Optical-Kermit: Optical map guided genome assembly

Miika Leinonen, Leena Salmela

University of Helsinki

5.2.2020

Genome assembly

Genome

CGGGTCGTTTTGTGTCCTCTGCACAAACGCCTAGGACCGGCGCCGTGCCC

⇓ **Use sequencing machine to produce smaller reads** ⇓

GGACCGGCGCCGTGCC
CTGCACAAACGCCTAGG
CTAGGACCGGCGCCGTGCC
CTCTGCACAAACGCCTA
GTTTTGTGTCCTCTG
TTTGTGTCCTCTGCACAA
AACGCCTAGGACCGGC
ACAAACGCCTAG
GTCCTCTGCACAAACGCCTA
GGTCGTTTTGTGTCC
TTTTGTGTCCTCTGCAC
CGTTTTGTGTCCTCT
GGTCGTTTTGTGTCCTC
GTCCTCTGCACAAACGC
CCTAGGACCGGCGCCG
GTCCTCTGCACAAACGCCTAGGA
AAACGCCTAGGACC

Genome assembly

Genome

? ?

⇑ **Reconstruct the unknown original genome using reads** ⇑

GGACCGGCGCCGTGCC
CTGCACAAACGCCTAGG
CTAGGACCGGCGCCGTGCC
CTCTGCACAAACGCCTA
GTTTTGTGTCCTCTG
TTTGTGTCCTCTGCACAA
AACGCCTAGGACCGGC
ACAAACGCCTAG
GTCCTCTGCACAAACGCCTA
GGTCGTTTTGTGTCC
TTTTGTGTCCTCTGCAC
CGTTTTGTGTCCTCT
GGTCGTTTTGTGTCCTC
GTCCTCTGCACAAACGC
CCTAGGACCGGCGCCG
GTCCTCTGCACAAACGCCTAGGA
AAACGCCTAGGACC

Guided assembly

- Hard problem
- Use additional information on top of the reads, like a reference genome
- We wanted to try to use optical maps

Guided genome assembly with Kermit

- Our contribution: Optical-Kermit, modified version of the original Kermit[1] program
- Original Kermit is an overlap graph based genome assembly program
- Kermit uses genetic maps as auxiliary information to guide the assembly
- We will talk about our work (Optical-Kermit) later, but first...

[1] Kermit: linkage map guided long read assembly; Riku Walve, Pasi Rastas, Leena Salmela; Algorithms for Molecular Biology volume 14, Article number: 8 (2019)

Kermit

- Uses an overlap graph to reconstruct the genome
- Nodes in the graph represent known sequences (reads, parts of reads,...)
- Edges represent overlaps between node sequences
- Paths in the graph give us longer consensus sequences that are likely to appear in the genome (*unitigs/contigs*)

