A little tour of assembly methods

Antoine Limasset & Camille Marchet
CRIStAL, Université de Lille, CNRS, France
Evomics 2024 – Český Krumlov

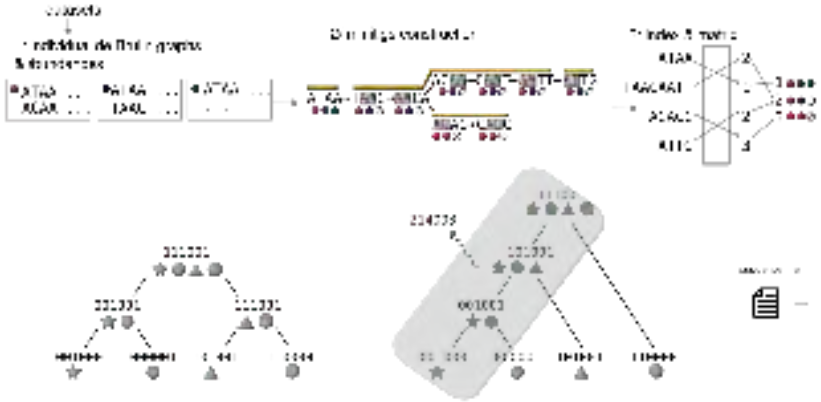
@Camillemrcht camille.marchet@univ-lille.fr
@BQPMalfoy antoine.limasset@univ-lille.fr

- Camille Marchet

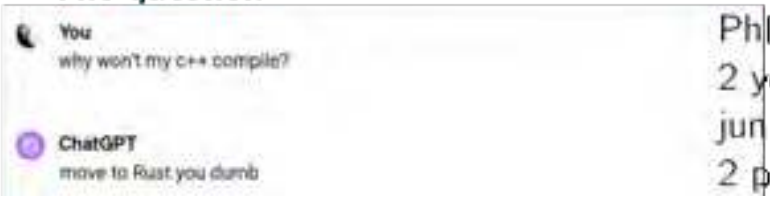
The scientist

I was here in 1921 like Jack in The Shining

The "species"

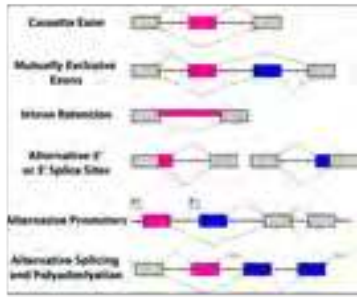


The question



The career

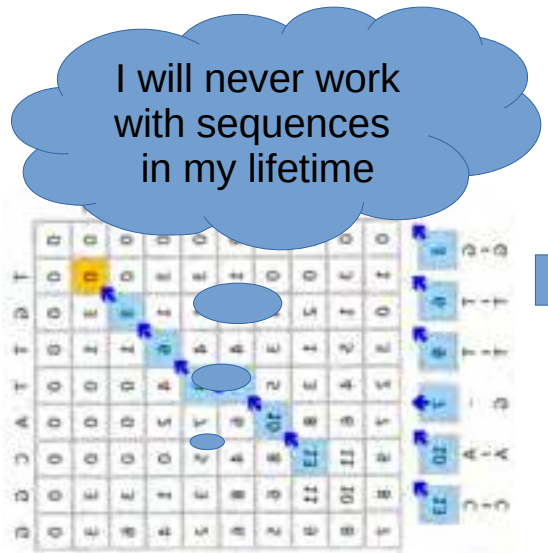
PhD in 2018
2 years postdoc
junior researcher since 2021 (Lille, France)
2 parental leaves



- Antoine Limasset

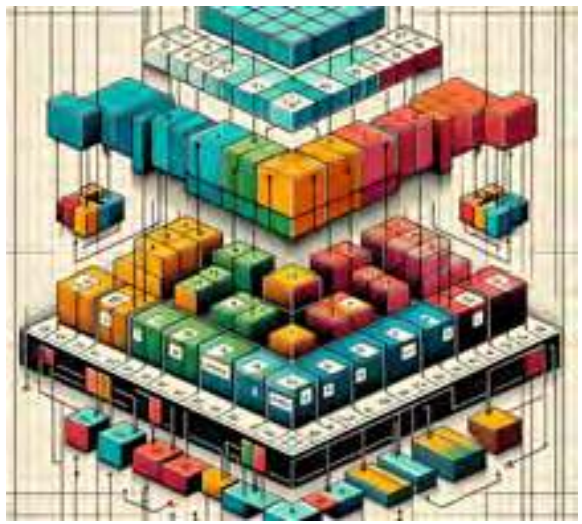
Computer scientist

Undergrad



PhD

Researcher



Leisure (according to ChadGPT)

Trivia:
Never used Blast
Never had biology class
Write one-use script in C++
(Cool) Tool naming is top priority
Mandatory Mahjong club for students
Working on kmers since 2012
Casually drink 1L energy drink a day
Crepes and cake mass producer

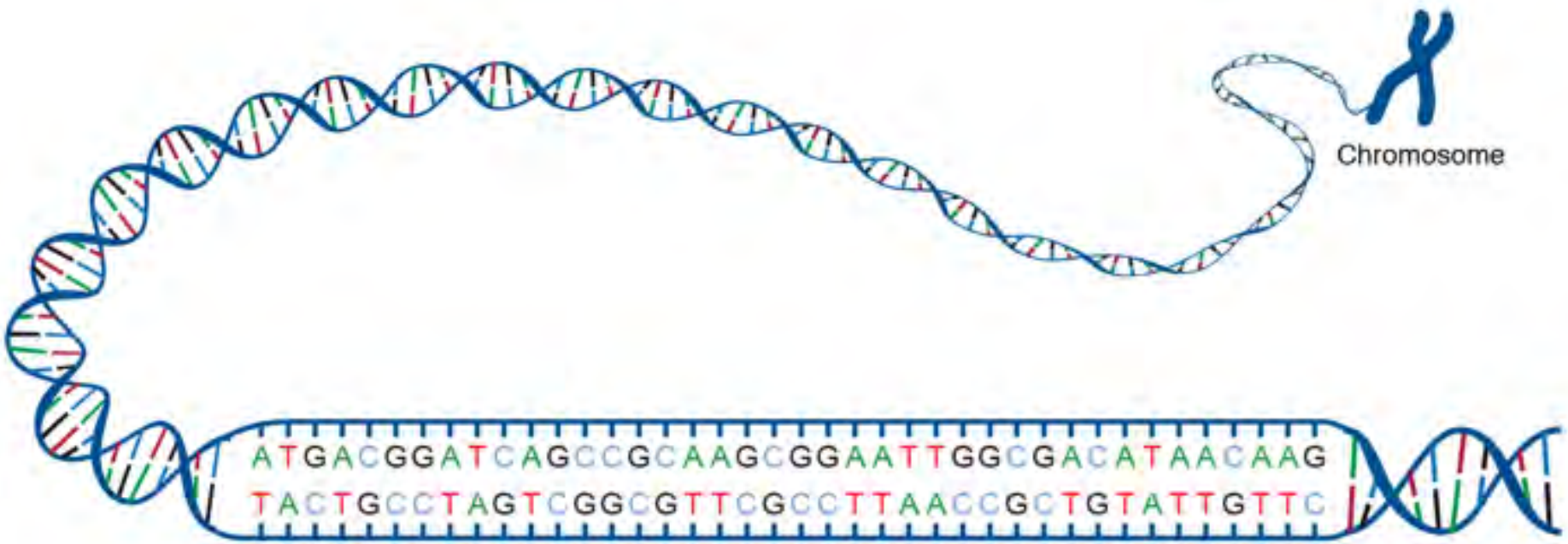
Content of this course

- How to reconstruct a genome with sequencing data?
- What are the main challenges?
- Which solutions have been proposed?

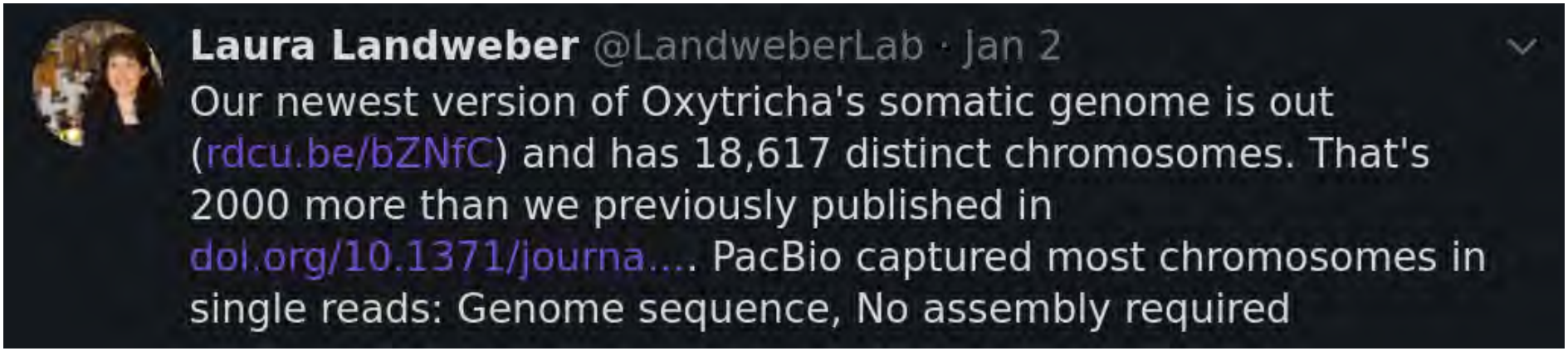
Bingo: find a book that we both love (French title).

genome size: $\sim$ 32 gigabases

- Accessing a genome



- Why do we need assembly?



- Reads are subsequences from the genome

- Reads are **shuffled** subsequences from the genome

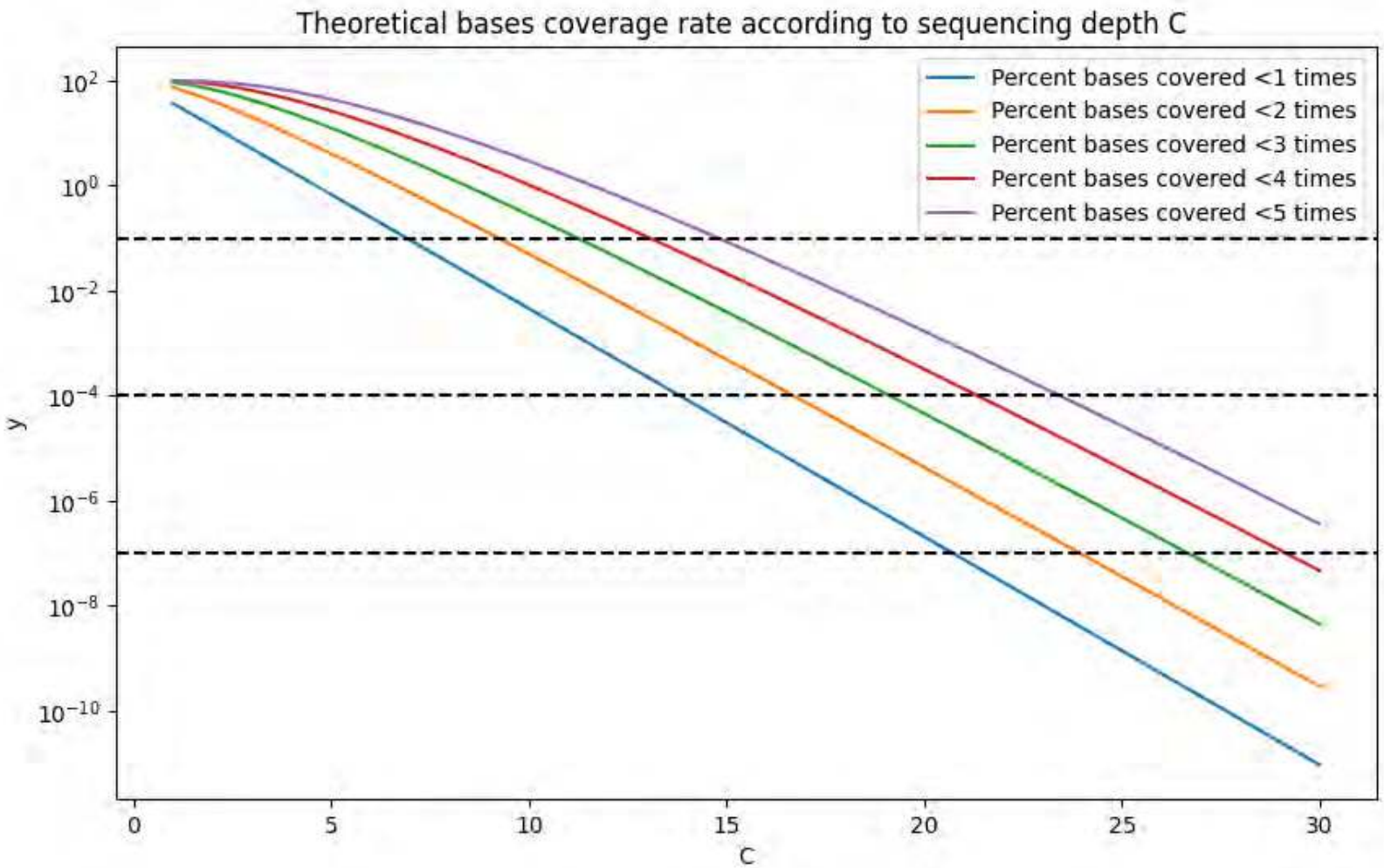
- Genome assembly task

- Genome sequencing: coverage

- Genome sequencing: coverage

- Genome sequencing: coverage

Poisson law



30-60X are often required for assembly projects

- First experiment: *Long, perfect boy*'s genome

Order according to overlaps

Overlaping reads are likely successive part of the genome

Check all reads for overlaps

For a given read, scan all others

Most cases

No overlap

Small overlaps

Can happen "by chance"

Longest overlaps

We are more confident in longer overlaps

- Back to coverage

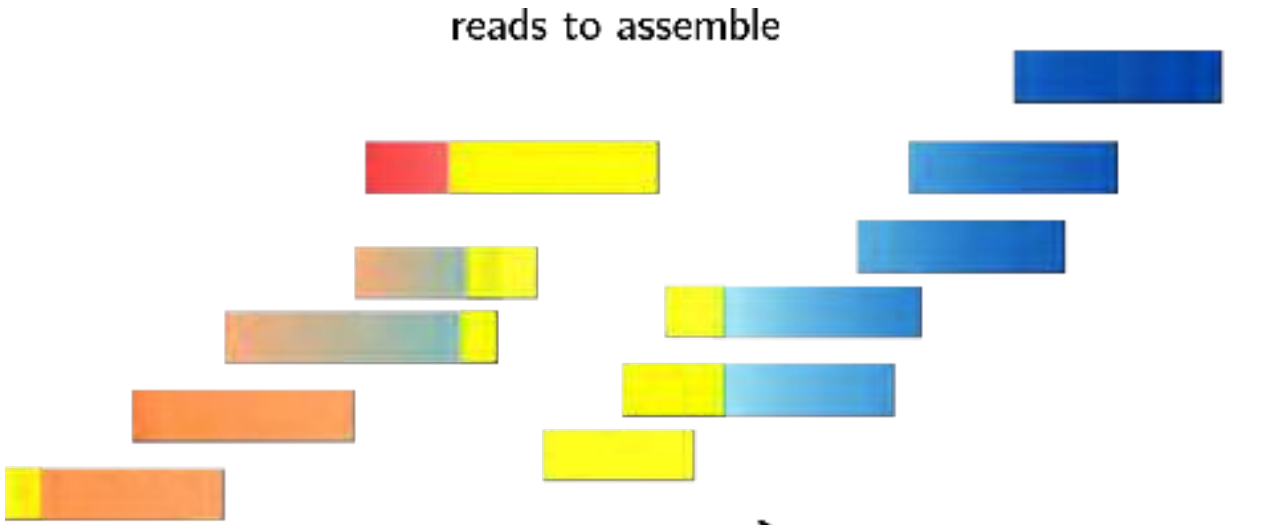
- Higher coverage, longer overlaps

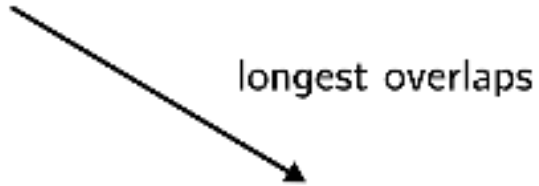
- Assemble according to overlaps

- How to compute the overlaps? Alignment?

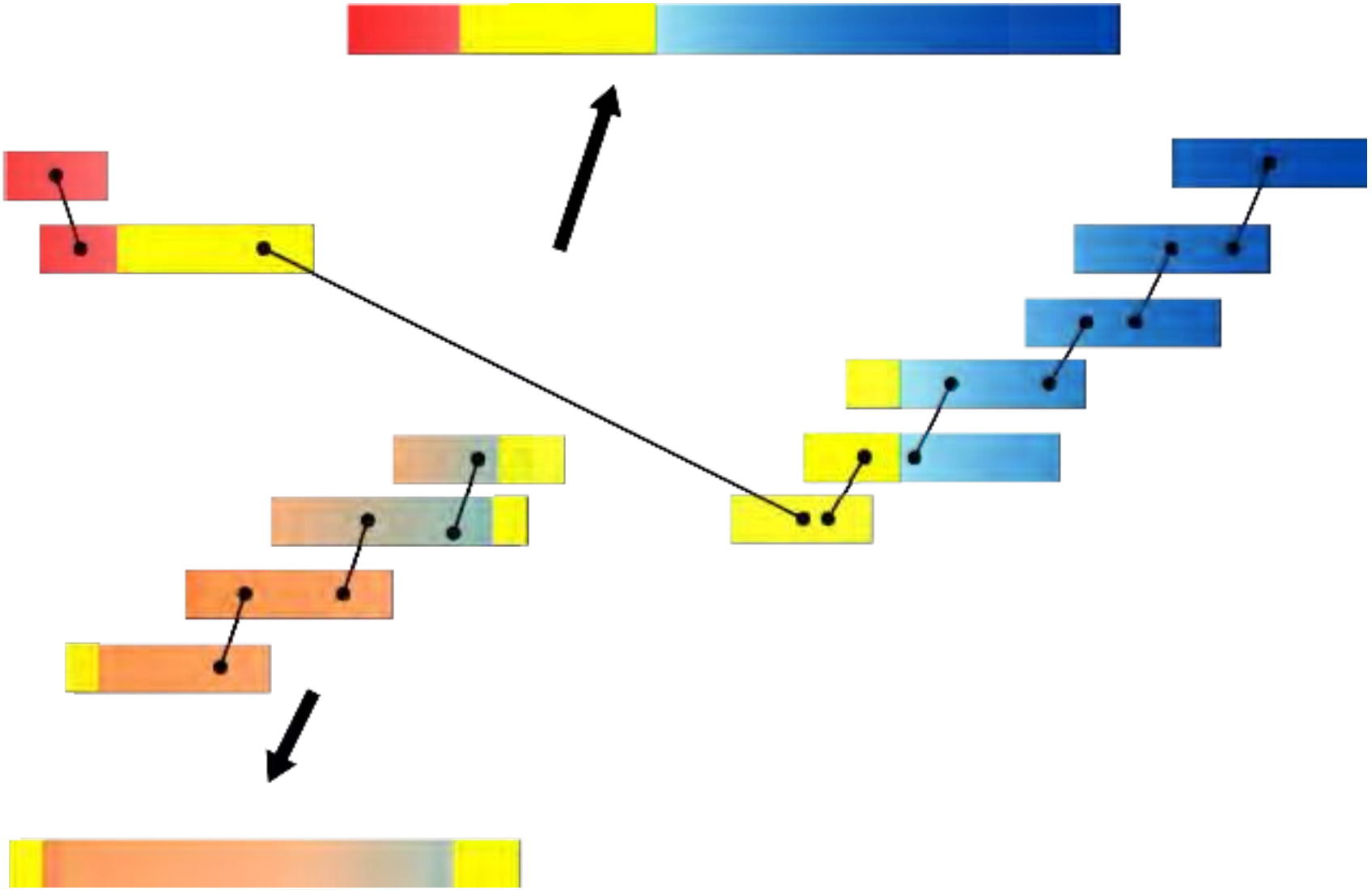
- How to compute the overlaps? Quick exact match!

Something weird happened

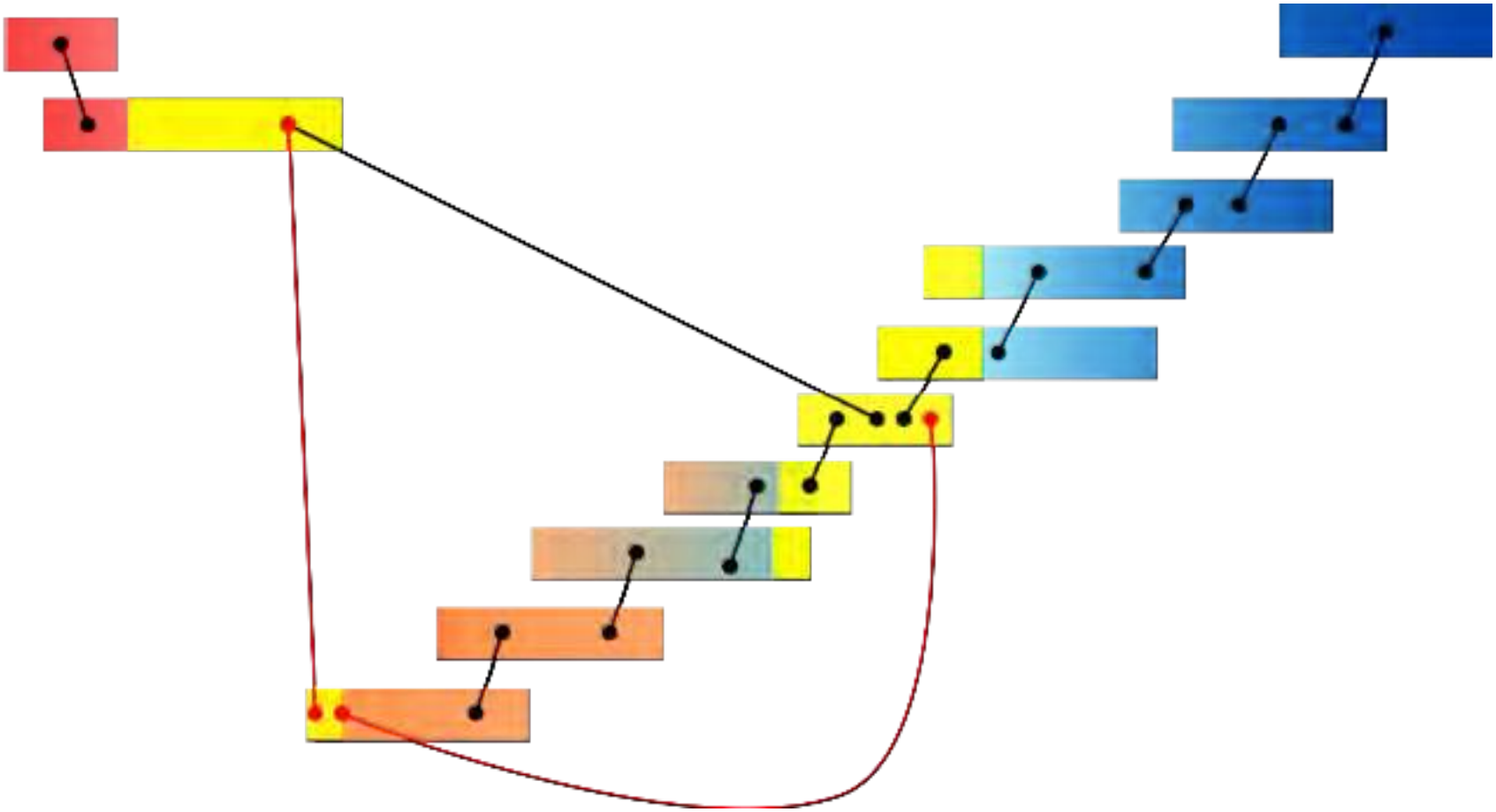




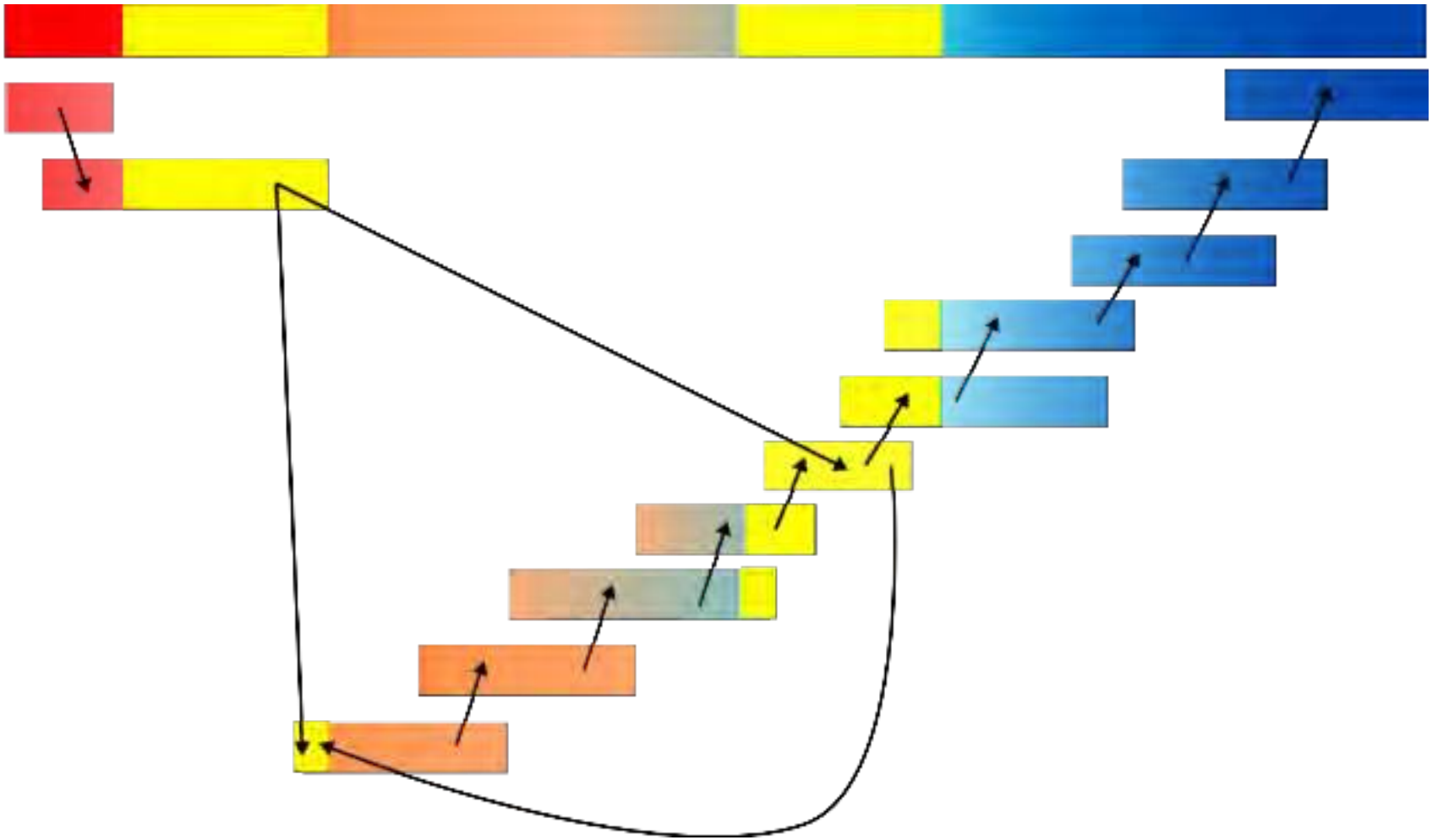
- All longest overlaps



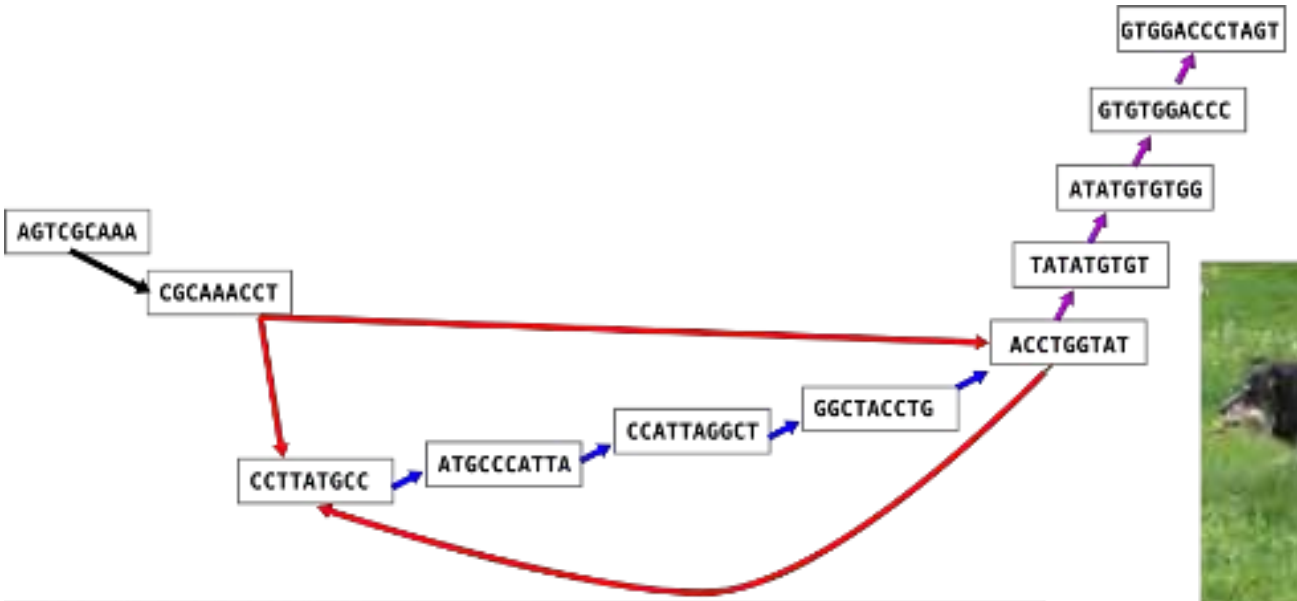
- Take into account other overlaps?



