HairSplitter: assembling long reads in an unknown number of haplotypes

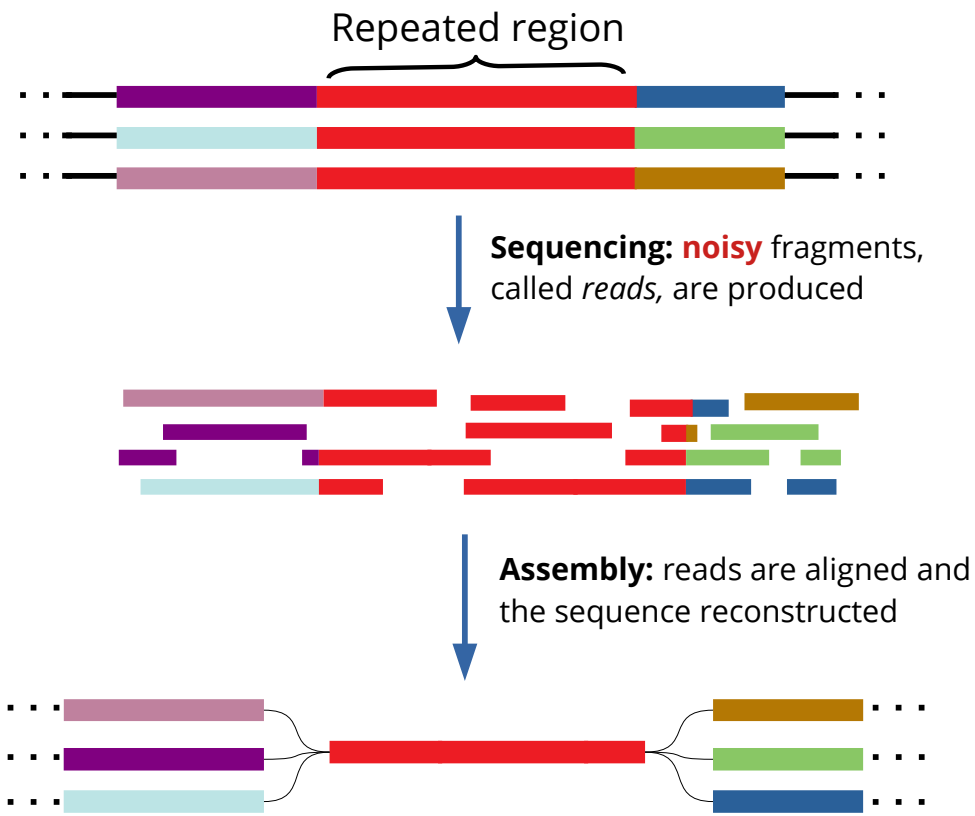
Roland Faure[1,2], Jean-François Flot[1], Dominique Lavenier[2]

[1]Université libre de Bruxelles (ULB)