Overlap graph

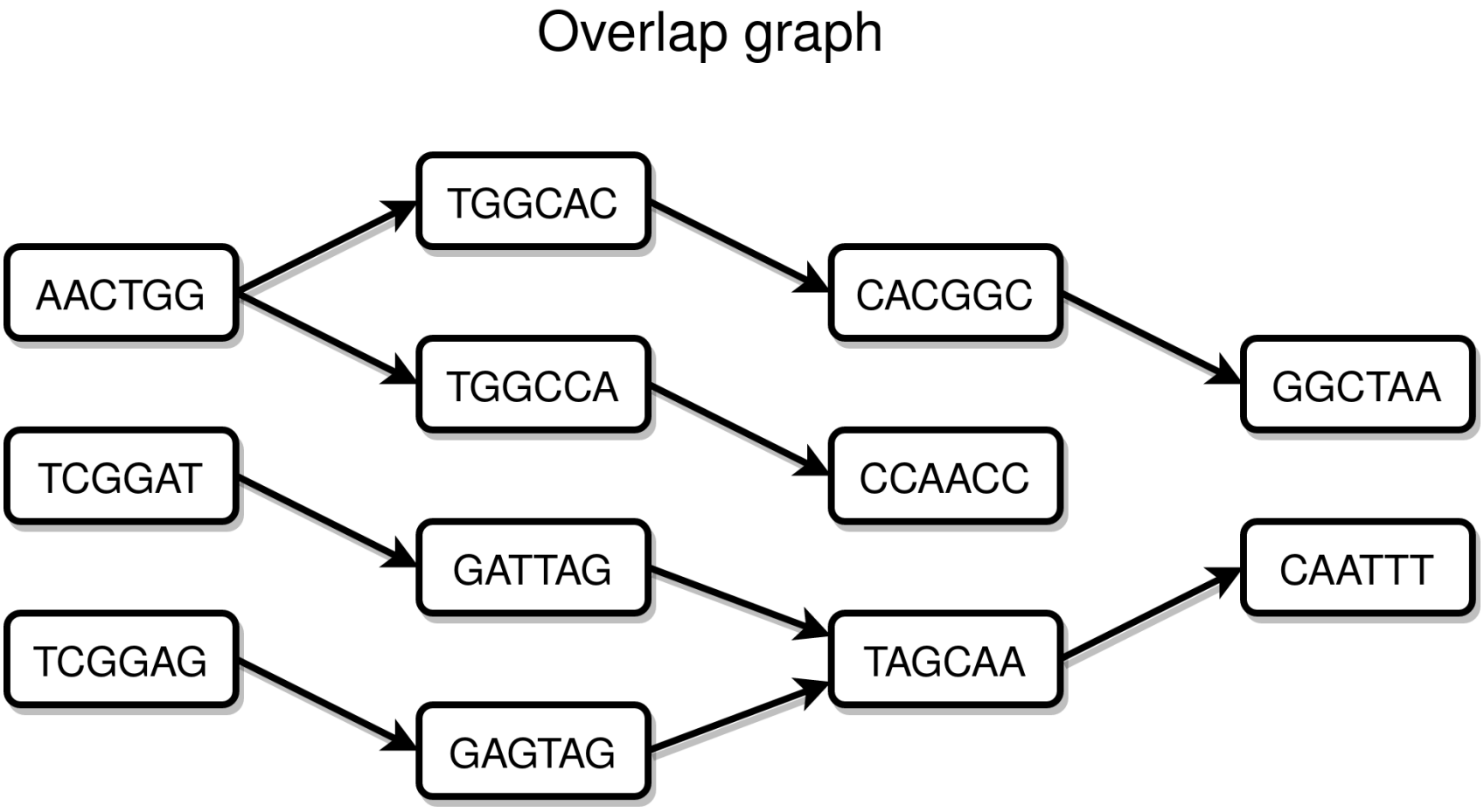

Figure 1: Example overlap graph. Vertices are known sequences. There is an edge between sequences if they overlap. Built using reads.

