GraphUnzip: using assembly graphs to improve assemblies

Roland Faure[1,2], Nadège Guiglielmoni[1,4], Jean-François Flot [1,3]

[1]Evolutionary Biology & Ecology
Université libre de Bruxelles (ULB)

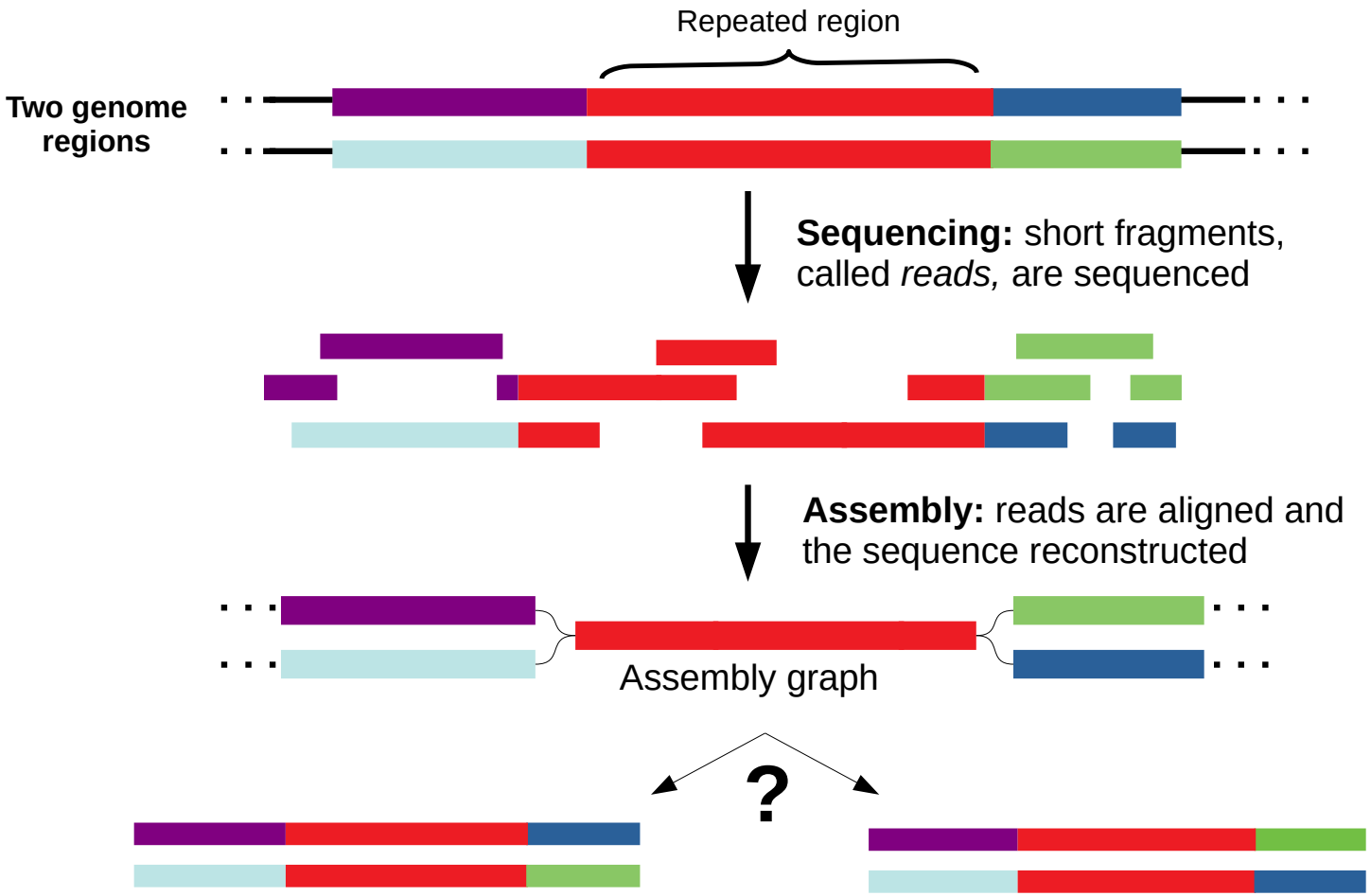
[2]Université de Rennes, Inria RBA

[3]Interuniversity Institute of Bioinformatics in Brussels

