A little tour of assembly methods

Camille Marchet & Antoine Limasset

Université de Lille, CNRS, France

EVOMICS Workshop on Genomics 2020 - Český Krumlov

Content of this course

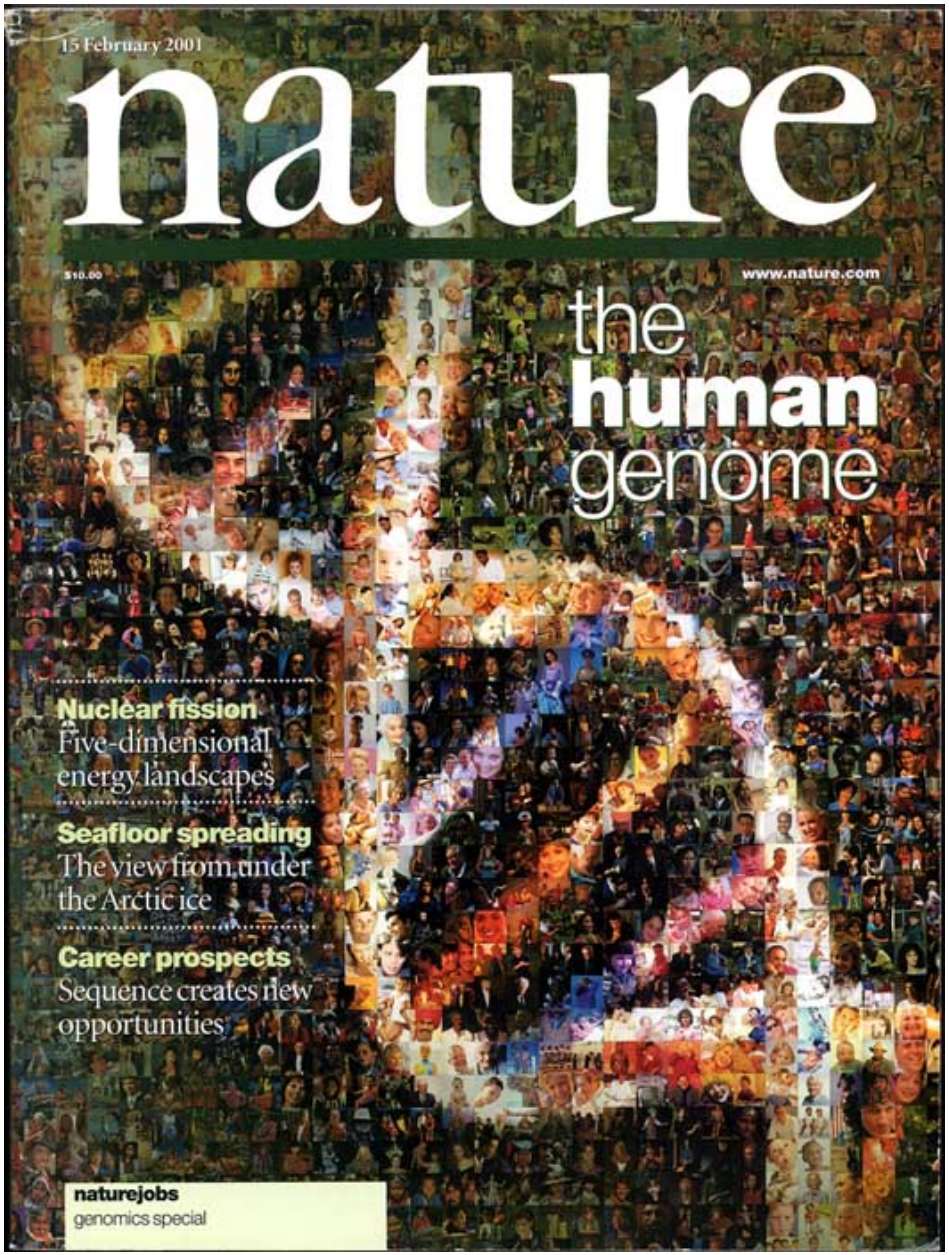
- How to reconstruct a genome with sequencing data?
- What are the main challenges?
- Which solutions have been proposed?

genome size: $\sim$ 32 gigabases

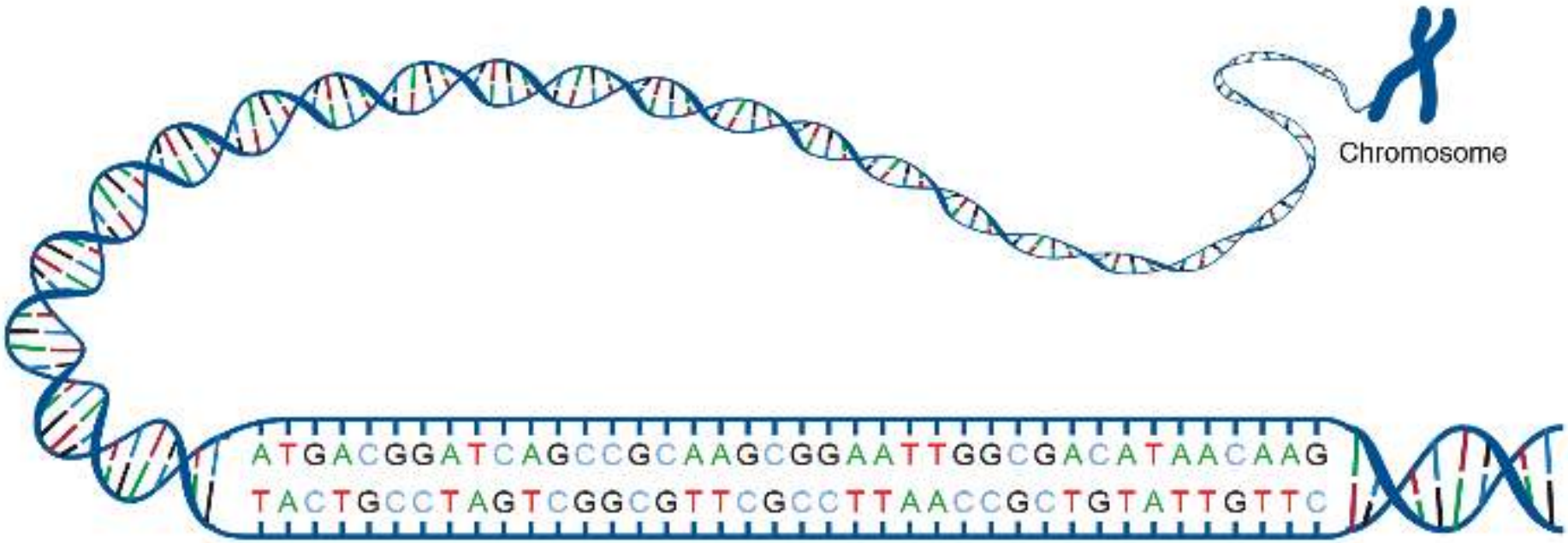
Content of this course

- What is assembly?
- What are the main problematics to bear in mind when conducting an assembly?
- What are the current limits of assembly?

Assembly


15 February 2001
nature
$10.00
www.nature.com
the human genome
Nuclear fission
Five-dimensional energy landscapes
Seafloor spreading
The view from under the Arctic ice
Career prospects
Sequence creates new opportunities
naturejobs
genomics special

Accessing a genome



From `www.genome.gov/genetics-glossary/acgt`

Reads are words from the genome

Reads are **shuffled** words from the genome

Genome assembly task

Using read overlaps

Genome sequencing: coverage

Genome sequencing: coverage

Genome sequencing: coverage

Genome sequencing: coverage

Genome sequencing: coverage

Assembly **idea number 1: Select the longest overlaps**

Assembly idea number 1: assemble the longest overlaps

Your time to shine!

The Greedy solution

The actual solution

What happened?

Do we expect many repeats?

Probability to have NO repeated word of size 31 in a 5 megabases genome

Input interpretation:

$$\left(\frac{4^{31}-1}{4^{31}}\right)^{1/2\,(5\times 10^6\,(5\times 10^6-1))}$$

Decimal approximation:

0.999997289498784302383172055421363836712023171938932024106...

From en.wikipedia.org/wiki/Birthday_problem

The burden of assembly: genomic repeats

Amount of repeats larger than a given size in *E. coli* genome

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

Genomic repeats are NOT random events

Greedy assemblers

- Simple and efficient scheme
- Rely on **local** best choice (greedy)
- May create errors because of local choices when there are repeats

History from the last century

Assembly **idea number 2: consider all overlaps**

Greedy solution

One piece solution

Multiple repeats

First solution

Second solution

Parsimonious solution: do not assemble

Repeats lead to the fragmentation of the assembly

Missing information also fragments the assembly

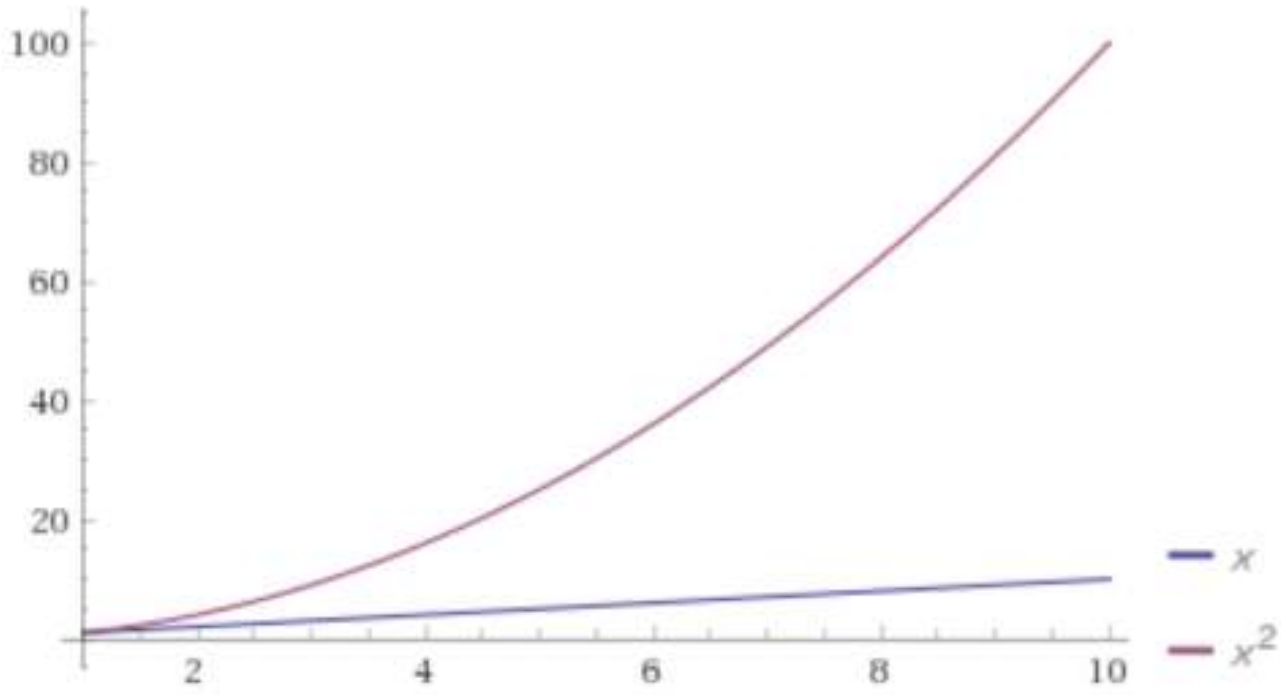
Assembly **concession number 1: output fragments**

In the real world, assemblers often provide pieces of genomes rather than complete ones

Overlap graph prerequisite : all overlaps

Overlap graph burden: number of reads

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



Linear: 2X data 2X time
Quadratic: 2X data 4X time

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...

Most overlaps are too small to be considered...
The overlap computation is not linear

Overlap graphs in a nutshell

- Graphs of overlaps between the reads
- Can provide a global solution for assembly
- Can be difficult in real cases because it requires a lot of computation (overlaps)

S. cerevisiae, *D. melanogaster*, human could be assembled using overlap graphs approaches (Celera (Myers et al. 2000), SGA (Simpson & Durbin 2011), ...)

Fast forward

Assembly **idea number 3: Focus on genome words**

k-mer definition

A k-mer is a word of size k

Exercise

List all **distinct** 4-mers and 5-mers of this set of reads:
CATAATGCA
TGCACATAA
CATAATGCA

Solution

CATAATGCA, TGCA**CATAA**, **CATAATGCA**

5-mers:

CATAA, ATAAT, TAATC, AATGC, ATGCA, TGCAC, GCACA, ACATA

CATAATGCA, **TGCA**CATAA, **CATAATGCA**

4-mers:

CATA, ATAA, TAAT, AATG, ATGC, TGCA, GCAC, CACA, ACAT

Genome words / read words

Genome words / read words

Reconstitute larger genomic words

The De Bruijn graph

De Bruijn graph assembly

De Bruijn graph time!