Overlap graph

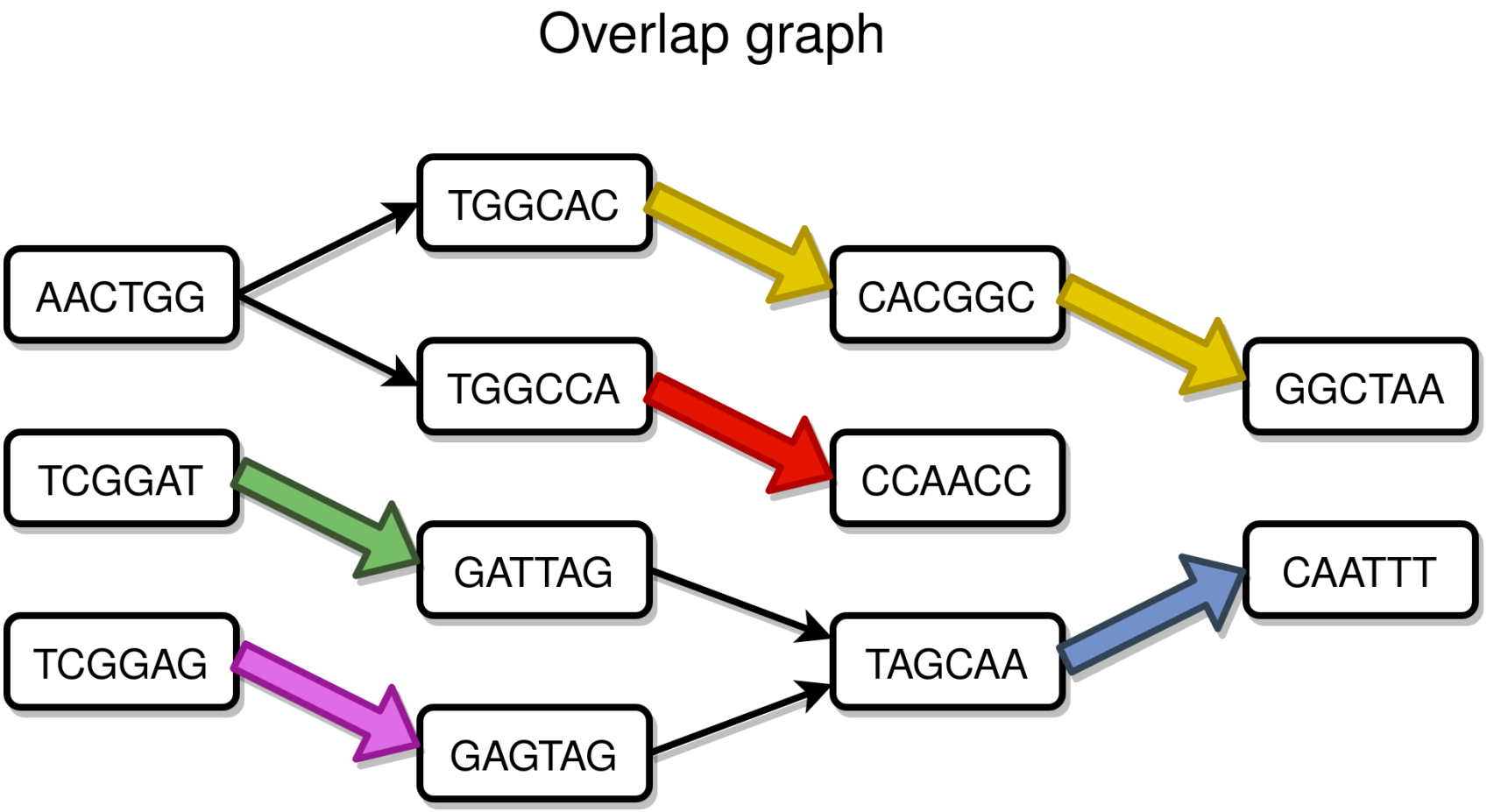

Figure 2: We can get unitigs by finding all non-branching paths in the graph. Unambiguous sequences.

5 unitigs can be found from this graph: TGGCACGGCTAA, TGGCCAACC, TCGGATTAG, TCGGAGTAG, TAGCAATTT

Kermit

- Kermit uses genetic maps to approximate the read positions in the genome i.e. to determine their relative order
- Reduces the number of overlaps we need to consider
- In practice, different regions of the genome are marked with distinct colors
- Reads are colored according to the color of their approximated locations
- Inconsistent edges of the graph can now be removed based the relative positions of the reads i.e. colors of the nodes