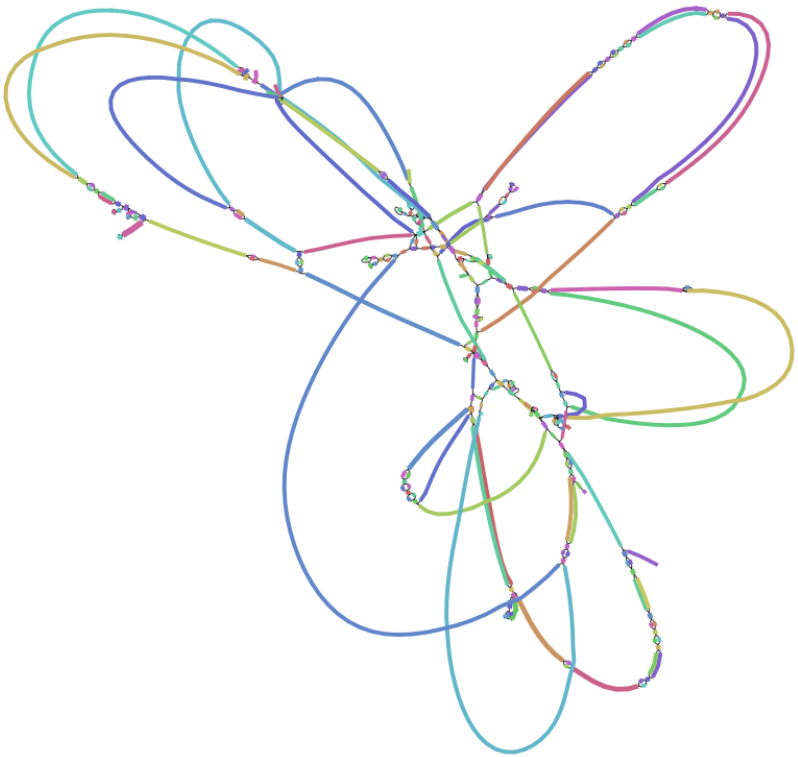
[4]Universität zu Köln

November 2021

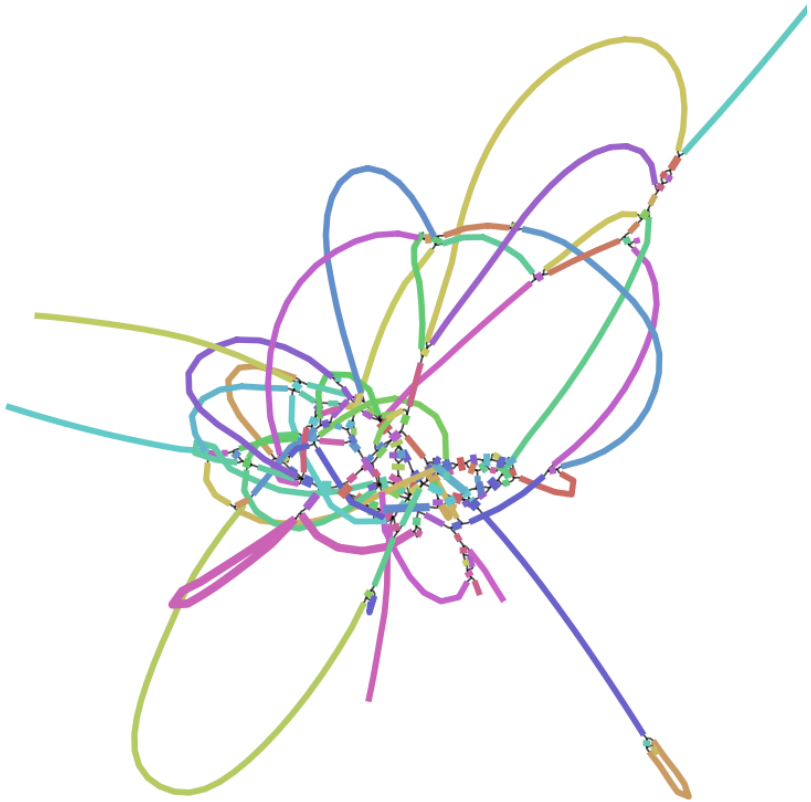
Genome assembly



Real assembly graphs

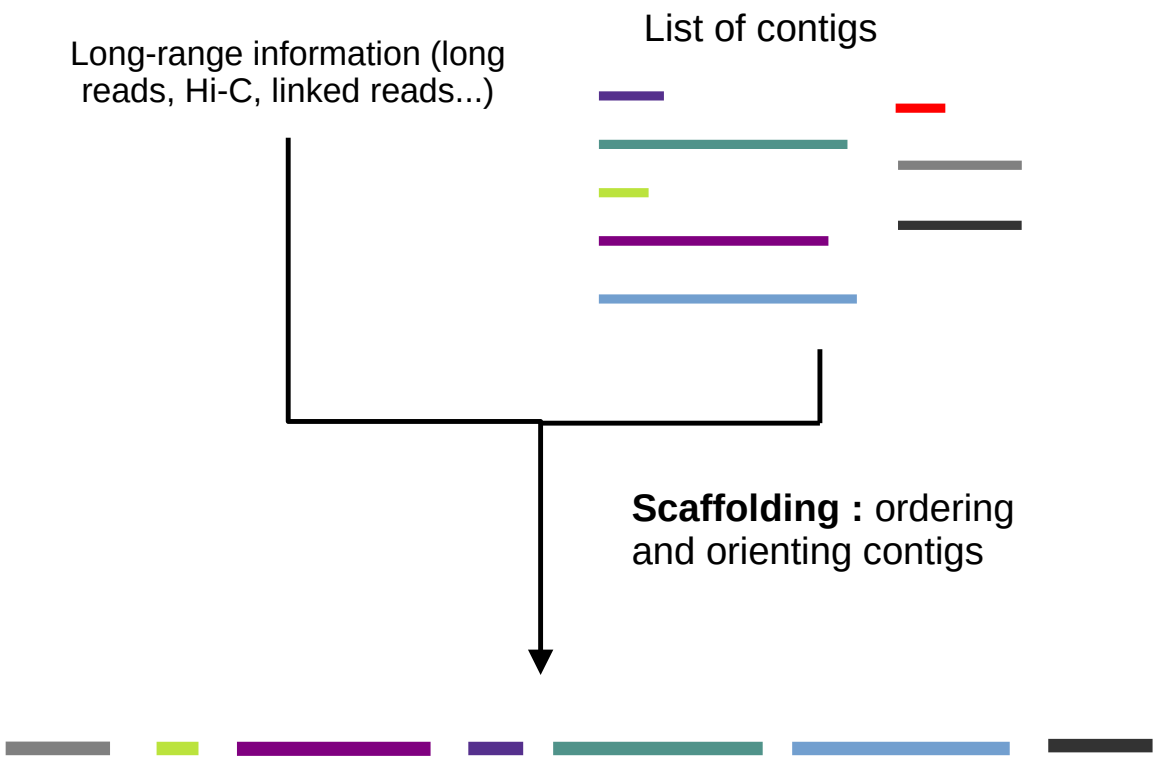


Animal: *Adineta vaga* (PacBio CLR + Shasta)

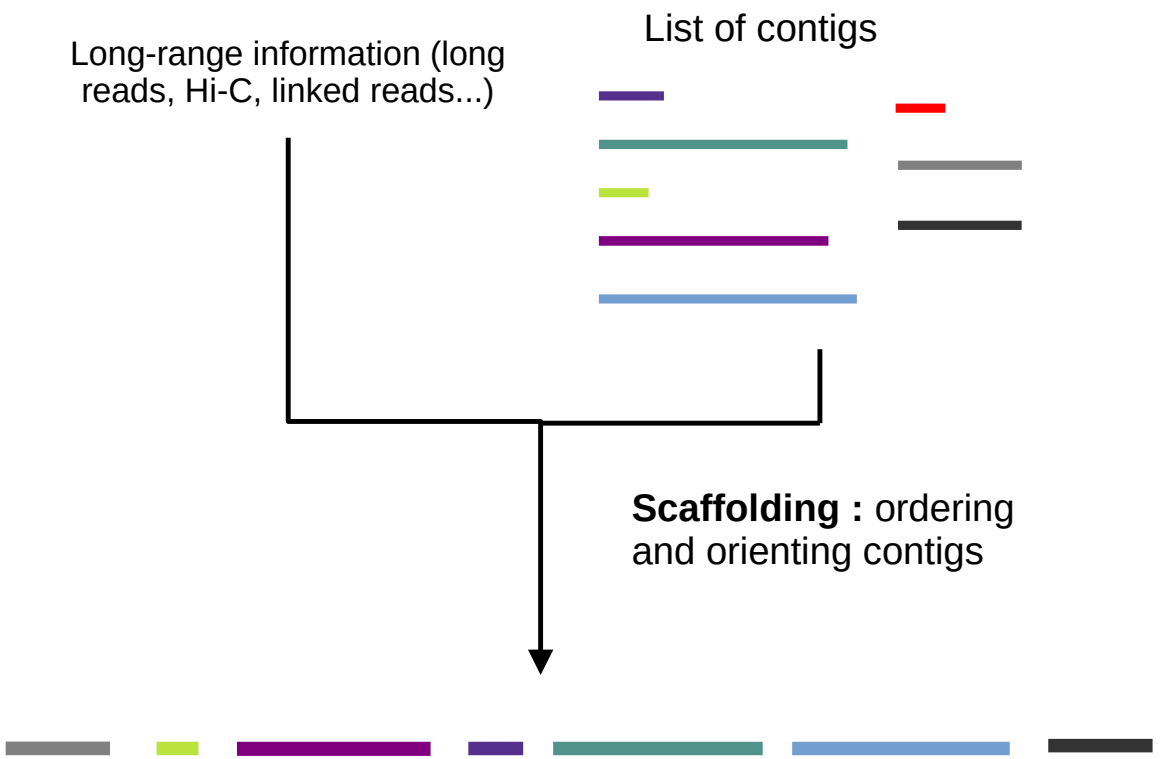


Bacterium: *Acidithiobacillus* sp. (Illumina + SPAdes)