- Look, a graph!



Unsafe paths in an overlap graph



spurious assembly result

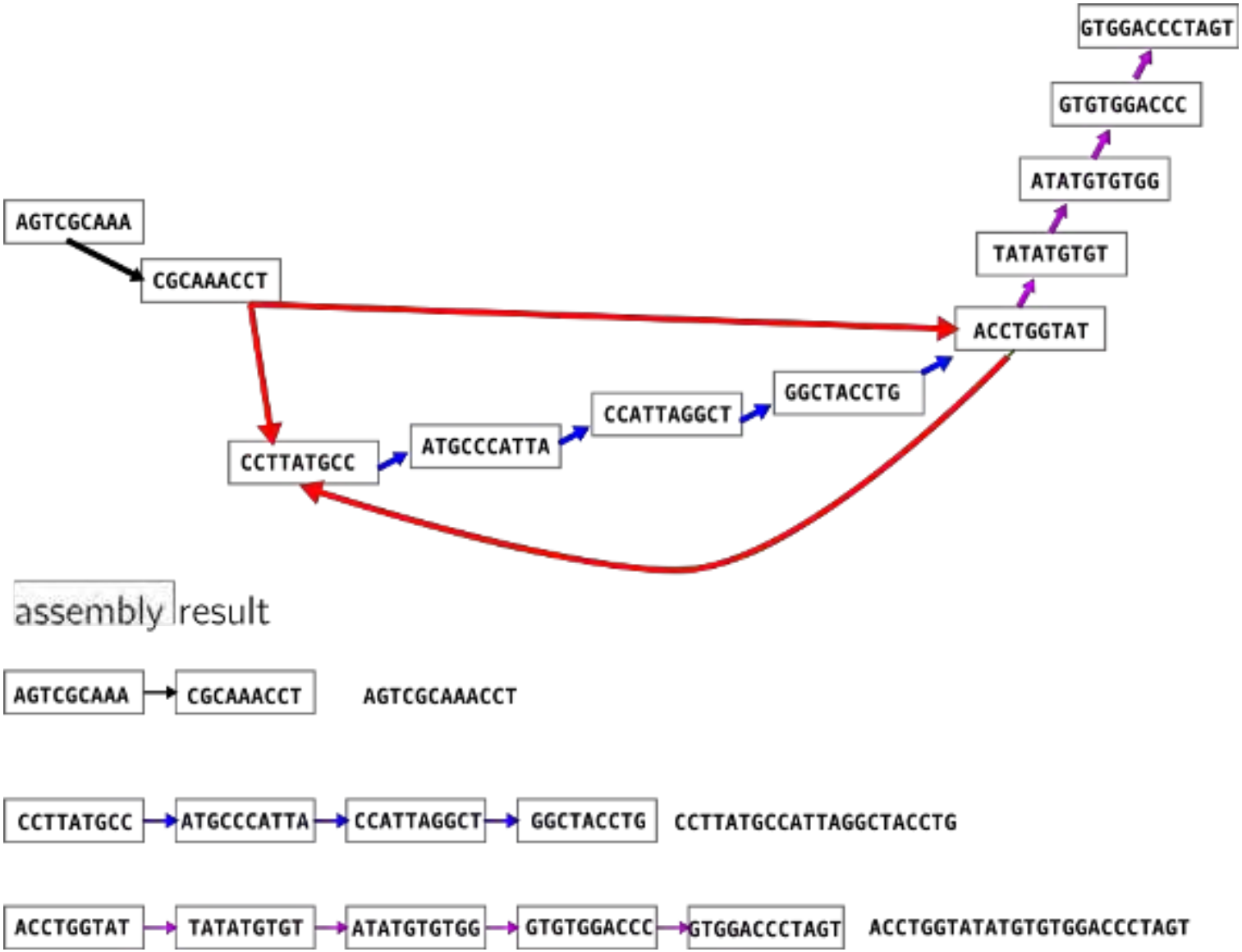
AGTCGCAAA → CGCAAACCT → ACCTGGTAT → TATATGTGT → ATATGTGTGG → GTGTGGACCC → GTGGACCCTAGT

AGTCGCAAACCTGGTATATGTGTGGACCCTAGT

CCTTATGCC → ATGCCCATTA → CCATTAGGCT → GGCTACCTG

CCTTATGCCATTAGGCTACCTG

Safe paths in an overlap graph



- Multiple repeats

- First solution

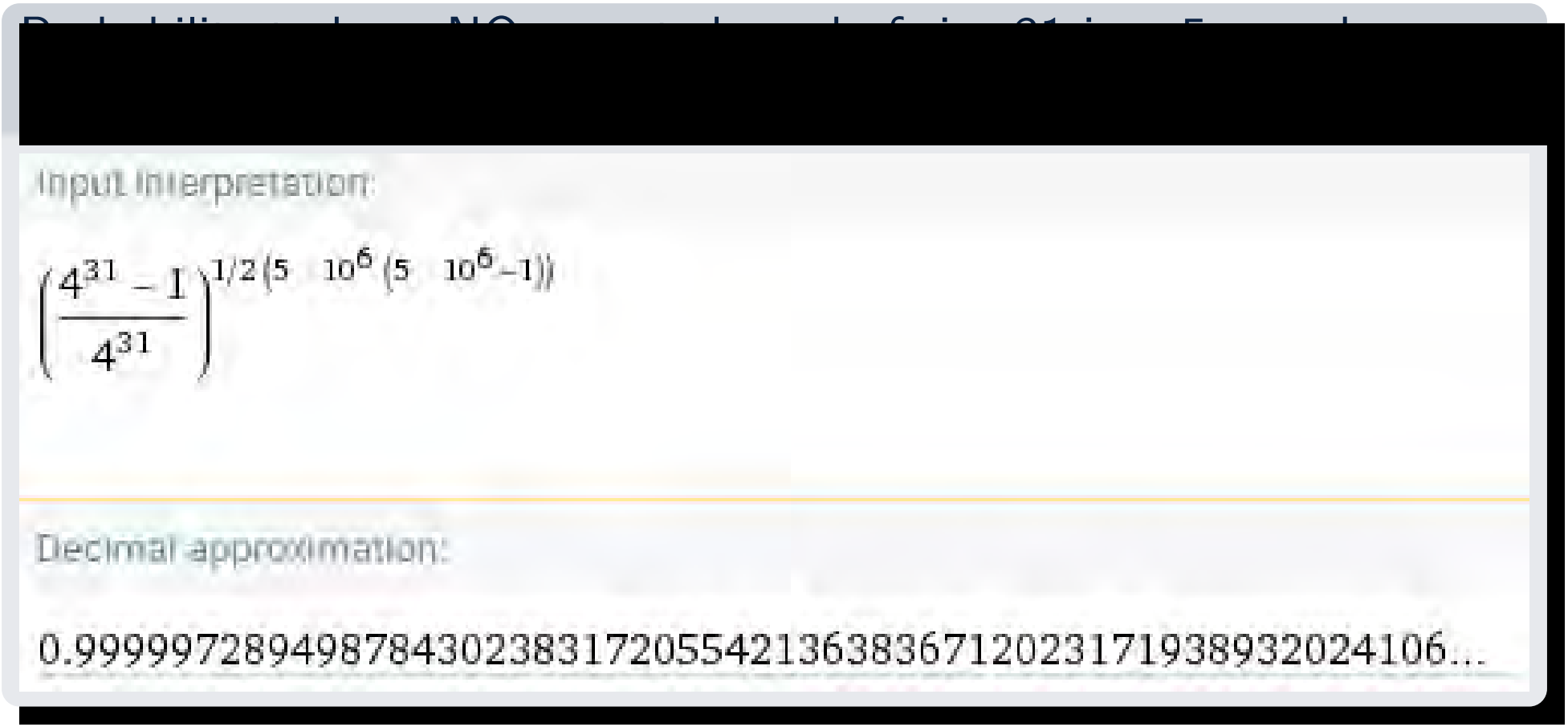
- Second solution

- Parsimonious solution: do not assemble

Repeats lead to the fragmentation of the assembly

Genomes pieces that make **con**sensus across the differents solution are called **Con**tigs

- Do we expect many repeats?



From `en.wikipedia.org/wiki/Birthday_problem`

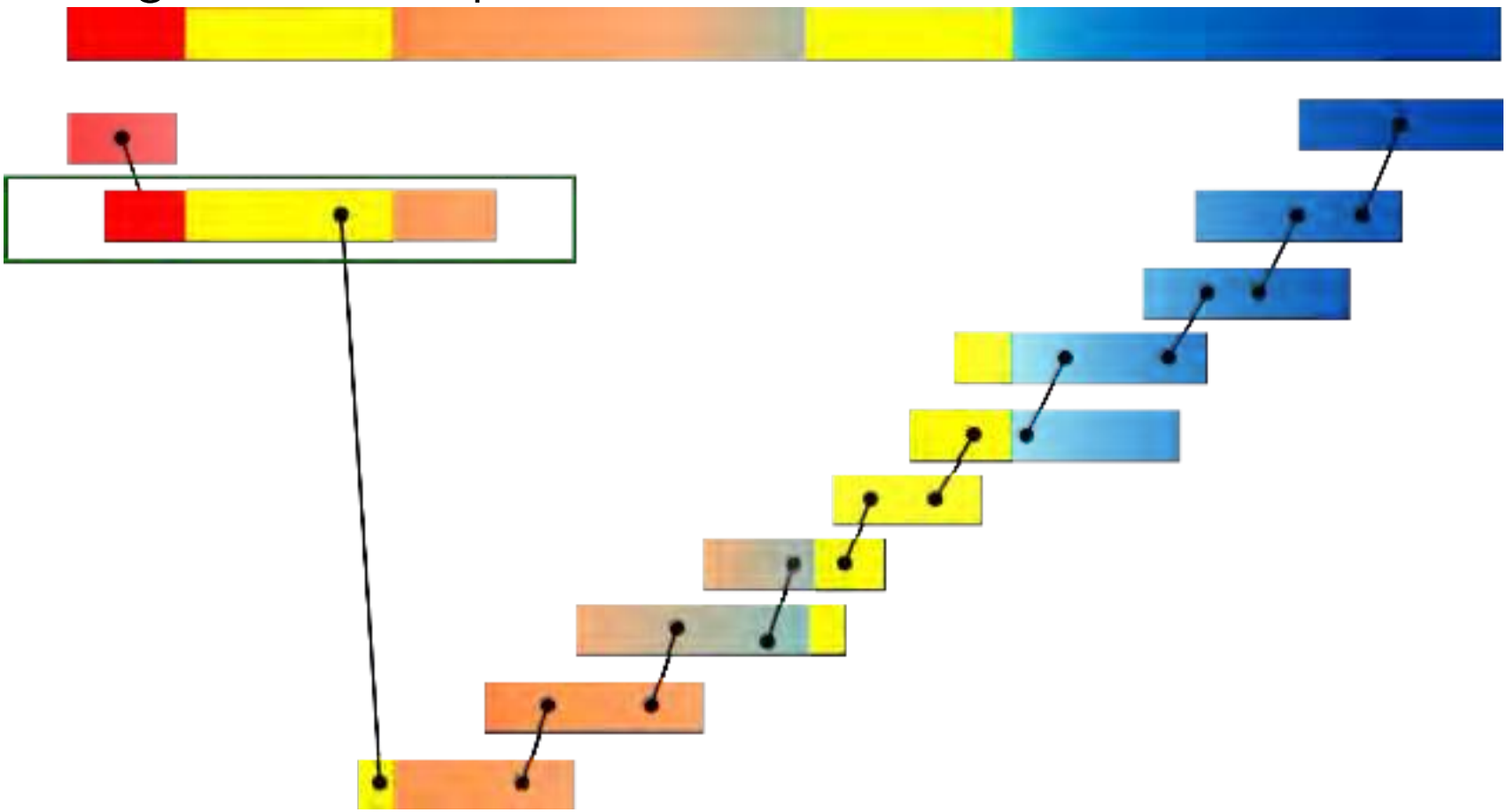
- The burden of assembly: genomic repeats