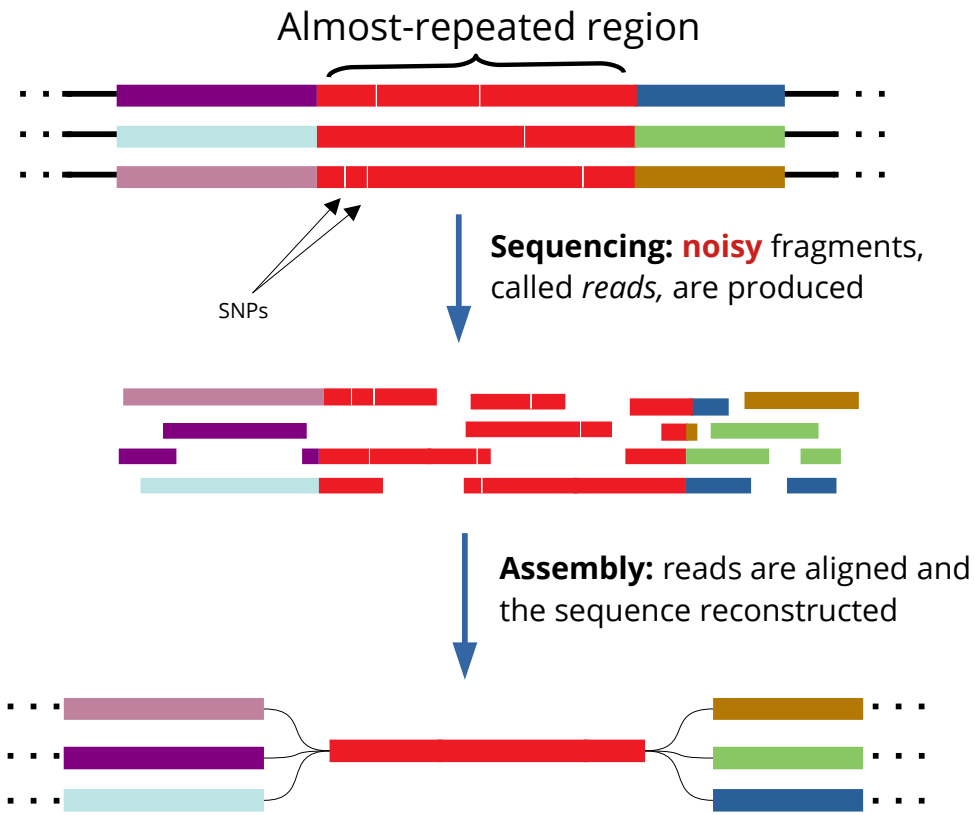
[2]Université de Rennes, IRISA

November 2022

Genome assembly

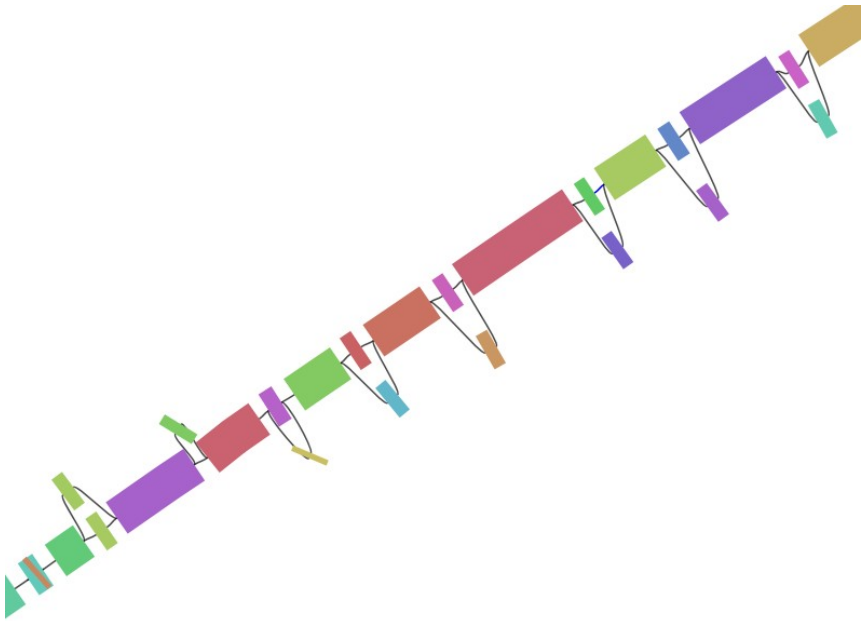


Genome assembly: similar regions get collapsed



- When divergence is small compared to the error rate of reads, SNPs are discarded as errors

Genome assembly: similar regions get collapsed



Screenshot of the Flye assembly of diploid *Adineta vaga*

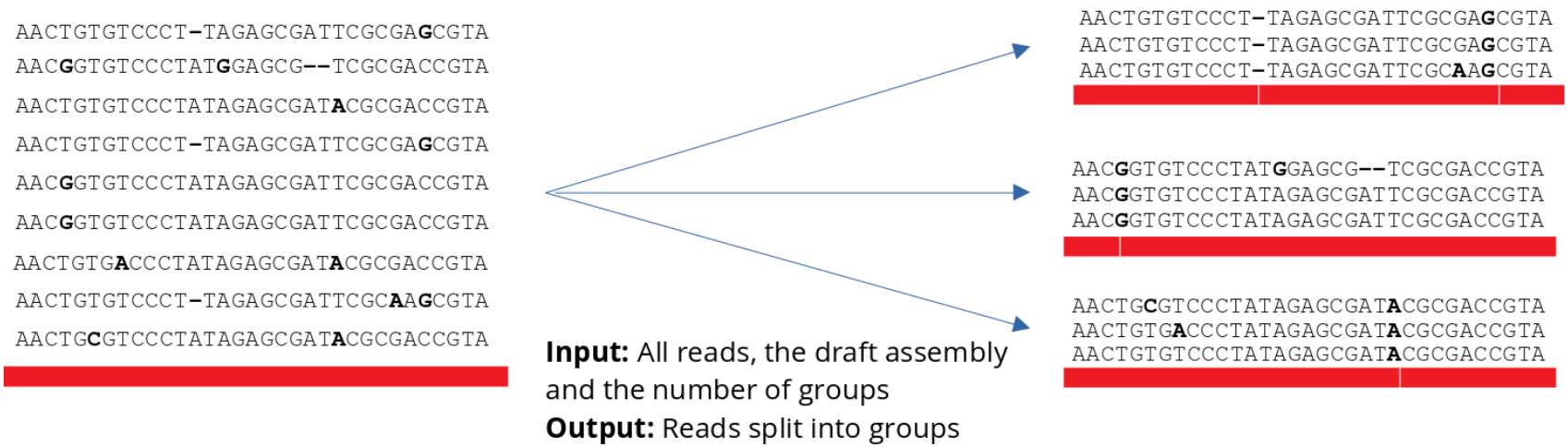
- Loss of heterozygous information!

Obtaining uncollapsed assemblies

- We want to recover the lost diversity

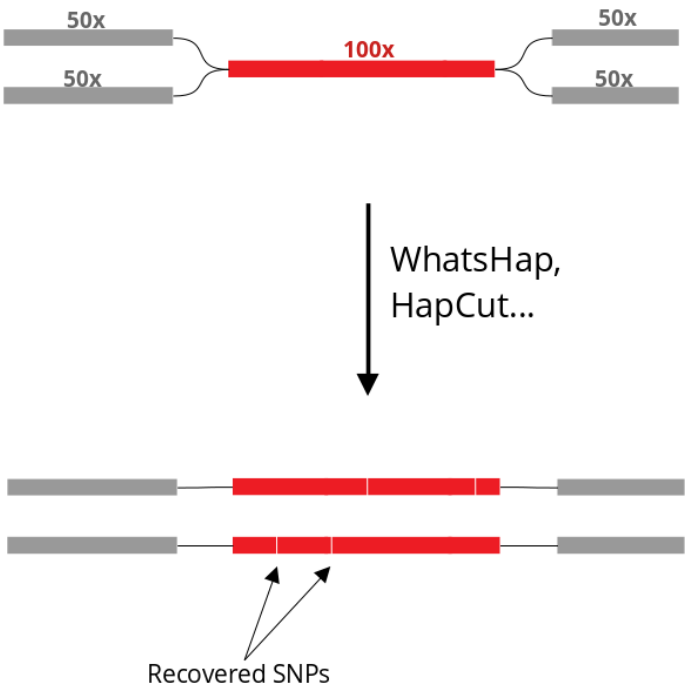
Obtaining uncollapsed assemblies

- We want to recover the lost diversity
- State of the art (long reads): phase the collapsed contigs using HapDup, WhatsHap, HapCut, H-PopG...