Finishing assemblies: scaffolding

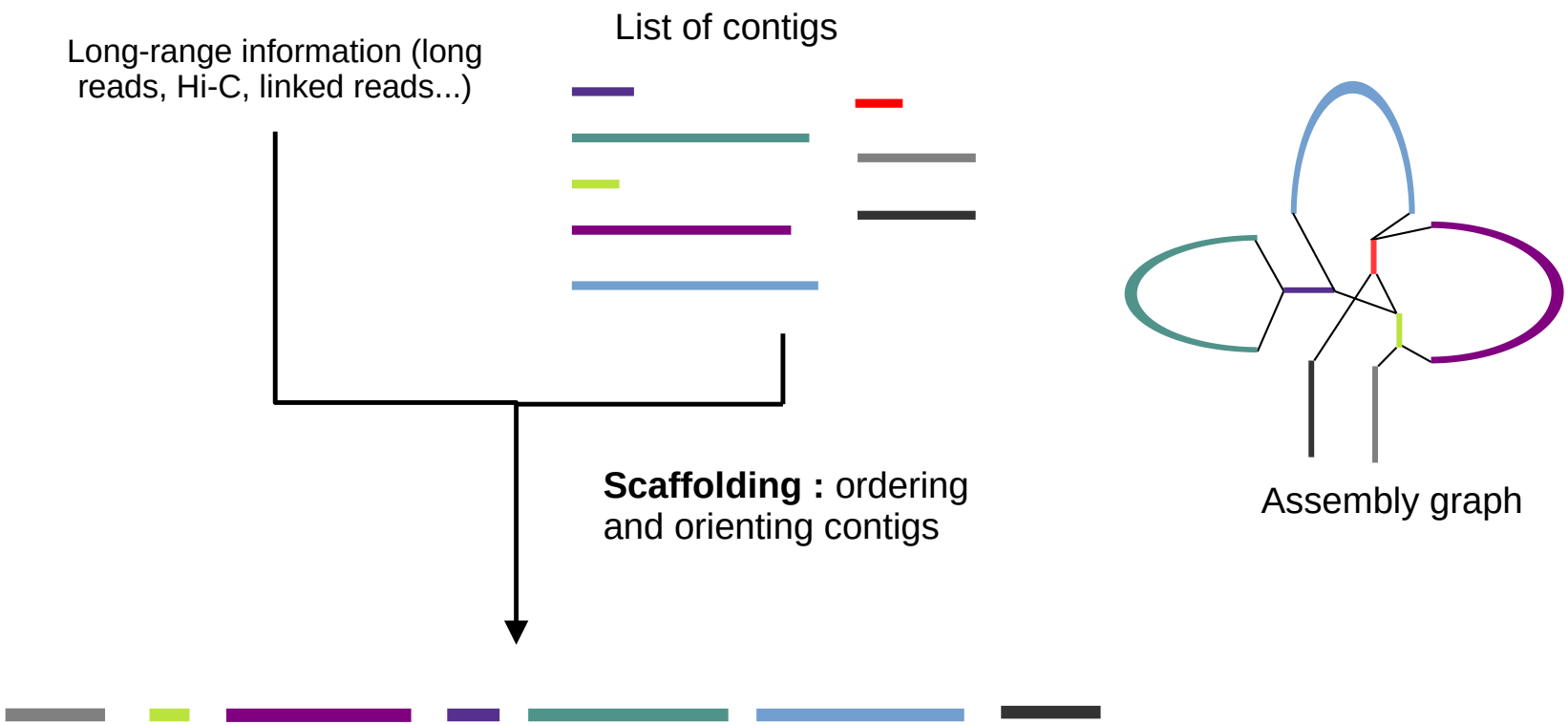


Finishing assemblies: scaffolding



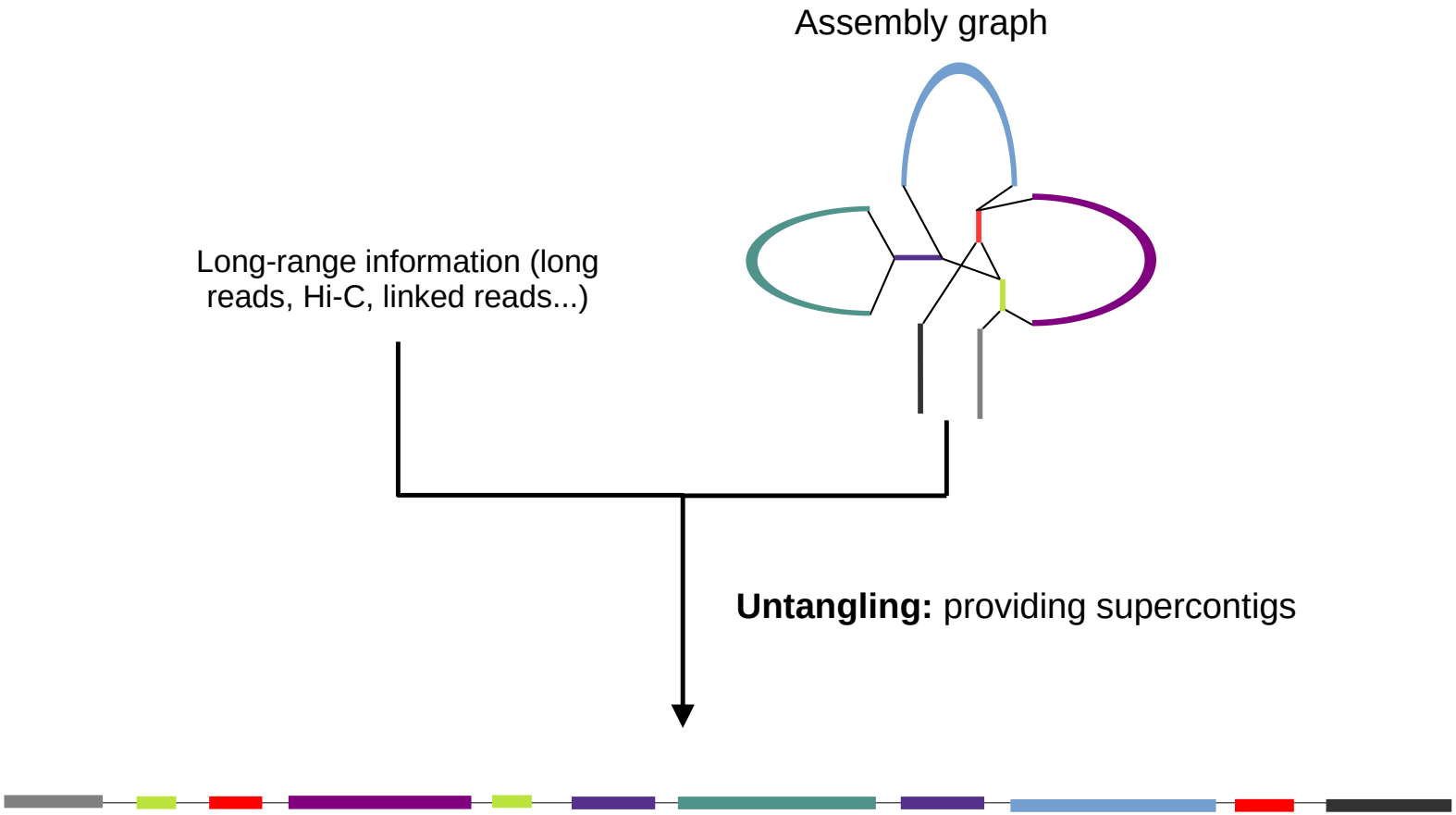
- Gaps of approximate length between contigs
- Scaffolding may lose contigs

Finishing assemblies: scaffolding



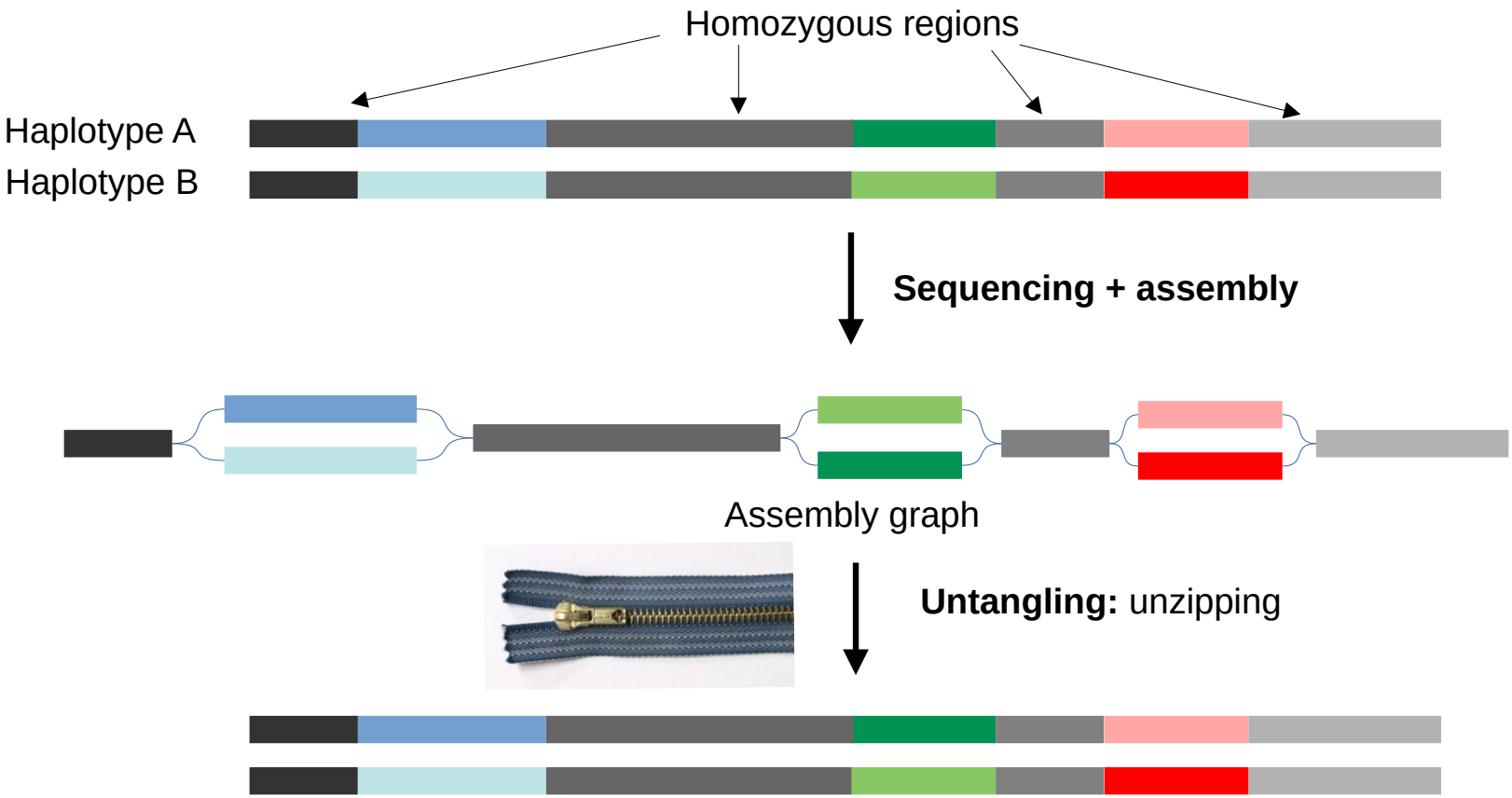
- Gaps of approximate length between contigs
- Scaffolding loses contigs

