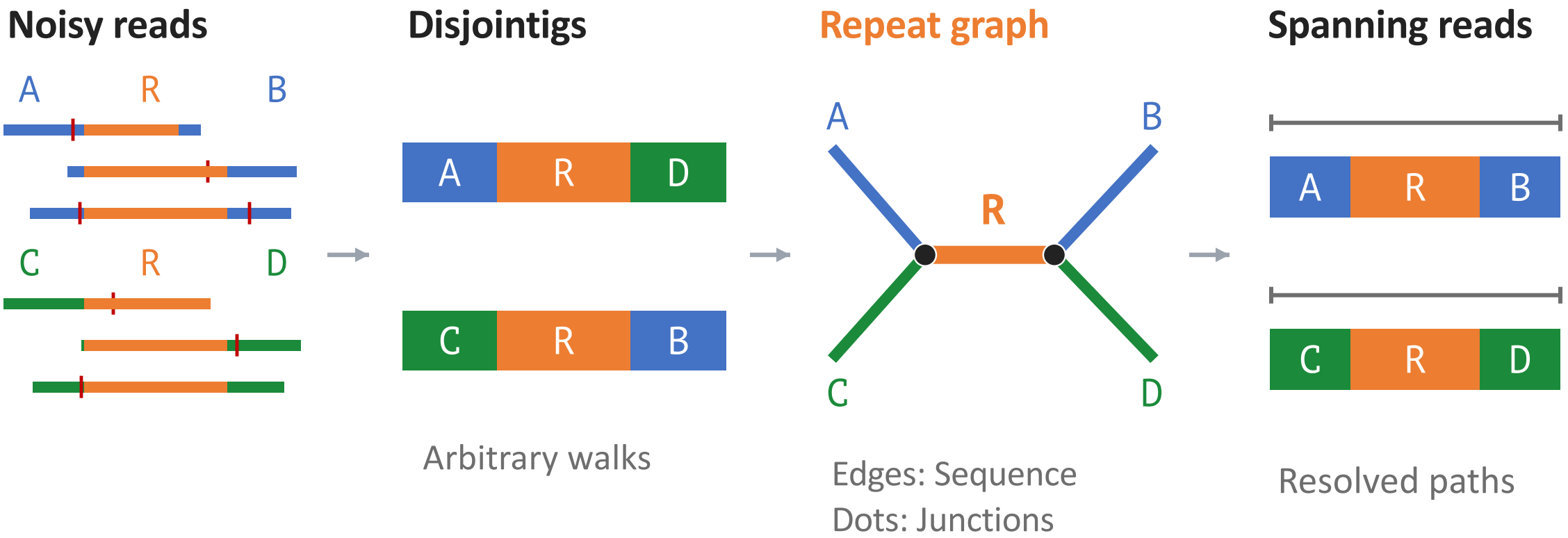
A Spanning Read Resolves a Connection

- A read reaching both unique flanks identifies their pairing



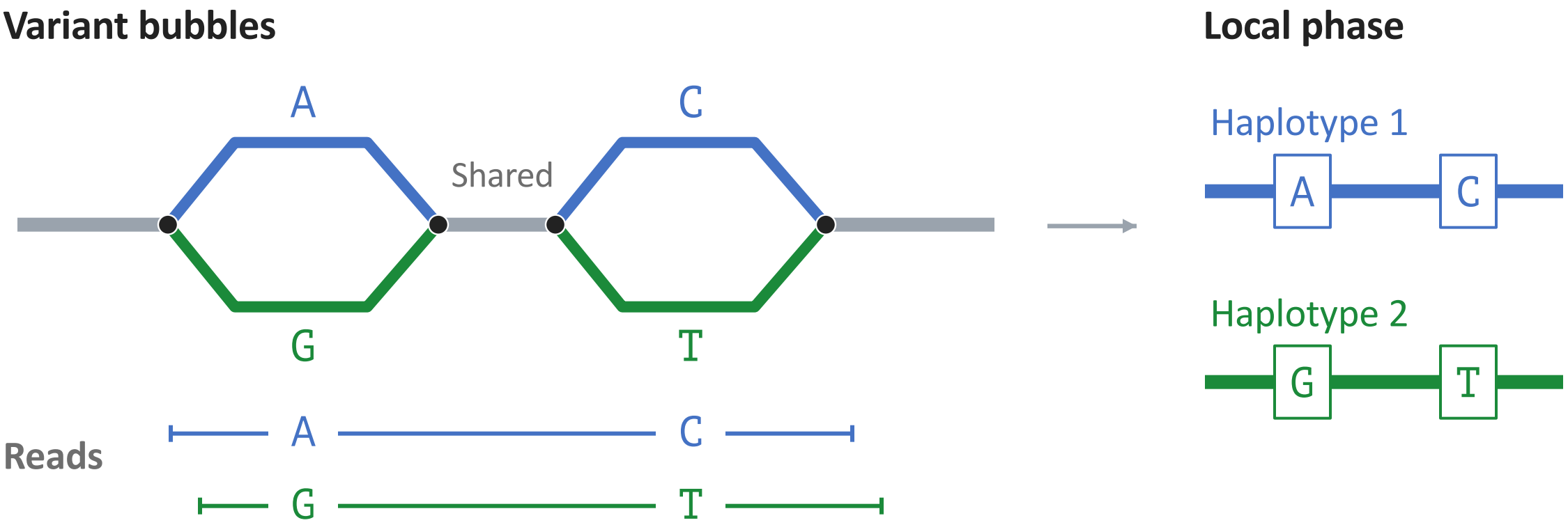
Flye: A Repeat Graph

- Build approximate repeat structure from long reads
- Use read paths to resolve supported repeat connections



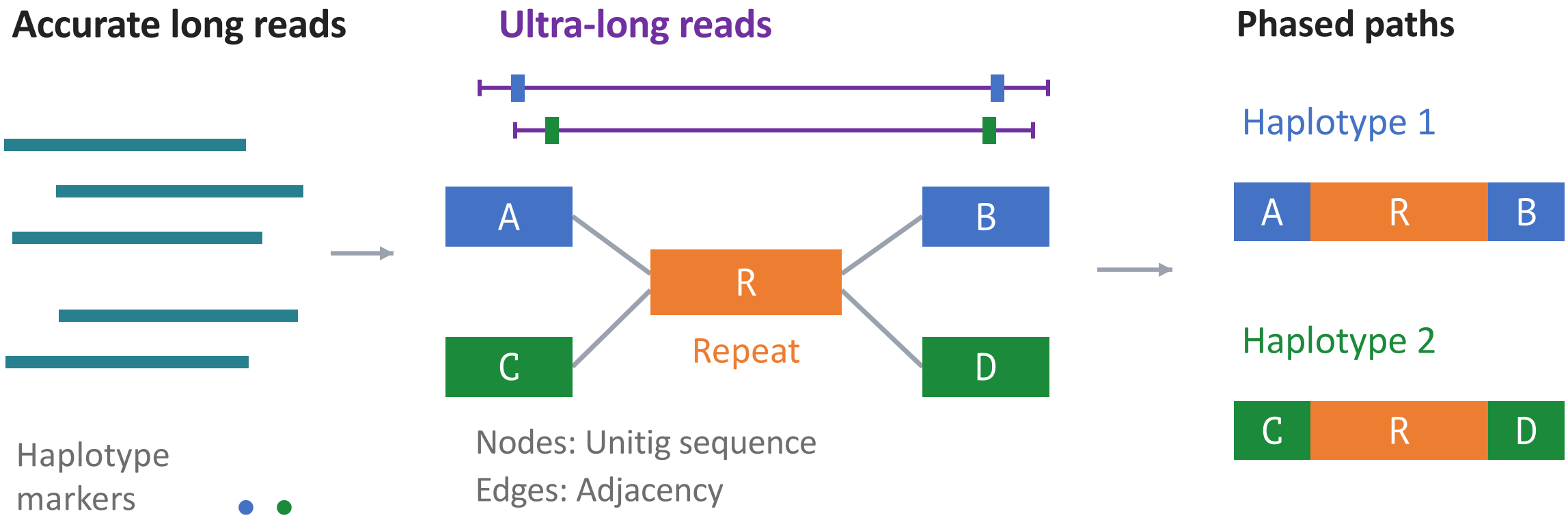
Hifiasm: Preserving Haplotype Paths

- Keep coherent alleles together through the assembly graph
- Accurate reads and phasing evidence help separate haplotypes



Verkko: Combining Read Evidence

- Accurate long reads construct the initial sequence graph
- Ultra-long reads and phasing evidence resolve more ambiguity



Genome Assembly: Key Ideas

- Coverage and read length bound what any assembler can resolve
- Overlaps connect reads through shared sequence
- Graph cleaning removes redundancy but can also lose real sequence
- Errors appear as rare tips and bubbles in every graph type
- Read paths help resolve repeats and preserve haplotypes
- Consensus improves bases, scaffolding connects contigs

Questions?

Next Lecture

- AI/ML in genomics

CMSC 829C: Algorithms, AI, and Hardware for Biological Data

Lecture 5: Genome Assembly

Fall 2026

September 15, 2026

Can Firtina | canfirtina@gmail.com | cs.umd.edu/~firtina

Assistant Professor of Computer Science at UMD

UNIVERSITY OF MARYLAND