Revisiting Graph-theoretic Models for Genome Assembly in the Era of Long Reads

Chirag Jain
Assistant Professor
Computational and Data Sciences
Indian Institute of Science

Based on Bioinformatics paper
https://doi.org/10.1093/bioinformatics/btad124

CiE (Computability in Europe) 2023
July 24, 2023

A human genome is a set of 46 strings

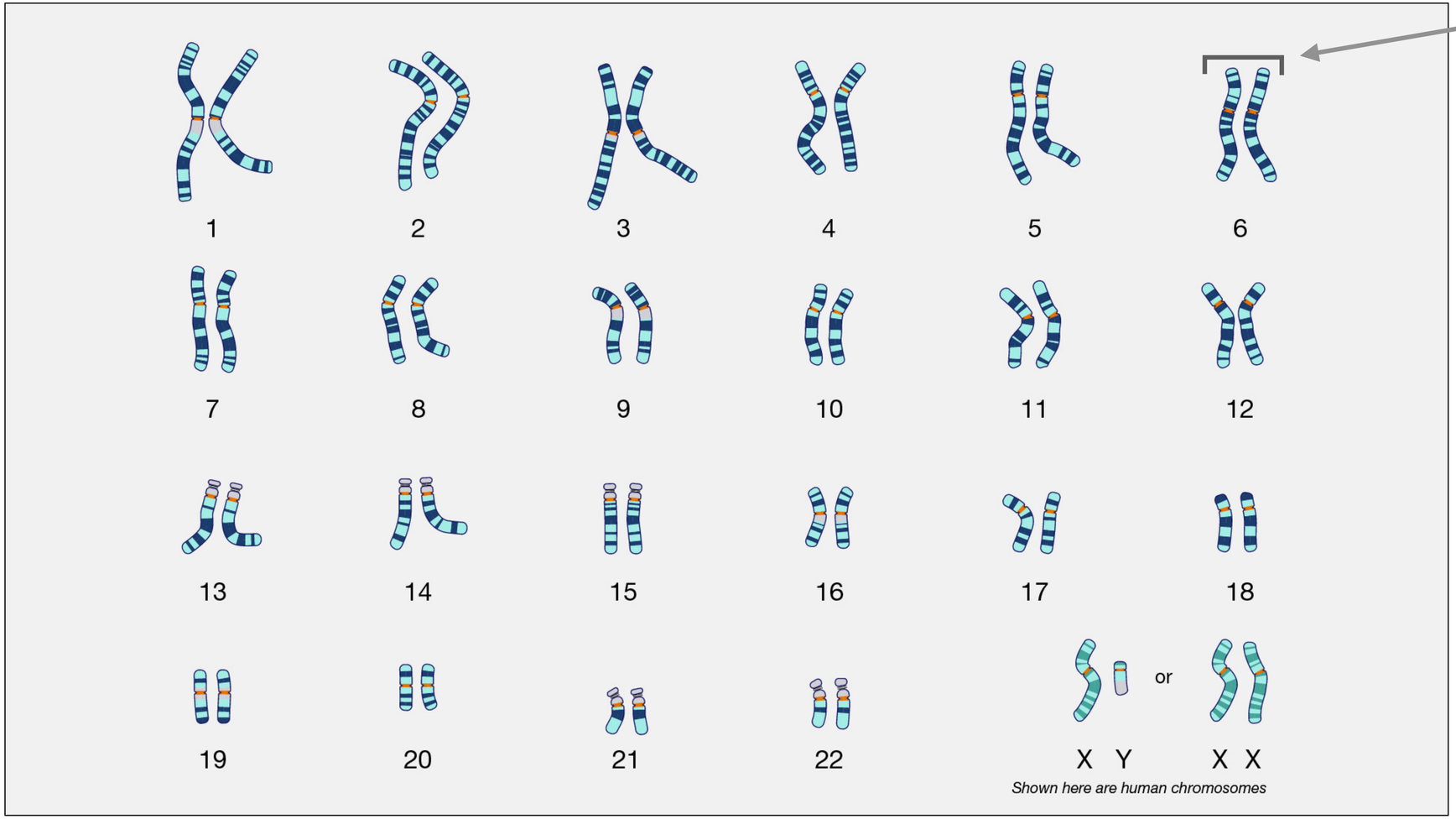
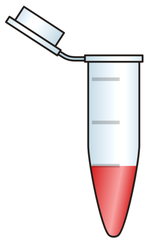


Image source: NHGRI

Two “copies”, inherited from father and mother (~99.9% identical)

Sum total length is about 6 billion characters

Human-genome sequencing



Human DNA

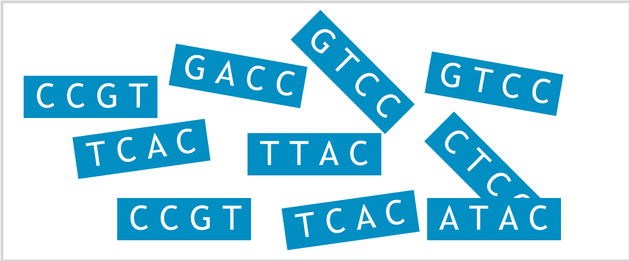
A sequencing instrument

(IDEAL)

```
TNTTCGTTTCTCTTCTGCTATCTGTGGTCTGTG
TCTGGGATGCTGACTGTCTCCCTGGGAAGGCAC
AACTCTCGTGACCCCAGCCTGGAGGCCCACATT
TTCTGCTTCTTTGTGATATCATCCTGTGCTGCC
GGCGCGACAAAATAGGGGTGATGGTTTGTGTT
GAGCACGCGCGGCAGAGAGGAAAAATGGGCTC
CCAAAGCGCCTCGGGAGATGGGGAGGGTAGCCA
```

Text file with the 46 strings

(ACTUAL)



Reads

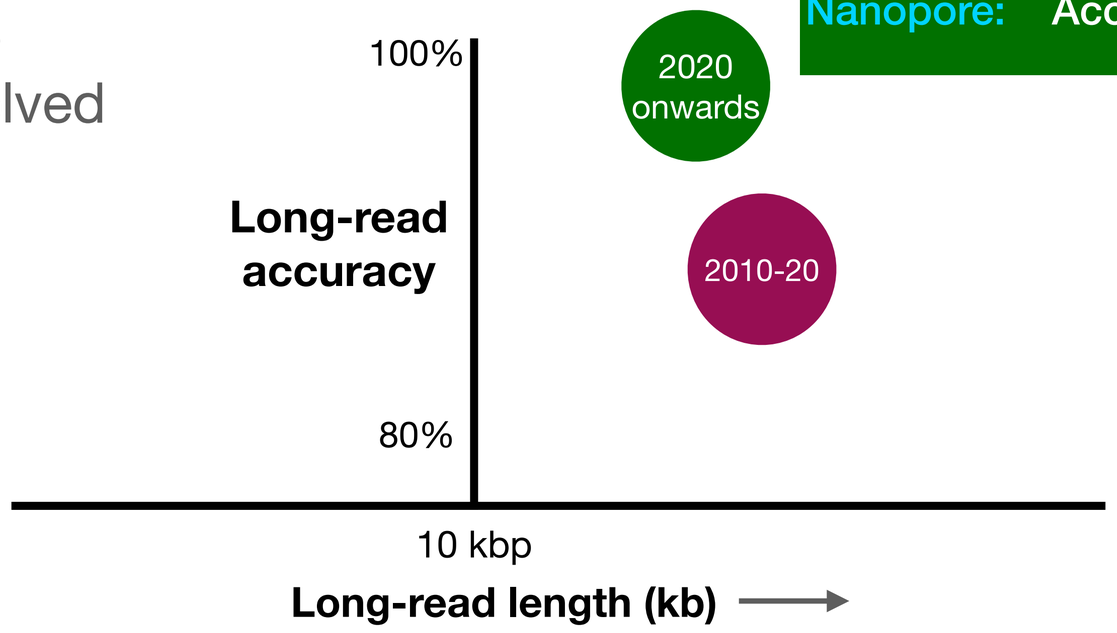
Genome assembly: Reconstruction of the original genome from reads

Latest: Long and accurate sequencing

- Enables *de novo* genome assembly of both maternal and paternal haplotypes
- Was not feasible using previous technologies

State-of-the-art assemblies:
~~3 Gbp (collapsed)~~
6 Gbp haplotype-resolved

PacBio HiFi: Accuracy = 99.8%, Avg. length > 10 kbp
Nanopore: Accuracy = ~99%, Avg. length > 40 kbp



Latest: Long and accurate sequencing

Thanks to long-read assemblers !