Use 7-mers

Solution

De Bruijn graph versus overlap graph

De Bruijn graphs abstract redundancy

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

De Bruijn graphs limitation

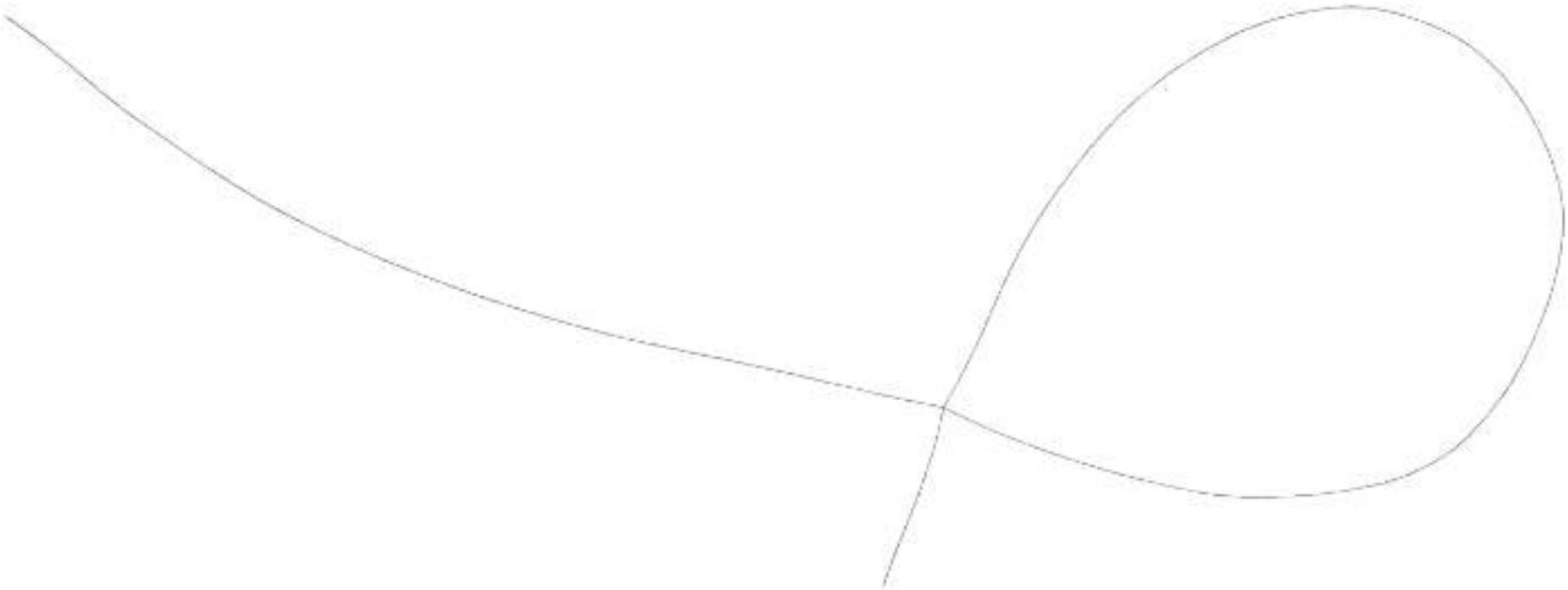
Fixed overlaps

De Bruijn graphs limitation

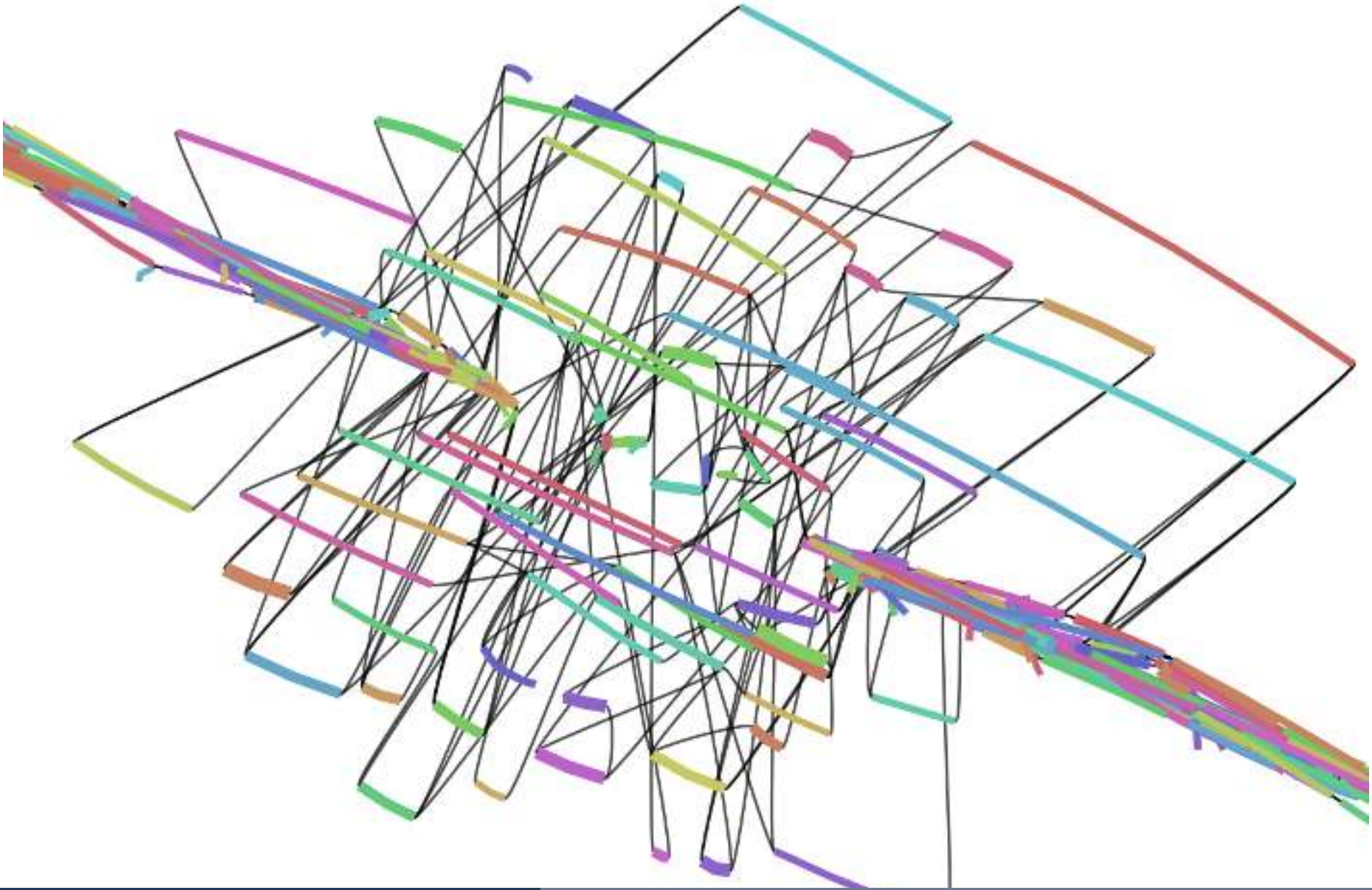
Repeats...

