ASSEMBLY OF HETEROZYGOUS GENOMES

Antoine Limasset - Camille Marchet - Jean François Flot*
Pierre Peterlongo*

Sequencing, Finishing and Analysis in the Future
(SFAF 2017, Santa Fe)

* co-last authors

Motivations

Assembling a diploid

- chrom. mom:
- chrom. dad:
- Input = reads

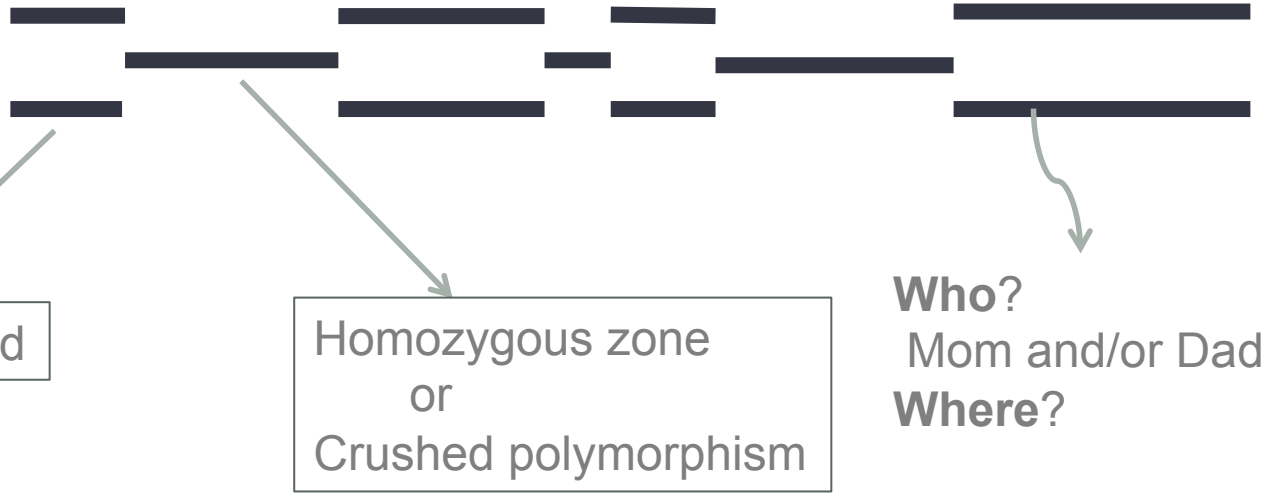
- What we want:

Assembling a diploid

- chrom. mom:
- chrom. dad:

- What we want:

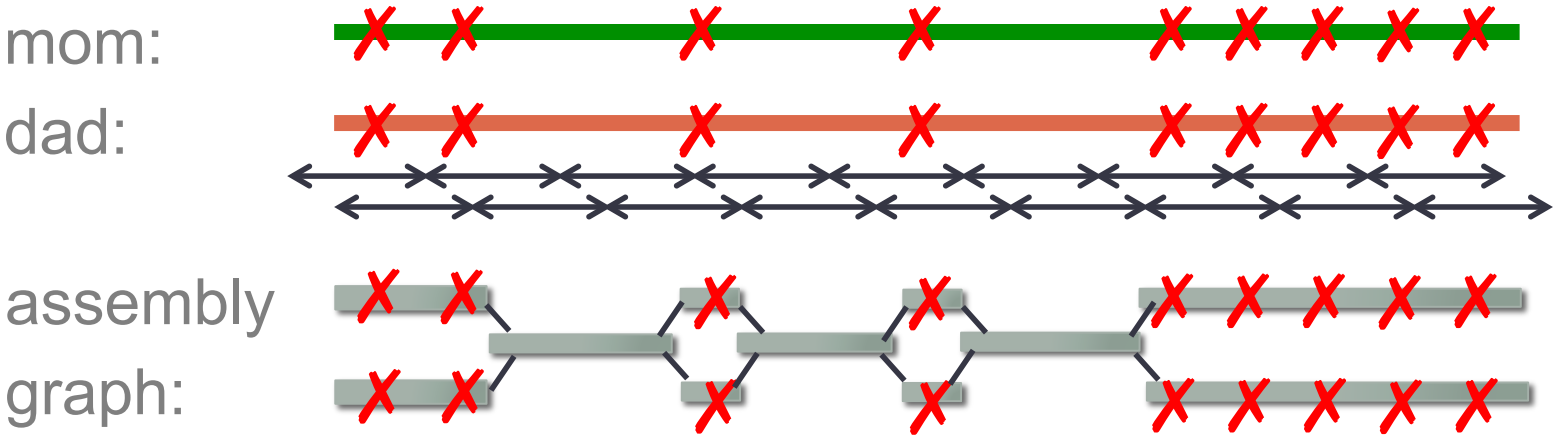
- What we got:



What happens in the “assembly graph”?

