Assembly of long, error-prone reads using repeat graphs

Mikhail Kolmogorov, Jeffrey Yuan, Yu Lin, and Pavel A. Pevzner

June 29, 2021

Johannes Hausmann, Luis Kress

Background

- Assembly: reconstruct target sequence from the reads

Background

- Assembly: reconstruct target sequence from the reads
- Different assemblers, different graph structures (De-Bruijn, Overlap-layout, String)

Background

- Assembly: reconstruct target sequence from the reads
- Different assemblers, different graph structures (De-Bruijn, Overlap-layout, String)
- Repeats → assembly fragmentation

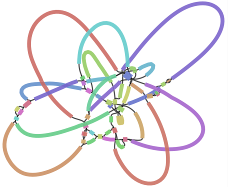


Figure 1: Tangled assembly graph

Background

- Assembly: reconstruct target sequence from the reads
- Different assemblers, different graph structures (De-Bruijn, Overlap-layout, String)
- Repeats → assembly fragmentation

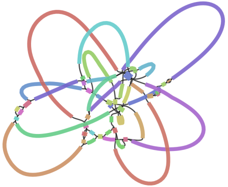


Figure 1: Tangled assembly graph

- Error rate long read ↔ short read

Background

- Assembly: reconstruct target sequence from the reads
- Different assemblers, different graph structures (De-Bruijn, Overlap-layout, String)
- Repeats → assembly fragmentation

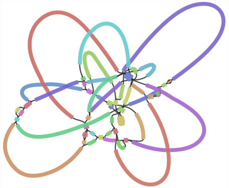


Figure 1: Tangled assembly graph

- Error rate long read ↔ short read
- Flye → resolve these repeats correctly

Disjointigs

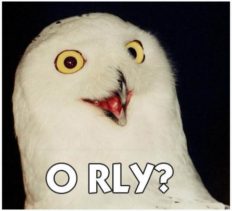
- Most assemblers spend much time on correct contig assembly

Disjointigs

- Most assemblers spend much time on correct contig assembly
- Flye uses a different approach [1]:

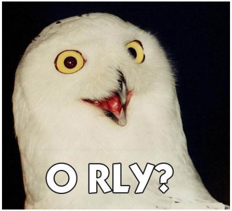
Disjointigs

- Most assemblers spend much time on correct contig assembly
- Flye uses a different approach [1]:
  - we don't care



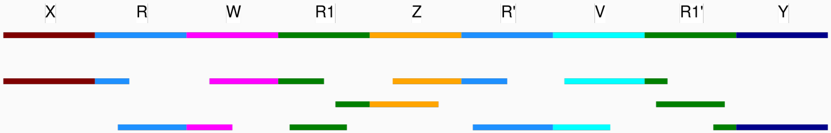
Disjointigs

- Most assemblers spend much time on correct contig assembly
- Flye uses a different approach [1]:
    - we don't care (at least at the initial stage)

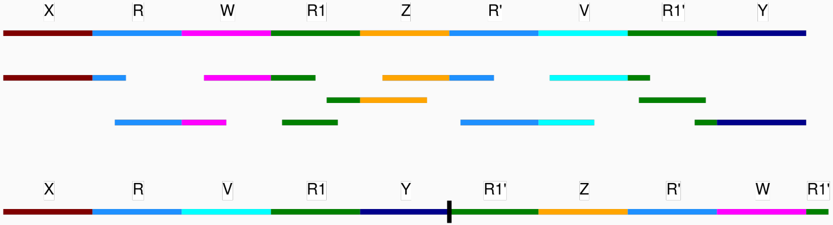


- Generate paths from overlapping reads without checking for correct repeat resolution → Disjointigs