An alternative to scaffolding

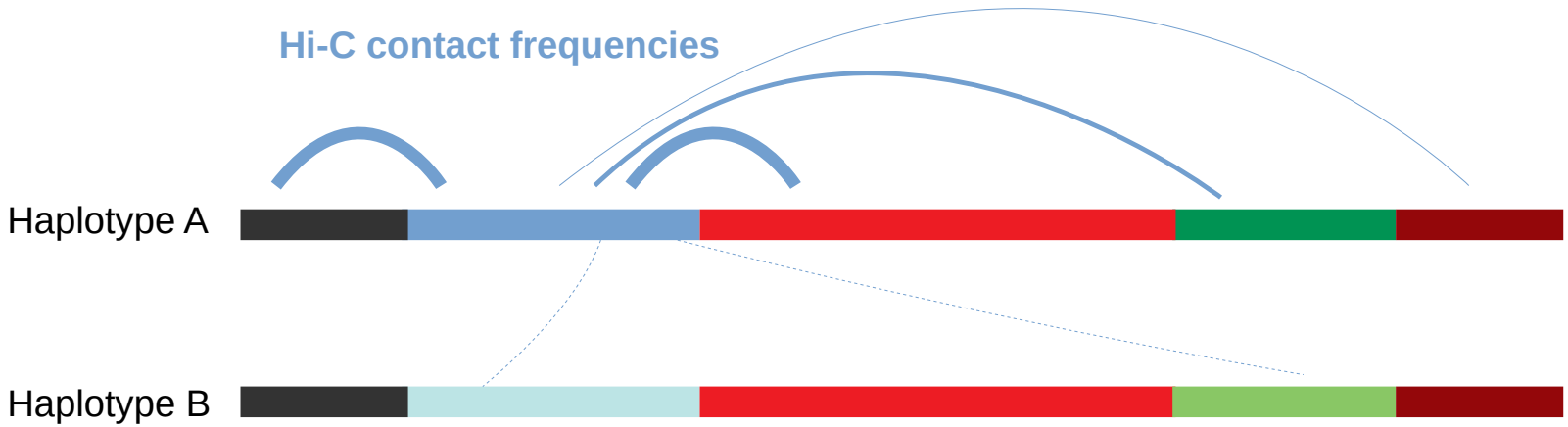


- Duplicating all contigs that are present in multiple copies
- Find paths through the graph to reach maximum contiguity

Particular case: multiploid genomes

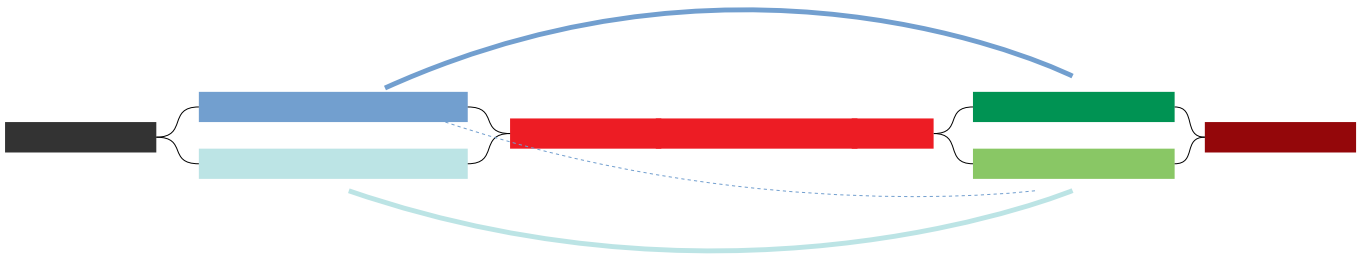


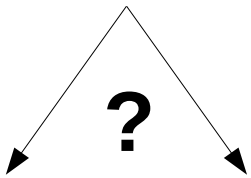
Integrating other types of data: Hi-C



- The closer the contigs the more frequent the contacts
- Intrachromosomal are more frequent than interchromosomal contacts