✗ Polymorphism

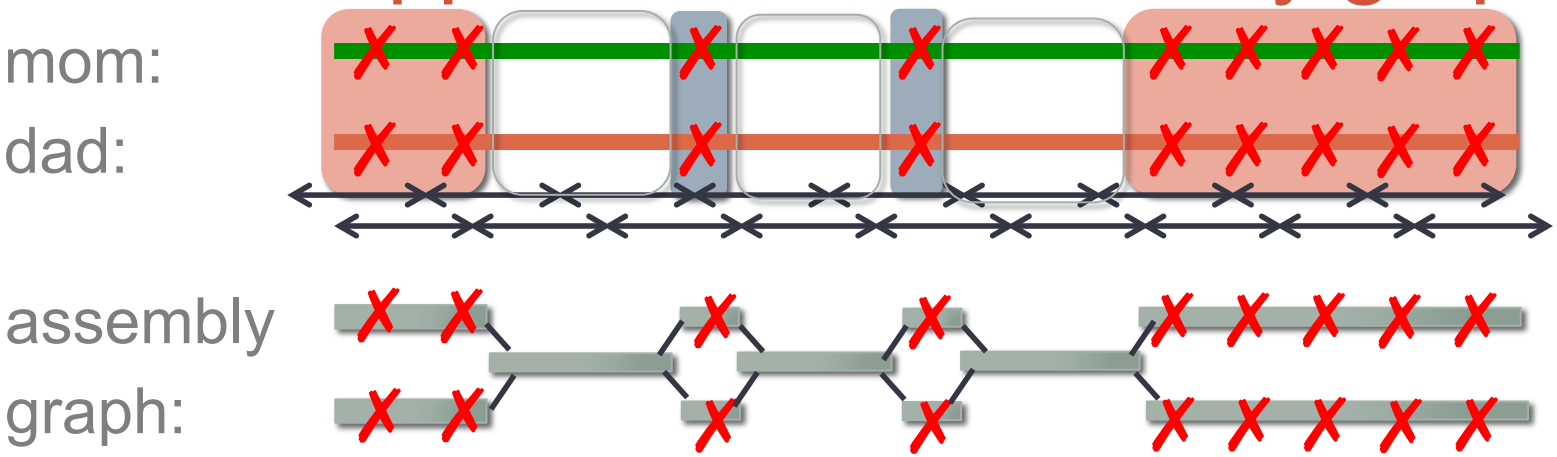
What happens in the “assembly graph”?



Polymorphism

Size of the information used in graph:
(read, kmer, long reads, read pairs…)

What happens in the “assembly graph”?



Polymorphism

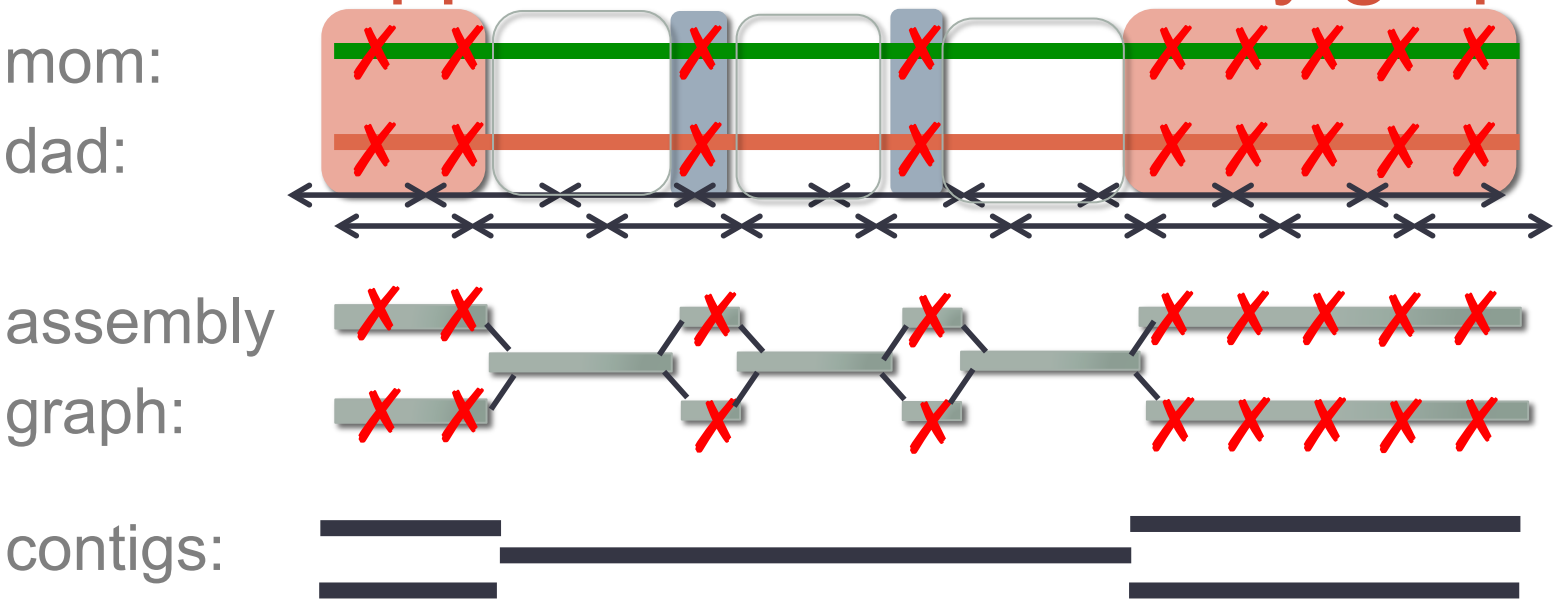
Size of the information used in graph:
(read, kmer, long reads, read pairs…)

“phased” polymorphism

“isolated” polymorphism

non-polymorphic regions

What happens in the “assembly graph”?



Polymorphism

Size of the information used in graph:
(read, kmer, long reads, read pairs…)

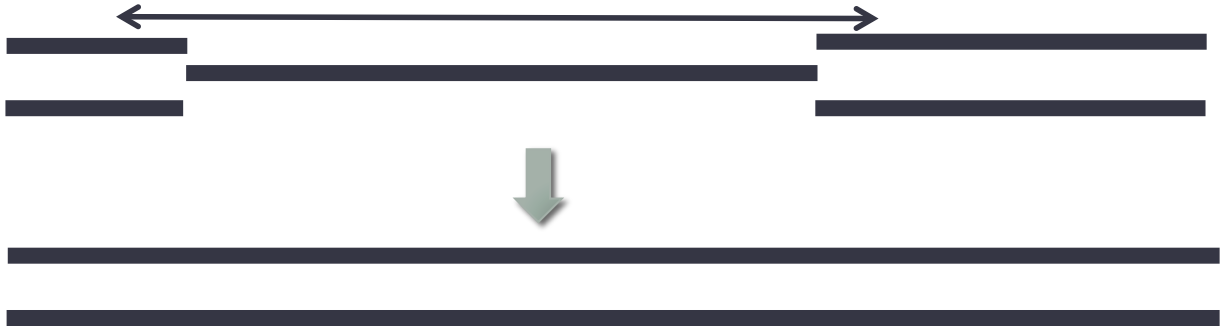
“phased” polymorphism

“isolated” polymorphism

non-polymorphic regions

size of the information used in graph:
(read, kmer, long reads, read pairs…)