- 15: 44,994
- 21: 1,169
- 31: 559
- 41: 323
- 51: 225
- 61: 192

Genomic repeats are NOT random events

With longer reads

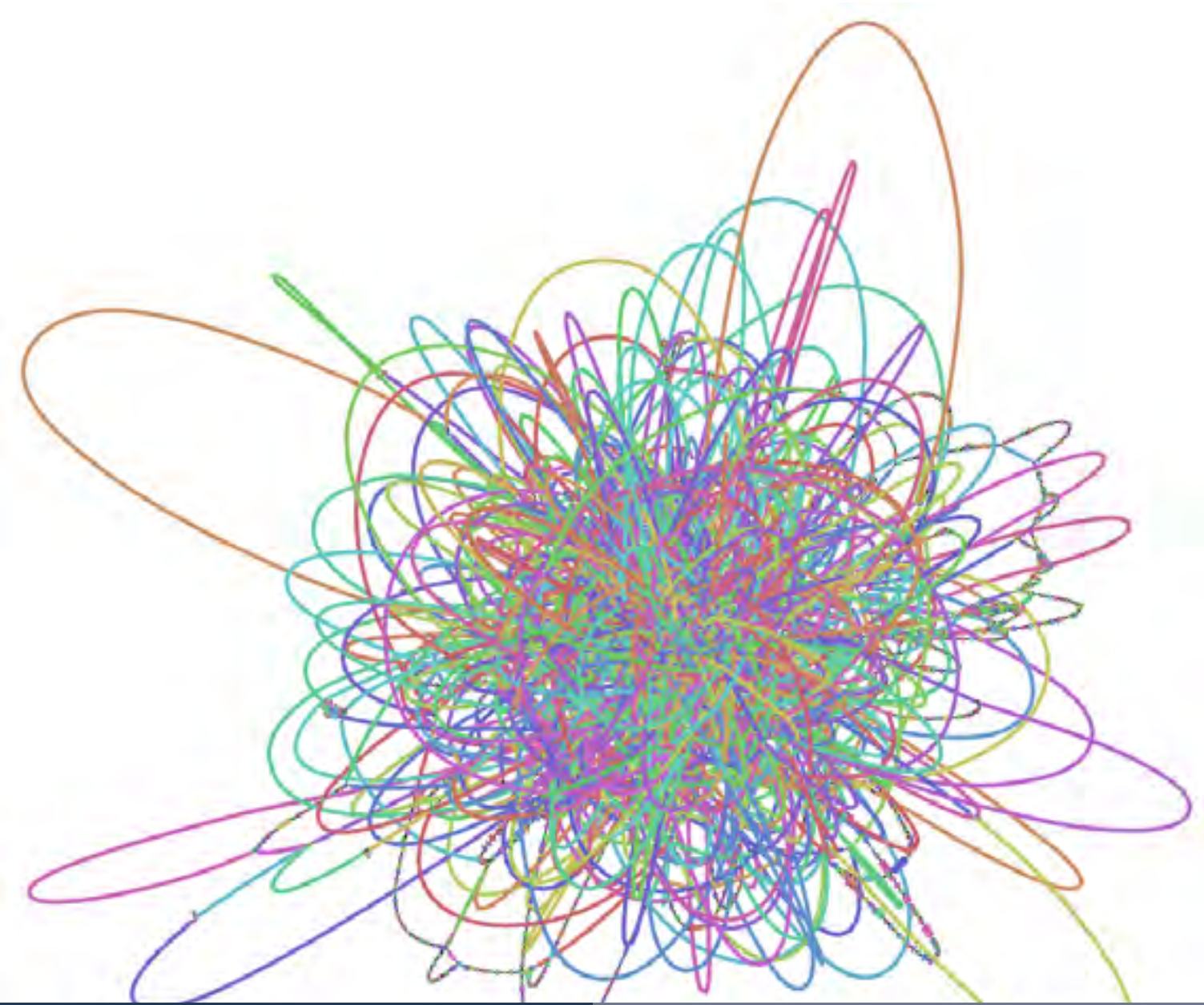
Reads longer than the repeat "solve" it



The graph becomes trivial to go traverse

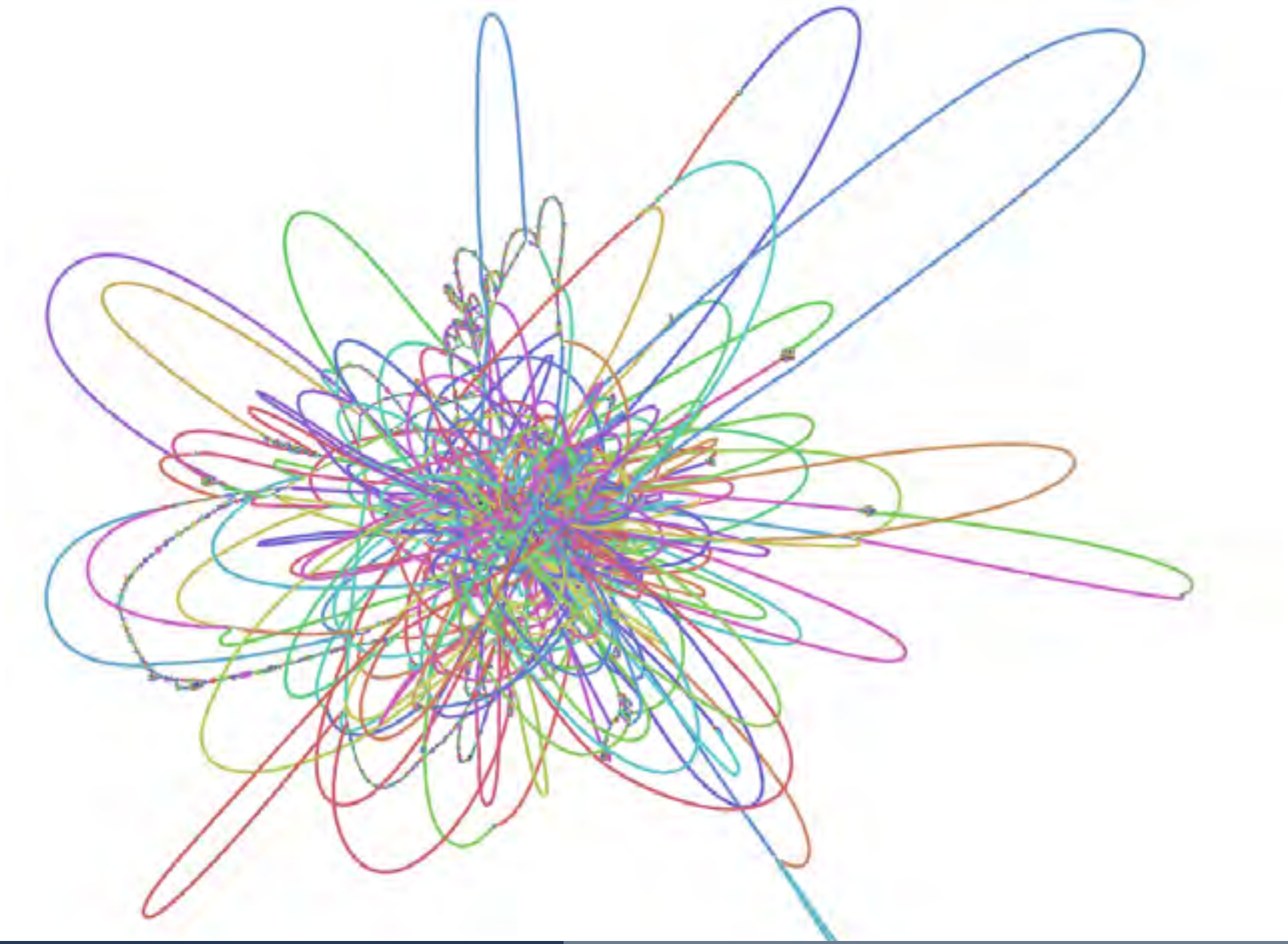
- Read length matters

Read size=21



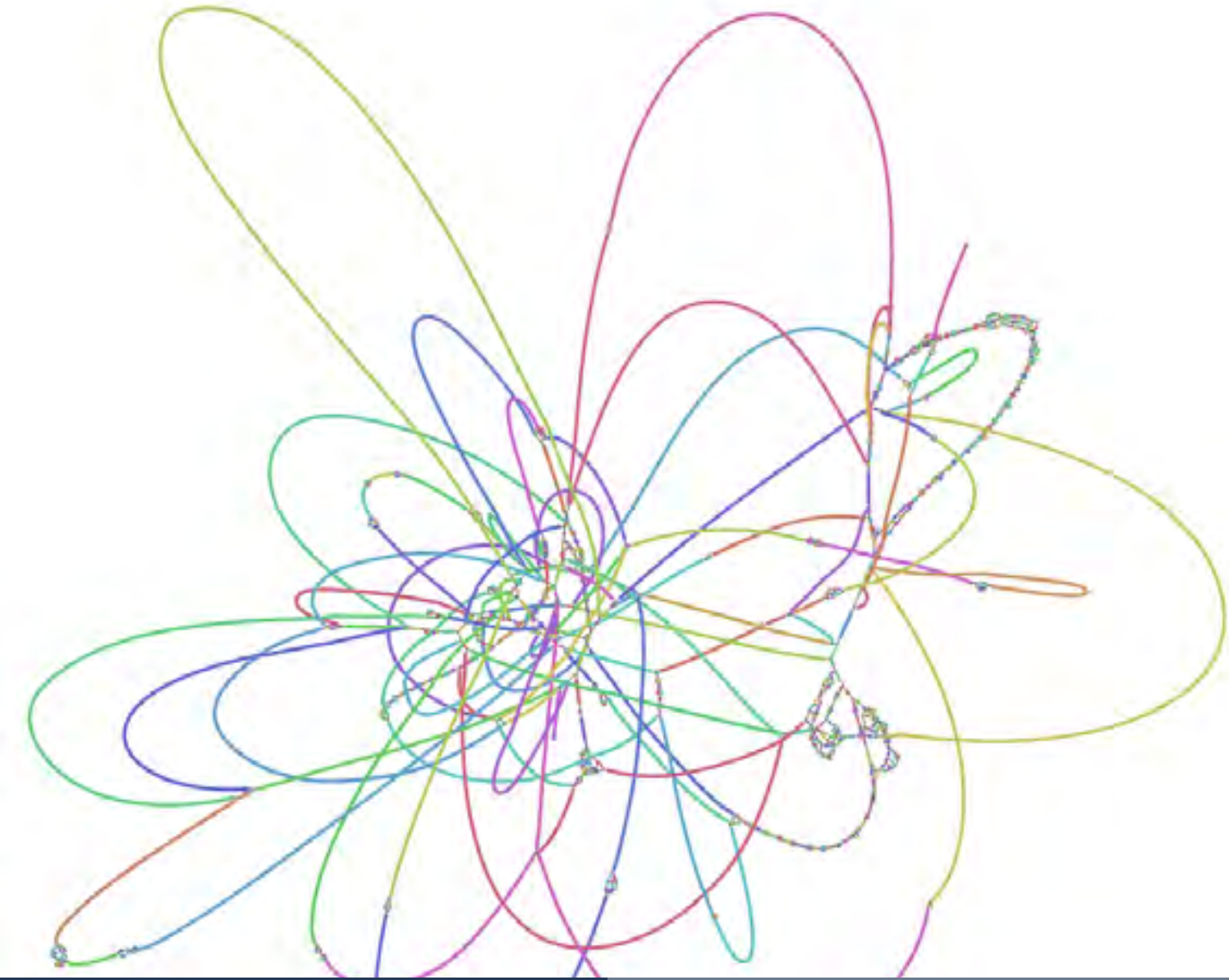
- Read length matters

Read size=31



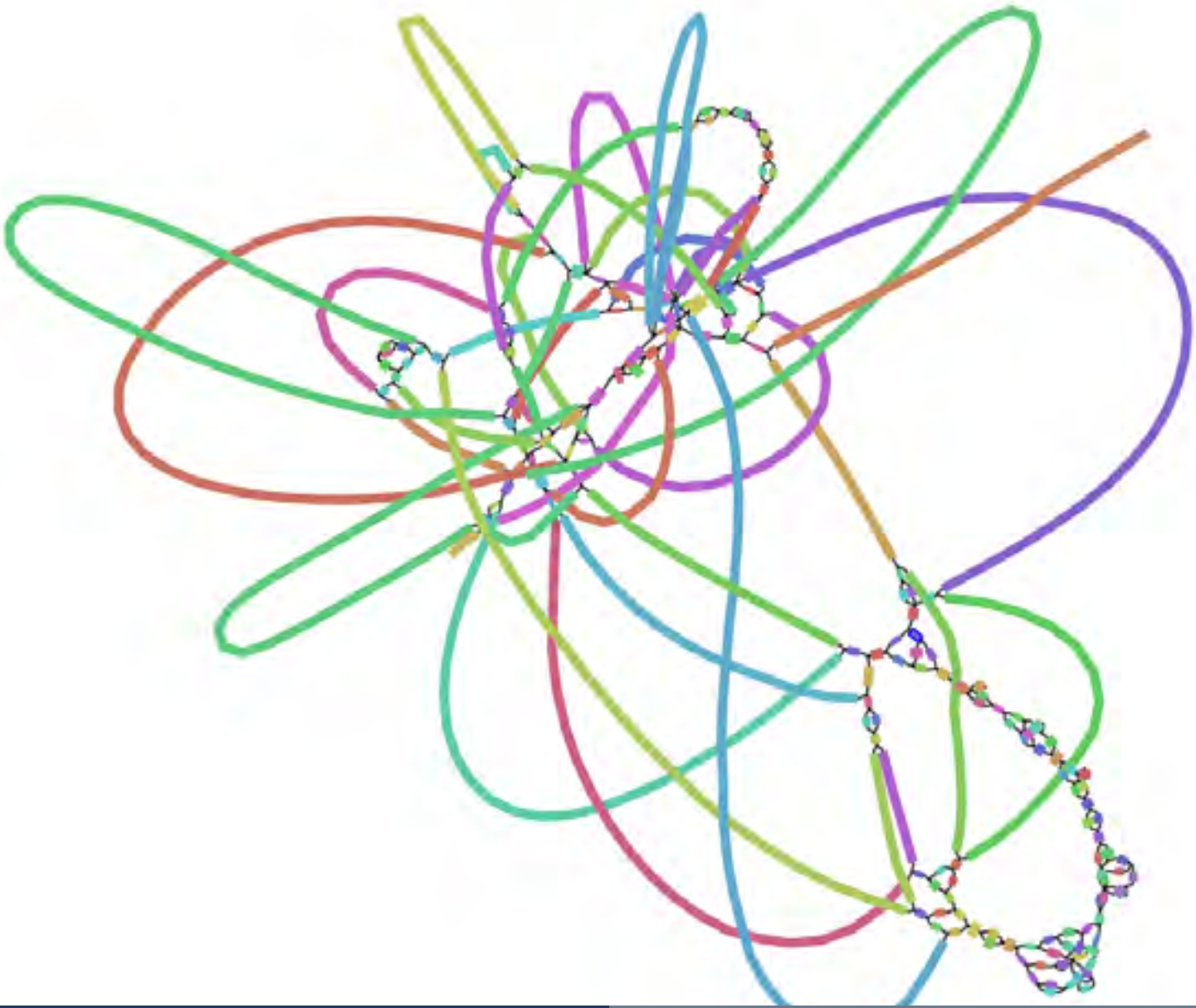
- Read length matters

Read size=63



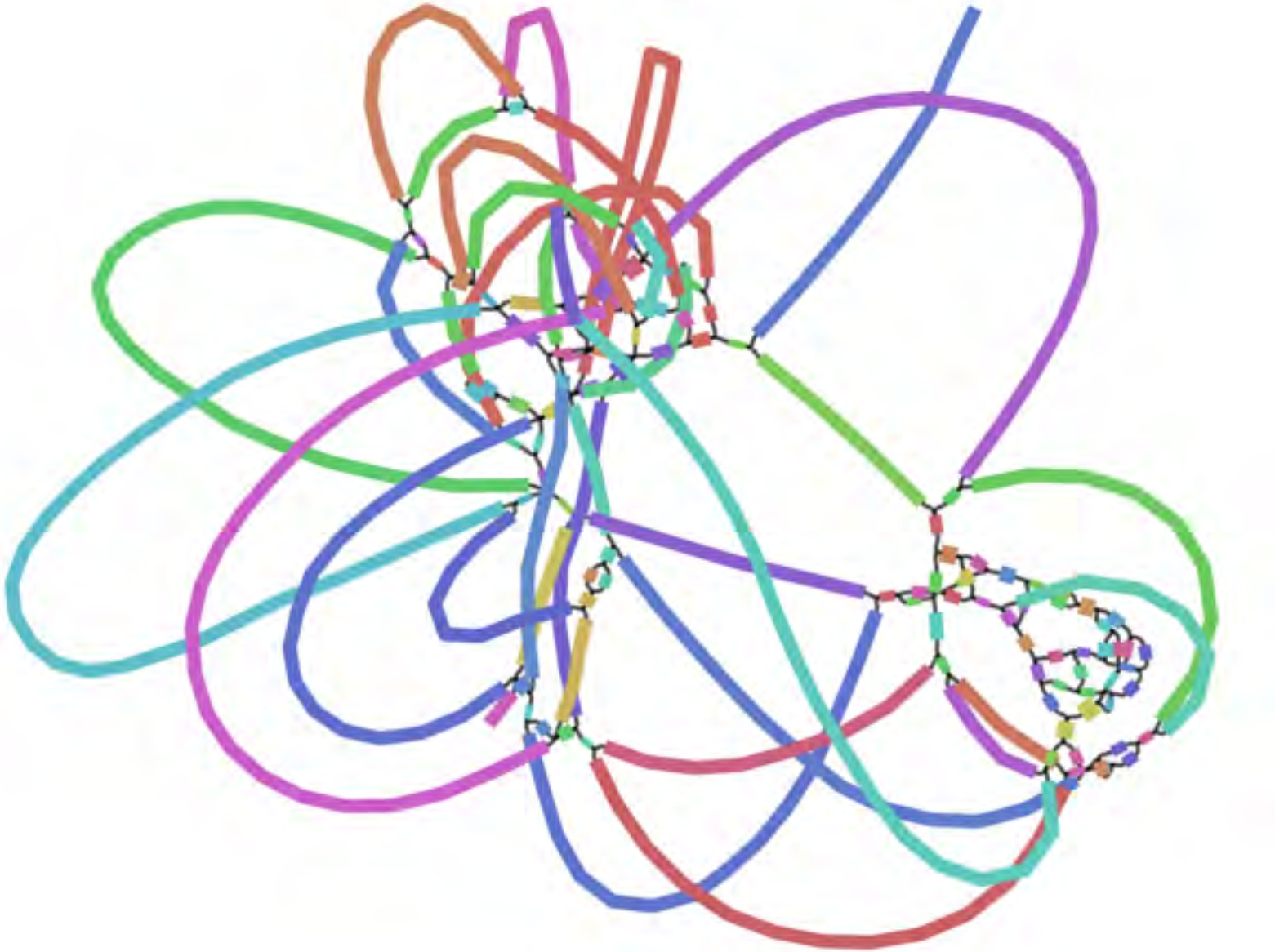
- Read length matters

Read size=255



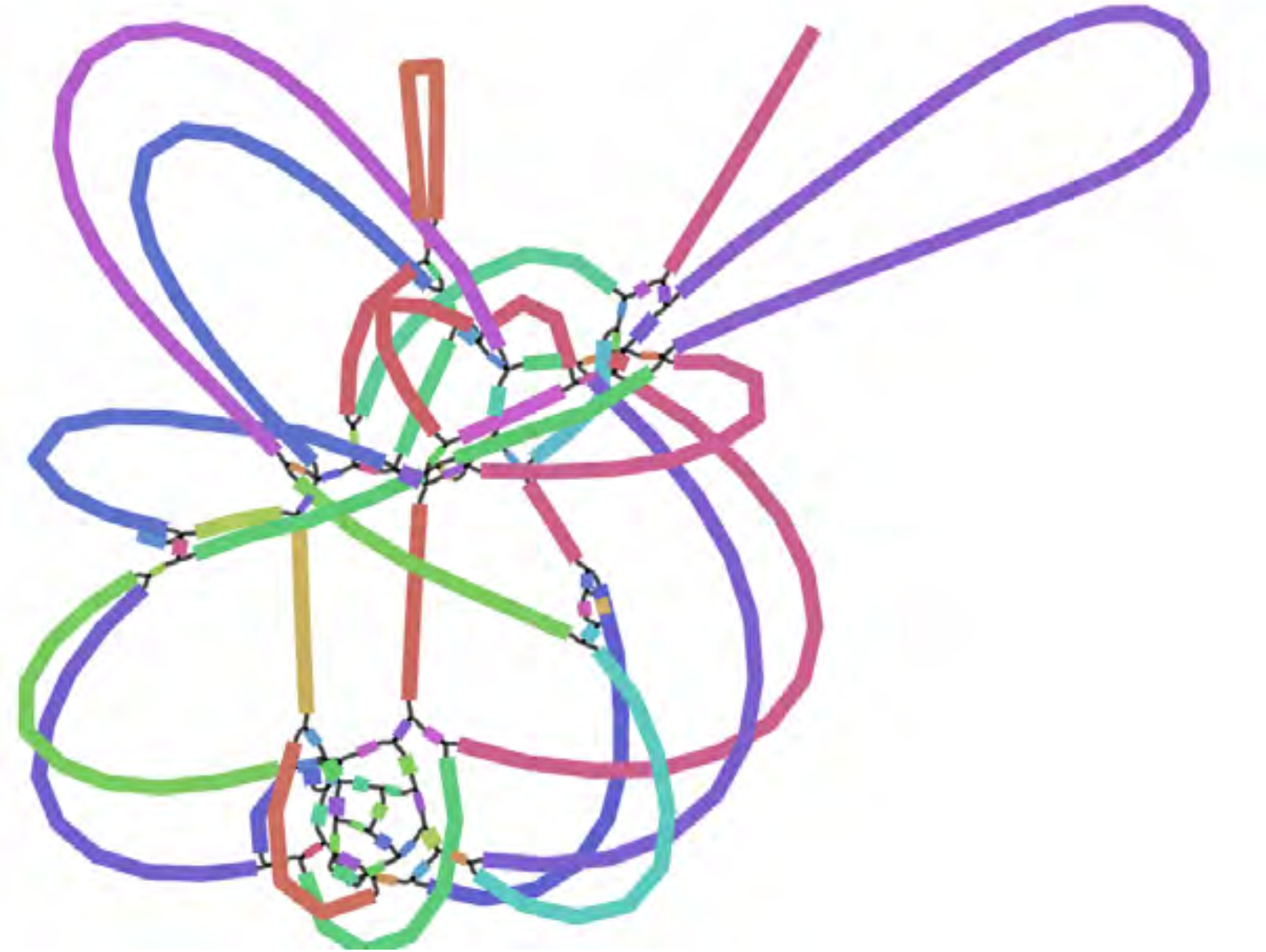
- Read length matters

Read size=500



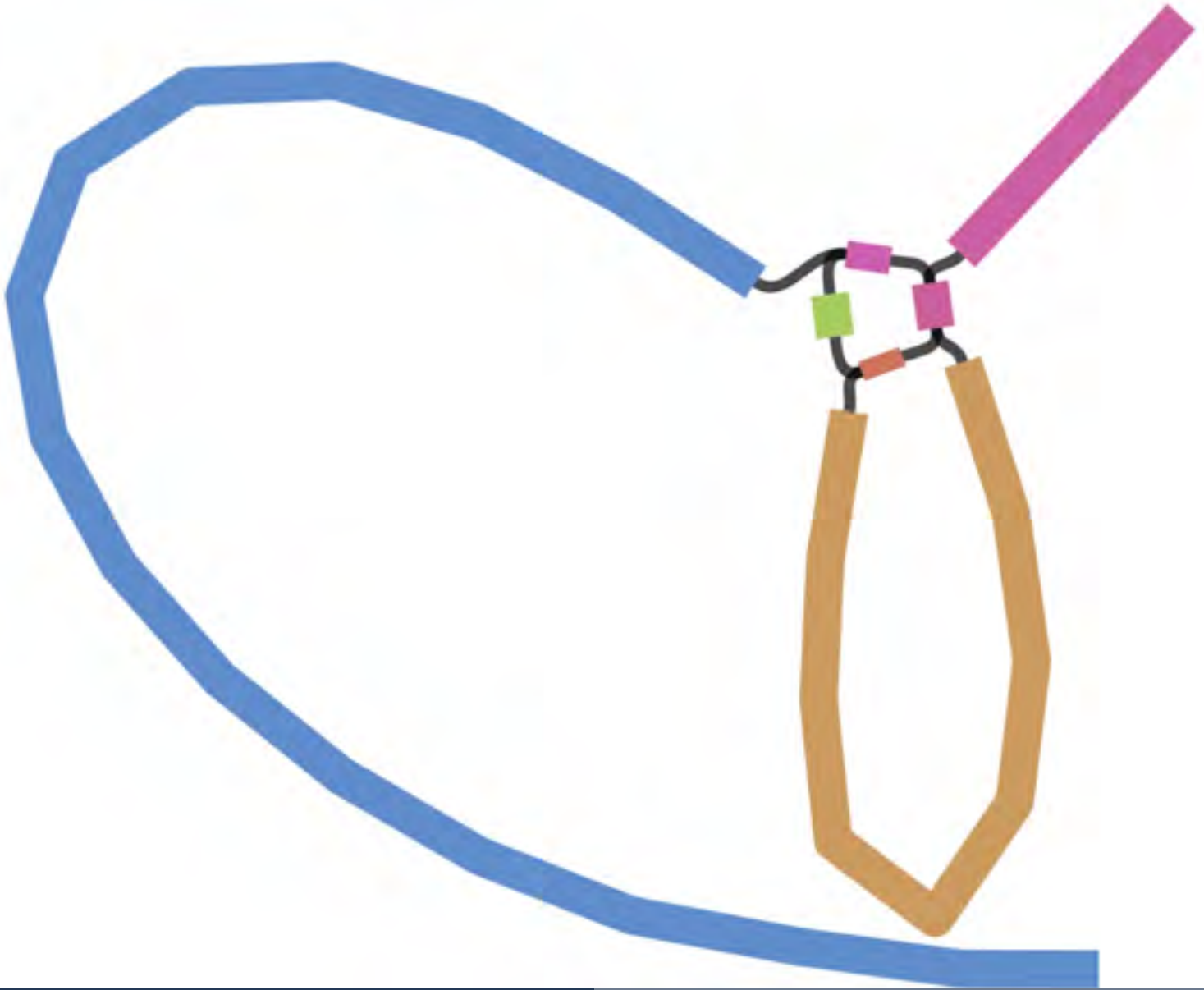
- Read length matters

Read size=1000



- Read length matters

Read size=2000



- Overlap graph simplifications

First (and most important) checkpoint

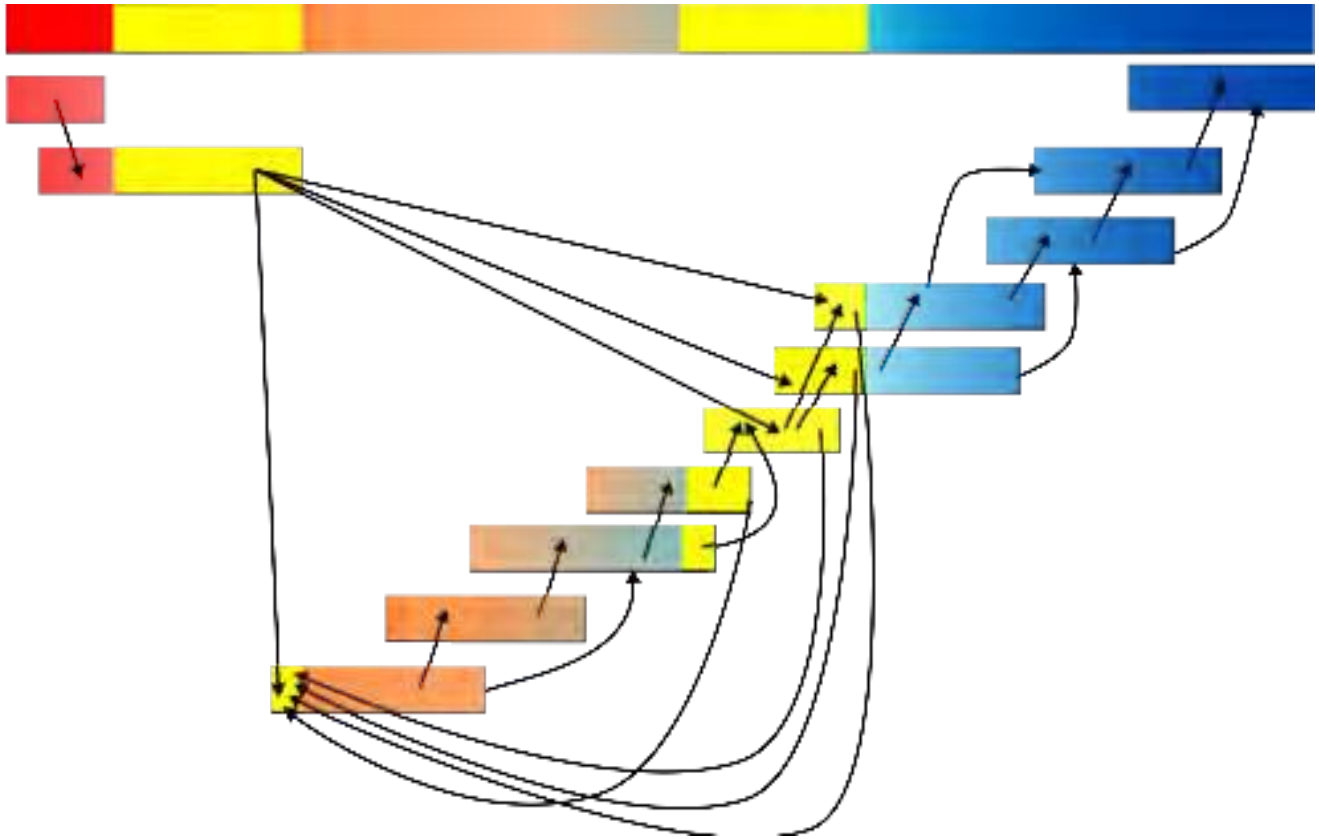
- Assembly orders reads using overlaps; longer overlaps are **generally** better.
- Multiple possible overlaps necessitate graphs for structuring information.
- Repeats longer than reads result in fragmented assembly (contigs).

- Compute overlaps

Detecting overlaps means a lot of comparisons

Compute overlaps

Even considering only the long overlaps means a lot of comparisons



Overlap graph burden: number of reads

$n(n-1)/2 = \mathcal{O}(n^2)$ possible overlaps for n reads

# Reads	# Overlaps
1000	499,500
10,000	50 million
100,000	5 billion
1 million	500 billion
10 million	50 trillion...

We have to be efficient and focus on "relevant" overlaps

Overlap graph burden: number of overlaps

For each base of the genome:

Read Coverage	Overlaps coverage
10	100
20	400
50	2,500
100	10,000

The amount of overlaps is not linear

Linear: 2X data 2X time

Quadratic: 2X data 4X time

- Another idea

- Another idea

- Another idea

- Another idea

- Another idea

- Context

- Context

- Context

- Assembly

- The de Bruijn graph

- Reconstitute larger genomic words

- de Bruijn graph assembly

- Exercise 1: de Bruijn graph time!

Hint: Use 7-mers

- Exercise 1: Solution

- de Bruijn graphs abstract redundancy

- de Bruijn graphs only rely on $k-1$ overlaps

- de Bruijn graphs limitation 1: Fixed overlaps

GGACT and ACTTA overlap is only of size 3 !

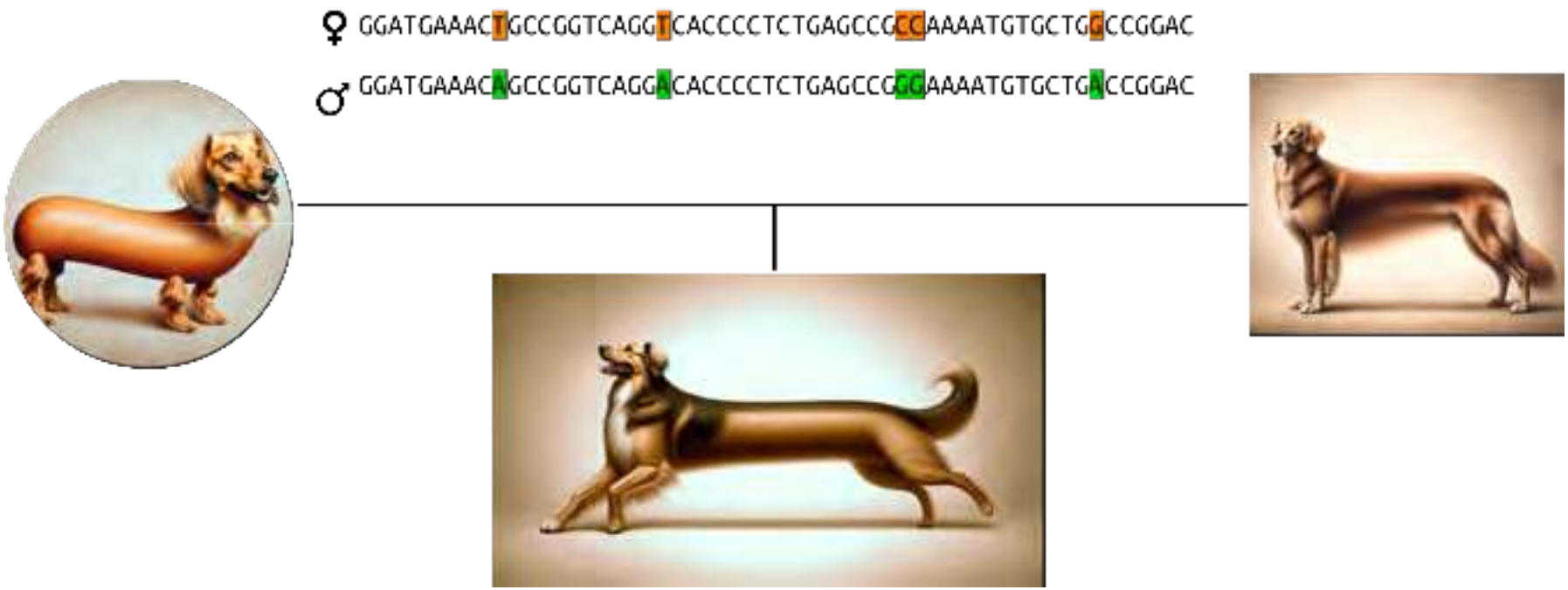
- de Bruijn graphs limitation 2: Repeats

each k-mer appears only once in a de Bruijn graph

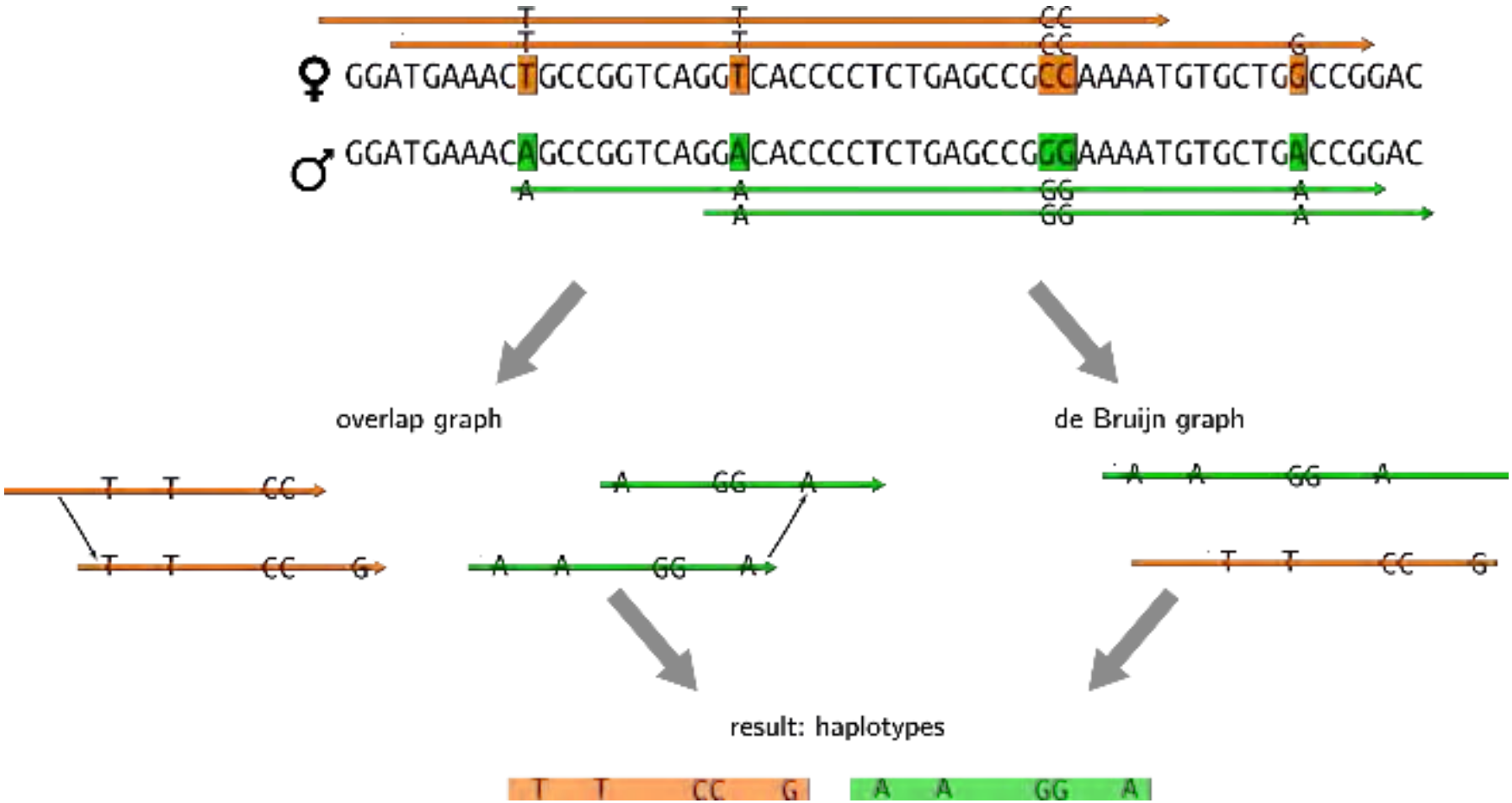
- de Bruijn graph limitation

- On the representation of de Bruijn graphs

- The boy is diploid!