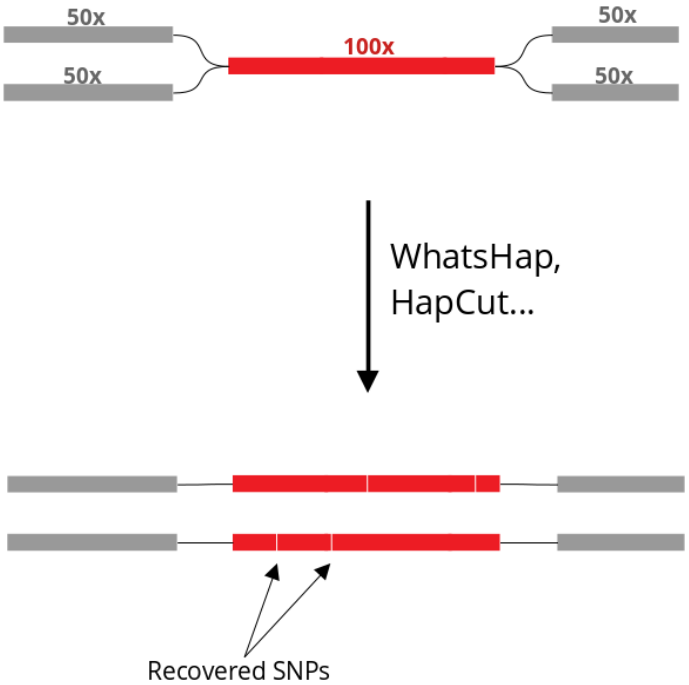


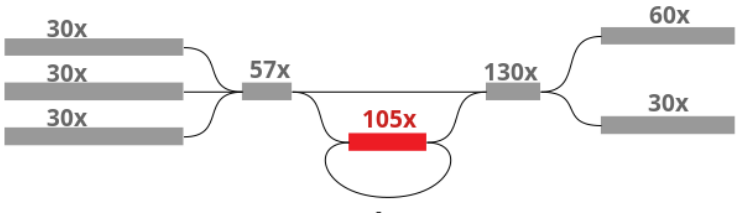
Let's recover the lost diversity!

Let's recover the lost diversity!

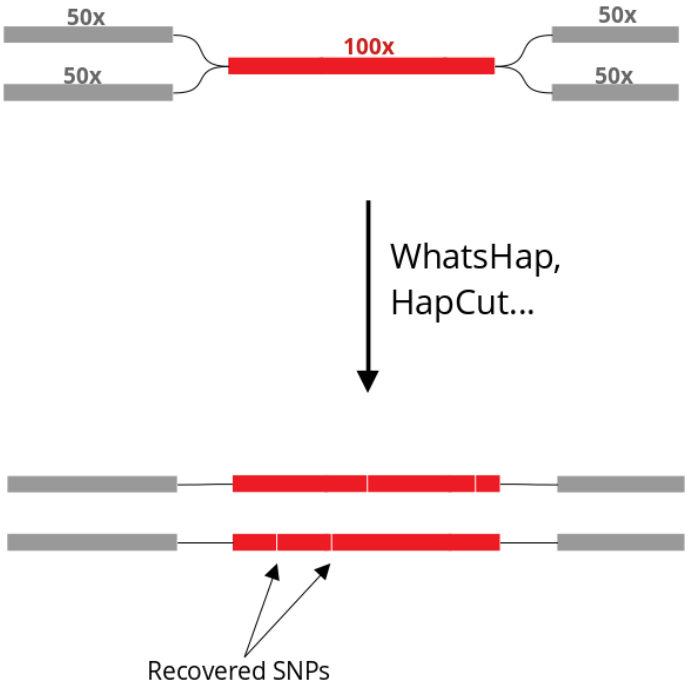


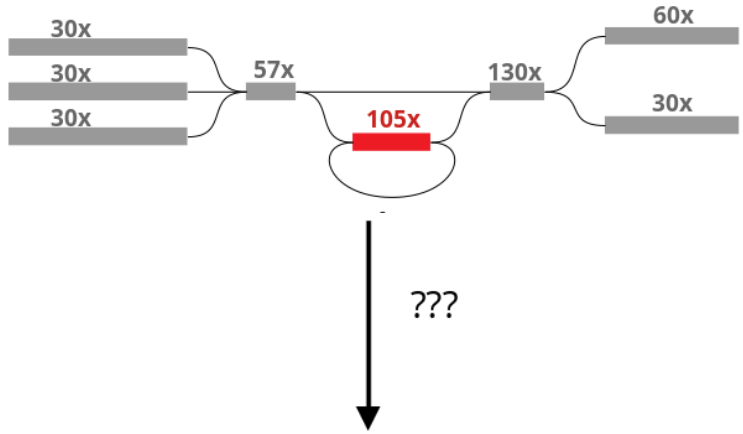
Let's recover the lost diversity!