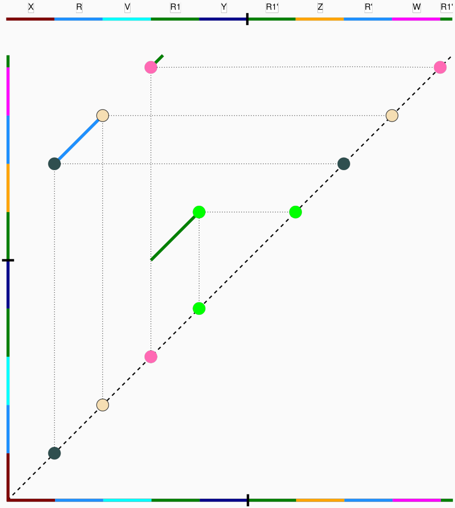
Repeat Graph Creation



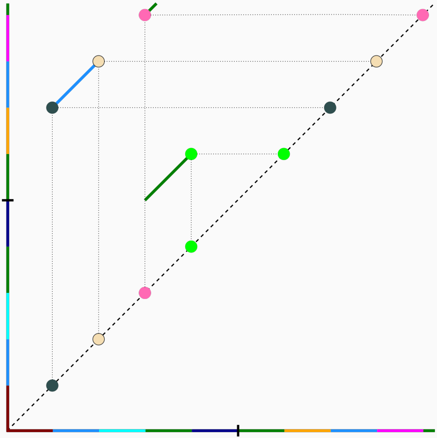
Repeat Graph Creation

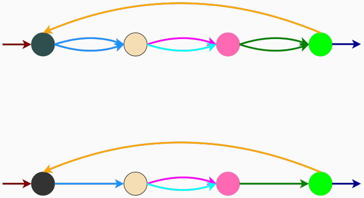


Repeat Graph Creation



Repeat Graph Creation





Results

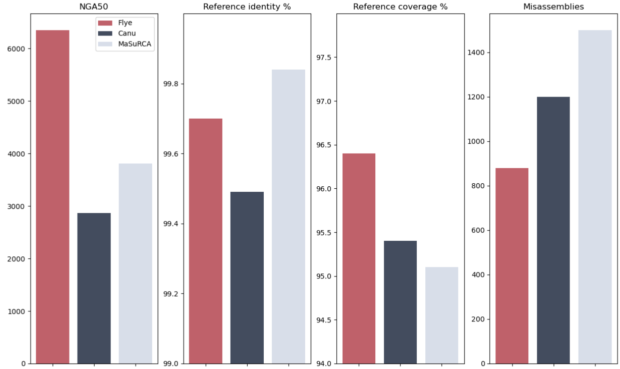


Figure 2: Results for HUMAN testset

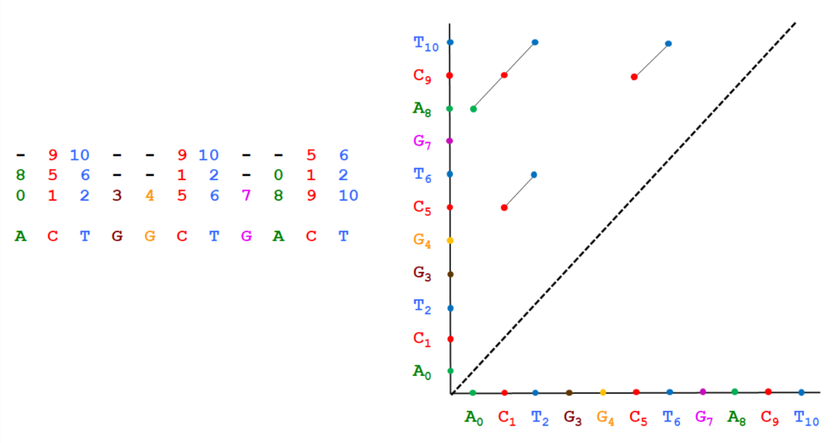
References i

M. Kolmogorov, J. Yuan, Y. Lin, and P. A. Pevzner.
Assembly of long, error-prone reads using repeat graphs.
Nature Biotechnology, 37(5):540–546, May 2019.

Git (presentation and poster)

Appendix

Dot plot creation



-	9	10	-	-	9	10	-	-	5	6
8	5	6	-	-	1	2	-	0	1	2
0	1	2	3	4	5	6	7	8	9	10
A	C	T	G	G	C	T	G	A	C	T

T_{10}
C_9
A_8
G_7
T_6
C_5
G_4
G_3
T_2
C_1
A_0
A_0 C_1 T_2 G_3 G_4 C_5 T_6 G_7 A_8 C_9 T_{10}

Figure 3: Dot plot creation

Repeat graphs

- generalization of de bruijn graphs
- structure
- creation
  - from disjointigs = random walk of reads on the repeat graph
  - means the repeat graph hasn't to be known

Difference repeat graph de Bruijn graph

- A-Bruijn graph (alignments) generalizes the de Bruijn graph
- We thus argue that the time has come to explain that the breakpoint graphs and the de Bruijn graphs are two identical data structures (if one ignores a cosmetic difference between them) as they both represent specific instances of a general notion of the A-Bruijn graph introduced in [13]. The A-Bruijn graphs are based on representing genomes as sets of labeled paths and further gluing identically labeled edges (breakpoint graphs) or vertices (de Bruijn graphs) in the resulting paths.
- de Bruijn graphs need correct bases
- otherwise tangled graph

Segmental duplications

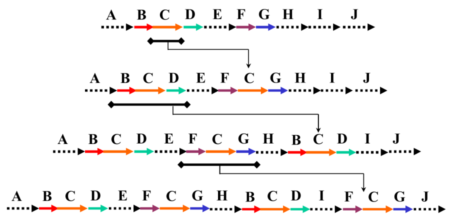

A B C D E F G H I J
A B C D E F C G H I J
A B C D E F C G H B C D I J
A B C D E F C G H B C D I F C G J

Figure 4: Segmental Duplications

- Segmental duplications are duplicated blocks of genomic DNA typically ranging in size from 1-200 kb (IHGSC 2001)
- They often contain sequence features such as high-copy repeats and gene sequences with intron-exon structure.

Contigity improvement

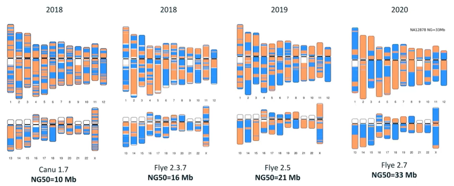


Figure 5: Contigity improvements

- colors are contigs

Contigity improvement

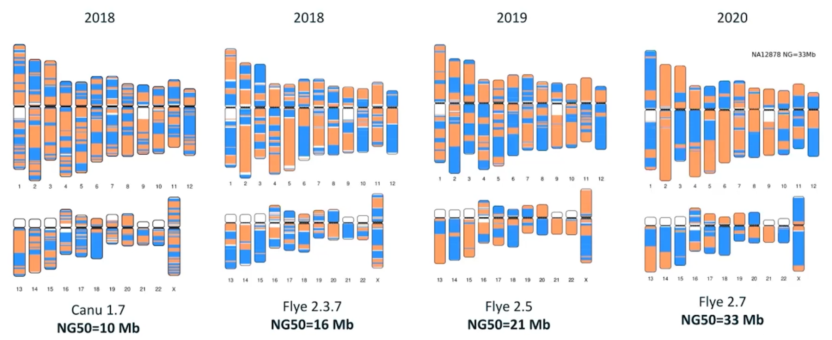


Figure 6: Contigity improvements

- colors are contigs → change in color means fragmentation