With longer information, we could "*phase" the polymorphism*



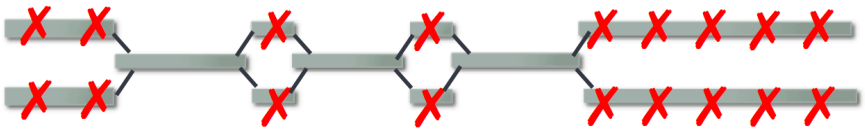
DECISIVE

Assembly paradigm dichotomy

string graph

read length info

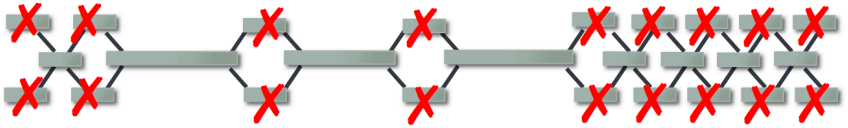
phase polymorphism



de Bruijn graph

k-mer length info

“bad” phasing

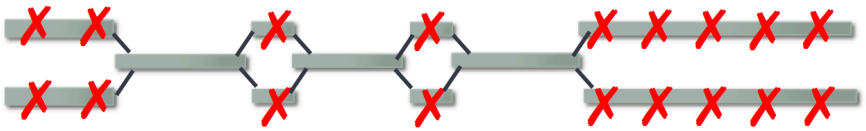


Assembly paradigm dichotomy

string graph

read length info

phase polymorphism



scaling issues

≈ quadratic

de Bruijn graph

k-mer length info

“bad” phasing

scales large instances

≈ linear

Assembly paradigm dichotomy

string graph

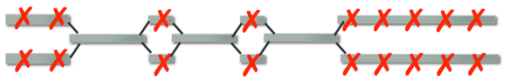
- read length info
  - phase polymorphism
- scaling issues

de Bruijn graph

- k-mer length info
  - “bad” phasing
- scales large instances

Our proposal, BWISE:

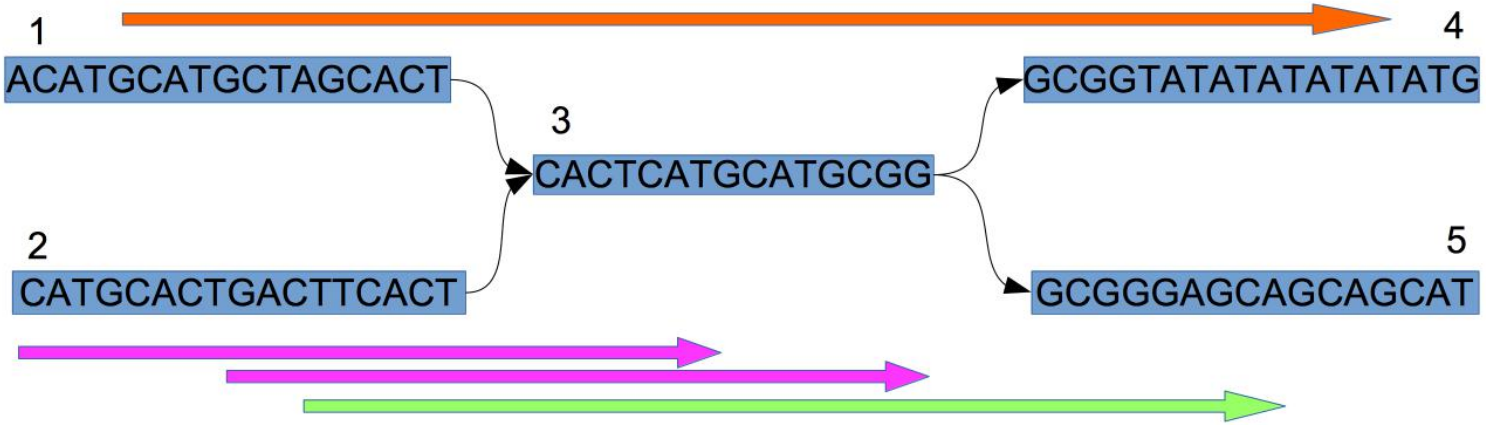
- read length info (and beyond)
  - phase polymorphism



- scales large instances

Algorithm overview

Main idea 1/2 – map reads on dBG



Super Reads (SR) produced by the mapped reads:

Shows the path 1,3,4:	ACATGCATGCTAGCACTCATGCATGCGGTATATATATATATG
Both show the path 2,3:	CATGCACTGACTTCACTCATGCATGCGG
Shows the path 2,3,5:	CATGCACTGACTTCACTCATGCATGCGGGAGCAGCAGCAT

Super reads:

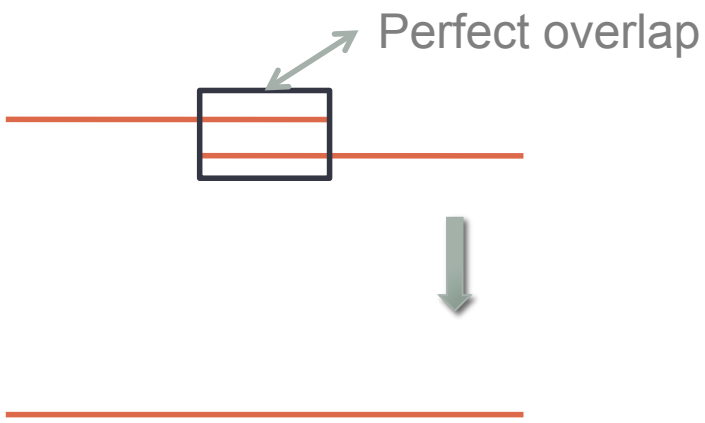
longer (or equal) than reads

exact paths from dBG.