Integrating other types of data: Hi-C





Hi-C frequencies support the correct unzipping of the graph

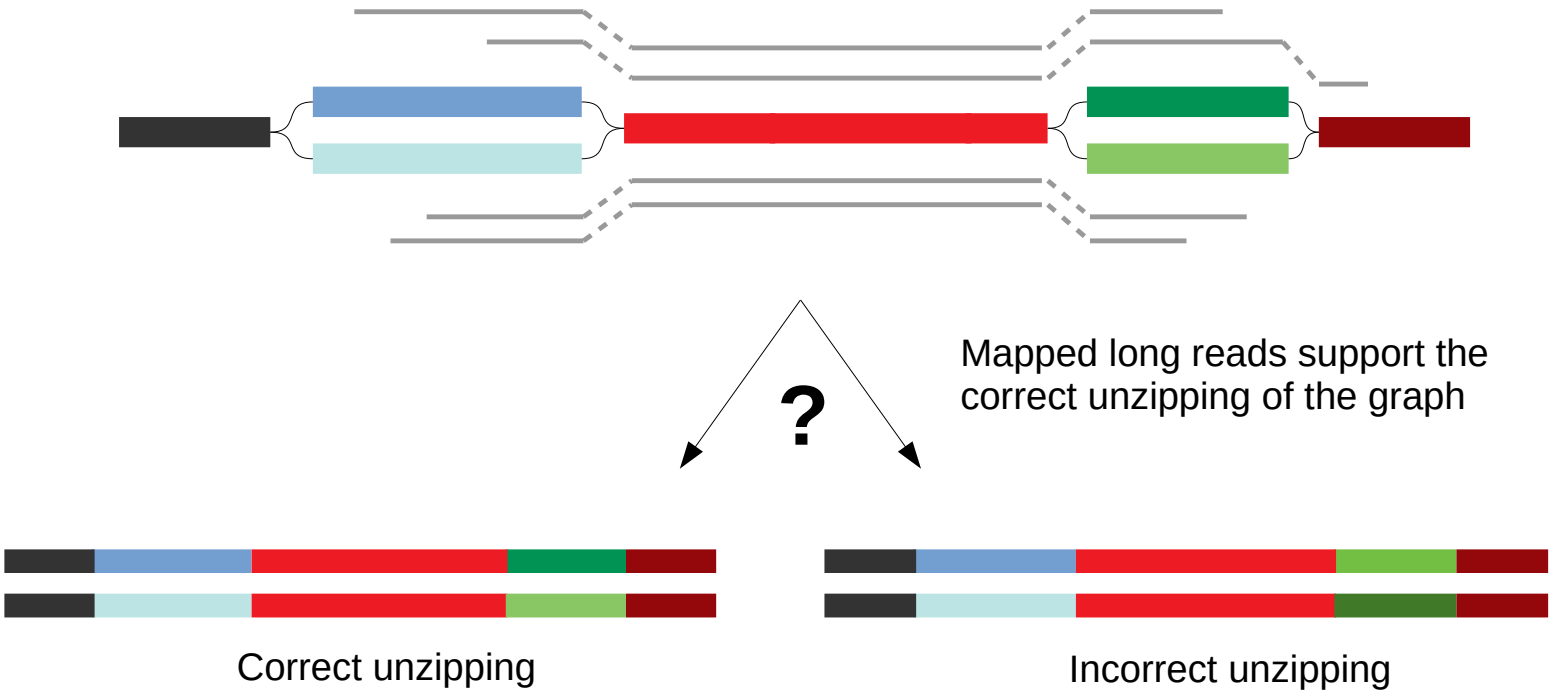
Correct unzipping

Incorrect unzipping

Integrating other types of data: (ultra-)long reads

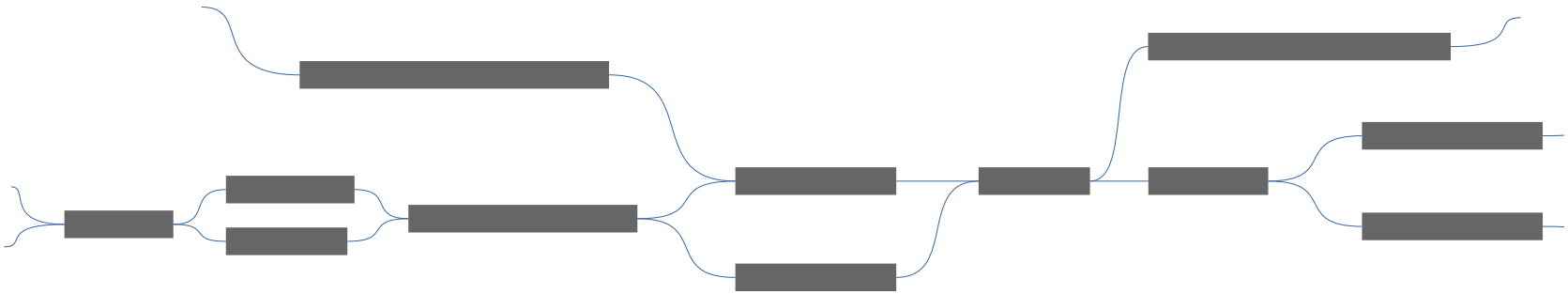
- Long reads can be mapped to the graph using e.g. GraphAligner
- PacBio (~20 kb) or Oxford Nanopore (10-100+ kb) can be used

Integrating other types of data: long reads

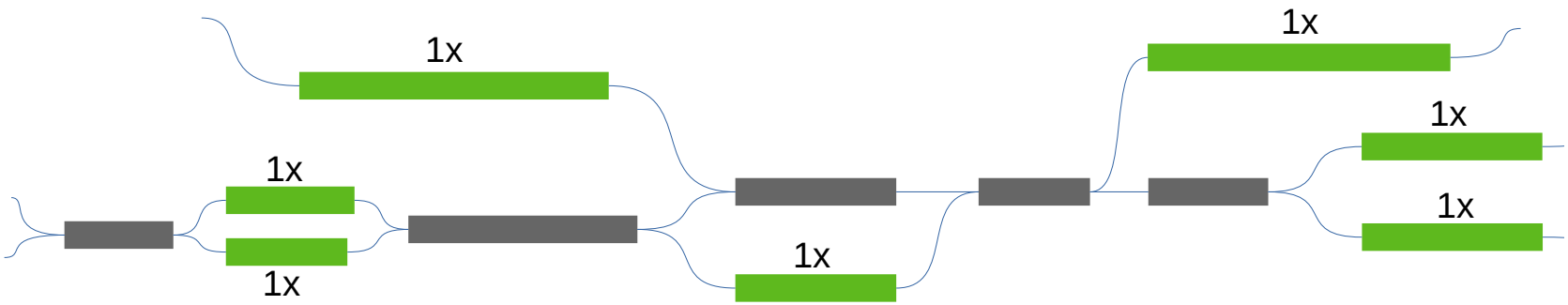


- An algorithm inspired by the program Unicycler

Algorithm: determining single-copy contigs

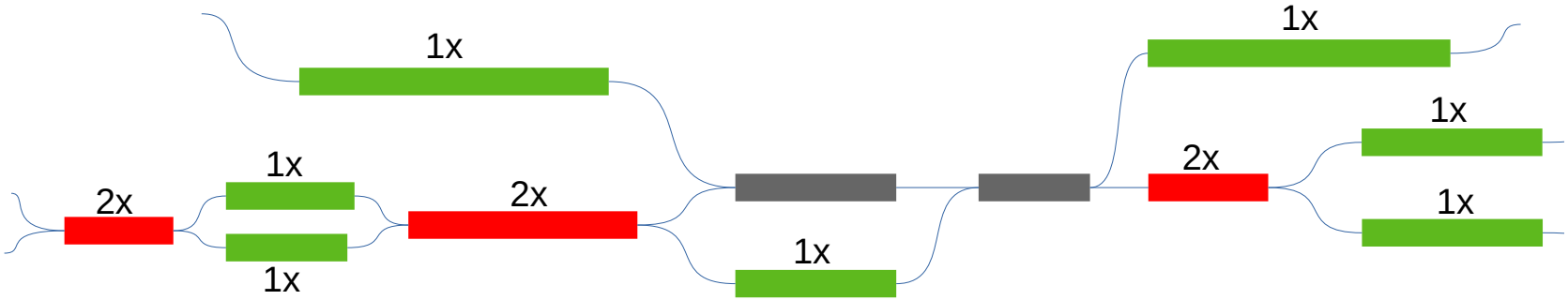


Algorithm: determining single-copy contigs



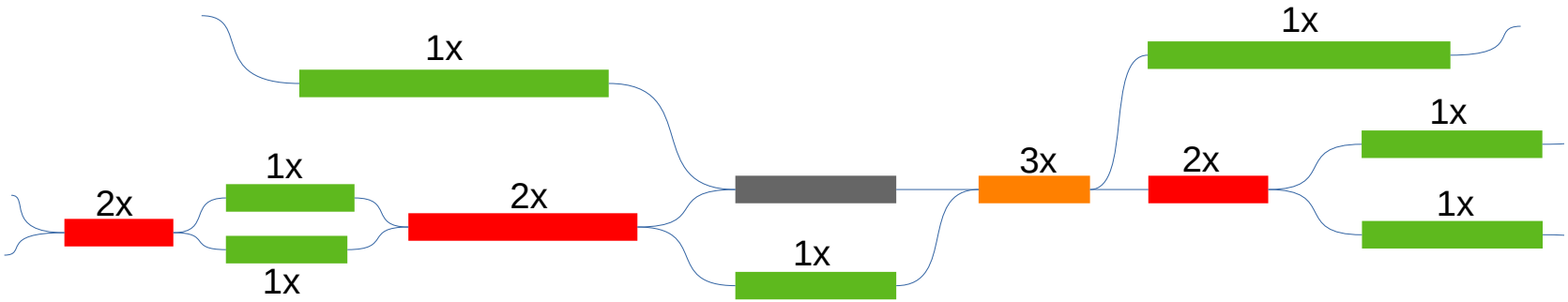
- Only one link left and right
- Coverage information

