

UC Riverside

UC Riverside Electronic Theses and Dissertations

Title

Novel Algorithmic Strategies for Improving De Novo Genome Assembly

Permalink

<https://escholarship.org/uc/item/35q306wt>

ISBN

9798241665003

Author

Chakravarty, Sakshar

Publication Date

2026-01-29

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Novel Algorithmic Strategies for Improving *De Novo* Genome Assembly

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Sakshar Chakravarty

March 2026

Dissertation Committee:

Professor Stefano Lonardi, Chairperson
Professor Evangelos Papalexakis
Professor Mingxun Wang
Professor Timothy Close

The Dissertation of Sakshar Chakravarty is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I am deeply grateful to my advisor, Professor Stefano Lonardi, without whose guidance and support I would not have reached this milestone. I truly believe he is the best advisor I could have asked for. He has always been kind, patient, and inspiring, making my Ph.D. journey both smooth and intellectually fulfilling. His consistent trust in my ideas and work has helped me grow into a confident and independent thinker. Without his mentorship, I would not have accomplished the publications and research achievements that form the foundation of this dissertation.

I would also like to sincerely thank all my co-authors throughout this Ph.D. journey, especially Hao Feng and Professor Glennis Logsdon, for their collaboration and valuable contributions. I extend my appreciation to the anonymous reviewers whose thoughtful feedback has significantly improved the quality of my published works. I am grateful to my Ph.D. committee members, Professors Evangelos Papalexakis, Mingxun Wang, and Timothy Close, for their constructive feedback and insightful comments that strengthened this dissertation. I also thank the Computer Science faculty and staff at UC Riverside for their support, as well as the funding agencies, NSF and NIH, whose financial support made this research possible.

I feel both fortunate and honored to have shared the lab with a group of brilliant minds and kind souls—Dr. Dipankar, Dr. Saleh, Amir, Faisal, Stamatis, Fatima, and Iftikhar. They have been my strongest supporters throughout my journey. I would also like to express my heartfelt thanks to my friends from the Riverside and Santa Barbara Bengali community, who welcomed me warmly and became my family thousands of miles away from home.

My wife, Laboni, has been a constant source of inspiration and motivation since she came into my life. Without her unwavering encouragement and support, I might not have been here

pursuing my Ph.D. so soon. She is the reason I continually strive to become a better version of myself every day.

My elder brother, Saumitra dada, will always be my academic idol. He introduced me to the beauty of Mathematics, Programming, and Biology, and throughout my academic journey, I have looked up to him for guidance and inspiration. He is the reason I pursued Computer Science and later ventured into the world of Bioinformatics.

Lastly, I am deeply indebted to my parents for always being there for me. They have given me the freedom to choose my path in life, both academically and professionally. I am grateful that I could honor their faith in me through every achievement, culminating in this Ph.D.

*To my beloved wife, **Laboni Sarker**, my elder brother, **Saumitra Chakravarty**, and my
parents, **Saraswati Rani Roy** and **Dulal Kumar Chakravarty**.*

ABSTRACT OF THE DISSERTATION

Novel Algorithmic Strategies for Improving *De Novo* Genome Assembly

by

Sakshar Chakravarty

Doctor of Philosophy, Graduate Program in Computer Science

University of California, Riverside, March 2026

Professor Stefano Lonardi, Chairperson

Approximately 8.7 million eukaryotic species exist on Earth, yet only a small fraction have been fully sequenced. Although sequencing costs continue to decline, *de novo* genome assembly remains challenging, requiring substantial expertise, computational resources, and time. From a computational standpoint, these challenges arise from the highly repetitive nature of eukaryotic genomes, short read lengths, uneven or ultra-deep sequencing coverage, sequencing errors, and chimeric reads. Despite decades of algorithmic advances, modern assemblers still struggle with highly repetitive regions that often harbor important functional and regulatory elements. The difficulty of producing complete assemblies is underscored by the 20-year effort required to resolve the final 8% of the human genome.

In this dissertation, we leverage single-copy k -mers (hereafter *unikmers*) for accurate read anchoring, overlap detection, and read partitioning in *de novo* genome assembly using highly accurate PacBio HiFi reads. *Unikmers* are defined as k -mers whose frequencies are close to the average sequencing depth, reflecting sequences that occur only once in the genome rather than sequencing errors. First, we introduce a statistical framework to identify *unikmers* directly from sequencing

reads without requiring a reference genome or a prior assembly. These *unikmers* are next applied to two separate genome assembly tasks.

As our first application, we introduce RAMbler, a genome assembler specialized for resolving complex repetitive regions from PacBio HiFi reads. RAMbler detects repetitive regions, barcodes the reads using *unikmers*, and uses these barcoded reads to generate a consensus assembly. We evaluate RAMbler against state-of-the-art assemblers including hifiasm, LJA, HiCANU, and Verkko on over 250 synthetic HiFi datasets spanning diverse repeat structures and sequencing conditions. RAMbler outperforms competing methods on most datasets and successfully reconstructs centromeres and other complex repeats in the human genome, achieving assembly quality comparable to the manually curated telomere-to-telomere human genome without manual intervention.

Finally, we address the largely unexplored problem of optimal read selection under ultra-deep sequencing conditions. We show that this problem is related to the NP-hard minimum tiling path problem and propose AWinK, a *unikmer*-based heuristic that selects informative read subsets. Across synthetic and real datasets, AWinK consistently produces highly contiguous and accurate assemblies, outperforming six alternative read-selection strategies.

Contents

1	Introduction	1
1.1	Limitations of Current Solutions	5
1.2	Single-copy k -mers for Genome Assembly	9
1.3	Thesis Organization	11
1.4	Publications	13
2	RAmbler: <i>De Novo</i> Genome Assembly of Complex Repetitive Regions	15
2.1	Introduction	15
2.1.1	Contributions	17
2.2	Methods	17
2.2.1	Problem Formulation	17
2.2.2	Repeat Identification	18
2.2.3	Analysis of k -mer Distribution	19
2.2.4	RAmbler’s Algorithm	23
2.3	Experimental Setup	27
2.3.1	Synthetic Data Generation	27
2.3.2	Performance Metrics	28
2.3.3	Parameter Optimization	29

2.4	Result Analysis	33
2.4.1	Comparing RAMbler, hifiasm, LJA, HiCANU, and Verkko on fixed-length repeats	33
2.4.2	Comparing RAMbler, hifiasm, LJA, HiCANU, and Verkko on variable-length repeats	37
2.4.3	Comparing RAMbler, hifiasm, LJA, HiCANU, and Verkko on repetitive regions with copy number variation	39
2.5	Discussion	42
2.6	Software Availability	53
3	RAMbler Resolves Complex Repeats in Human Chromosomes 8, 19 and X	54
3.1	Introduction	54
3.2	Methods	55
3.3	Performance Metrics	57
3.3.1	Assembly quality index (AQI)	57
3.3.2	Normalized error count in a window (N_w)	58
3.4	Parameter Optimization	58
3.5	Result Analysis	59
3.5.1	Experimental Results on <i>Homo sapiens</i>	59
3.5.2	Experimental Results on <i>Arabidopsis thaliana</i>	89
3.6	Discussion	93
4	Towards Optimal Selection of Ultra-Deep Sequencing Reads for <i>De Novo</i> Genome Assembly	94

4.1	Introduction	94
4.2	Preliminary Analysis	99
4.3	Methods	100
4.3.1	Problem Formulation	100
4.3.2	Optimal Selection	102
4.3.3	Analysis of k -mer Distribution	105
4.3.4	AWinK's Algorithm	107
4.4	Experimental Setup	110
4.4.1	Synthetic Data Generation	110
4.4.2	Performance Metrics	110
4.5	Result Analysis	111
4.5.1	Experimental Results on Synthetic Data	111
4.5.2	Experimental Results on Real Data	120
4.5.3	Runtime and Memory Usage	120
4.6	Discussion	121
4.7	Software Availability	123
5	Collaborative Projects	124
5.1	Prediction of DNA Methylation with Long-Range State-Space Models	124
5.1.1	Introduction	124
5.1.2	Problem Formulation	127
5.1.3	Methods	128
5.1.4	Experimental Results	131

5.1.5	Discussion	141
5.1.6	Software Availability	145
5.2	Visualizing the localization of multigene families with GFViewer	146
5.2.1	Background	146
5.2.2	Description	147
5.2.3	Software Availability	152
6	Conclusions	153
	Bibliography	156
A	RAmbler	171
A.1	Command-line listing	171
A.2	Supplemental Code	173
B	AWinK	174
B.1	Command-line listing	174
B.2	Supplemental Code	175
C	GFViewer	176
C.1	Test data	176
C.2	Supplemental Code	179

List of Figures

1.1	The value of contig N50 for over three thousand plant assemblies ordered by the year of publication; the type of sequencing technology is indicated by the color of each point; Legend: ‘HiFi’ = PacBio HiFi; ‘NGS’ = Illumina; ‘Other TGS’ = {PacBio CLR, ONT} (source: [134])	3
1.2	The distribution of contig N50 of plant genomes and pan-genomes ordered by the year of publication between 2000 and 2023 along with their level of completeness .	4
1.3	Number of human genomes assembled between 2000 and 2020 at either Chromosome or non-Chromosome level	5
1.4	The level of completeness for human genome assembly achieved over time between 2000 and 2021 (source: [106]). Note: Includes X but not Y chromosome and Count excludes mitochondrial DNA	10
1.5	A summary of components and tasks involved in <i>de novo</i> genome assembly pipeline	12

2.1	A frequency histogram of unresolved repeats in seven plant genomes (2015); the x -axis represents the repeat size (in kb); the y -axis represents the number of copies in each genome of those repeats' sizes (Figure reproduced with permission [24]) . . .	16
2.2	The problem formulation: tandem repeats $\alpha_1, \alpha_2, \dots, \alpha_t$ are over-collapsed by traditional assemblers; we want to reconstruct all the copies of the repetitive regions . . .	18
2.3	PacBio HiFi mapping coverage for Chromosome XII in <i>Saccharomyces cerevisiae</i> illustrated using NucFreq; the coverage spike indicates the presence of a repetitive region which is known to contain ~ 150 tandemly repeated copies of a 9.1 kb rDNA unit	19
2.4	PacBio HiFi k -mer distribution for $k = 17$ (A1), $k = 21$ (B1) and $k = 25$ (C1) for <i>Saccharomyces cerevisiae</i> ; precision, recall, and F1-score for $unikmers$ when $k = 17$ (A2), $k = 21$ (B2) and $k = 25$ (C2) for longer and longer intervals centered at the average sequencing coverage	20
2.5	Oxford Nanopore k -mer distribution for $k = 17$ (A1), $k = 21$ (B1) and $k = 25$ (C1) for <i>Saccharomyces cerevisiae</i> dataset (SRA accession SRR18365585, $\sim 50\times$ coverage); precision, recall, and F1-score for $unikmers$ when $k = 17$ (A2), $k = 21$ (B2) and $k = 25$ (C2) for larger and larger intervals centered at the average sequencing coverage	21

2.6	The algorithmic pipeline used in RAmbler	24
2.7	The three types of edges in our overlap graph	26
2.8	Performance of RAmbler for difference combinations of tolerance $to = \{1, 5, 10, 15, 20\}$, threshold, $th = \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$, repeat size = $\{10, 15, 20\}$ kb, number of copies = $\{2, 5, 10\}$, mutation rate $p = \{\frac{1}{100}, \frac{1}{250}, \frac{1}{500}, \frac{1}{1000}, \frac{1}{2000}\}$, on synthetic reads with coverage depth = $\{20\times, 30\times, 40\times\}$; the rows correspond to different values of to , the columns correspond to different values of th ; (A) number of contigs; the best performance is achieved for $to = \{15, 20\}$ and $th = \{10, 15\}$, (B) number of misassembled contigs; the best performance is achieved for $to = \{15, 20\}$ and $th = \{10, 15, 20, 25\}$, (C) effective genome fraction per contig ζ ; the best performance is achieved for $to = \{10, 15, 20\}$ and $th = \{10, 15, 20\}$, and (D) normalized NG50 η ; the best performance is achieved for $to = \{15, 20\}$ and $th = 15$	31
2.9	Assembly scores achieved by RAmbler for different combinations of tolerance $to = \{1, 5, 10, 15, 20\}$, threshold $th = \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$, repeat size = $\{10, 15, 20\}$ kb, number of copies = $\{2, 5, 10\}$, mutation rate $p = \{\frac{1}{100}, \frac{1}{250}, \frac{1}{500}, \frac{1}{1000}, \frac{1}{2000}\}$, on synthetic reads with coverage depth = $\{20\times, 30\times, 40\times\}$; the rows correspond to different values of to , the columns correspond to different values of th ; the optimal values are indicated by tick marks ($to = \{15, 20\}$ and $th = 15$) . .	32

2.10	Assembly statistics for RAMbler, hifiasm, LJA, HiCANU, and Verkko for 36 different combinations of synthetic data with repetitive regions having repeat sizes $\{15, 20\}$ kb, number of copies $\{5, 10\}$, mutation rate $p = \{1/250, 1/500, 1/1000\}$ and coverage depth $\{20\times, 30\times, 40\times\}$; (A) number of contigs, (B) effective genome fraction per contig ζ , (C) normalized NG50 η , and (D) assembly-score	34
2.11	Assembly statistics for RAMbler, hifiasm, LJA, HiCANU and Verkko on synthetic data with $30\times$ coverage depth for nine different combinations of repetitive regions containing five copies of a repeat size $= \{15, 20, 25\}$ kb and variable-length (up to $\pm 5\%$ per copy), mutation rate $p = \{\frac{1}{250}, \frac{1}{500}, \frac{1}{1000}\}$; (A) number of contigs, (B) number of misassembled contigs, (C) effective genome fraction per contig ζ , and (D) normalized NG50 η	36
2.12	Assembly score for RAMbler, hifiasm, LJA, HiCANU and Verkko on synthetic data with $30\times$ coverage depth for nine different combinations of repetitive regions containing five copies of a repeat size $= \{15, 20, 25\}$ kb and variable-length (up to $\pm 5\%$ per copy), mutation rate $p = \{\frac{1}{250}, \frac{1}{500}, \frac{1}{1000}\}$	37

2.13	Relationship among RAMbler’s performance, mutation rate p , and number of true uni/k mers; (A) Performance of RAMbler for several choices of the mutation rate p over 27 different combinations of repetitive regions with repeat sizes $\{10, 15, 20\}$ kb, number of copies $\{2, 5, 10\}$, coverage depth = $\{20\times, 30\times, 40\times\}$; (B) Number of true uni/k mers as a function of mutation rate and repeat unit’s sizes (five copies) .	52
3.1	RAMbler extension for resolving cycles in an overlap graph	56
3.2	NucFreq plots illustrating HiFi read mapping coverage for the comparison of RA.HG38.HG01346 assembly against GRCh38.p13 (HG38) of human Chromosomes 8, 19, and X; in all plots, blocks (i), (ii), and (iii) are non-centromeric repeats in Chromosome 19; blocks (iv) and (v) are the centromeric regions of Chromosomes 8 and X, respectively	61
3.3	Mapping coverage for HiFi HG01346 reads ($\sim 26\times$ coverage) on all chromosomes in GRCh38.p13 (HG38) and T2T-CHM13.v2.0 (T2T)	63
3.4	Alignment of the centromeres of GRCh38.p13 (HG38) and T2T-CHM13.v2.0 (T2T) against the reference optical map GM08331	70

3.5	Comparing RAmbler's assemblies (RA.HG38.HG01346 and RA.T2T.CHM13) against (A, B) the GRCh38.p13 assembly (HG38) and (C, D) the T2T-CHM13.v2.0 assembly (T2T) of human Chromosomes 8, 19, and X; in all plots, blocks (i), (ii), and (iii) are non-centromeric repeats in Chromosome 19; blocks (iv) and (v) are the centromeric regions of Chromosomes 8 and X, respectively; (A, C) NucFreq plots illustrating HiFi read mapping coverage (clipped at 60×); (B, D) CRAQ Circos plots illustrating (a) assembly quality index (AQI) score (higher is better), (b) base errors (fewer is better), and (c) misjoins (fewer is better)	72
3.6	NucFlag plots illustrating HiFi read mapping coverage for the comparison of (A) RA.HG38.HG01346 assembly against GRCh38.p13 (HG38) and (B) RA.T2T.CHM13 assembly against T2T-CHM13.v2.0 (T2T) of human Chromosomes 8, 19, and X; in all plots, blocks (i), (ii), and (iii) are non-centromeric repeats in Chromosome 19; blocks (iv) and (v) are the centromeric regions of Chromosomes 8 and X, respectively	73
3.7	Comparing assemblies RA.HG38.HG002 and RA.HG38.HG00733 against GRCh38.p13 (HG38) of human Chromosomes 8, 19, and X; in all plots, blocks (i), (ii), and (iii) are non-centromeric repeats in Chromosome 19; blocks (iv) and (v) are the centromeric regions of Chromosomes 8 and X, respectively; (A, C) NucFreq plots illustrating HiFi read mapping coverage; (B, D) CRAQ Circos plots illustrating (a) assembly quality index (AQI) score (higher is better), (b) base errors (lower is better), and (c) misjoins (lower is better)	78

3.8	NucFlag plots illustrating HiFi read mapping coverage for the comparison of (A) RA.HG38.HG002 assembly and (B) RA.HG38.HG00733 assembly against GRCh38.p13 (HG38) of human Chromosomes 8, 19, and X; in all plots, blocks (i), (ii), and (iii) are non-centromeric repeats in Chromosome 19; blocks (iv) and (v) are the centromeric regions of Chromosomes 8 and X, respectively	79
3.9	Synteny plots of RAmbler’s assemblies against GRCh38.p13 assembly (HG38) and the T2T-CHM13.v2.0 assembly (T2T) on human Chromosomes 19, 8, and X using SyRI; (A) Synteny plots between HG38 (blue), RA.HG38.HG002 (orange) and T2T (green); (B) Synteny plots between HG38 (blue), RA.HG38.HG00733 (orange) and T2T (green); (C) Synteny plots between HG38 (blue), RA.HG38.HG01346 (orange) and T2T (green)	80
3.10	Comparing RAmbler’s assembly (RA.T2T.CHM13) quality against the T2T-CHM13.v2.0 assembly (T2T) on human Chromosomes 19, 8, and X. Blocks (i), (ii), and (iii) are non-centromeric repeats in Chromosome 19; blocks (iv) and (v) are the centromeric regions of Chromosomes 8 and X; (A) NucFreq plots illustrating ultra-long ONT (link, $\sim 17\times$ coverage (≥ 100 kb)) read mapping coverage for T2T (top), and RAmbler (bottom); (B) CRAQ Circos plots illustrating regional AQI scores, base errors, and misjoins	82

3.11 Comparing RA.MAT002.HG002 against the maternal HG002 assembly (MAT002) of human Chromosomes 8, 19, and X; in all plots, blocks (i), (ii), and (iii) are non-centromeric repeats in Chromosome 19; blocks (iv) and (v) are the centromeric regions of Chromosomes 8 and X, respectively; (A) NucFreq plots illustrating HiFi read mapping coverage; (B) CRAQ Circos plots illustrating (a) assembly quality index (AQI) score (higher is better), (b) base errors (fewer is better), and (c) misjoins (fewer is better)	84
3.12 NucFlag plots illustrating HiFi read mapping coverage for the comparison of RA.MAT002.HG002 assembly against the maternal HG002 assembly (MAT002) of human Chromosomes 8, 19, and X; in all plots, blocks (i), (ii), and (iii) are non-centromeric repeats in Chromosome 19; blocks (iv) and (v) are the centromeric regions of Chromosomes 8 and X, respectively	85
3.13 Synteny plots of RAmbler’s assembly against T2T-CHM13.v2.0 (T2T) and the maternal HG002 assembly (MAT002) on human Chromosomes 19, 8, and X using SyRI; (A) Synteny plots between RA.T2T.CHM13 (blue), and T2T (orange); (B) Synteny plots between RA.MAT002.HG002 (blue) and MAT002 (orange)	87
3.14 Dotplot of contigs longer than 5 Mb produced by hifiasm using PacBio HiFi reads; (A) SRA accession ERR6210723 ($\sim 50\times$ coverage), and (B) SRA accession CRR302668 ($\sim 100\times$ coverage) using the TAIR10.1 assembly as reference	89

3.15	PacBio HiFi mapping coverage (SRA accession ERR6210723) for Chromosome 5 in <i>A. thaliana</i> illustrated using NucFreq; the coverage spikes indicate the presence of repetitive regions	90
4.1	Running hifiasm on (1) subset of reads selected at sequencing depth of 300× with uniform SLICES and (2) RANDOM sets at 300× (from a set of real HiFi read with sequencing depth of 3000×) for the genome of <i>B. divergens MOI</i> ; (top) length of the three longest assembled contigs produced by hifiasm; (bottom) number of assembled contigs produced by hifiasm	96
4.2	Running hifiasm on subset of reads selected using various strategies and sequencing depths from a set of real HiFi read with sequencing depth of 3000× for the genome of <i>B. divergens MOI</i> ; (top) length of the three longest assembled contigs produced by hifiasm; (bottom) number of assembled contigs produced by hifiasm	97
4.3	Problem formulation. Given a dataset of ultra-deep sequencing reads (top), AWinK selects a subset of reads that when given to hifiasm, it will produce a contiguous and complete <i>de novo</i> assembly (bottom); if the positions of the reads were known, the minimum tiling path (MTP) is the smallest subset of the reads that can be assembled to a complete genome (middle)	101

4.4	The length of the six longest contigs produced by hifiasm on a subset of reads selected (i) at random (200× coverage), (ii) with best average QV (200×), (iii) with longest (200×), and (iv) produced by the “oracle” at 5×, 10×, and 20× coverage selected from a set of PBSim synthetic HiFi reads with a sequencing depth of 1000× for the genome of <i>C. elegans</i> (contig1 is chromosome V); numerical values are available in Table 4.1	102
4.5	Dot plots comparing three hifiasm assemblies to the <i>C. elegans</i> reference genome using a subset of reads produced by the “oracle” at 5×, 10×, and 20× coverage selected from a set of PBSim synthetic HiFi reads with a sequencing depth of 1000×	103
4.6	The frequency distribution of <i>unikmers</i> in 20 kbp non-overlapping sliding windows across the six chromosomes of <i>C. elegans</i>	106
4.7	The frequency distribution of <i>unikmers</i> in a data set of synthetic HiFi reads at 1000× coverage for <i>C. elegans</i>	107
4.8	The algorithmic pipeline used in AWinK	108

4.9	Length of the six longest assembled contigs (top) and number of assembled contigs (bottom) produced by hifiasm on subset of reads selected by <i>AWinK</i> at different sequencing depths from synthetic HiFi reads with sequencing depth of 1,000× for the genome of <i>C. elegans</i> ; numerical values are available in Table 4.3	113
4.10	The genome fraction and the sequence identity of the assemblies produced by running hifiasm on various subset of reads selected by <i>AWinK</i> at different coverage depths from a set of PBSim simulated HiFi data with sequencing depth of 1,000× for the genome of <i>C. elegans</i>	116
4.11	NG50 and NGA50 of the assemblies produced by running hifiasm on various subset of reads selected by <i>AWinK</i> at different coverage depths from a set of PBSim simulated HiFi data with sequencing depth of 1,000× for the genome of <i>C. elegans</i>	116
4.12	Length of the six longest assembled contigs (top) and number of assembled contigs (bottom) produced by hifiasm on subset of reads selected by different strategies and coverage depths from a set of PBSim simulated HiFi data with sequencing depth of 1,000× for the genome of <i>C. elegans</i>	117
4.13	A visualization of the alignment between the three longest contigs of the hifiasm assembly from <i>AWinK</i> selected reads at 50× coverage and the optical map for <i>B. divergens MO1</i>	119