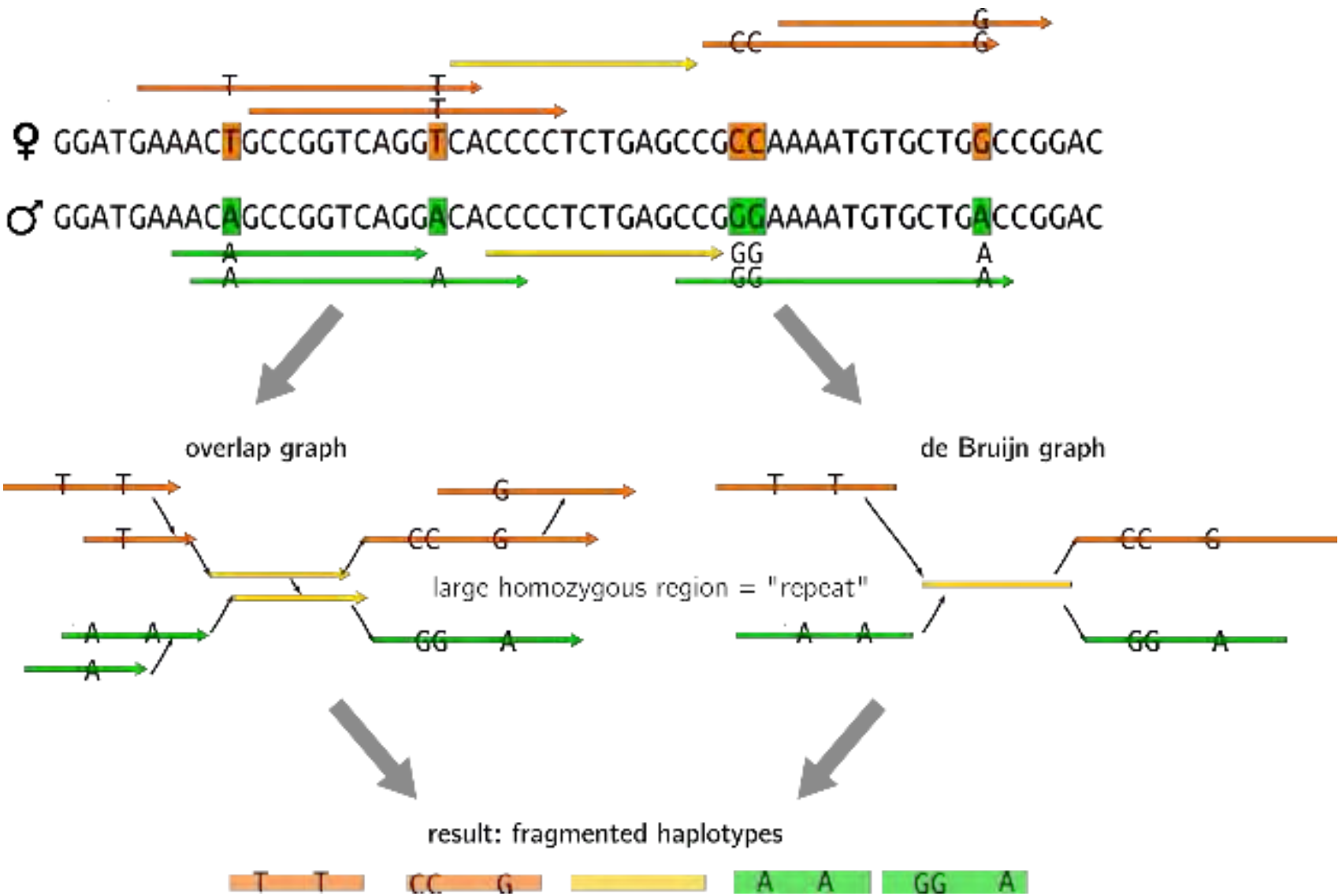


Ploidy and very long reads



Haplotypes can be "phased"

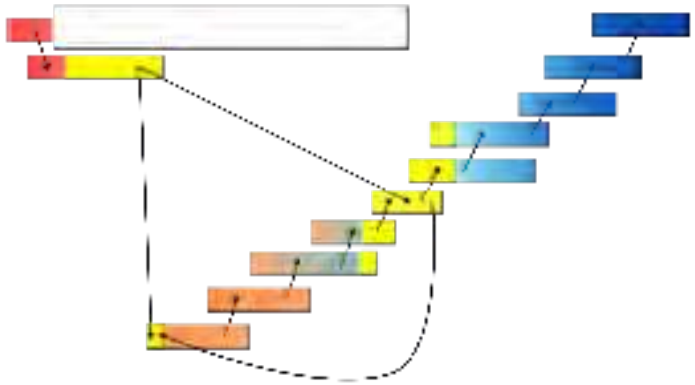
- Homozygous vs heterozygous regions



Assembly concession: assembly can be fragmented due to ploidy

• Method checkpoint: de Bruijn graph versus overlap graph

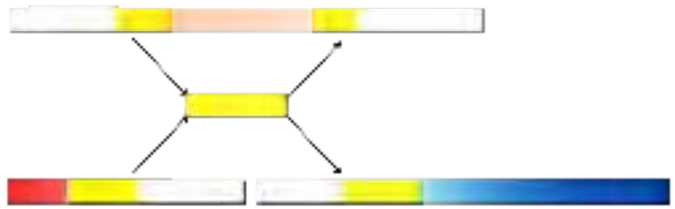
Overlap graph



quadratic growth with coverage

issue with repeats larger than reads

(compacted) de Bruijn graph



abstracts coverage

issue with repeats larger than k

- Data checkpoint: results for the *long, perfect boy*

  - Contigs can reach the chromosome's order of magnitude in length (megabases)
  - Breaks due to large repeats
  - Haplotypes can be partially reconstructed

- Second experiment: *noisy, super long boy*'s genome

- de Bruijn graph or overlap graph?

- Sequencing errors and k-mers

- Erroneous k-mers do not connect

Most k-mers will contain at least an error and will be useless
(not connected to the other reads)

$\rightarrow$ **de Bruijn graph out!**

- Overlap graph: inexact matches

Quadratic alignment for each pair of reads
Quadratic number of comparisons to perform ...

- Overlap graph: drop alignment

Procedure called *anchor chaining*.

- How to get accurate contigs from noisy reads?

- Using coverage to remove noise: consensus

- Exercise 2: Perform a consensus

```
read1: ACTTCGAACGT
read2: TCGATCGTTT
read3: GATCAGTTTAG
read4: TCATTTCGTA
read5: GTTTCGTCCG
ref:   ACTCGAATGTTTTCCTACG
```

- Exercise 2: Perform a consensus - solution

```
read1: ACTTCGAACGT
read2:   T-CGA-TC-GTTT
read3:     GA-TCAGTTT-AG
read4:       TCA-TTT-CGTA
read5:         GTTT-CGTCCG
ref:   ACT-CGAAT--GTTTTCCTACG
cons:  ACT-CGA-TCAGTTT-CGTACG
```

- Consensus before assembly: correction

- Consensus during assembly (hence the OL**C**)

- Consensus during assembly. Yes but:

- Consensus after assembly: polishing

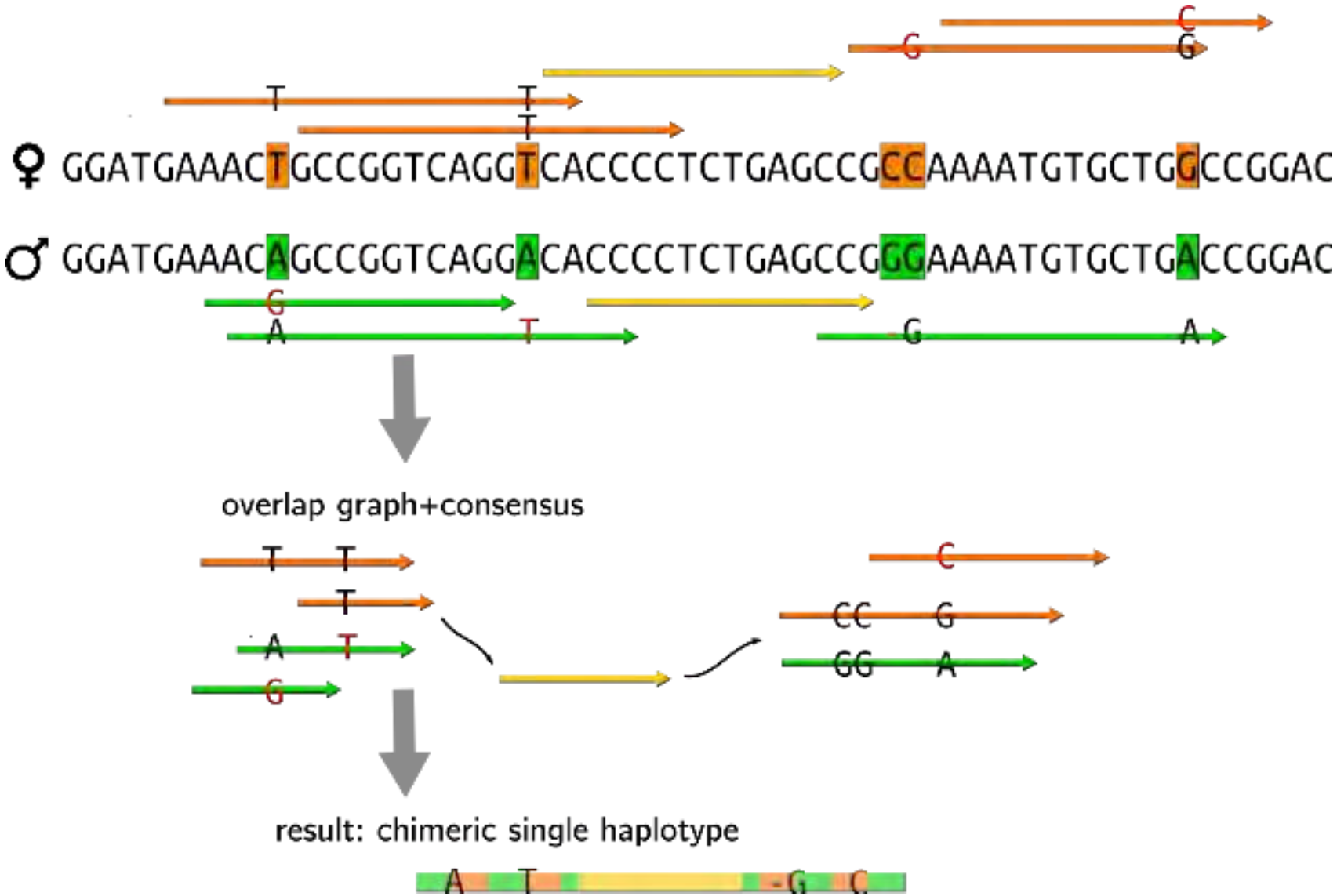
Correction/Consensus during assembly/Polishing

- Correction $\times$
  - Redundancy:100X coverage $\rightarrow$ 100X more bases to correct
- Consensus during assembly $\approx$
  - Do not use all reads
- Polishing $\checkmark$
  - Correct each base of the genome once
  - Use all reads

- Consensus destroys heterozygosity

$\rightarrow$ a mix between the two alleles

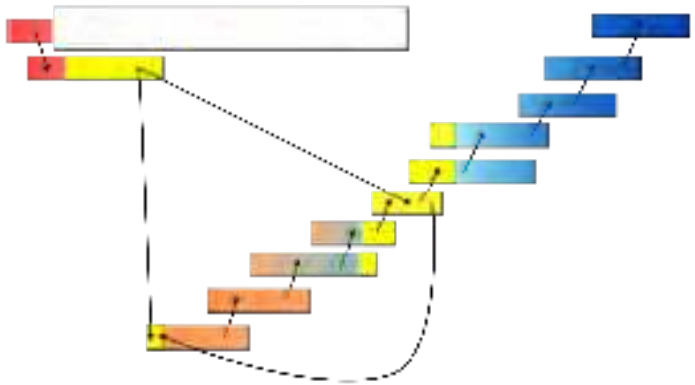
- Consensus destroys heterozygosity



Assembly concession: "haploid" assembly due to errors

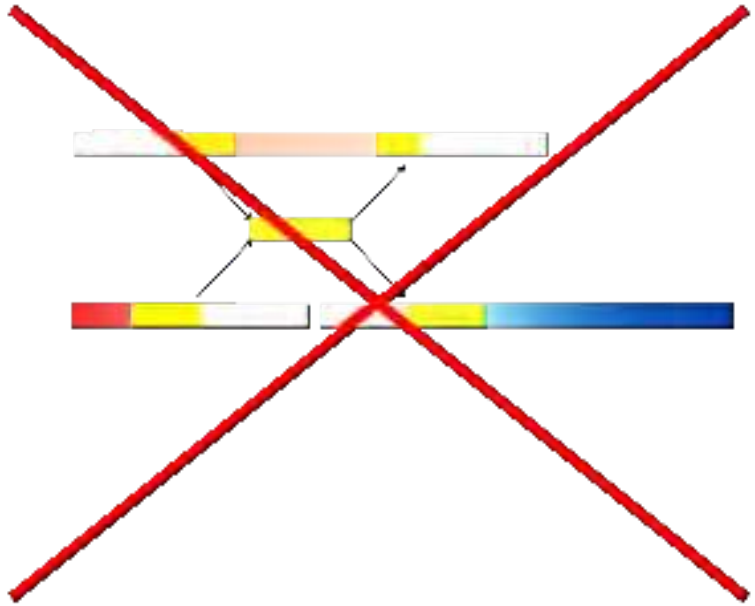
- Method checkpoint: de Bruijn graph vs overlap graph

Overlap graph



can deal with approximate overlaps
(anchor chaining)
polishing

(compacted) de Bruijn graph



too many errors prohibit connectivity

- Data checkpoint: results for the *noisy super long boy*

  - Contigs can reach the chromosome's order of magnitude in length (megabases)
  - Breaks due to very large repeats
  - Contigs are chimeras of haplotypes

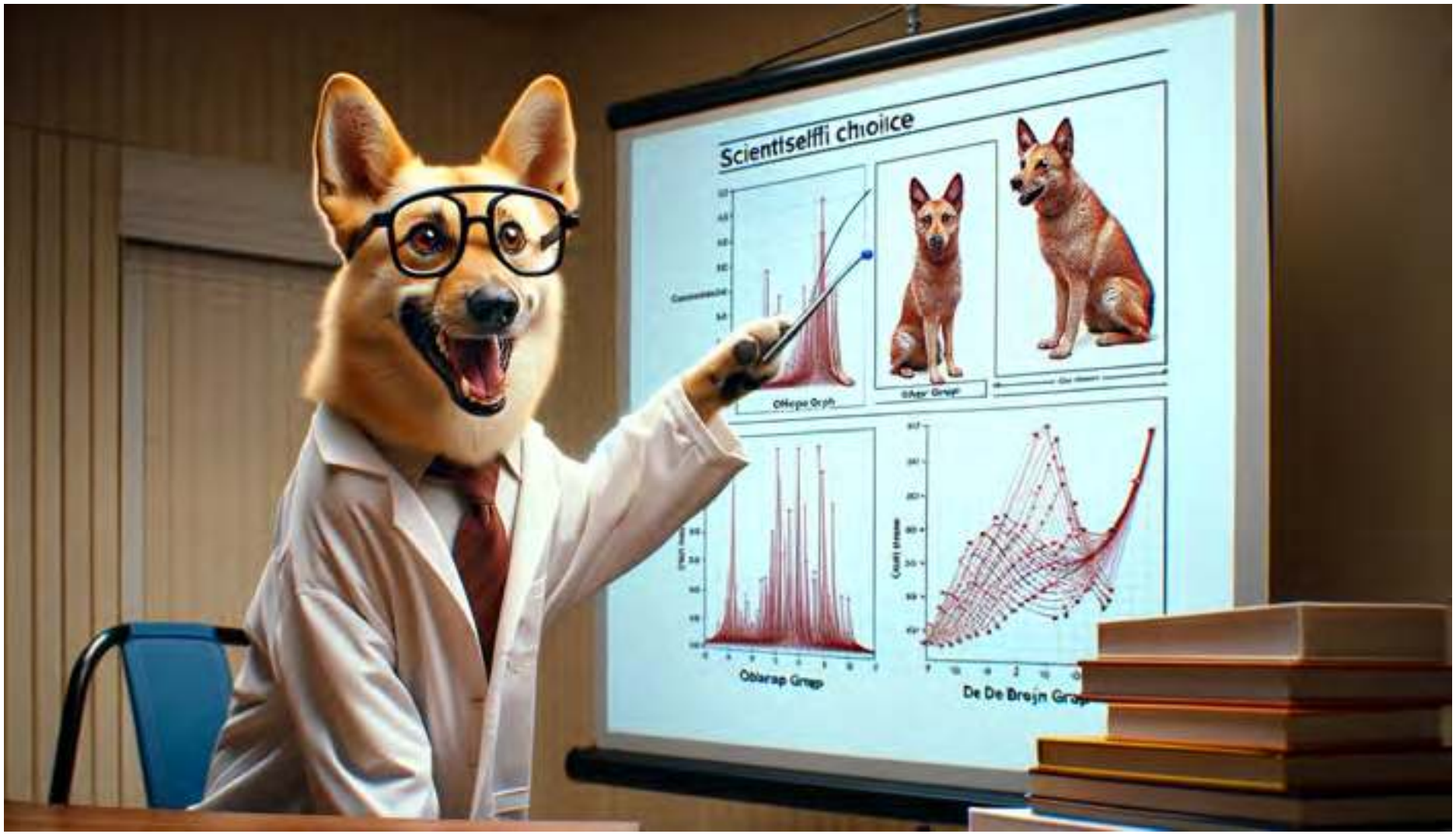
- Third experiment: *short boy*'s genome

100kb region from the genome
(only for the record, we actually don't have it)

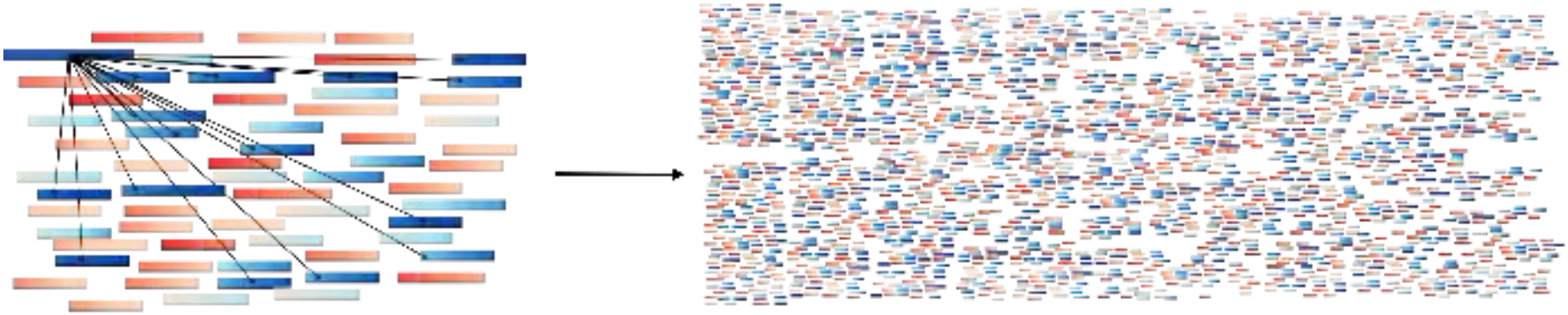
Genome size
1 billion bases

Reads
1 billion
size 100 bases
<1% error

de Bruijn graph or overlap graph?



Scalability issue for the overlap graph



At equal coverage we got:

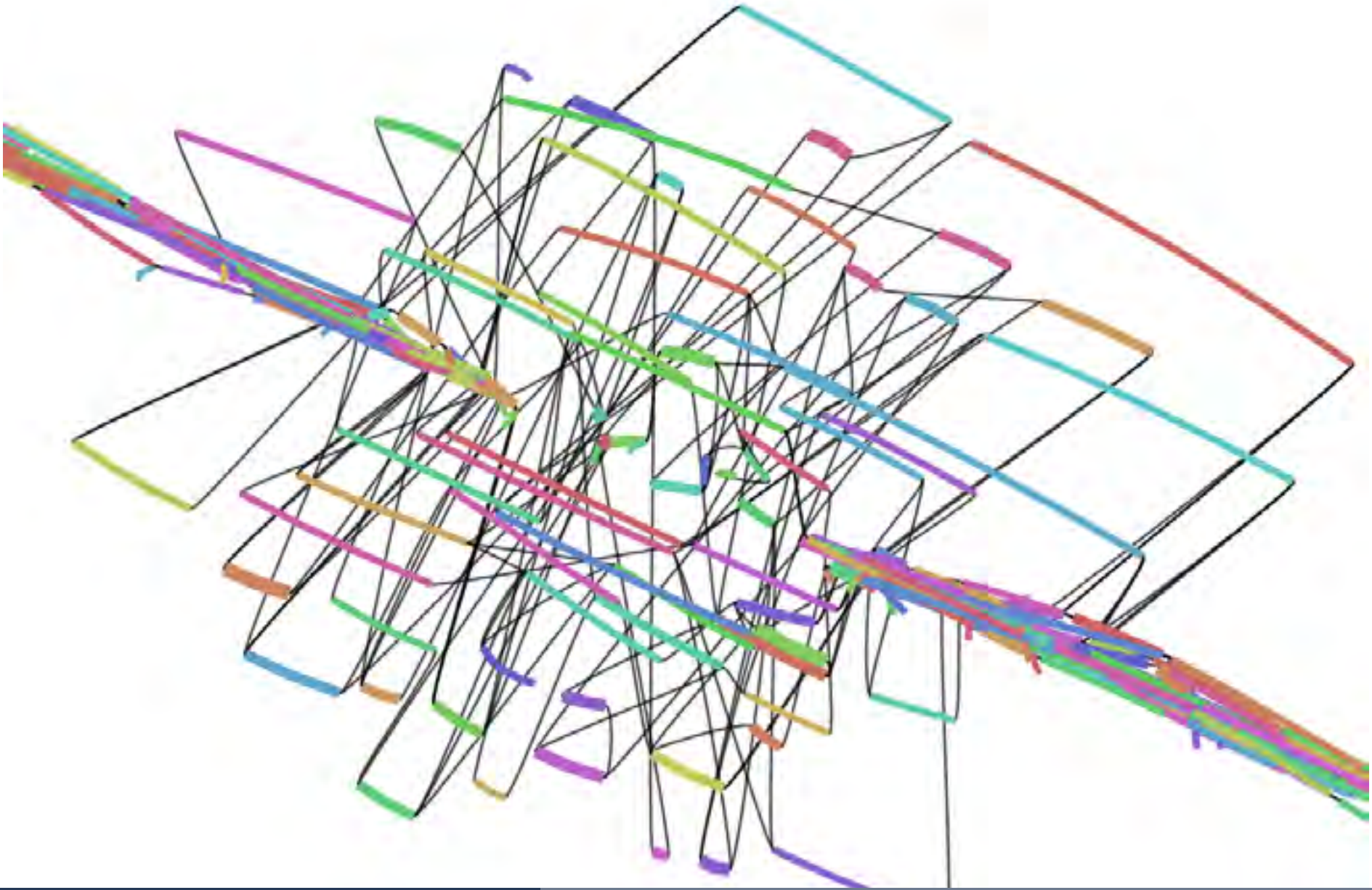
1000 $\times$ more reads $\rightarrow$ 1 million $\times$ more overlaps to check!