Colored overlap graph

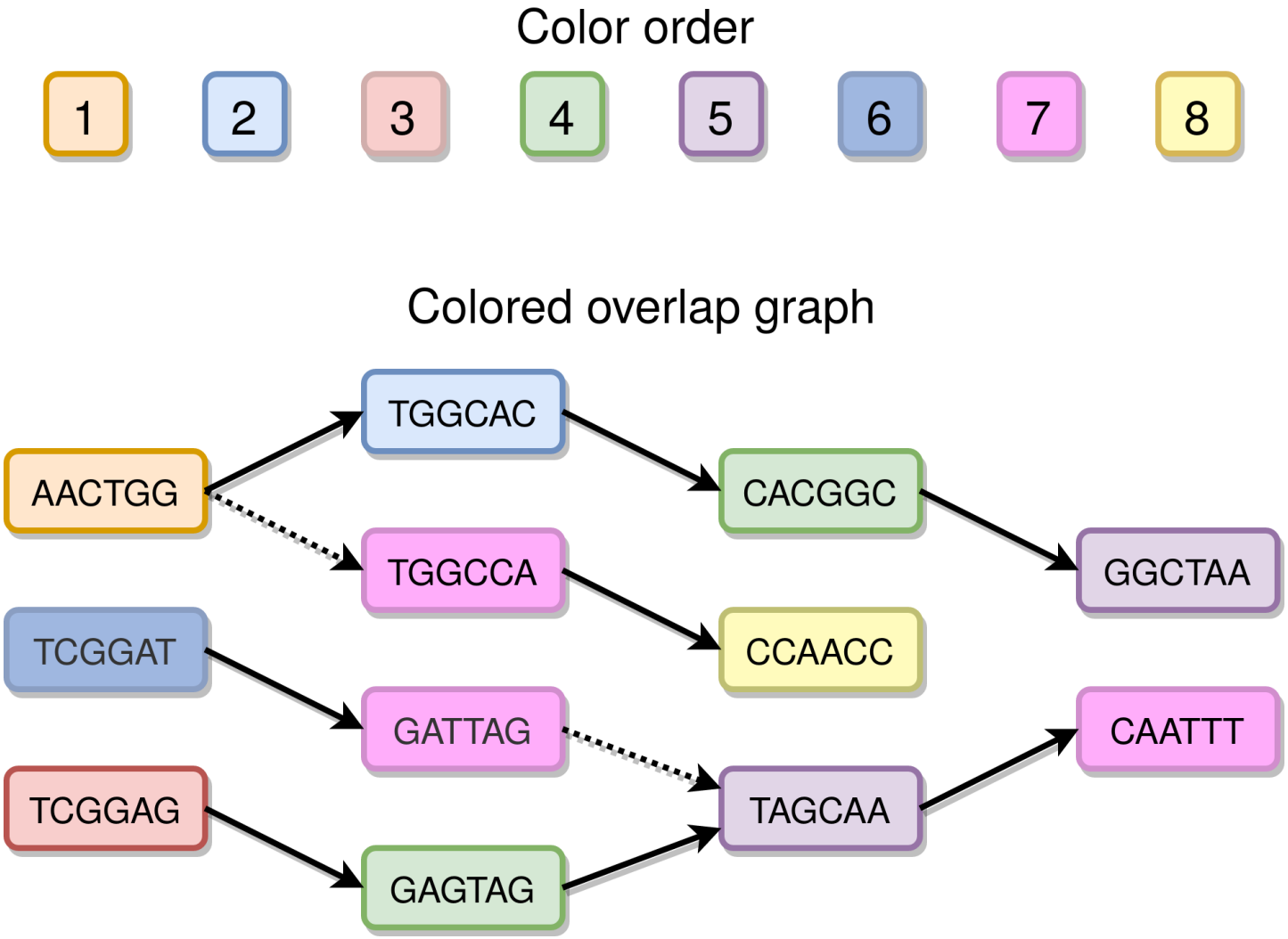

Figure 3: Graph trimming. Orange to pink edge is removed because there are missing colors between them. Pink to gray edge is removed because their order is wrong.

