Long Read Genome Assembly

A Case Study in Python

Group D: SMRT-y pants
Dylan Estep, Ethan Gaskin, Akshat Gupta, Zhen Yang

Background

- Short-read sequencing is known for its cost-effectiveness, accuracy
- However, complexities of genomes, such as repetitive regions, pose a challenge when assembling the genomes from short amplified fragments
- Long reads enhance de novo assembly, overcoming amplification bias even though with higher error rates

Problem Statement

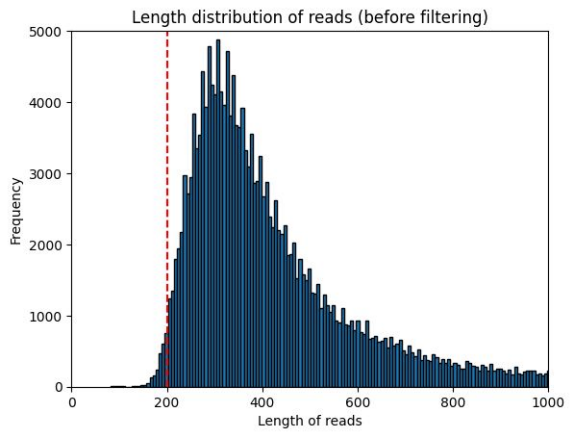
De Novo Genome Reconstruction Problem: Reconstruct a genome from error-prone, single-molecule DNA sequencing reads (long reads) without a reference genome

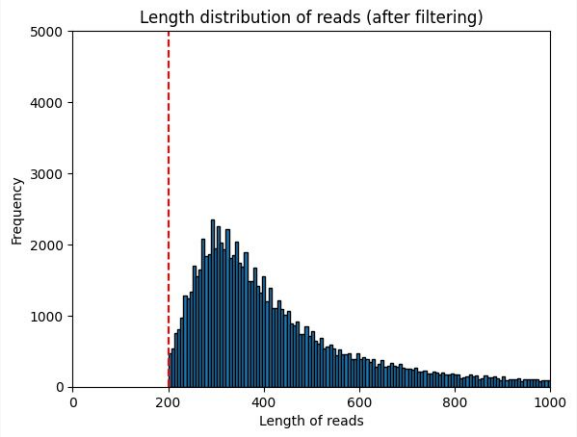
- Input: fastq file consisting of
  - N single-molecule sequencing reads
  - quality scores of base calls for each read
- Output: Long contiguous sequences (contigs) that are highly likely to be in the genome used to produce these reads

Methods

Methods Overview

- Quality Control (NanoFilt -q 10 -l 200 --minGC 0.3)
- Overlap-Layout-Consensus (OLC)
  - Overlap: Detect overlapping reads
  - Error Correction: FalconSense
  - Layout: Group overlapped reads into layouts of reads coming from same genomic region
  - Consensus: Overlay reads in each layout and compute consensus seq (contigs)





Whole-genome, long-read

Oxford Nanopore

106,084 reads

Salmonella enterica serotype Hadar

Ref. 1

Find Overlap Using MHAP

S_1: CATGGACCGA

CAT GAC
ATG ACC
TGG CCG
GGA CGA

GCAGTACCGA :**S_2**

GTA CGA
AGT CCG
CAG ACC
GCA TAC

S_1: CATGGACCGA

| | | | |

S_2 GCAGTACCGA

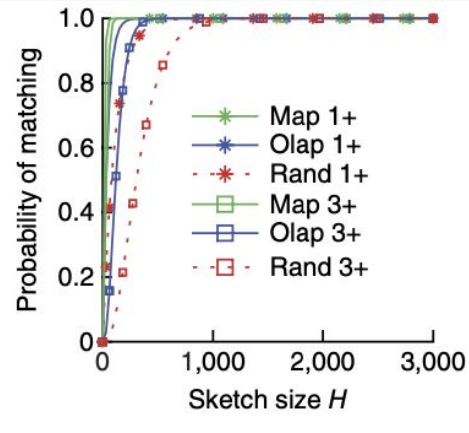
$J(S_1, S_2) \approx 2/4 = 0.5$

min-mers

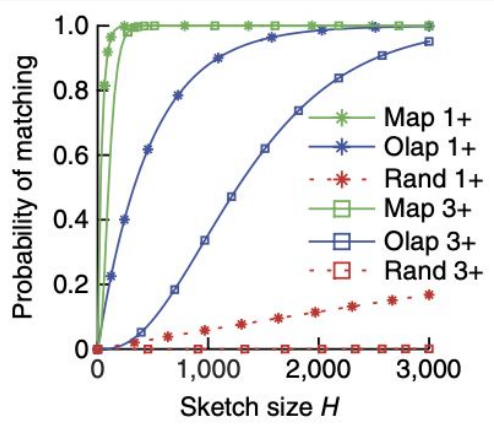
[5, 1, 2, 15]
Sketch (S_1)