5.1	The HyenaDNA model structure; two Hyena layers constitute the architecture backbone (pre-training); a downstream decoder layer is used for classification (fine-tuning); input sequences are processed using a single-nucleotide tokenizer; the right side shows the components of the Hyena layer: the Hyena operator combines long convolutions with element-wise gating, and the long convolutions are implicitly parameterized by an MLP [94].	129
5.2	Context and species-specific accuracy comparisons between HyenaDNA and AMPS (sequence-only and sequence+annotation). In 15 contexts/species combinations out of 18 (83.4%), HyenaDNA models (irrespective of their sizes) outperformed AMPS with and without annotations.	134
5.3	Comparing the precision of HyenaDNA-436k and AMPS for several contexts and species	135
5.4	Comparing the recall of HyenaDNA-436k and AMPS for several contexts and species	136
5.5	Cross-context testing accuracy comparisons between HyenaDNA and AMPS. HyenaDNA shows consistent improvement in accuracy scores over AMPS, especially in cross-context cases between CG and CHG, with the 940k model outperforming both classes of AMPS in 12 of 12 such cases (100%).	137

5.6	Cross-species testing accuracy comparisons between HyenaDNA and AMPS. HyenaDNA-940k outperforms AMPS without annotations in a majority of cases, but falls behind AMPS with annotations.	138
5.7	Comparing DNABERT-2 and HyenaDNA-436k's runtime for different input sequence lengths.	139
5.8	Comparing the predictive accuracy of DNABERT-2 and HyenaDNA-436k on the cucumber CG context with respect to different input sequence length (both models were fine-tuned using a window size of 3,200 bp)	140
5.9	Comparing the accuracy of HyenaDNA-436k trained with embedding dropout only versus HyenaDNA-436k trained with both embedding and residual dropouts; the model was pre-trained on the cucumber genome and then fine-tuned using residual dropout of 0.1, in addition to 0.1 embedding dropout	141
5.10	Comparing HyenaDNA-436k's accuracy for various choices of the training dataset size (25%, 50%, 75%, and 100%) on the cucumber genome	142
5.11	Comparing the accuracy of HyenaDNA-436k using a window of 3,200 bp vs. 6,400 bp for all contexts and species; AMPS' accuracy with and without annotations is also included as a baseline	143

5.12 The homepage of GFViewer web application	148
5.13 Multi-gene family localization plot for top-6 gene families of <i>Babesia divergens</i>	
<i>MOI</i> across its chromosomes	151

List of Tables

1.1	DNA sequencing technologies (source: [33, 111])	2
1.2	A brief history of genome assembly tools; here, ‘DBG’ refers to De Bruijn Graph, ‘OLC’ refers to Overlap-Layout-Consensus, ‘SG’ refers to String Graph, and ‘Hybrid’ refers to a combination of more than one Data Structures into a single assembly pipeline	8
2.1	Data sets for <i>S. cerevisiae</i>	19
2.2	PBSim’s parameters and their values	27
2.3	RAmbler’s main parameters	30
2.4	Assembly score of RAmbler, hifiasm, LJA, HiCANU, and Verkko on 36 synthetic data sets; numbers in bold indicate the best assembly score on each row	35

2.5	Assembly score results for nine different data sets of synthetic HiFi reads with coverage $30\times$, based on a repetitive region with a variable-length (up to $\pm 5\%$ per copy) repeat unit of length $\{15, 20, 25\}$ kb, and $p = \{1/250, 1/500, 1/1000\}$; numbers in bold indicate the best assembly score on each row	38
2.6	Number of contigs and assembly size for the assemblies produced by RAMbler, hifiasm, LJA, and HiCANU; assemblers were given in input synthetic HiFi reads with coverage $30\times$ generated from synthetic diploid genomes with repeat unit of length $\{5, 10, 15\}$ kb; in each synthetic genome the primary haplotype (hap1) had five copies of the repeat unit, while the secondary haplotype (hap2) had 3-5 copies	43
2.7	Number of contigs and assembly size for the assemblies produced by RAMbler, hifiasm, LJA, and HiCANU; assemblers were given in input synthetic HiFi reads with coverage $30\times$ generated from synthetic diploid genomes with repeat unit of length $\{20, 50, 100\}$ kb; in each synthetic genome the primary haplotype (hap1) had five copies of the repeat unit, while the secondary haplotype (hap2) had 3-5 copies	44

2.8	Number of contigs and assembly size for the assemblies produced by RAmbler, hifiasm, LJA, and HiCANU; assemblers were given in input synthetic HiFi reads with coverage $30\times$ generated from synthetic diploid genomes with repeat unit of length $\{5, 10, 15\}$ kb; in each synthetic genome the primary haplotype (hap1) had ten copies of the repeat unit, while the secondary haplotype (hap2) had 6, 8 or 10 copies	45
2.9	Number of contigs and assembly size for the assemblies produced by RAmbler, hifiasm, LJA, and HiCANU; assemblers were given in input synthetic HiFi reads with coverage $30\times$ generated from synthetic diploid genomes with repeat unit of length $\{20, 50, 100\}$ kb; in each synthetic genome the primary haplotype (hap1) had ten copies of the repeat unit, while the secondary haplotype (hap2) had 6, 8 or 10 copies	46
2.10	Number of repeat copies resolved and number of haplotype switching in the assemblies produced by RAmbler, hifiasm, LJA, and HiCANU; assemblers were given in input synthetic HiFi reads with coverage $30\times$ generated from synthetic diploid genomes with repeat unit of length $\{5, 10, 15\}$ kb; in each synthetic genome the primary haplotype (hap1) had five copies of the repeat unit, while the secondary haplotype (hap2) had 3-5 copies	47

2.11	Number of repeat copies resolved and number of haplotype switching in the assemblies produced by RAmblr, hifiasm, LJA, and HiCANU; assemblers were given in input synthetic HiFi reads with coverage 30× generated from synthetic diploid genomes with repeat unit of length {20, 50, 100} kb; in each synthetic genome the primary haplotype (hap1) had five copies of the repeat unit, while the secondary haplotype (hap2) had 3-5 copies	48
2.12	Number of repeat copies resolved and number of haplotype switching in the assemblies produced by RAmblr, hifiasm, LJA, and HiCANU; assemblers were given in input synthetic HiFi reads with coverage 30× generated from synthetic diploid genomes with repeat unit of length {5, 10, 15} kb; in each synthetic genome the primary haplotype (hap1) had ten copies of the repeat unit, while the secondary haplotype (hap2) had 6, 8 or 10 copies	49
2.13	Number of repeat copies resolved and number of haplotype switching in the assemblies produced by RAmblr, hifiasm, LJA, and HiCANU; assemblers were given in input synthetic HiFi reads with coverage 30× generated from synthetic diploid genomes with repeat unit of length {20, 50, 100} kb; in each synthetic genome the primary haplotype (hap1) had ten copies of the repeat unit, while the secondary haplotype (hap2) had 6, 8 or 10 copies	50

2.14	Summary of assembly performance of RAMbler, hifiasm, LJA, and HiCANU on synthetic diploid genomes with copy number variation	51
3.1	Data sets for <i>H. sapiens</i>	55
3.2	Data sets for <i>A. thaliana</i>	55
3.3	Comparison of the assemblies produced by RAMbler, hifiasm, LJA, HiCANU and Verkko on the HiFi reads extracted around the centromeres of human Chromosomes 8 and X, for several choices of input assemblies and HiFi reads. Note: ‘-’ means that the assembler timed out after 96 hours, ‘N/A’ means that the assembler finished execution but produced no output, numbers in bold indicate the assemblies with the fewest contigs in each row	62
3.4	Assembly statistics for RAMbler, hifiasm, LJA, HiCANU and Verkko on three non-centromeric repeats of Chromosome 19 from the T2T-CHM13.v2.0 assembly (T2T) and the maternal HG002 assembly (MAT002) using the corresponding HiFi reads as inputs	65
3.5	Assembly statistics for RAMbler, hifiasm, LJA, HiCANU and Verkko on three non-centromeric repeats of Chromosome 19 using HiFi reads HG002, HG00733, HG01346 and GRCh38.p13 (HG38) as input assembly	66

3.6	Changes in HiFi reads' coverage for each Chromosome when segmental duplications (SD) are masked	67
3.7	Comparing the number of selected HiFi HG002 reads when the GRCh38.p13 assembly (HG38) or the maternal HG002 assembly (MAT002) were used as the input reference assembly to Rambler	68
3.8	Comparing Rambler's assemblies (RA.HG38.HG002, RA.HG38.HG00733, and RA.HG38.HG01346) for selected regions of human Chromosomes 19, 8, and X against the GRCh38.p13 assembly (HG38) using CRAQ; CRAQ reports assembly coverage rate (i.e., the fraction of the genome assembled), Regional Assembly Quality Indicator (R-AQI) score (higher is better) and Structural Assembly Quality Indicator (S-AQI) score (higher is better); CRAQ was provided Illumina reads ($\sim 13.5\times$ coverage) and long PacBio HiFi reads ($\sim 66\times$ coverage) for HG002; Illumina reads ($\sim 12\times$ coverage) and long PacBio HiFi reads ($\sim 33\times$ coverage) for HG00733; Illumina reads ($\sim 10\times$ coverage) and long PacBio HiFi reads ($\sim 26\times$ coverage) for HG01346; numbers in bold indicate the best scores	75

3.9	QUAST scores for human Chromosomes 8, 19, and X from GRCh38.p13 (HG38) and RAmbler (RA.T2T.CHM13, RA.HG38.HG002, RA.HG38.HG00733 and RA.HG38.HG01346) using T2T-CHM13.v2.0 (T2T) as the ground truth; the length of the chromosomes are 146,259,331 bp, 61,707,364 bp, and 154,259,566 bp, respectively, and from GRCh38.p13 (HG38) and RAmbler (RA.MAT002.HG002 and RA.HG38.HG002) using the maternal strand of HG002 (MAT002) as the ground truth; the length of the chromosomes are 146,740,790 bp, 61,317,360 bp, and 154,341,406 bp respectively	76
3.10	CRAQ report on assembly coverage and errors obtained using short Illumina reads (SRA accession SRR10035392, $\sim 17\times$ coverage) and PacBio HiFi reads (SRA accession SRR11292120-SRR11292123, $\sim 32.4\times$ coverage) on human Chromosomes 19, 8, and X from T2T-CHM13.v2.0 (T2T) and RAmbler (RA.T2T.CHM13); numbers in bold indicate the best scores	81
3.11	CRAQ report on assembly coverage and errors obtained using short Illumina reads (SRA accession SRR10035392, $\sim 17\times$ coverage) and ultra-long Oxford Nanopore reads (link, $\sim 17\times$ coverage (≥ 100 kb)) on human Chromosomes 19, 8, and X from T2T-CHM13.v2.0 (T2T), and RAmbler (RA.T2T.CHM13); numbers in bold indicate the best scores	83

3.12	CRAQ report on assembly coverage and errors obtained using short Illumina reads (illuminaHG002, $\sim 13.5\times$ coverage) and PacBio HiFi reads (hifiHG002, $\sim 66\times$ coverage) on human Chromosomes 19, 8, and X from the maternal HG002 assembly (MAT002) and RAmbler (RA.MAT002.HG002); numbers in bold indicate the best scores	86
3.13	Time and memory usage for RAmbler for the reconstruction of the centromeres of Chromosome 8 and X	88
3.14	Assembly statistics for RAmbler, hifiasm, LJA, HiCANU and Verkko on three centromeric repeats in Chromosome 5 of <i>A. thaliana</i> using TAIR10.1 as input assembly; Symbol ‘-’ means that the assembler timed out after 48 hours, ‘N/A’ means that the assembler finished execution with no output	91
3.15	Repeat resolution statistics for RAmbler, hifiasm, LJA, HiCANU and Verkko on real HiFi reads (SRA accession ERR6210723) for three repeats of <i>A. thaliana</i> in Chromosome 5; HiCANU required more than 40 hours to complete all these assemblies and Verkko failed to generate outputs in one case	92

4.1	The length of the six longest contigs, number of assembled contigs and the total assembly length produced by hifiasm on a subset of reads selected (i) at random (200× coverage), (ii) with best average QV (200×), (iii) with longest (200×), and (iv) produced by the “oracle” at 5×, 10×, and 20× coverage selected from a set of PBSim synthetic HiFi reads with a sequencing depth of 1000× for the genome of <i>C. elegans</i> (contig1 is chromosome V)	103
4.2	PBSim’s parameters and their optimal values	110
4.3	Length of the six longest assembled contigs, number of assembled contigs and the total assembly length produced by hifiasm on subset of reads selected by AWinK at different sequencing depths from synthetic HiFi reads with sequencing depth of 1000× for the genome of <i>C. elegans</i>	114
4.4	Average length of short contigs produced by hifiasm on subset of reads selected by AWinK at sequencing depths = {100×, 150×, 200×} from synthetic HiFi reads with sequencing depth of 1000× for the genome of <i>C. elegans</i>	115
4.5	Comparing the metrics for hifiasm assemblies produced by various read selection strategies for <i>C. elegans</i> ; numbers in bold indicate the best metric	118

4.6	Runtime and Peak memory usage for hifiasm assemblies produced by AWinK selected reads of <i>B. divergens</i> <i>MOI</i> and <i>C. elegans</i> for the target coverage depths of 50× and 100×	121
5.1	Hyperparameters of HyenaDNA	130
5.2	Data sets used for training the models in this study; data sets were capped to a maximum size of 500,000 with a 50:50 split between methylated and unmethylated samples; the right-most column shows the predictive accuracy of HyenaDNA-436k on each context and species	132
5.3	Plant genomes used in this study	133
5.4	A toy example for gene-family data file	149
5.5	Summary of parameters and options available in GFViewer; (* indicates the required parameters)	150

Chapter 1

Introduction

Approximately 8.7 million eukaryotic species exist on our planet, yet only a few tens of thousands have had their genomes sequenced [90]. Prokaryotes are also likely to number in the tens of millions. Assembling a large eukaryotic genome is still a challenging task: it requires significant expertise, money, computational resources, and time. To enable Life Scientists to access the genomes of more species, the quality of sequencing data must continue to increase while the cost must continue to decrease. At the same time, more efficient and accurate assembly algorithms must become available.

Given the broad biological impact of obtaining the genome for a new organism, *de novo* genome assembly is one of the most critical problems in computational biology. Despite tremendous algorithmic progress, the problem is not yet completely solved. The assembly problem remains challenging due to the highly repetitive content of eukaryotic genomes, short read length, uneven sequencing coverage, non-uniform sequencing errors and chimeric reads [22, 135]. Repetitive regions (or segmental duplications) are the primary reasons for which *de novo* genome assemblies are often

Generation	Name	Avg. read length	Accuracy (%)	Throughput (Mb/h)	Cost/Gbp (USD)
1st	Sanger	500-600 bp	99.99	0.1	Very high
2nd (NGS)	Illumina	50-600 bp	99.9	170-67000	5-150
	SOLiD	35-50 bp	99.9	5-15	60-130
3rd	PacBio CLR	10-30 kbp	87	1300	7.2-43.3
	Oxford Nanopore	10-100 kbp	92-97	700	7-100
	PacBio HiFi	10-25 kbp	99.8	< CLR/ONT	~100

Table 1.1: DNA sequencing technologies (source: [33, 111])

fragmented and incomplete. A large fraction of eukaryotic genomes is made of repetitive elements, including satellite DNA, mini-satellites, micro-satellites, and DNA/RNA transposons. Several studies have shown that expansions or mutations of these repetitive regions are linked to a variety of human diseases, ranging from neurological diseases to cancers (see [115, 57, 49] for a review). The inability of sequencing instruments to generate reads that can span long repeats has consistently challenged genome assembly algorithms.

Table 1.1 describes different generations of sequencing technologies and platforms with respect to the average length of reads, accuracy, throughput and cost. The third generation of sequencing technology on the market, e.g., Pacific Biosciences (PacBio) [37, 102, 109, 55, 66] and Oxford Nanopore (ONT) [30, 103, 7, 81], offers longer reads at a higher cost per base than the second generation, but the sequencing error rate is much higher. The introduction of PacBio HiFi sequencing at the end of 2019 has been a “game-changer” in genome assembly, because it can produce read lengths typically ranging 10-25 kb with accuracy greater than 99.8% [133]. HiFi sequencing greatly improved human assemblies [96, 101, 116, 44] and many other genomes (see, e.g.,

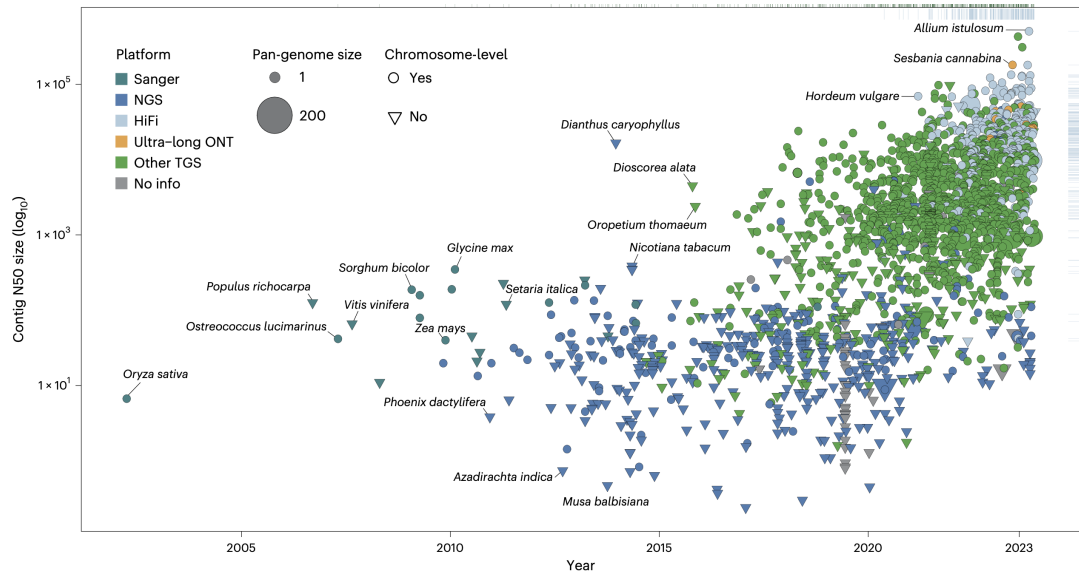


Figure 1.1: The value of contig N50 for over three thousand plant assemblies ordered by the year of publication; the type of sequencing technology is indicated by the color of each point; Legend: ‘HiFi’ = PacBio HiFi; ‘NGS’ = Illumina; ‘Other TGS’ = {PacBio CLR, ONT} (source: [134])

[134, 51]).

The number of species being sequenced and assembled is increasing rapidly. A study published on 04/2024 reports on a comprehensive analysis of over three thousand plant genomes assembled from 2000 to 2023 [134]. From 2000 to 2020, 1,144 genomes of 782 plant species were sequenced and assembled. From 2021 to 2023, 2,373 genomes of 1,031 plant species, including 793 newly sequenced species, were sequenced and assembled. Interestingly, out of the 2,373 newly assembled genomes, only 63 were chromosome-level assemblies, which is a testimony to the difficulty of achieving such a level of completeness. Figure 1.1 (reproduced from [134]) shows the value of contig N50 for these three thousand plant assemblies ordered by the year of publication. A few observations are in order. (i) PacBio HiFi (light blue) dominates for genome assembly in

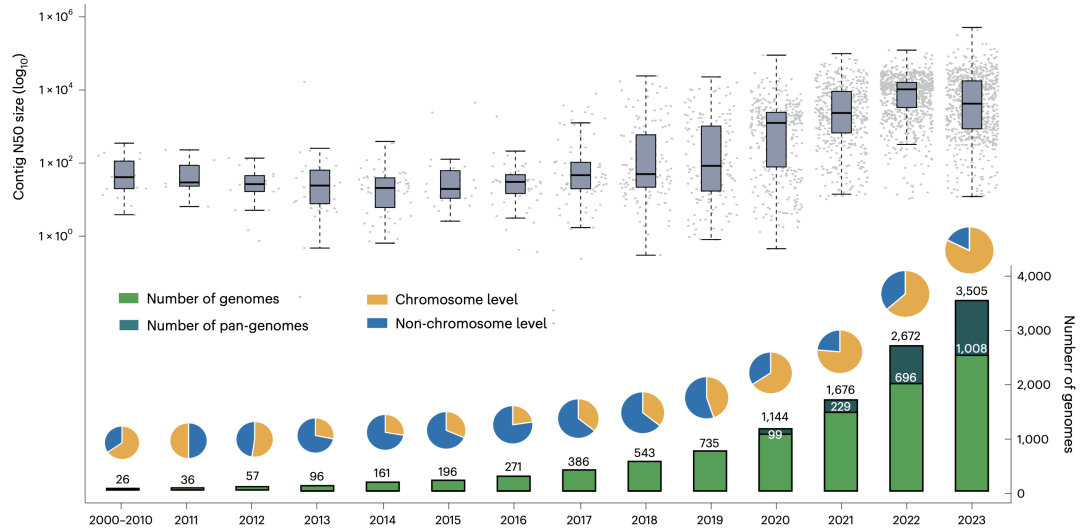


Figure 1.2: The distribution of contig N50 of plant genomes and pan-genomes ordered by the year of publication between 2000 and 2023 along with their level of completeness

the last few years and increasing contiguity (N50), (ii) ultra-long Oxford Nanopore (orange) is rare, likely due to its high cost, (iii) Illumina (dark blue color) has not been used much in the last few years, probably because it yields low contiguity assemblies, (iv) the “other-TGS” platforms which includes PacBio CLR and ONT are still heavily used. The authors of [134] reported that PacBio CLR, PacBio HiFi and ONT accounted for 94.0% of the genome sequencing projects in the past three years, with the remaining 6.0% using Illumina technology. For the few chromosome-level assemblies, hifiasm [26] was the assembler predominantly used for contig assembly, nextpolish [54] was the tool predominantly used for polishing, and 3D-DNA [36] was the most used scaffolding tool. Additionally, Figure 1.2 shows the distribution of contig N50 of plant genomes and pan-genomes assembled between 2000 and 2023 depicting the ratio between the numbers of individual plant genomes and pan-genomes assembled at either Chromosome or non-Chromosome level.

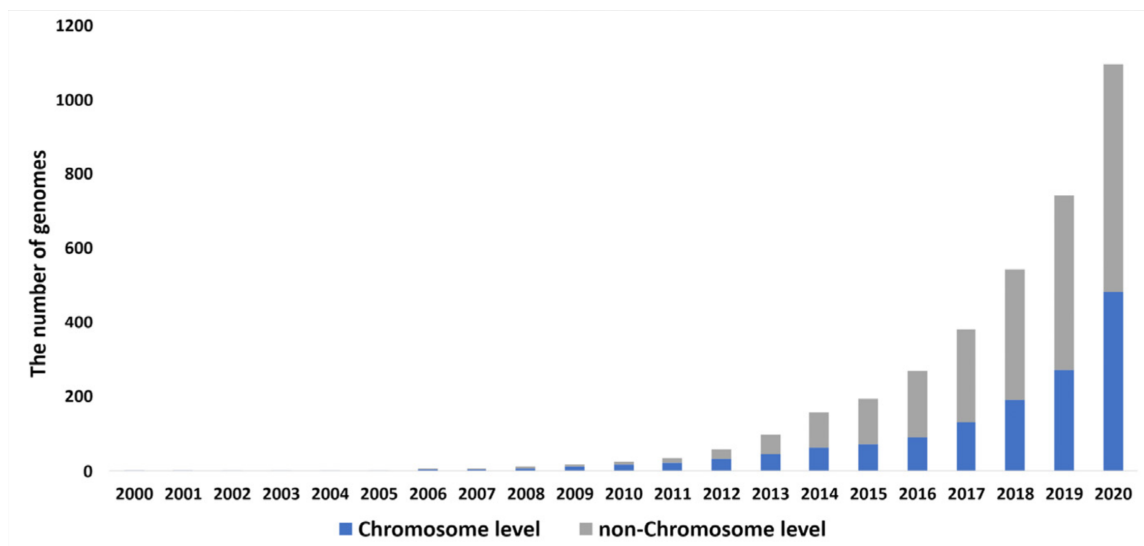


Figure 1.3: Number of human genomes assembled between 2000 and 2020 at either Chromosome or non-Chromosome level

1.1 Limitations of Current Solutions

The state-of-the-art in genome assembly is probably best illustrated by 30+ years of efforts to obtain the complete sequence of the human genome. In the summer of 2000, the Human Genome Sequencing Consortium announced a draft of the human genome sequence that accounted for 90% of the genome and contained more than 150,000 gaps. In 2003, the consortium announced that it had generated an “essentially complete” human genome sequence that accounted for 92% of the human genome and had about 400 gaps. The telomere-to-telomere (T2T) human genome sequencing project took advantage of PacBio HiFi and ultra-long ONT reads and in April 2023 (twenty years later), they finally closed almost all the gaps in the remaining 8% of the genome and achieved 99.9% completeness [87, 79, 53, 95, 105]. Figure 1.3 shows the number of human genomes assembled be-

tween 2000 and 2020 at either Chromosome or non-Chromosome level and Figure 1.4 illustrates the level of completeness for human genome assembly achieved over time. As shown in Figure 1.4, by 2021, researchers had resolved nearly all previously incomplete regions of the human reference genome, producing an assembly of approximately 3.05 billion base pairs, with an estimated 0.3% of the sequence still containing errors.