Overlap graph hardly scales to such a large number of reads/overlaps

$\rightarrow$ **Overlap graph out!**

- de Bruijn graph on a real dataset

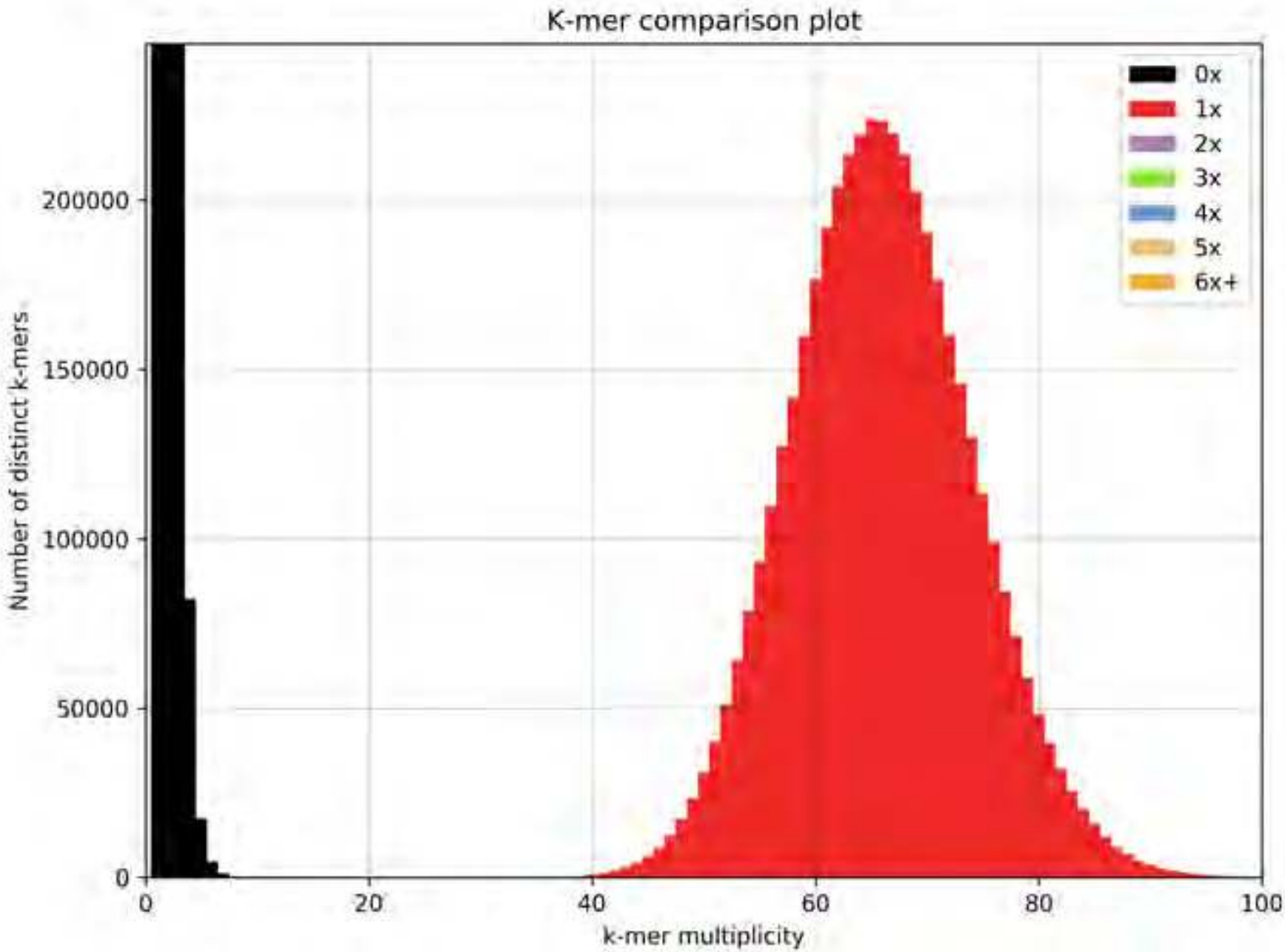
- de Bruijn graph on a real dataset ZOOMED IN



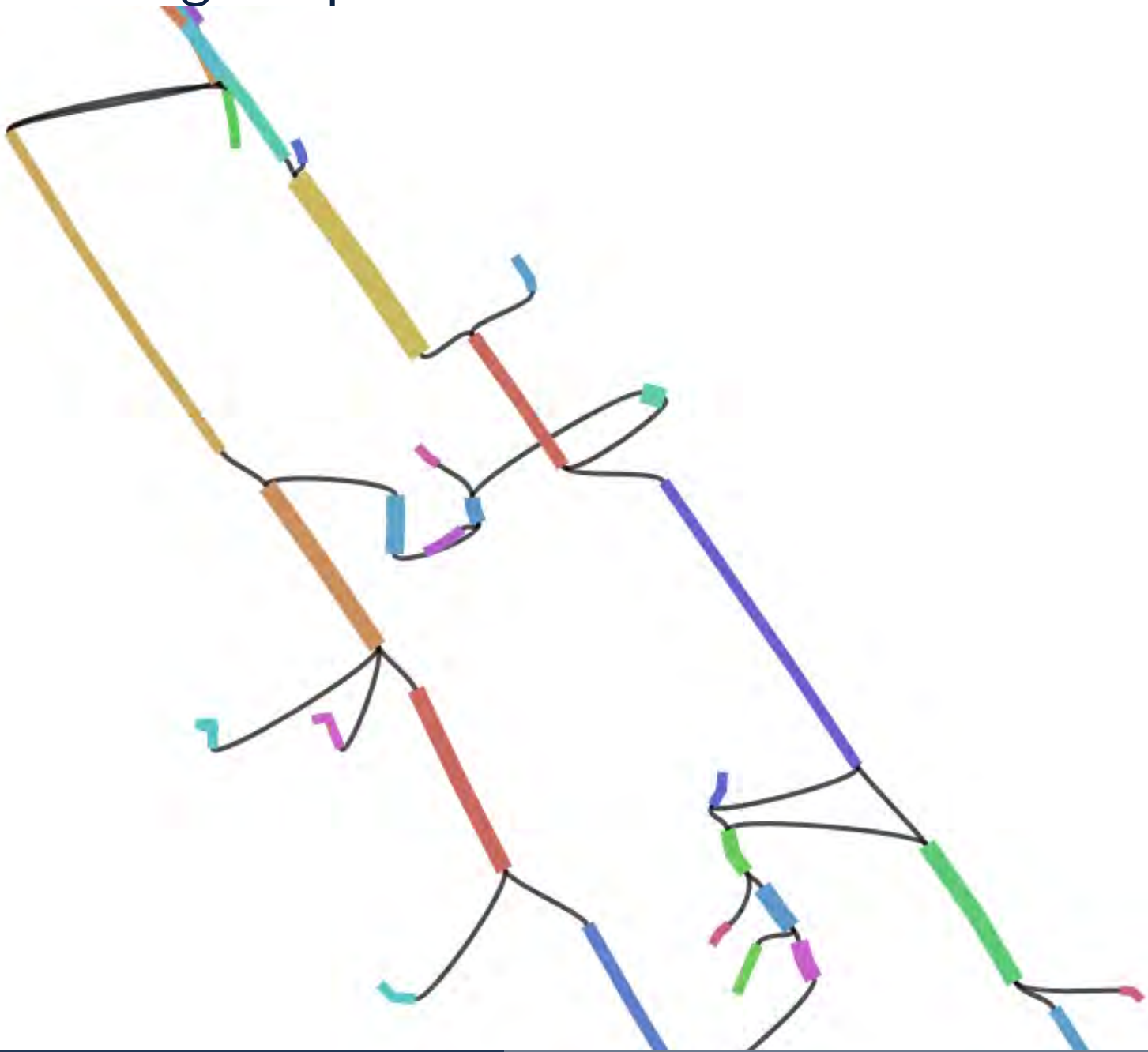
- Erroneous k-mers vs genomic k-mers

Erroneous k-mers are seen less than genomic ones

- *K*-mer histogram

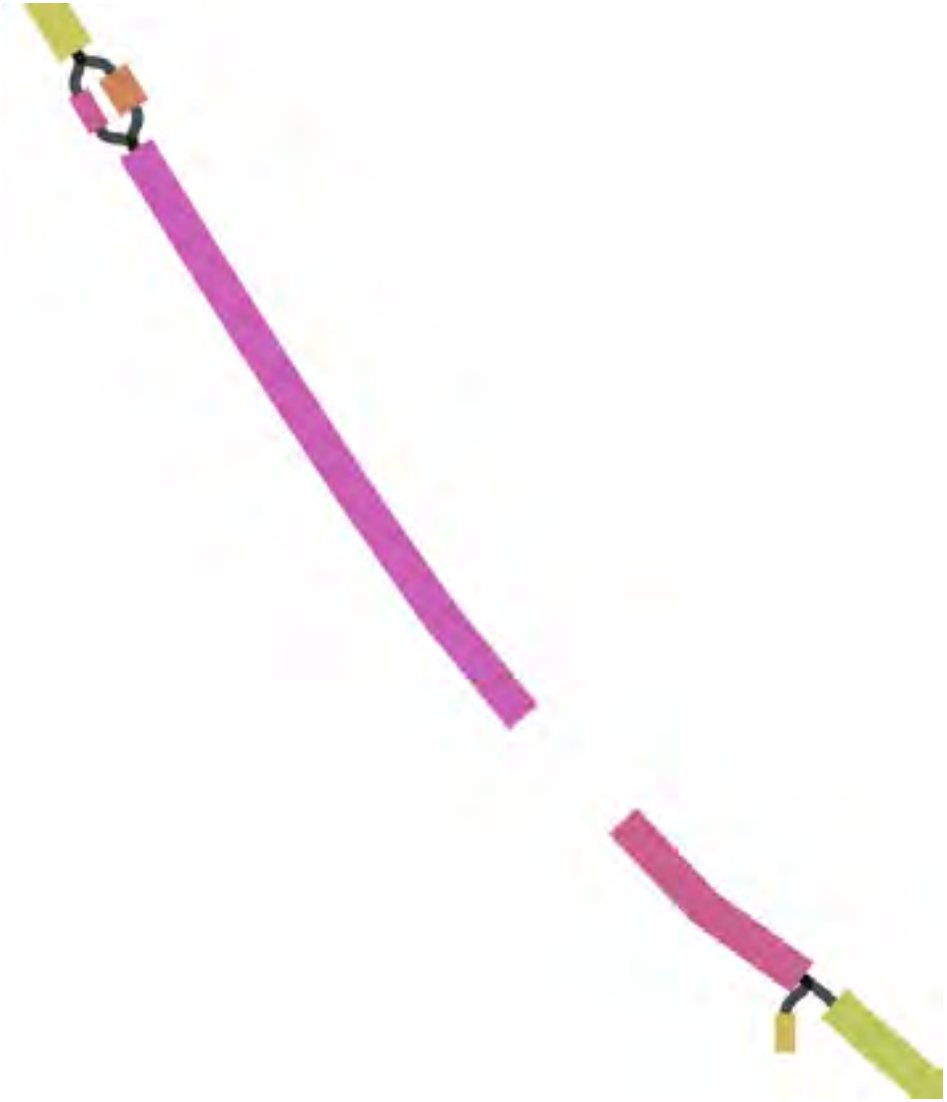


- Removing unique k-mers

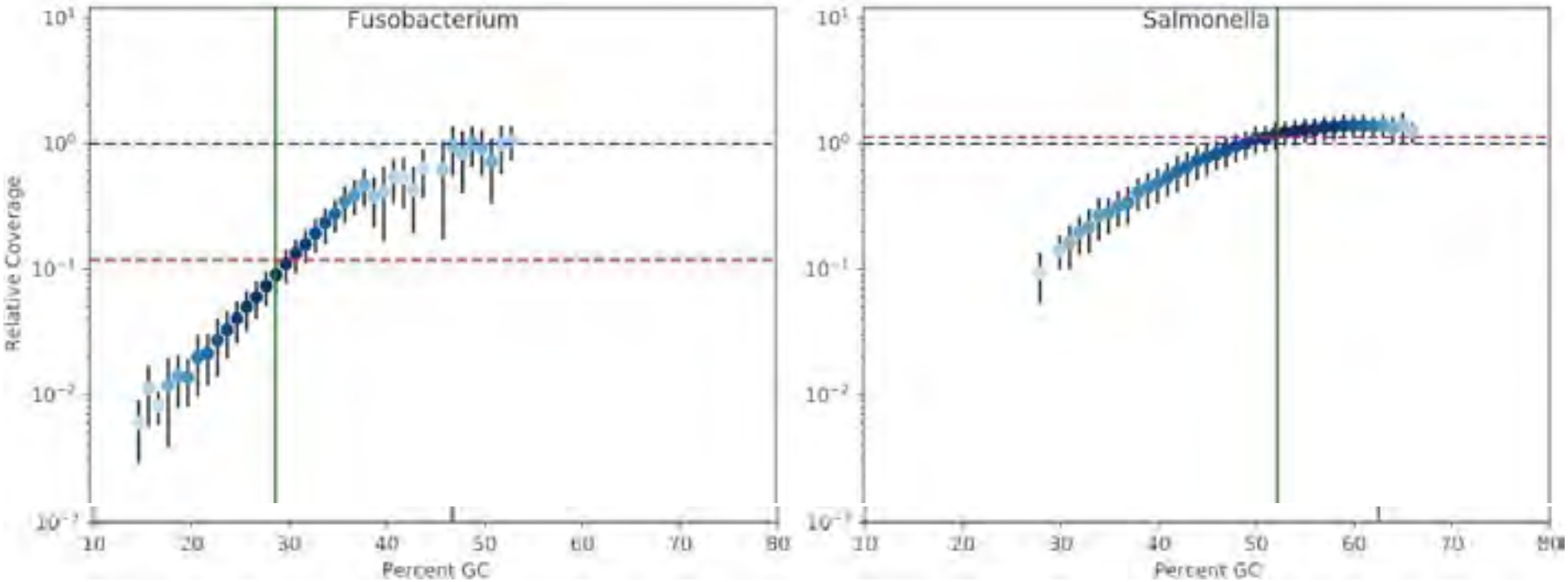


- Removing k-mers seen less than 3 times

- Removing k-mers seen less than 4 times



GC bias

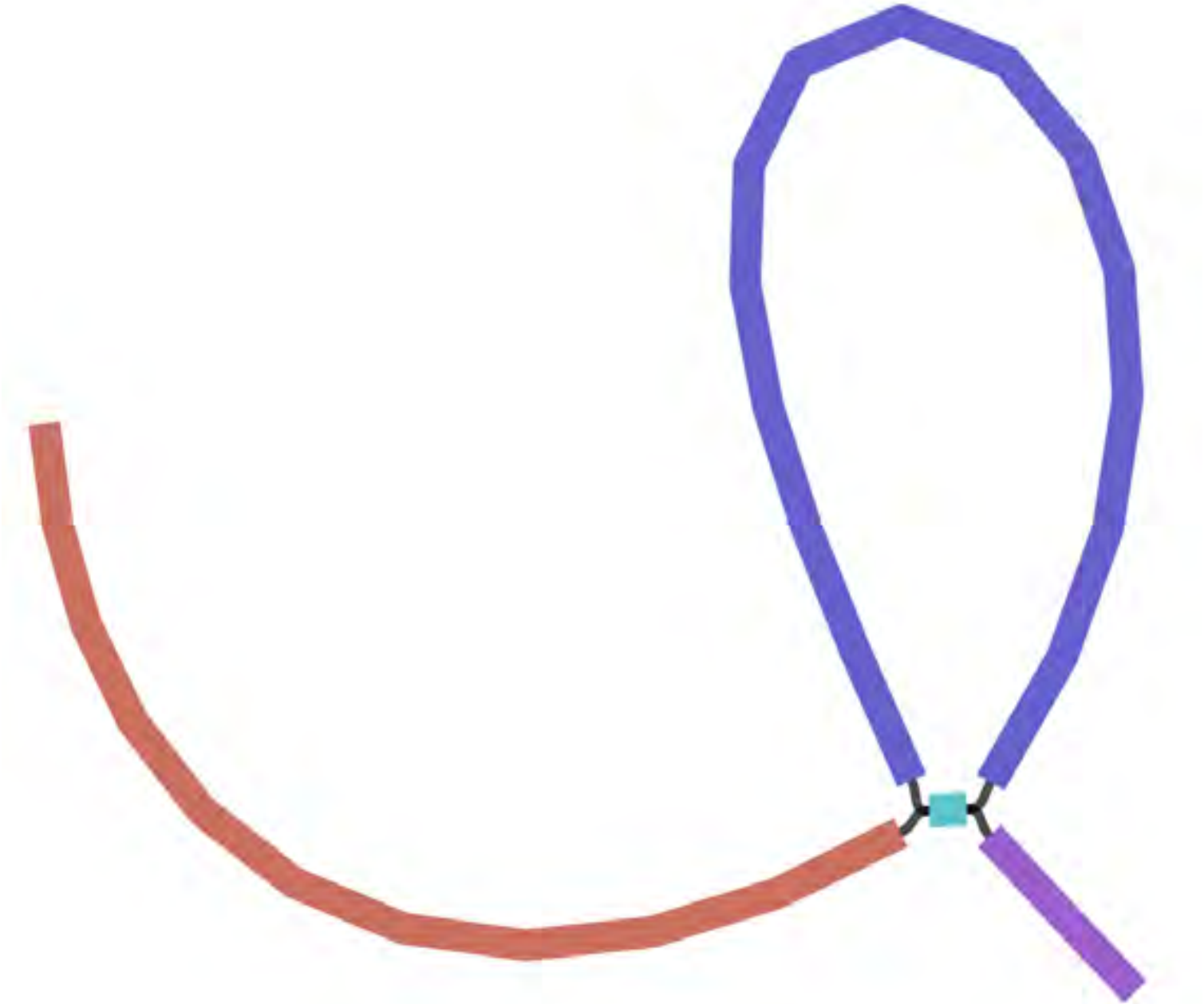

Fusobacterium
Salmonella
Relative Coverage
Percent GC
10^{1}
10^{0}
10^{-1}
10^{-2}
10
20
30
40
50
60
70
80

- Errors in de Bruijn graphs

- Errors in de Bruijn graphs

- Errors in de Bruijn graphs

- Almost assembled phage !



- de Bruijn graph on my diploid genome

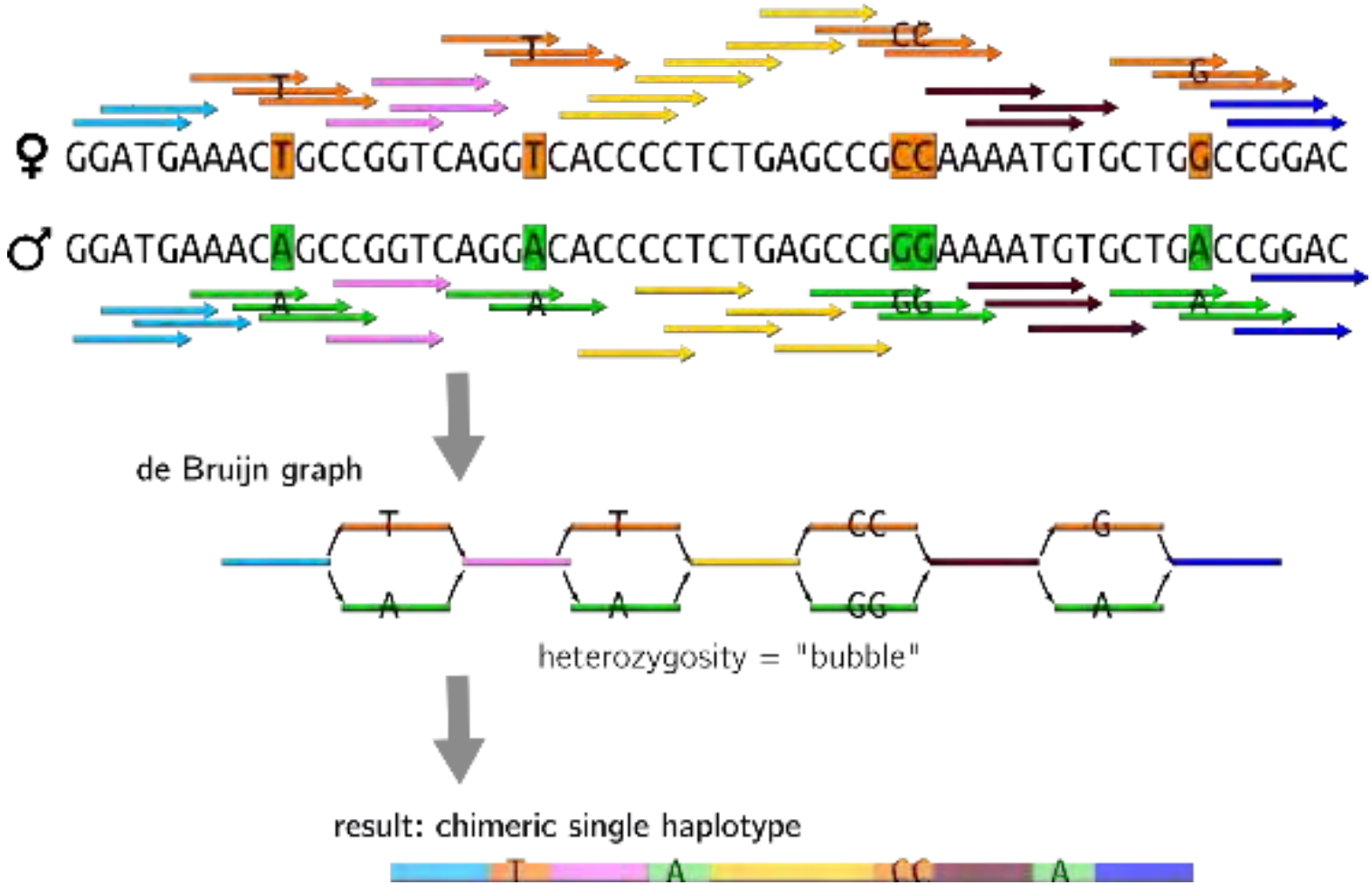
- Ploidy and de Bruijn graph

- Bubble crushing

- Variants are not "lost"

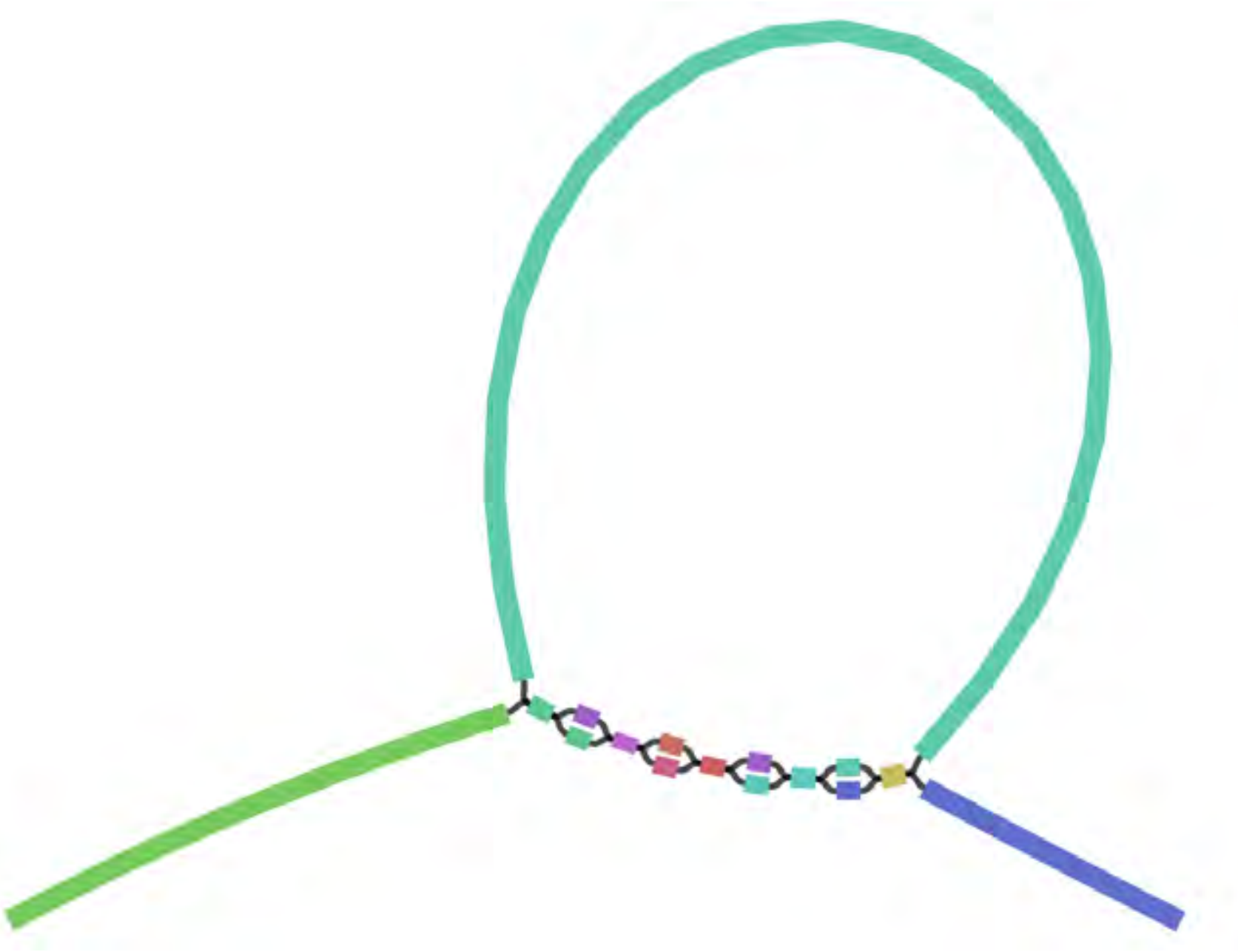
We can align the reads to the assembly and do variant calling

Haploid assembly



Assembly concession: haplotypes are collapsed when using short reads

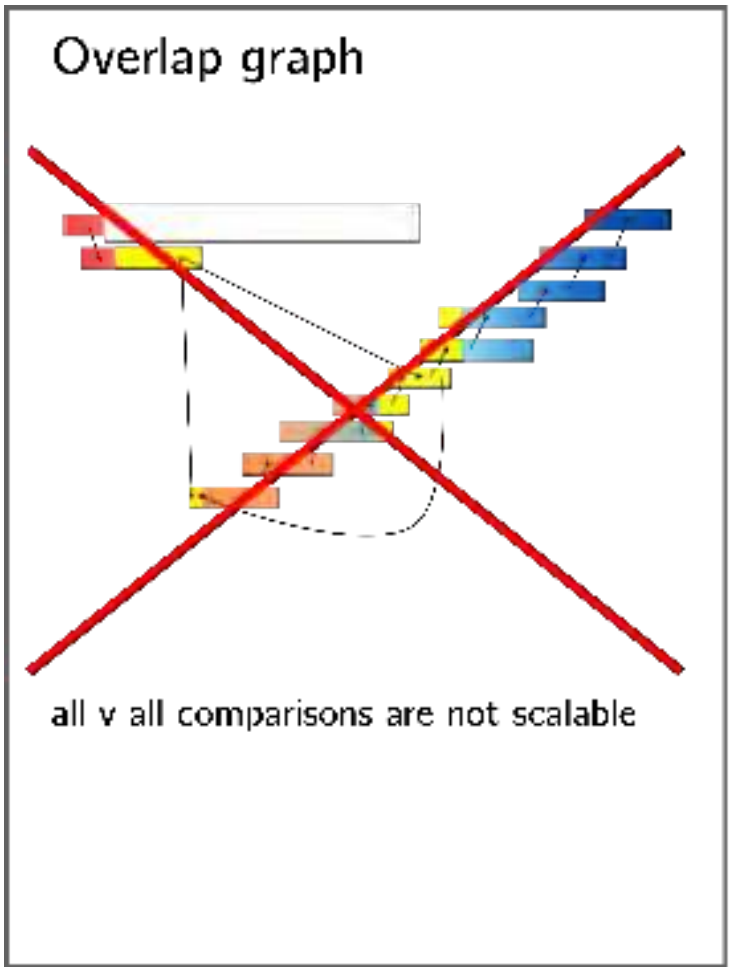
- Paralog genes/repeats

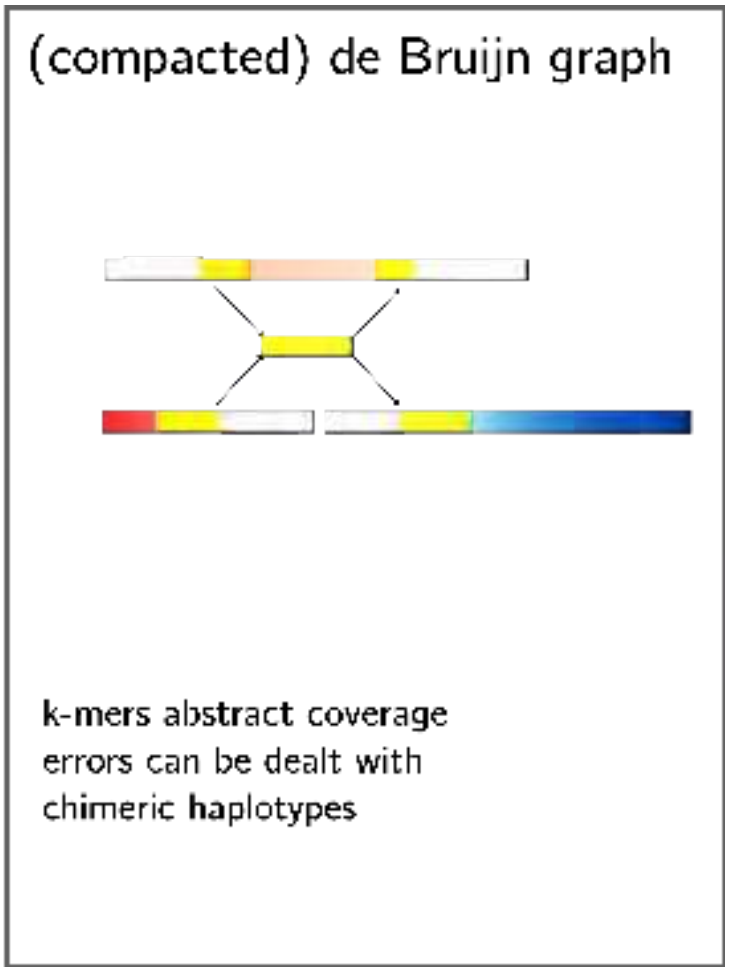


- Paralog genes/repeats

- Paralog genes/repeats in graph

- Method checkpoint: de Bruijn graph versus overlap graph





- Data checkpoint: *short boy* results

  - Very fragmented assembly of short contigs (mostly below 100kb)
  - Very high base accuracy
  - Contigs are chimeras of haplotypes
  - Can miss extreme GC content

- Fourth experiment: *golden boy*'s genome

Billion $ project $\rightarrow$ cancelled

(Time accurate) recap

[illegible]

No longer used for assembly (too expensive)

[illegible]

De Bruijn graph assembly
Fragmented haploid assembly

[illegible]

Overlap graph assembly (+ polishing)
Contiguous haploid assembly

[illegible]

Overlap graph or de Bruijn graph assembly
Contiguous diploid assembly

- Back to the present

Challenge 1: Scalability

- Human Genome project (2001)
- 1000 Genomes project (2015)
- 10k Genomes project (2016)
- 100k Genomes project (2018)
- 500K UK genomes (2023)

Many ambitious sequencing projects beyond human: Earth biogenome project, Vertebrate genome project ...