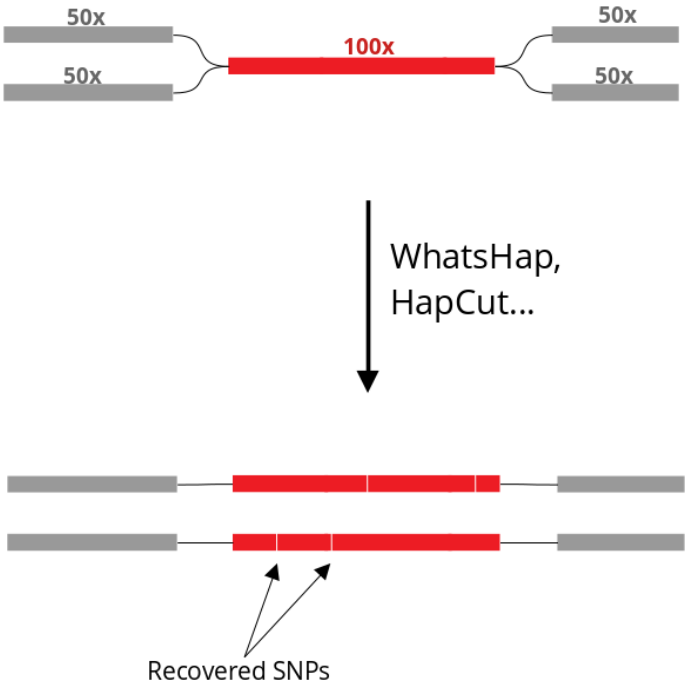


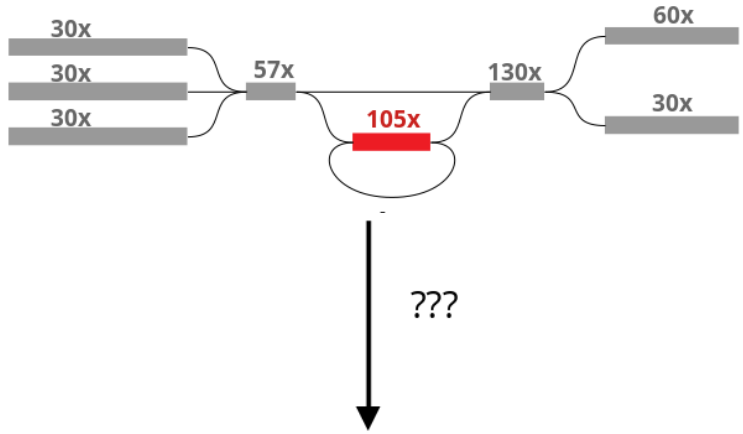
Let's recover the lost diversity!





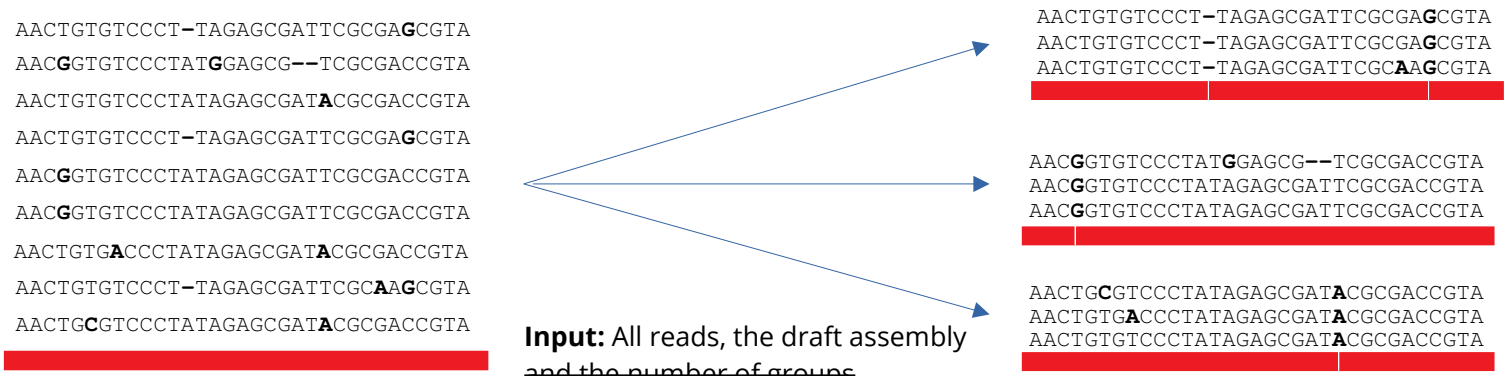
Let's recover the lost diversity!





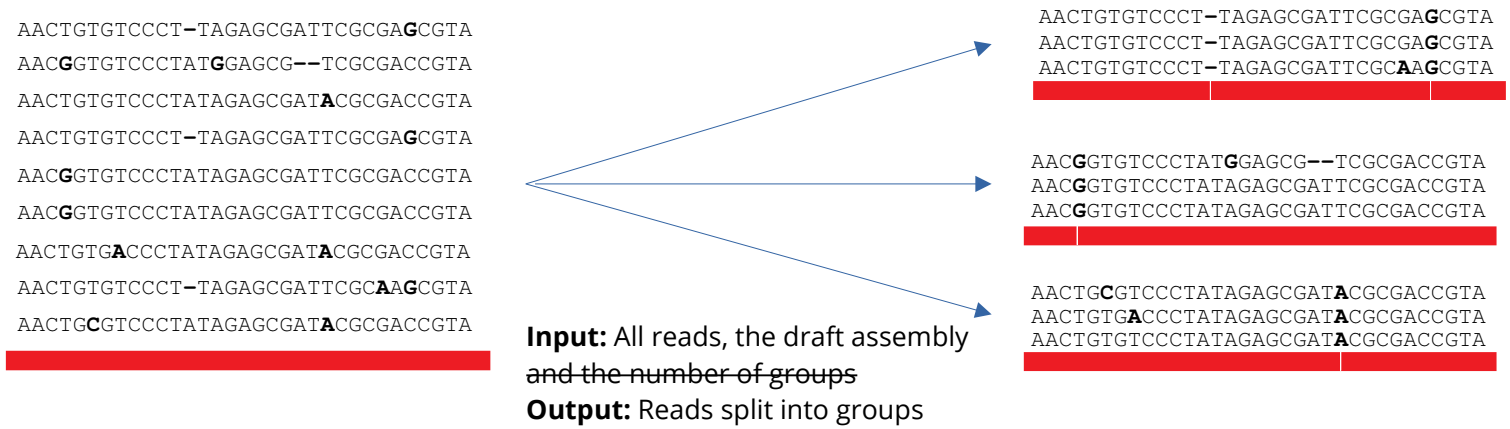
- In many cases the copy-number of a contig is unknown: polyploid genomes, repeats, metagenomic assemblies...