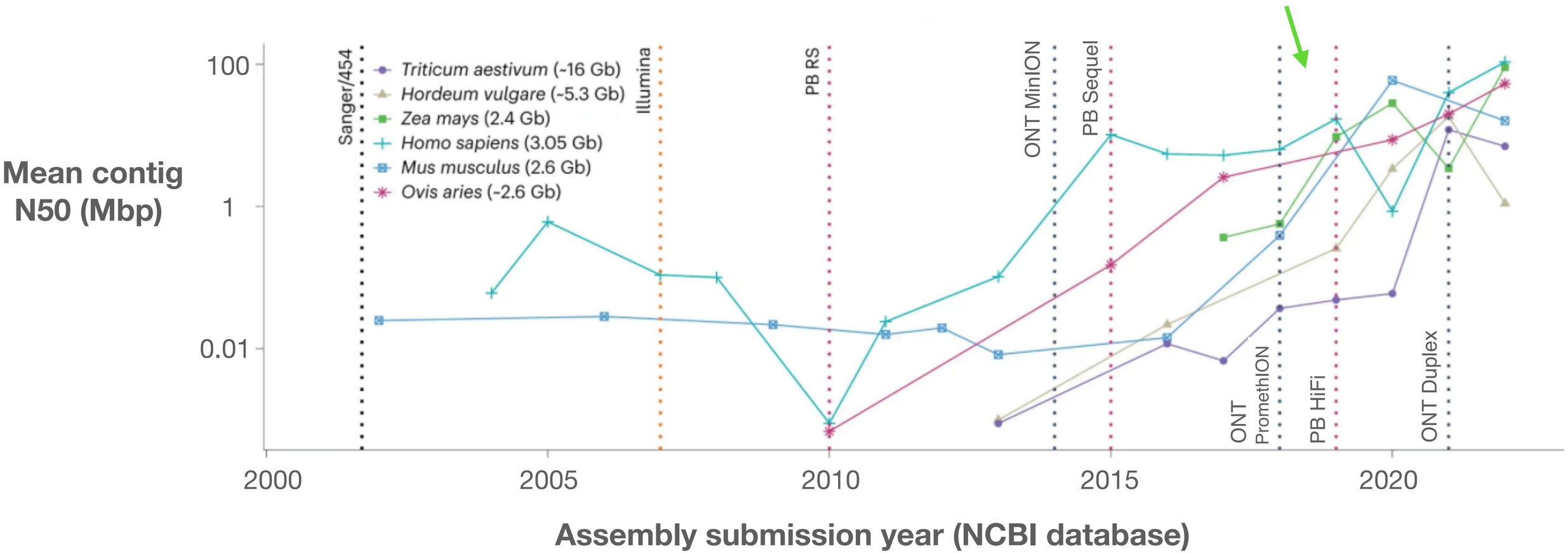


Figure from Kovaka et al. 2023, “Approaching complete genomes, transcriptomes and epi-omes with accurate long-read sequencing”

Graph-theoretic models for assembly

- Input: Set of reads R
  - **De Bruijn graph** : $B_k(R)$ [Idury and Waterman 1995]
    - Vertices are distinct k-mers observed in R
    - An edge implies a suffix-prefix overlap of length $k-1$ between two k-mers

Graph-theoretic models for assembly

- Input: Set of reads R
  - **De Bruijn graph** : $B_k(R)$ [Idury and Waterman 1995]
    - Vertices are distinct k-mers observed in R
    - An edge implies a suffix-prefix overlap of length $k-1$ between two k-mers
  - **Overlap graph** : $O_k(R)$
    - Vertices are input reads
    - An edge implies an exact suffix-prefix match of length $\geq k$ between two reads

Graph-theoretic models for assembly

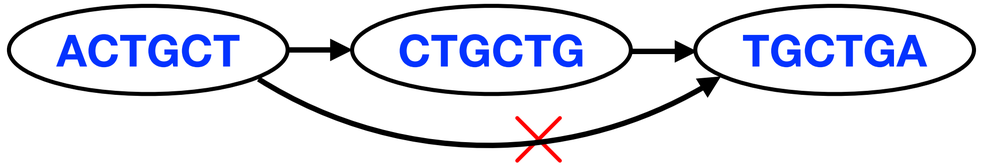
- Input: Set of reads R
  - **De Bruijn graph** : $B_k(R)$ [Idury and Waterman 1995]
    - Vertices are distinct k-mers observed in R
    - An edge implies a suffix-prefix overlap of length $k-1$ between two k-mers
  - **Overlap graph** : $O_k(R)$
    - Vertices are input reads
    - An edge implies an exact suffix-prefix match of length $\geq k$ between two reads
  - **String graph** : $S_k(R)$ [Myers 1995, 2005]
    - Subgraph of $O_k(R)$
    - Next slide…

String graph

- Used in most long-read assemblers
- **Sub-graph of overlap graph** [Myers 1995, 2005]
  - Keep only the longest suffix-prefix overlap between a pair of reads
  - Remove contained reads
  - Remove transitive edges

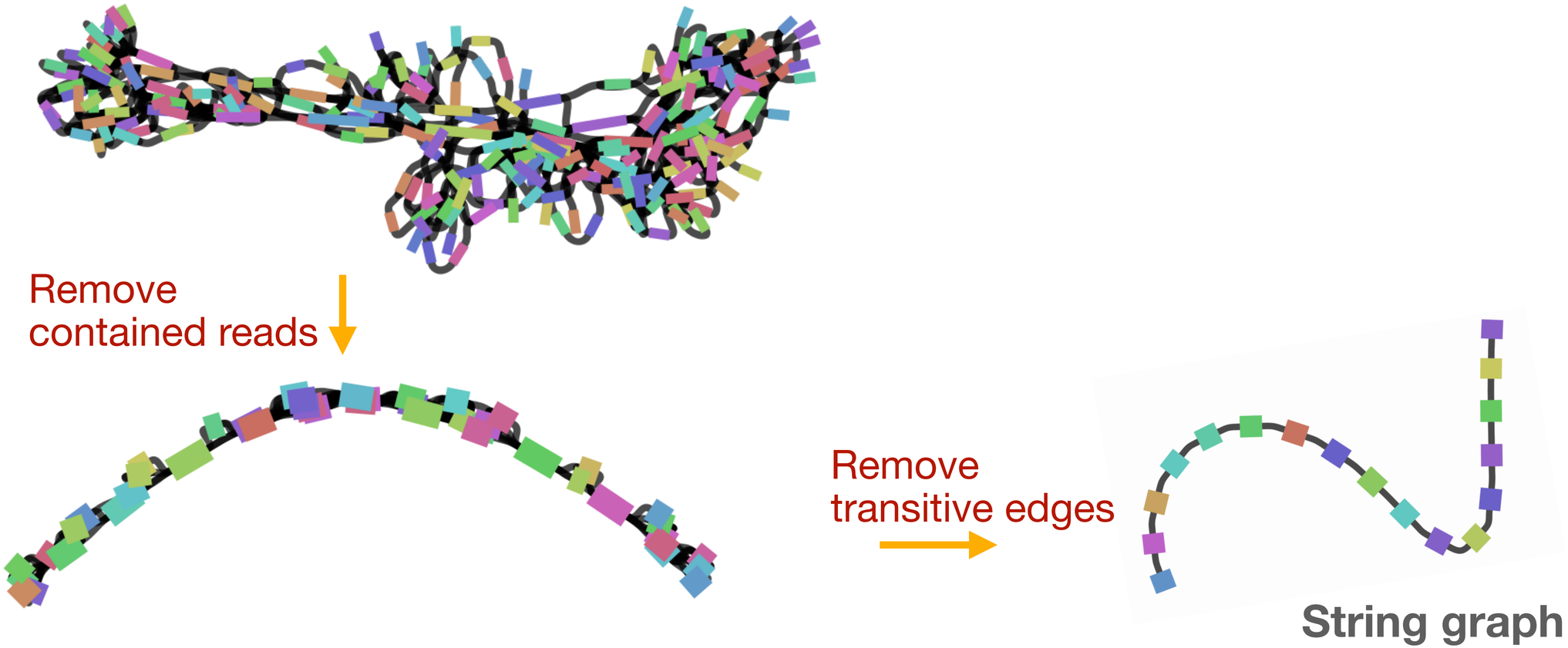
```
ACTGCTTAC
 CTGCTT
```

```
ACTGCT
 CTGCTG
  TGCTGA
```



Graph sparsification cannot be avoided in practice

Overlap graph obtained using a subset of simulated nanopore reads from human chr20



Are graph models “coverage-preserving”?