The data for the T2T-CHM13 included 30× PacBio HiFi, 120× of ultra-long Oxford Nanopore, 70× of PacBio CLR, 50× of 10X Genomics, Bionano optical maps, and 70× Illumina HiC [87, 79]. CHM13 is a haploid genome: it originated from an ovum without maternal chromosomes fertilized by one sperm, leading to two nearly identical paternal haploid complements with an X chromosome, eliminating the need to phase haplotypes and avoid associated diploid assembly errors. Completing the T2T-CHM13 assembly required substantial resources, including manual curation by dozens of people over several years, with different groups focused on each chromosome.

Another recent human genome project further exemplifies the challenges of obtaining an error-free gapless assembly. The recent HG002 human genome assembly was carried out by the Human Pangenome Reference Consortium (HPRC) using PacBio HiFi, PacBio CLR, Oxford Nanopore, ultra-long Oxford Nanopore, HiC (Illumina), 10X Genomics, Bionano optical map, and Strand-Seq. HPRC used the Verkko assembler [105], followed by intense manual curation, namely (i) ONT-based patching to resolve HiFi coverage gaps, (ii) manual resolution of tangles, and (iii) Strand-Seq and HiC based scaffolding. Despite these efforts and cost, the v0.7 assembly released at the end of 2022 had nine multi-million base pair gaps in chromosomes 13, 14, 15, 21, and 22 corresponding to the rDNA arrays and 352 regions were flagged having assembly issues. A team of curators identified more than 33,000 patches to obtain v1.0 after an additional year of work. The

nine multi-million base pair gaps in chromosomes 13, 14, 15, 21, and 22 are still present in the v1.0 assembly.

The assembly of HG002 offered an opportunity to test and compare software pipelines for genome assembly. In a companion paper [59], many assembly pipelines were tested, including CANU [70], HiCANU [96], CrossStitch, DipAsm [44], FALCON Unzip [29], Flye [69], hifiasm [26], MaSuRCA [143], NECAT [25], Peregrine [28], Shasta [114] and wtdbg2 [110] (Table 1.2 provides a summary of different genome assembly tools utilizing different sequencing technologies along with brief descriptions). The authors noted that “*Bionano maps and HiC revealed a handful to several hundred scaffolding errors per assembly, including: missed joins, misjoins, and erroneous inversions or false duplications. There were also within contigs errors: chimeric joins without a gap; sequence expansions; and sequence collapses.*” They also noted that “*There was no one approach, without using a highly curated reference, that was free of one or more scaffold or contig errors in an automated process . . . each approach avoided at least one type of error that others did not.*”

As mentioned previously, in the plant world, out of the 2,373 plant genomes reported in [134], only 63 were telomere-to-telomere (or chromosome-level) assemblies. Regarding the notion of T2T, the authors of [134] noted that “*despite the claimed high quality, genome quality varies substantially owing to a lack of independent third-party assessment using uniform parameters . . . In this study, genomes were considered to be T2T only if they contained at least half of the total number of chromosomes with both ends (that is, telomeres) assembled.*” In the context of their paper, T2T does not mean gapless.

As it is evident from all these efforts to obtain a gapless T2T assembly for human and plant

Tool	Year/s	Data Structures	Technologies	Description
Velvet [139]	2007/11	DBG	Illumina	Small genomes
Edena [50]	2008	OLC	Illumina	Small bacterial genomes
ABYSS [118, 58]	2009/17	DBG	Sanger, NGS	Large genome assembly of short reads (genomic & transcriptomic)
Ray [13]	2010	DBG	Roche/454, Illumina	A suite of assemblers including <i>de novo</i> , metagenomic, ontology and taxonomic profiling
SGA [117]	2012	SG	NGS	Large genomes from high coverage short reads
SPAdes [9]	2012/21	DBG	NGS, PacBio SMRT, ONT	Small genomes, single-cell
Falcon [29]	2014/17	Hybrid	PacBio SMRT	Diploid genomes
Miniasm [74]	2016	SG	PacBio SMRT, ONT	Small and large, haploid/diploid genomes, memory efficient
CANU [70, 96]	2001/18	Hybrid	PacBio SMRT, ONT, HiFi	Small and large, haploid/diploid genomes
Flye [69]	2019	DBG	PacBio SMRT, ONT, HiFi	Long-read assembly for managing repetitive and complex genomic regions via repeat graphs
hifiasm [26]	2021	Hybrid	PacBio HiFi	Specialized for haplotype resolved assembly
LJA [8]	2022	DBG	PacBio HiFi	Automated assemblies of long, HiFi reads
Verkko [105]	2023	Hybrid	PacBio HiFi, ONT	Telomere-to-telomere assembly of diploid chromosomes

Table 1.2: A brief history of genome assembly tools; here, ‘DBG’ refers to De Bruijn Graph, ‘OLC’ refers to Overlap-Layout-Consensus, ‘SG’ refers to String Graph, and ‘Hybrid’ refers to a combination of more than one Data Structures into a single assembly pipeline

genomes, eukaryotic *de novo* genome assembly requires not only a significant amount of resources to generate all kinds of sequencing data and genome-wide maps, but also intense and laborious manual curation. In this thesis, we address some of these challenges by focusing on the problem of closing the gaps, e.g., assembling complex repetitive sequences [21, 22] and dealing with the assembly problem of ultra-deep sequencing HiFi data.

The problem of reconstructing repetitive regions (segmental duplications) has been addressed a few times in the literature, but all published works assume that the input reads have a relatively high sequencing error rate (e.g., PacBio CLR or ONT). For instance, the Segmental Duplication Assembler (SDA) constructs graphs where paralogous sequence variants define the nodes and long-read sequences provide either attraction and repulsion edges allowing the tool to parti-

tion and assemble noisy long reads corresponding to distinct paralogs [128]. Experimental results, however, show that most of the SDA assembled paralogs are not linked to their flanking unique sequences, i.e., remain orphan contigs [34]. In addition, SDA “*is no longer maintained and should not be used . . . assembly tools like Flye, HiCanu, and hifiasm outperform any results previously possible with SDA.*” (quote from the [SDA github](#)). HERA addresses the same problem by building an undirected overlap graph consisting of two types of nodes: anchoring nodes to represent pre-assembled contigs and read nodes to represent long noisy reads [34]. Noisy raw reads are corrected in HERA by using the self-correction module in CANU [70], then the overlaps between all sequences including the contigs and the corrected reads are identified using BWA. Using the overlap graph, HERA tries to maximize the number of identified anchoring node pairs that represent adjacent contigs. CentroFlye [16] was also designed with ultra-long error-prone reads in mind. While experimental results showed that centroFlye can reconstruct the centromeres of human chromosomes 6 and X, its memory usage is prohibitive (800 GB of RAM to reconstruct the X centromere).

1.2 Single-copy k -mers for Genome Assembly

A single copy k -mer (hereafter called *unikmer*) is a k -mer that occurs a number of times approximately equal to the average sequencing depth μ in the input reads. Please note that *unikmers* are not k -mers that appear only once in the reads: those k -mers typically correspond to sequencing errors. For instance, if the average sequencing depth is $\mu = 30\times$ and $k = 21$, all 21-mers that appear about 30 times in the set of all reads are *unikmers*. Since these k -mers are expected to appear only once in the genome, they can be used as “anchors”. The main idea is that we can determine a reliable mapping of a read r in the genome if the set of *unikmers* in r matches another set of *unikmers*

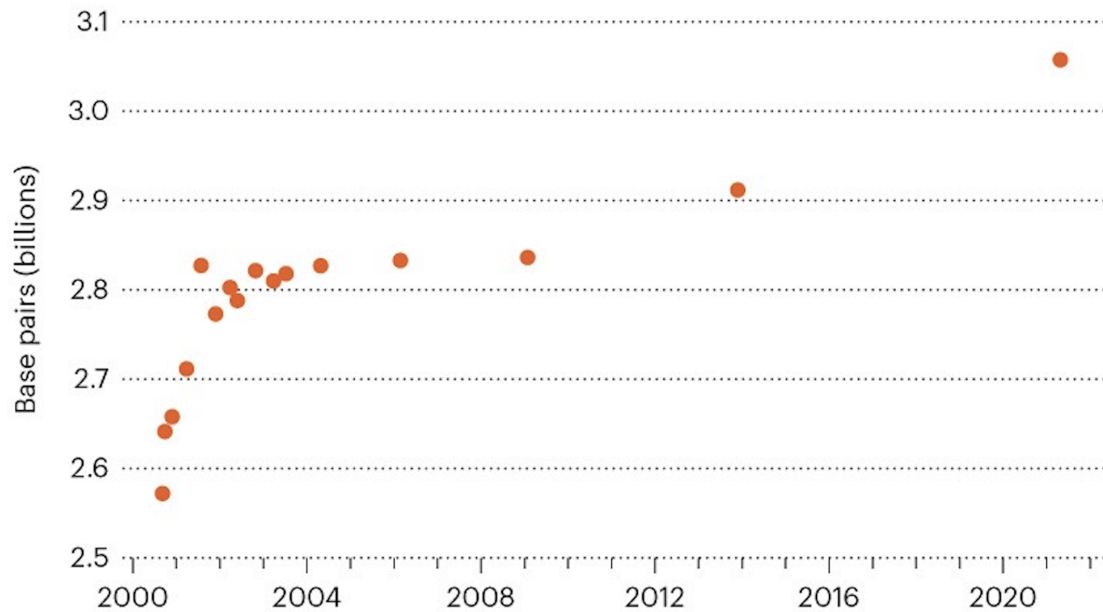


Figure 1.4: The level of completeness for human genome assembly achieved over time between 2000 and 2021 (source: [106]). Note: Includes X but not Y chromosome and Count excludes mitochondrial DNA

in a particular position in the genome (in the same order and relative distance). Similarly, we can determine whether two reads are likely to be overlapping in the genome if they share a sufficient number of common *unimers* (in the same order and relative distance). Please observe that for this idea to work, the reads must be essentially error-free, thus the need for PacBio HiFi reads.

When dealing with the assembly of repetitive regions, we assume that the copies in the repetitive region (called *monomers*) are not identical to each other. If all the monomers were identical, the problem of resolving all the copies of the repetitive region would be impossible (unless we can get reads long enough that they would span the entire repetitive region). We can therefore rely on the presence of mutations (or variants) to distinguish and partition the HiFi reads that belong to

different repeat copies within a repetitive region. In other words, we expect each monomer to have its own “signature” of variations. Each mutation is likely to induce a *unikmer*, i.e., a k -mer that occurs a number of times approximately equal to the average sequencing depth in the reads.

To the best of our knowledge, the first use of single-copy k -mers dates to 2010 [122]. In their paper, Sudmant et al., define the notion of a *singly unique nucleotide* (SUN) sequence in a genome. The authors claimed that “*SUNs are a distinct class of paralogous sequence variants that uniquely tag a specific paralog*” and used them to provide accurate estimates of copy number variation and multi-copy genes in the human genome. Ten years later, the same lab developed a method to assemble human chromosome 8 by exploiting the use of single-copy k -mers to resolve complex repetitive regions in its centromere [79]. CentroFlye [16] also used *rare* k -mers (although not necessarily single-copy) to classify reads that belong to the repetitive regions. The same group used the idea of using rare k -mers for the problem of mapping long reads accurately in the presence of repeats [88, 15, 17]. The latest incarnation of this idea appears in the mapping tool UniAligner [17], which uses shortest unique substrings (i.e., variable-length k -mers) as anchors for the dynamic programming alignment.

1.3 Thesis Organization

Figure 1.5 summarizes different components and tasks involved in a general *de novo* genome assembly pipeline. As mentioned above, despite thirty years of tremendous algorithmic progress driven by advances in sequencing technology, the problem of *de novo* genome assembly is not yet completely solved. In this thesis, we focus on two specific aspects of the problem that are still unresolved utilizing *unikmers* and PacBio HiFi reads exclusively, namely (i) assembling highly repetitive regions

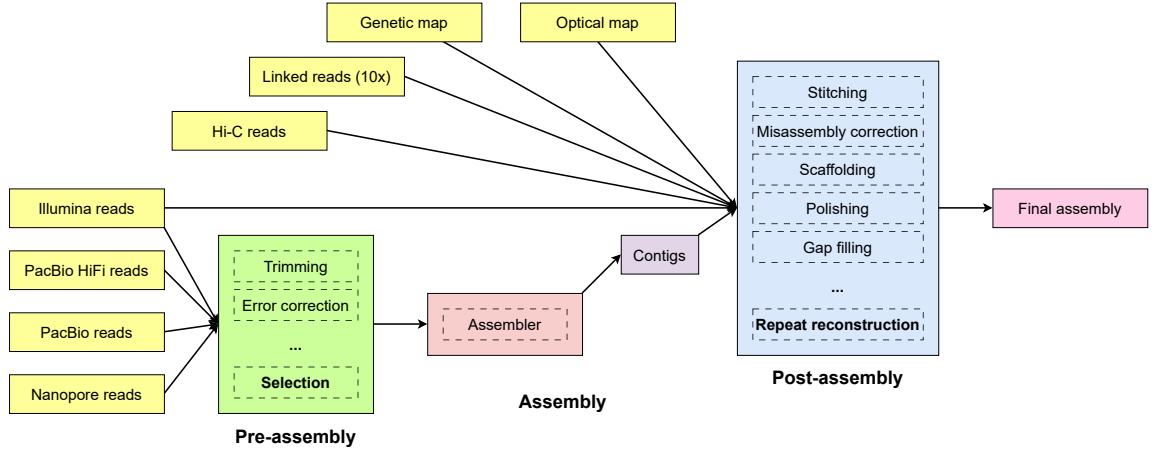


Figure 1.5: A summary of components and tasks involved in *de novo* genome assembly pipeline

of the genome [21, 22] which is a post-assembly task, and (ii) assembling ultra-deep sequencing data (i.e., sequencing depth higher than $1,000\times$) via optimal selection of reads [23] which is a pre-assembly task as shown in Figure 1.5.

In Chapter 2, we introduce a statistical framework for extracting *unikmers* from long-read sequencing data and propose a novel repeat assembler. We conduct extensive experimental evaluations on synthetic datasets to benchmark our approach against several state-of-the-art assemblers. In Chapter 3, we extend this repeat assembly framework to handle cycles in overlap graphs, enabling improved reconstruction of more complex repetitive regions. We then apply our method to assemble multiple repetitive regions from the genomes of *Homo sapiens* and *Arabidopsis thaliana*, and assess assembly quality by comparison with the best available telomere-to-telomere (T2T)-level reference assemblies. In Chapter 4, we address the challenge of *de novo* genome assembly under ultra-deep sequencing coverage by proposing an approximation algorithm that leverages *unikmers* for optimal read selection. We evaluate the effectiveness of this approach through comprehensive comparisons with several widely used read selection strategies. Chapter 5 presents additional studies arising from

collaborative research efforts. In Section 5.1, we investigate DNA methylation patterns across six plant species and introduce a deep learning framework inspired by large language models (LLMs) to predict cytosine methylation at low-coverage sites [40]. In Section 5.2, we describe a visualization tool for analyzing the genomic localization of multigene families [119, 120]. Finally, we summarize our contributions and discuss future research directions in Chapter 6.

1.4 Publications

This section lists the publications produced as part of the research conducted for this thesis:

- **Chakravarty S**, Logsdon G, and Lonardi S. 2023. “Rambler: de novo genome assembly of complex repetitive regions”. In *Proceedings of the 14th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics (BCB '23)*. Association for Computing Machinery, New York, NY, USA, Article 13, 1. <https://doi.org/10.1145/3584371.3612971> → Chapter 2
- **Chakravarty S**, Logsdon G, and Lonardi S. “Rambler resolves complex repeats in human Chromosomes 8, 19, and X”. *Genome Research*. 2025; 35(4): 863–876. <https://doi.org/10.1101/gr.279308.124> → Chapter 3
- **Chakravarty S** and Lonardi S. 2025. “Towards optimal selection of ultra-deep sequencing reads for de novo genome assembly”. In *Proceedings of the 16th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics (BCB '25)*. Association for Computing Machinery, New York, NY, USA, Article 42, 1–6. <https://doi.org/10.1145/3765612.3767210> → Chapter 4

- Feng H, **Chakravarty S**, and Lonardi S. “Prediction of DNA Methylation with Long-Range State-Space Models”. *IEEE Transactions on Computational Biology and Bioinformatics*. 2025; 22(6): 3584–3590. <https://doi.org/10.1109/TCBBIO.2025.3634738> → Section 5.1
- Singh P, Lonardi S, Liang Q, Vydyam P, Khabirova E, Fang T, Gihaz S, Thekkiniath J, Munshi M, Abel S, Ciampossin L, Batugedara G, Gupta M, Lu X M, Lenz T, **Chakravarty S**, Cornillot E, Hu Y, Ma W, Gonzalez L M, Sánchez S, Estrada K, Sánchez-Flores A, Montero E, Harb O S, Le Roch K G, Mamoun C B. “*Babesia duncani* multi-omics identifies virulence factors and drug targets”. *Nature Microbiology*. 2023; 8(5): 845–859. <https://doi.org/10.1038/s41564-023-01360-8> → Section 5.2
- Singh P, Vydyam P, Fang T, Estrada K, Gonzalez L M, Grande R, Kumar M, **Chakravarty S**, Berry V, Ranwez V, Carcy B, Depoix D, Sánchez S, Cornillot E, Abel S, Ciampossin L, Lenz T, Harb O, Sánchez-Flores A, Montero E, Le Roch K G, Lonardi S, Mamoun C B. “Insights into the evolution, virulence and speciation of *Babesia MOI* and *Babesia divergens* through multiomics analyses”. *Emerging Microbes & Infections*. 2024; 13(1). <https://doi.org/10.1080/22221751.2024.2386136> → Section 5.2

Chapter 2

RAmbler: *De Novo* Genome Assembly of Complex Repetitive Regions

2.1 Introduction

Repetitive regions (or segmental duplications) are the primary reasons for which *de novo* genome assemblies are often fragmented and incomplete. A large fraction of eukaryotic genomes is made of repetitive elements, including satellite DNA, mini-satellites, micro-satellites, and DNA/RNA transposons [91, 64, 126, 14]. For instance, Figure 2.1 illustrates the frequency of repeats present in seven plant species [24]. Several studies have shown that expansions or mutations of repetitive regions are linked to a variety of human diseases, ranging from neurological diseases to cancers (see [78] for a review). Although many repeats were considered non-functional, they have been shown to impact gene expression, contributing to genetic disorders [49, 57, 115].

The problem of reconstructing repetitive regions (segmental duplications) has been ad-

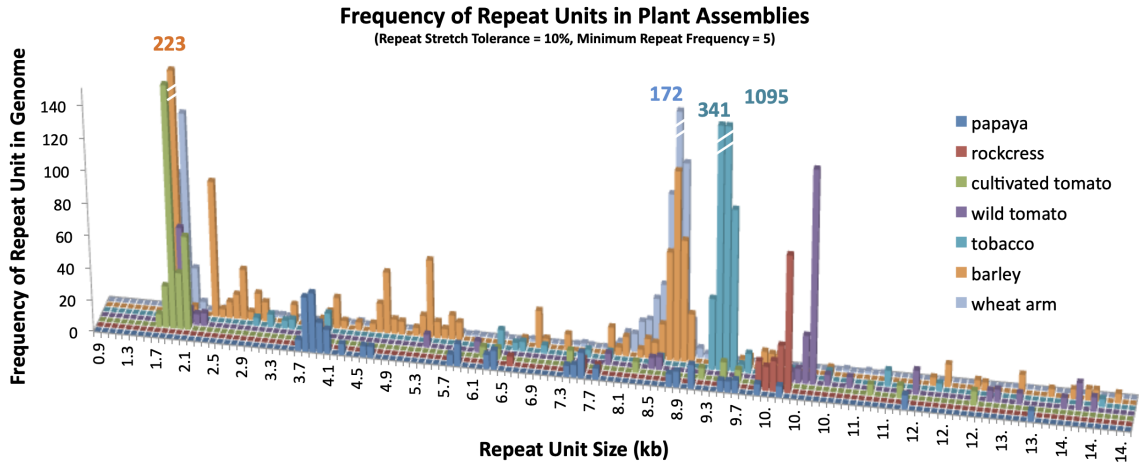


Figure 2.1: A frequency histogram of unresolved repeats in seven plant genomes (2015); the x -axis represents the repeat size (in kb); the y -axis represents the number of copies in each genome of those repeats' sizes (Figure reproduced with permission [24])

addressed several times in the literature, e.g., [19, 128, 16]. The Segmental Duplication Assembler (SDA) by [128] is no longer maintained. More recently, [16] proposed CentroFlye to address the problem of reconstructing human centromeres. CentroFlye is a specialized assembler that uses error-prone ONT or PacBio CLR reads. It also requires additional information such as higher-order repeats (HORs) or monomers. This limits its applicability to species for which HORs or monomers are known. In addition, CentroFlye is very demanding in terms of computational resources. While SDA, CentroFlye and the assembler for human Chromosome 8 by [79] are specialized tools for reconstructing segmental duplications or centromeres, they were designed for error-prone long reads. To the best of our knowledge, there is no specialized assembler for reconstructing repeats that uses PacBio HiFi reads exclusively.

2.1.1 Contributions

In this chapter, (1) we carry out the first statistical analysis on the method to extract single copy k -mers for PacBio HiFi and Oxford Nanopore reads; (2) we develop a new assembly method, called RAMbler, that takes advantage of single copy k -mers to resolve complex repetitive regions; (3) we compare RAMbler against state-of-the-art assemblers (hifiasm [26], LJA [8], HiCANU [96] and Verkko [105]) on more than 250 synthetic data sets; (4) we introduce a new quality metric, called assembly score, for genome assemblies that incorporates completeness, contiguity and number of mis-assemblies in one consolidated measure.

2.2 Methods

2.2.1 Problem Formulation

We assume that (1) the genome G contains n repetitive regions $\{R_1, \dots, R_i, \dots, R_n\}$, (2) each repetitive region R_i is composed of t_i tandem copies of a string α_i , (3) each tandem copy has sufficient variations that allows it to be distinguished from another copy; we assume that each copy contains SNPs with probability p (e.g., $p = 1/100$), and its length can increase or decrease by at most $L\%$. Given a set T of HiFi reads and the draft genome G , the objective is to produce a set $\{F_1, \dots, F_i, \dots, F_n\}$ of n assemblies, where each F_i is as “similar as possible” to R_i . In particular, if the assembly F_i contains t'_i copies of the repeat unit, we want t'_i as close as t_i as possible (see Figure 2.2). For synthetic data, we measure the quality of the assembly F_i by comparing it to R_i using QUAST [48], i.e., we report the fraction of R_i covered by F_i (ideally 100%), the number of mis-assembled contigs in F_i (ideally zero), and the number of contigs in F_i (ideally

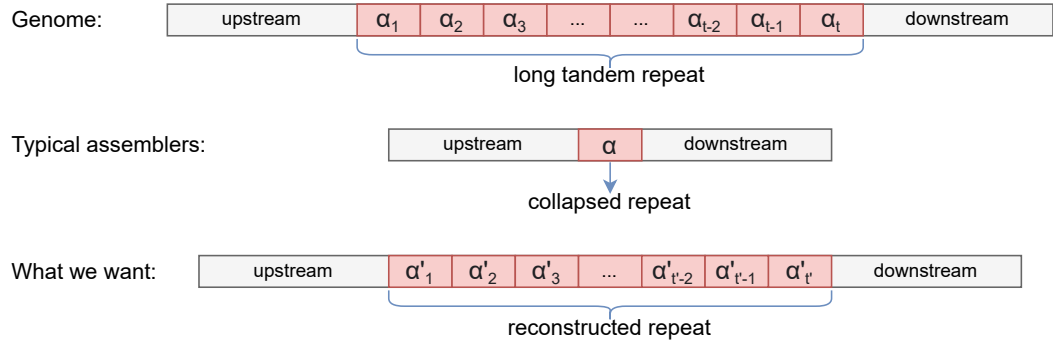


Figure 2.2: The problem formulation: tandem repeats $\alpha_1, \alpha_2, \dots, \alpha_t$ are over-collapsed by traditional assemblers; we want to reconstruct all the copies of the repetitive regions

one). When the ground truth is unavailable (i.e., for real data sets), a qualitative assessment of the assembly's accuracy can be obtained by aligning F_i with the corresponding repeat unit α_i . The alignment, visualized as a dot plot, can provide a qualitative measure on how well the repeat units are assembled.