HairSplitter



Input: All reads, the draft assembly ~~and the number of groups~~
Output: Reads split into groups

HairSplitter



- *Hairsplitter:* a person who argues about differences that are too small to be important - *Britannica.com*

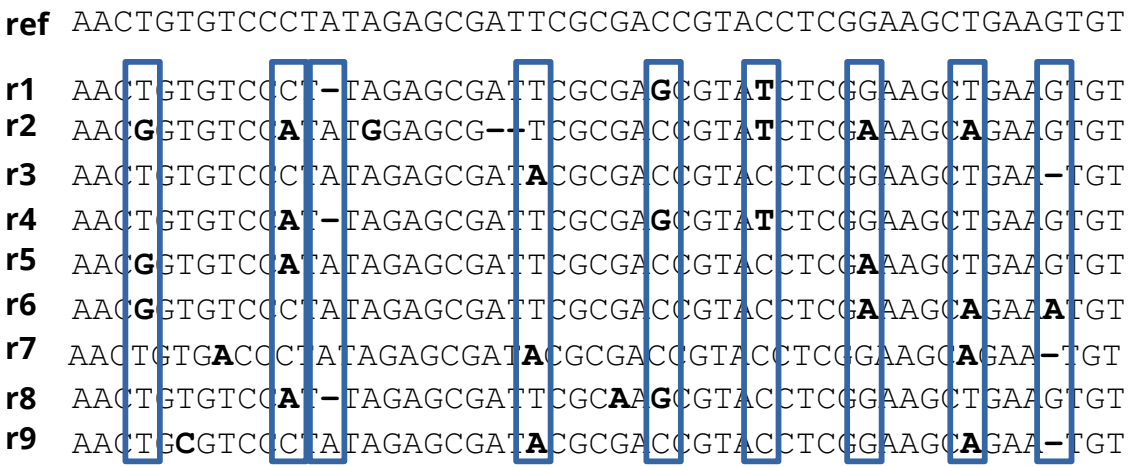
Splitting contigs: algorithm

```
ref AACTGTGTCCCTATAGAGCGATTCGCGACCGTACCTCGGAAGCTGAAGTGT

r1  AACTGTGTCCCT-TAGAGCGATTCGCGAGCGTATCTCGGAAGCTGAAGTGT
r2  AACGGTGTCCATATGGAGCG--TCGCGACCGTATCTCGAAAGCAGAAGTGT
r3  AACTGTGTCCCTATAGAGCGATACGCGACCGTACCTCGGAAGCTGAA-TGT
r4  AACTGTGTCCAT-TAGAGCGATTCGCGAGCGTATCTCGGAAGCTGAAGTGT
r5  AACGGTGTCCATATAGAGCGATTCGCGACCGTACCTCGAAAGCTGAAGTGT
r6  AACGGTGTCCCTATAGAGCGATTCGCGACCGTACCTCGAAAGCAGAAGTGT
r7  AACTGTGACCCTATAGAGCGATACGCGACCGTACCTCGGAAGCAGAA-TGT
r8  AACTGTGTCCAT-TAGAGCGATTCGCAAGCGTACCTCGGAAGCTGAAGTGT
r9  AACTGCGTCCCTATAGAGCGATACGCGACCGTACCTCGGAAGCAGAA-TGT
```

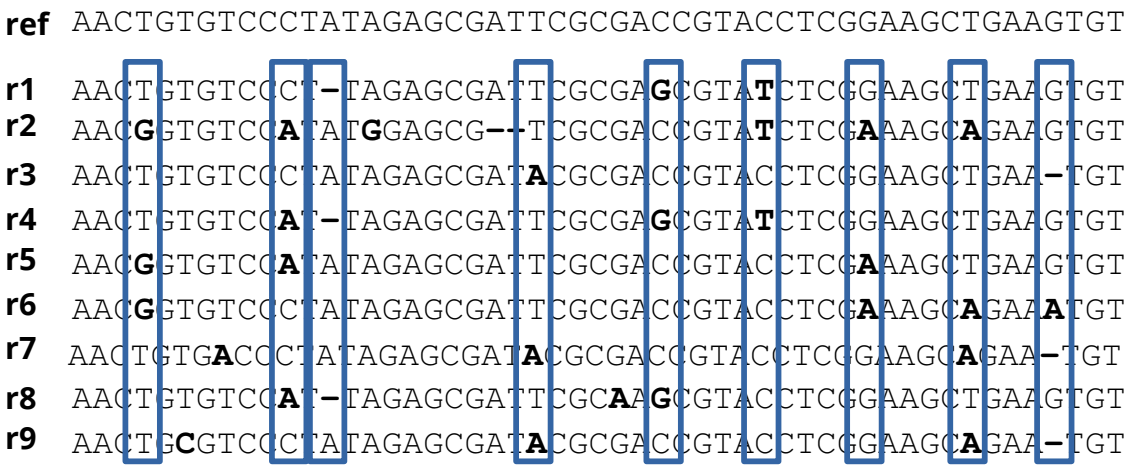
- Reads are aligned on the (collapsed) contig

Splitting contigs: algorithm



- Positions with high divergence are selected

Splitting contigs: algorithm



(TGTTGGTTT): {r2,r5,r6}, {r1,r3,r4,r7,r8,r9}
(CACAACCAC): {r2,r4,r5,r8}, {r1,r3,r6,r7,r9}
(-AA-AAA-A): {r1,r4,r8}, {r2,r3,r5,r6,r7,r9}
(TTATTTATA): {r3,r7,r9}, {r1,r2,r4,r5,r6,r8}
(GCCGCCCGC): {r1,r4,r8}, {r2,r3,r5,r6,r7,r9}

(TTCTCCCCC): {r1,r2,r4}, {r3,r5,r6,r7,r8,r9}
(GAGGAAGGG): {r2,r5,r6}, {r1,r3,r4,r7,r8,r9}
(TATTTAATA): {r2,r6,r7,r9}, {r1,r3,r4,r5,r8}
(GG-GGA-G-): {r3,r7,r9}, {r1,r2,r4,r5,r8}