[5, 1, 6, 6]
Sketch (S_2)

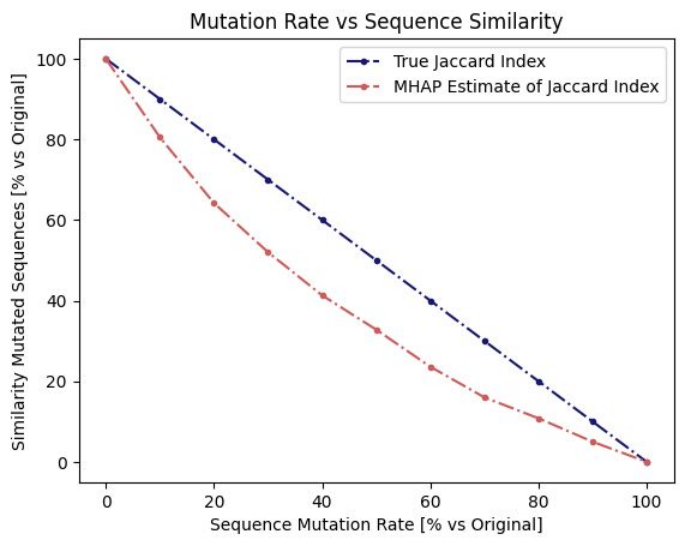
Choose Parameter k, *H*



k = 10



k = 16



H = 1256

Ref. 2

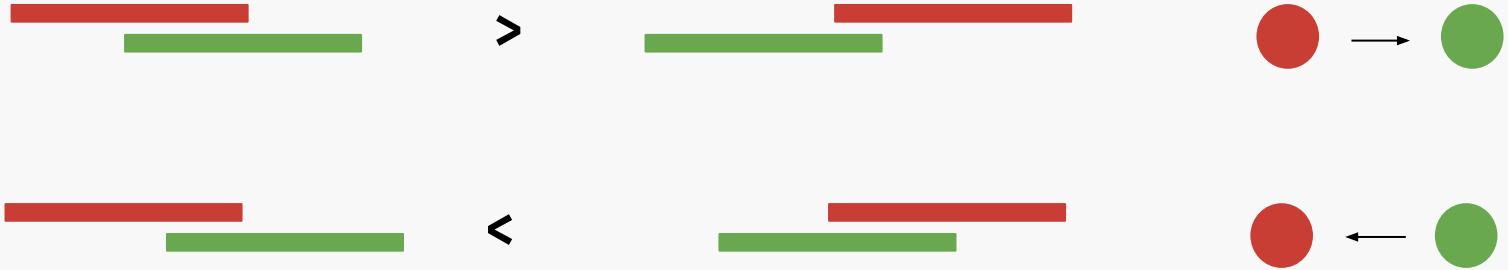
Parallelization

- Assume MHAP with sketch size H tells us (on average) each read has Z candidate reads to overlap with (N = 100,000, H = 1,256, Z = 1,000):
  - O(N*H) = 125,600,000 hashes to computed (for all N sketches)
  - $O(N^2H^2)$ = 1,577,536,000,000,000 calcs for Jaccard mtx (N^2 elements)
  - O(N*Z) = 100,000,000 alignments will be necessary (2% of total)
    - Better than $100{,}000^2/2$ = 5,000,000,000 alignments!
- Multiple embarrassingly parallel steps for MHAP
  - Pairwise Jaccard Index computations are independent & symmetric
  - Computing sketches for each read is independent
- Future parallelization: Pairwise alignments from MHAP independent
  - Efficient memoization should allow us to avoid recomputation