Main idea 2/2 – Super reads graph

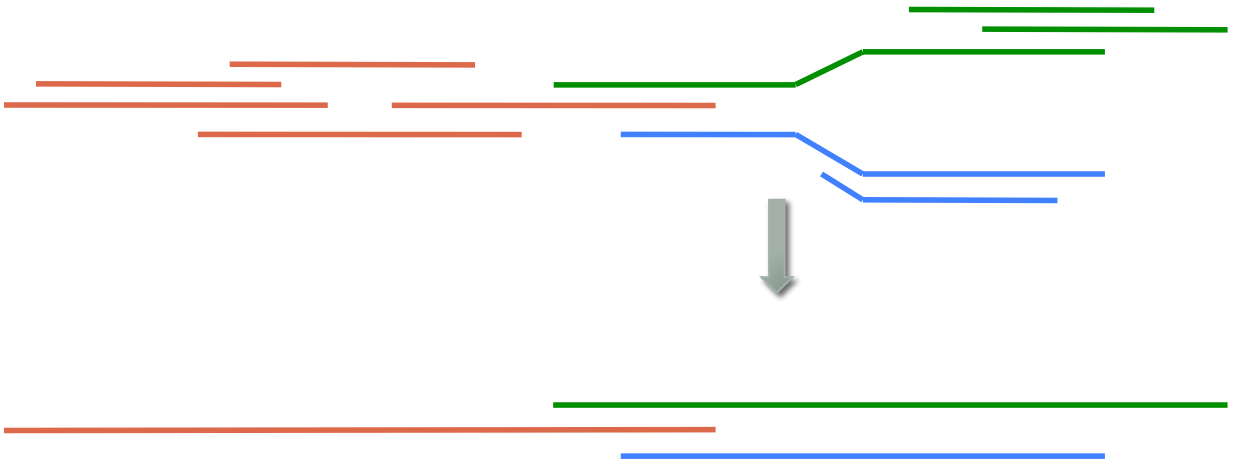
- Connect super reads (exact prefix/suffix links)
- Remove redundancies
- Detect simple paths

  Consequence: increases again SR size



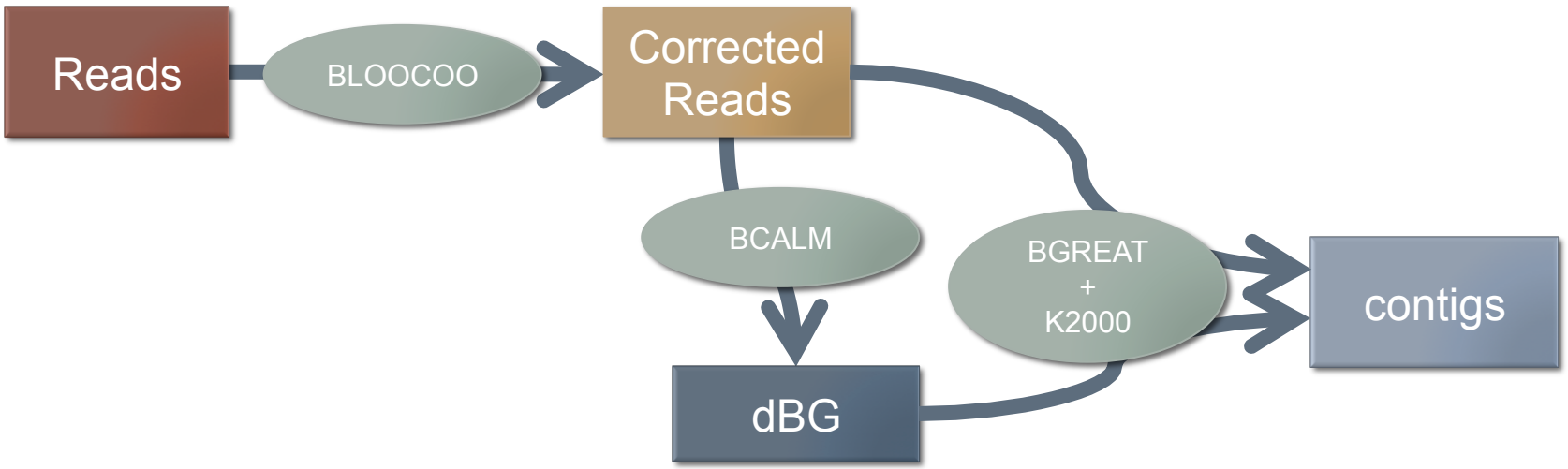
Main idea –
Super reads graph ~ string graph

- Connect super reads (exact prefix/suffix links)
- Remove redundancies
- Detect simple paths