2.2.2 Repeat Identification

To identify the repetitive regions $\{R_1, \dots, R_n\}$, we map the HiFi reads T against the draft genome assembly G . Since unresolved tandem repeats are collapsed in the draft assembly, they can be identified by a spike in mapping coverage. For instance, Figure 2.3 shows the mapping coverage of an unresolved tandem repeat in Chromosome XII of *Saccharomyces cerevisiae* which is known to contain ~ 150 tandemly repeated copies of a 9.1 kb rDNA unit [63, 68]. This region is the only unresolved non-telomeric gap in the current *S. cerevisiae* assembly. Table 2.1 summarizes the data sets for *S. cerevisiae* used in this study.

Organism	Reference Genome	Illumina (NGS)		Oxford Nanopore (ONT)		PacBio HiFi (HiFi)	
		Accession/Link	Coverage	Accession/Link	Coverage	Accession/Link	Coverage
Yeast (<i>S. cerevisiae</i>)	R64 (S288C)	-	-	SRR15597407	~140×	SRR13577847	~85.5×
				SRR18365585	~50×	SRR18210286	~188×

Table 2.1: Data sets for *S. cerevisiae*

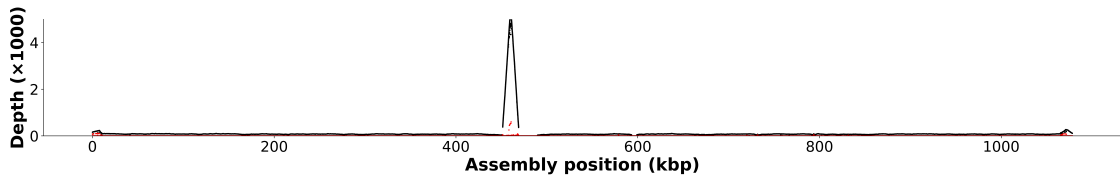


Figure 2.3: PacBio HiFi mapping coverage for Chromosome XII in *Saccharomyces cerevisiae* illustrated using NucFreq; the coverage spike indicates the presence of a repetitive region which is known to contain ~150 tandemly repeated copies of a 9.1 kb rDNA unit

2.2.3 Analysis of k -mer Distribution

Recall that we assume that the copies in the repetitive region are not identical to each other. If all the copies were identical, the problem of resolving repeats would be impossible, unless one can produce reads so long that they span the entire repetitive region. We rely on the presence of the SNPs to distinguish and partition the HiFi reads that belong to different repeat copies within a repetitive region. When distinct SNPs are present among the different copies, we expect those copies to have their own SNP signatures.

Each SNP is likely to induce a unique (or single-copy) k -mer, i.e., a k -mer that occurs a number of times approximately equal to the expected sequencing coverage. We call these k -mers,

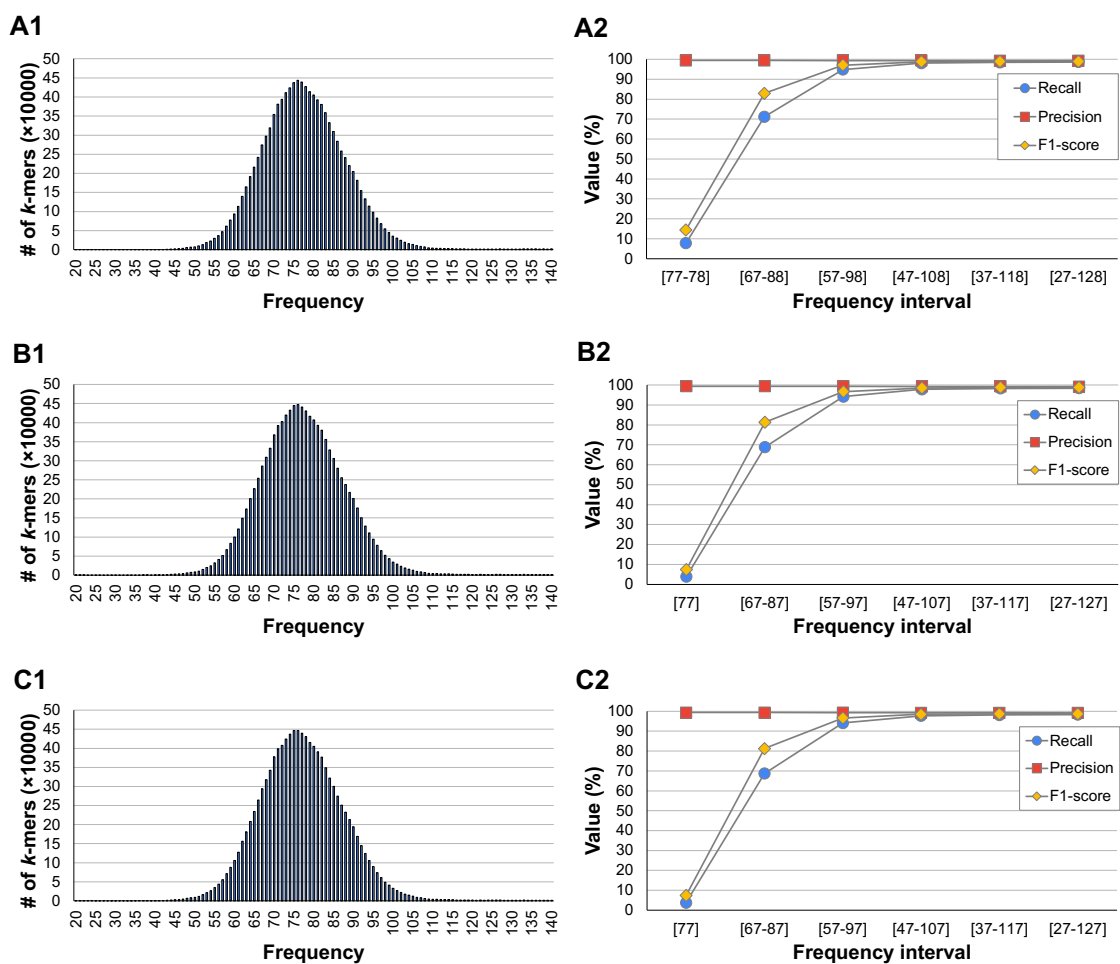


Figure 2.4: PacBio HiFi k -mer distribution for $k = 17$ (A1), $k = 21$ (B1) and $k = 25$ (C1) for *Saccharomyces cerevisiae*; precision, recall, and F1-score for unikmers when $k = 17$ (A2), $k = 21$ (B2) and $k = 25$ (C2) for longer and longer intervals centered at the average sequencing coverage

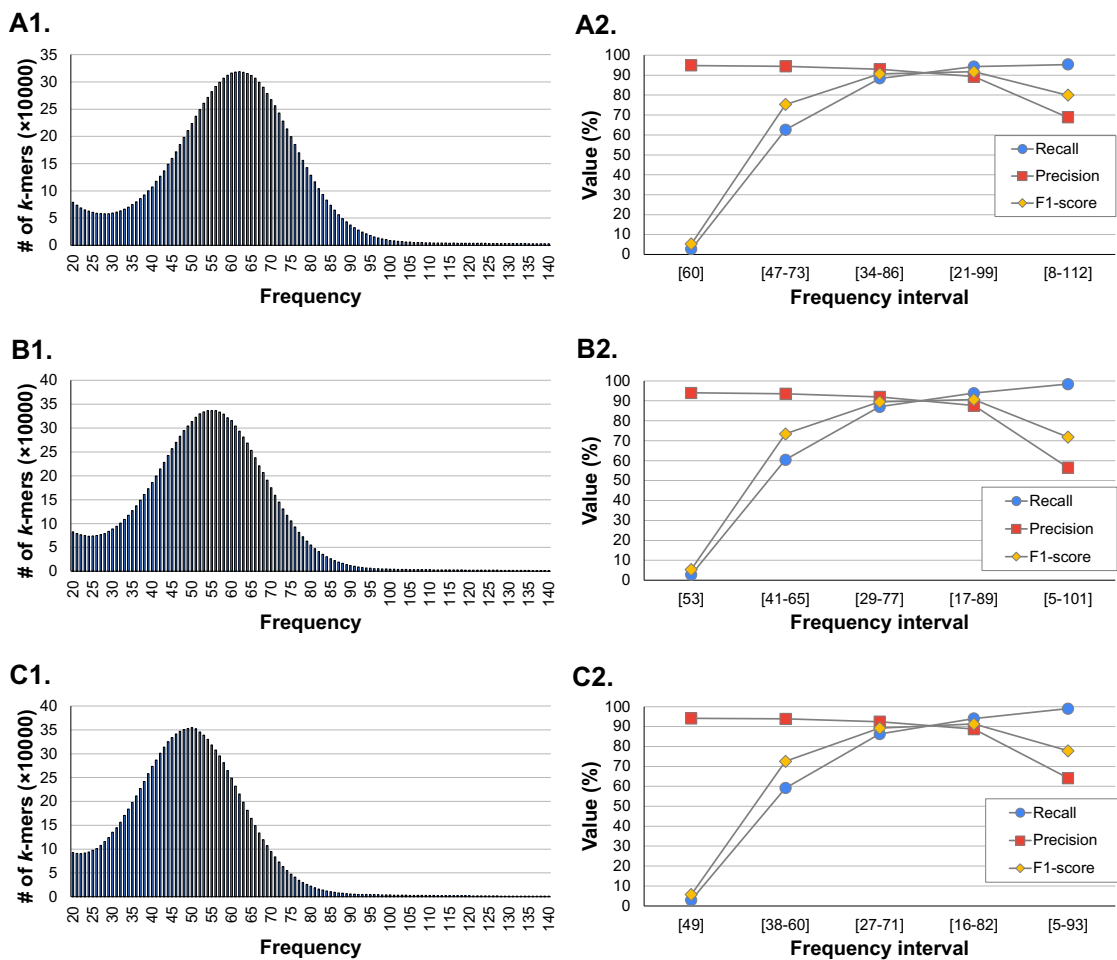


Figure 2.5: Oxford Nanopore k -mer distribution for $k = 17$ (A1), $k = 21$ (B1) and $k = 25$ (C1) for *Saccharomyces cerevisiae* dataset (SRA accession SRR18365585, $\sim 50\times$ coverage); precision, recall, and F1-score for unikmers when $k = 17$ (A2), $k = 21$ (B2) and $k = 25$ (C2) for larger and larger intervals centered at the average sequencing coverage

unikmers. Unikmers (called SUNKs in [79] or SUNs in [122]) were crucial to resolve the assembly of human Chromosome 8, but to the best of our knowledge they have not been used in any other assembly method. Please note that *unikmers* are NOT *k*-mers that appear only once in the reads: those *k*-mers correspond to sequencing errors.

One of the contributions of our study is to provide a method to identify *unikmers* from the reads, and analyze its accuracy and precision. RAmbler finds *unikmers* by selecting all *k*-mers in the HiFi reads that have a number of occurrences within the interval $[\mu - t\sigma, \mu + t\sigma]$, where μ is the average sequencing depth of the HiFi reads, σ is the standard deviation of the sequencing depth, and k and t are user-defined parameters.

We investigate how to choose k and t in the following analysis. (1) We determine the set of true *unikmers* in the *Saccharomyces cerevisiae* genome to serve as the ground truth. (2) We compute the *k*-mer distribution for a set of real HiFi reads (SRA accession SRR13577847) and ONT reads (SRA accession SRR18365585) for *S. cerevisiae* using Jellyfish [85]. The left column of Figure 2.4 shows the *k*-mer distribution for HiFi reads for $k = 17$ (Figure 2.4-A1), $k = 21$ (Figure 2.4-B1) and $k = 25$ (Figure 2.4-C1). Odd integers in the range $[17, 25]$ are typical choices for k to estimate genome size (see, e.g., [129, 104]) or the construction of the de Bruijn graphs for eukaryotic genomes (see, e.g., [139]). Observe that the distributions are almost identical, which indicate that any of these *k*-mer choices would be appropriate. (3) We compute the average sequencing depth μ and the standard deviation of the sequencing depth σ from the *k*-mer distribution. (4) The *k*-mers in the HiFi reads that have a number of occurrences within the interval $[\mu - t\sigma, \mu + t\sigma]$ for $t = 0, 1, 2, 3, 4, 5$ are compared against the true *unikmers*: true positive, false positive, true negative and false negative are recorded.

The results of this analysis (precision, recall and F1-score) for HiFi reads are shown in the right column of Figure 2.4. The x-axis represents the choice of t , i.e., longer and longer intervals centered around the mean (the first interval is for $t = 0$, the second is for $t = 1$, etc.). Figure 2.4-A2 shows the results for $k = 17$, Figure 2.4-B2 illustrates the results for $k = 21$ and Figure 2.4-C2 shows the results for $k = 25$. Observe that in all cases precision and recall are very close to 100% as soon as $t = 3$. For instance, there are 11,137,337 21-mers that occur $[47 - 107]$ times in the HiFi reads, i.e., at most $t = 3$ standard deviations away from the average coverage. Of those, 11,058,290 are truly *unikmers* which correspond to a precision of 99.29%; only 79,047 are false positives (0.71% of the total). For $t = 3$, this method recalls 97.84% of the *unikmers* in the genome. Almost identical results can be obtained from $k = 17$ or $k = 25$. This analysis indicates that selecting k -mers that have a number of occurrences in the interval $\mu \pm 3\sigma$ in HiFi reads can recover almost 98% of the true single-copy k -mers in the genome with a false positive rate less than 1%. The same analyses carried out on ONT reads show that precision, recall, and F1-score for ONT reads are slightly lower than those obtained from PacBio HiFi reads, likely due to the higher rate of sequencing errors in ONT reads (see Figure 2.5).

Based on this analysis, we used $k = 21$ and $t = 3$ for all the experiments (unless otherwise noted).

2.2.4 RAMbler's Algorithm

A. Determine the reads corresponding to repetitive regions. As mentioned in Section 2.2.2, RAMbler identifies repetitive regions by mapping all HiFi reads T against the draft genome. RAMbler generates the plot of the read coverage across the genome using NucFreq [128]. Unresolved

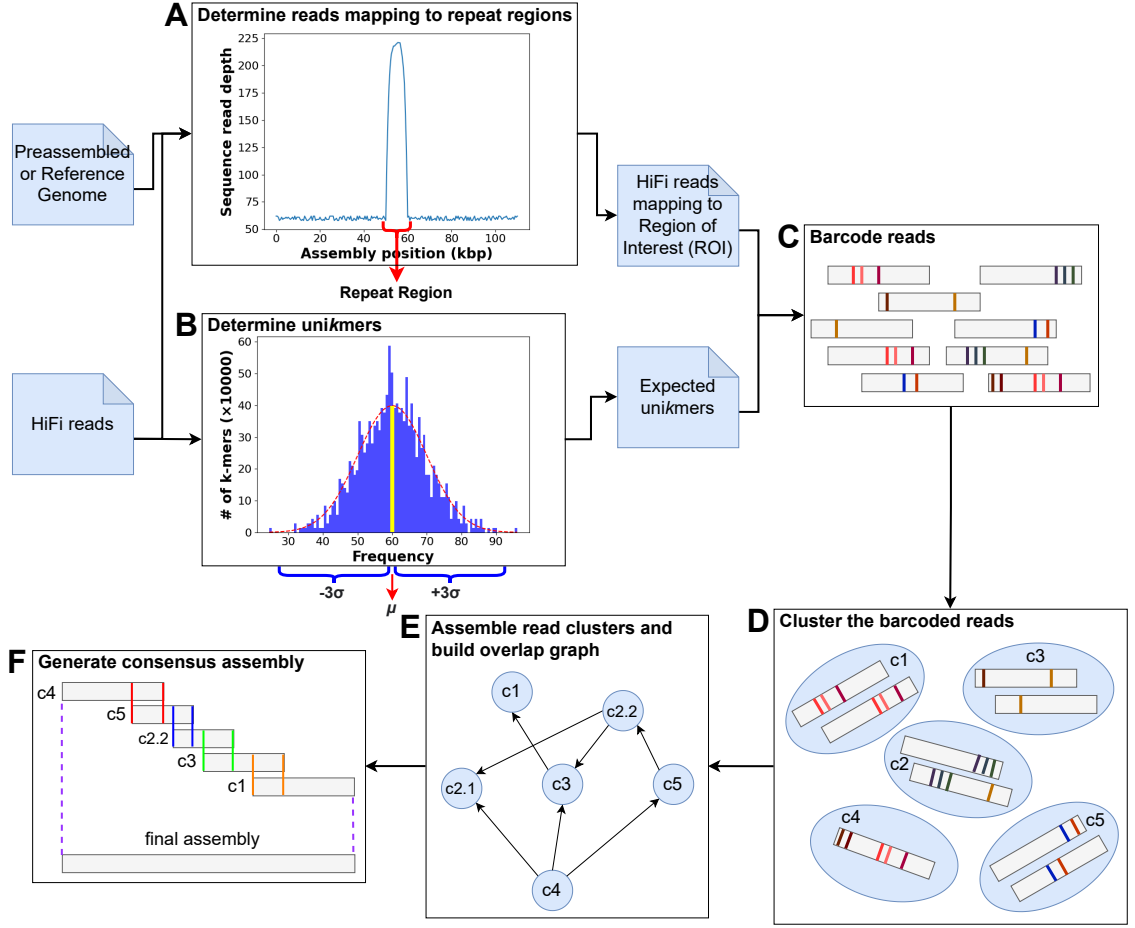


Figure 2.6: The algorithmic pipeline used in RAMbler

repetitive regions produce distinctive peaks in the coverage plot as illustrated in Figure 2.3. Then, RAMbler selects the reads that map to the repetitive regions, as well as the reads extending 50 kb upstream and downstream of the peak, as shown in Figure 2.6, step A. We call the set of HiFi reads extracted in this step T_r , where r identifies the repetitive region.

B. Determine unikmers. RAMbler uses Jellyfish on the entire set of HiFi reads T to obtain the count distribution of 21-mers genome-wide. RAMbler calculates the mean μ and standard deviation σ of the distribution by excluding 21-mers that appear less than 5 times since these are most likely

induced by sequencing errors. RAMbler then selects the 21-mers that fall within the interval $[\mu - 3\sigma, \mu + 3\sigma]$. According to our analysis in Section 2.2.3, these 21-mers are true *unikmers* with high probability (Figure 2.6, step B).

C. Barcode reads. RAMbler uses the set of *unikmers* to barcode the HiFi reads T_r (Figure 2.6, step C). RAMbler searches for exact occurrences of the *unikmers* in the reads T_r or their reverse complement. The set of *unikmers* present in a read and their location is the *barcode* of that read. For each read, RAMbler stores pairs (u, j) , where u is a *unikmer* and j is the location within the read.

D. Cluster the barcoded reads. RAMbler compares the barcode of all pairs of reads to identify shared *unikmers*. This pairwise comparison allows RAMbler to determine which reads are overlapping. Two reads are overlapping if they share at least th *unikmers*, and the set of relative distances between the shared *unikmers* match within a tolerance up to to base pairs. Overlaps are stored in the *barcode graph*: each node in the barcode graph represents a read; nodes in the graph are connected by an edge if the corresponding reads are overlapping according to the criteria described above. Once the graph is completed, RAMbler identifies clusters of reads by determining the connected components of the barcode graph (Figure 2.6, step D).

E. Assemble read clusters and build overlap graph. RAMbler carries out individual local assemblies for each set of clustered reads using a standard HiFi assembler (hifiasm in this case). Each read cluster is assembled in one or more contigs. RAMbler then uses minimap2 [75] to align assembled contigs to each other. Any contig that is fully contained within another longer contig is removed. RAMbler constructs an overlap graph based on the overlap information provided by min-

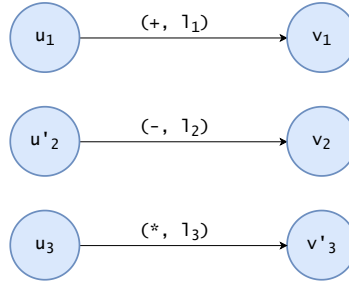


Figure 2.7: The three types of edges in our overlap graph

imap2: each node in the overlap graph represents a contig; nodes are connected by edges if they have a suffix-prefix overlap (Figure 2.6, step E). It is worth noting there could be multiple suffix-prefix overlaps between a pair of contigs. RAMbler retains the overlap with the highest percentage of identity as long as the overlap is at least $mo = 1000$ bps. Furthermore, these suffix-prefix overlaps can occur between the positive or negative strands, resulting in three types of edges. Each edge is labeled by a pair (t, l) , where $t \in \{+, -, *\}$, and l is the length of the suffix-prefix overlap (Figure 2.7). Given an edge (u, v) in the overlap graph, its type t is (1) “+” when there is an overlap between contig u and contig v , (2) “−” when there is an overlap between the reverse complement of contig u and contig v , (3) “*” when there is an overlap between contig u and the reverse complement of contig v .

F. Generate consensus assembly. At this stage, RAMbler needs an estimate of the size of the repetitive region. The size is obtained by computing the ratio between the total number of bases in the reads T_r (extracted in step A) and the average coverage depth.

To compute the final assembly, RAMbler enumerates all possible paths using DFS and generates a set of candidate assemblies. When computing the sequence consensus for suffix-prefix

Parameter	Value	Parameter	Value
Mean read length	13,500	Mean read accuracy	0.998
Standard deviation of read length	1,500	Standard deviation of read accuracy	0.01
Minimum read length	7,500	Minimum read accuracy	0.95
Maximum read length	20,000	Maximum read accuracy	1.0

Table 2.2: PBSim’s parameters and their values

overlaps, if the suffix and the prefix do not match, RAMbler arbitrarily picks the base either from the suffix or the prefix (Figure 2.6, step F). Among all assemblies, RAMbler selects the one that best matches the estimated length of the repetitive region.

A summary of RAMbler’s main parameters k , t , th , to , and mo with their default values is shown in Table 2.3. The optimization of these parameters is discussed in Sections 2.2.3 and 2.3.3.

2.3 Experimental Setup

2.3.1 Synthetic Data Generation

We generated synthetic repetitive regions by selecting a combination of (1) repeat unit size: 10 kb, 15 kb, 20 kb and 25 kb; (2) 2, 5, and 10 copies of the repeat unit; (3) mutation rate in each copy of the repeat $p = \{1/100, 1/250, 1/500, 1/1000, 1/2000\}$. For each combination, we generated HiFi reads using PBSim [97] on the CCS model with read coverage of $10\times$, $20\times$, $30\times$, and $40\times$. The other parameters required to be set before generating reads are provided in Table 2.2.

2.3.2 Performance Metrics

Metrics such as NG50, genome fraction, number of contigs, and number of mis-assemblies have been traditionally employed to evaluate the quality of an assembly on synthetic data when the reference genome is known. However, each of these metrics alone does not fully capture all the desired qualities of a “good assembly”. To address this shortcoming we introduce here a new metric called the *assembly score* that summarizes in one number the quality of an assembly in terms of accuracy, contiguity, and completeness. The assembly score is based on two preliminary metrics, as explained next.

A. Effective genome fraction per contig (ζ). Consider an assembly that consists of a single contig but contains one mis-assembly. To correct the mis-assembly, the contig needs to be broken, resulting in the creation of an additional contig. Based on this observation we define the *effective number of contigs* as the sum of the number of contigs in the assembly and the number of mis-assemblies. We propose to calculate the effective genome fraction per contig as follows

$$\zeta = \frac{\text{genome_fraction}}{\text{\#contigs} + \text{\#mis-assembled contigs}} \quad (2.1)$$

As said, metric ζ takes into account the number of mis-assemblies and penalizes the score accordingly. An assembly that covers 100% of the reference genome (without any mis-assemblies) would yield $\zeta = 1.0$. By considering the effective genome fraction per contig, we can assess the assembly quality while accounting for the presence of mis-assemblies, thereby providing a more comprehensive evaluation.