De Bruijn graph limitation

De Bruijn graph on a real dataset

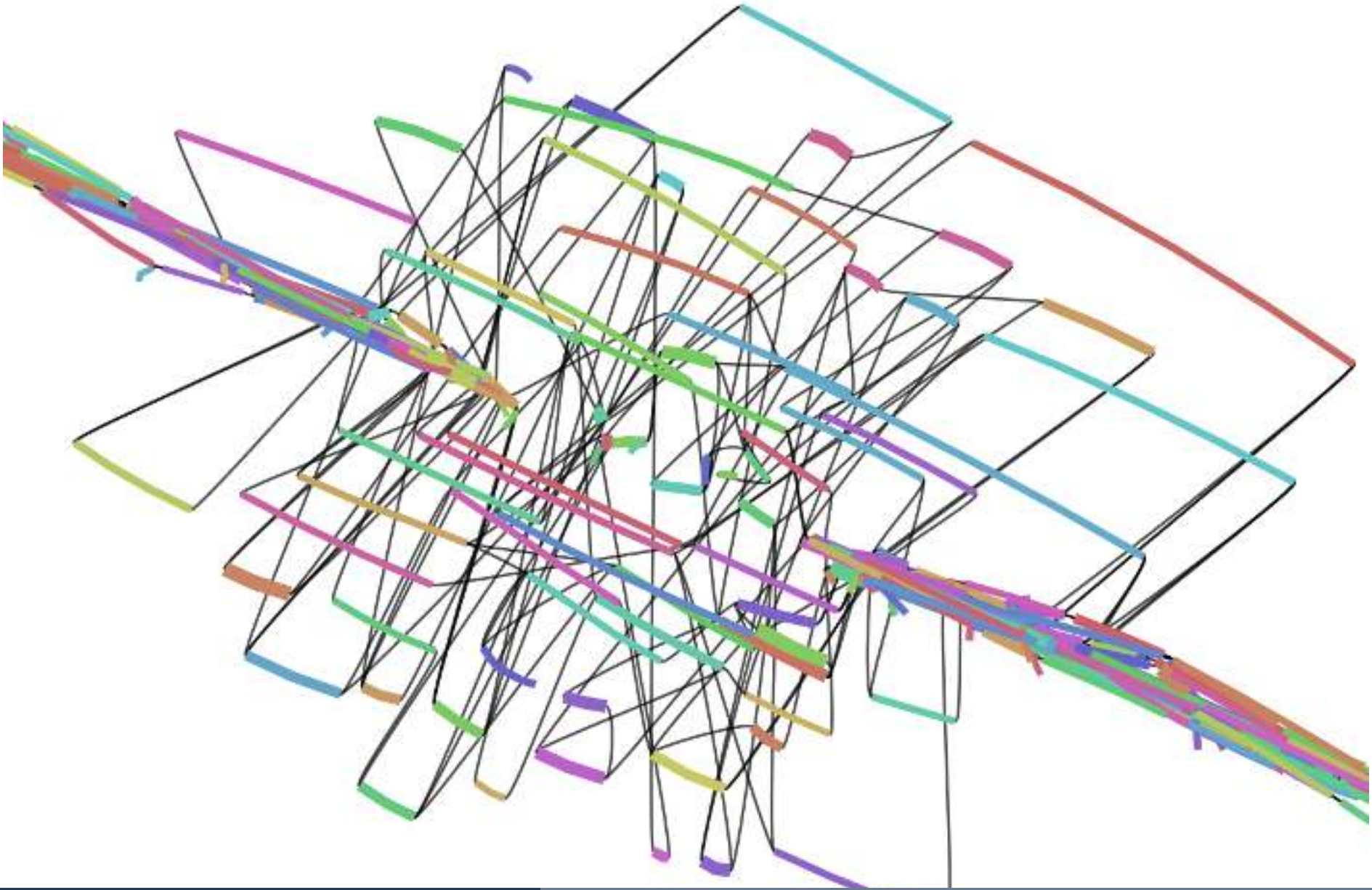


De Bruijn graph on real dataset ZOOMED IN



On the representation of De Bruijn graphs

De Bruijn graph on a real dataset ZOOMED IN

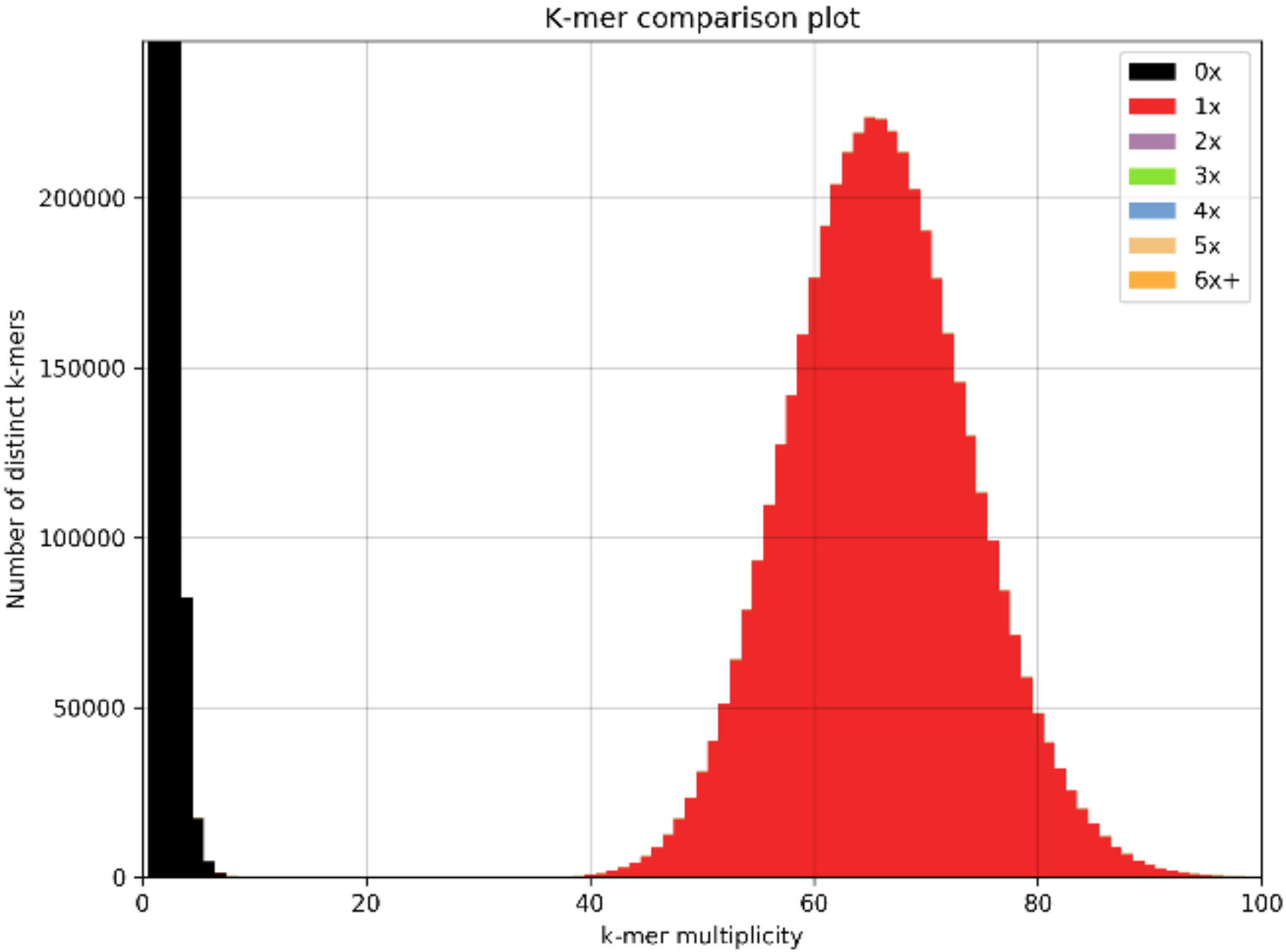


Dealing with sequencing errors

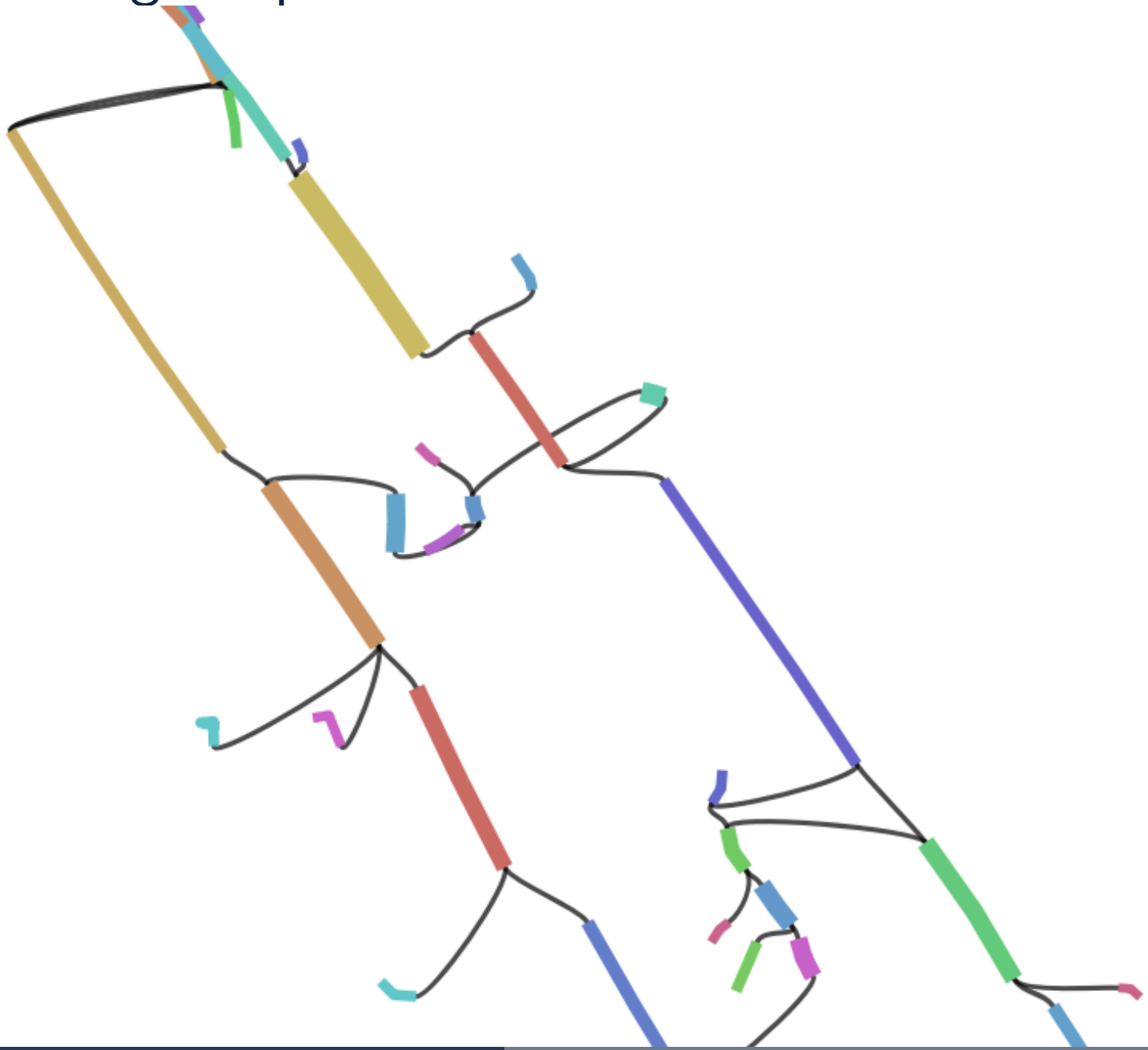
Erroneous k-mers vs genomic k-mers

Erroneous k-mers are seen less than genomic ones

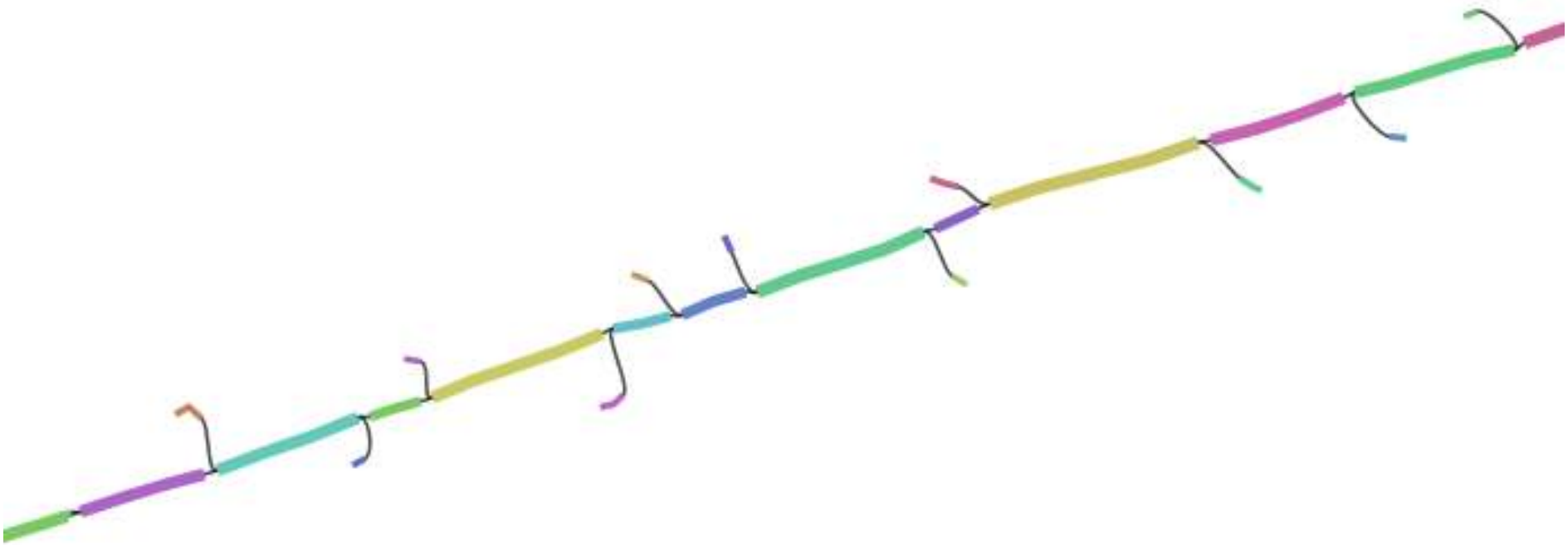
K-mer histogram



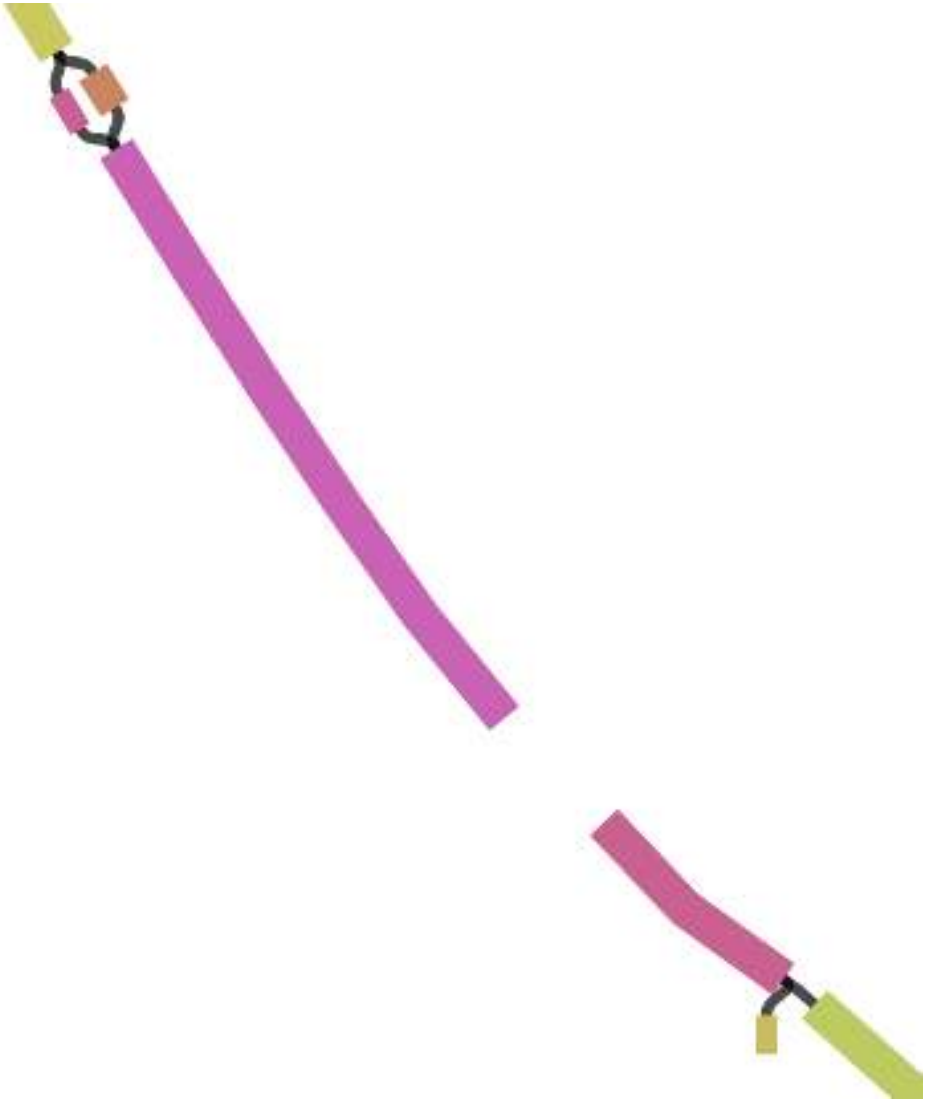
Removing unique k-mers



Removing k-mers seen less than 3 times



Removing k-mers seen less than 4 times

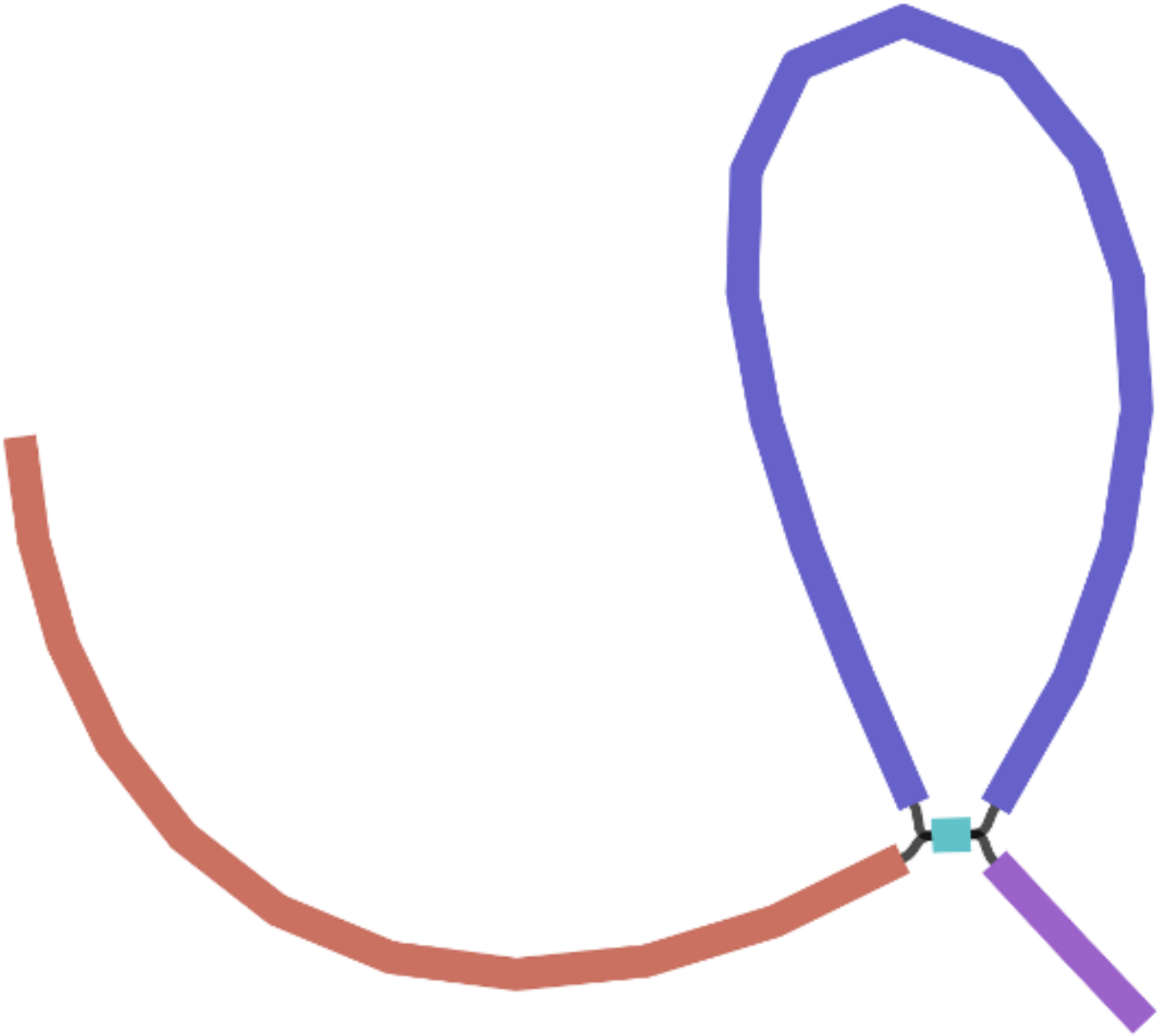


Errors in De Bruijn graphs

Errors in De Bruijn graphs

Errors in De Bruijn graphs

(Almost assembled phage !)