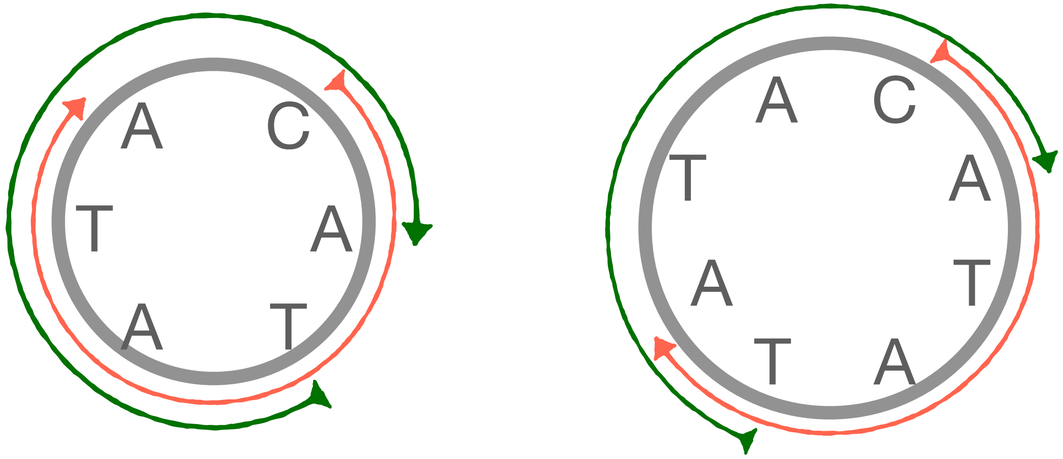
- Suppose input reads cover the entire genome, do we have a guarantee that the “true” chromosomes can be spelled as a walk in the graph?

⚠️

Are graph models "coverage-preserving"?

- **Suppose input reads cover the entire genome, do we have a guarantee that each candidate chromosome can be spelled as a walk in the graph?** ✓

Say $R =$ TATACA, CATATA

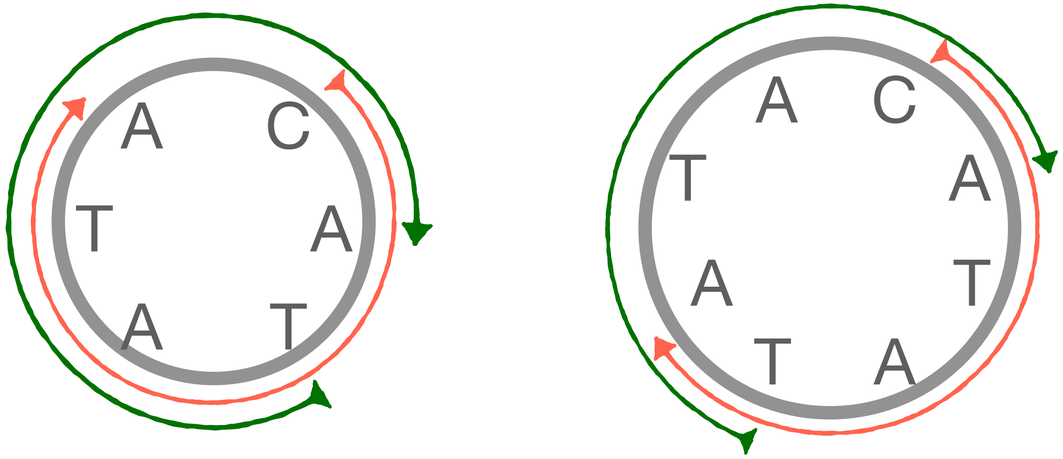


Candidate 1 Candidate 2

Are graph models "coverage-preserving"?

- **Suppose input reads cover the entire genome, do we have a guarantee that each candidate chromosome can be spelled as a walk in the graph?**

Say $R =$ TATACA, CATATA



Candidate 1 Candidate 2

- Circular string z is a candidate if $\exists l_1, l_2 \in \mathbb{N}, l_2 > l_1$ such that all intervals of length l_1 in z include the starting position of at least one read of length l_2

Theoretical evaluation

- **Input:** set of reads R

Graph model	Guarantee?	Proof technique
de Bruijn graph $B_k(R)$	YES $\forall\ k \leq l_2 - l_1 + 1$	By contradiction
Overlap graph $O_k(R)$	YES $\forall\ k \leq l_2 - l_1$	Algorithm to identify a closed walk in the graph for each candidate
String graph $S_k(R)$	NO for any k	Counter-example

Consistent with prior works [e.g., Hui et al. ISIT 2016]

Jain "Coverage-preserving sparsification of overlap graphs for long-read assembly" *Bioinformatics* 2023

Theoretical evaluation

- **Input:** set of reads R

Graph model	Guarantee?	Proof technique
de Bruijn graph $B_k(R)$	YES $\forall\ k \leq l_2 - l_1 + 1$	By contradiction
Overlap graph $O_k(R)$	YES $\forall\ k \leq l_2 - l_1$	Algorithm to identify a closed walk in the graph for each candidate
String graph $S_k(R)$	NO for any k	Counter-example

Consistent with prior works [e.g., Hui et al. ISIT 2016]

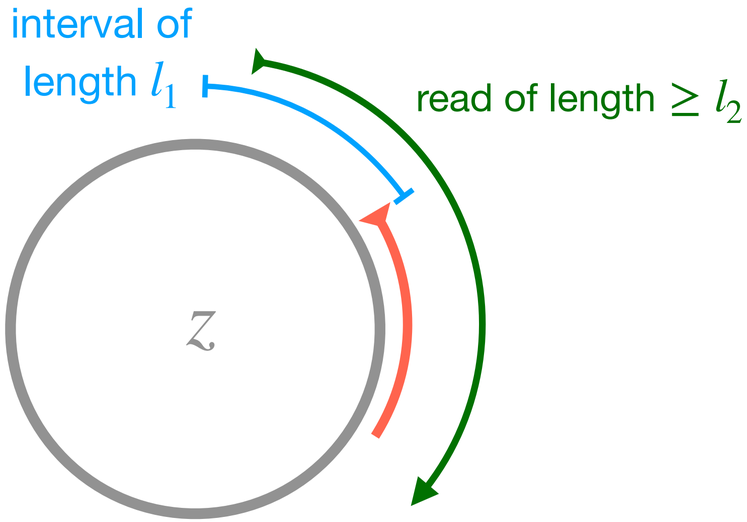
Jain "Coverage-preserving sparsification of overlap graphs for long-read assembly" *Bioinformatics* 2023

Proof sketch

Assembly-graph model	Coverage-preserving?	Proof technique
de Bruijn graph $B_k(R)$	YES $\forall\ k \leq l_2 - l_1 + 1$	By contradiction

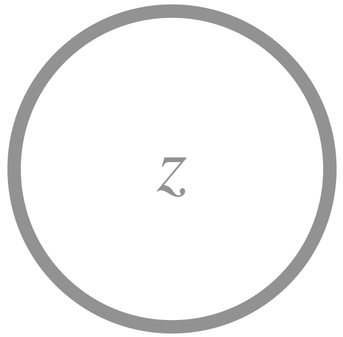
- Assume there is a candidate chromosome z not spelled by graph