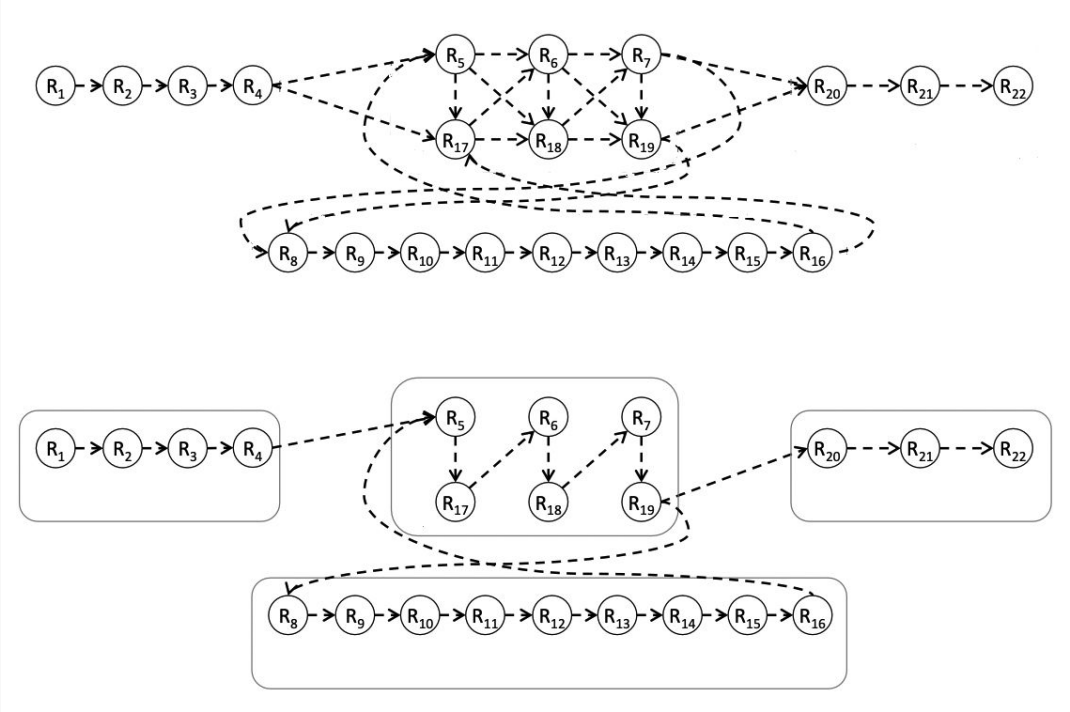
Overlap: Generating directed overlap graph

Error Correction by FalconSense: for each read, generate fitting alignment from all overlapping reads, then obtain consensus for each position in the read

Additionally perform **overlap alignments** in *both* directions to ascertain direction of edge that should be drawn in overlap graph, based on higher score