Colored overlap graph

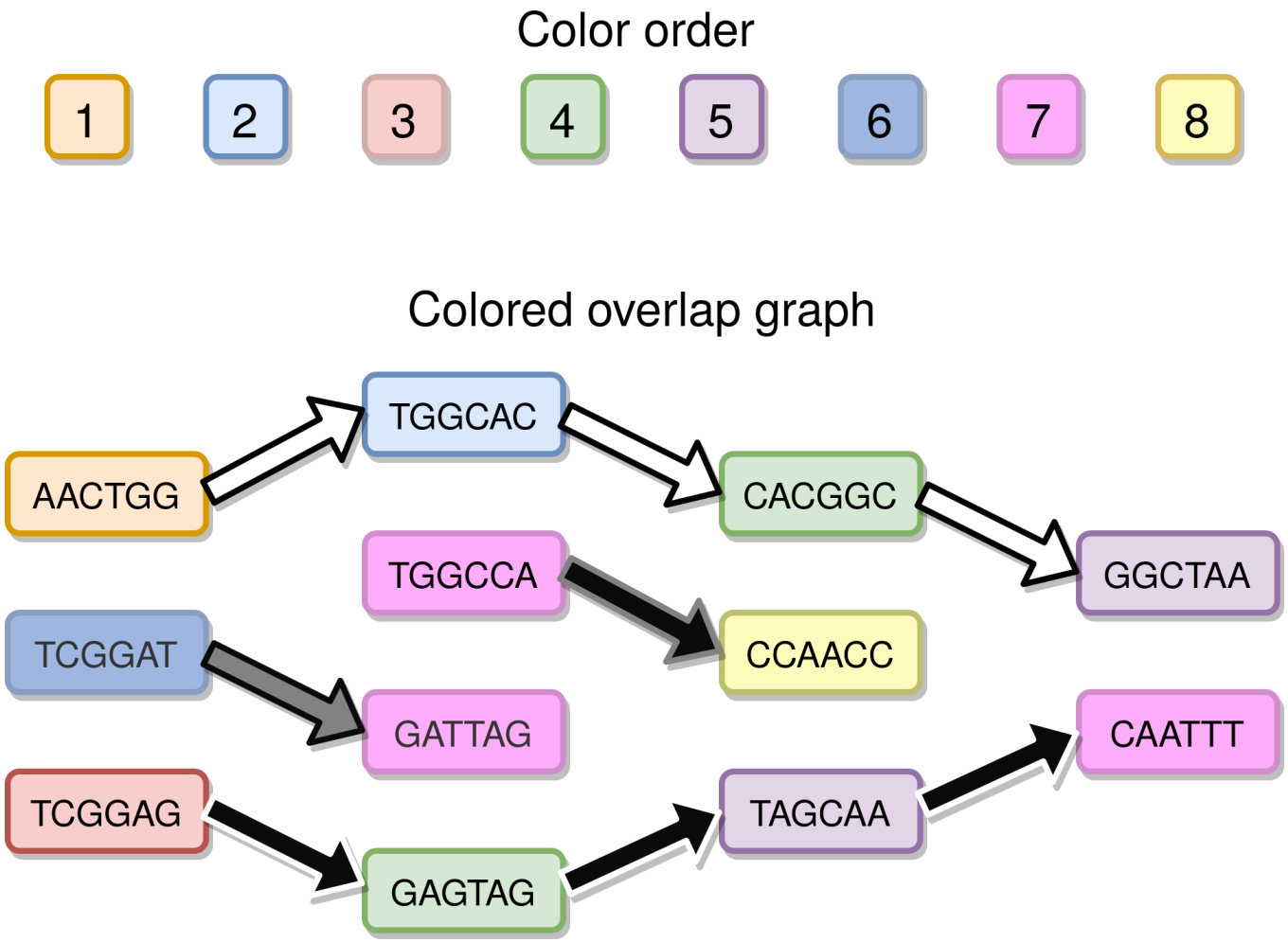

Figure 4: Get contigs by finding all non-branching paths.