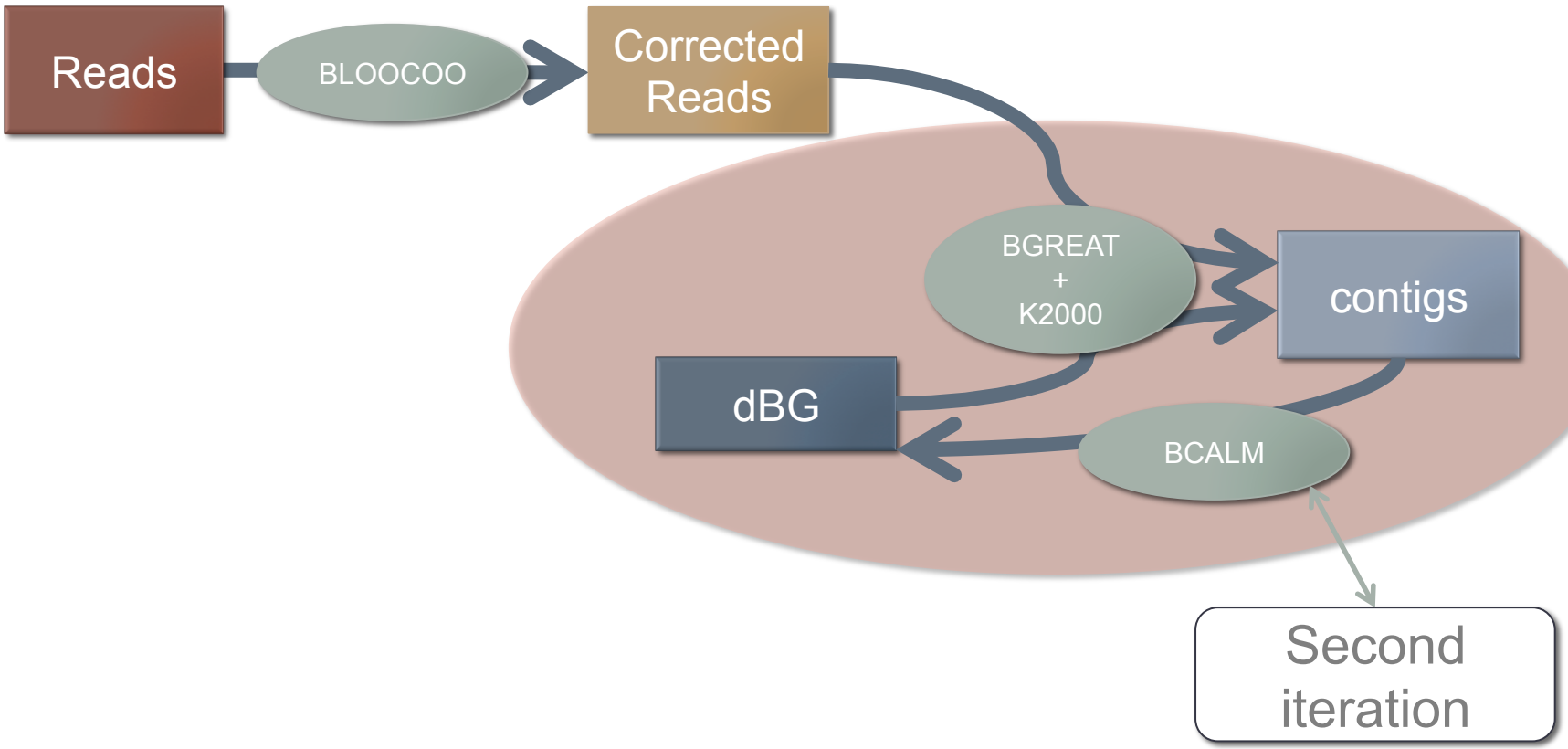
Output contigs

In practice



[BLOOCOO] Benoit, G., Lavenier, D., Lemaitre, C., Rizk, G.: Bloocoo, a memory efficient read corrector (ECCB 2014)
[BCALM] Chikhi, R., Limasset, A., et al.: On the representation of de bruijn graphs (Lecture Notes in Computer Science, vol. 8394)
[BGREAT] Limasset, A., Cazaux, B., Rivals, E., & Peterlongo, P.: Read mapping on de Bruijn graphs. BMC Bioinformatics (2016)

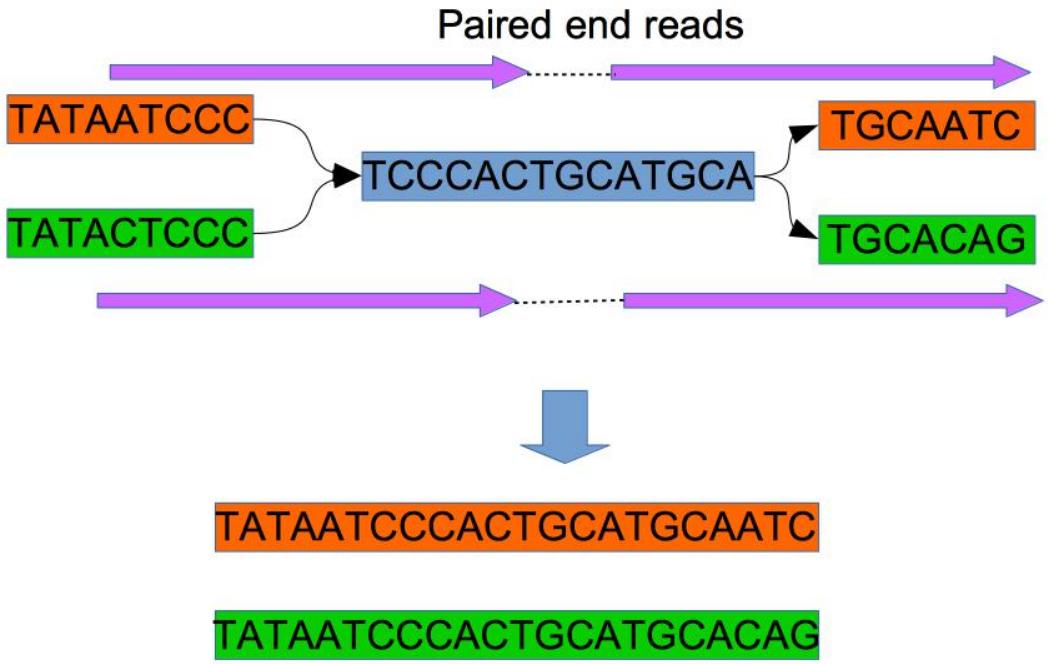
In practice



[BLOOCOO] Benoit, G., Lavenier, D., Lemaitre, C., Rizk, G.: Bloocoo, a memory efficient read corrector (ECCB 2014)
[BCALM] Chikhi, R., Limasset, A., et al.: On the representation of de bruijn graphs (Lecture Notes in Computer Science, vol. 8394)
[BGREAT] Limasset, A., Cazaux, B., Rivals, E., & Peterlongo, P.: Read mapping on de Bruijn graphs. BMC Bioinformatics (2016)