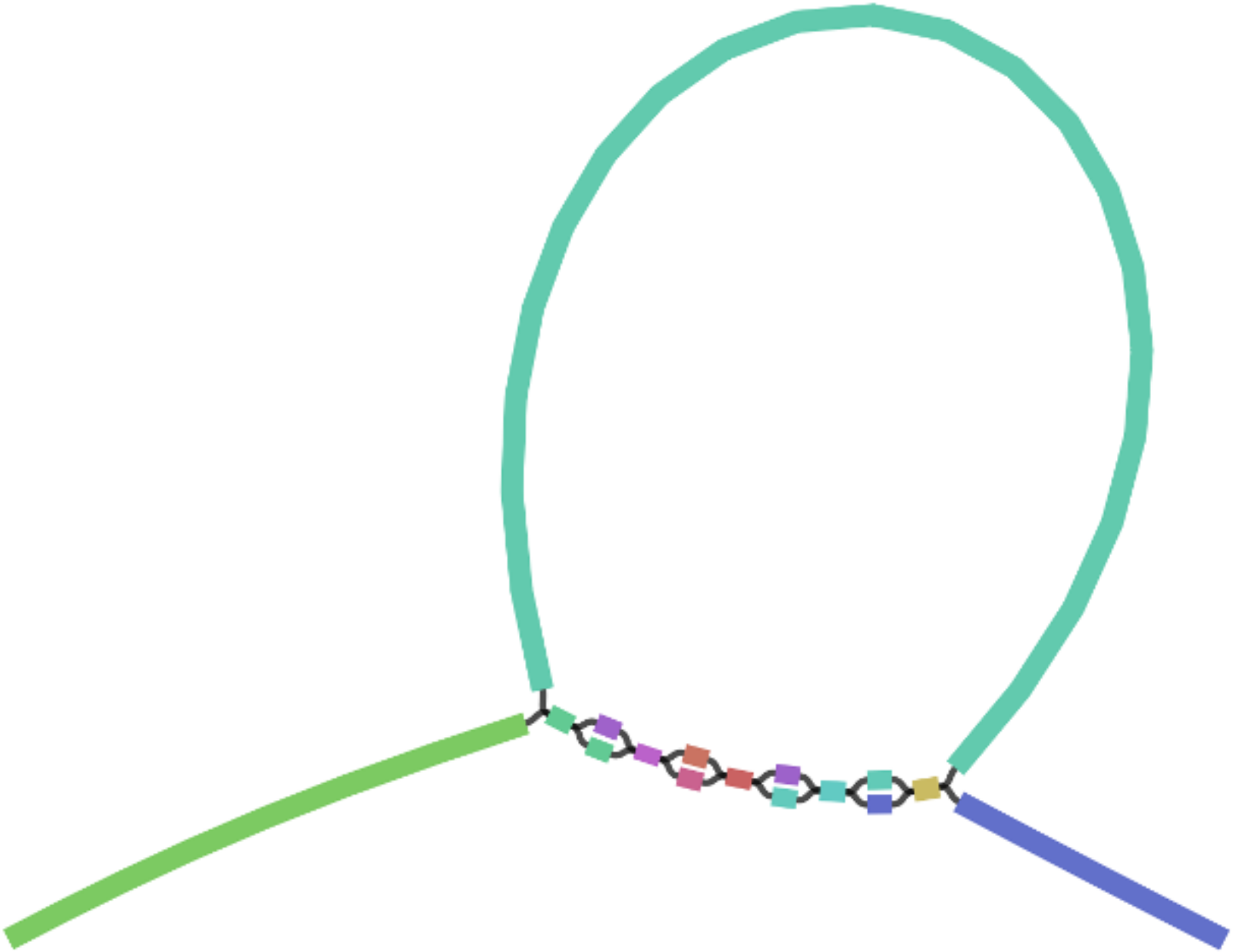
De Bruijn graph on an eukaryota

Two or more genomes per individual

Two or more genomes per individual

Assembly **concession number 2**: collapse variability

Paralog genes/repeats



Paralog genes/repeats in graph

An assembler is a set of heuristics

Graph cleaning heuristics

- Nodes coverage
- Graph local/global topology
- Reads that can be mapped on nodes
- Estimated coverage/genome size
- ...

An assembly is a model

Different tools can produce very similar assemblies

A single tool can produce very different assemblies with small changes of parameters(!)

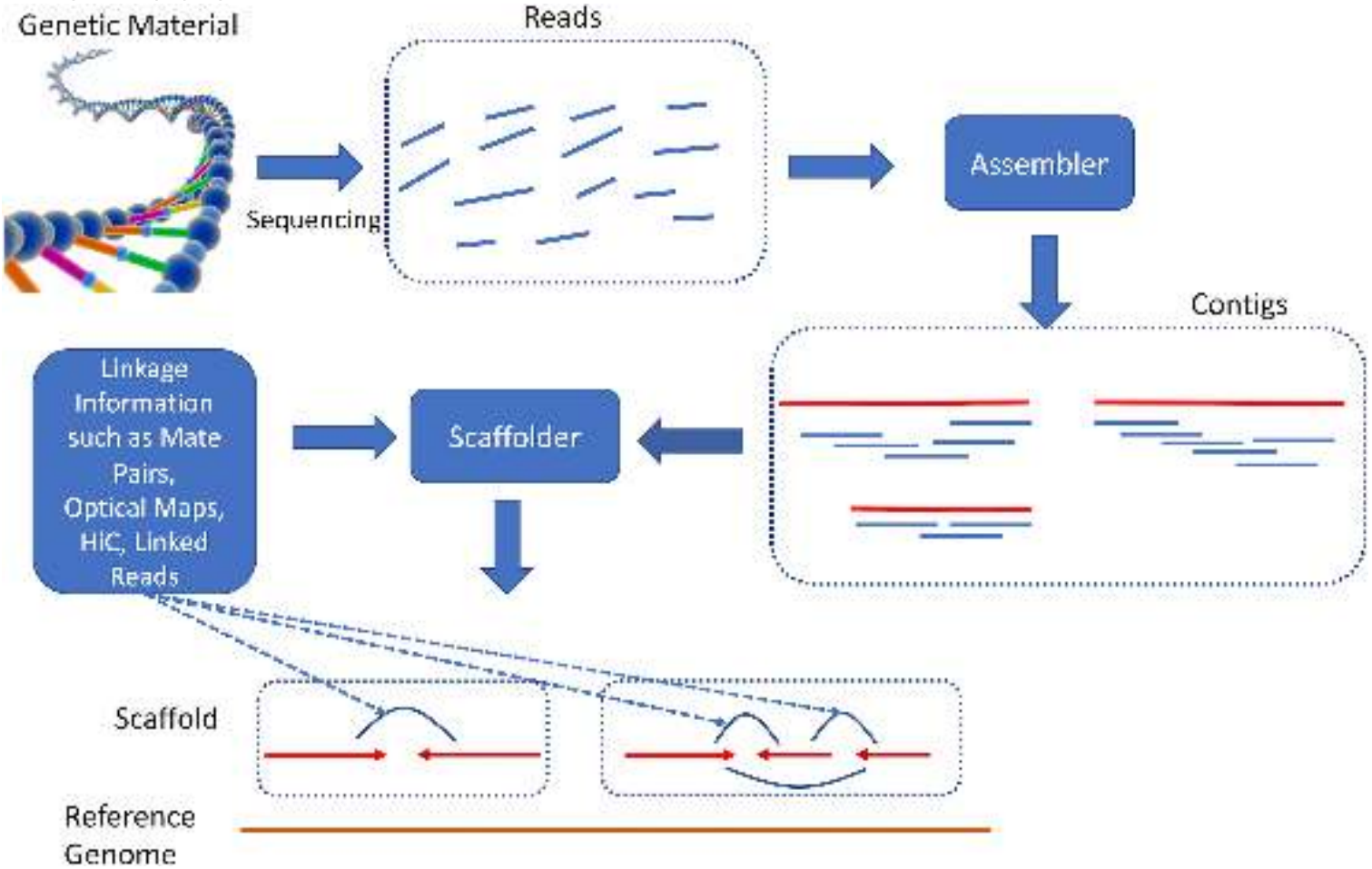
De Bruijn graphs in a nutshell

- Graph of words of size k, k-1 overlaps
- Collapses identical k-mers
- Very successful, have replaced the overlap graphs with high throughput sequencing data
- Still outputs fragments of the genome

white spruce, 20 gigabases

Scaffolding

Softwares can improve the assembly continuity by using other kinds of information



From "Modern technologies and algorithms for scaffolding assembled genomes" Plos Computational Biology

Second generation sequencing

NextSeq Series

HiSeq 4000 System

HiSeq X Series‡

NovaSeq 6000 System

- Short reads $\approx 150bp$
- Low error rate $\approx 1\%$
- High throughput (up to billion of reads per run)
- GC bias

Mainly assembled using De Bruijn graphs

State-of-the-art

Your toolkit for the practical session

- SPAdes
- Megahit
- Minia

Other notable assemblers

- SGA
- Discovar denovo
- Abyss

Third generation sequencing

- Long reads $\approx 10 - 100kbp$
- High error rate $\approx 10 - 12\%$
- High throughput (up to millions of reads per run)

Nanopore VS Pacbio

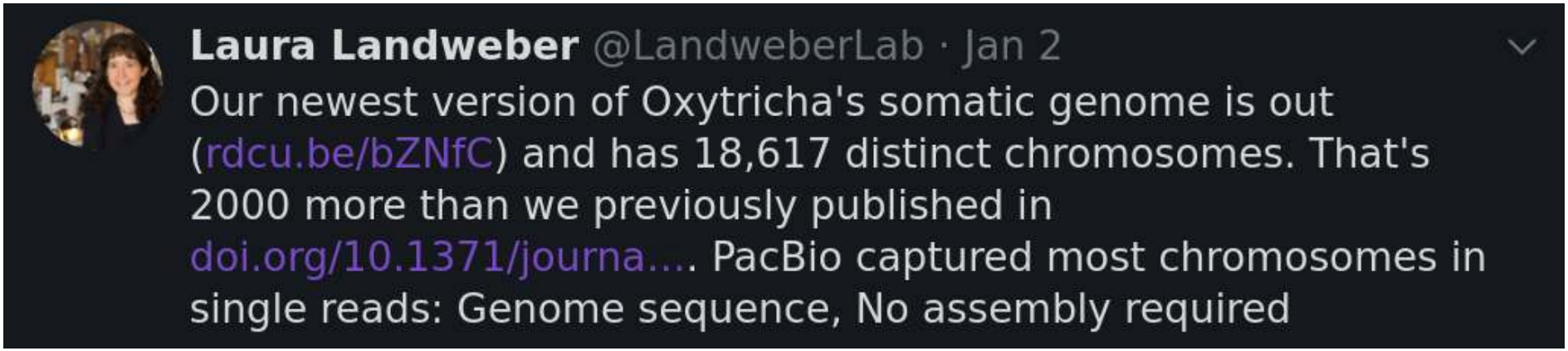
Nanopore

- Portable
- Ultra long reads (100kbases, some reads reach the megabase level)
- Mostly deletions