B. Normalized NG50 (η). While NG50 is an essential metric for evaluating the contiguity of an assembly, it depends on the size of the reference genome, making it challenging to use it to compare an assembler’s performance across genomes of different lengths. To address this limitation, we normalize NG50 by the size of the reference genome, yielding a metric called η defined as follows

$$\eta = \frac{\text{NG50}}{|\text{reference_genome}|} \quad (2.2)$$

By normalizing NG50 with respect to the reference genome size, η is constrained within the range of $[0, 1]$, with a perfect assembly achieving $\eta = 1$. Observe that it is possible that η may exceed 1 when the assembly is over-inflated, i.e., longer than the actual genome. In general, a higher value of η indicates a better assembly quality, as long as it is smaller than 1.

C. Assembly score. The assembly score is defined by computing the harmonic mean of ζ and η ,

$$\text{assembly_score} = \frac{2\zeta\eta}{\zeta + \eta} \quad (2.3)$$

Observe that while ζ is always within $[0, 1]$, η can exceed 1, which can result in an assembly score greater than 1. Nevertheless an assembly score closer to 1 indicates a higher quality assembly. This score enables a holistic assessment of the assembly’s quality, taking into account accuracy, contiguity, and completeness, which are all equally important.

2.3.3 Parameter Optimization

RAmbler has five main parameters (summarized in Table 2.3). Based on the analysis in Section 2.2.3, we determined that $k = 21$ and $t = 3$. To find the optimal values for to and th , we conducted a grid search where $to = \{1, 5, 10, 15, 20\}$ and $th = \{5, 10, 15, \dots, 50\}$ (50 combinations). Parameter mo (minimum overlap) was set to 1,000 bp.

	Parameter	Value
k	length of <i>unikmers</i>	21
t	k -mers in the interval $[\mu - t\sigma, \mu + t\sigma]$ of the k -mer distribution considered <i>unikmers</i>	3
th	minimum number of <i>unikmers</i> shared between two reads to declare them overlapping	15
to	maximum allowed tolerance between the position of two consecutive shared <i>unikmers</i>	15
mo	minimum overlap between contigs that is allowed in the overlap graph	1,000 bp

Table 2.3: RAMbler’s main parameters

RAMbler was tested on 135 synthetic data sets, obtained from the combinations of different choices of the repeat unit size $\{10, 15, 20\}$ kb, repeat copies $\{2, 5, 10\}$, mutation rate $p = \{1/100, 1/250, 1/500, 1/1000, 1/2000\}$, and read coverage depth $\{20\times, 30\times, 40\times\}$. Figure 2.8 shows the experimental results for different metrics (namely number of contigs, number of misassembled contigs, effective genome fraction per contig and normalized NG50), with the best choices for to and th highlighted in colored rectangles.

When considering the number of contigs (Figure 2.8A), observe that RAMbler achieved the best performance for $to = \{15, 20\}$ and $th = \{10, 15\}$. Regarding the number of misassembled contigs produced by RAMbler (Figure 2.8B), the best values were $to = \{15, 20\}$ and $th = \{10, 15, 20, 25\}$. In terms of effective genome fraction per contig (ζ) (Figure 2.8C), RAMbler had better results with $to = \{10, 15, 20\}$ and $th = \{10, 15, 20\}$. When considering the normalized NG50 (η) (Figure 2.8D), the best outcomes were obtained with $to = \{15, 20\}$ and $th = 15$. By combining all these metrics in the assembly score (defined in Section 2.3.2.C), we determined that the optimal values were $to = 15$ and $th = 15$ (Figure 2.9).

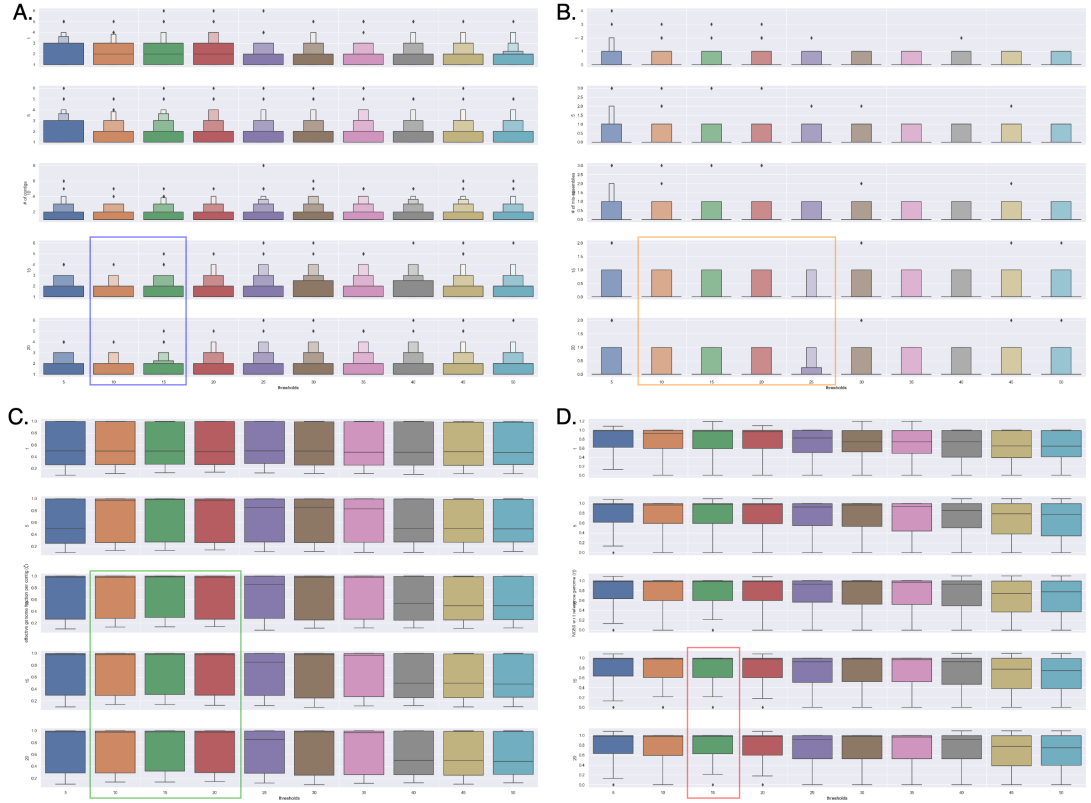


Figure 2.8: Performance of RAMbler for difference combinations of tolerance $to = \{1, 5, 10, 15, 20\}$, threshold, $th = \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$, repeat size = $\{10, 15, 20\}$ kb, number of copies = $\{2, 5, 10\}$, mutation rate $p = \{\frac{1}{100}, \frac{1}{250}, \frac{1}{500}, \frac{1}{1000}, \frac{1}{2000}\}$, on synthetic reads with coverage depth = $\{20\times, 30\times, 40\times\}$; the rows correspond to different values of to , the columns correspond to different values of th ; (A) number of contigs; the best performance is achieved for $to = \{15, 20\}$ and $th = \{10, 15\}$, (B) number of misassembled contigs; the best performance is achieved for $to = \{15, 20\}$ and $th = \{10, 15, 20, 25\}$, (C) effective genome fraction per contig ζ ; the best performance is achieved for $to = \{10, 15, 20\}$ and $th = \{10, 15, 20\}$, and (D) normalized NG50 η ; the best performance is achieved for $to = \{15, 20\}$ and $th = 15$

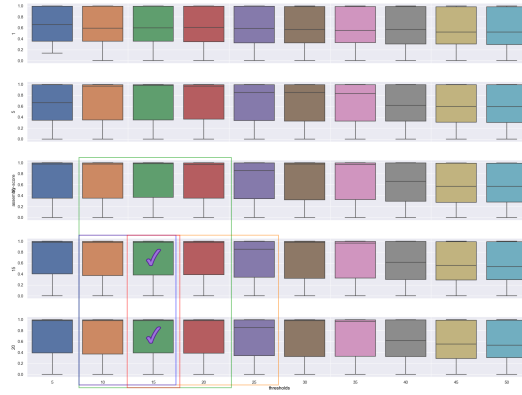


Figure 2.9: Assembly scores achieved by RAMbler for different combinations of tolerance $to = \{1, 5, 10, 15, 20\}$, threshold $th = \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$, repeat size = $\{10, 15, 20\}$ kb, number of copies = $\{2, 5, 10\}$, mutation rate $p = \{\frac{1}{100}, \frac{1}{250}, \frac{1}{500}, \frac{1}{1000}, \frac{1}{2000}\}$, on synthetic reads with coverage depth = $\{20\times, 30\times, 40\times\}$; the rows correspond to different values of to , the columns correspond to different values of th ; the optimal values are indicated by tick marks ($to = \{15, 20\}$ and $th = 15$)

2.4 Result Analysis

2.4.1 Comparing RAMbler, hifiasm, LJA, HiCANU, and Verkko on fixed-length repeats

We conducted an extensive performance comparison of RAMbler with other state-of-the-art HiFi assemblers, namely hifiasm, LJA, HiCANU, and Verkko, using 36 synthetic data sets. SDA is no longer maintained and CentroFlye requires additional auxiliary information (HORs and monomers), thus they were both excluded from this comparison. We generated repetitive regions with two choices of the repeat unit size $\{15, 20\}$ kb, two choices for the number of copies $\{5, 10\}$, and three values of mutation rate $p = \{1/250, 1/500, 1/1000\}$. Synthetic HiFi reads were generated using PBSim with coverage depths of $\{20\times, 30\times, 40\times\}$ (see Section 2.3.1 for details).

Figure 2.10 summarizes the results in terms of four different performance metrics. In Figure 2.10A, we compare the number of contigs produced by the different assemblers, where a single contig would be the ideal assembly. Observe that RAMbler consistently produced the lowest number of contigs among all the assemblers. Figures 2.10B, C, and D show the results in terms of effective genome fraction per contig (ζ), normalized NG50 (η), and assembly score (numerical scores are in Table 2.4), respectively. For all these metrics, the best assembly is the one that gets closer to 1.0, and RAMbler achieved higher values on all of them compared to the other assemblers.

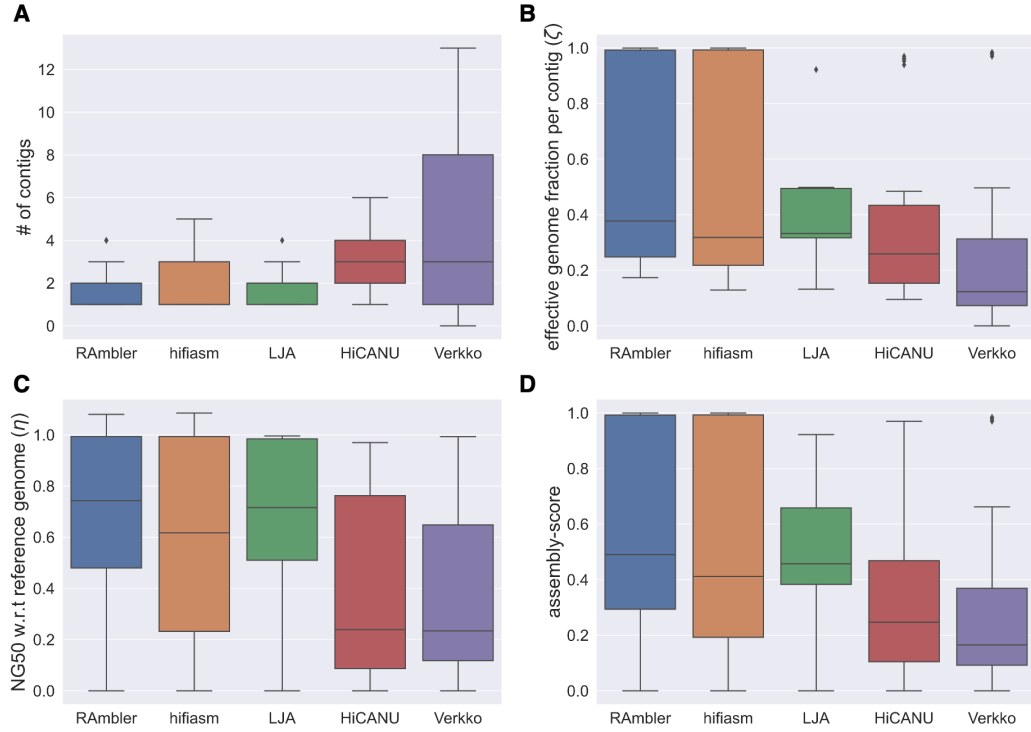


Figure 2.10: Assembly statistics for RAMbler, hifiasm, LJA, HiCANU, and Verkko for 36 different combinations of synthetic data with repetitive regions having repeat sizes $\{15, 20\}$ kb, number of copies $\{5, 10\}$, mutation rate $p = \{1/250, 1/500, 1/1000\}$ and coverage depth $\{20\times, 30\times, 40\times\}$; (A) number of contigs, (B) effective genome fraction per contig ζ , (C) normalized NG50 η , and (D) assembly-score

Dataset				Assembly score				
Repeat unit size	# of copies	p	Coverage	RAmbler	hifiasm	LJA	HiCANU	Verkko
15 kb	5	1/250	20	0	0.992421	0.362516	0	0.103149
			30	0.993685	0.993682	0.66135	0.279093	0.336791
			40	0.992072	0.996525	0.664114	0.596943	0.472819
		1/500	20	0.442281	0.986855	0.64146	0.236397	0.273222
			30	0.680773	0.548473	0.658083	0.388264	0
			40	0.999324	0.684225	0.661653	0.960054	0.982176
		1/1000	20	0.979912	0.996561	0	0	0.141093
			30	0.999289	0.999281	0.921949	0.402803	0.170338
			40	0.996749	0.996706	0.404907	0.57493	0.212441
	10	1/250	20	0.378988	0.085941	0.213113	0.164032	0.111124
			30	0.381808	0.194383	0.326858	0.566358	0.464188
			40	0.697521	0.164275	0.651712	0.436001	0.645386
		1/500	20	0.503772	0.18682	0	0	0.092472
			30	0.397242	0.214671	0.655464	0.13711	0.170362
			40	0.208417	0.290759	0.64724	0.645192	0
		1/1000	20	0.27603	0.325317	0.395507	0.117697	0.105373
			30	0.246113	0.42346	0.445131	0.166698	0.264898
			40	0.23118	0.423723	0.434755	0.969895	0.984059
20 kb	5	1/250	20	0.993377	0.993362	0.613642	0	0.158968
			30	0.995045	0.995115	0.651152	0.143198	0.320305
			40	0.995057	0.995112	0.414397	0.939415	0
		1/500	20	0.989612	0.99032	0.290029	0	0.228378
			30	0.99039	0.310043	0.406863	0.257214	0.551989
			40	0.998585	0.998585	0.659891	0.962357	0
		1/1000	20	0.994817	0.596527	0.211835	0	0.125832
			30	0.61509	0.255791	0.662906	0.228406	0
			40	0.265227	0.305306	0.661373	0.411638	0.97693
	10	1/250	20	0.321816	0.16656	0	0	0.218633
			30	0.475847	0.399262	0.658377	0.373103	0.139977
			40	0.73109	0.86521	0.457416	0.264454	0.661972
		1/500	20	0.362629	0.077912	0.258614	0	0.090594
			30	0.299438	0.310596	0.414516	0.110915	0.146964
			40	0.336683	0	0.660191	0.374036	0.970193
		1/1000	20	0.187074	0.177739	0.389218	0.087065	0
			30	0.270521	0	0.456605	0.116092	0
			40	0.194154	0	0.559774	0.95271	0

Table 2.4: Assembly score of RAMbler, hifiasm, LJA, HiCANU, and Verkko on 36 synthetic data sets; numbers in bold indicate the best assembly score on each row

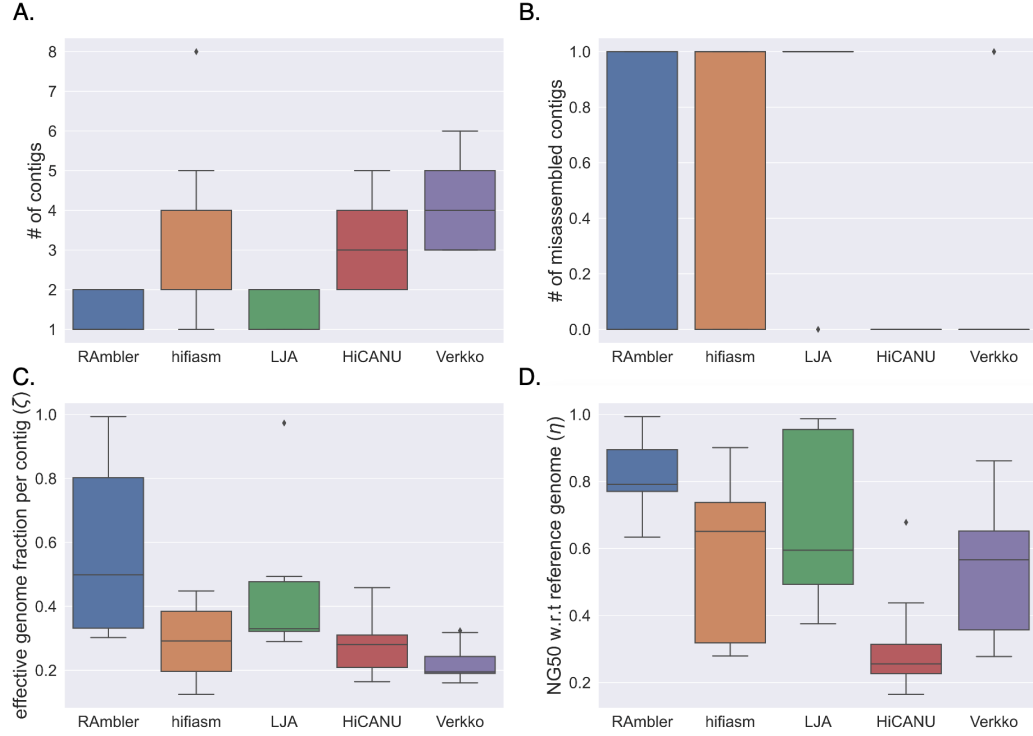


Figure 2.11: Assembly statistics for RAMbler, hifiasm, LJA, HiCANU and Verkko on synthetic data with $30\times$ coverage depth for nine different combinations of repetitive regions containing five copies of a repeat size = $\{15, 20, 25\}$ kb and variable-length (up to $\pm 5\%$ per copy), mutation rate $p = \{\frac{1}{250}, \frac{1}{500}, \frac{1}{1000}\}$; (A) number of contigs, (B) number of misassembled contigs, (C) effective genome fraction per contig ζ , and (D) normalized NG50 η

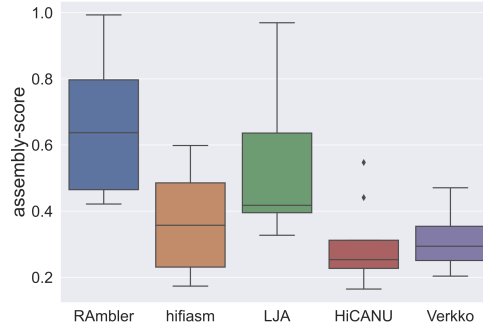


Figure 2.12: Assembly score for RAmbler, hifiasm, LJA, HiCANU and Verkko on synthetic data with $30\times$ coverage depth for nine different combinations of repetitive regions containing five copies of a repeat size = $\{15, 20, 25\}$ kb and variable-length (up to $\pm 5\%$ per copy), mutation rate $p = \{\frac{1}{250}, \frac{1}{500}, \frac{1}{1000}\}$

2.4.2 Comparing RAmbler, hifiasm, LJA, HiCANU, and Verkko on variable-length repeats

To evaluate RAmbler’s ability to resolve tandem repeats in the presence of variable-length repeat units, we created synthetic repetitive regions in which each repeat copy can vary up to $\pm 5\%$ of the length of a repeat unit. We used five copies of repeat units of $\{15, 20, 25\}$ kb with mutation rates $p = \{1/250, 1/500, 1/1000\}$. Synthetic HiFi reads with a coverage depth of $30\times$ were generated with PBSim. Table 2.5 summarizes assembly score results for these nine data sets. RAmbler outperformed the other assemblers in six out of nine runs. Figure 2.11 shows that RAmbler, hifiasm and LJA produce contigs with either zero or one misassembled contig. It also shows that RAmbler, hifiasm and LJA produce more contiguous assemblies than HiCANU and Verkko. In general, while HiCANU and Verkko rarely introduce misassemblies, they produce more fragmented assem-

Synthetic data set			Assembly score				
Repeat unit size	Number of copies	p	RAmbler	hifiasm	LJA	HiCANU	Verkko
15 kb	5	1/250	0.493	0.357	0.327	0.239	0.251
		1/500	0.993	0.545	0.658	0.165	0.204
		1/1000	0.454	0.426	0.346	0.547	0.408
20 kb		1/250	0.422	0.599	0.418	0.441	0.276
		1/500	0.797	0.309	0.618	0.312	0.246
		1/1000	0.896	0.213	0.395	0.206	0.294
25 kb		1/250	0.637	0.486	0.636	0.227	0.354
		1/500	0.773	0.173	0.970	0.254	0.471
		1/1000	0.465	0.231	0.409	0.278	0.345

Table 2.5: Assembly score results for nine different data sets of synthetic HiFi reads with coverage $30\times$, based on a repetitive region with a variable-length (up to $\pm 5\%$ per copy) repeat unit of length $\{15, 20, 25\}$ kb, and $p = \{1/250, 1/500, 1/1000\}$; numbers in bolt indicate the best assembly score on each row

blies than RAmbler, hifiasm and LJA. This is reflected by the RAmbler's best assembly score in Figure 2.12.

2.4.3 Comparing RAMbler, hifiasm, LJA, HiCANU, and Verkko on repetitive regions with copy number variation

Note that RAMbler was not designed to produce a haplotype-resolved assembly. An important question is what assembly would RAMbler produce in case there are copy number variations between the two haplotypes. To address this question we carried out several experiments on synthetic diploid genomes, as follows. (i) We created a synthetic repetitive region where the primary haplotype (hap1) contained either 5 or 10 copies of a repeat unit. Each repeat unit was {5, 10, 15, 20, 50, 100} kb long, with a mutation probability of {1/250, 1/500, 1/1000}. (ii) We produced the secondary haplotype (hap2) as follows. When the hap1 had 5 copies, hap2 had {3, 4, 5} copies. If hap1 had 10 copies, hap2 had {6, 8, 10} copies. (iii) We added a 50 kb sequence upstream and downstream of the repetitive region on both hap1 and hap2. (iv) We used PBSim to generate $30\times$ -coverage HiFi reads from these $2 \times 6 \times 3 \times 3 = 108$ synthetic diploid genomes (hap1+hap2). (v) We assembled the synthetic HiFi reads using RAMbler, hifiasm, LJA, HiCANU, and Verkko.

The goal of these experiments was to evaluate the quality of the assemblies produced by RAMbler, hifiasm, LJA, HiCANU, and Verkko on these synthetic diploid genomes using various metrics. We recorded the total number of contigs, the overall assembly size, the number of resolved repeat copies, and the number of contigs with haplotype switching. On these synthetic HiFi datasets with minimal divergence between haplotypes, LJA, HiCANU, and Verkko are expected to produce phased assemblies, but not fully haplotype-resolved assemblies due to the limited length of HiFi reads. They typically require either long-range sequencing data (e.g., ultra-long ONT or Hi-C reads) or phasing data (e.g., maternal and paternal reads) for full haplotype resolution. In HiFi-only mode, LJA, HiCANU, and Verkko generate a single FASTA file containing both haplotypes. hifiasm

instead, attempts to resolve haplotypes even in HiFi-only mode by joining phased assembly blocks to achieve greater contiguity. It produces separate FASTA files for primary and alternate assemblies. Thus, the assemblies produced by LJA, HiCANU, and Verkko are expected to have a smaller number of haplotype switches, but more fragmented than hifiasm. Since LJA, HiCANU, and Verkko are expected to resolve repeat copies from both haplotypes, their total assembly size is expected to be close to the sum of the two haplotype lengths. Instead, since hifiasm assigns phased blocks to either the primary or the alternate assembly, it is expected to incur more haplotype switch errors, and generate pseudo-haplotypes that capture hap1 as primary and hap2 as alternate (or vice versa). RAMbler also prioritizes long contig generation and outputs a single FASTA file representing the primary pseudo-haplotype. This assembly is expected to capture hap1, which contains the larger repeat copy count, possibly including some haplotype switches.