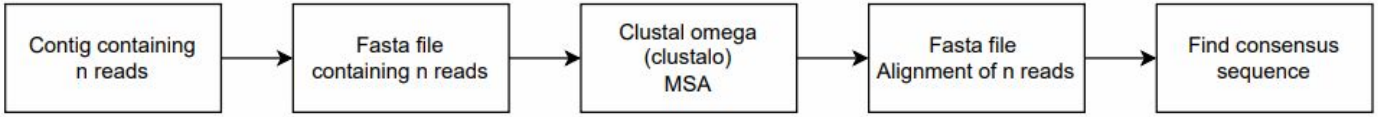
Layout



maximal nonbranching paths = contigs

Ref. 4

Consensus

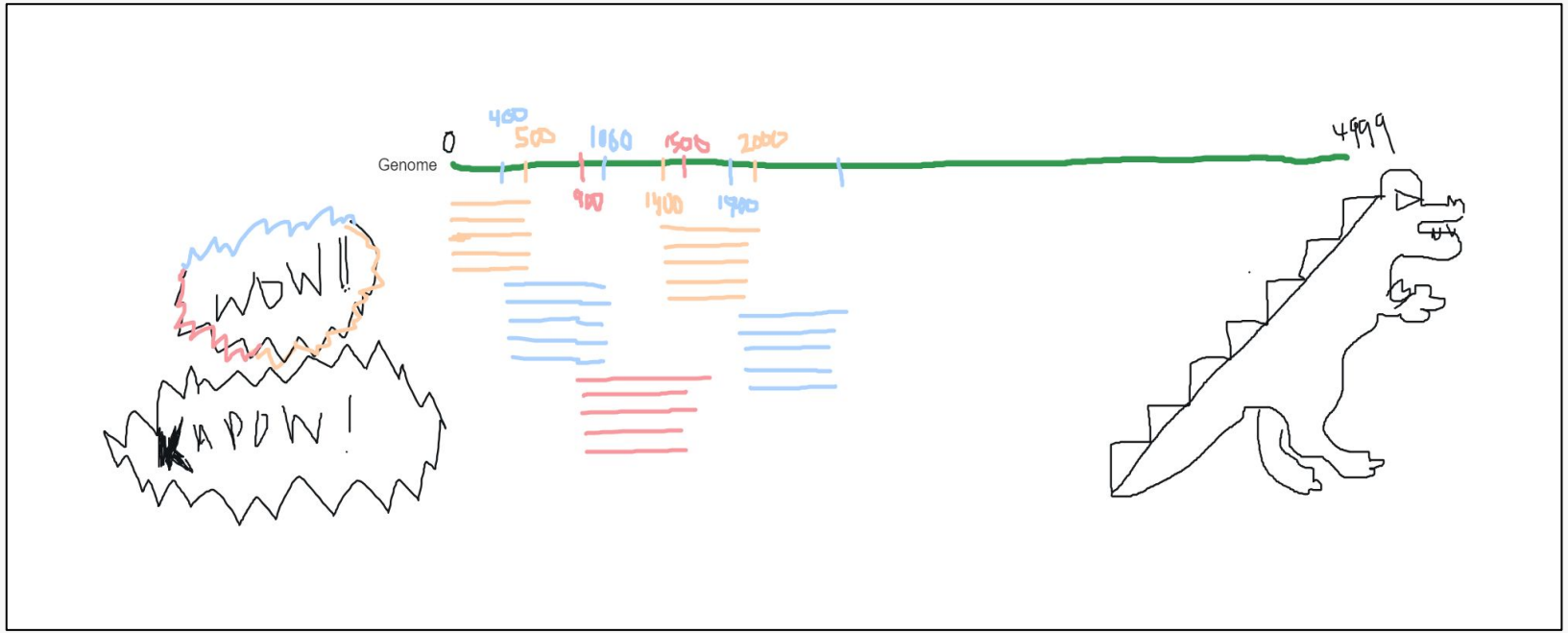


Finding the consensus from the MSA

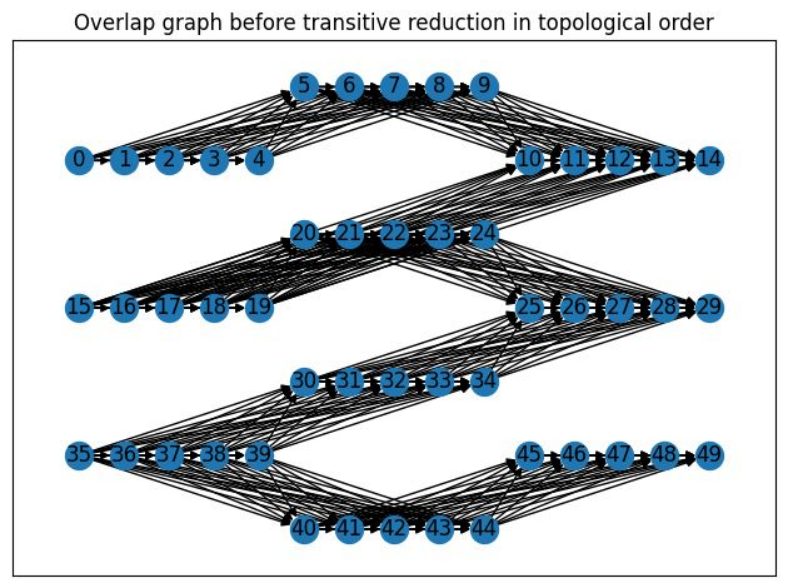
```
TTAACCTTGGTTTTGAACTTGAACACTTAGGGGATTG
        TGGTTTTGAACTTGAACACTTAGGGGATTGAAGATTCAACAA
                AACTTGAACACTTAGGGGATTGAAGATTCAACAACCCTAAAGCT
                        CACTTAGGGGATTGAAGATTCAACAACCCTAAAGCTTGGG
                                ATTGAAGATTCAACAACCCTAAAGCTTGGGGT
                                        ACCCTAAAGCTTGGGGTA
                                                AGCTTGGGGTAAAAC
TTAACCTTGGTTTTGAACTTGAACACTTAGGGGATTGAAGATTCAACAACCCTAAAGCTTGGGGTAAAAC
```