$\Rightarrow$ At least one k-mer in z is absent from the set of vertices

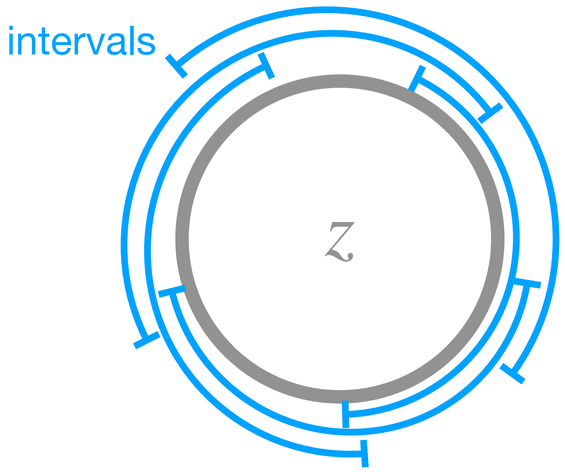


Proof sketch

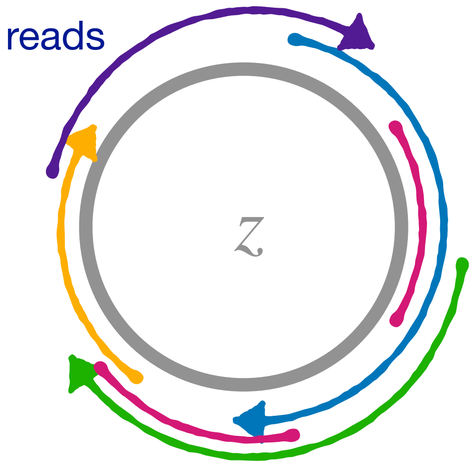
Assembly-graph model	Coverage-preserving?	Proof technique
Overlap graph $O_k(R)$	YES $\forall\ k \leq l_2 - l_1$	Proposed algorithm to identify a closed walk for each candidate



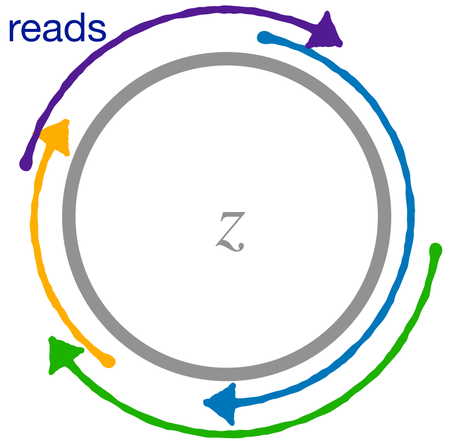
Candidate circular chromosome z



All intervals of length l_1 in z



Pull out reads of length $\geq l_2$ that start in the intervals



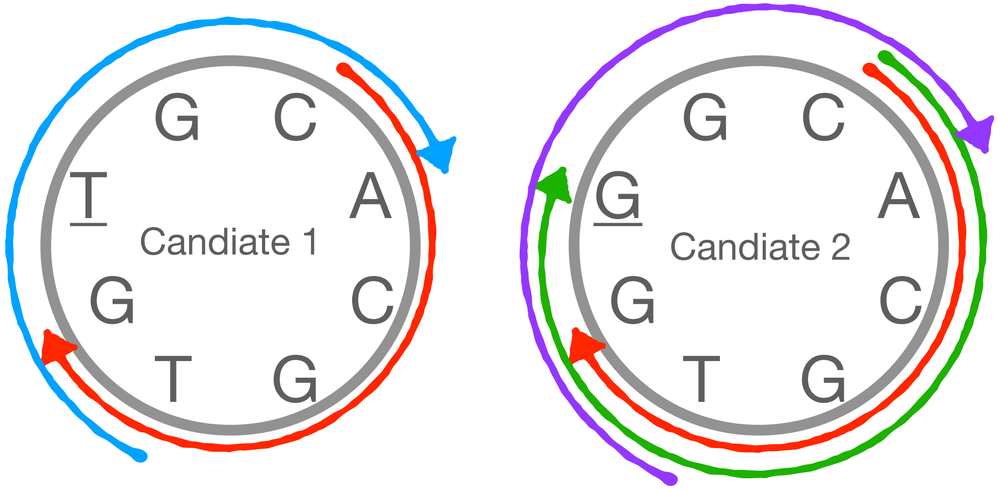
Identify a subset of these reads with suffix-prefix overlaps

Counter example for string graph

Assembly-graph model	Coverage-preserving?
String graph	NO for any k

Say $R =$

TGTGCA
CACGTG
CACGTGG
TGGGCA



⚠ Candidate 1 cannot be spelled in graph after contained read CACGTG is removed

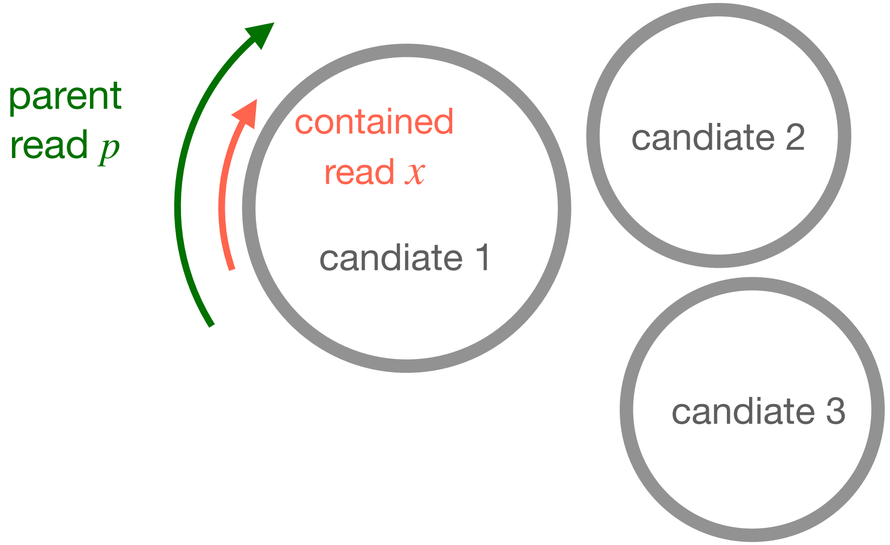
Further questions addressed

- Does it really matter for genome assembly quality in reality?
- Is there an alternate method to sparsify overlap graph that is practical and provably-good?
- Good heuristics to recover non-redundant contained reads?

‘Safe’ rules to sparsify overlap graph

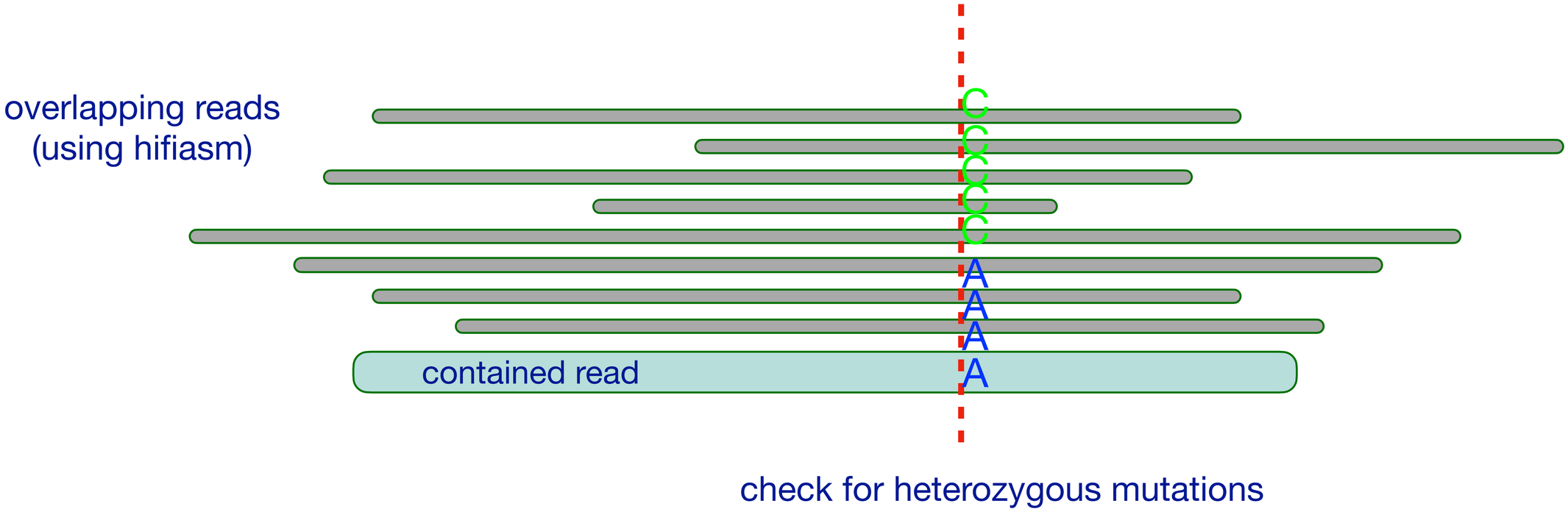
Practical?