We used BLAST [86] to align all the repeat units in hap1 and hap2 to each target assembly. We defined a repeat unit R to be *resolved* by an assembly A if the BLAST output indicated a perfect identity (i.e., 99.99% or higher) of R in A . Any alignment with less than perfect identity was disregarded. We also recorded whether any contig of a target assembly contained a mix of repeat units from both haplotypes, which indicated switching errors.

The experimental results are summarized in Tables 2.6 to 2.13. We omitted Verkko from the tables because it failed to produce any output in most cases. Verkko failed 92 runs out of 108 for unknown reasons (the logs were uninformative). Tables 2.6 to 2.9 report the number of contigs and the total assembly size. Tables 2.10 to 2.13 report the number of resolved repeat copies and the number of contigs containing haplotype switching errors. In Tables 2.10 to 2.13, red numbers indicate incorrect repeat copy counts (neither matching either haplotype copy counts nor their sum),

blue numbers indicate copy counts that match the sum of copies for both haplotypes (hap1+hap2), and grey cells indicate assemblies with haplotype switches.

Tables 2.6, 2.7, 2.10 and 2.11 show the experimental results when hap1 had 5 copies and hap2 had 3 – 5 copies. For these data sets, RAMbler produced an assembly containing 5 copies of the repeat unit in 47 cases out of 54 (87%) with 8 haplotype switches. hifiasm produced the correct number of copies for hap1 in 48 cases out of 54 (89%), and the correct number of copies for hap2 in 46 cases out of 54 (85%) with 7 and 1 haplotype switches, respectively. RAMbler however, always produced a single contig, while hifiasm produced 2-3 contigs in some cases (the average number of contigs was 1.09 for both primary and alternate assembly). LJA resolved hap1+hap2 repeat copies in 19 cases out of 54 (35%), and sometimes only hap1 repeat copies in 2 cases out of 54 (4%). LJA produced more fragmented assemblies (the average number of contigs was 3.65), with a total of 7 haplotype switching errors. HiCANU resolved hap1+hap2 repeat copies in 10 cases out of 54 (19%), and sometimes only hap1 repeat copies in 9 cases out of 54 (17%). HiCANU produced more fragmented assemblies (the average number of contigs was 7.49), with a total of 12 haplotype switching errors. Overall, LJA and HiCANU performed worse than RAMbler and hifiasm on these datasets, producing assemblies that were more fragmented and contained less repeat copies.

Tables 2.8, 2.9, 2.12 and 2.13 summarize the cases where hap1 had 10 copies and hap2 had 6, 8 or 10 copies. RAMbler resolved 10 copies of the repeat unit in 32 cases out of 54 (59%) with 14 haplotype switching errors. The average number of contigs over all RAMbler's assemblies was 1.26. hifiasm resolved 10 repeat copies in 40 cases out of 54 (74%) with 17 haplotype switching errors. The average number of contigs over all hifiasm's assemblies was 1.06 for the primary, and 1.07 for the alternate. LJA's assemblies matched either hap1 or hap1+hap2 in 19 cases out of 54

(35%) with 18 haplotype switching errors. The average number of contigs over all LJA's assemblies was 7.52. HiCANU's assemblies matched either hap1 or hap1+hap2 in 16 cases out of 54 (30%) with 21 haplotype switching errors. The average number of contigs over all HiCANU's assemblies was 13.81. LJA and HiCANU, again, performed worse than RAMbler and hifiasm on these datasets.

In summary (see Table 2.14), RAMbler captured the larger repeat count between the two haplotypes in most cases. It frequently produced a single contig with a low number of haplotype switching errors, relying exclusively on HiFi reads.

2.5 Discussion

We introduced RAMbler, a reference-guided genome assembler aimed at resolving complex repetitive regions. To the best of our knowledge, there is no other specialized assembler for resolving complex repeats that uses HiFi reads exclusively. Both SDA and CentroFlye expect as input error-prone PacBio CLR or ONT long reads. SDA is no longer maintained and CentroFlye requires very high computational resources and additional information about the centromeres of interest.

RAMBler leverages *uni*k*mers* to detect overlaps and locally assemble the HiFi reads. By taking advantage of shared *uni*k*mers*, RAMbler can select safe and informative overlaps that are difficult to identify by traditional assemblers. Traditional HiFi assemblers rely on highly accurate (but not necessarily perfect) overlaps to build a string/overlap graph, which is a method that works very

Repeat unit size (bp)	Haplotype 1 genome size (bp)	Haplotype 2		<i>p</i>	Number of contigs					Assembly size (bp)				
		Number of copies	Genome size (bp)		RAmbler	hifiasm		LJA	HiCANU	RAmbler	hifiasm		LJA	HiCANU
						primary	alternate				primary	alternate		
5,000	125,000	3	115,000	1/250	1	1	1	2	2	124,014	124,044	114,044	236,342	153,146
				1/500	1	1	1	2	2	123,792	124,733	114,733	238,326	158,925
				1/1000	1	1	1	2	1	124,516	124,506	114,506	234,154	118,297
		4	120,000	1/250	1	1	1	1	2	124,772	124,772	119,772	124,610	146,101
				1/500	1	1	1	2	1	122,954	124,190	119,190	237,137	120,875
				1/1000	1	1	1	2	2	119,684	124,678	119,678	242,786	155,645
		5	125,000	1/250	1	1	1	2	3	124,816	124,845	124,845	248,430	167,028
				1/500	1	1	1	2	2	124,459	124,453	124,453	245,850	172,225
				1/1000	1	1	1	2	2	124,516	124,509	124,509	247,492	137,479
10,000	150,000	3	130,000	1/250	1	1	1	2	2	149,704	149,768	129,768	275,648	196,186
				1/500	1	1	1	2	1	149,460	149,522	129,522	277,752	124,994
				1/1000	1	1	1	2	2	149,938	149,894	129,894	279,752	178,305
		4	140,000	1/250	1	1	1	2	2	149,740	149,734	139,734	281,318	199,970
				1/500	1	1	1	2	2	149,272	149,277	139,277	286,966	203,767
				1/1000	1	1	1	2	2	149,606	149,606	139,606	287,786	183,662
		5	150,000	1/250	1	1	1	1	2	149,824	149,827	149,827	149,599	170,422
				1/500	1	1	1	2	4	149,686	149,747	149,747	246,773	203,411
				1/1000	1	1	1	6	4	148,757	149,564	149,564	259,079	211,558
15,000	175,000	3	145,000	1/250	1	1	1	2	4	174,718	174,719	144,719	314,955	244,935
				1/500	1	1	1	2	8	174,012	174,557	144,557	303,079	255,277
				1/1000	1	1	1	2	5	174,244	174,304	144,304	318,018	231,616
		4	160,000	1/250	1	1	1	3	4	174,683	174,649	159,649	333,299	243,259
				1/500	1	1	1	1	5	174,163	174,166	159,166	173,611	246,750
				1/1000	1	1	1	3	3	174,696	174,683	159,683	275,311	248,097
		5	175,000	1/250	1	1	1	2	6	173,791	173,792	173,792	234,321	253,340
				1/500	1	1	1	5	3	174,765	174,847	174,847	301,644	224,413
				1/1000	1	1	1	4	3	219,085	174,650	174,650	242,524	249,085

Table 2.6: Number of contigs and assembly size for the assemblies produced by RAmbler, hifiasm, LJA, and HiCANU; assemblers were given in input synthetic HiFi reads with coverage $30\times$ generated from synthetic diploid genomes with repeat unit of length $\{5, 10, 15\}$ kb; in each synthetic genome the primary haplotype (hap1) had five copies of the repeat unit, while the secondary haplotype (hap2) had 3-5 copies

Repeat unit size (bp)	Haplotype 1 genome size	Haplotype 2		p	Number of contigs					Assembly size (bp)				
		Number of copies	Genome size (bp)		RAmbler	hifiasm		LJA	HiCANU	RAmbler	hifiasm		LJA	HiCANU
	primary					alternate	primary				alternate			
20,000	200,000	3	160,000	1/250	1	1	1	2	5	199,897	199,884	159,884	358,247	282,097
				1/500	1	1	1	2	3	199,853	199,854	159,854	358,946	283,088
				1/1000	1	1	1	5	9	238,526	238,525	72,170	289,383	352,156
		4	180,000	1/250	1	1	1	2	3	199,378	199,490	179,490	334,210	247,370
				1/500	1	1	1	2	7	198,949	199,456	179,456	367,218	295,081
				1/1000	1	1	1	4	8	199,560	199,562	179,562	311,928	358,247
		5	200,000	1/250	1	1	1	3	3	199,555	199,555	199,555	397,504	268,243
				1/500	1	1	1	2	4	199,349	199,344	199,344	397,630	233,761
				1/1000	1	1	2	4	6	199,723	219,716	202,723	376,751	353,151
50,000	350,000	3	250,000	1/250	1	1	1	4	5	349,544	349,540	249,541	619,973	433,210
				1/500	1	1	1	3	6	399,665	349,654	249,654	550,810	389,677
				1/1000	1	1	1	12	14	448,922	349,542	249,542	556,744	616,291
		4	300,000	1/250	1	1	1	2	13	349,296	349,335	299,335	599,897	550,346
				1/500	1	1	1	3	7	349,649	349,679	299,679	661,998	534,924
				1/1000	1	2	1	6	8	349,634	364,122	299,685	572,956	533,915
		5	350,000	1/250	1	1	1	4	19	349,651	349,646	349,646	704,151	667,578
				1/500	1	1	1	4	14	449,784	449,965	183,554	655,052	457,076
				1/1000	1	1	1	5	16	349,220	349,219	308,939	611,751	656,526
100,000	600,000	3	400,000	1/250	1	1	1	6	15	598,560	599,235	399,237	1,035,574	803,380
				1/500	1	1	2	5	18	599,336	699,758	310,632	983,101	802,113
				1/1000	1	2	1	17	24	599,644	615,325	399,668	963,902	867,330
		4	500,000	1/250	1	1	1	4	15	599,834	599,831	499,831	1,111,210	829,144
				1/500	1	1	2	6	16	599,322	599,323	522,346	1,131,188	832,552
				1/1000	1	2	2	10	22	599,015	410,687	679,805	1,010,779	987,162
		5	600,000	1/250	1	1	1	3	14	599,784	599,779	599,779	1,143,983	930,005
				1/500	1	1	1	5	23	599,418	699,418	322,057	1,225,788	1,032,964
				1/1000	1	3	2	10	28	699,614	565,106	640,252	1,215,171	1,114,022

Table 2.7: Number of contigs and assembly size for the assemblies produced by RAmbler, hifiasm, LJA, and HiCANU; assemblers were given in input synthetic HiFi reads with coverage $30\times$ generated from synthetic diploid genomes with repeat unit of length $\{20, 50, 100\}$ kb; in each synthetic genome the primary haplotype (hap1) had five copies of the repeat unit, while the secondary haplotype (hap2) had 3-5 copies

Repeat unit size (bp)	Haplotype 1 genome size (bp)	Haplotype 2		<i>p</i>	Number of contigs					Assembly size (bp)				
		Number of copies	Genome size (bp)		RAmbler	hifiasm		LJA	HiCANU	RAmbler	hifiasm		LJA	HiCANU
						primary	alternate				primary	alternate		
5,000	150,000	6	130,000	1/250	1	1	1	2	2	149,798	149,807	129,807	278,735	204,153
				1/500	1	1	1	2	2	149,600	149,584	129,584	273,803	175,265
				1/1000	1	1	1	3	3	149,821	149,816	129,816	157,944	185,932
		8	140,000	1/250	1	1	1	2	4	149,814	149,814	139,814	281,722	204,717
				1/500	1	1	1	2	3	149,598	149,700	139,700	287,414	160,116
				1/1000	1	1	1	4	3	149,146	149,698	139,699	213,196	154,213
		10	150,000	1/250	1	1	1	2	3	149,520	149,522	149,522	294,735	209,336
				1/500	2	1	1	2	3	157,601	149,596	149,596	299,977	177,134
				1/1000	1	1	1	2	4	149,533	149,534	149,534	251,604	218,802
10,000	200,000	6	160,000	1/250	1	1	1	3	3	198,907	198,943	158,943	356,387	284,989
				1/500	1	1	1	3	3	208,918	198,966	158,966	313,973	270,487
				1/1000	1	1	1	6	4	198,803	198,799	158,799	367,402	277,901
		8	180,000	1/250	1	1	1	2	3	199,909	199,903	179,903	259,132	313,321
				1/500	1	1	1	2	3	259,751	199,808	179,808	334,697	250,747
				1/1000	1	1	1	3	5	159,892	199,802	179,802	280,797	338,852
		10	200,000	1/250	1	1	1	2	8	199,143	199,138	199,138	286,264	311,887
				1/500	1	1	1	3	6	199,387	199,386	199,386	342,217	329,246
				1/1000	1	1	1	2	12	199,921	199,912	199,912	343,594	406,570
15,000	250,000	6	190,000	1/250	1	1	1	5	5	249,666	249,666	189,666	453,741	336,368
				1/500	1	1	1	3	8	249,774	249,774	189,783	454,197	364,820
				1/1000	2	1	2	3	8	218,440	264,806	204,702	318,102	374,178
		8	220,000	1/250	1	1	1	2	5	249,801	249,801	219,801	464,700	404,715
				1/500	1	1	1	2	8	219,761	309,760	115,902	461,678	371,045
				1/1000	2	1	1	10	7	393,419	249,891	219,891	463,866	399,737
		10	250,000	1/250	1	1	1	4	3	263,324	369,706	83,795	479,210	394,700
				1/500	1	1	1	2	3	219,559	249,554	249,554	495,509	387,150
				1/1000	3	1	0	7	10	235,122	324,922	0	437,369	420,817

Table 2.8: Number of contigs and assembly size for the assemblies produced by RAmbler, hifiasm, LJA, and HiCANU; assemblers were given in input synthetic HiFi reads with coverage $30\times$ generated from synthetic diploid genomes with repeat unit of length $\{5, 10, 15\}$ kb; in each synthetic genome the primary haplotype (hap1) had ten copies of the repeat unit, while the secondary haplotype (hap2) had 6, 8 or 10 copies

Repeat unit size (bp)	Haplotype 1 genome size (bp)	Haplotype 2		p	Number of contigs					Assembly size (bp)				
		Number of copies	Genome size (bp)		RAmbler	hifiasm		LJA	HiCANU	RAmbler	hifiasm		LJA	HiCANU
						primary	alternate				primary	alternate		
20,000	300,000	6	220,000	1/250	1	1	1	4	4	398,624	298,623	218,623	534,817	373,343
				1/500	1	1	1	3	6	299,828	299,831	219,831	445,980	364,339
				1/1000	1	1	1	3	4	298,587	263,652	219,618	389,514	388,469
		8	260,000	1/250	1	1	1	2	10	398,674	299,601	259,601	559,192	468,951
				1/500	1	1	1	6	7	299,403	299,466	259,466	504,472	467,505
				1/1000	3	1	1	5	5	296,036	299,811	259,811	502,147	392,303
		10	300,000	1/250	1	1	1	5	12	299,429	299,473	299,479	549,615	495,926
				1/500	1	1	1	2	10	319,347	299,345	299,345	594,195	451,130
				1/1000	1	1	2	13	19	239,745	419,694	199,766	466,960	556,297
50,000	600,000	6	400,000	1/250	1	1	2	6	10	749,565	749,564	261,223	994,336	715,594
				1/500	1	1	1	7	13	598,937	598,942	398,942	925,192	726,548
				1/1000	2	1	1	15	26	679,365	614,317	399,952	976,037	1,077,795
		8	500,000	1/250	1	1	1	10	18	598,481	599,775	499,775	1,098,032	817,291
				1/500	1	1	1	2	16	599,795	599,791	499,791	1,097,101	838,820
				1/1000	1	1	1	19	35	599,763	599,764	499,764	1,018,018	1,246,016
		10	600,000	1/250	3	1	1	8	18	845,259	598,997	598,997	1,230,662	998,718
				1/500	1	1	1	8	22	599,707	599,707	599,707	1,207,967	1,165,785
				1/1000	1	3	1	32	43	749,327	614,349	560,593	1,104,150	1,344,584
100,000	1,100,000	6	700,000	1/250	1	1	1	7	31	1,399,769	1,099,766	699,766	1,661,520	1,384,759
				1/500	1	1	1	4	29	1,098,917	1,099,745	699,742	1,761,361	1,298,079
				1/1000	3	2	3	55	42	1,021,424	971,330	946,443	1,910,184	1,912,471
		8	900,000	1/250	1	1	1	10	23	1,598,393	1,099,455	899,455	1,927,926	1,546,332
				1/500	2	1	1	11	26	1,146,069	1,098,945	898,945	1,978,251	1,432,989
				1/1000	1	1	1	38	61	1,399,349	1,599,474	899,474	1,971,769	2,237,345
		10	1,100,000	1/250	1	1	1	13	33	1,099,462	1,099,460	1,099,460	2,199,749	1,822,173
				1/500	1	1	1	7	37	1,099,255	1,199,265	934,535	2,256,628	1,942,282
				1/1000	2	1	1	26	55	376,999	1,299,672	1,099,672	2,352,688	2,362,493

Table 2.9: Number of contigs and assembly size for the assemblies produced by RAmbler, hifiasm, LJA, and HiCANU; assemblers were given in input synthetic HiFi reads with coverage $30\times$ generated from synthetic diploid genomes with repeat unit of length $\{20, 50, 100\}$ kb; in each synthetic genome the primary haplotype (hap1) had ten copies of the repeat unit, while the secondary haplotype (hap2) had 6, 8 or 10 copies

Repeat unit size (bp)	# of copies in hap2	p	RAmbler				hifiasm (primary)				hifiasm (alternate)				LJA				HiCANU			
			# copies resolved			# switch errors	# copies resolved			# switch errors	# copies resolved			# switch errors	# copies resolved			# switch errors	# copies resolved			# switch errors
			hap1	hap2	sum		hap1	hap2	sum		hap1	hap2	sum		hap1	hap2	sum		hap1	hap2	sum	
5,000	3	1/250	5	0	5	0	5	0	5	0	0	3	3	0	5	3	8	0	5	3	8	0
		1/500	5	0	5	0	5	0	5	0	0	3	3	0	5	3	8	0	5	3	8	0
		1/1000	5	0	5	0	5	0	5	0	0	3	3	0	5	3	8	0	2	2	4	1
	4	1/250	5	0	5	0	5	0	5	0	0	4	4	0	5	0	5	0	5	1	6	0
		1/500	5	0	5	0	5	0	5	0	0	4	4	0	5	3	8	0	5	0	5	0
		1/1000	0	4	4	0	5	0	5	0	0	4	4	0	4	3	7	0	5	4	9	0
	5	1/250	0	5	5	0	0	5	5	0	5	0	5	0	5	5	10	0	4	5	9	1
		1/500	0	5	5	0	0	5	5	0	5	0	5	0	5	4	9	0	4	4	8	0
		1/1000	0	5	5	0	0	5	5	0	5	0	5	0	5	5	10	0	1	5	6	0
10,000	3	1/250	5	0	5	0	5	0	5	0	0	3	3	0	4	3	7	0	5	3	8	0
		1/500	5	0	5	0	5	0	5	0	0	3	3	0	3	3	6	0	0	3	3	0
		1/1000	5	0	5	0	5	0	5	0	0	3	3	0	5	3	8	0	1	3	4	0
	4	1/250	5	0	5	0	5	0	5	0	0	4	4	0	5	4	9	0	5	4	9	0
		1/500	5	0	5	0	5	0	5	0	0	4	4	0	5	4	9	0	5	2	7	0
		1/1000	5	0	5	0	5	0	5	0	0	4	4	0	4	3	7	0	3	3	6	0
	5	1/250	0	5	5	0	0	5	5	0	5	0	5	0	5	0	5	0	5	1	6	0
		1/500	5	0	5	0	0	5	5	0	5	0	5	0	4	4	8	0	4	2	6	1
		1/1000	0	5	5	0	0	5	5	0	5	0	5	0	5	5	10	1	3	4	7	0
15,000	3	1/250	5	0	5	0	5	0	5	0	0	3	3	0	4	3	7	0	5	3	8	1
		1/500	5	0	5	0	5	0	5	0	0	3	3	0	3	1	4	0	1	1	2	0
		1/1000	5	0	5	0	5	0	5	0	0	3	3	0	4	3	7	0	1	0	1	0
	4	1/250	5	0	5	0	5	0	5	0	0	4	4	0	5	4	9	0	4	1	5	0
		1/500	5	0	5	0	5	0	5	0	0	4	4	0	4	0	4	0	2	2	4	0
		1/1000	5	0	5	0	5	0	5	0	0	4	4	0	5	4	9	0	4	2	6	0
	5	1/250	0	5	5	0	5	0	5	0	0	5	5	0	2	4	6	0	2	3	5	0
		1/500	5	0	5	0	5	0	5	0	0	5	5	0	5	5	10	2	1	1	2	1
		1/1000	4	4	8	1	0	5	5	0	5	0	5	0	5	5	10	2	4	4	8	1

Table 2.10: Number of repeat copies resolved and number of haplotype switching in the assemblies produced by RAmbler, hifiasm, LJA, and HiCANU; assemblers were given in input synthetic HiFi reads with coverage $30\times$ generated from synthetic diploid genomes with repeat unit of length $\{5, 10, 15\}$ kb; in each synthetic genome the primary haplotype (hap1) had five copies of the repeat unit, while the secondary haplotype (hap2) had 3-5 copies

Repeat unit size (bp)	# of copies in hap2	<i>p</i>	RAmbler				hifiasm (primary)				hifiasm (alternate)				LJA				HiCANU			
			# copies resolved			# switch	# copies resolved			# switch	# copies resolved			# switch	# copies resolved			# switch	# copies resolved			# switch
			hap1	hap2	sum	errors	hap1	hap2	sum	errors	hap1	hap2	sum	errors	hap1	hap2	sum	errors	hap1	hap2	sum	errors
20,000	3	1/250	5	0	5	0	5	0	5	0	0	3	3	0	4	2	6	0	5	1	6	1
		1/500	5	0	5	0	5	0	5	0	0	3	3	0	5	3	8	0	5	3	8	0
		1/1000	4	3	7	1	4	2	6	1	0	2	2	0	5	3	8	0	1	2	3	1
	4	1/250	5	0	5	0	5	0	5	0	0	4	4	0	5	3	8	0	3	4	7	0
		1/500	3	2	5	1	5	0	5	0	0	4	4	0	5	4	9	0	4	2	6	0
		1/1000	5	0	5	0	5	0	5	0	0	4	4	0	5	4	9	0	5	0	5	0
	5	1/250	0	5	5	0	0	5	5	0	5	0	5	0	5	3	8	0	3	2	5	0
		1/500	0	5	5	0	0	5	5	0	5	0	5	0	3	5	8	0	2	2	4	0
		1/1000	5	0	5	0	3	2	5	1	4	4	8	0	4	4	8	1	5	4	9	1
50,000	3	1/250	5	0	5	0	5	0	5	0	0	3	3	0	2	2	4	0	3	2	5	0
		1/500	4	2	6	1	5	0	5	0	0	3	3	0	4	2	6	0	1	1	2	0
		1/1000	5	2	7	1	5	0	5	0	0	3	3	0	4	3	7	0	3	2	5	0
	4	1/250	5	0	5	0	5	0	5	0	0	4	4	0	4	3	7	0	1	2	3	0
		1/500	5	0	5	0	5	0	5	0	0	4	4	0	5	3	8	0	4	1	5	0
		1/1000	5	0	5	0	5	0	5	0	0	4	4	0	5	3	8	1	1	2	3	0
	5	1/250	5	0	5	0	5	0	5	0	0	5	5	0	4	5	9	0	2	1	3	0
		1/500	4	3	7	1	2	4	6	1	3	0	3	0	5	5	10	0	5	5	10	1
		1/1000	5	0	5	0	0	5	5	0	5	0	5	0	1	2	3	0	1	3	4	0
100,000	3	1/250	5	0	5	0	5	0	5	0	0	3	3	0	4	2	6	0	5	3	8	1
		1/500	5	0	5	0	5	1	6	1	5	3	8	1	5	3	8	0	1	1	2	0
		1/1000	5	0	5	0	5	0	5	0	0	3	3	0	4	3	7	1	2	2	4	0
	4	1/250	5	0	5	0	5	0	5	0	0	4	4	0	2	4	6	0	2	3	5	0
		1/500	5	0	5	0	1	3	4	1	4	1	5	0	4	3	7	0	1	1	2	0
		1/1000	4	1	5	1	4	0	4	0	4	4	8	0	4	4	8	0	0	1	1	0
	5	1/250	0	5	5	0	0	5	5	0	5	0	5	0	3	4	7	1	3	3	6	0
		1/500	5	0	5	0	3	2	5	1	0	3	3	0	3	4	7	0	1	0	1	0
		1/1000	4	2	6	1	5	5	10	1	5	2	7	0	4	5	9	0	5	5	10	1