History

How long to assemble a human genome?

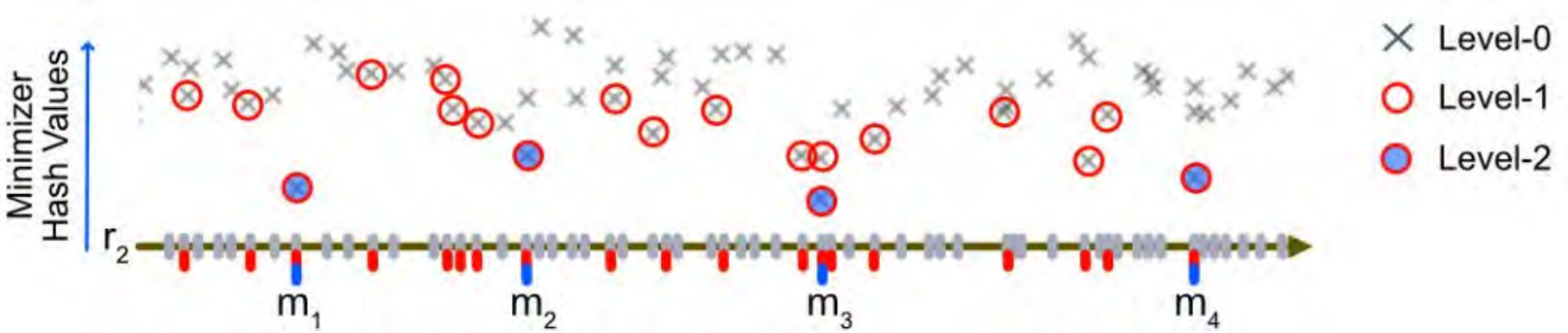
- Sanger: MANY CPU **years**
- Illumina (Overlap graph): 2 CPU **months**
- Illumina (De Bruijn graph): A CPU **day**
- Long reads (Alignment): 2 CPU **years**
- Long reads (Anchors chaining): 20 CPU **days**
- HiFi (Anchors chaining): 2 CPU **days**
- HiFi (De Bruijn graph): A CPU **hour**

Algorithms and data structures matter!

Also long and precise reads are easier to assemble

Very fast genome assembly with HiFi

Human genome assembled within 2 hours (Peregrine assembler) and 10 minutes (RMBG assembler)



- Telomere to telomere assembly?

Challenge 2: Telomere to telomere chromosomes

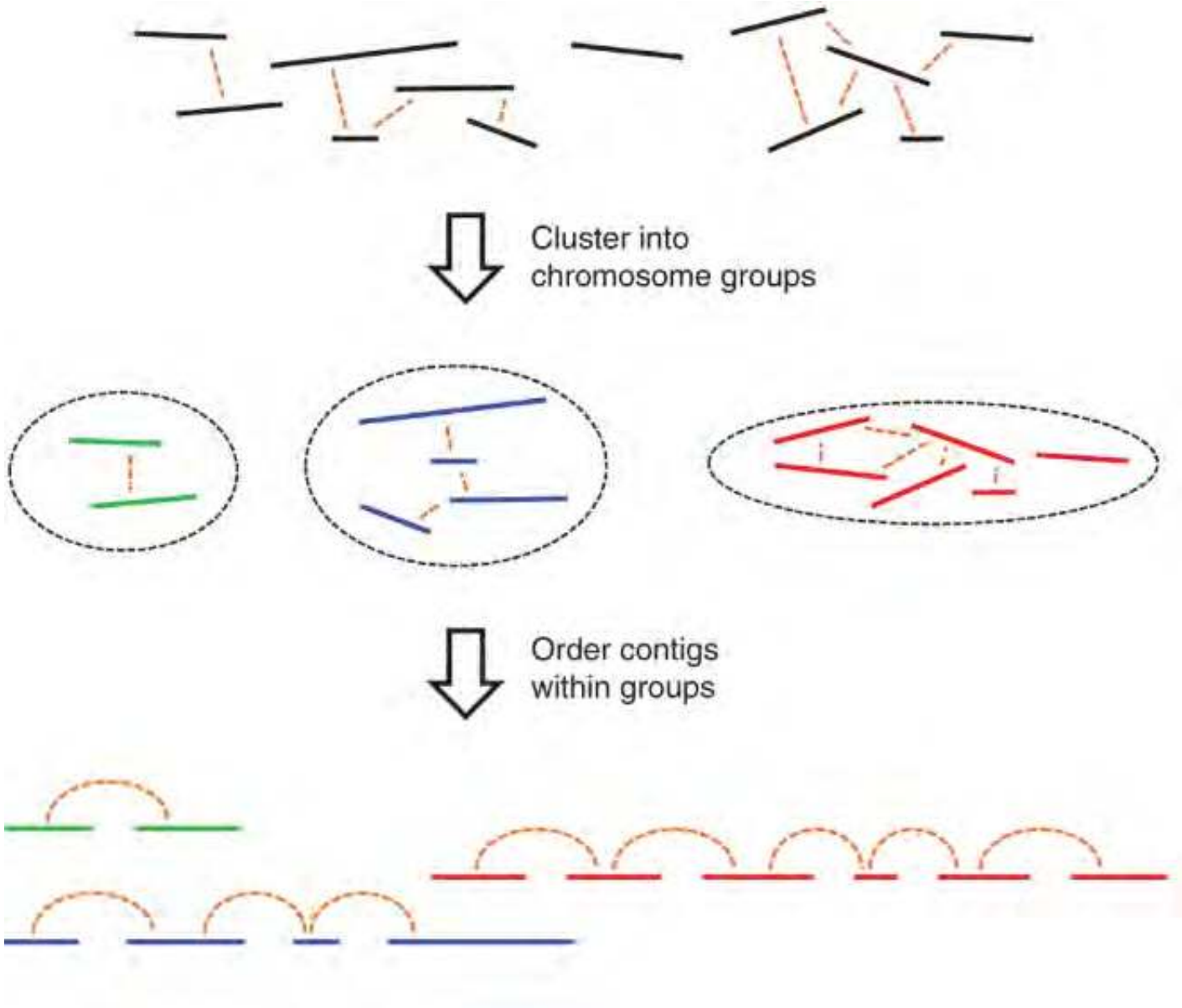
Main problems:

- Very large exact repeats
- Very similar sequences accross the genome
- Low complexity regions
- Mosaic repeats

Need long distance information **AND** high base accuracy

Scaffolding

Use long range information to order contigs into "scaffolds"

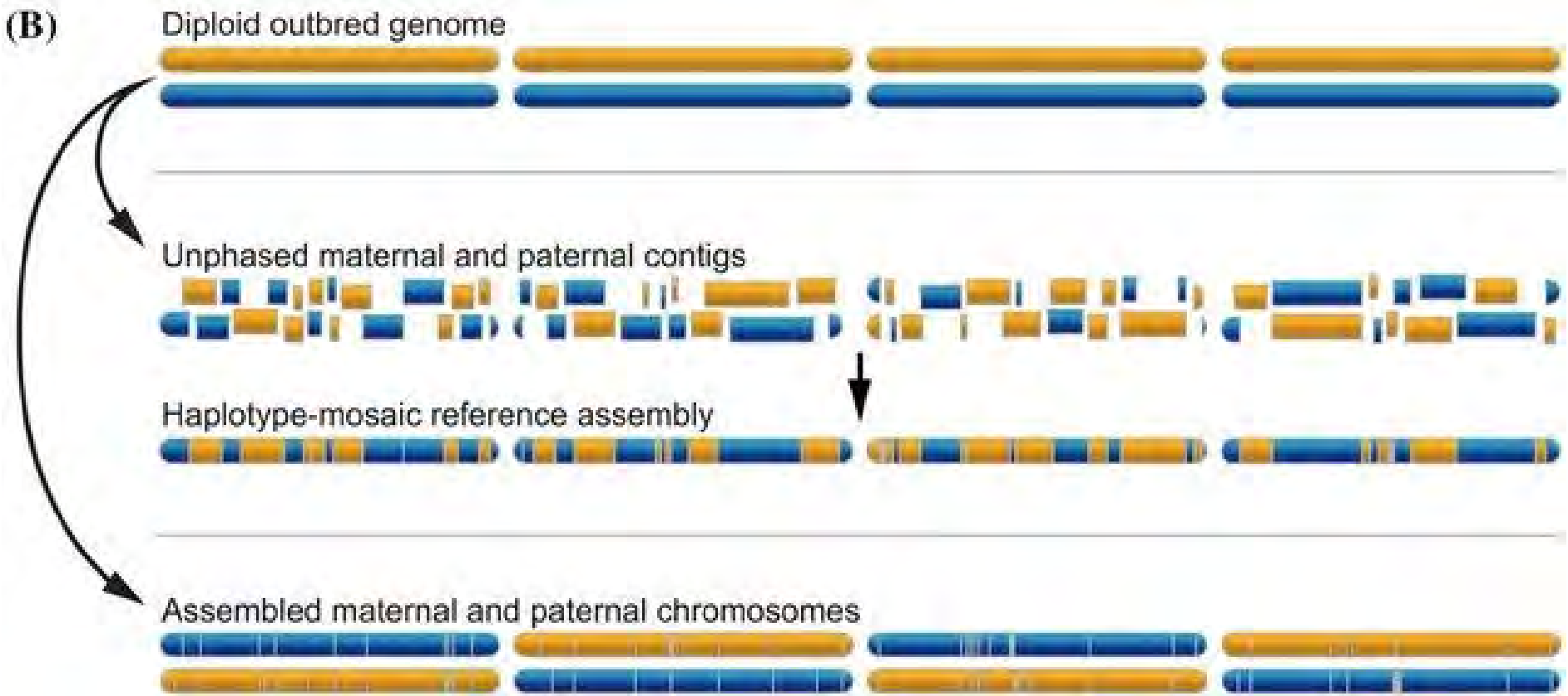


- Telomere-to-Telomere consortium

Has produced in 2021 a complete human genome with one contig per chromosome !

- 30x PacBio HiFi
- 120x coverage of Oxford Nanopore (ultra long reads)
- 70x PacBio CLR
- Arima Genomics HiC
- BioNano DLS
- 100 authors from 50 labs

- Diploid assembly



Telomere-to-Telomere diploid human reference

T2T-YAO released in 2023 a complete human genome with one contig per chromosome !

- 92x PacBio HiFi
- 336x coverage of Oxford Nanopore (ultra long reads)
- 70x PacBio CLR
- 584x Arima Genomics HiC
- BioNano DLS
- Illumina HiSeq 150bp for the son and parents (with 278x and 116x coverage, respectively).

The human genome is not THAT hard

Hall of fame of largest assembled genomes of their time:

- Pine (20Gb)

The human genome is not THAT hard

Hall of fame of largest assembled genomes of their time:

- Pine (20Gb)
- Axolotl (32Gb)

The human genome is not THAT hard

Hall of fame of largest assembled genomes of their time:

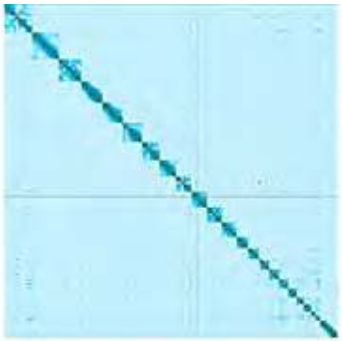
- Pine (20Gb)
- Axolotl (32Gb)
- Lungfish (43Gb)

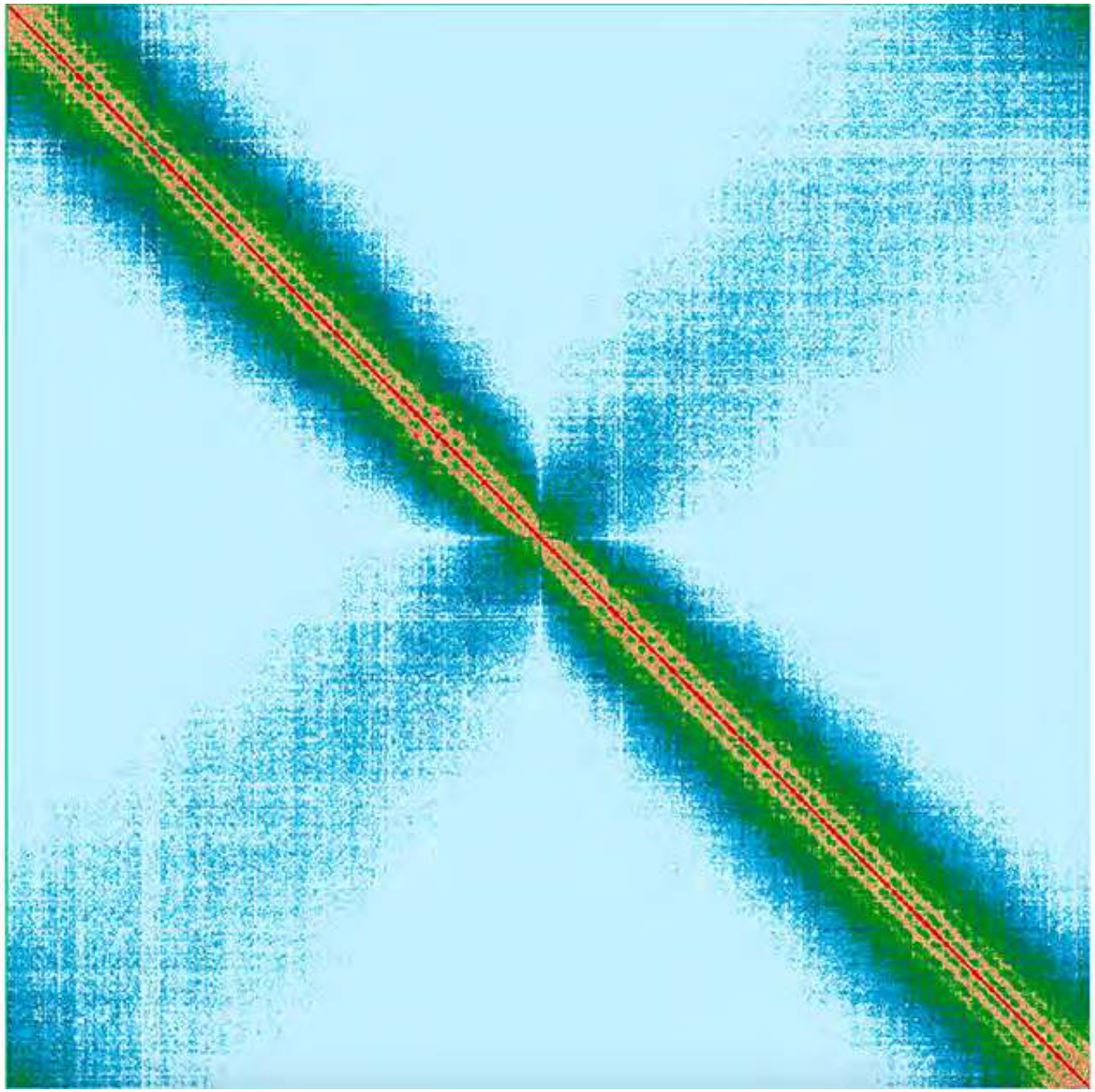
The human genome is not THAT hard

Hall of fame of largest assembled genomes of their time:

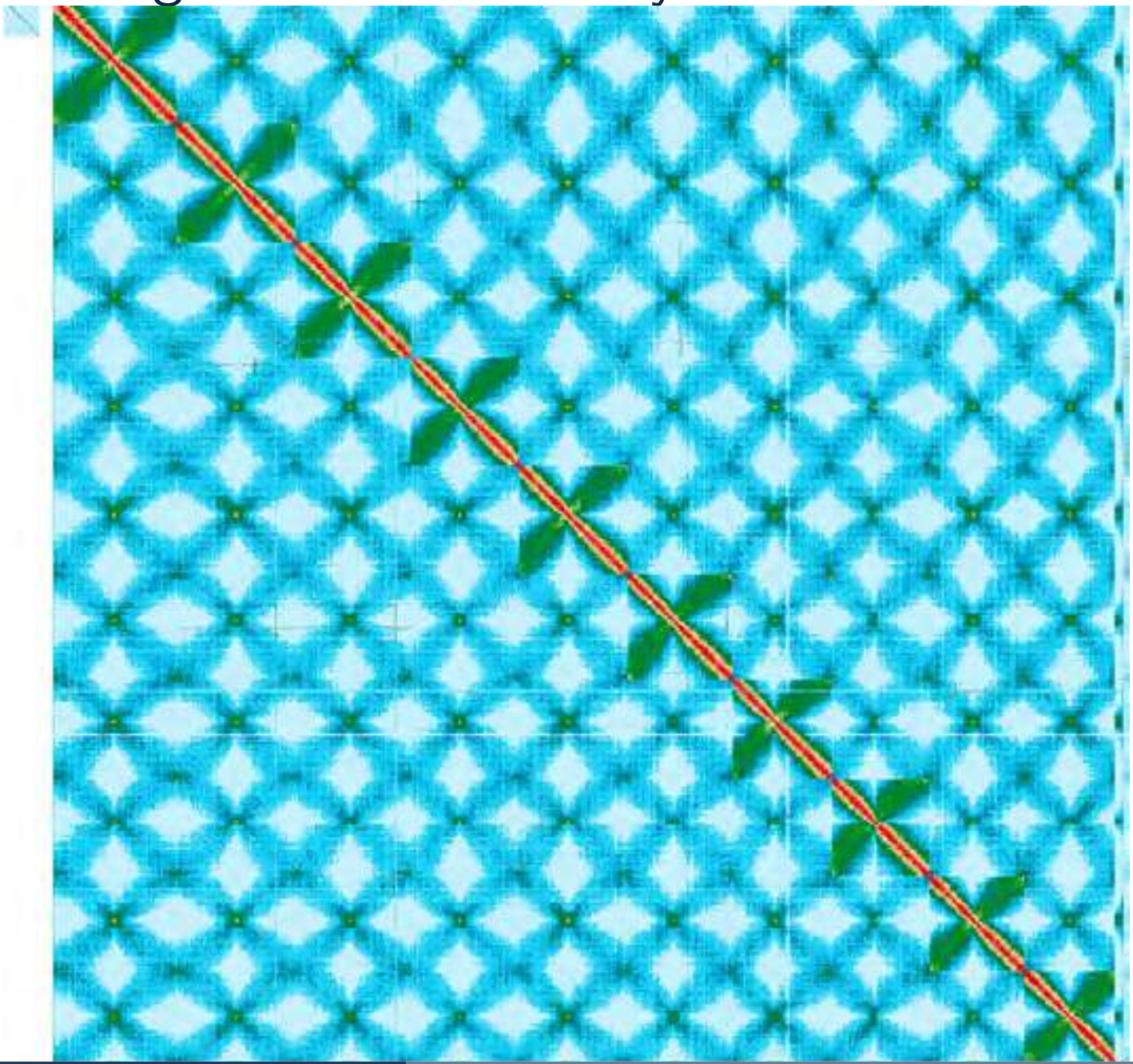
- Pine (20Gb)
- Axolotl (32Gb)
- Lungfish (43Gb)
- Mistletoe (90Gb)
- Metagenomes ...