Pacbio

- More mature
- HiFi reads (99% identity)
- Mostly insertions

Long reads killed the assembly star



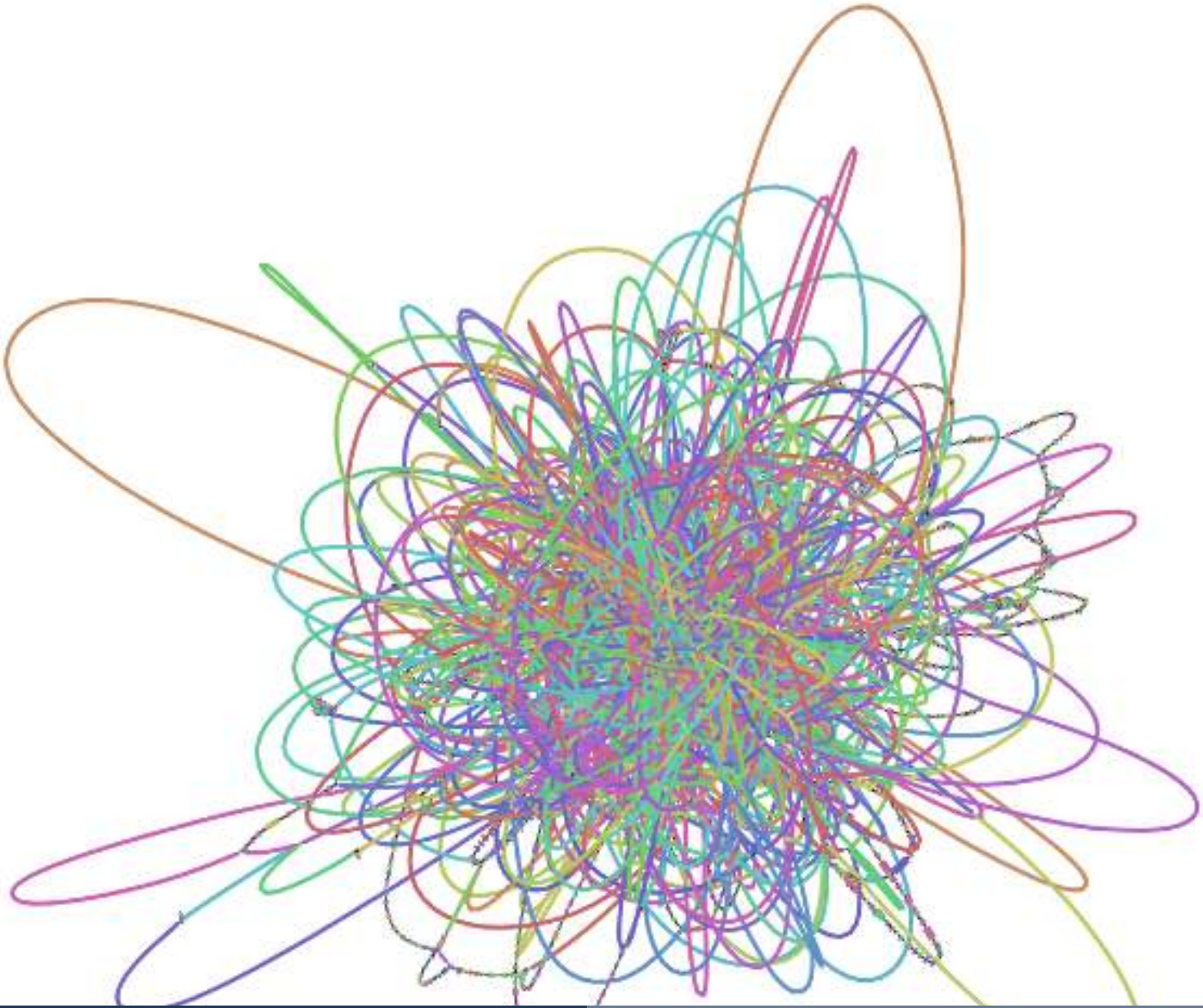
Repeats spanning

Repeats spanning

Repeats spanning

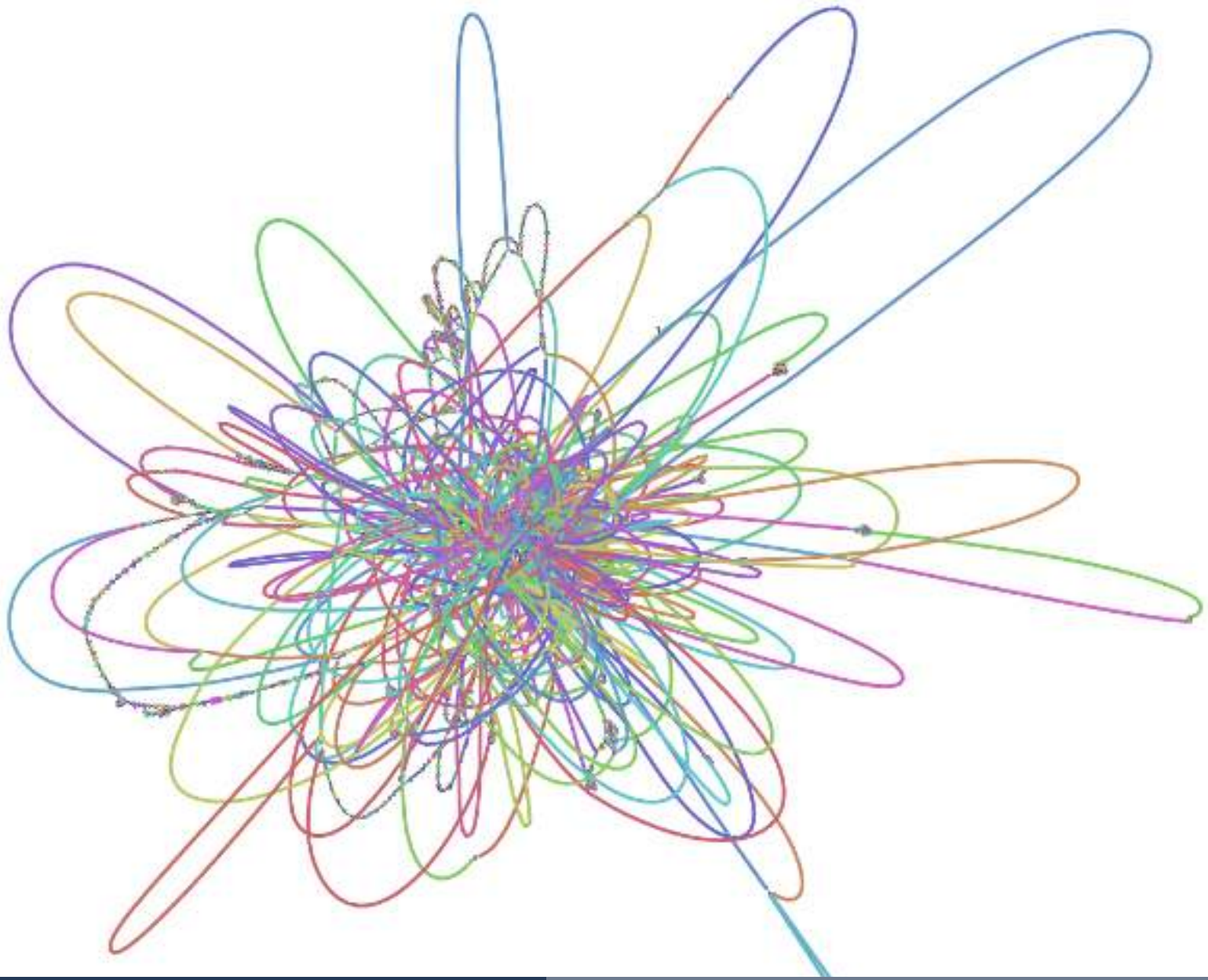
Read length matter

Read size=21



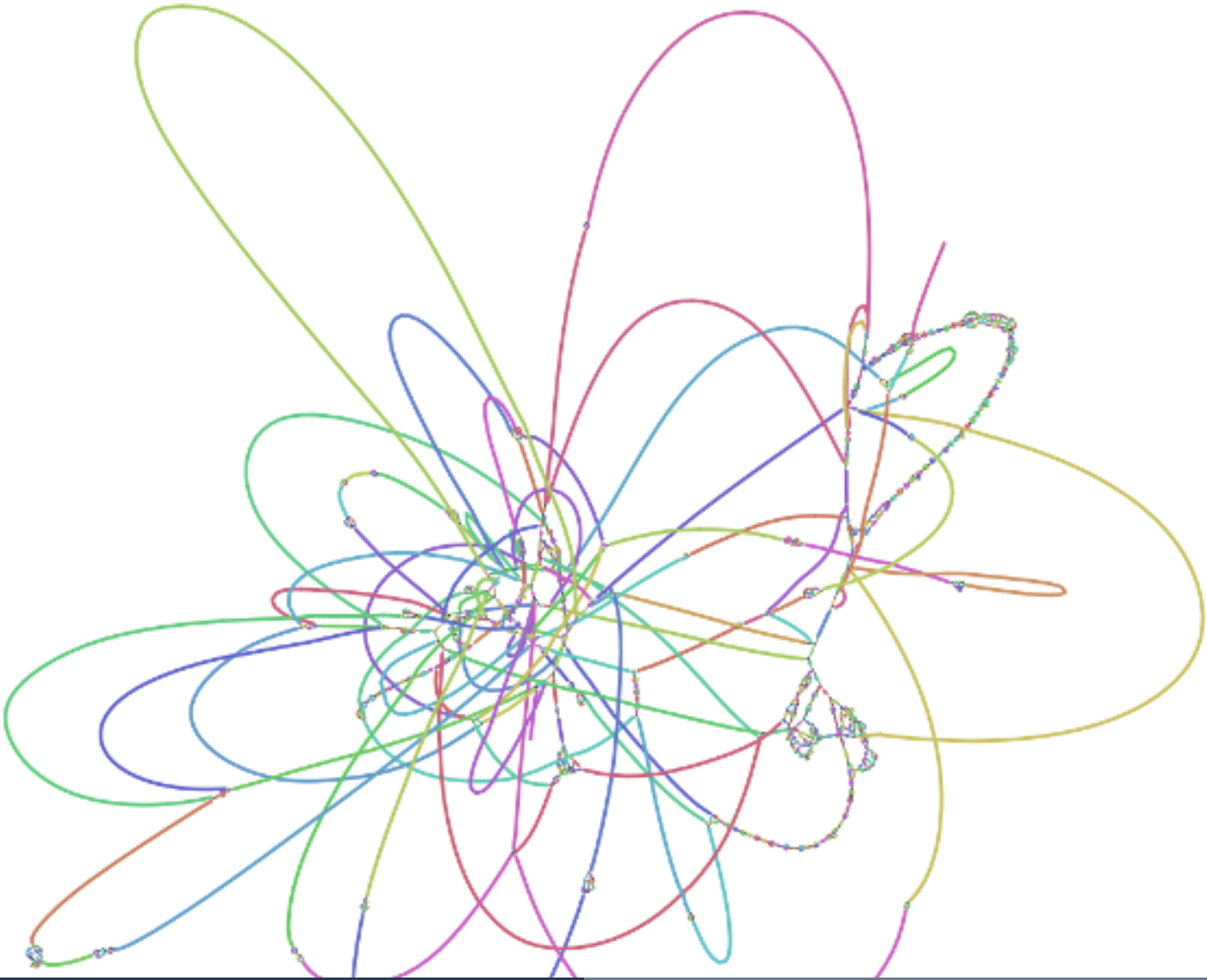
Read length matter

Read size=31



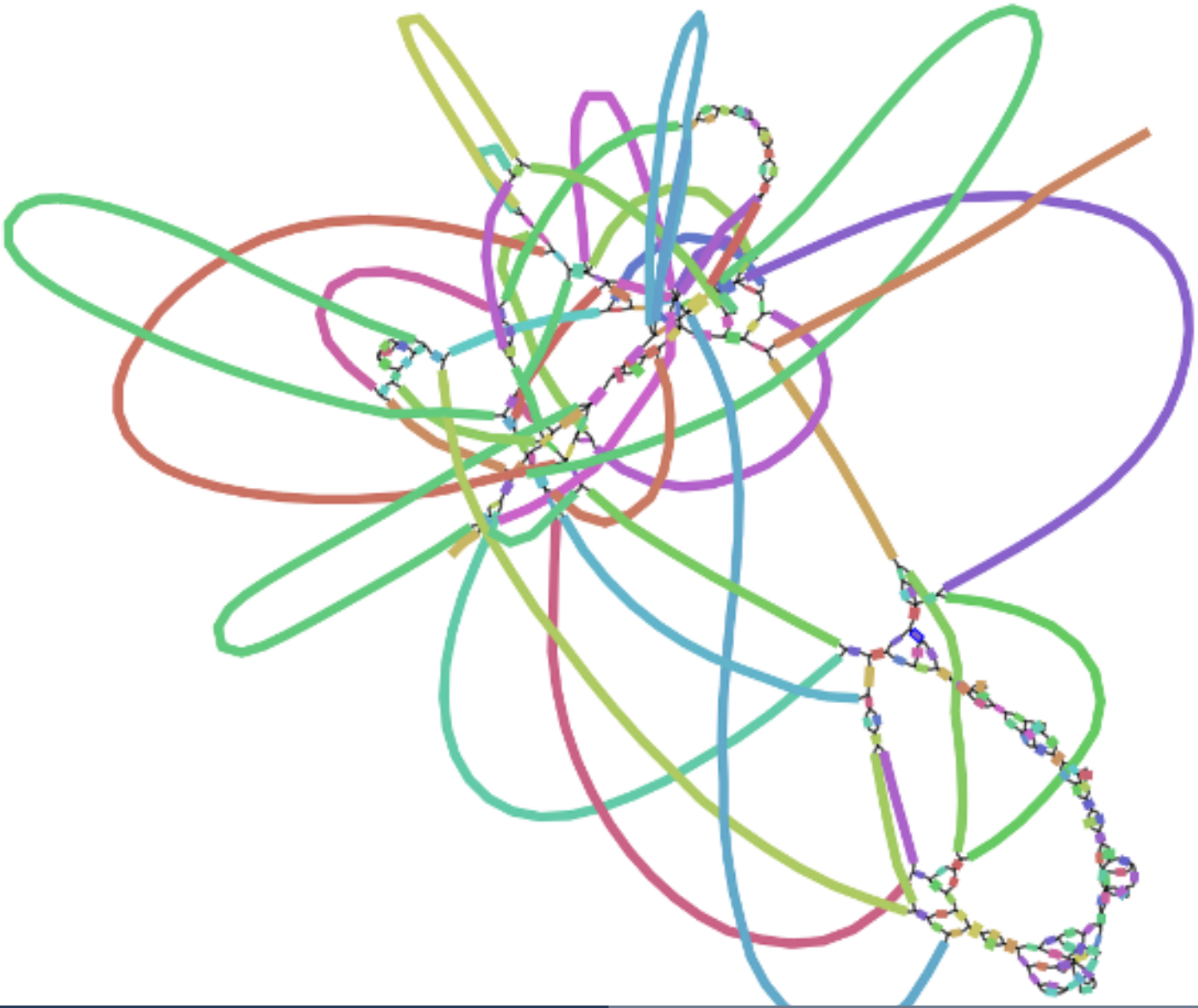
Read length matter

Read size=63



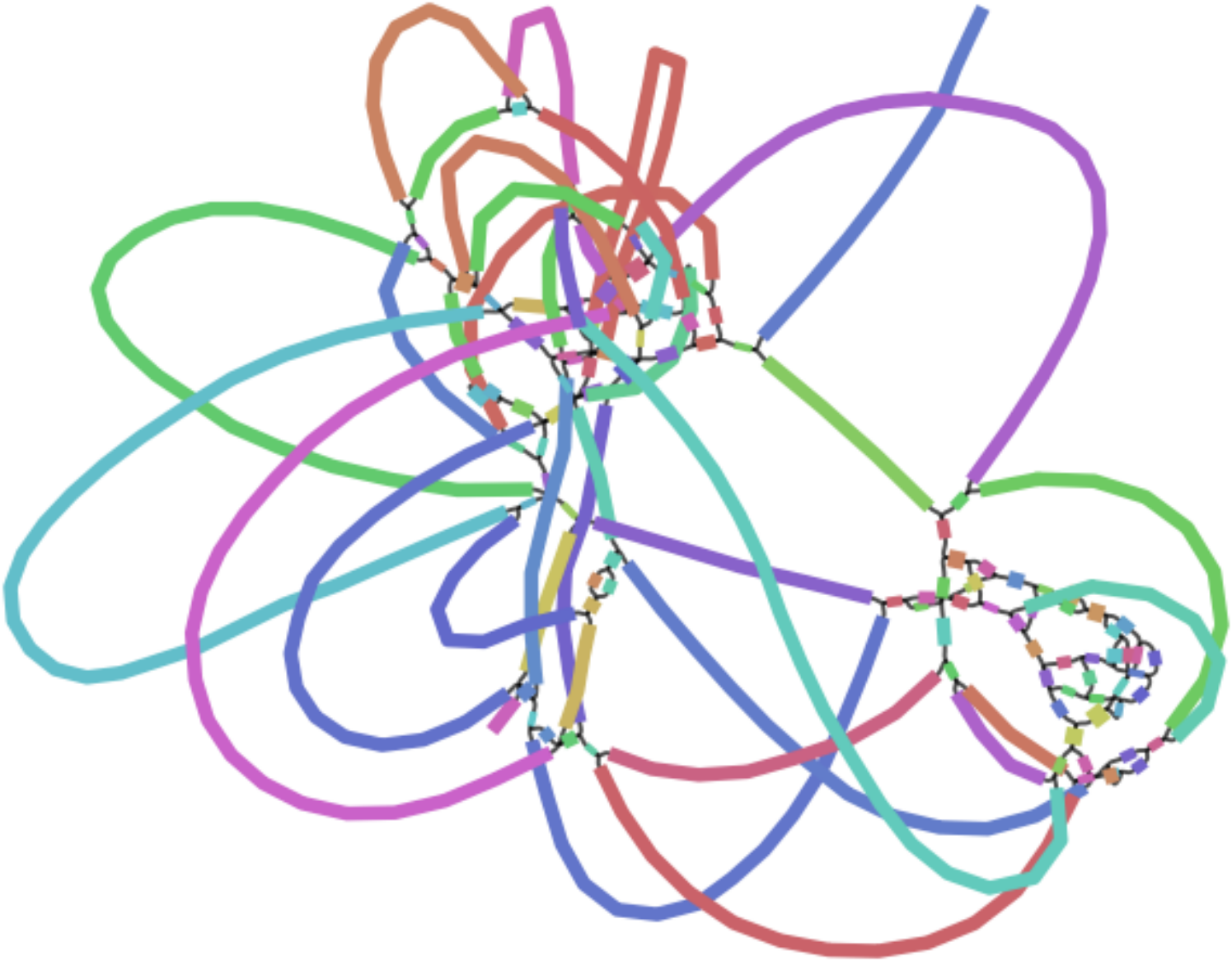
Read length matter

Read size=255



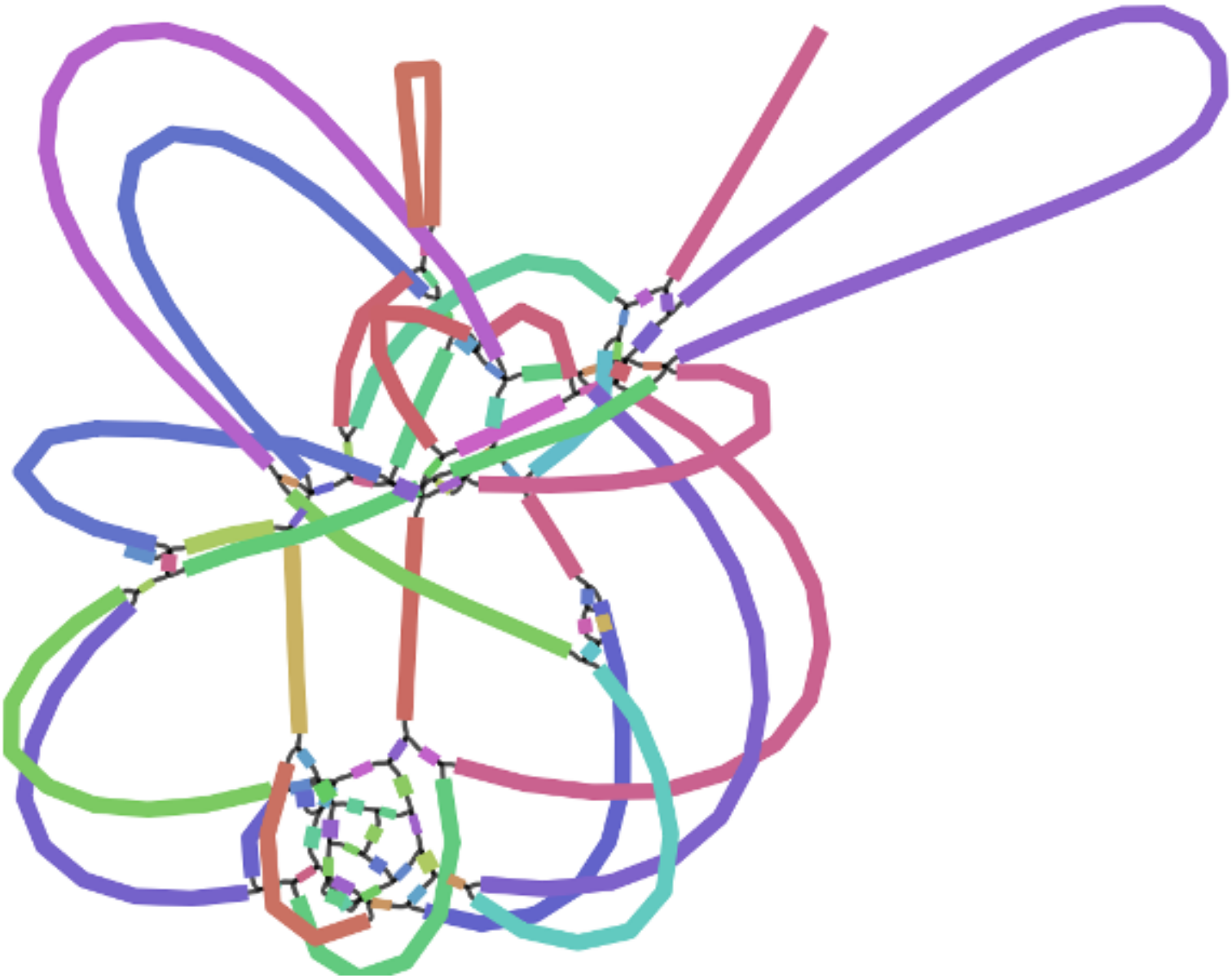
Read length matter

Read size=500



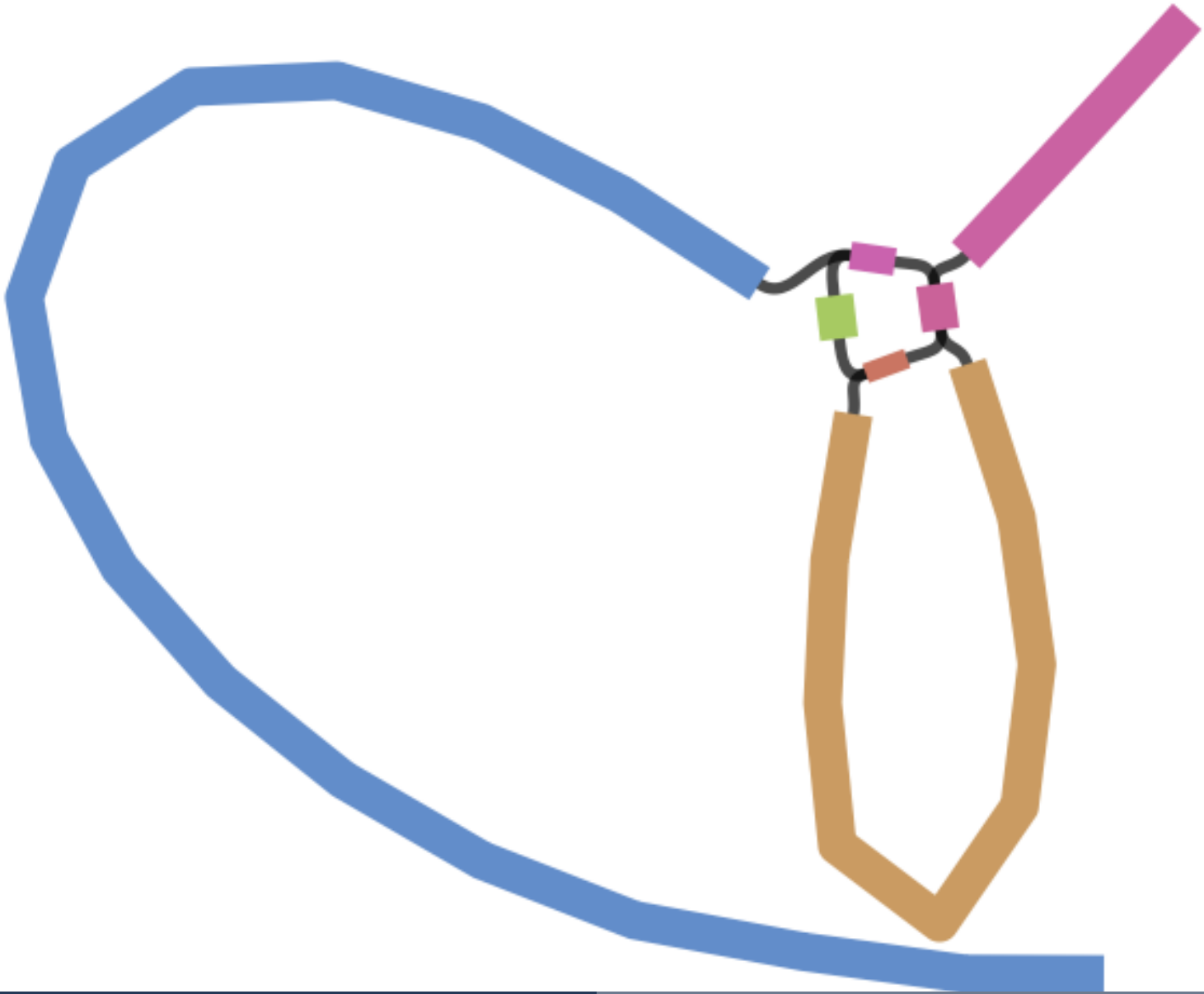
Read length matter

Read size=1000

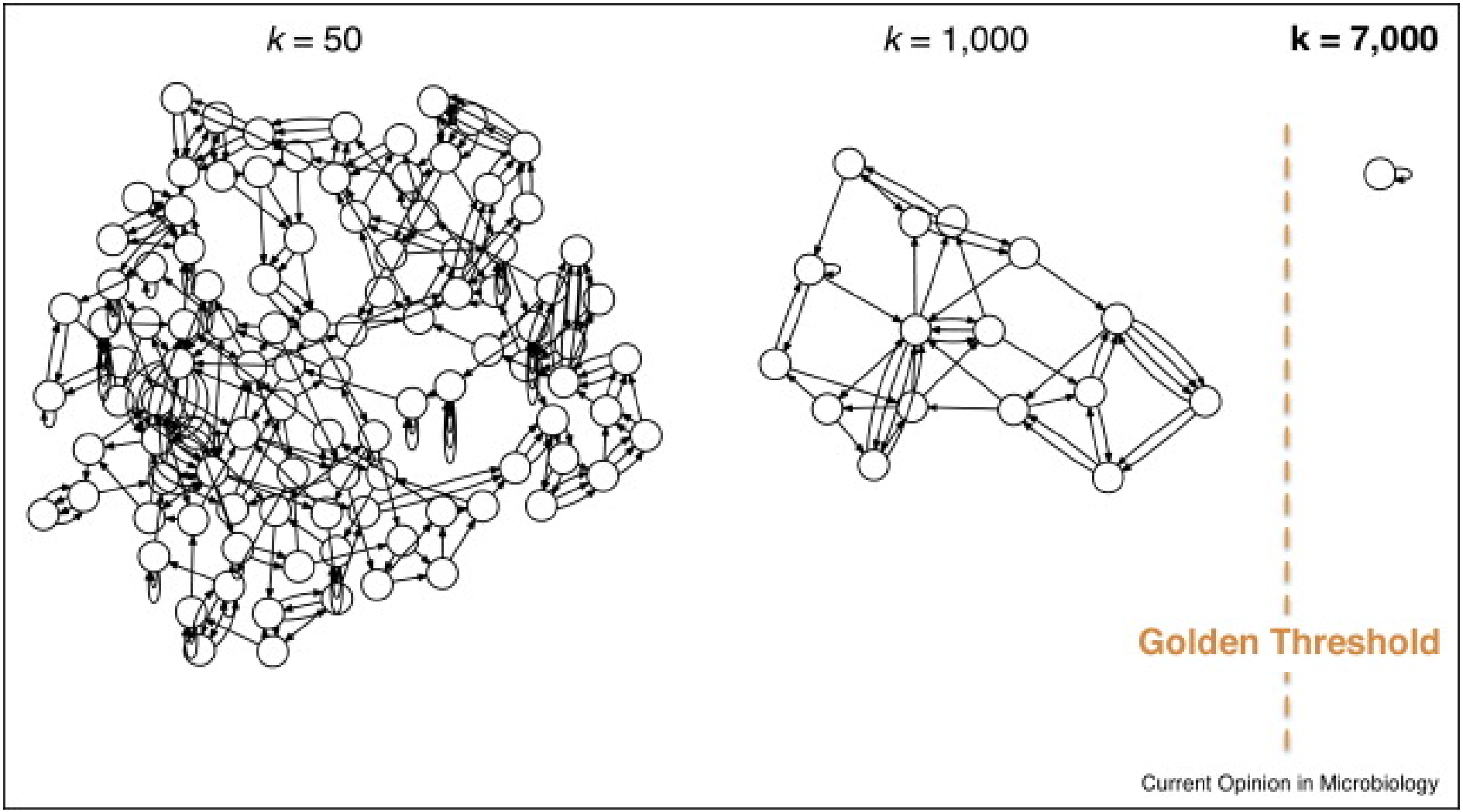


Read length matter

Read size=2000

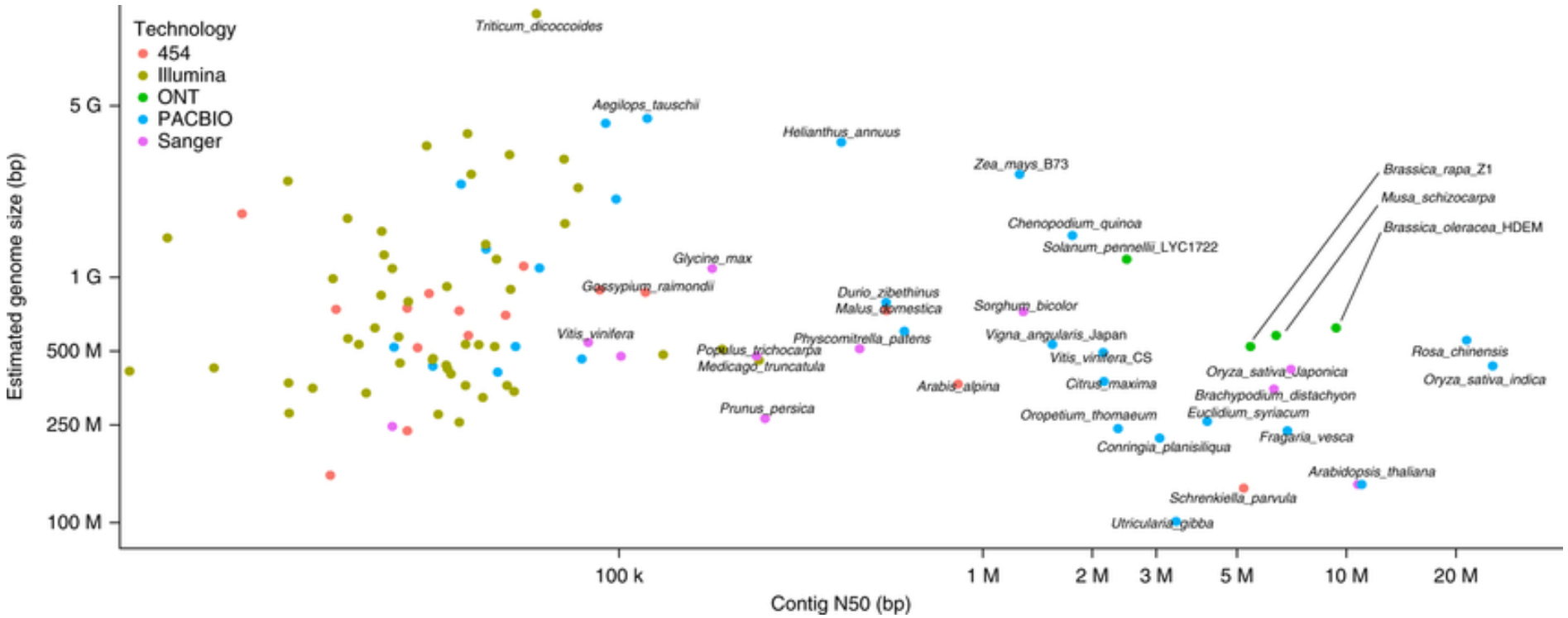


Great hope for assembly



From "One chromosome, one contig: complete microbial genomes from long-read sequencing and assembly" Current Opinion in Microbiology 2015

Great hope for assembly