- It is *safe* to remove a vertex (or edge) if the set of circular string walks remains unchanged ⚠
- Transitive edge reduction in [Myers 2005] is *safe* ✓
- Removing a contained read is *safe* if it maps to only a single candidate genome at a unique location [formal proof in paper] ⚠



Heuristic - 1

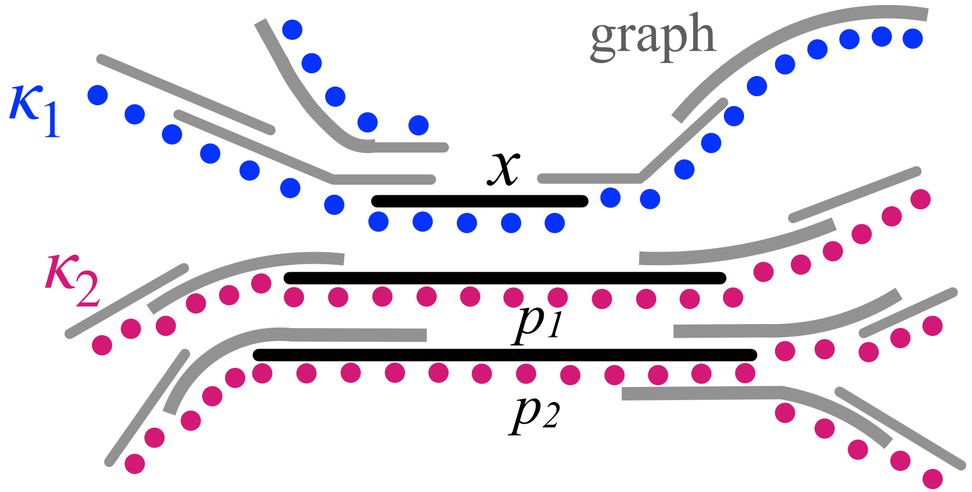
- By computing all-versus-all read alignments, we can estimate if a contained read maps uniquely within a single haplotype (either paternal or maternal)



Heuristic - 2

- Estimate if contained read contributes a “novel” string walk in the graph
- κ_1 = set of k-mers observed within bounded length string walks in the assembly graph from a contained read
- κ_2 = set of k-mers observed similarly from its “parent” reads
- Remove contained read if $\kappa_1 \subseteq \kappa_2$

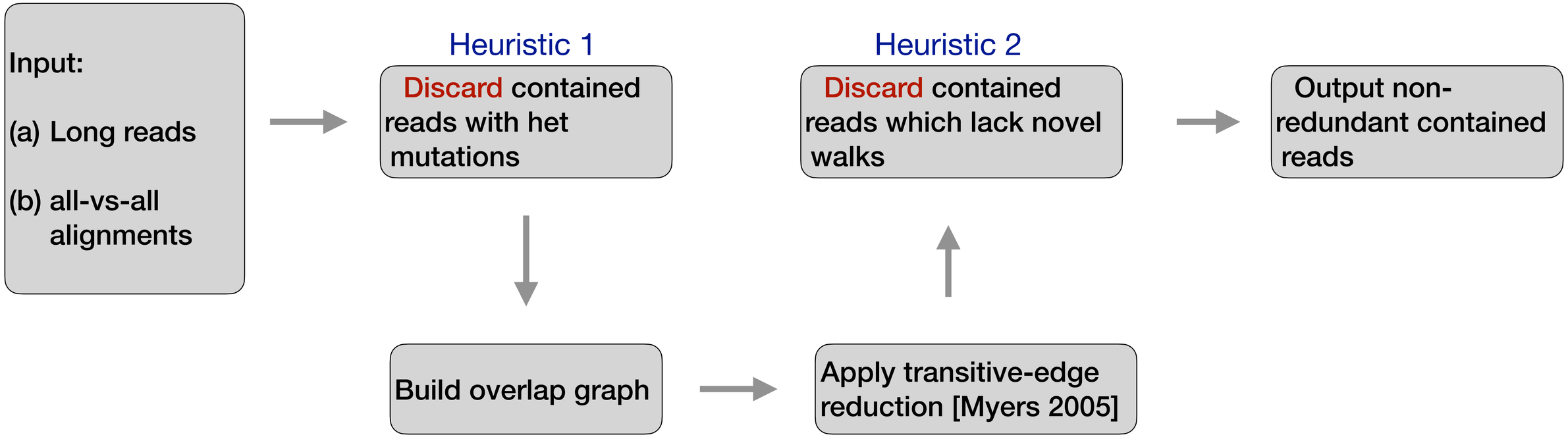
read x CTGCTT
p_1 AACTGCTTACTC
p_2 ACTGCTTGG



ContainX

Prototype implementation in C++

github.com/at-cg/ContainX



Benchmark datasets

- Simulated **error-free** long reads; length distribution matches real data
- Human genomes: CHM13 (haploid), HG002 (diploid)

Data set	Count of reads	N50 length	Max length
HAPLOID-20x-ONT-1	3.7M	40K	570K
HAPLOID-20x-ONT-2	3.7M	40K	540K
HAPLOID-20x-HiFi-1	2.9M	21K	49K
HAPLOID-20x-HiFi-2	2.9M	21K	49K
DIPLOID-30x-ONT-1	5.3M	40K	540K
DIPLOID-30x-ONT-2	5.3M	40K	570K
DIPLOID-30x-HiFi-1	4.2M	21K	49K
DIPLOID-30x-HiFi-2	4.2M	21K	49K

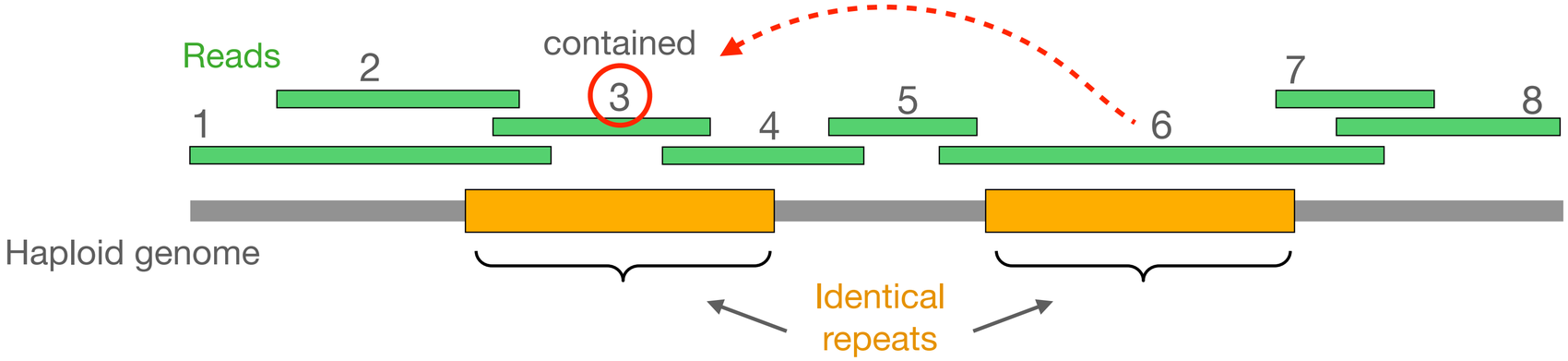
Cheng et al. "Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm" (2021)

Nurk et al. "The complete sequence of a human genome" (2022)

Coverage gaps observed by removing contained reads

- Step 1: Identify contained reads by all-vs-all read alignments
- Step 2: map non-contained reads to genome