- The human genome seems small



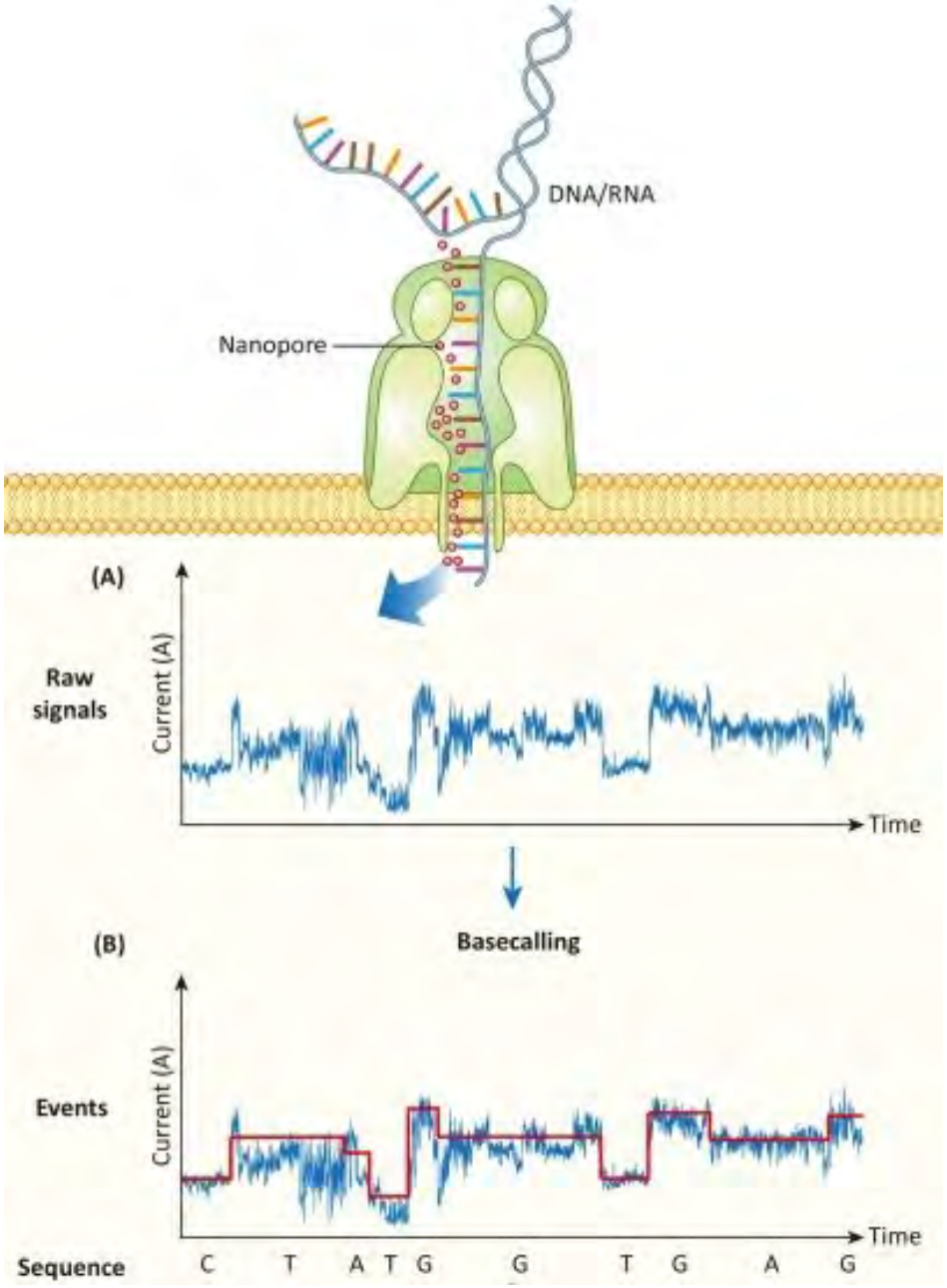


- The human genome seems really small



- Challenge 3: Base level accuracy

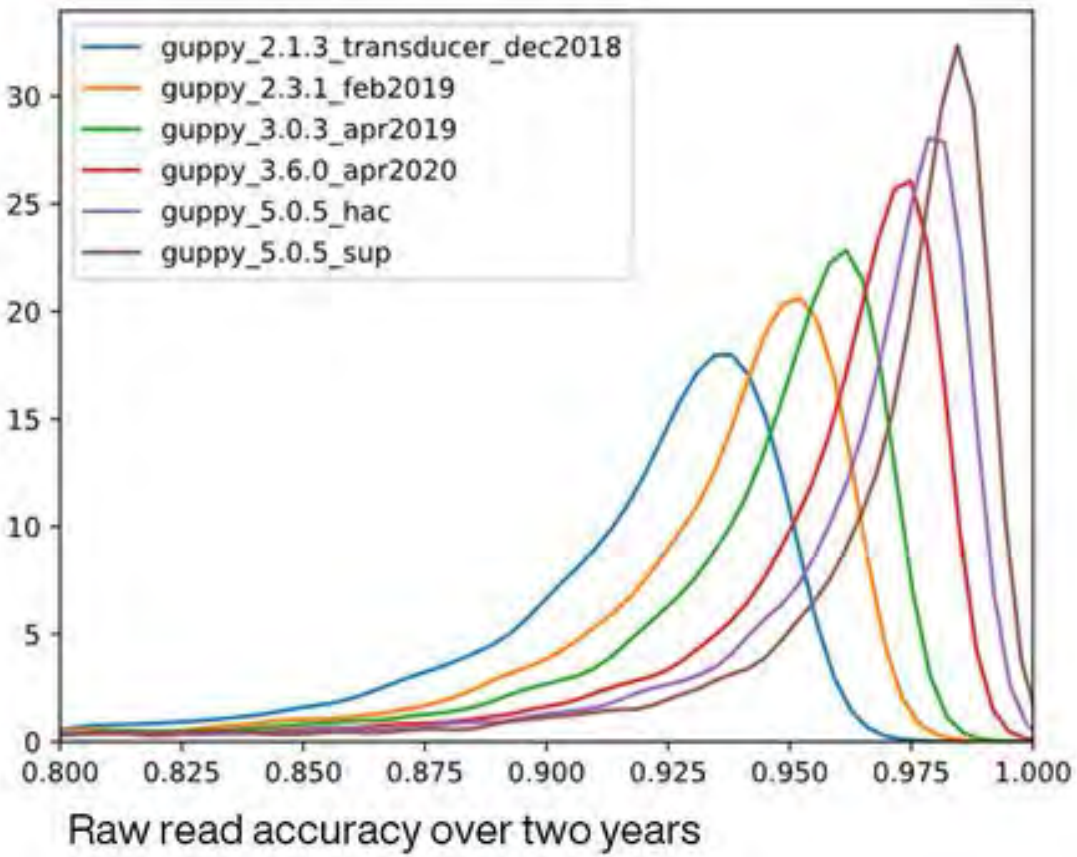
- Homopolymers are hard to read



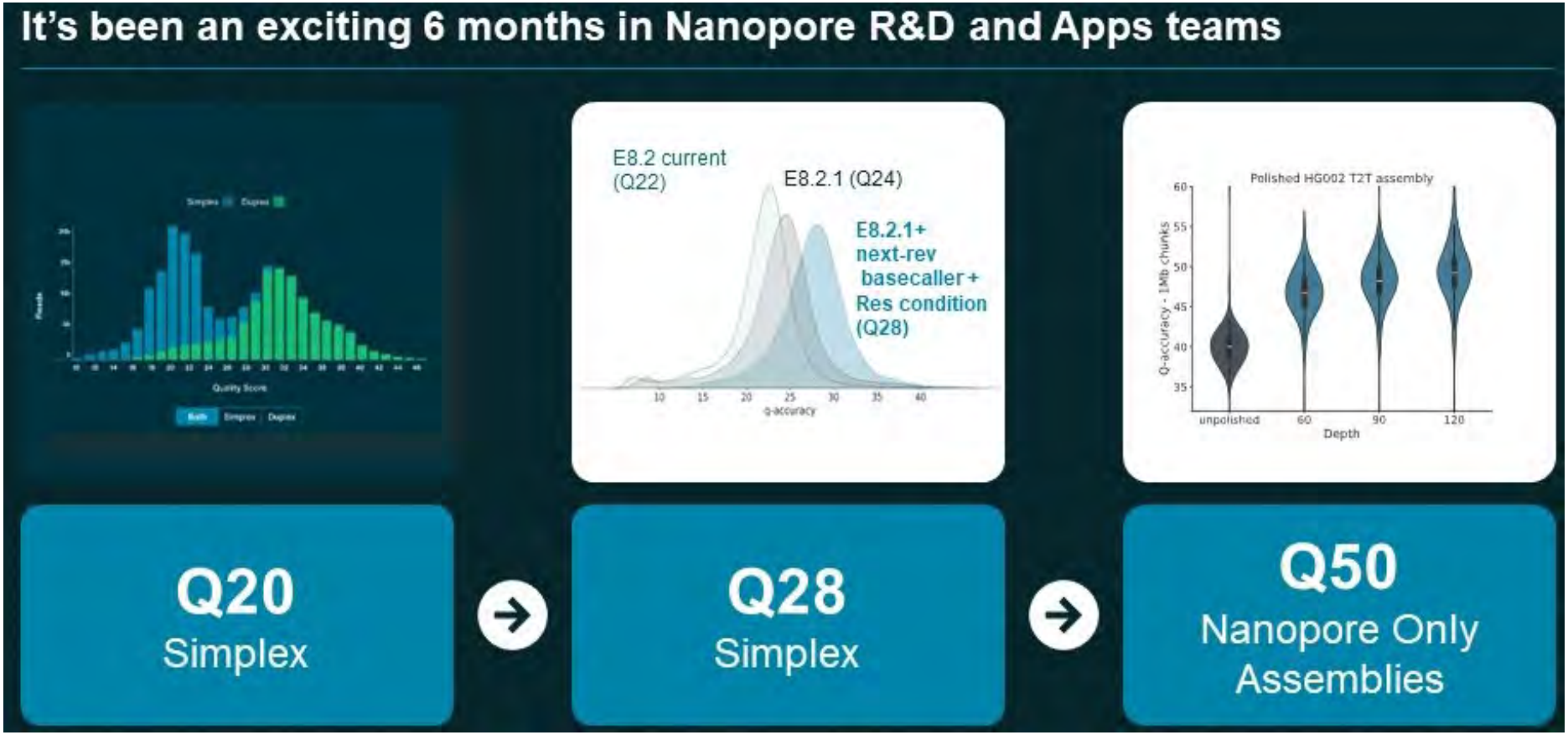
Systematic errors

Polishing with Illumina data can improve the final error rate

- Basecalling progress: Guppy years



- Basecalling progress: Dorado years



Replication outside nanopore HQ

Latest post from Ryan Wick's bioinformatics blog (`rrwick.github.io/`) report Q20 reads accuracy and Q60 assemblies on 9 bacterial assemblies

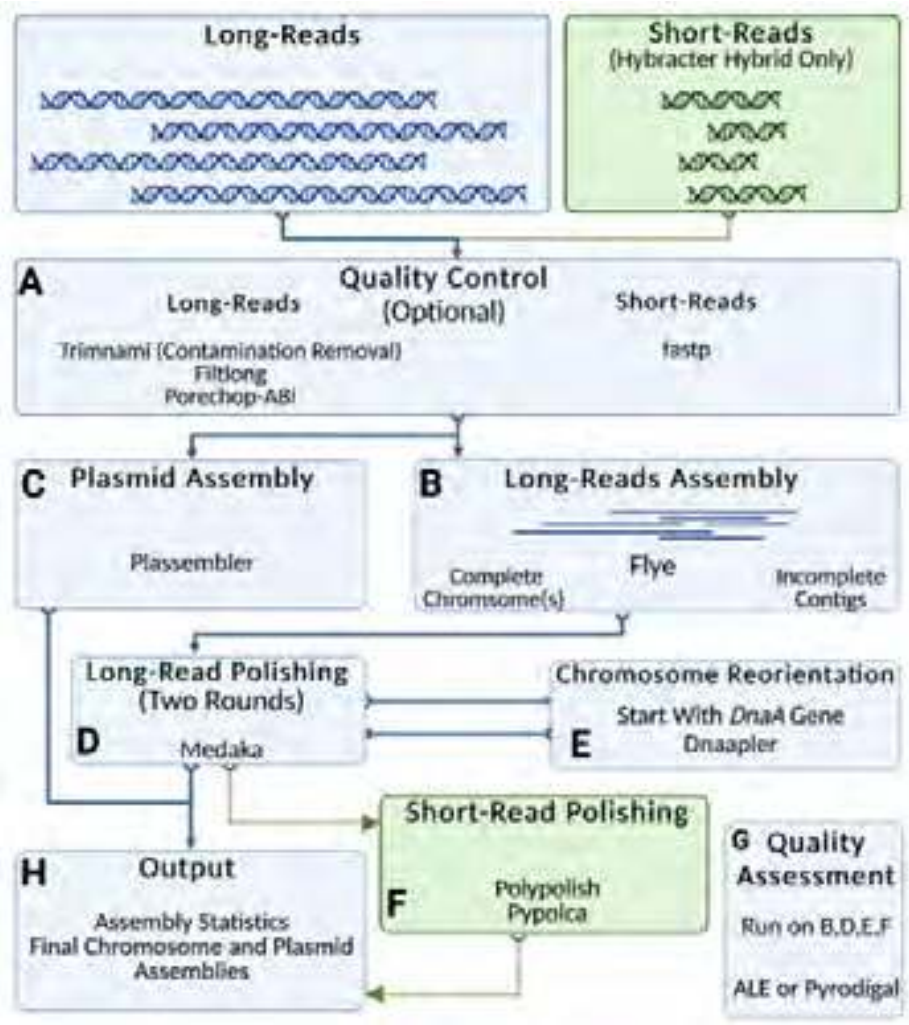
Average	**Read accuracy**	**Assembly accuracy**
mean	97.7%, Q16.4	4 errors, Q60.43
median	99.1%, Q20.5	2 errors, Q60.43
mode	99.4%, Q22.2	NA

HiFi-like Nanopore data ?

(Near) error-less very long reads we have several promising improvements ahead:

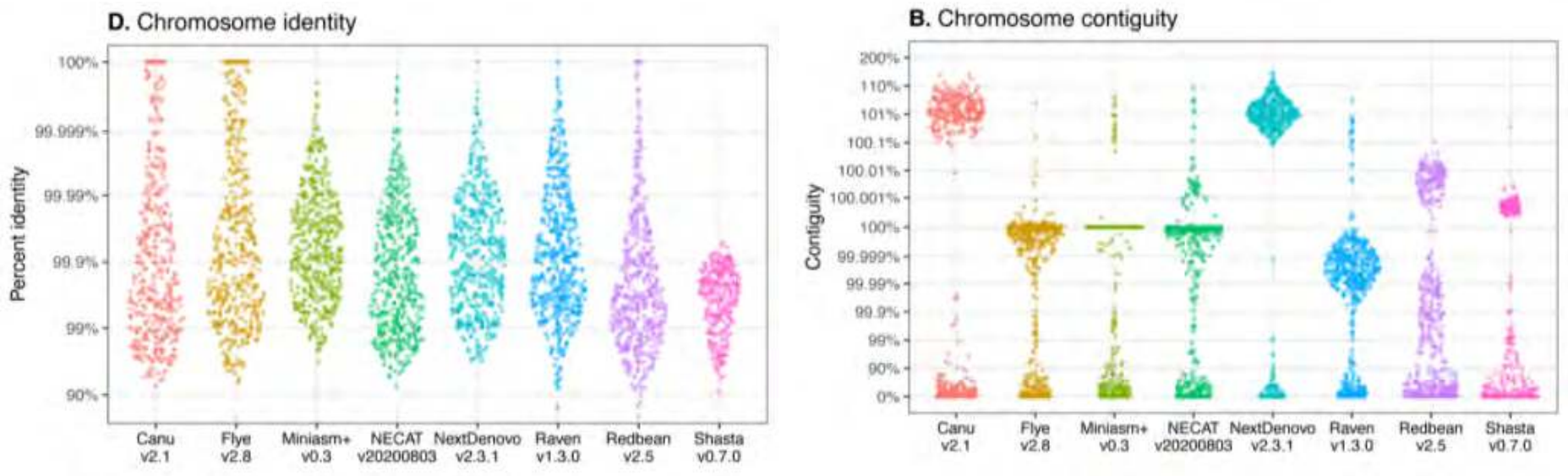
- Very fast assembly
- T2T chromosomes with less data
- Higher consensus accuracy
- Poplyploid assemblies

Challenge 4 : Assembly as a software



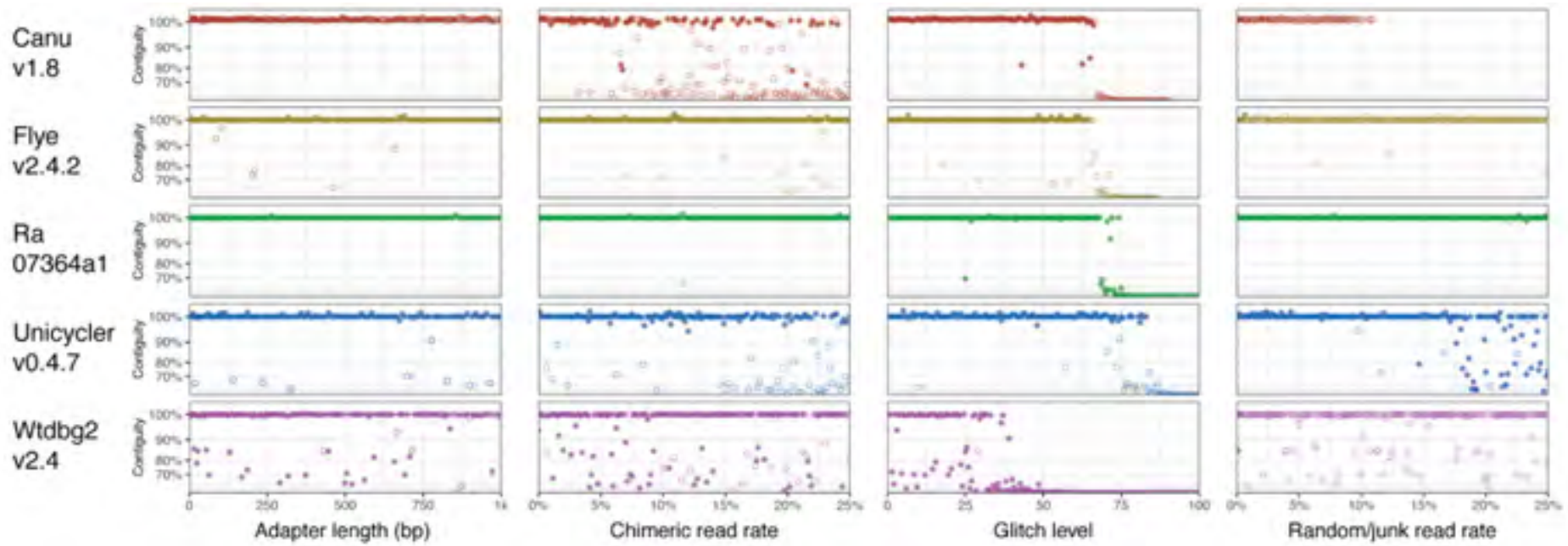
From Hybracter: Enabling Scalable, Automated, Complete and Accurate Bacterial Genome Assemblies

Assemblers behave differently



From `github.com/rrwick/Long-read-assembler-comparison`

- Software robustness



From `github.com/rrwick/Long-read-assembler-comparison`

- An assembly is a model

1. Assemblies contain errors
2. Different tools can produce very similar assemblies
3. A single tool can produce very different assemblies with small changes of parameters(!)

The (first) end

Advanced points

If we have time, we'll review everything (while doing this course, I doubt it . . .)
Else, pick one:

- Multiple k assembly in de Bruijn graphs
- HiFi de Bruijn Assembly
- An overlap graph limitation with noisy long reads (and current fixes)
- The repeat graph

• Coming back to a de Bruijn graph limitation: fixed overlaps

GGACT and ACTTA overlap is only of size 3 !

- A too small k is not a solution
- We would like larger k's but miss connections

- Multiple k assembly

Most de Bruijn graph assemblers can now perform several assemblies with different k-mer sizes to produce an improved super assembly

Build DBG with k=5 and k=7 from those reads
AAAATCGATCTC
TCTCATCGAATT

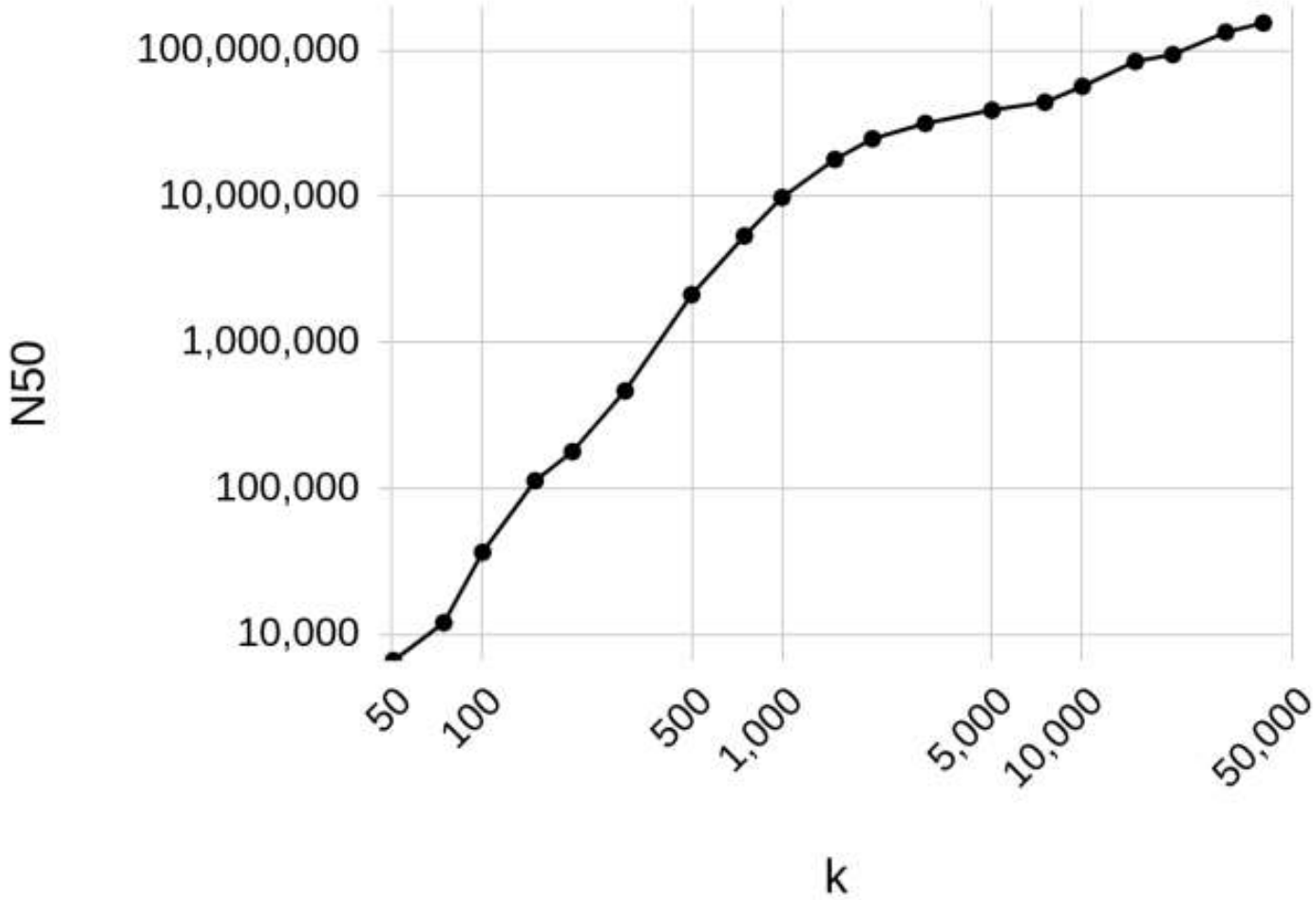
- Multiple k assembly

We are missing GATCTCA and ATCTCAT in the second graph.
But they are present in the first graph!

- Multiple k assembly

HiFi de Bruijn graph Assembly

Using very large K (K=500 to K=5000) de Bruijn graphs to assemble



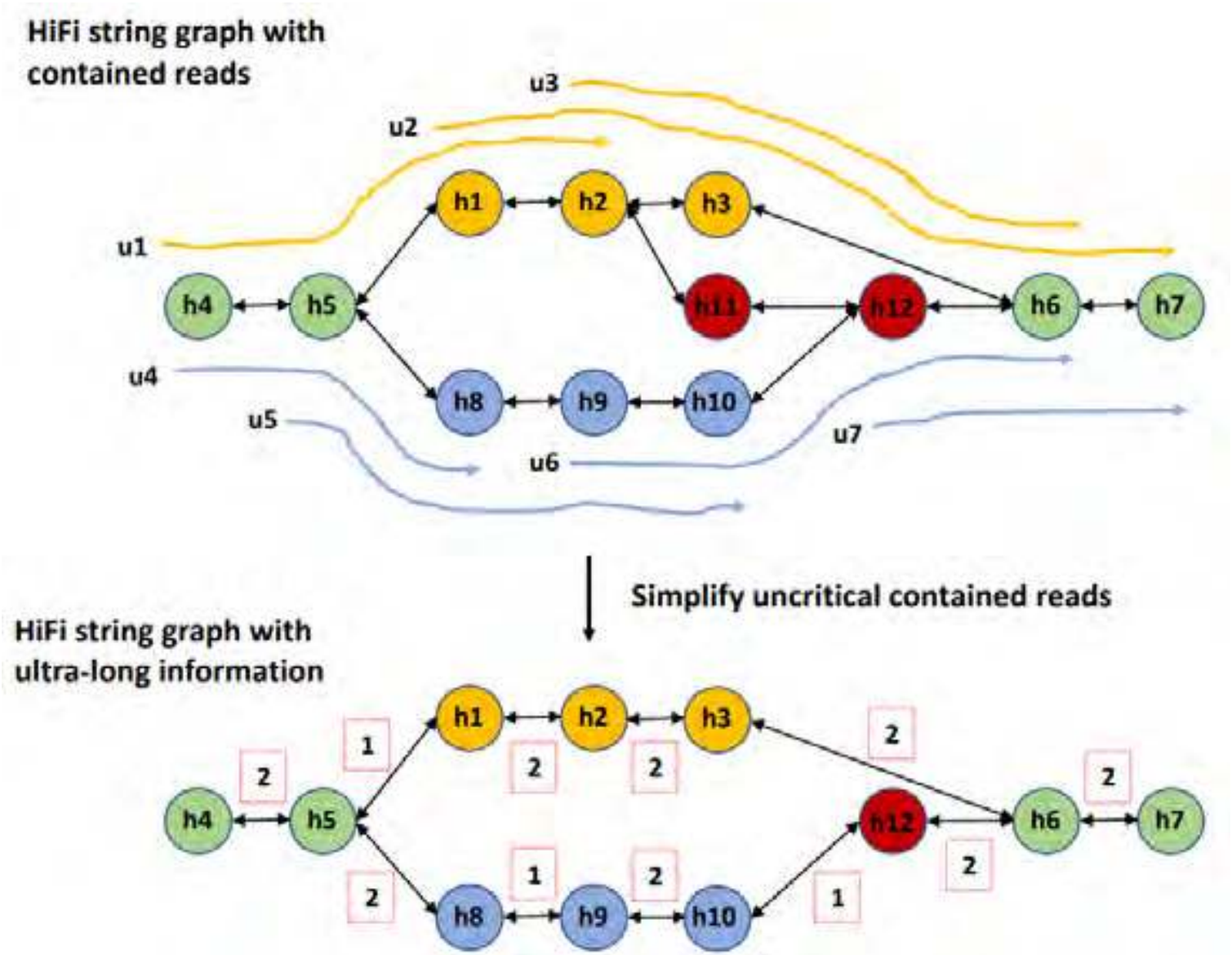
- Coming back to the overlap graph simplifications

- An overlap graph limitation when using noisy reads

- An overlap graph limitation when using noisy reads

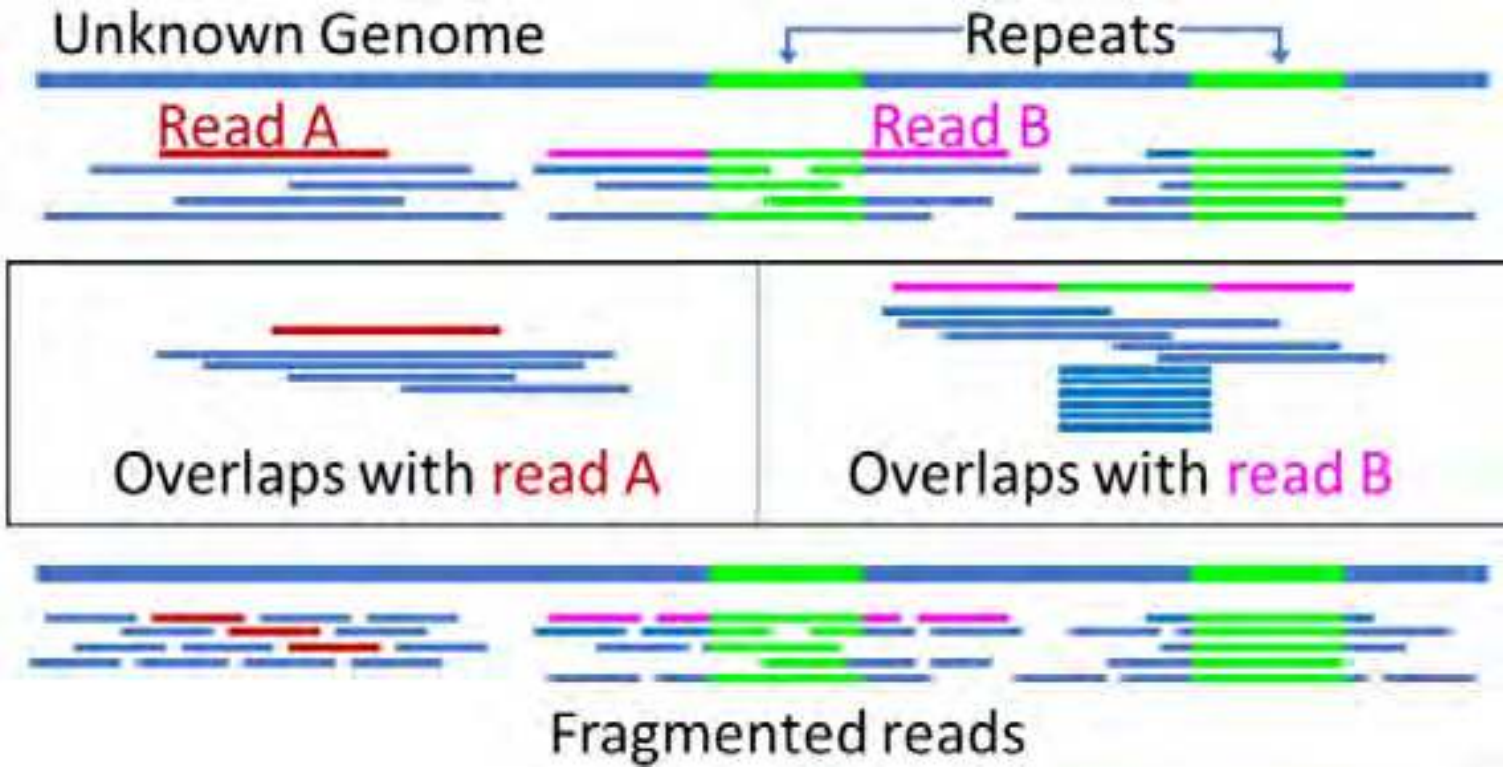
- An overlap graph limitation when using noisy reads