From "Chromosome-scale assemblies of plant genomes using nanopore long reads and optical maps" Nature Plants 2018

Which assembly strategy is best suited?

- Long reads $\approx 10kbp$
- High error rate $\approx 12\%$
- High throughput (up to millions of reads per run)

Based on long reads properties, which assembly solution would you choose and why?

Vote!

- ~~Fish~~
- Greedy
- Overlap graph
- ~~Birbs~~
- De Bruijn graph

Long reads for assembly: De Bruijn graph?

Long reads for assembly: overlap graph?

Supposed to be super expensive!

Longer reads, better overlaps

- Less reads for the same coverage
- Larger overlaps

5Mb bacteria example with 100X coverage

Short reads

- 5 million 100bp reads
- 99 bp average overlap

Long reads

- 50,000 10kbp reads
- 9,900 bp average overlap

Very long reads

- 5,000 100kbp reads
- 99,000 bp average overlap

Anchors chaining in overlap graph

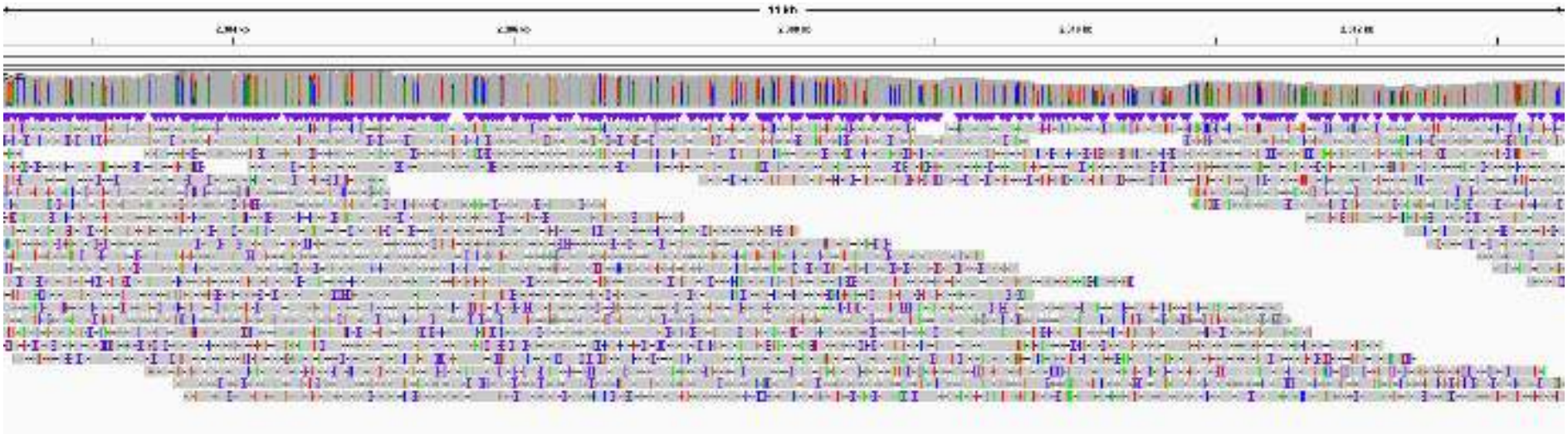
Minimap's job.

Long reads for assembly: overlap graphs

Sequencing errors