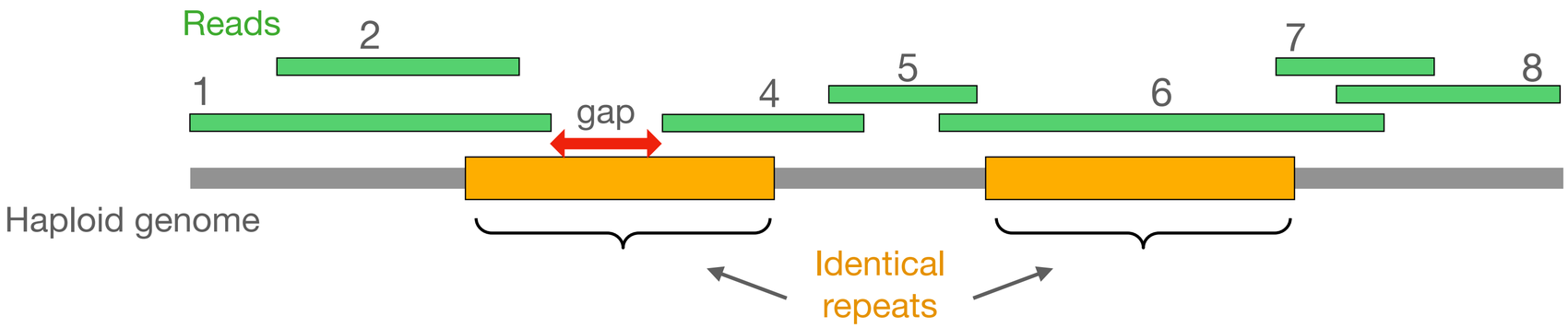
Data	Count of contained reads	Coverage-gaps	
		Count	Maximum length
HAPLOID-20x-ONT-1	3.2M	0	-
HAPLOID-20x-ONT-2	3.2M	0	-
HAPLOID-20x-HiFi-1	1.9M	0	-
HAPLOID-20x-HiFi-2	1.9M	0	-



Coverage gaps observed by removing contained reads

- Step 1: Identify contained reads by all-vs-all read alignments
- Step 2: map non-contained reads to genome

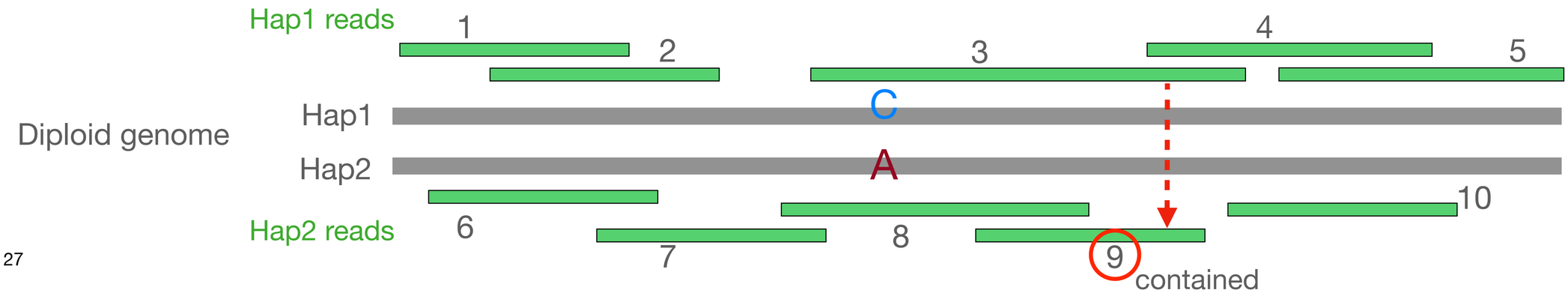
Data	Count of contained reads	Coverage-gaps	
		Count	Maximum length
HAPLOID-20x-ONT-1	3.2M	0	-
HAPLOID-20x-ONT-2	3.2M	0	-
HAPLOID-20x-HiFi-1	1.9M	0	-
HAPLOID-20x-HiFi-2	1.9M	0	-



Coverage gaps observed by removing contained reads

- Step 1: Identify contained reads by all-vs-all read alignments
- Step 2: map non-contained reads to genome

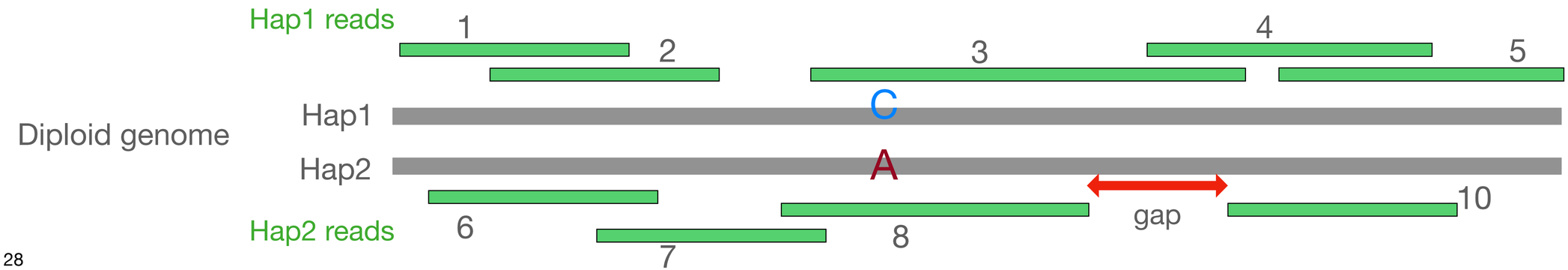
Data	Count of contained reads	Coverage-gaps	
		Count	Maximum length
DIPLOID-30x-ONT-1	4.6M	46	53K
DIPLOID-30x-ONT-2	4.6M	54	101K
DIPLOID-30x-HiFi-1	2.5M	1	2K
DIPLOID-30x-HiFi-2	2.5M	1	0.2K



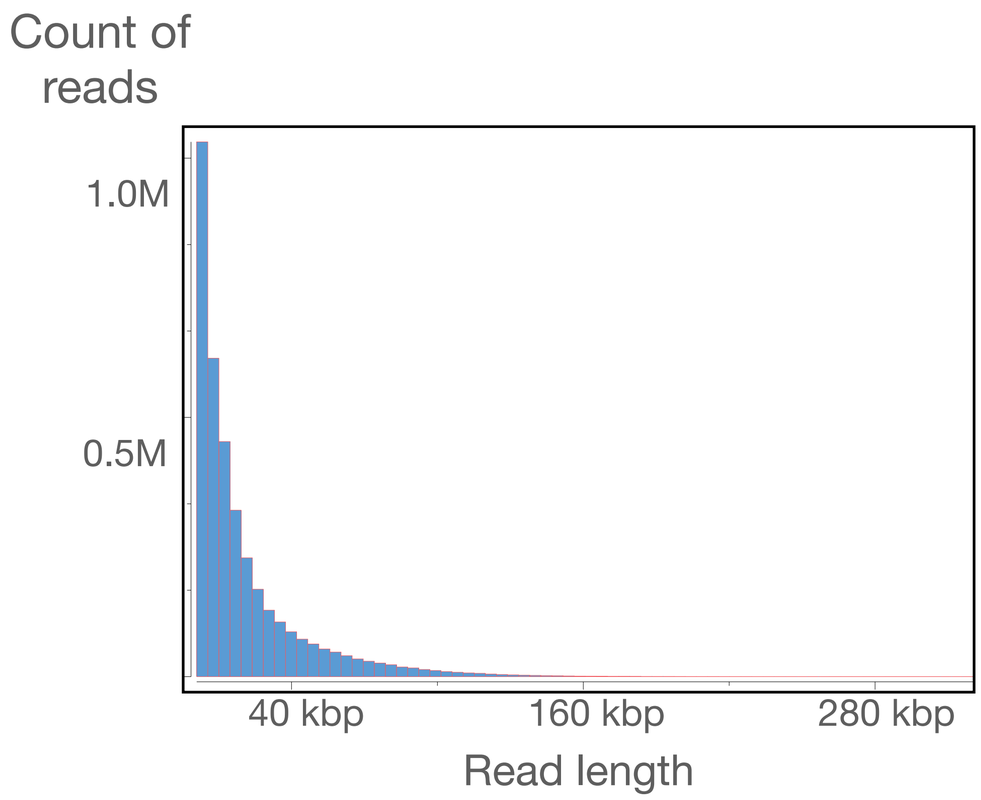
Coverage gaps observed by removing contained reads

- Step 1: Identify contained reads by all-vs-all read alignments
- Step 2: map non-contained reads to genome