Table 2.11: Number of repeat copies resolved and number of haplotype switching in the assemblies produced by RAmbler, hifiasm, LJA, and HiCANU; assemblers were given in input synthetic HiFi reads with coverage $30\times$ generated from synthetic diploid genomes with repeat unit of length $\{20, 50, 100\}$ kb; in each synthetic genome the primary haplotype (hap1) had five copies of the repeat unit, while the secondary haplotype (hap2) had 3-5 copies

Repeat unit size (bp)	# of copies in hap2	<i>p</i>	RAmbler				hifiasm (primary)				hifiasm (alternate)				LJA				HiCANU			
			# copies resolved			# switch	# copies resolved			# switch	# copies resolved			# switch	# copies resolved			# switch	# copies resolved			# switch
			hap1	hap2	sum	errors	hap1	hap2	sum	errors	hap1	hap2	sum	errors	hap1	hap2	sum	errors	hap1	hap2	sum	errors
5,000	6	1/250	10	0	10	0	10	0	10	0	0	6	6	0	9	6	15	0	10	6	16	0
		1/500	10	0	10	0	10	0	10	0	0	6	6	0	9	4	13	0	10	2	12	0
		1/1000	10	0	10	0	10	0	10	0	0	6	6	0	10	6	16	1	9	3	12	1
	8	1/250	10	0	10	0	10	0	10	0	0	8	8	0	9	6	15	0	10	3	13	0
		1/500	10	0	10	0	10	0	10	0	0	8	8	0	10	7	17	0	3	3	6	1
		1/1000	6	4	10	1	8	7	15	1	6	8	14	1	10	8	18	4	7	7	14	2
	10	1/250	10	0	10	0	10	0	10	0	0	10	10	0	10	9	19	0	10	7	17	0
		1/500	7	6	13	0	10	0	10	0	0	10	10	0	9	10	19	0	1	10	11	1
		1/1000	0	10	10	0	0	10	10	0	10	0	10	0	10	10	20	0	5	7	12	0
10,000	6	1/250	10	0	10	0	10	0	10	0	0	6	6	0	7	6	13	0	8	6	14	0
		1/500	10	1	11	1	10	0	10	0	0	6	6	0	6	3	9	0	7	4	11	0
		1/1000	10	0	10	0	10	0	10	0	0	6	6	0	10	6	16	4	7	3	10	0
	8	1/250	10	0	10	0	10	0	10	0	0	8	8	0	6	5	11	0	10	6	16	0
		1/500	10	6	16	1	10	0	10	0	0	8	8	0	8	8	16	0	4	6	10	0
		1/1000	2	3	5	1	8	2	10	1	3	5	8	1	10	8	18	1	7	6	13	1
	10	1/250	10	0	10	0	0	10	10	0	10	0	10	0	7	8	15	0	6	8	14	0
		1/500	0	10	10	0	10	0	10	0	0	10	10	0	6	8	14	0	8	8	16	0
		1/1000	10	0	10	0	10	0	10	0	0	10	10	0	9	10	19	0	6	8	14	1
15,000	6	1/250	10	0	10	0	10	0	10	0	0	6	6	0	10	6	16	0	6	6	12	0
		1/500	10	0	10	0	10	0	10	0	0	6	6	0	9	6	15	0	6	4	10	0
		1/1000	3	6	9	0	7	4	11	1	5	2	7	1	10	4	14	2	10	6	16	1
	8	1/250	10	0	10	0	10	0	10	0	0	8	8	0	8	4	12	0	10	5	15	0
		1/500	0	8	8	0	9	4	13	1	0	5	5	0	10	7	17	0	10	8	18	2
		1/1000	10	7	17	0	10	0	10	0	0	8	8	0	10	7	17	4	8	5	13	1
	10	1/250	10	1	11	1	9	8	17	1	0	3	3	0	9	6	15	0	9	6	15	0
		1/500	0	8	8	0	0	9	9	0	8	0	8	0	9	9	18	0	7	8	15	0
		1/1000	2	4	6	0	8	6	14	1	0	0	0	0	10	10	20	2	10	10	20	2

Table 2.12: Number of repeat copies resolved and number of haplotype switching in the assemblies produced by RAmbler, hifiasm, LJA, and HiCANU; assemblers were given in input synthetic HiFi reads with coverage $30\times$ generated from synthetic diploid genomes with repeat unit of length $\{5, 10, 15\}$ kb; in each synthetic genome the primary haplotype (hap1) had ten copies of the repeat unit, while the secondary haplotype (hap2) had 6, 8 or 10 copies

Repeat unit size (bp)	# of copies in hap2	<i>p</i>	RAmbler				hifiasm (primary)				hifiasm (alternate)				LJA				HiCANU			
			# copies resolved			# switch	# copies resolved			# switch	# copies resolved			# switch	# copies resolved			# switch	# copies resolved			# switch
			hap1	hap2	sum	errors	hap1	hap2	sum	errors	hap1	hap2	sum	errors	hap1	hap2	sum	errors	hap1	hap2	sum	errors
20,000	6	1/250	10	5	15	1	9	0	9	0	0	6	6	0	7	6	13	0	7	3	10	0
		1/500	10	0	10	0	6	4	10	1	4	2	6	1	9	4	13	0	4	5	9	0
		1/1000	10	0	10	0	7	3	10	1	3	3	6	1	9	6	15	0	6	5	11	1
	8	1/250	10	5	15	1	10	0	10	0	0	8	8	0	8	6	14	2	4	4	8	0
		1/500	10	0	10	0	9	2	11	1	0	6	6	0	9	8	17	0	6	7	13	0
		1/1000	3	3	6	0	8	0	8	0	2	8	10	1	10	8	18	1	5	5	10	0
	10	1/250	10	0	10	0	0	10	10	0	10	0	10	0	7	7	14	0	10	10	20	3
		1/500	2	9	11	1	10	0	10	0	0	10	10	0	10	10	20	0	5	2	7	1
		1/1000	7	0	7	0	7	8	15	1	10	10	20	1	10	10	20	2	10	10	20	3
50,000	6	1/250	10	3	13	1	9	3	12	1	1	3	4	1	6	4	10	0	5	2	7	1
		1/500	10	0	10	0	10	0	10	0	0	6	6	0	10	4	14	1	4	2	6	0
		1/1000	9	1	10	1	10	0	10	0	0	6	6	0	10	6	16	0	7	3	10	0
	8	1/250	10	0	10	0	10	0	10	0	0	8	8	0	10	8	18	1	2	4	6	0
		1/500	10	0	10	0	6	4	10	1	4	4	8	1	9	8	17	0	3	5	8	1
		1/1000	10	0	10	0	10	0	10	0	0	8	8	0	10	8	18	5	10	8	18	4
	10	1/250	8	7	15	0	7	3	10	1	3	7	10	1	7	7	14	0	4	3	7	0
		1/500	0	10	10	0	10	0	10	0	0	10	10	0	7	10	17	0	10	10	20	1
		1/1000	8	5	13	1	10	4	14	1	0	10	10	0	10	10	20	9	5	6	11	0
100,000	6	1/250	10	3	13	1	10	0	10	0	0	6	6	0	10	6	16	1	4	2	6	0
		1/500	10	0	10	0	10	0	10	0	0	6	6	0	8	5	13	0	10	6	16	1
		1/1000	9	1	10	0	3	6	9	0	8	1	9	1	10	6	16	5	10	6	16	1
	8	1/250	10	5	15	1	10	0	10	0	0	8	8	0	4	5	9	0	3	5	8	0
		1/500	4	6	10	0	5	5	10	1	5	3	8	1	8	6	14	0	5	3	8	0
		1/1000	10	3	13	1	2	8	10	1	8	0	8	0	10	8	18	3	2	2	4	0
	10	1/250	10	0	10	0	0	10	10	0	10	0	10	0	8	9	17	0	6	1	7	0
		1/500	10	0	10	0	9	2	11	1	0	8	8	0	6	10	16	0	6	1	7	0
		1/1000	1	2	3	0	0	10	10	0	10	0	10	0	10	10	20	4	6	6	12	2

Table 2.13: Number of repeat copies resolved and number of haplotype switching in the assemblies produced by RAmbler, hifiasm, LJA, and HiCANU; assemblers were given in input synthetic HiFi reads with coverage $30\times$ generated from synthetic diploid genomes with repeat unit of length $\{20, 50, 100\}$ kb; in each synthetic genome the primary haplotype (hap1) had ten copies of the repeat unit, while the secondary haplotype (hap2) had 6, 8 or 10 copies

# of copies in hap1	# test cases		RAmbler	hifiasm		LJA	HiCANU
				primary	alternate		
5	54	avg. # contigs	1	1.09	1.09	3.65	7.48
		# match hap1	47	48	46	2	9
		# match hap1/hap2	48	50	47	3	12
		# match total	0	1	1	19	10
		# mismatch	6	3	6	32	32
		# switch error	8	7	1	7	12
10	54	avg. # contigs	1.26	1.06	1.07	7.52	13.81
		# match hap1	32	40	43	1	6
		# match hap1/hap2	33	41	44	1	12
		# match total	0	0	1	18	10
		# mismatch	21	13	9	35	32
		# switch error	14	17	12	18	21

Table 2.14: Summary of assembly performance of RAmbler, hifiasm, LJA, and HiCANU on synthetic diploid genomes with copy number variation

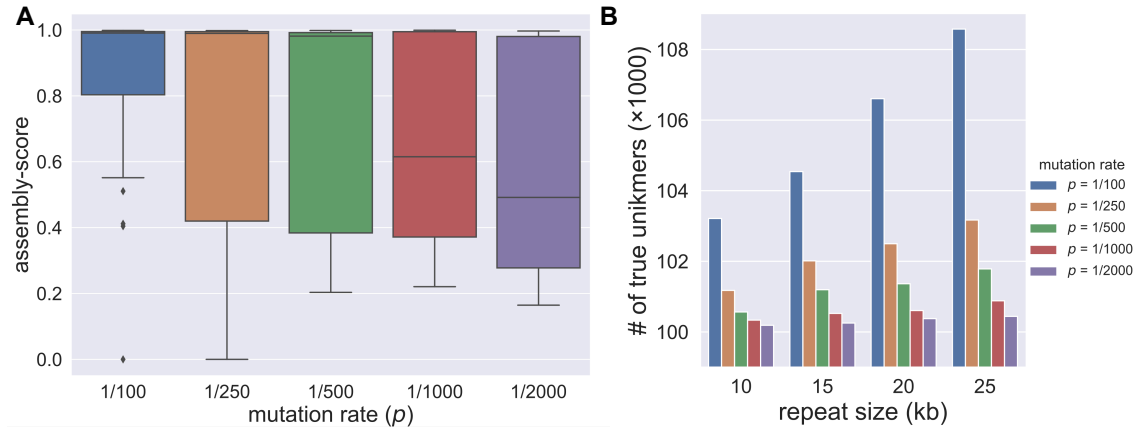


Figure 2.13: Relationship among RAMbler's performance, mutation rate p , and number of true unikmers; (A) Performance of RAMbler for several choices of the mutation rate p over 27 different combinations of repetitive regions with repeat sizes $\{10, 15, 20\}$ kb, number of copies $\{2, 5, 10\}$, coverage depth $= \{20\times, 30\times, 40\times\}$; (B) Number of true unikmers as a function of mutation rate and repeat unit's sizes (five copies)

well for non-repetitive regions of the genome. For instance, hifiasm carries out an all-pairs sequence alignment of the HiFi reads before building the string graph. Highly repetitive regions generate an overwhelming number of prefix-suffix overlaps, which are difficult to process. In contrast, the use of unikmers allows RAMbler to detect informative overlaps without the need of sifting through a very large number of sequence alignments. The use of unikmers also allows RAMbler to eliminate the need of the error correction step to compensate for rare sequencing errors in HiFi reads, as it is done in hifiasm.

RAMBler still has some limitations. In simulations, its performance drops significantly when the mutation rate p is smaller than $1/1000$ (see Figure 2.13A). In this case, the individual copies inside the repetitive regions are almost identical, thus there are not enough unikmers to re-

solve them. This observation is supported by Figure 2.13B, which illustrates that the number of uni/k mers decreases with p . However, the mutation rate in eukaryotic genomes is generally higher than one SNP over a thousand base pairs [108, 56, 98]. The current implementation of RAMbler is not multi-threaded which penalizes its runtime. This is an opportunity to improve RAMbler's runtime. In addition, RAMbler generates pseudo-haplotype assemblies from HiFi data, while LJA, Hi-CANU, and Verkko produce phased assemblies and hifiasm always attempts to produce haplotype-resolved assemblies. HiCANU, hifiasm and Verkko support trio-based enhanced phasing. hifiasm and Verkko can use Hi-C reads to further improve haplotype resolution. Currently our tool lacks these features, which we plan to add in future versions of RAMbler.

2.6 Software Availability

RAMbler is available from GitHub (<https://github.com/ucrbioinfo/rambler>) and as Supplemental Code in the Appendix A.

Chapter 3

RAmbler Resolves Complex Repeats in Human Chromosomes 8, 19 and X

3.1 Introduction

In this chapter, (1) we extend RAMbler, introduced in Chapter 2, for handling cycles in overlap graphs; (2) we analyze the impacts of HiFi sequencing coverage bias at repetitive regions on RAMbler’s performance; (3) we also perform thorough analysis of reference bias in RAMbler; (4) we carry out experiments to ensure that setting the value of parameter *mo* (minimum overlap for the overlap graph) to 1,000 bp is stringent enough to avoid spurious overlaps in the human genome; (5) Using NucFreq [128], CRAQ [76], QUAST [48], and SyRI [45], we show that RAMbler can resolve complex repeats (both centromeric and non-centromeric) in human Chromosomes 8, 19, and X from HiFi data to a quality comparable to the telomere-to-telomere (T2T) human genome assembly. (6) Due to the lack of specialized HiFi assemblers for repeats, we compare RAMbler

Reference Genome	Sample	Illumina (NGS)		Oxford Nanopore (ONT)		PacBio HiFi (HiFi)	
		Accession/Link	Coverage	Accession/Link	Coverage	Accession/Link	Coverage
T2T-CHM13v2.0	CHM13	SRR10035392	$\sim 17\times$	T2T_rel2 (link)	$\sim 50\times$	SRR11292120-23	$\sim 32.4\times$
				T2T_rel2 ($\geq 100\text{Kb}$)	$\sim 17\times$		
HG002v1.0.1	HG002	runs: 1-3, 17 (link)	$\sim 13.5\times$	-	-	link	$\sim 66\times$
GRCh38.p13	HG00733	28 runs (1 2 3 4)	$\sim 12\times$	-	-	link	$\sim 33\times$
	HG01346	SRR769453-54	$\sim 10\times$	-	-	SRR25618906	$\sim 26\times$

Table 3.1: Data sets for *H. sapiens*

Organism	Reference Genome	Illumina (NGS)		Oxford Nanopore (ONT)		PacBio HiFi (HiFi)	
		Accession/Link	Coverage	Accession/Link	Coverage	Accession/Link	Coverage
Thale cress (<i>A. thaliana</i>)	TAIR10.1	SRR28338978	$\sim 43\times$	-	-	ERR6210723	$\sim 50\times$
		CRR302669	$\sim 88\times$	CRR302667	$\sim 475\times$	CRR302668	$\sim 100\times$

Table 3.2: Data sets for *A. thaliana*

against four general-purpose state-of-the-art assemblers, namely hifiasm [26], LJA [8], HiCANU [96] and Verkko [105]) on five real *H. sapiens* (Table 3.1) and two real *A. thaliana* (Table 3.2) data sets and show that RAMbler outperforms them.

3.2 Methods

The initial algorithmic pipeline used in RAMbler as illustrated in Figure 2.6 was not developed for handling cycles in overlap graphs which are built in step E. While working with long and complex repetitive regions present in human genome, the overlap graphs may end up being cyclic. For successful repeat reconstruction using RAMbler with real data sets, we extended its step F of generating

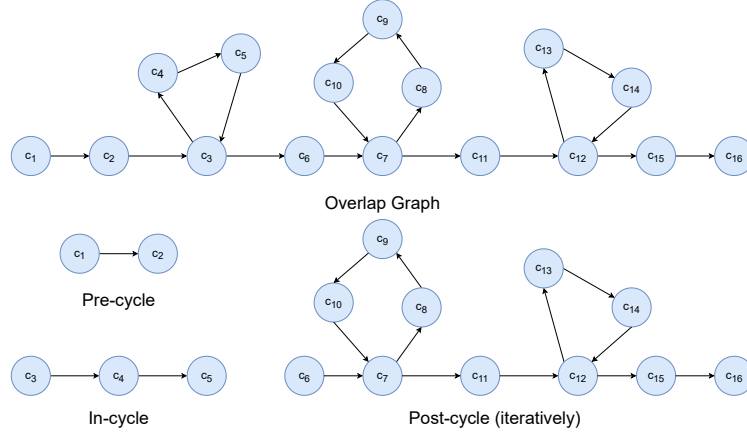


Figure 3.1: RAMbler extension for resolving cycles in an overlap graph

consensus assembly for handling cycles.

To compute the final assembly, RAMbler first determines whether the overlap graph is acyclic. If it is acyclic, RAMbler executes this step as described in Section 2.2.4. On the contrary, when the overlap graph has cycles, RAMbler partitions the graph into three components: an acyclic pre-cycle subgraph, the cycle itself, and a post-cycle subgraph (which could be cyclic), as shown in Figure 3.1. RAMbler repeats this process iteratively on the post-cycle subgraph until no cycles remain. Once the graph is completely decomposed in a set of acyclic subgraphs, RAMbler generates an assembly for each subgraph as described in the previous paragraph. RAMbler then enumerates all possible combinations of partial assemblies and selects the combination such that the sum of the individual assembly's length best matches the estimated length of the repetitive region.

Besides, we have added a component to the RAMbler pipeline that allows automatic integration of the RAMbler reconstructed repetitive regions back into the input draft genome.

3.3 Performance Metrics

We used CRAQ (Clipping information for Revealing Assembly Quality) to assess the assemblies produced by RAmbler [76]. CRAQ utilizes both NGS short reads and long reads for identifying and classifying errors in a given draft assembly. It reports assembly errors at different scales by transforming error counts into corresponding assembly quality indicators (AQIs) that reflect assembly quality at both regional and structural levels. CRAQ can distinguish between assembly errors and heterozygous loci based on the ratio of mapping coverage and the effective number of clipped reads; (1) Clip-based Regional Errors (CREs): If a region with clipped NGS reads is spanned by long reads with only SNP cluster features, and (2) Clip-based Structural Errors (CSEs): if the mapped long reads around a region with errors exhibit clipping features, i.e., the NGS reads simultaneously show clipping or no coverage.

3.3.1 Assembly quality index (AQI)

$$AQI = 100e^{-0.1N/L} \quad (3.1)$$

where N represents the cumulative normalized CRE or CSE count, and L indicates the total length of the assembly in the mega-base unit. Observe that a perfect assembly will yield an AQI score of 100. To avoid excessive impacts of specific regions enriched in errors (e.g, pericentromeric regions) on the overall AQI values, error counts were normalized within a sliding window of $0.0001 \times (\text{total assembly size})$.

3.3.2 Normalized error count in a window (N_w)

$$N_w = \sum_{i=1}^m i^{-1} \quad (3.2)$$

where m is the actual number of CRE/CSEs in the block. R-AQI and S-AQI can be calculated separately for CREs and CSEs.

3.4 Parameter Optimization

Recall that Table 2.3 summarizes the five main parameters of RAMbler where the value of mo (minimum overlap for the overlap graph) was set to 1,000 bp in all our experiments (both on synthetic and real data). In order to ensure that $mo = 1,000$ was sufficiently stringent to avoid spurious overlaps in the human genome, we carried out the following experiment. (i) We collected the centromeric regions for all human chromosomes in the T2T assembly (excluding the rDNA regions). (ii) We extracted 1 kb sequences every 100 bp in these centromeric regions. (iii) We computed an alignment between all pairs of sequences extracted in Step (ii) using minimap2 [75]. Out of 2,860,045 sequences generated in Step (ii), only three had an identical full match to another sequence in the set. There was (i) a 1 kb sequence on Chromosome 16 starting at position 33,712,533 matching a 1 kb sequence starting at position 35,108,033, (ii) a 1 kb sequence on Chromosome 16 starting at position 33,713,733 matching a 1 kb sequence starting at position 35,109,233, and (iii) a 1 kb sequence on Chromosome 3 starting at position 95,760,872 matching a 1 kb sequence starting at position 95,843,772. This analysis demonstrates that spurious overlaps that are 1 kb or longer are extremely rare in the human genome, even in repetitive regions like the human centromeres. We

used windows every 100 bp due to the computational cost aligning pairwise all these sequences. The minimap2 alignment file for 2.86 M sequences was 1.43 TB. The number of 1 kb sequence every 10 bp would be about 28.6 M, the number of 1 kb sequence every bp would be about 286 M. Since the alignment file grows quadratically with the number of sequences, it would quickly become infeasible to process. We believe that the conclusions would still hold with $10\times$ or $100\times$ more data.

3.5 Result Analysis

3.5.1 Experimental Results on *Homo sapiens*

We used three human genome assemblies as a reference, namely GRCh38.p13 (hereafter called HG38), T2T-CHM13.v2.0 (T2T), and the maternal strand of HG002 (MAT002). We used PacBio HiFi reads from four different cell lines, namely CHM13 (from the T2T project), HG002, HG00733, and HG01346 (these last three from the human pan-genome project [77]). Table 3.1 reports the accession numbers and statistics for these human data sets, while Table 3.2 summarizes accession numbers and statistics for *A. thaliana*.

We used RAmbler to assemble some of the repetitive regions in the reference assemblies HG38, T2T and MAT002, using various sets of HiFi reads. We used the following naming convention to identify the assemblies produced by RAmbler. Hereafter, a RAmbler assembly using reference assembly G and HiFi reads H will be denoted by $RA.G.H$. For example, the RAmbler assembly using the HG38 assembly and the HG00733 HiFi reads will be called $RA.HG38.HG00733$.

We focused on five complex repetitive regions within the human genome: the centromeres of Chromosome 8 and X, and three non-centromeric regions from Chromosome 19. While the selection of these regions was somewhat arbitrary, it was motivated by a few factors: (1) these

regions were over-collapsed in the HG38 assembly (Figure 3.2), (2) Chromosomes 8, 19, and X have no unplaced contigs within the HG38 assembly, and (3) Chromosome 19 contains a few unresolved non-centromeric repeats (Figure 3.3). Our intent was to show that using newer HiFi reads, RAMbler could improve some of the over-collapsed repetitive regions, in particular in the HG38 assembly.

We carried out five experiments on human data sets. The first two experiments were aimed at demonstrating RAMbler’s ability to resolve repeats without manual curation. For these two experiments, we used as inputs (1) the T2T assembly with CHM13 HiFi reads, and (2) the MAT002 assembly with HG002 HiFi reads. The other three experiments focused on RAMbler’s performance. For these three, we used as input the HG38 assembly and HiFi reads from (3) HG002, (4) HG00733, and (5) HG01346.