- Each position partitions the reads in two groups

Splitting contigs: algorithm

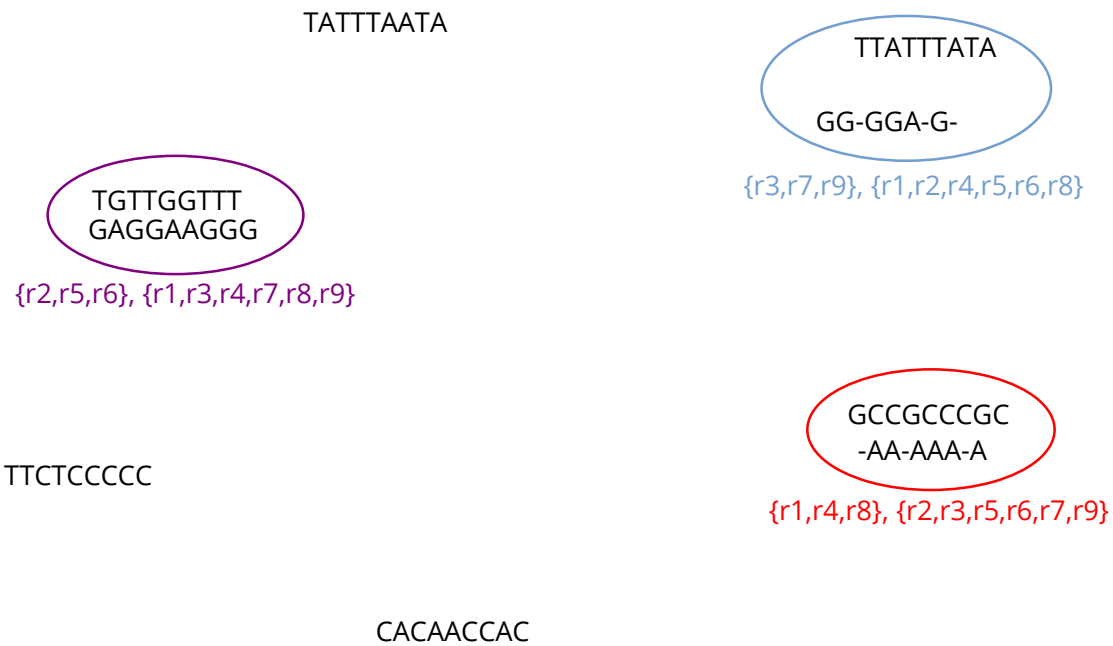
- We can use the *V-measure* (V) between two partitions to define a distance d between two positions:

$$d(TGTTGGTTT, GAGGAAGGG) = V\begin{pmatrix}\{r2,r5,r6\}\{r1,r3,r4,r7,r8,r9\}\\ \{r2,r5,r6\}\{r1,r3,r4,r7,r8,r9\}\end{pmatrix} = 0$$

$$d(TGTTGGTTT, CACAACCAC) = V\begin{pmatrix}\{r2,r5,r6\}\{r1,r3,r4,r7,r8,r9\}\\ \{r2,r4,r5,r8\}\{r1,r3,r6,r7,r9\}\end{pmatrix} = 0.16$$

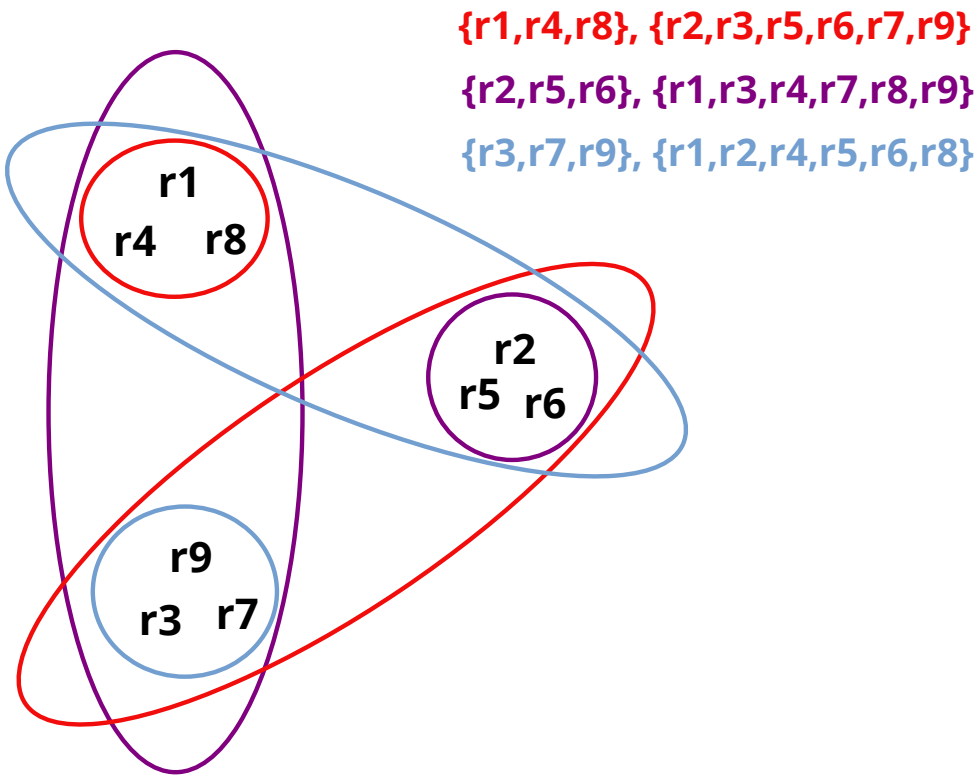
- Use this distance to cluster positions

Splitting contigs: algorithm



- Recurring partitions correspond to SNPs
- Isolated positions correspond to error-prone positions (the probability of two random partitions being close decreases exponentially with coverage)