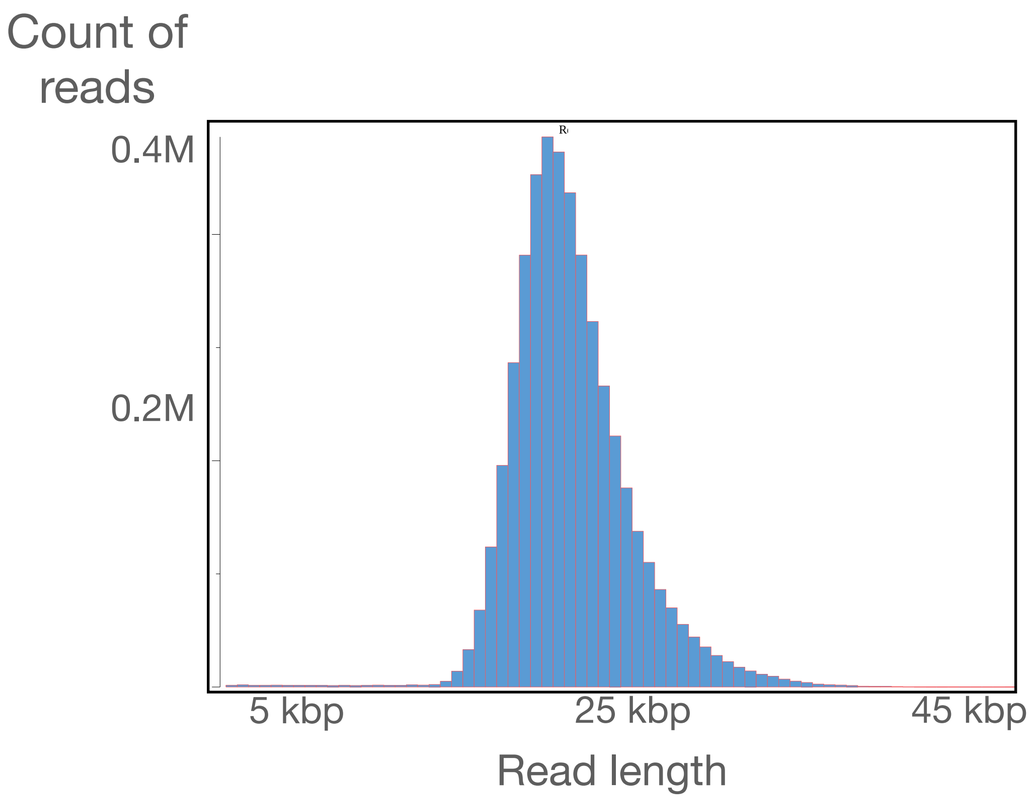
Data	Count of contained reads	Coverage-gaps	
		Count	Maximum length
DIPLOID-30x-ONT-1	4.6M	46	53K
DIPLOID-30x-ONT-2	4.6M	54	101K
DIPLOID-30x-HiFi-1	2.5M	1	2K
DIPLOID-30x-HiFi-2	2.5M	1	0.2K



Read length distributions

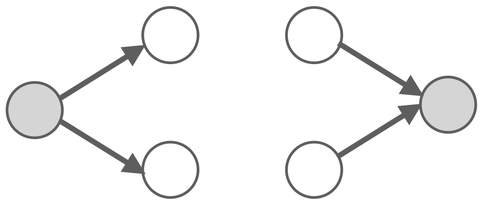


Oxford Nanopore (ONT)



PacBio HiFi

Evaluation of proposed heuristics



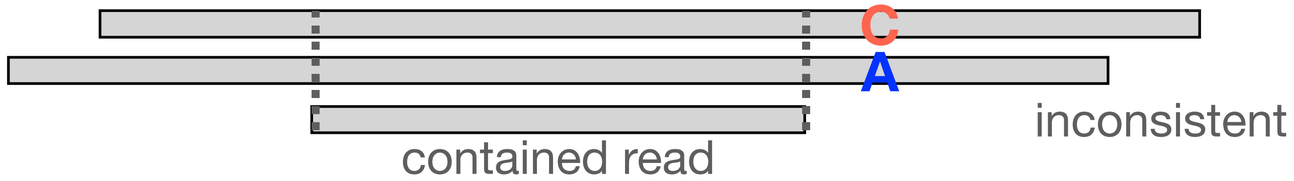
Data	Method	Count of contained reads retained	Count of junction vertices	Gaps introduced in the genome
DIPLOID-30x-ONT-1				
	ContainX			
DIPLOID-30x-HiFi-1				
	ContainX			

LOWER IS BETTER

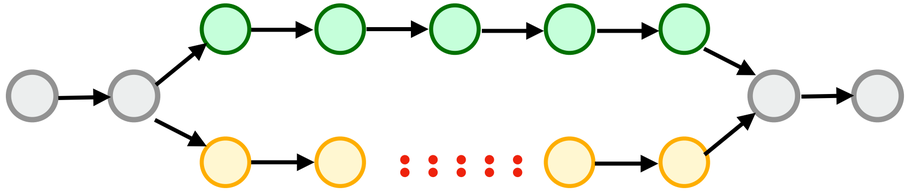
LOWER IS BETTER

Existing solutions besides ContainX

- Other solutions to identify “useful” contained reads
- **[Hui et al. ISIT 2016]**
  - Contained read is removed if it has an inconsistent pair of parent reads



- **Hifiasm [Cheng et al. 2021]**
  - Recovers contained reads which join a broken haplotype walk

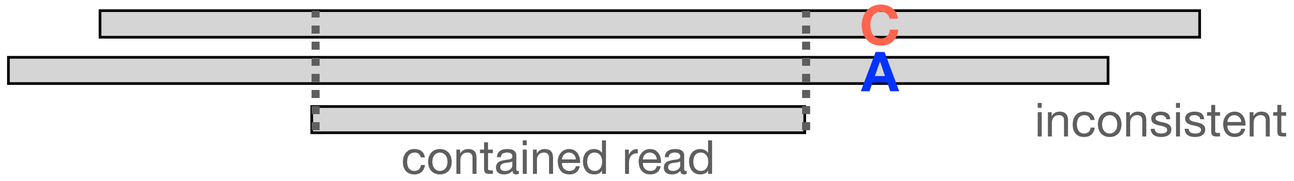


Existing solutions besides ContainX

- Other solutions to identify “useful” contained reads

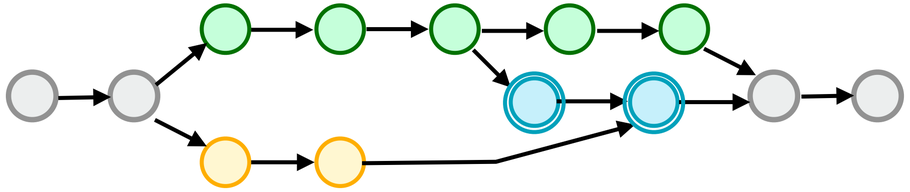
- **[Hui et al. ISIT 2016]**

  - Contained read is removed if it has an inconsistent pair of parent reads



- **Hifiasm Hybrid: PacBio HiFi + ultra-long ONT** **[Cheng et al. 2023]**

  - Identifies useful contained reads by aligning ultra-long nanopore reads to graph



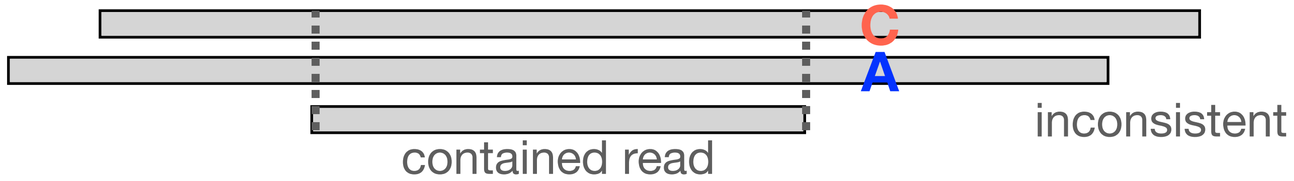
vertices from contained reads

Existing solutions besides ContainX

- Other solutions to identify “useful” contained reads

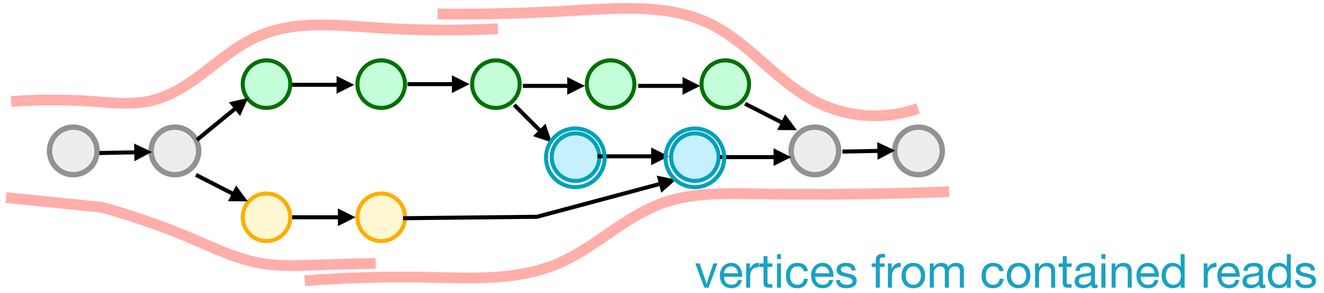
- **[Hui et al. ISIT 2016]**

  - Contained read is removed if it has an inconsistent pair of parent reads



- **Hifiasm Hybrid: PacBio HiFi + ultra-long ONT** **[Cheng et al. 2023]**

  - Identifies useful contained reads by aligning ultra-long nanopore reads to graph