Read threading alternative

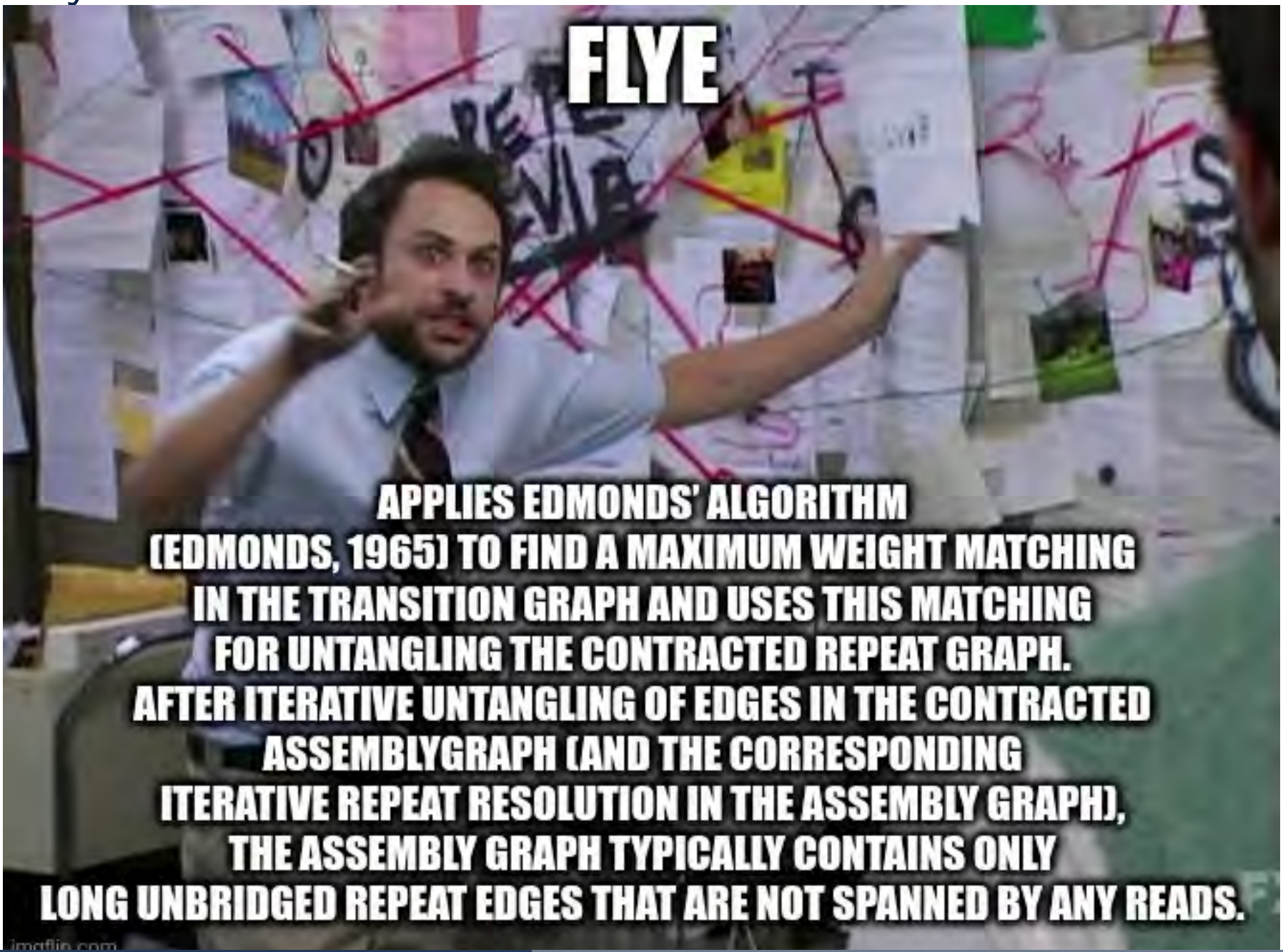


Fragmented read alternative

The RAFT tool fragments the reads that does not cover repeats to avoid read inclusion problems.



- Flye


FLYE
APPLIES EDMONDS' ALGORITHM
(EDMONDS, 1965) TO FIND A MAXIMUM WEIGHT MATCHING
IN THE TRANSITION GRAPH AND USES THIS MATCHING
FOR UNTANGLING THE CONTRACTED REPEAT GRAPH.
AFTER ITERATIVE UNTANGLING OF EDGES IN THE CONTRACTED
ASSEMBLYGRAPH (AND THE CORRESPONDING
ITERATIVE REPEAT RESOLUTION IN THE ASSEMBLY GRAPH),
THE ASSEMBLY GRAPH TYPICALLY CONTAINS ONLY
LONG UNBRIDGED REPEAT EDGES THAT ARE NOT SPANNED BY ANY READS.

- Repeat graph

- Repeat graph

- Repeat graph

- Repeat graph

The end (Thank you for your attention)

Slides for the practical session

Evaluate assembly

Two cases:

Reference based

Align contigs to the reference and compare them considering the reference as the ground truth (QUAST).

Reference free

- Reads analysis (QUAST)
- Kmer analysis (Merqury)
- Assembly graph analysis (Bandage)

QUAST statistics

Alignment-based statistics	ABySS	MEGAHIT	SPAdes	Velvet
Genome fraction (%)	98.661	98.424	98.113	97.997
Duplication ratio	1.043	1	1	1
# genomic features	4525 + 75 part	4511 + 64 part	4489 + 50 part	4486 + 56 part
Largest alignment	248 481	235 933	285 096	264 944
Total aligned length	4 776 214	4 568 317	4 553 809	4 550 150
NGA50	69 801	122 647	133 309	112 446
LGA50	21	14	12	14
Misassemblies				
# misassemblies	4	0	0	4
Misassembled contigs length	231 767	0	0	435 515
Per base quality				
# mismatches per 100 kbp	2.09	2.69	1.03	3.19
# indels per 100 kbp	0.57	1.31	0.29	1.98
# N's per 100 kbp	24.59	0	17.55	94.19
Statistics without reference				
# contigs	176	95	92	90
Largest contig	248 481	235 933	285 196	264 944
Total length	4 777 853	4 571 292	4 557 363	4 552 266
Total length (>= 1000 bp)	4 757 929	4 562 458	4 548 710	4 544 453
Total length (>= 10000 bp)	4 562 801	4 478 614	4 466 223	4 475 223
Total length (>= 50000 bp)	3 248 113	3 833 793	3 812 315	3 817 904
BUSCO completeness				
Complete BUSCO (%)	98.65	98.65	98.65	98.65
Partial BUSCO (%)	0	0	0	0
Predicted genes				
# predicted genes (unique)	3717	3595	3587	3576

- Assembly continuity

N50

N50 can be described as a weighted median statistic such that 50% of the entire assembly is contained in contigs or scaffolds equal to or larger than this value.

- The catsembler

- Assembly continuity

N50

N50 can be described as a weighted median statistic such that 50% of the entire assembly is contained in contigs or scaffolds equal to or larger than this value.

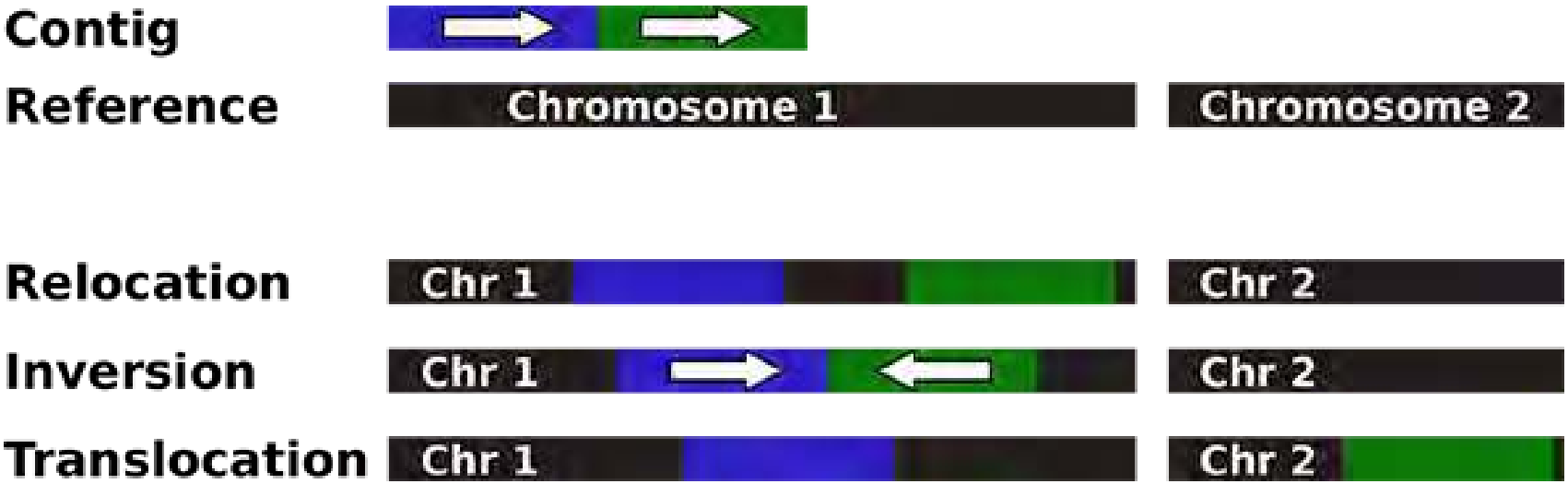
N75

N75 is the same statistic for 75% of the assembly

NGA50

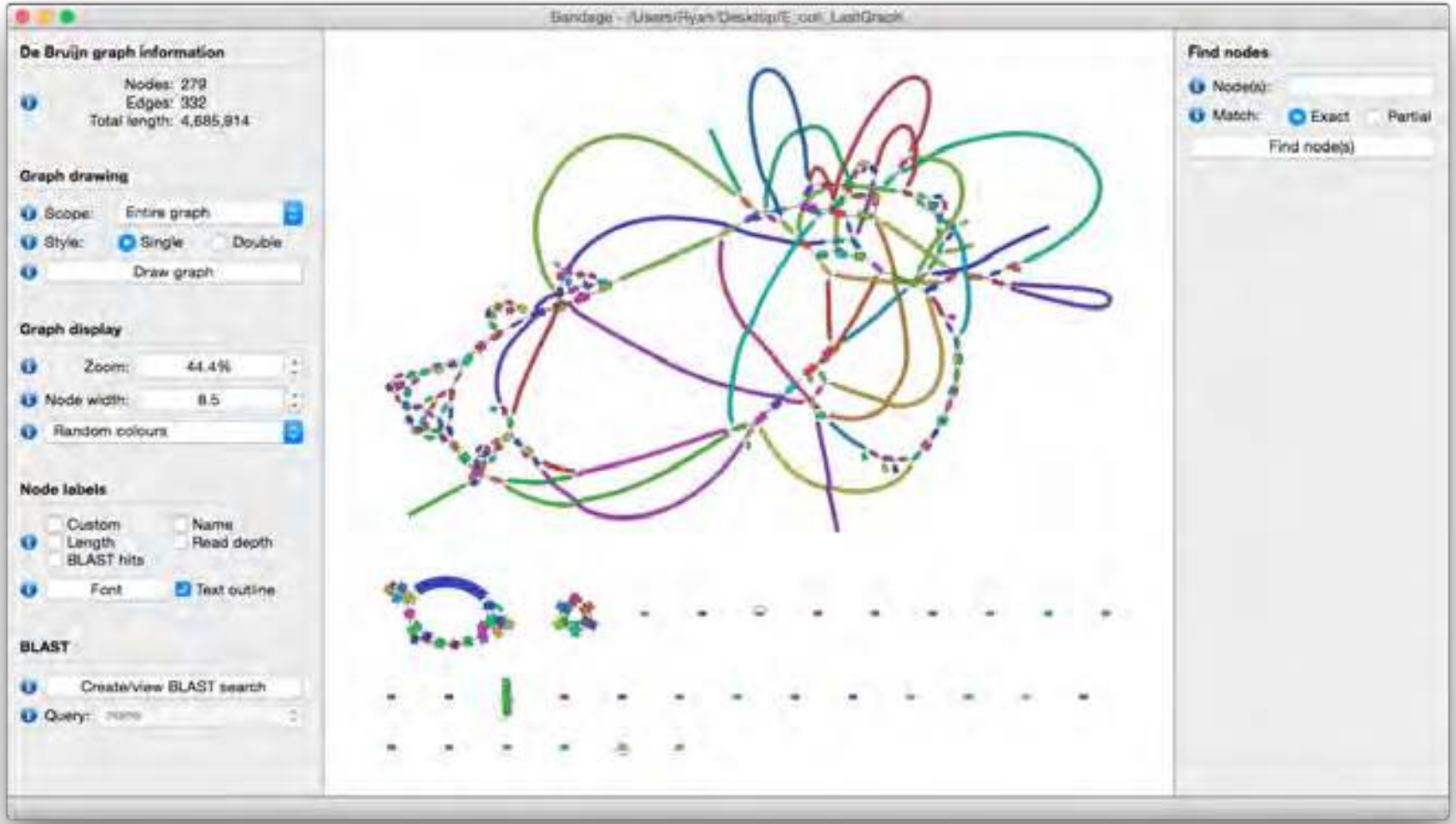
Similar to the N50 but only takes into account contigs/scaffolds that can be **aligned** on the reference genome and consider 50% of the **genome size** instead of the assembly size

- Misassemblies



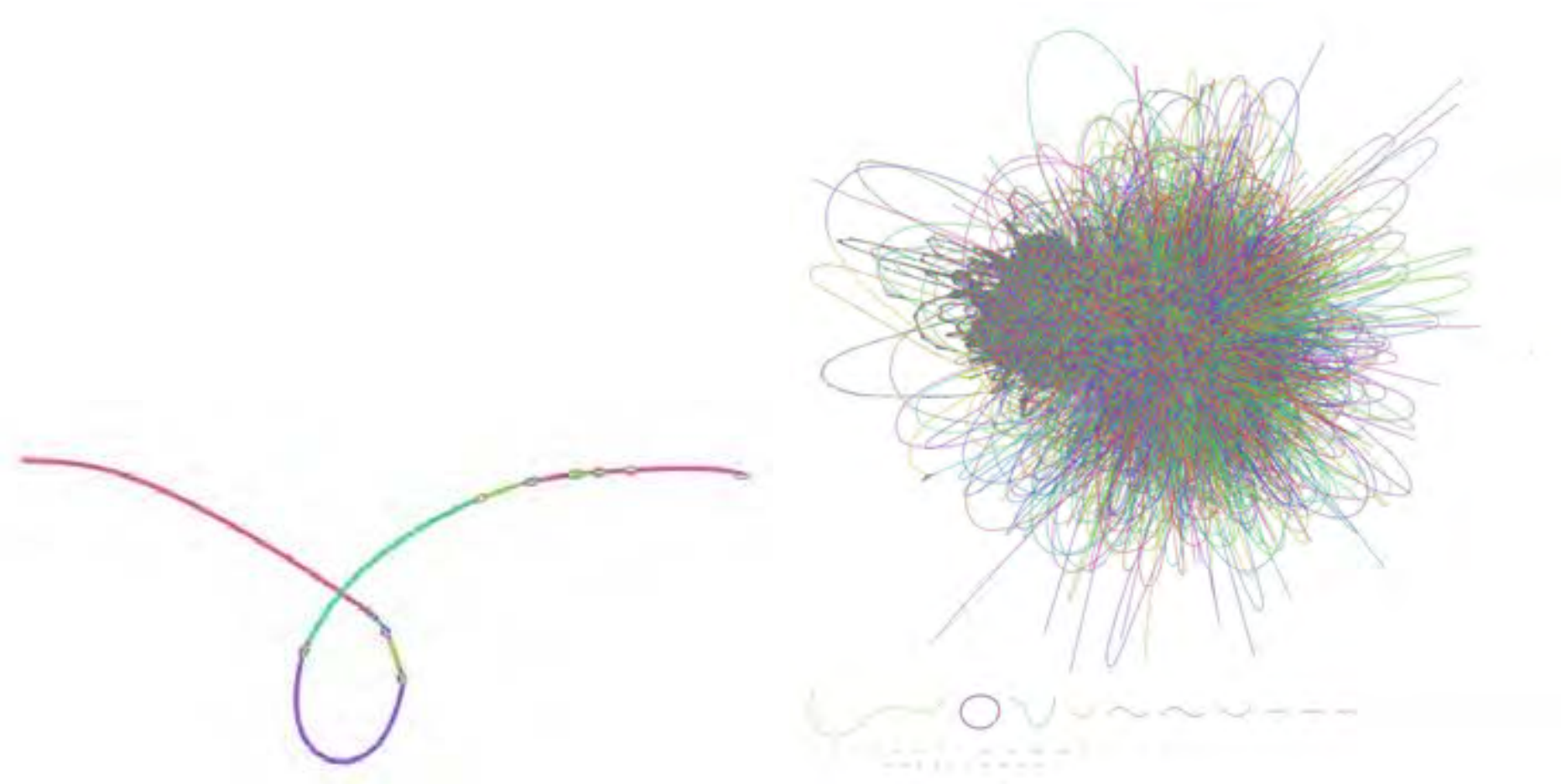
• Visualize assembly

Bandage tool can visualize assembly graphs (GFA)

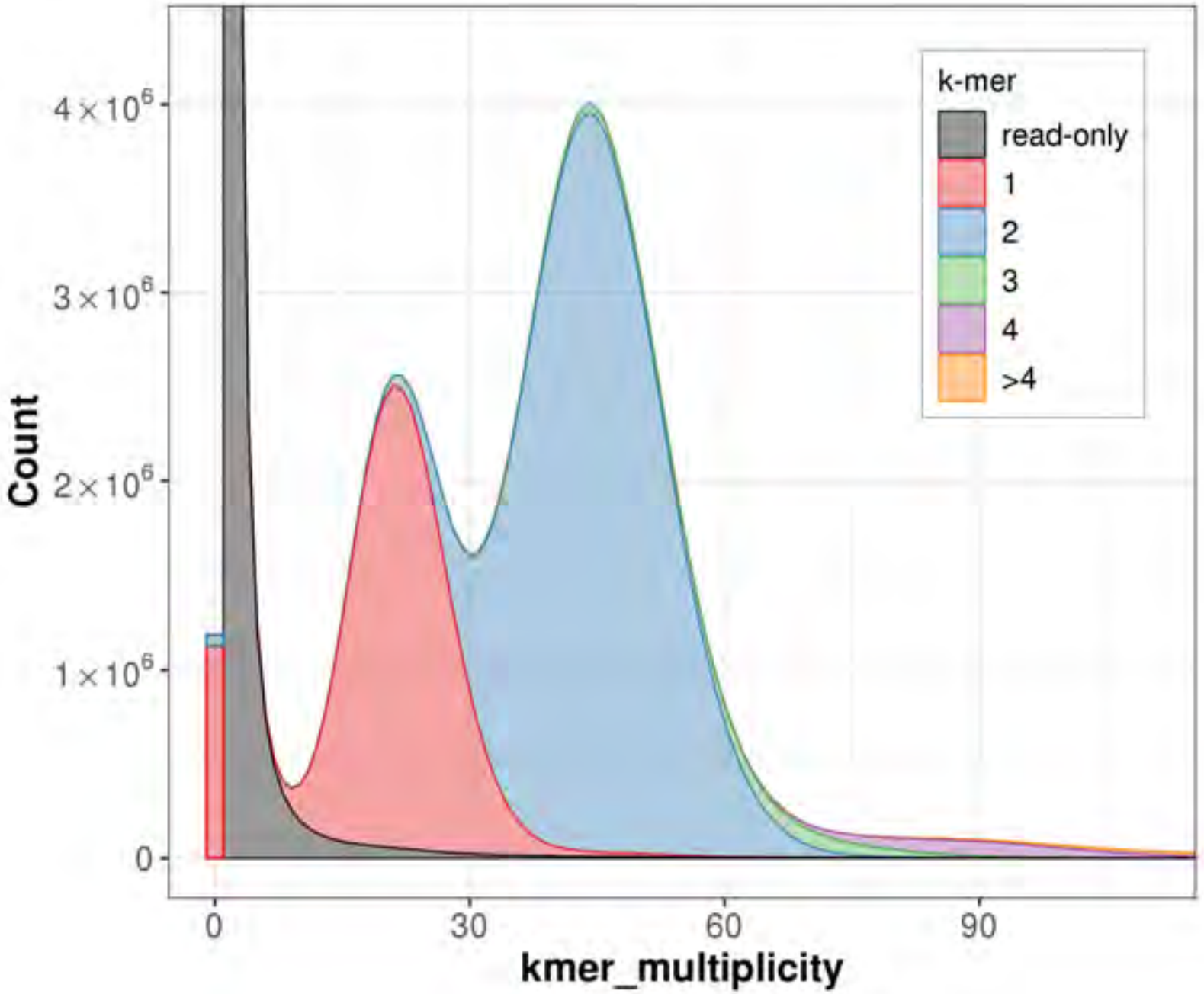


• Visualize assembly

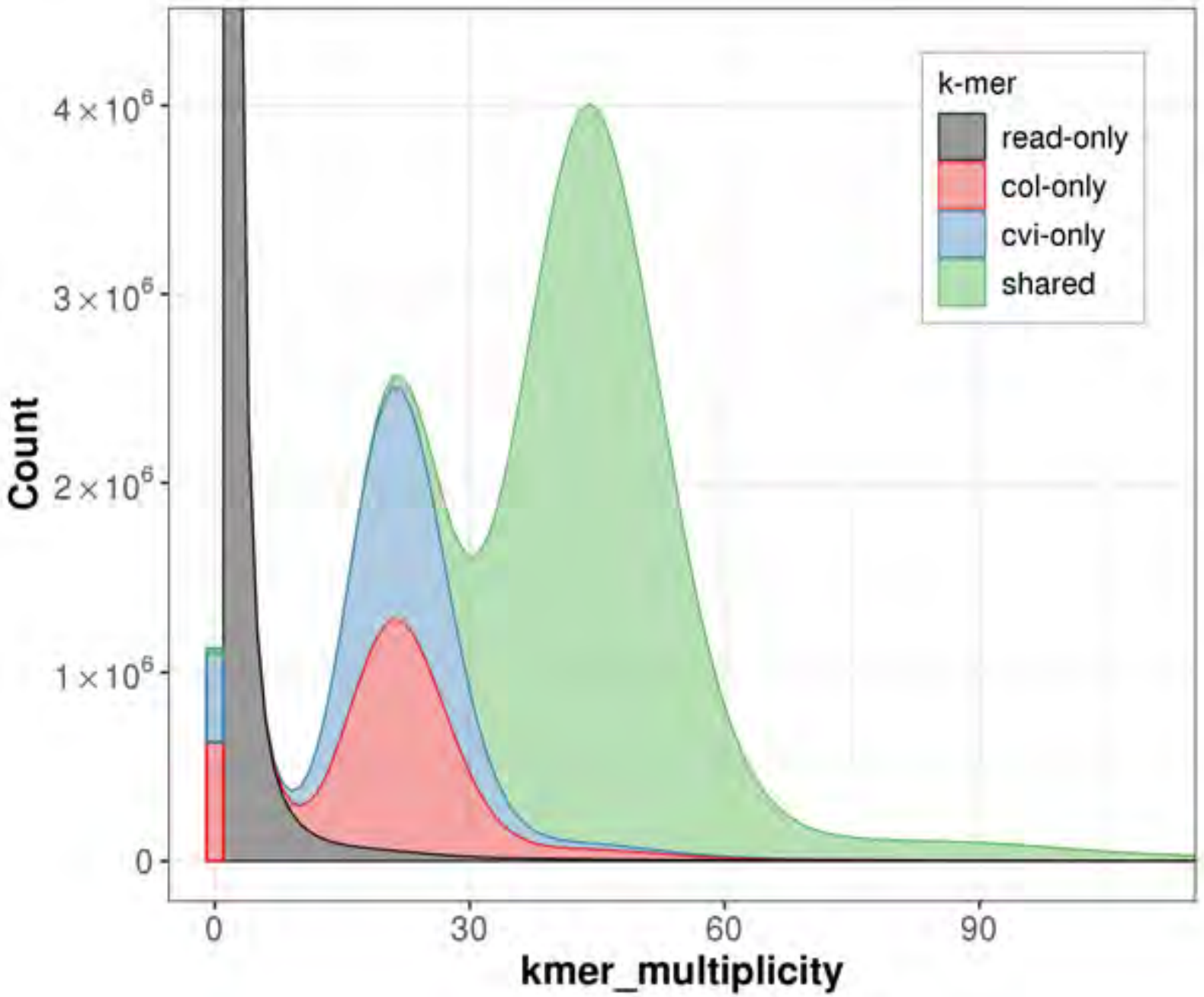
Bandage tool can visualize assembly graphs (GFA)



- *K*-mer spectrum visualization with merqury



- Trio *K*-mer spectrum visualization with KAT



SPAdes assembler

- Designed to assemble megabase-sized genomes
- Multiple k de Bruijn graph assembly from short reads
- Can use long reads to solve repeats

Short reads

Long reads

Hifiasm assembler

- Build an overlap graph from HiFi reads
- Generate both haploid and diploid assemblies
- Can use (very) long reads to solve repeats

Mandatory

HiFi reads

Optional

Long reads

Flye assembler

- Build a repeat graph from long reads
- Can use any kind of long reads
- Can also assemble metagenomes

HiFi/Long reads

HiFi/Long reads

Unicycler (long read mode)

- Build a overlap graph from long reads
- Polish the assembly
- Also has a short-reads-first similar to SPAdes