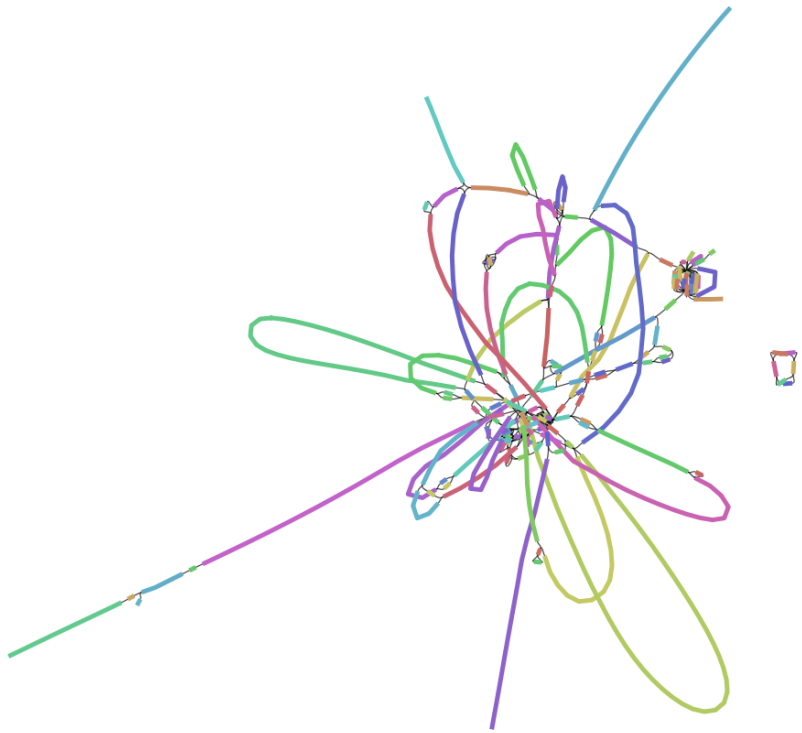
Splitting contigs: algorithm



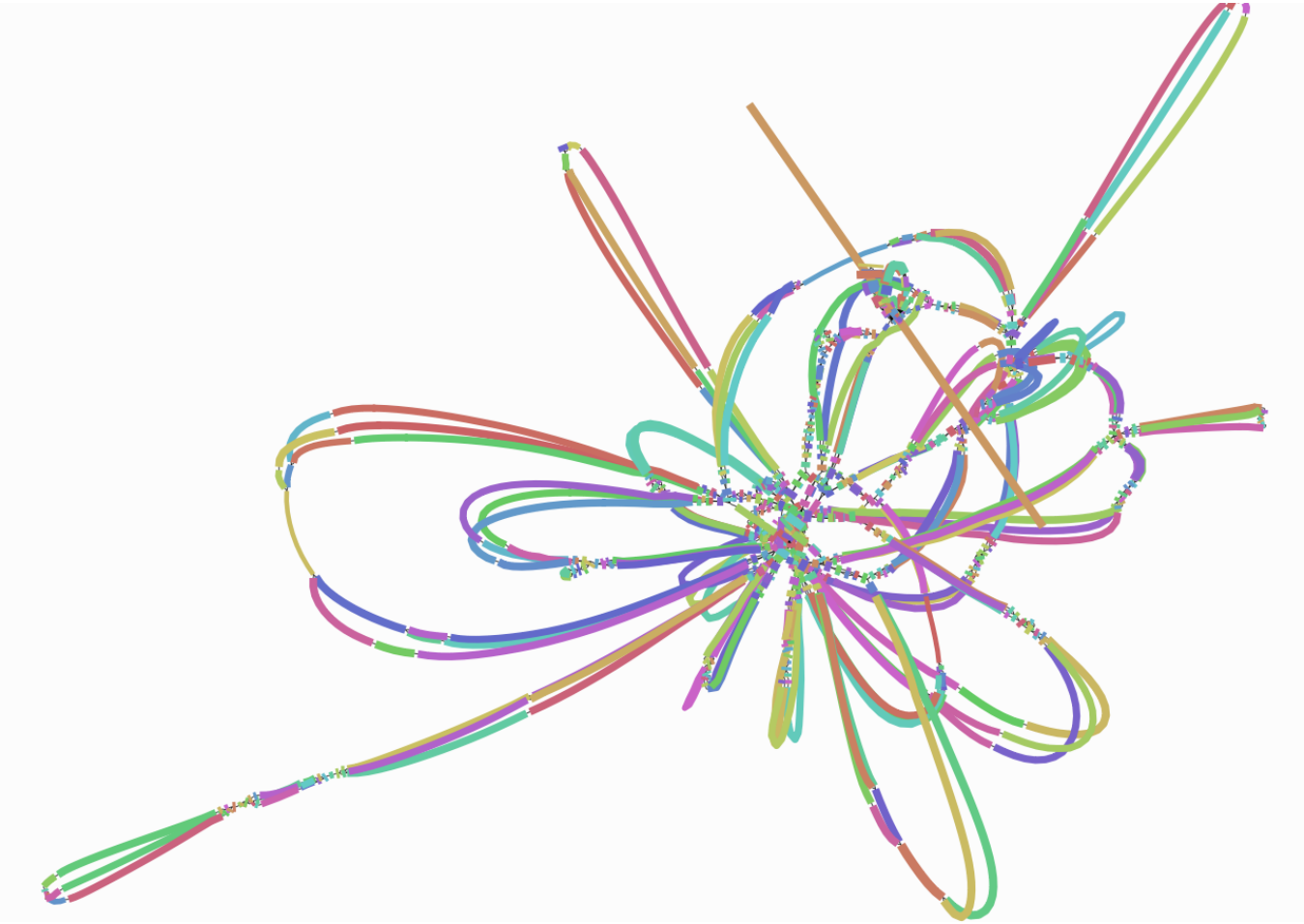
- Reads are separated, then the 3 versions of the contig are created using Racon

Dataset

- *In silico* mix of 3 strains of *Saccharomyces cerevisiae*
- Nanopore reads with $> 5\%$ error rate
- Assembled with Flye *--keep-haplotypes*
- We get a collapsed assembly



Result!



Result!