Insertion and deletion made calling almost impossible

Sequencing errors



Miniasm

Long reads can be assembled without taking care of the sequencing errors ("Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences", Bioinformatics '16)

Genome characteristics and structure can be quickly estimated

Using coverage to remove noise: Consensus

Exercise: Perform a consensus

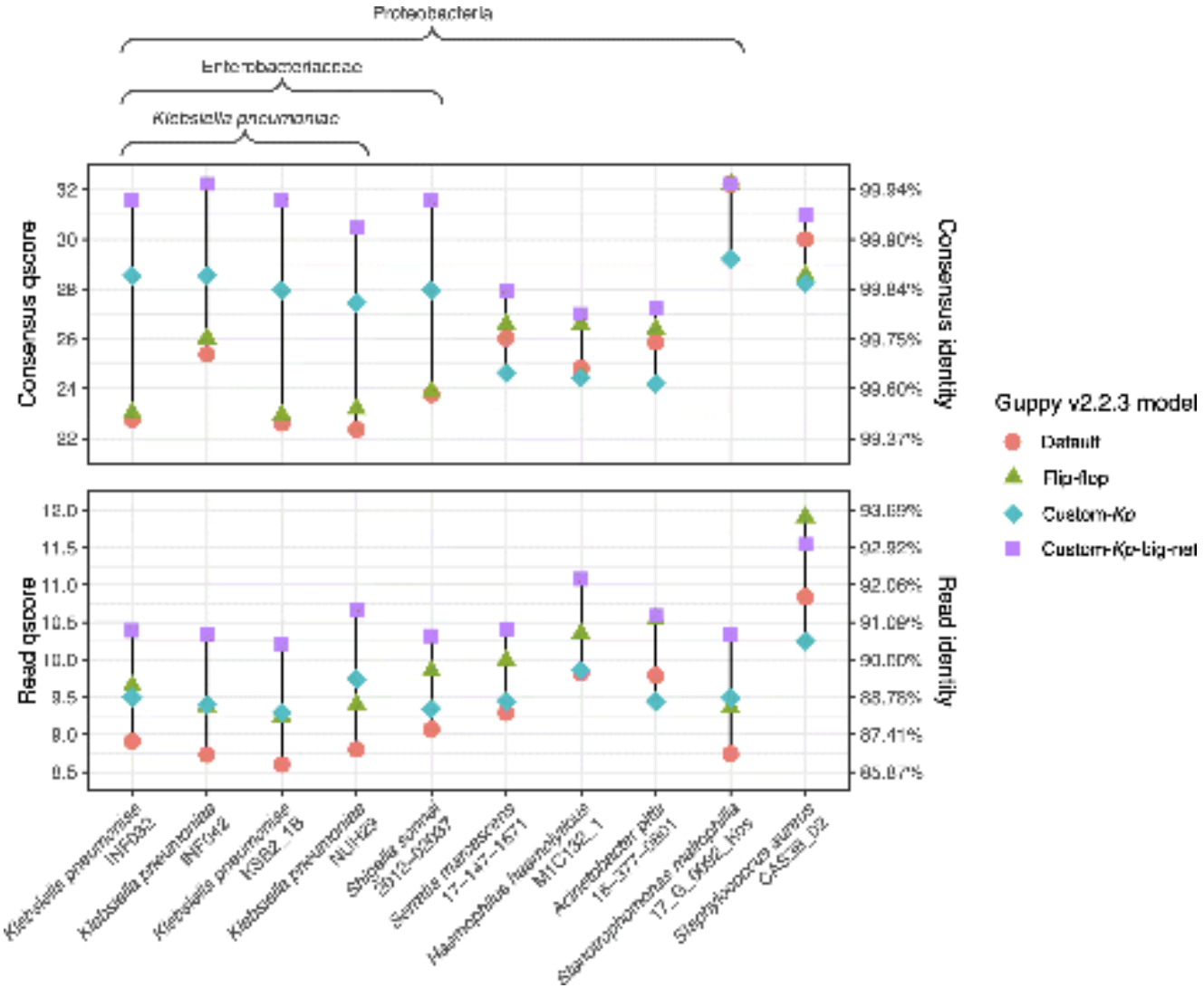
Exercise: Perform a consensus - solution

Consensus during assembly

Consensus after assembly: polishing

Consensus after assembly: polishing

Systematic errors

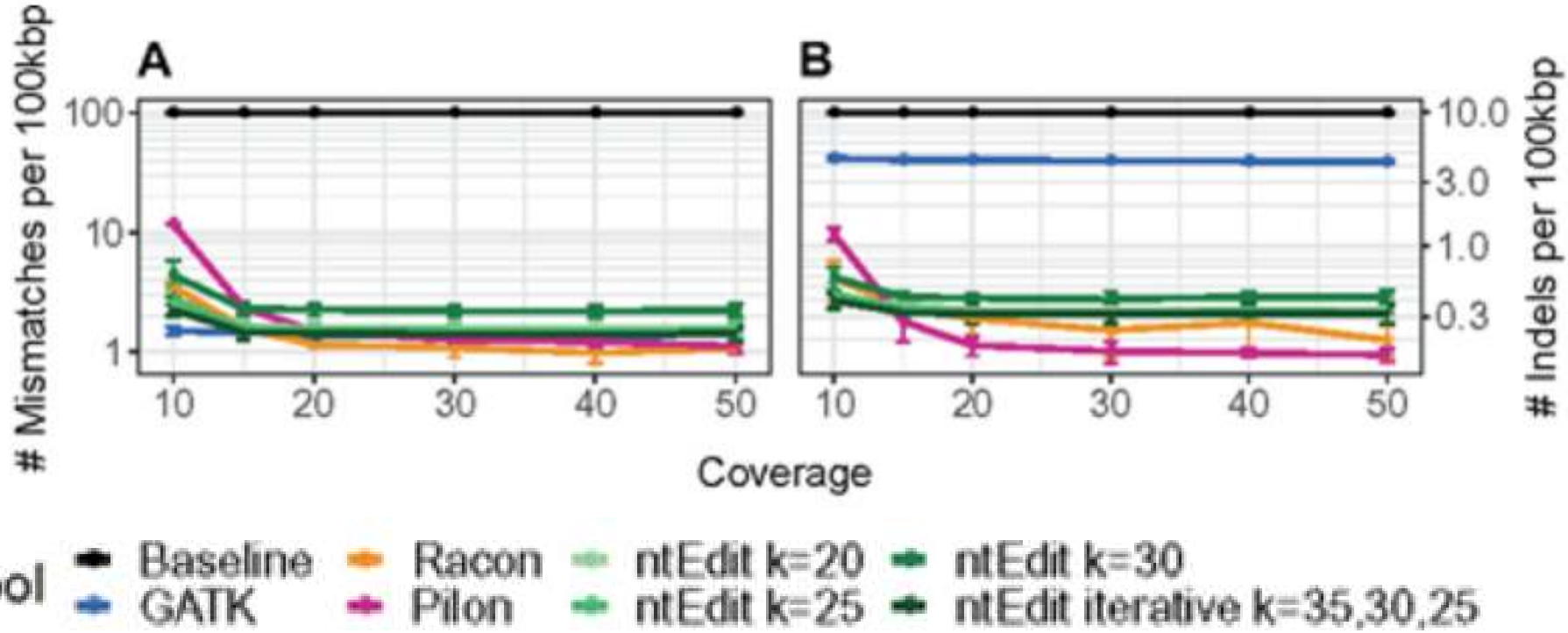


From "Performance of neural network basecalling tools for Oxford Nanopore sequencing" Genome biology 2019

Polishing using accurate reads

Systematics errors

Polishing with Illumina data can improve the final error rate



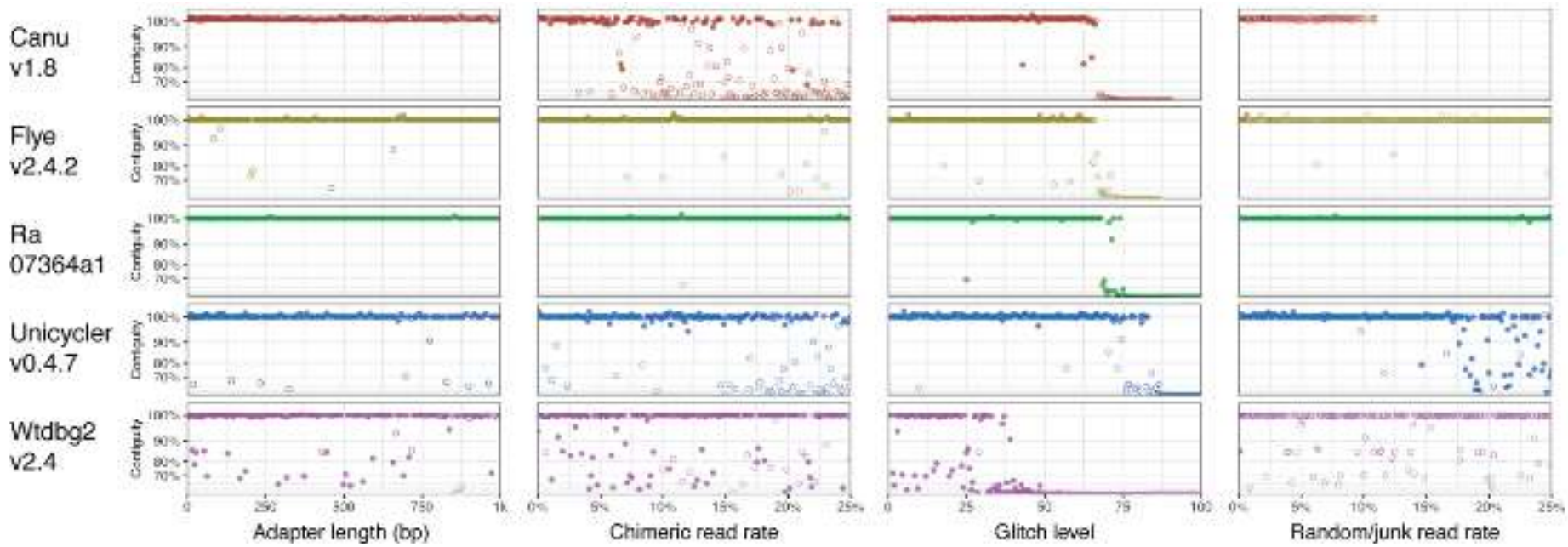
From ntEdit: scalable genome sequence polishing Bioinformatics 2019

Correction before assembly

Long reads for assembly: assembly solved?