4 unitigs can be found from this graph: AACTGGCACGGCTAA, TGGCCAACC, TCGGATTAG, TCGGAGTAGCAATTT

Optical-Kermit

- Now let's move on to our modification to the original Kermit program
- We use optical maps to approximate the read locations in the genome

Optical maps

- DNA sequence can be split at specific restriction sites with *restriction enzymes*
- For example, enzyme XhoI splits DNA sequence whenever it finds a restriction site 'CTCGAG'
- The lengths of the resulting fragments are called an optical map

Optical maps

Example sequence

TACTAGTCTCGAGCCGTAGGCATCTCGAGAAACGCGTCCGCTCGAGGGAGTGCA

⇓ **Apply restriction enzyme (XhoI recognizes restriction sites 'CTCGAG')** ⇓

TACTAGTCTCGAGCCGTAGGCATCTCGAGAAACGCGTCCGCTCGAGGGAGTGCA

⇓ **XhoI cuts sequence after the first Cs** ⇓

TACTAGTC||TCGAGCCGTAGGCATC||TCGAGAAACGCGTCCGC||TCGAGGGAGTGCA

⇓ **Measure fragment lengths** ⇓

8||16||17||13

⇓ **Optical map** ⇓

$[8, 16, 17, 13]$

Optical maps

- Optical map of the genome can be obtained experimentally with the help of restriction enzymes
- Read optical maps are obtained computationally

Initial idea

- Compare genome and read optical maps
- Find the approximate read locations by aligning optical map fragment lengths

Problems with the approach

- Even with long reads and multiple restriction enzymes, we got relatively short optical maps
- If the number of fragments is too low, no reliable alignment can be made

Second idea

- We still wanted to use optical maps, but needed longer sequences
- Do an initial assembly without auxiliary information, *pre-coloring assembly*
- Build optical maps for the resulting pre-coloring contigs
- Contigs are much longer than individual reads, so their optical maps also have more fragments
- Reliable contig-to-reference optical map alignments

Second idea

- We can now approximate the contig positions in the genome
- Fragments of the reference genome optical map are colored with distinct colors
- Contig optical maps are colored based on their alignment with the reference optical map