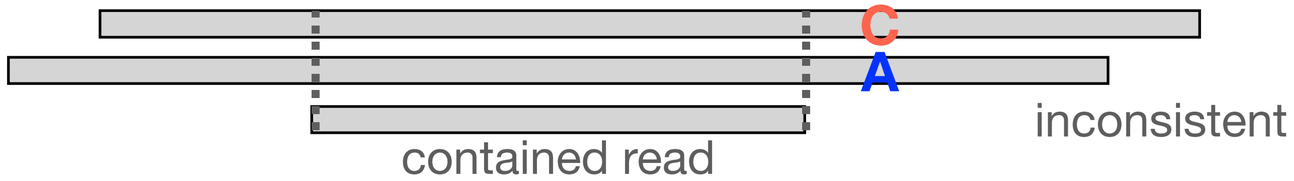


Existing solutions besides ContainX

- Other solutions to identify “useful” contained reads

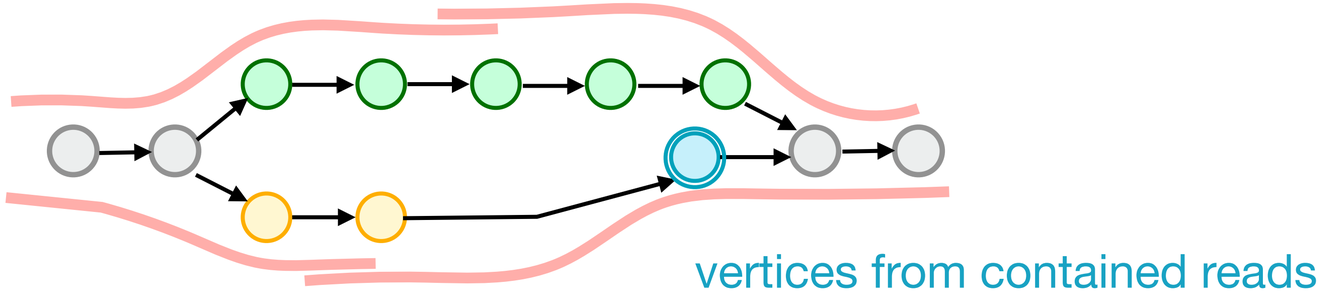
- **[Hui et al. ISIT 2016]**

  - Contained read is removed if it has an inconsistent pair of parent reads

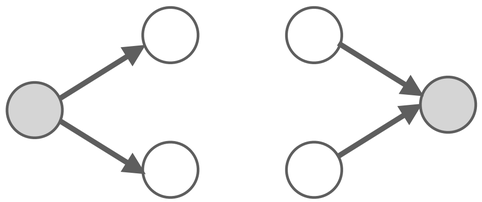


- **Hifiasm Hybrid: PacBio HiFi + ultra-long ONT** **[Cheng et al. 2023]**

  - Identifies useful contained reads by aligning ultra-long nanopore reads to graph



Evaluation of proposed heuristics

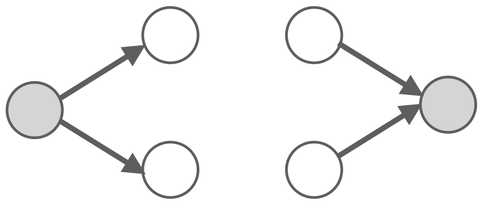


Data	Method	Count of contained reads retained	Count of junction vertices	Gaps introduced in the genome
DIPLOID-30x-ONT-1	Retain all			
	Hui-2016			
	ContainX			
	Remove all			
DIPLOID-30x-HiFi-1	Retain all			
	Hui-2016			
	ContainX			
	Remove all			

LOWER IS BETTER

LOWER IS BETTER

Evaluation of proposed heuristics

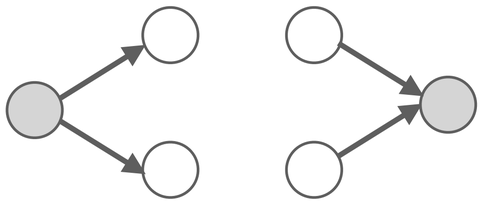


Data	Method	Count of contained reads retained	Count of junction vertices	Gaps introduced in the genome
DIPLOID-30x-ONT-1	Retain all	2.8M	2.5M	0
	Hui-2016			
	ContainX			
	Remove all	0	38.9K	46
DIPLOID-30x-HiFi-1	Retain all	2.5M	3.4M	0
	Hui-2016			
	ContainX			
	Remove all	0	158.4K	1

LOWER IS BETTER

LOWER IS BETTER

Evaluation of proposed heuristics

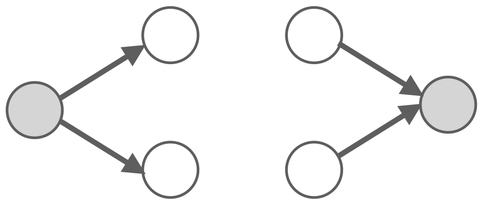


Data	Method	Count of contained reads retained	Count of junction vertices	Gaps introduced in the genome
DIPLOID-30x-ONT-1	Retain all	2.8M	2.5M	0
	Hui-2016	2.5M	2.3M	0
	ContainX	28.5K	53.9K	2
	Remove all	0	38.9K	46
DIPLOID-30x-HiFi-1	Retain all	2.5M	3.4M	0
	Hui-2016	2.5M	3.3M	0
	ContainX	39.8K	184.1K	0
	Remove all	0	158.4K	1

LOWER IS BETTER

LOWER IS BETTER

Evaluation of proposed heuristics



Data	Method	Count of contained reads retained	Count of junction vertices	Gaps introduced in the genome
DIPLOID-30x-ONT-1	Retain all	2.8M	2.5M	0
	Hui-2016	2.5M	2.3M	0
	ContainX	28.5K	53.9K	2
	Hifiasm	4.0K	1.7K	33
	Remove all	0	38.9K	46
DIPLOID-30x-HiFi-1	Retain all	2.5M	3.4M	0
	Hui-2016	2.5M	3.3M	0
	ContainX	39.8K	184.1K	0
	Hifiasm	164	36.9K	0
	Remove all	0	158.4K	1
			LOWER IS BETTER	LOWER IS BETTER

* Hifiasm is an end-to-end genome assembler, uses multiple graph sparsification heuristics

Conclusions

- Provably-good graph models will be useful for reliable and accurate human genome reconstruction
- String graph model is used commonly, but

  it violates the ‘safety’ guarantee, both in theory and practice.
- Optimal sparsification of overlap graphs remains unsolved. We proposed *safe* rules and promising heuristics.

chirag@iisc.ac.in
github.com/at-cg/ContainX

Link to publication:
https://doi.org/10.1093/bioinformatics/btad124

Acknowledgement

Mehak Bindra (Project staff), Sudhanva Shyam Kamath (Ph.D. student)

Valuable feedback from:

- Prof. Sunil Chandran, Prof. Debnath Pal [IISc]
- Haowen Zhang [Georgia Tech]
- Mile Sikic, Robert Vaser [Genome Institute of Singapore]
- Brian Walenz, Sergey Nurk [NHGRI]
- Haoyu Cheng [Dana Farber Cancer Institute]

IISc

विज्ञान एवं प्रौद्योगिकी विभाग
DEPARTMENT OF
SCIENCE & TECHNOLOGY

NERSC