Algorithm: inferring multiplicities



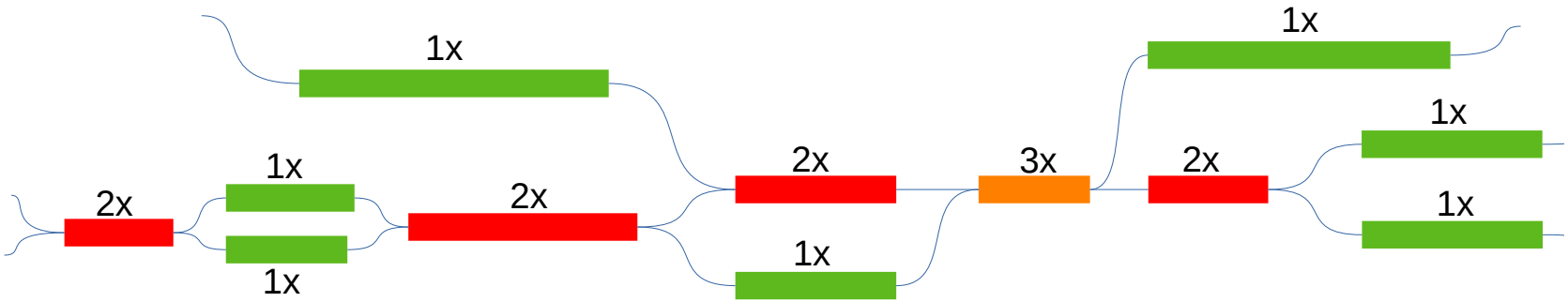
- Spreading the multiplicity

Algorithm: inferring multiplicities



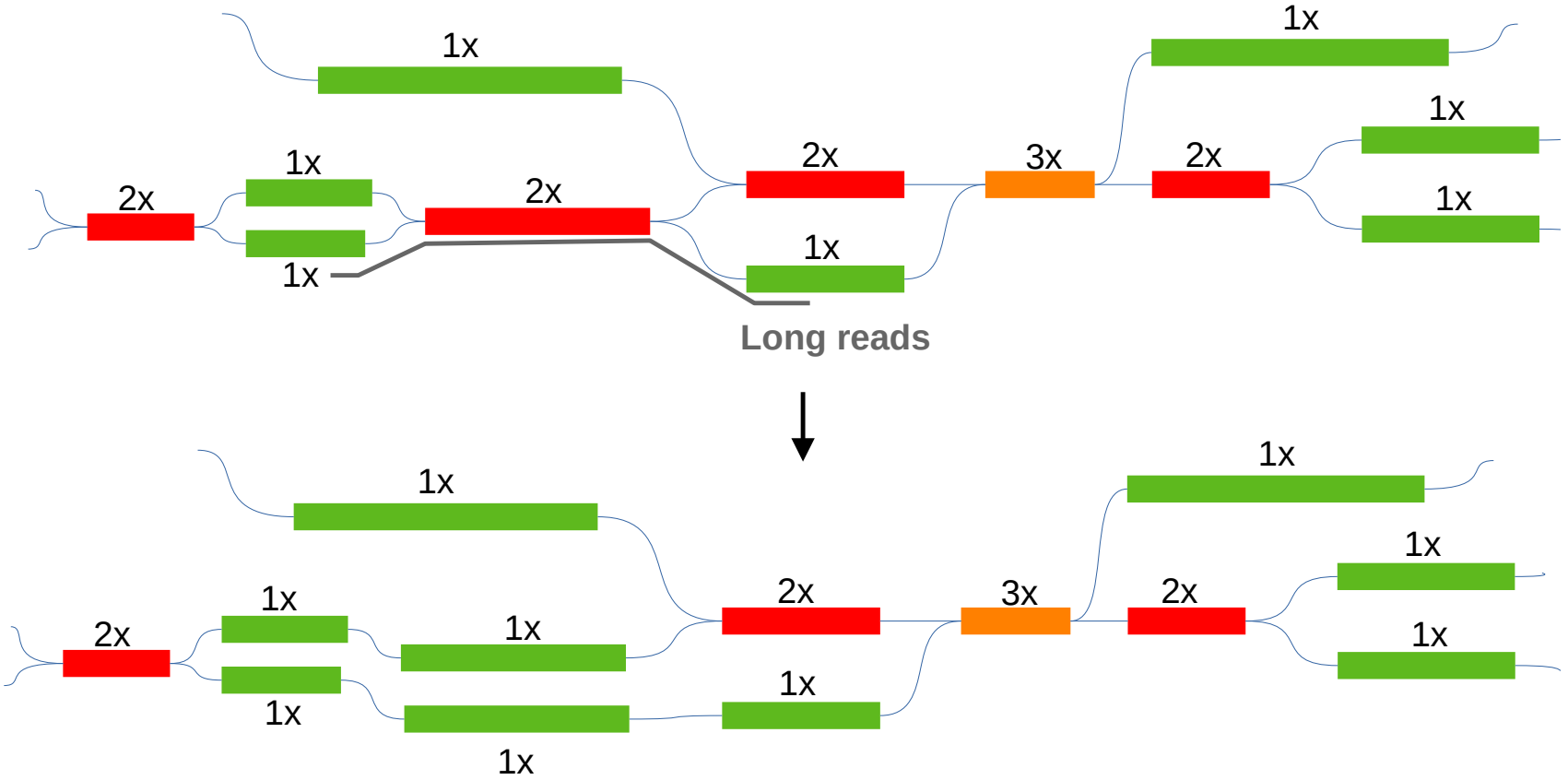
- Spreading the multiplicity

Algorithm: inferring multiplicities



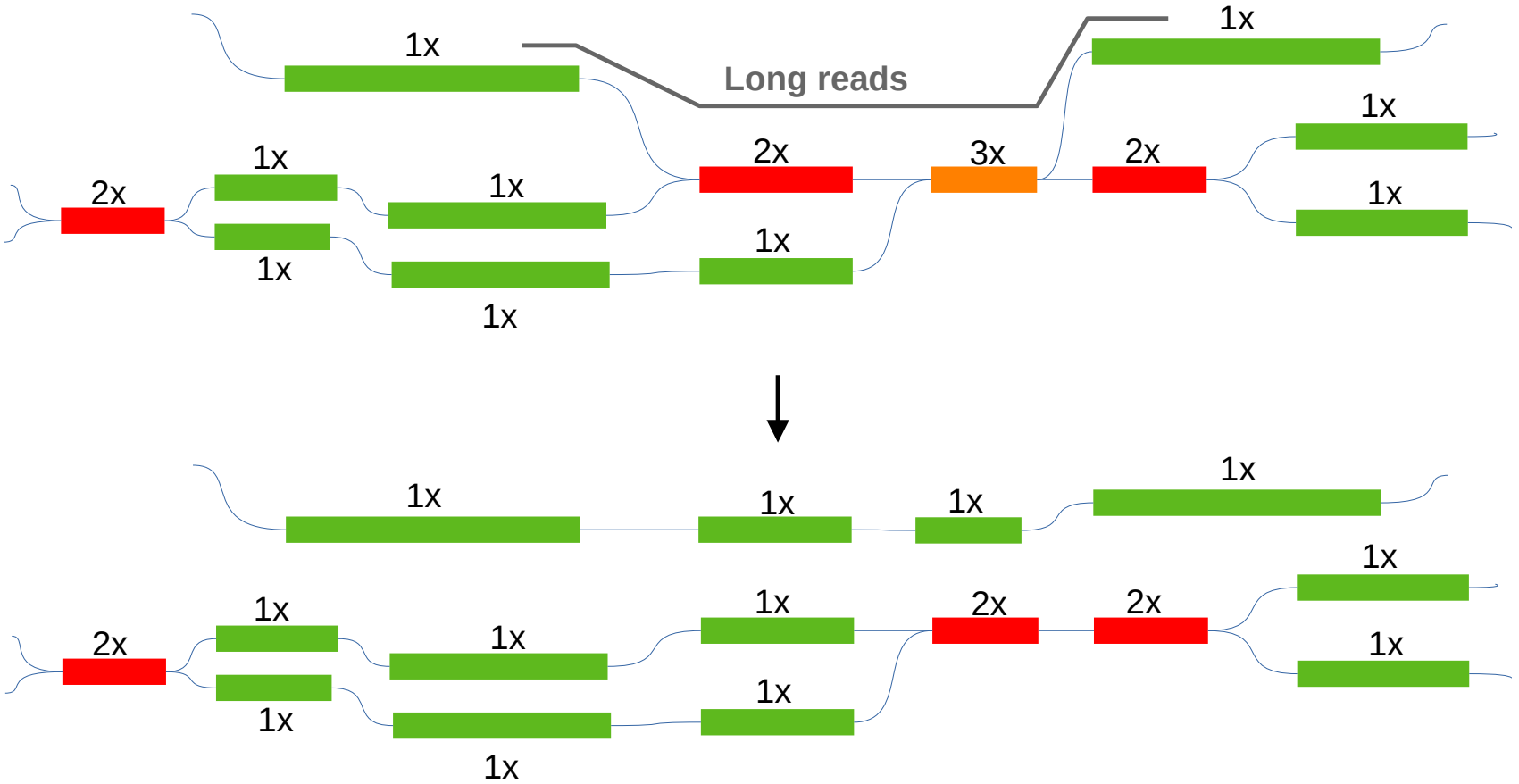
- Spreading the multiplicity

Algorithm: building bridges



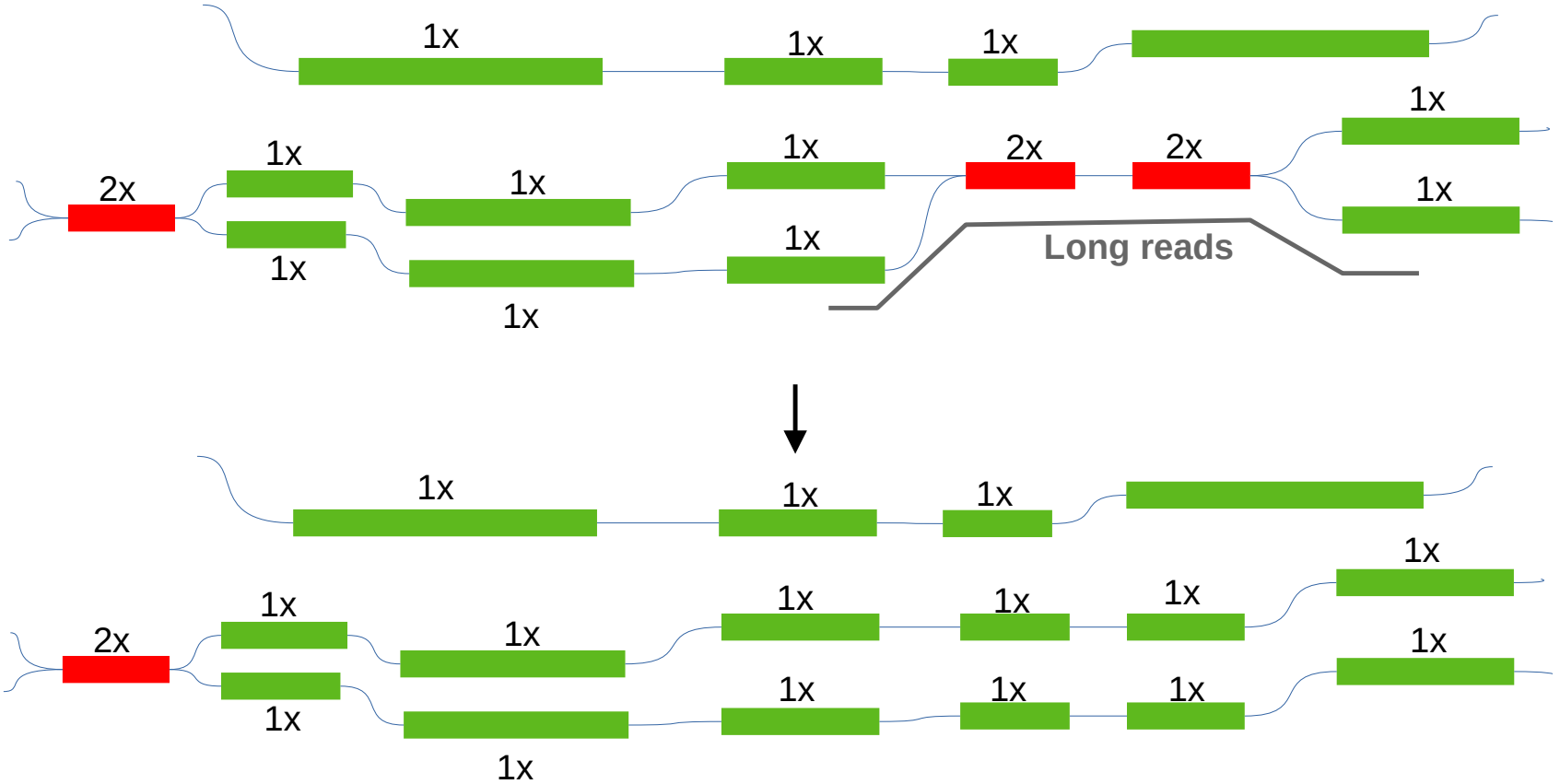
- We look at long reads building "bridges" between haploid contigs

Algorithm: building bridges



- We look at long reads building "bridges" between haploid contigs