Optical map coloring

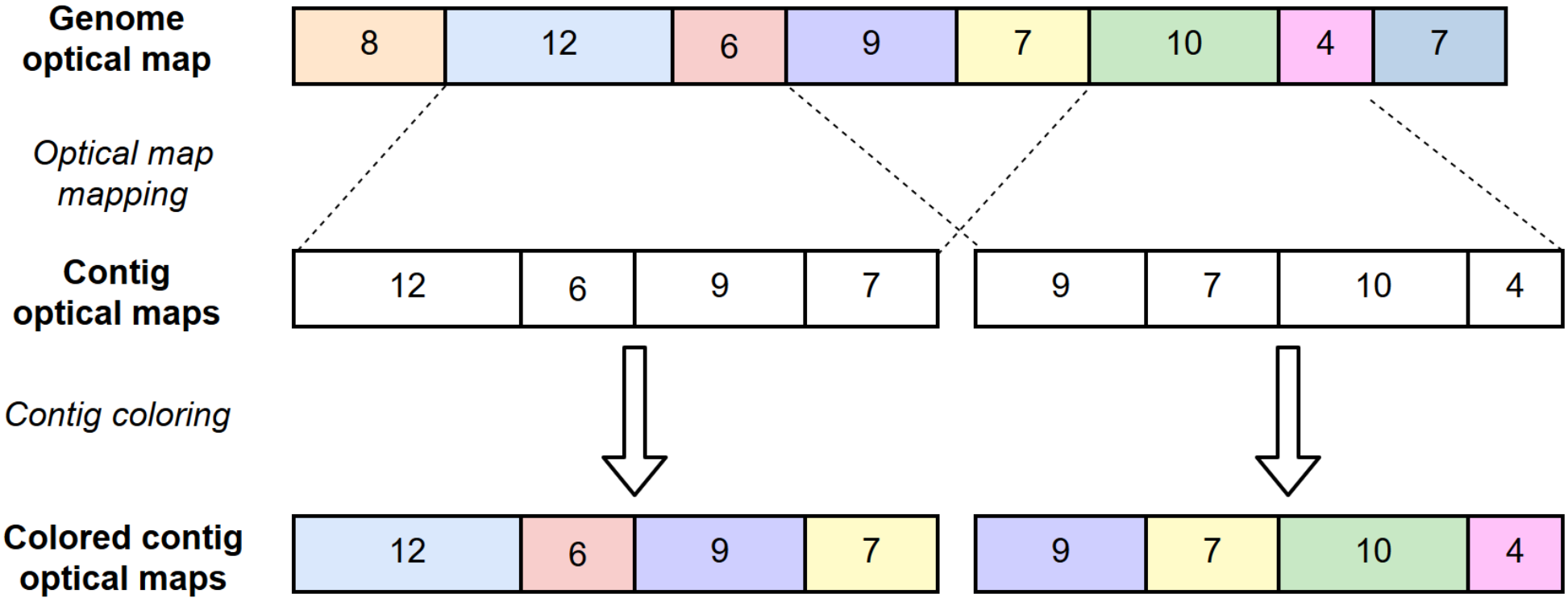

Figure 5: Optical map coloring. Fragments of the genome optical map are colored. Contigs optical maps are *mapped* to the genome optical map. Color contig optical map fragments.

Second idea

- Ultimate goal is to approximate read positions
- Use a sequence aligner to align reads to contigs, and color the reads accordingly