Why a second iteration

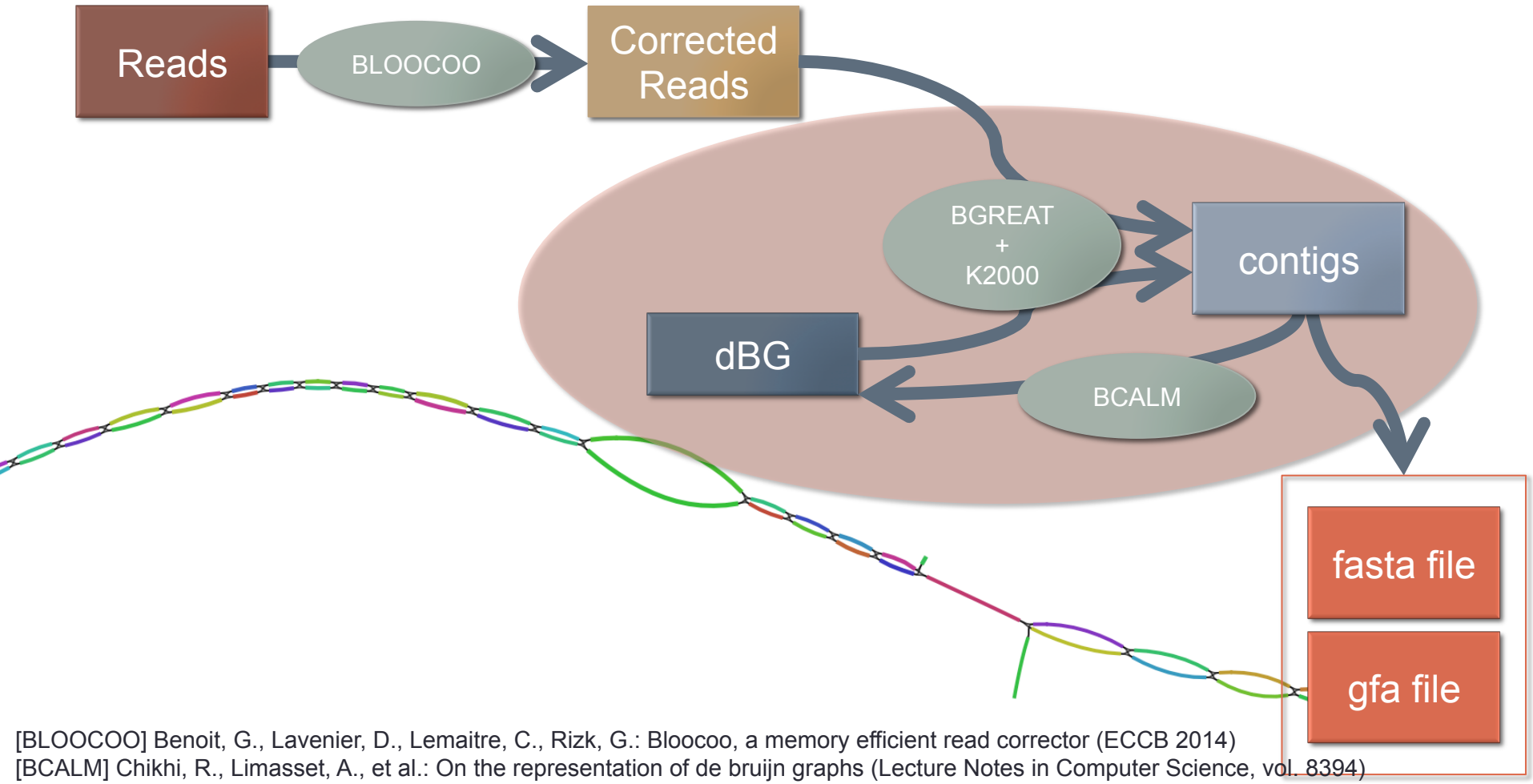
- Super reads → dBG
- Map back read pairs



Super reads:

longer (or equal) than **pairs of reads**
exact paths from dBG

In practice



[BLOOCOO] Benoit, G., Lavenier, D., Lemaitre, C., Rizk, G.: Bloocoo, a memory efficient read corrector (ECCB 2014)
[BCALM] Chikhi, R., Limasset, A., et al.: On the representation of de bruijn graphs (Lecture Notes in Computer Science, vol. 8394)
[BGREAT] Limasset, A., Cazaux, B., Rivals, E., & Peterlongo, P.: Read mapping on de Bruijn graphs. BMC Bioinformatics (2016)

(preliminary) results

Bwise parameters : k_1=51
k_2=201

Haploid simulations

	E. coli	C. elegans	Human
SPAdes	contigs: 71 N50: 178,400	contigs: 6,550 N50: 103,128	
Platanus	contigs: 272 N50: 133,252	contigs: 21,413 N50: 103,128	
Bwise	contigs: 56 N50: 210,994	contigs: 2,527 N50: 122,287	contigs:132,272 N50: 74,148