Algorithm: building bridges



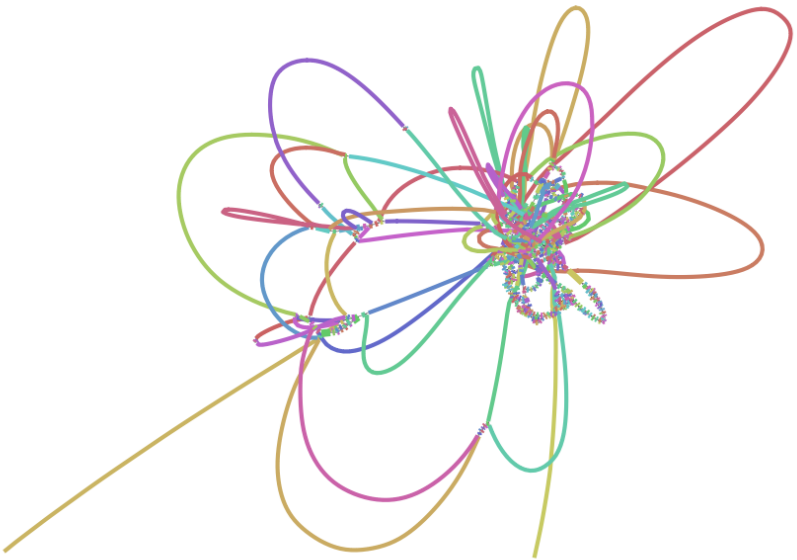
- We look at long reads building "bridges" between haploid contigs

Two test datasets

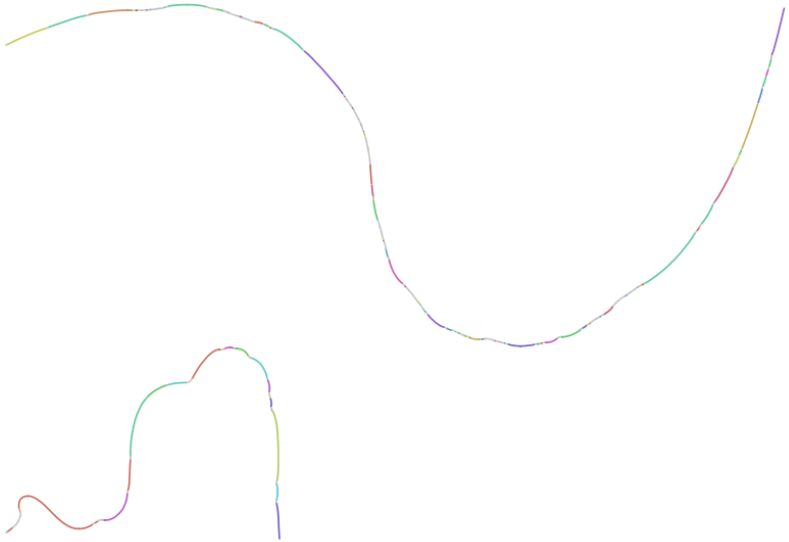
- Genome: *Escherichia coli*, Sakai strain
- Short Illumina reads simulated with IDBA-sim_reads
- Long PacBio HiFi reads simulated with Badread

- Genome: "diploid" *Escherichia coli*, one haplotype Sakai strain and one haplotype K12 strain
- Short Illumina reads simulated with IDBA-sim_reads
- Long PacBio HiFi reads simulated with Badread