Read coloring

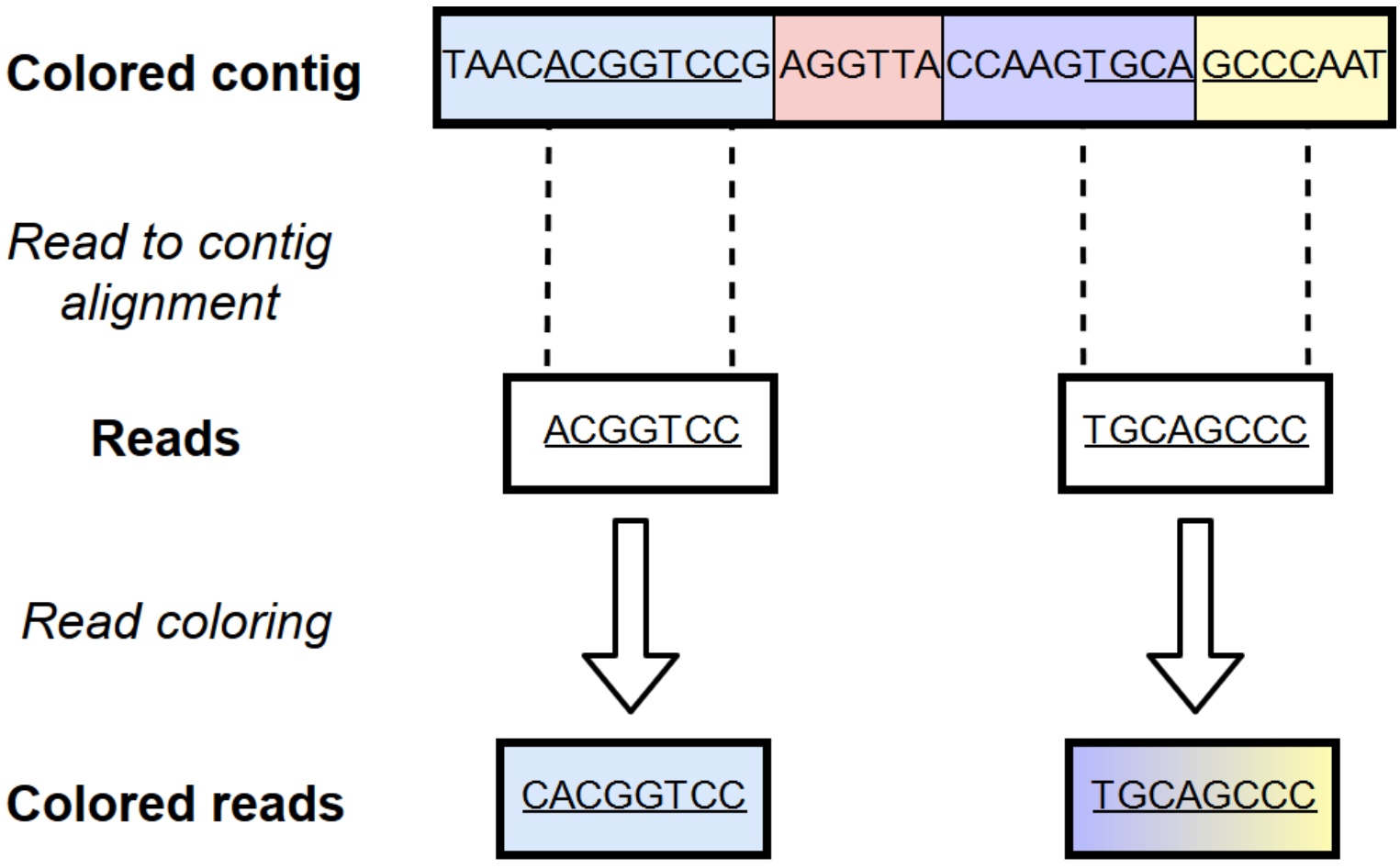

Figure 6: Read coloring. Reads are aligned with the contigs. Reads are colored with the colors of the fragments they cover.

Ready to run

- Reads are now colored i.e. we have some idea about their relative order
- (In reality alignments are not this simple, need to find appropriate score thresholds, some reads will be left uncolored)
- We have everything we need for the assembly
- Use the exact same approach as with the original Kermit: build a colored overlap graph, trim it, find consensus sequences
- This time we get new contigs, *post-coloring contigs*

Optical-Kermit pipeline summary

1. Use reads to produce pre-coloring contigs
2. Create optical maps of the pre-coloring contigs and the reference genome
3. Align pre-coloring contig optical maps to the reference optical map
4. Align reads to pre-coloring contigs
5. Use reads-to-contigs alignment information with contigs-to-reference optical map alignment information to color the reads
6. Give colored reads to Kermit to produce the final product, post-coloring contigs

Experiment results

C.elegans assembly	miniasm	Optical-Kermit
Number of contigs	111	94
Number of $>$ 50Kbp contigs	75	61
Contigs total length (Kbp)	100 430	100 215
$>$ 50Kbp contigs total length (Kbp)	99 770	99 637
Misassemblies	11	8
NGA50 (Kbp)	2 656	3 028

Table 1: *C.elegans* (a roundworm) assembly results. 6 (+1) chromosomes, 100 273 Kbp length. Simulated PacBio reads.

Experiment results

A.thaliana assembly	miniasm	Optical-Kermit
Number of contigs	539	276
Number of $>$ 50Kbp contigs	114	105
Contigs total length (Kbp)	133 818	125 297
$>$ 50Kbp contigs total length (Kbp)	119 877	199 817
Misassemblies	58	63
NGA50 (Kbp)	1 442	1 648

Table 2: *A.thaliana* (a flowering plant) assembly results. 5 (+2) chromosomes, 118 058 Kbp length. Real PacBio reads.

Conclusion

- Optical-Kermit introduces a flexible way to utilize optical maps automatically to guide genome assembly
- The results seem positive, so Optical-Kermit can be an effective way to guide genome assembly with optical maps