- Our "solution genome" = the 3 genomes assembled separately

	Size of assembly (Mbp)	Missing k-mers	Missing haplotype-specific k-mers
Solution	35.8	-	-
Before Hairsplitter	12.7	29.4 %	55.3 %
After Hairsplitter	33.9	8.5 %	16 %

Comparison with WhatsHap polyphase

- edge_132 of the assembly

	proportion of unassigned reads	proportion of mis-assigned reads
HairSplitter	0%	4.5%
Whatshap-polyphase	51%	8.2%

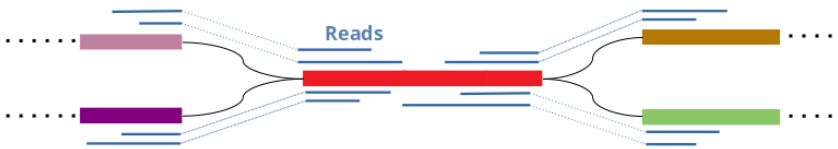
- HairSplitter much faster because does not need variant calling

Splitting the assembly

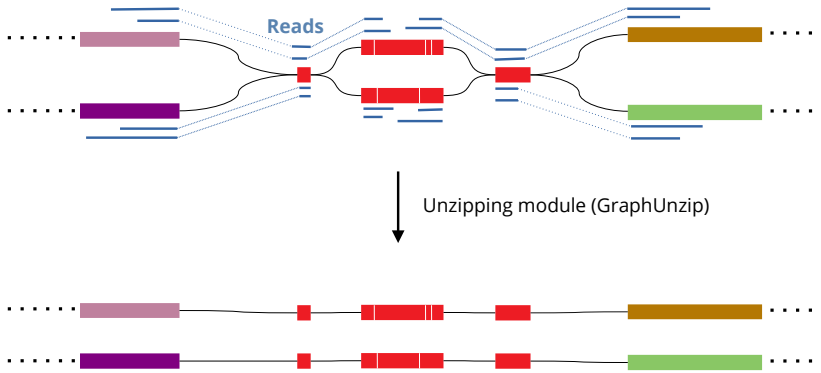
- First phase all contigs

Splitting the assembly

- First phase all contigs
- Then improve the contiguity of the assembly!



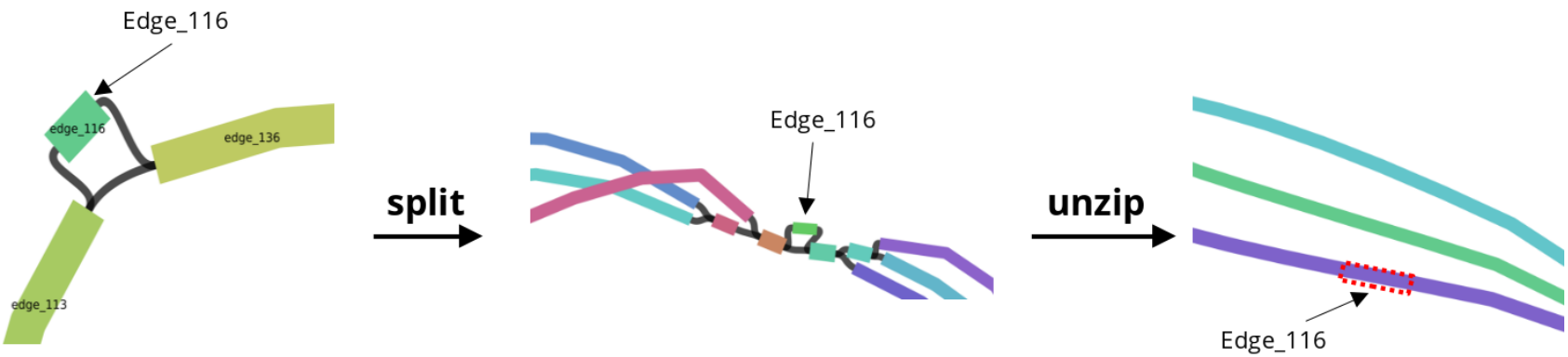
Before HairSplitter



After HairSplitter

Result!

- Example on the triploid yeast



- N50 went from 373 to 465kbp (assembled separately: 767kbp)

Pros and cons of HairSplitter

Limitations of HairSplitter:

- Not *very* fast: it re-polishes the whole assembly
- Limited in the number of haplotypes

Strengths of HairSplitter:

- Very modular, can be used with any assembler
- Naive: makes no assumption on ploidy, parameter-free
- Safe: won't artificially duplicate contigs

Take-home message

- HairSplitter splits collapsed assemblies from "draft" assemblies obtained by any means
- HairSplitter can recover haplotypes and distinguish repeated elements
- Only needs sequencing reads, potentially error-prone

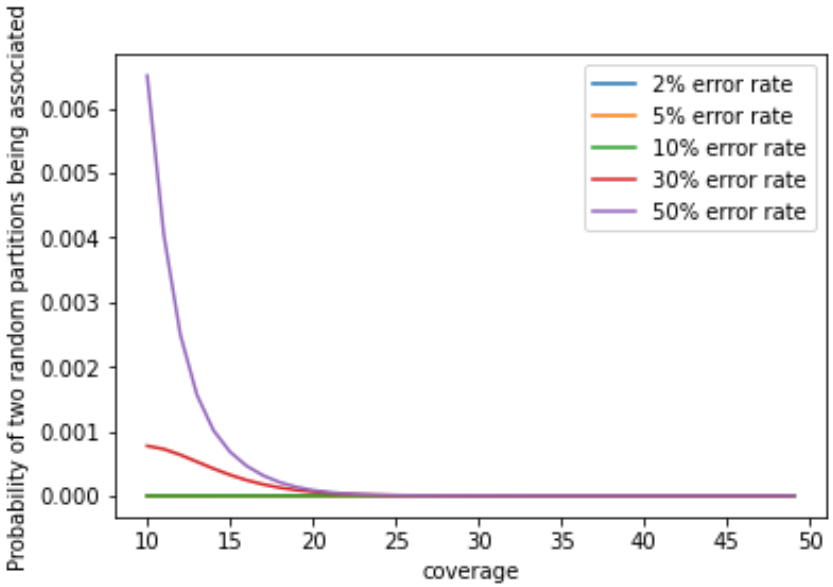
Take-home message

- HairSplitter splits collapsed assemblies from "draft" assemblies obtained by any means
- HairSplitter can recover haplotypes and distinguish repeated elements
- Only needs sequencing reads, potentially error-prone
- Not really available yet (github.com/RolandFaure/HairSplitter)

Acknowledgements

- Dominique Lavenier and Jean-François Flot for their supervision
- The EEB-EBE and GenScale teams

Similarity between random partitions



Probability of two positions being clustered together by HS, in function of the error rate at these position

- With high coverage, the erroneous positions won't cluster