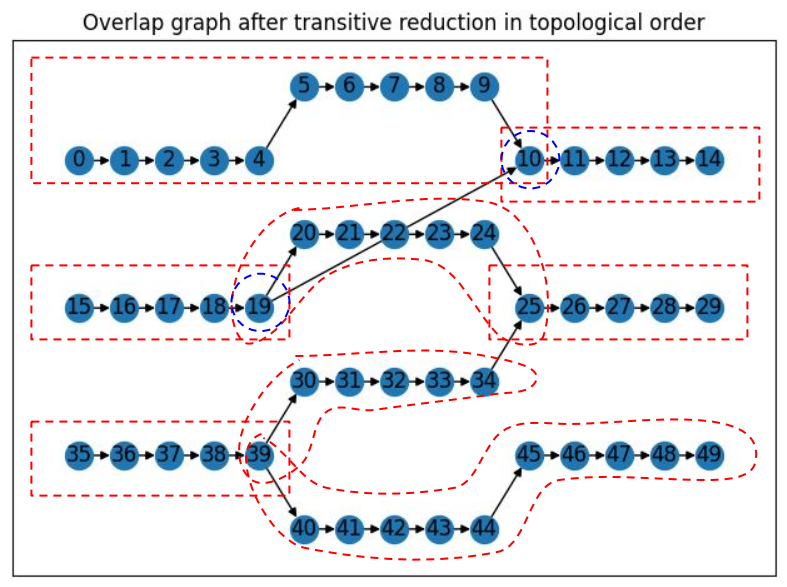
Results

The Perfect OLC Dataset...



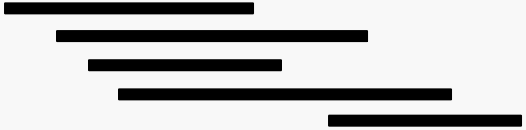
Why Transitive Reduction Matters?





Imperfect OLC Datasets...

- Case 1
  - A synthetic genome is broken down into different sizes of reads (depth x1)
  - No sequencing errors introduced into the reads



- Case 2
  - A synthetic genome is broken down into different sizes of reads (depth x1)
  - Sequencing errors (< 10%) are introduced into the reads

- **Our Results**: (1) More contigs than reads, (2) Most contigs consists of two reads, (3) Many disjoint components (1 large connected component)

Conclusions

- Optimization is an essential step to handle complex biological data.
- Generating the overlap graph is the computational bottleneck of genome assembly.
- The performance of our algorithm is very poor compared to Spades, even on the toy data!
- Python might not be the best language to handle computationally intensive tasks.

Future Steps

- Continue to run the pipeline on the real data (expensive)
- Parallelize Jaccard Index Calculation and alignment step
- Benchmark assembly results to existing tools
- Troubleshoot the layout step for different sets of synthetic data
- Characterize effects of different sequencing error rates and depths on our algorithm's performance for the "Perfect OLC Dataset"

References

1. Amarasinghe, S. L., Su, S., Dong, X., Zappia, L., Ritchie, M. E., & Gouil, Q. (2020). Opportunities and challenges in long-read sequencing data analysis. Genome Biology, 21(1). https://doi.org/10.1186/s13059-020-1935-5
2. Berlin, K., Koren, S., Chin, C., Drake, J., Landolin, J. M., & Phillippy, A. M. (2015). Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. Nature Biotechnology, 33(6), 623–630. https://doi.org/10.1038/nbt.3238
3. Chu, J., Mohamadi, H., Warren, R. L., Yang, C., & Birol, İ. (2016). Innovations and challenges in detecting long read overlaps: an evaluation of the state-of-the-art. Bioinformatics, 33(8), 1261–1270. https://doi.org/10.1093/bioinformatics/btw811
4. Sung, W. (2017). Algorithms for Next-Generation Sequencing. In Chapman and Hall/CRC eBooks. https://doi.org/10.1201/9781315374352
5. Zhang, H., Jain, C., & Aluru, S. (2020). A comprehensive evaluation of long read error correction methods. BMC Genomics, 21(S6). https://doi.org/10.1186/s12864-020-07227-0

k-mer	Γ_1	Γ_2	Γ_3	Γ_4
GCA	36	19	14	57
CAG	18	13	56	39
AGT	11	54	33	28
GTA	44	27	**6**	49
TAC	49	44	27	**6**
ACC	**5**	48	47	26
CCG	22	**1**	60	43
CGA	24	7	50	45