Assembly is not solved yet

Sometime the software fail



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

Long reads for assembly: assembly solved?

Assembly is not solved yet

Sometimes the data cannot solve the problem

- Very large repeated region
- Low local coverage
- Chimeric/noisy reads

Long reads assemblers

Your toolkit for the practical

- Flye
- Miniasm
- Raven

Other notable assemblers

- Canu
- Mecat
- Redbean
- ...

Long read assembly summary

- Not resolved: correction before or after assembly (polishing)
- Overlap graphs with quick overlap computation
- Long reads can span repeats and improve assemblies
- Methods to polish contigs

Conclusions

Take home messages

- Short reads: De Bruijn graphs / Long reads: Overlap graph
- Repeats are the core issue
- Output fragments of genomes (**contigs**)
- Several steps and heuristics in practice

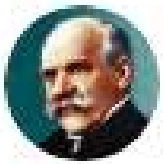
Challenges

- Difficult to reconstruct haplotypes
- Scaling on large genomes
- Robustness to noisy data

Topics we did not review today

- Read correction
- Multi k assembly
- Use of paired-end, mate-pair, 10X, Hi-C . . .)

The end



PopGenGoogling
@popgengoogling

i trust you to figure out your own genome

Traduire le Tweet
3:01 AM · 14 déc. 2019 · Twitter for iPhone

37 Retweets **267** J'aime

Practical session

Datasets

Short Illumina reads 150bp
Long PacBio reads

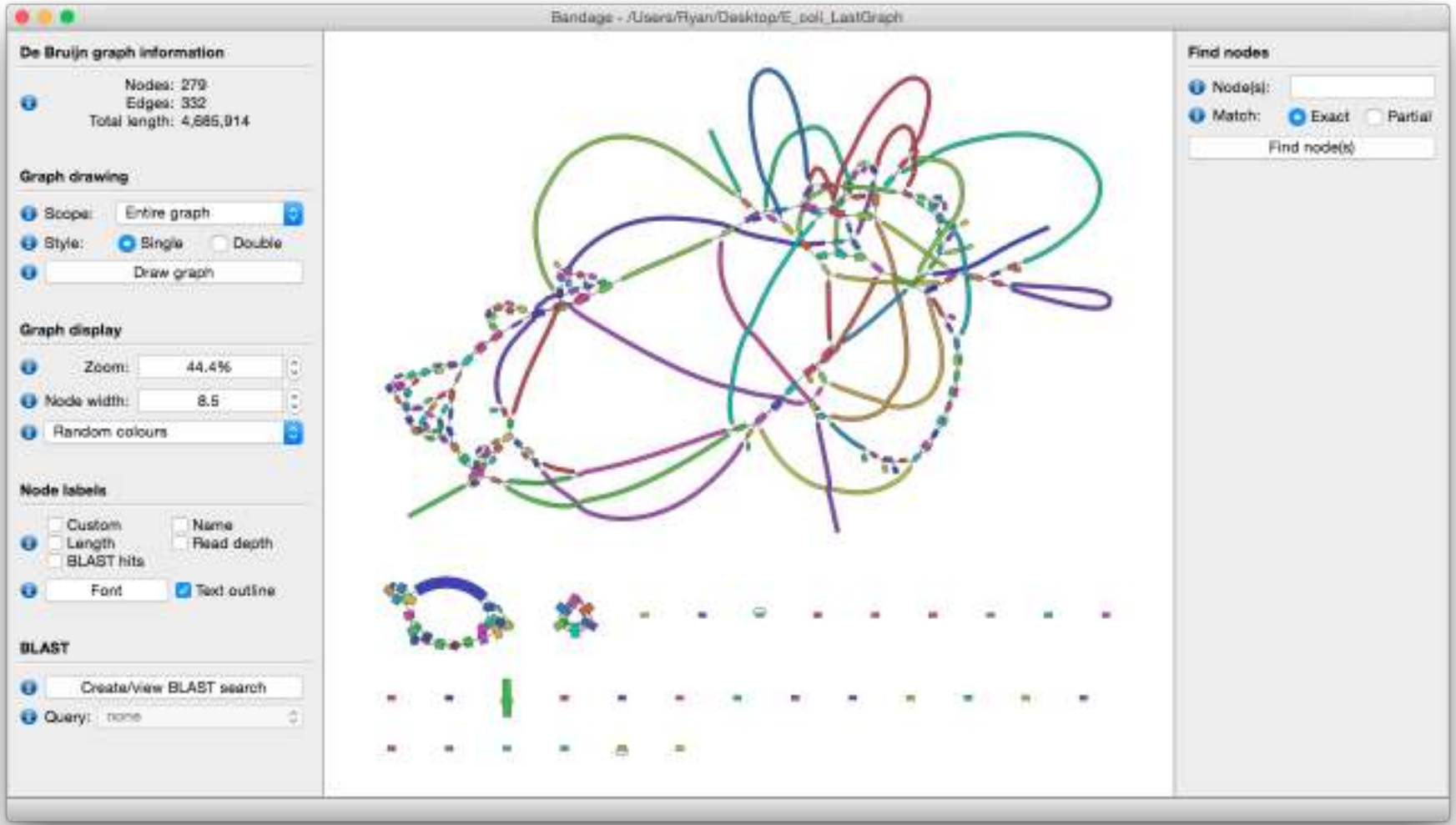
Steps

- Perform a draft assembly
- Evaluate it
- (Try to) Perform a better assembly

You will read the assembler website and papers to decide which one you want to use

Visualize assembly

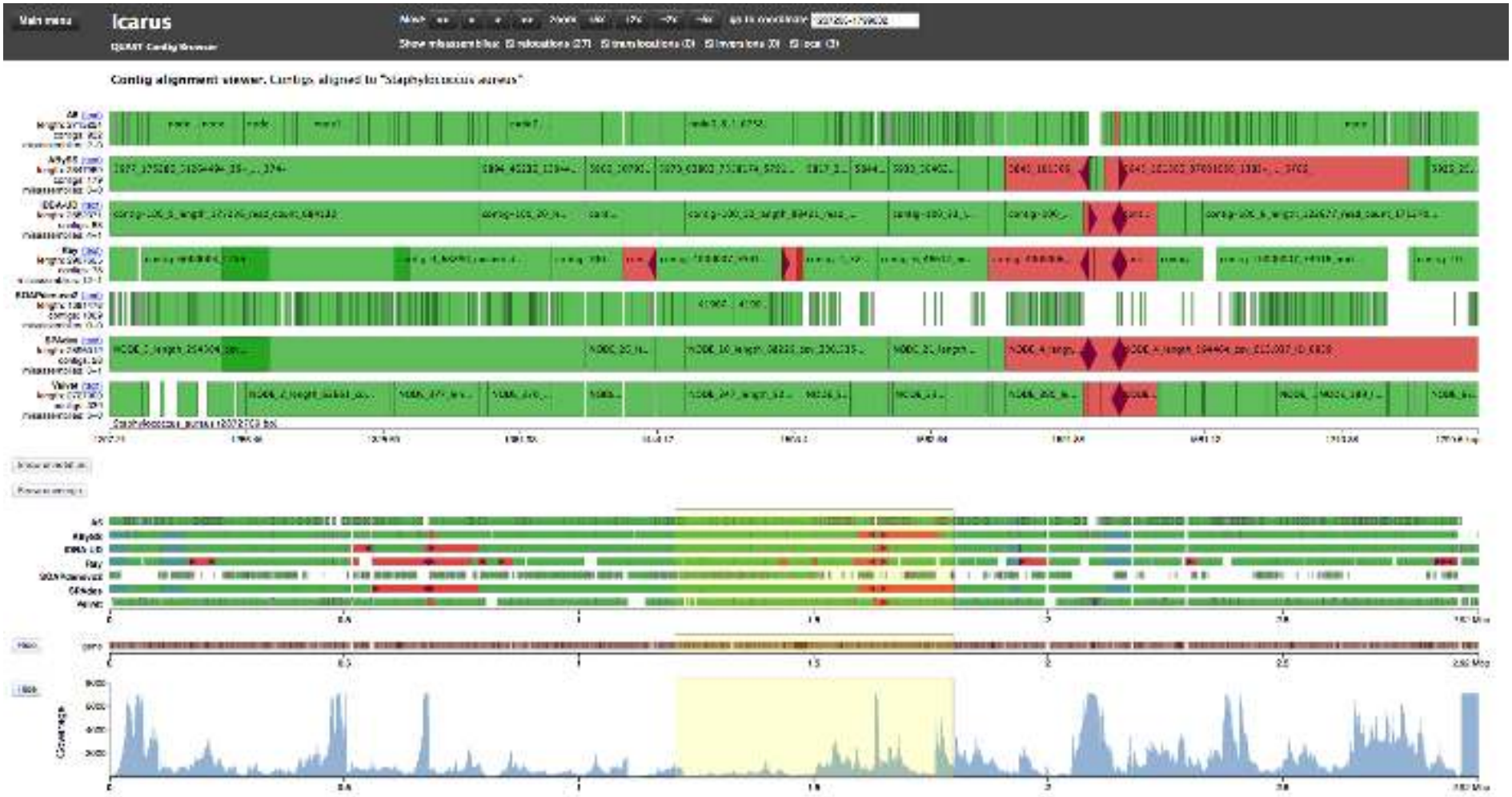
Bandage tool can visualize assembly graphs (GFA)



From `rwick.github.io/Bandage`

Evaluate assembly

Contigs can be mapped and compared to a reference/closely related genome



From `quast.bioinf.spbau.ru/manual.html`

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.

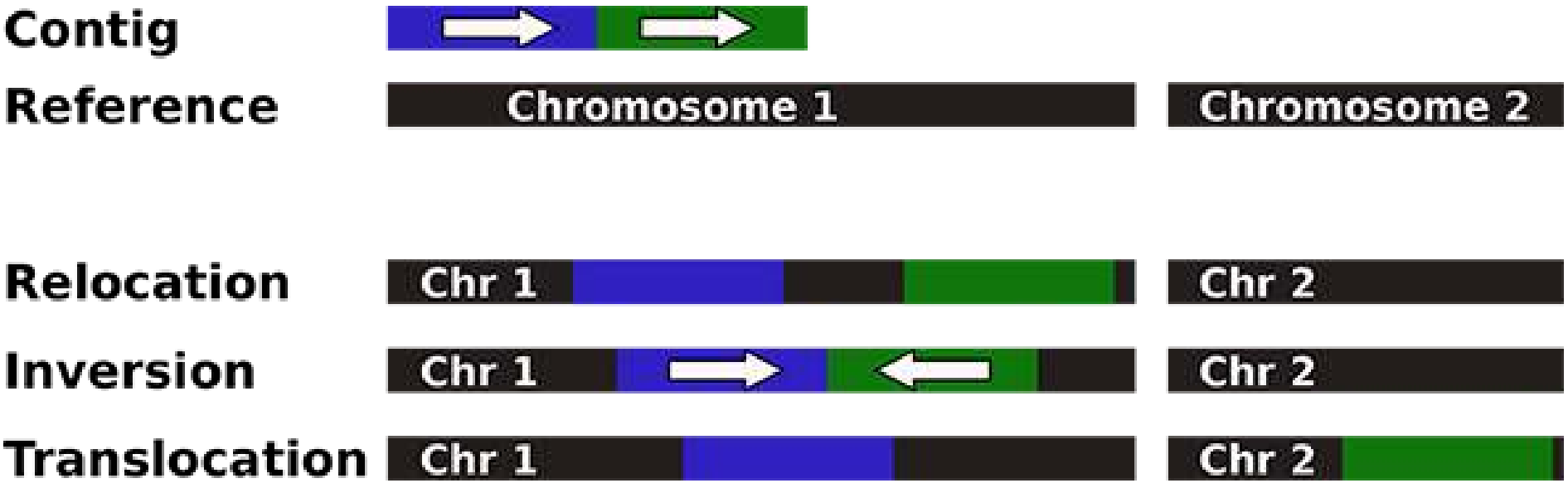
N75

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

Misasemblies



Multiple k assembly

Most De Bruijn graph assemblers can now perform several assemblies with different k-mer sizes to produce an improved "super" assembly
(will be discussed in metagenomic session)

Advice

Using a single size of k-mer will allow the assembly to go way faster during the practical