GraphUnzip: haploid *E. coli*

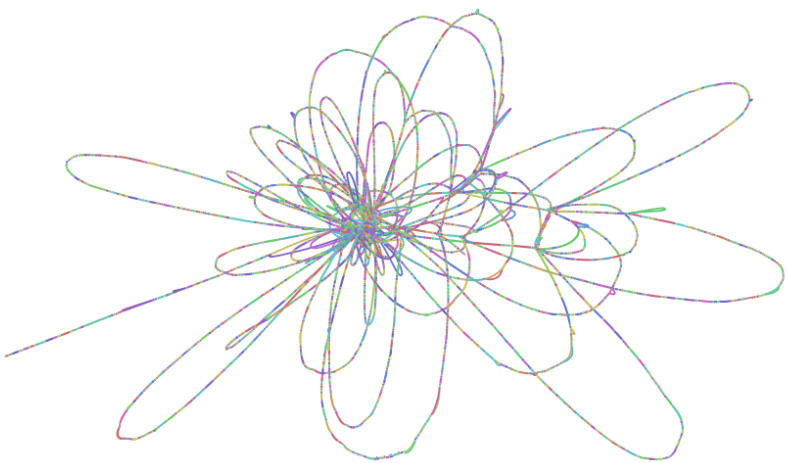


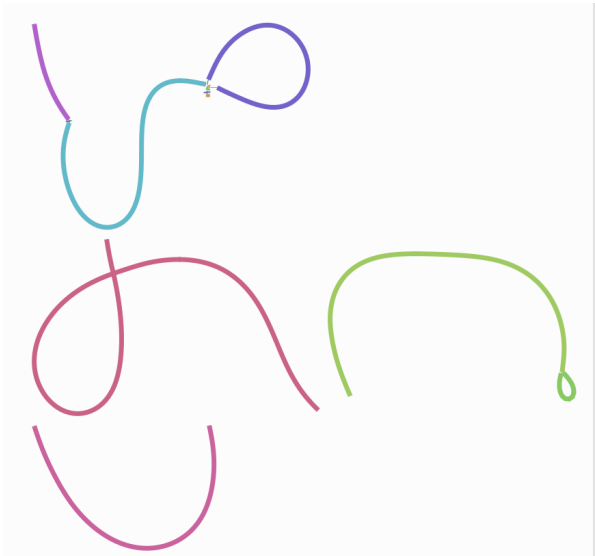
- SPAdes short-read assembly

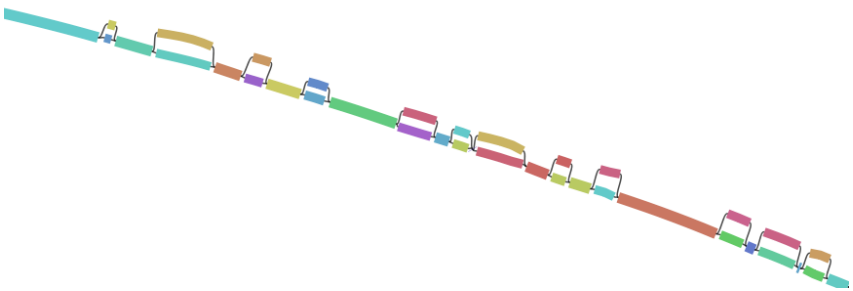


- Untangled with GraphUnzip
- 0 errors in untangling

GraphUnzip: diploid *E. coli*







- SPAdes short-read assembly

- Untangled with GraphUnzip
- 99.99% of the genome in 7 contigs
- 0 misassemblies, missing 2kbp

Benchmark description

- Scaffolding tools:
  - LongStitch
  - SLR
  - npScarf
- Hybrid assembly tools:
  - OPERA-MS (metagenome assembler)
  - Unicycler

Results

Assembly metrics: haploid assembly (5.6 Mb)

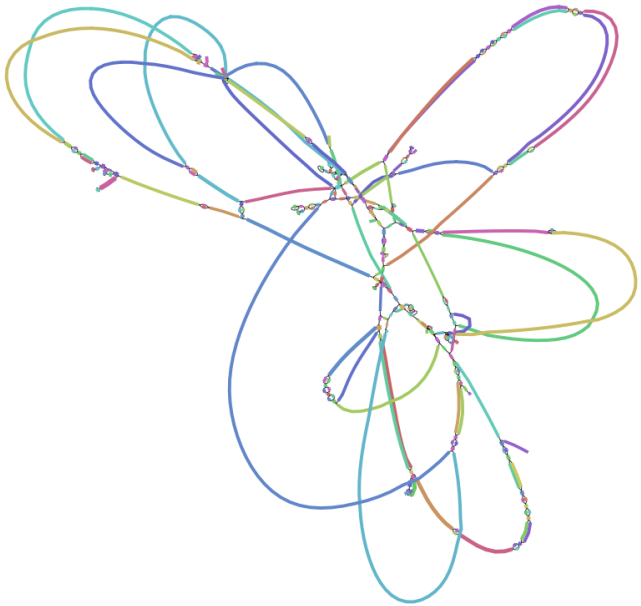
	completeness (%)	Misassemblies	N50 (Mb)	N90 (Mb)	Size (Mb)
LongStitch	96	**0**	1.5	0.14	5.9
SLR	96	**0**	0.15	0.006	5.6
npScarf	99	17	3.1	0.26	5.8
OPERA-MS	96	**0**	0.15	0.006	5.6
Unicycler	**100**	**0**	**5.5**	**5.5**	5.5
GraphUnzip	**100**	**0**	1.8	1.7	5.8

Assembly metrics: diploid assembly (10.3 Mb)

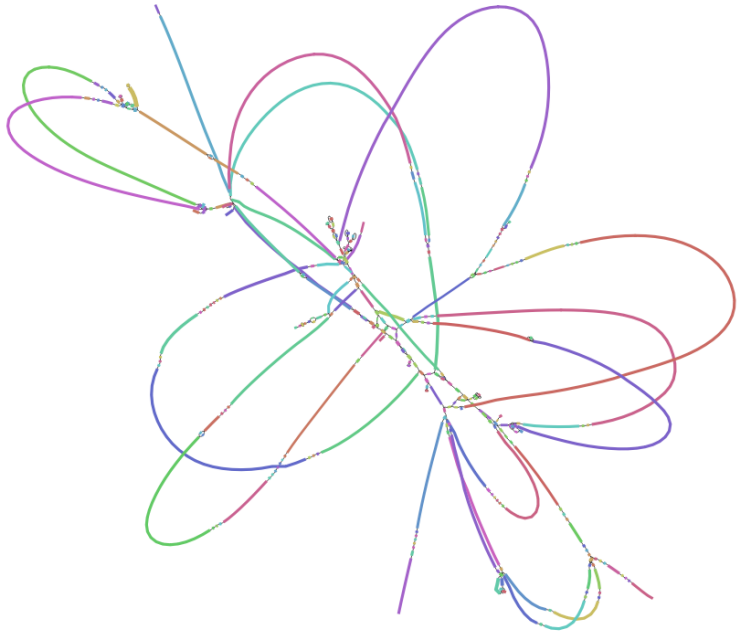
	completeness (%)	Misassemblies	N50 (Mb)	N90 (Mb)	Size (Mb)
LongStitch	76	**0**	0.002	0.0005	9.5
SLR	82	44	0.115	0.0005	10.8
npScarf					
OPERA-MS	87	492	0.41	0.0006	15.1
Unicycler	70	43	0.64	0.006	7.4
GraphUnzip	**100**	**0**	**1.5**	**0.29**	10.3

- GraphUnzip is clearly the best on the diploid assembly

Unzipping of *Adineta vaga* with long reads



HiFi assembled with hifiasm



Assembly after GraphUnzip with Nanopore

N50 : 6.3 Mb $\rightarrow$ 10.3 Mb

- GraphUnzip can also be used to combine HiFi and Nanopore

Pros and cons of GraphUnzip

Limitations of GraphUnzip:

- Blind trust in the input assembly
- Haplotypes not explicitly separated

Strengths of GraphUnzip:

- Very modular, can be used with any assembler
- Fast and memory-efficient (all examples ran on laptop)
- Naive: makes no assumption on ploidy, parameter-free

Take-home message

- GraphUnzip is the first standalone software to untangle assembly graphs using long-range data
- GraphUnzip can use Hi-C or long reads to do so
- Available at `github.com/nadegeguiglielmoni/GraphUnzip`

Acknowledgements

- Nadège Guiglielmoni and Jean-François Flot for their guidance
- The EEB-EBE and GenScale teams
- This work has been supported by a grant from the Société Française de Bioinformatique