0.6GB RAM; 7min

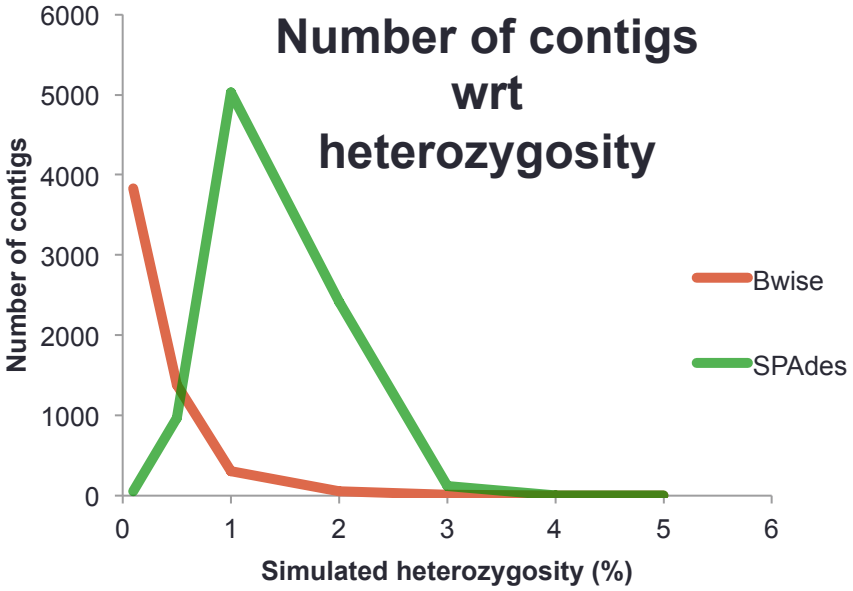
50GB RAM; <48h

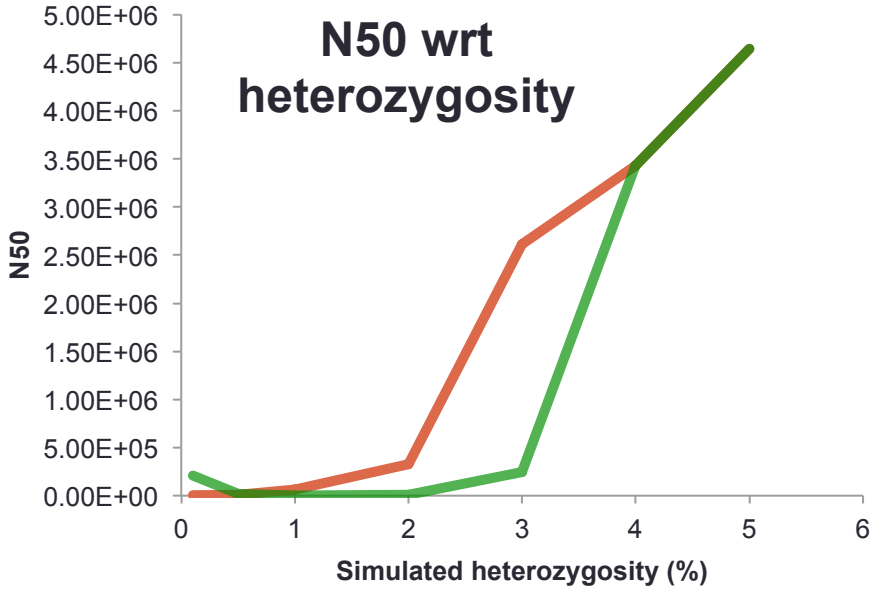
Bwise & Platanus ≈ 2x more misassemblies than SPAdes

Simulated 100x Miseq 2*250 reads 800 fragment size 1% error.

Bwise parameters : k_1=51
k_2=201

E. coli, simulated heterozygosity



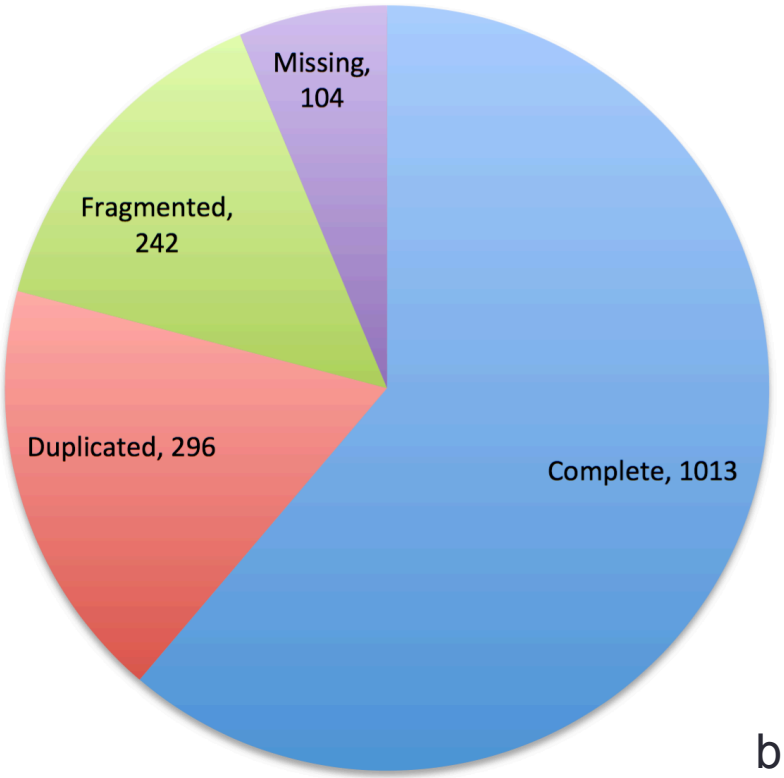


Simulated 100x Miseq 2*250 reads 800 fragment size 1% error.

Melinaea marsaeus Assembly

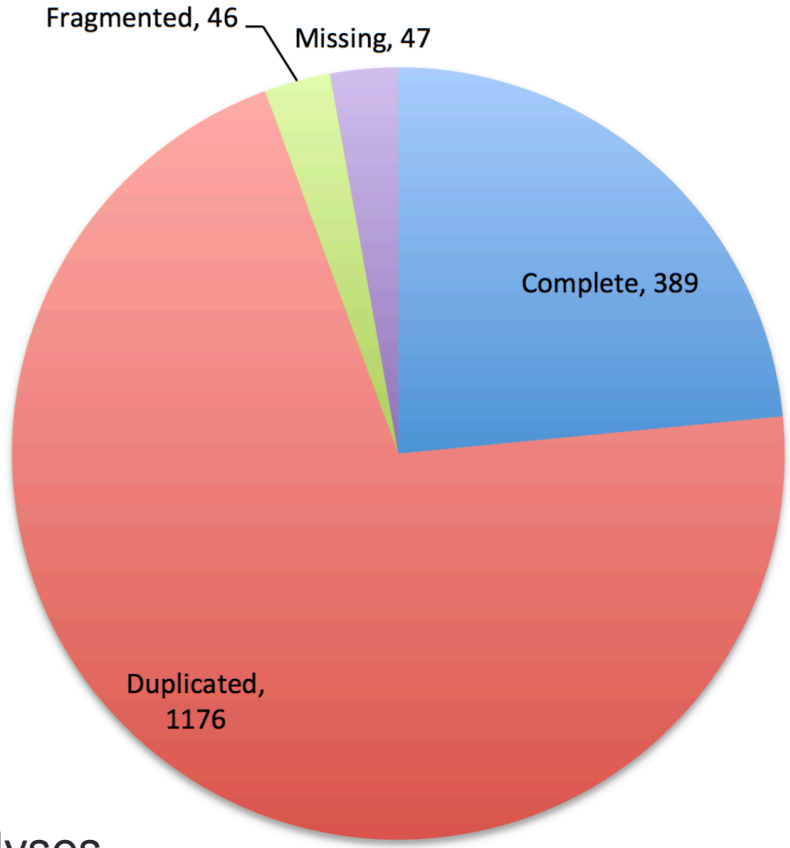
Platanus:

- N50: 8,129 bp



Bwise:

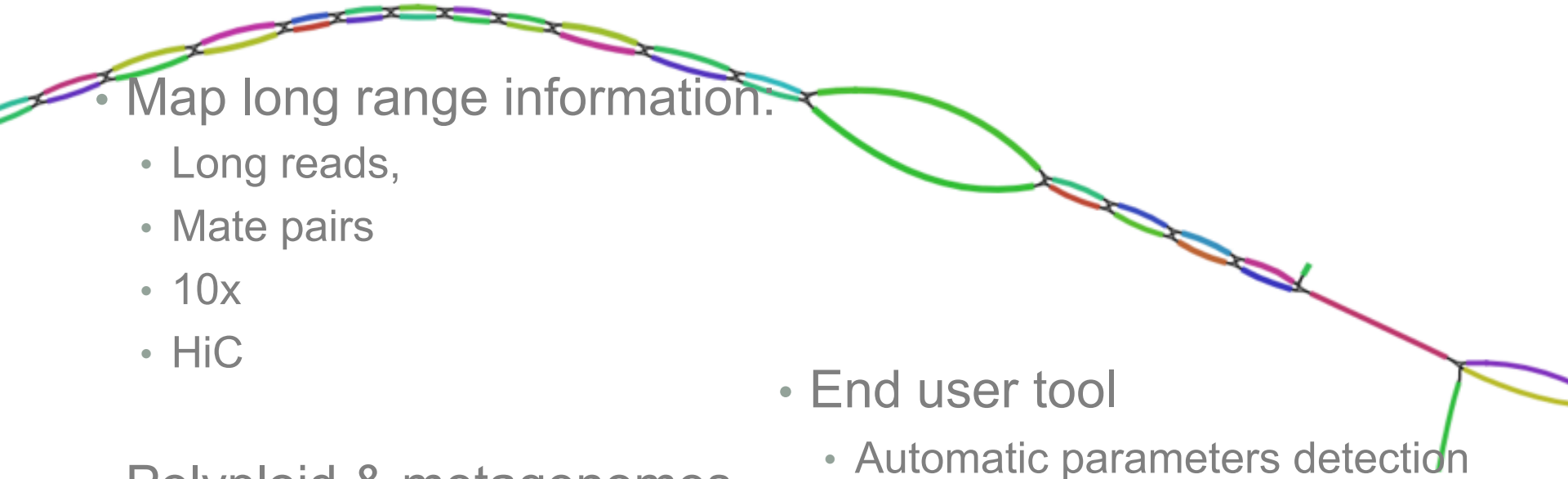
- N50: 15,093 bp



busco analyses

Future
&
(my) questions

Future



- Map long range information:
  - Long reads,
  - Mate pairs
  - 10x
  - HiC
- Polyploid & metagenomes
- Deal with low heterozygosity
- End user tool
  - Automatic parameters detection
  - Pipeline factorization
  - Output representation?

Thanks

Tool: https://github.com/Malfoy/BWISE

Contacts: bwise@inria.fr

Bwise parameters : k_1=51
k_2=201

E. coli, simulated heterozygosity

Simulated 100x Miseq 2*250 reads 800 fragment size.

BWISE : diploid assembly
SPAdes : haploid assembly

	0.1%	0.5%
SPAdes	contigs: 55 N50: 210,905	contigs: 973 N50: 17,487
Platanus	contigs: 926 N50: 40,837	contigs: 2,224 N50: 11,128
Bwise	contigs 3,835 N50: 4,520	contigs: 1,370 N50: 10,390

	1%	3%	5%
SPAdes	contigs: 5,027 N50: 1,827	contigs: 121 N50: 246,674	contigs: 3 N50: 4,639,654
Platanus	contigs: 2,887 N50: 6,526	contigs: 1,614 N50: 10,019	contigs: 711 N50: 22,275
Bwise	contigs 307 N50: 64,217	contigs: 8 N50: 2,610,736	contigs: 2 N50: 4,639,654