We carried out these five experiments on the centromeric regions of Chromosomes 8 and X, as well as three non-centromeric regions on Chromosome 19. We ran RAMbler, hifiasm, LJA, HiCANU, and Verkko on the subset of HiFi reads mapped to these regions. SDA was excluded because it is obsolete for reasons explained in Chapter 1. CentroFlye was also excluded because we were unable to run it on our 2.8 GHz 32-core processor server with 512GB of RAM. We tried first to reconstruct the centromere of Chromosome X using the HiFi reads and the HORs provided by the authors, but CentroFlye failed during the error-correction step. When we tried to reconstruct the same centromere using the CHM13 ONT reads from [16], CentroFlye ran out of memory. The instructions claim that CentroFlye requires about 800GB of RAM to complete the assembly of Chromosome X.

The comparative results for RAMbler, hifiasm, LJA, HiCANU, and Verkko on the cen-

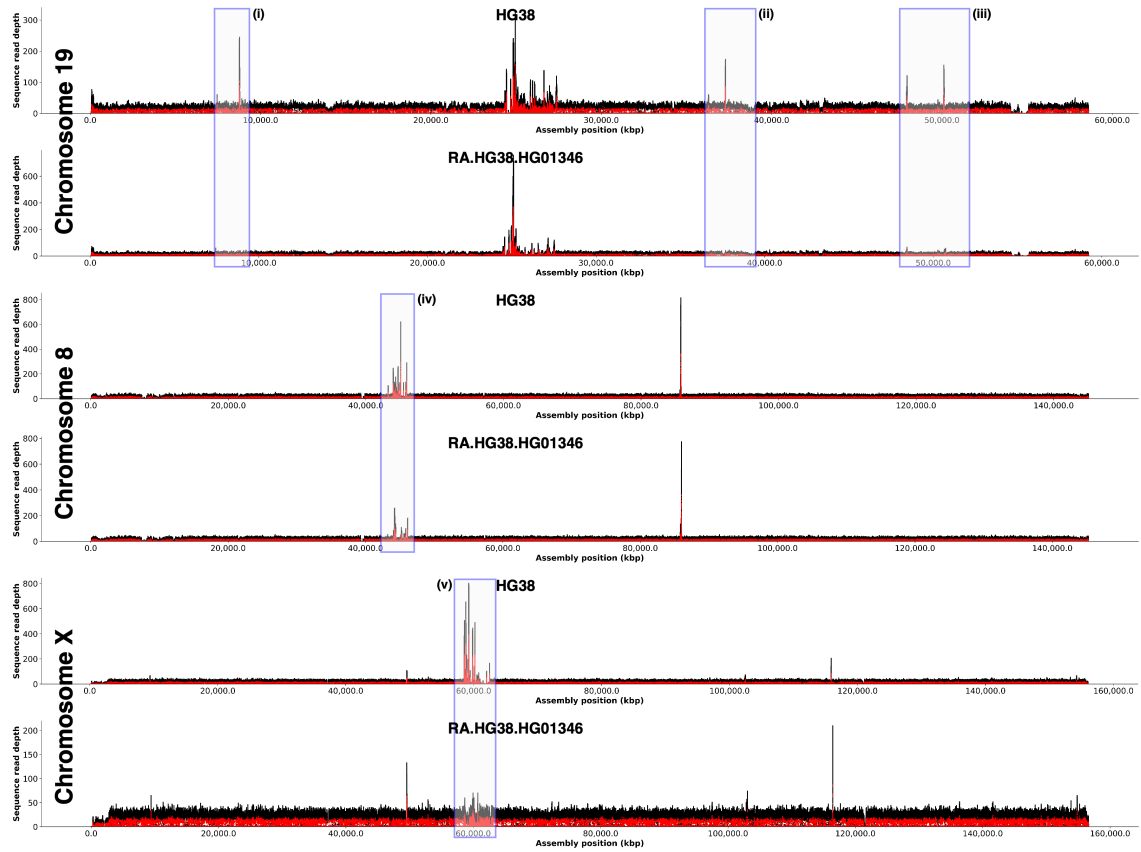


Figure 3.2: NucFreq plots illustrating HiFi read mapping coverage for the comparison of RA.HG38.HG01346 assembly against GRCh38.p13 (HG38) of human Chromosomes 8, 19, and X; in all plots, blocks (i), (ii), and (iii) are non-centromeric repeats in Chromosome 19; blocks (iv) and (v) are the centromeric regions of Chromosomes 8 and X, respectively

Chr	Input assembly	HiFi reads	Depth	Repetitive region		# Selected reads	Selected reads sum (Mb)	Expected size (Mb)		Assembler				
				start (Mb)	end (Mb)					Rambler	hifiasm	LJA	HiCANU	Verkko
8	HG38	HG002	67.5×	43.5	46.5	15,620	249.038	3.689	# contigs	2	3	39	-	66
									Total (bp)	3,885,290	3,632,973	7,573,022		8,186,622
									Longest contig	3,627,559	1,843,336	2,701,877		1,374,826
		HG00733	32.8×	43.5	46.5	7,241	99.356	3.029	# contigs	1	3	74	44	103
									Total (bp)	3,805,715	3,692,973	6,653,621	6,279,128	7,267,808
									Longest contig	3,805,715	3,483,042	1,384,931	1,002,430	1,710,467
		HG01346	26×	42.0	48.0	9,579	177.120	6.812	# contigs	3	5	91	73	134
									Total (bp)	7,553,039	6,637,843	12,660,338	13,840,156	13,671,832
									Longest contig	6,229,877	6,213,919	4,017,234	6,043,945	2,981,689
	T2T	CHM13	32.4×	43.5	46.5	5,591	101.370	3.129	# contigs	1	1	1	10	1
									Total (bp)	3,040,965	3,040,965	3,040,948	3,343,237	3,032,370
									Longest contig	3,040,965	3,040,965	3,040,948	3,033,722	3,032,370
	MAT002	HG002	66×	43.5	47.0	14,393	229.495	3.477	# contigs	1	3	26	-	51
									Total (bp)	3,271,813	3,309,709	6,861,347		7,415,002
									Longest contig	3,271,813	1,776,606	2,701,877		1,374,826
X	HG38	HG002	34.2×	57.5	63.0	9,665	155.687	4.552	# contigs	1	1	1	1	N/A
									Total (bp)	4,678,725	4,678,719	4,663,887	4,669,944	
									Longest contig	4,678,725	4,678,719	4,663,887	4,669,944	
		HG00733	31.8×	57.25	64.0	15,333	209.268	6.581	# contigs	2	3	114	103	170
									Total (bp)	5,653,597	5,279,187	12,115,053	13,853,128	13,649,195
									Longest contig	3,813,195	3,408,909	1,675,295	2,970,002	3,984,187
		HG01346	25.5×	57.25	64.0	9,679	179.004	7.020	# contigs	1	5	74	92	140
									Total (bp)	7,378,373	7,072,503	12,305,066	14,376,243	13,718,592
									Longest contig	7,378,373	3,232,572	4,247,266	3,280,524	4,251,271
	T2T	CHM13	34×	57.5	62.0	8,488	153.883	4.526	# contigs	1	2	3	22	10
									Total (bp)	4,468,109	4,533,307	4,536,587	5,081,155	4,575,146
									Longest contig	4,468,109	4,486,438	4,074,724	2,016,666	1,940,381
	MAT002	HG002	33×	57.5	62.0	9,362	150.793	4.569	# contigs	2	2	3	-	4
									Total (bp)	4,566,785	4,566,777	4,542,009		4,546,175
									Longest contig	4,542,010	4,542,004	4,180,812		4,141,556

Table 3.3: Comparison of the assemblies produced by Rambler, hifiasm, LJA, HiCANU and Verkko on the HiFi reads extracted around the centromeres of human Chromosomes 8 and X, for several choices of input assemblies and HiFi reads. Note: ‘-’ means that the assembler timed out after 96 hours, ‘N/A’ means that the assembler finished execution but produced no output, numbers in bold indicate the assemblies with the fewest contigs in each row

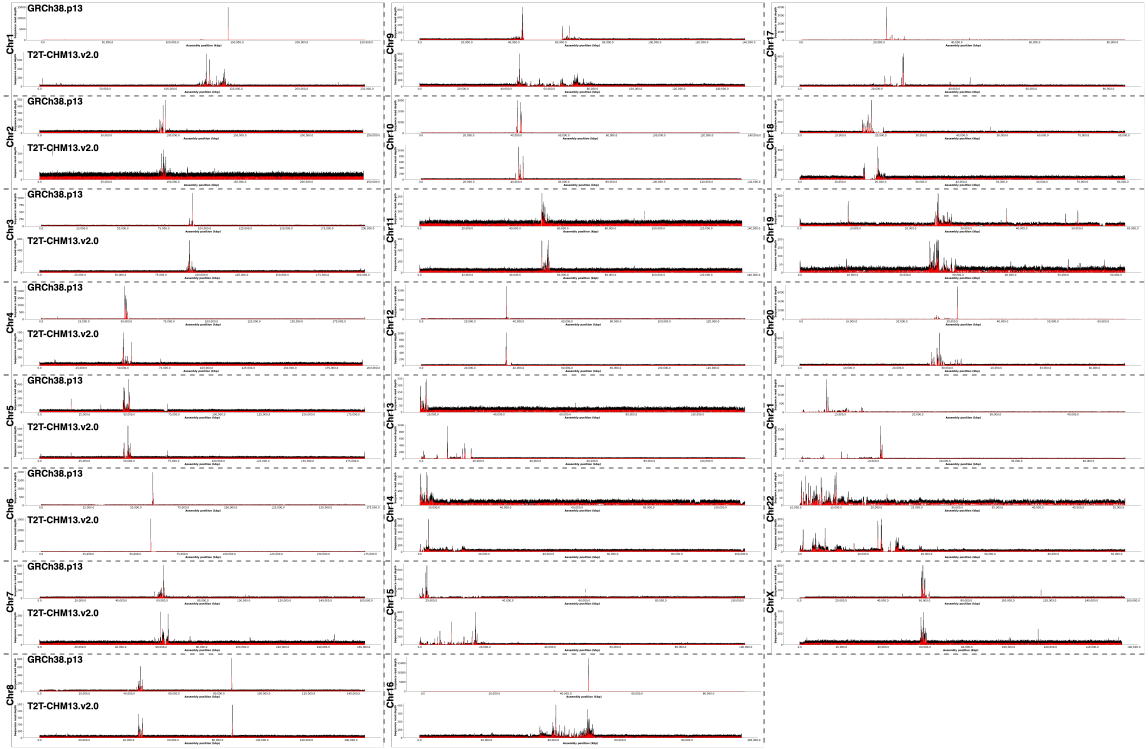


Figure 3.3: Mapping coverage for HiFi HG01346 reads ($\sim 26\times$ coverage) on all chromosomes in GRCh38.p13 (HG38) and T2T-CHM13.v2.0 (T2T)

tromeric regions of Chromosomes 8 and X are summarized in Table 3.3. Columns 2, 3, and 4 indicate the input assembly, the HiFi sample, and the chromosome-level sequencing depth, respectively. Columns 5 and 6 indicate the boundaries of the regions to be re-assembled. Columns 7 and 8 indicate the number of HiFi reads mapped to the target region and the total number of bases in these reads, respectively. In Column 9, we show RAMbler’s estimate of the size of the repetitive region, obtained by computing the ratio between the total number of bases of the selected HiFi reads and the average chromosome-level coverage depth. While it has been reported that repetitive regions can affect HiFi sequencing coverage [95], we determined that this coverage bias would have a small impact on RAMbler’s estimation of the length of the repetitive region to be assembled.

When we masked all repetitive regions from the human genome, the changes in HiFi coverage was relatively small. Table 3.6 shows that on Chromosomes 8, 19 and X, the change in coverage after repeat masking was smaller than 4%. Overall, the variations in coverage were under 9% at the chromosome-level and under 3% at the genome level. The largest change was observed for Chromosome 17 (8.57% variation in coverage using the HG002 HiFi reads). Such a coverage fluctuation would mean that, in the worst case, a repetitive region with ten copies of the monomer could be underestimated by one copy. Columns 11-15 show the assembly statistics for RAMbler, hifiasm, LJA, HiCANU, and Verkko. Observe that RAMbler consistently produced the least fragmented assemblies, and in most cases the longest contig produced by RAMbler was the longest among all assemblers. In many cases the total assembly size generated by RAMbler was close to the expected assembly size, although hifiasm also produced assemblies whose total size was also consistent with the expectation. Similar observations could be made for the three non-centromeric regions on Chromosome 19 (Table 3.4 for T2T and MAT002, Table 3.5 for HG38).

Unlike *de novo* assemblers which consistently produce the same outputs for the same set of input reads, RAMbler's assembly depends on the reference used as it relies on the input assembly to identify the reads that belong to a repetitive region. Observe that in Table 3.3 we used the same set of HiFi reads (HG002) with two reference assemblies, namely HG38 and MAT002. Despite the difference between HG38 and MAT002, in particular in the centromeric regions, the analysis in Table 3.7 shows that more than 85% of the selected reads were shared. As a result, the RAMbler assemblies are relatively consistent. For instance, on the centromere of Chromosome X, the length difference for the longest contigs was about 3%. On the centromere of Chromosome 8, the length

Input assembly	HiFi reads	Coverage	Repetitive region		# Selected reads	Selected reads sum (Mb)	Expected size (Mb)		Assembler				
			start (Mb)	end (Mb)					RAmbler	hifiasm	LJA	HiCANU	Verkko
T2T	CHM13	30.5×	8	10	3,428	61.473	2.016	# contigs	1	1	1	8	1
								Total (bp)	2,050,034	2,050,034	2,040,669	2,294,767	2,047,043
								Longest contig	2,050,034	2,050,034	2,040,669	2,047,040	2,047,043
			38.5	41.5	5,154	92.475	3.032	# contigs	1	1	1	10	4
								Total (bp)	3,042,992	3,042,992	3,042,941	3,309,519	3,125,757
								Longest contig	3,042,992	3,042,992	3,042,941	3,037,943	2,793,414
			50	54	6,422	114.784	3.763	# contigs	1	1	1	7	1
								Total (bp)	4,051,224	4,051,222	4,051,225	4,251,698	4,051,206
								Longest contig	4,051,224	4,051,222	4,051,225	4,044,012	4,051,206
MAT002	HG002	60×	8	10	7,709	123.020	2.050	# contigs	1	3	14	15	23
								Total (bp)	2,082,651	2,149,253	4,034,730	4,275,016	4,213,623
								Longest contig	2,082,651	2,082,652	734,364	2,002,029	751,170
			38.5	41.5	11,249	180.077	3.001	# contigs	1	2	21	14	21
								Total (bp)	3,029,470	3,077,117	5,880,214	6,065,751	5,925,770
								Longest contig	3,029,470	3,041,587	1,395,837	3,023,701	1,581,545
			50	54	13,854	220.561	3.676	# contigs	1	1	23	17	41
								Total (bp)	4,040,445	4,040,444	7,849,165	8,056,324	8,090,995
								Longest contig	4,040,445	4,040,444	1,721,763	4,035,820	1,602,573

Table 3.4: Assembly statistics for RAmbler, hifiasm, LJA, HiCANU and Verkko on three non-centromeric repeats of Chromosome 19 from the T2T-CHM13.v2.0 assembly (T2T) and the maternal HG002 assembly (MAT002) using the corresponding HiFi reads as inputs

Input assembly	HiFi reads	Coverage	Repetitive region		# Selected reads	Selected reads sum (Mb)	Expected size (Mb)		Assembler				
			start (Mb)	end (Mb)					RAmbler	hifiasm	LJA	HiCANU	Verkko
HG38	HG002	61×	8	10	8,183	130.625	2.141	# contigs	1	1	14	12	23
								Total (bp)	2,203,257	2,225,655	4,344,711	4,496,850	4,509,590
								Longest contig	2,203,257	2,225,655	770,032	2,223,194	780,635
			36	38	8,657	138.825	2.276	# contigs	1	1	9	5	14
								Total (bp)	2,272,627	2,272,627	4,386,424	4,502,066	4,586,729
								Longest contig	2,272,627	2,272,627	1,169,520	1,355,295	1,346,483
			47	51	14,573	232.556	3.812	# contigs	1	2	31	22	53
								Total (bp)	4,283,959	4,313,161	8,347,481	8,639,631	8,658,198
								Longest contig	4,283,959	4,294,647	1,651,643	3,958,242	1,565,100
	HG00733	33×	8	10	5,236	70.550	2.138	# contigs	1	1	52	33	72
								Total (bp)	2,214,533	2,214,533	4,221,660	4,370,351	4,541,272
								Longest contig	2,214,533	2,214,533	672,674	2,104,790	538,893
			36	38	5,529	74.748	2.265	# contigs	1	1	27	48	76
								Total (bp)	2,292,968	2,292,968	3,552,638	4,493,447	4,362,704
								Longest contig	2,292,968	2,292,968	697,275	2,287,243	320,798
			47	51	10,283	137.920	4.179	# contigs	2	12	32	31	62
								Total (bp)	4,197,958	7,150,118	8,335,398	8,595,284	8,670,120
								Longest contig	3,217,913	4,215,911	843,864	4,279,895	783,679
	HG01346	24.5×	8	10	2,926	53.411	2.180	# contigs	1	1	25	18	31
								Total (bp)	2,190,205	2,198,110	4,198,112	4,460,687	4,395,359
								Longest contig	2,190,205	2,198,110	705,013	1,538,054	580,887
			36	38	2,992	55.055	2.247	# contigs	1	1	21	20	42
								Total (bp)	2,303,078	2,303,078	3,744,185	4,229,499	4,120,267
								Longest contig	2,303,078	2,303,078	531,893	2,296,104	451,056
			47	51	5,216	95.285	3.889	# contigs	1	2	36	28	64
								Total (bp)	4,244,920	4,329,037	8,412,138	8,695,951	8,722,578
								Longest contig	4,244,920	4,271,958	793,309	4,069,987	656,797

Table 3.5: Assembly statistics for RAmbler, hifiasm, LJA, HiCANU and Verkko on three non-centromeric repeats of Chromosome 19 using HiFi reads HG002, HG00733, HG01346 and GRCh38.p13 (HG38) as input assembly

Chromosome statistics			HiFi reads: HG002			HiFi reads: HG00733			HiFi reads: HG01346		
Chr	Length (bp)	Length of SDs (bp)	Coverage	Coverage (SDs masked)	Change (%)	Coverage	Coverage (SDs masked)	Change (%)	Coverage	Coverage (SDs masked)	Change (%)
1	248,956,422	14,814,748	65.75	65.30	0.68	34.07	32.88	3.49	25.83	25.28	2.13
2	242,193,529	10,341,536	69.02	70.67	2.39	33.97	34.75	2.30	26.72	27.30	2.17
3	198,295,559	3,642,738	69.89	71.52	2.33	34.06	34.83	2.26	26.66	27.24	2.18
4	190,214,555	5,059,972	72.27	72.82	0.76	34.49	34.17	0.93	27.66	27.54	0.43
5	181,538,259	5,948,025	70.77	72.84	2.92	33.94	35.01	3.15	26.8	27.61	3.02
6	170,805,979	4,071,190	69.6	71.19	2.28	33.11	33.90	2.39	26.20	26.81	2.33
7	159,345,973	13,686,825	67.68	71.05	4.98	33.6	35.18	4.70	26.37	27.65	4.85
8	145,138,636	3,143,789	67.61	68.10	0.72	32.78	33.07	0.88	25.94	26.18	0.93
9	138,394,717	13,547,270	59.75	60.90	1.92	30.15	30.57	1.39	23.66	24.00	1.44
10	133,797,422	7,574,751	69.74	70.85	1.59	35.86	35.52	0.95	27.60	27.58	0.07
11	135,086,622	5,719,550	67.50	69.48	2.93	34.01	35.05	3.06	26.39	27.18	2.99
12	133,275,309	3,027,056	68.23	70.01	2.61	33.89	34.78	2.63	26.21	26.81	2.29
13	114,364,328	3,363,038	59.80	61.53	2.89	28.27	29.03	2.69	22.60	23.25	2.88
14	107,043,718	2,952,467	57.97	58.58	1.05	28.58	28.95	1.29	22.24	22.62	1.71
15	101,991,189	8,669,292	52.15	54.87	5.22	26.70	27.82	4.19	20.68	21.73	5.08
16	90,338,345	9,086,009	62.23	64.38	3.45	34.45	35.55	3.19	26.25	27.37	4.27
17	83,257,441	7,711,895	61.83	67.13	8.57	33.34	35.81	7.41	24.94	26.58	6.58
18	80,373,285	1,938,507	67.91	69.06	1.69	32.71	33.28	1.74	25.57	26.00	1.68
19	58,617,616	3,985,208	61.11	62.98	3.06	33.61	34.70	3.24	24.35	25.23	3.61
20	64,444,167	2,894,651	71.59	74.02	3.39	37.99	36.60	3.66	28.73	27.61	3.90
21	46,709,983	6,163,779	62.94	61.22	2.73	31.45	29.44	6.39	24.88	23.52	5.47
22	50,818,468	5,996,890	48.68	47.69	2.03	26.12	26.22	0.38	20.20	19.93	1.34
X	156,040,895	10,321,878	34.20	35.21	2.95	31.75	32.66	2.87	25.52	26.29	3.02
Total	3,031,042,417	153,661,064	64.73	66.19	2.26	32.56	33.03	1.44	25.3	25.71	1.62

Table 3.6: Changes in HiFi reads' coverage for each Chromosome when segmental duplications (SD) are masked

Chr	Input Assembly	Region	# Selected reads	# Common reads	Common (%)
8	HG38	Centromere	15,620	14,392	92.14
	MAT002		14,393		99.99
X	HG38		9,665	8,217	85.02
	MAT002		9,362		87.77
19	HG38	8 Mb - 10 Mb	8,183	7,548	92.24
	MAT002	8 Mb - 10 Mb	7,709		97.91
	HG38	36 Mb - 38 Mb	8,657	7,453	86.09
	MAT002	38.5 Mb - 40.5 Mb	7,775		95.86
	HG38	47 Mb - 51 Mb	14,573	12,454	85.46
	MAT002	50 Mb - 54 Mb	13,854		89.89

Table 3.7: Comparing the number of selected HiFi HG002 reads when the GRCh38.p13 assembly (HG38) or the maternal HG002 assembly (MAT002) were used as the input reference assembly to RAmbler

difference for the longest contigs was approximately 10%.

While the assembly contiguity statistics reported in Tables 3.3 to 3.5 are important, they did not tell us whether these assemblies had high quality. In order to quantitatively assess the assembly quality, we used a two-pronged approach. First, we used the manually-curated T2T assembly as the ground truth and we measured the agreement between the RAmbler assemblies and the T2T assembly using QUAST [48] and SyRI [45]. This approach, however, can be problematic because

the HiFi reads used for the RAmbler assemblies did not originate from the cell line used to generate the T2T assembly. Human centromeres are known to exhibit high variation across individuals [4]. For instance, the NucFreq coverage plots in Figure 3.3 show coverage spikes in the centromeric regions of the T2T assembly when the HG01346 reads were mapped to it. As a consequence some level of divergence would be expected in RAmbler’s centromeric assembly when compared to the T2T assembly. In order to obtain both qualitative and quantitative assessments of RAmbler’s improvements, we measured the assembly quality using CRAQ [76] and NucFreq [128]. CRAQ is a tool that measures assembly quality without the need of a reference; in particular, it can detect local and global assembly errors based on the alignment information of mapped reads (long and short). CRAQ was given in input Illumina reads (which were not used in the assembly) and HiFi reads (which were used for the assembly). While we expected a reduction in coverage spikes in the repetitive regions of the NucFreq plots because those HiFi reads were also used in the assembly, we were interested in measuring the extent of this reduction. Additionally, we used NucFlag (an extension of NucFreq) to flag assembly errors or collapses along with the coverage plots. Note that the alignments of centromeres in Figure 3.4 shows that the centromeres are missing in the optical maps (GM08331). Since the centromeres are highly repetitive and lack the nicking/restriction enzyme sites, the optical map assembly pipeline over-compresses those regions. Hence optical maps cannot be used to validate the assembly quality at the centromeres, or in other complex repetitive region.

To establish RAmbler’s assembly quality, we carried out two sets of experiments. In the first set, we show that RAmbler can drastically improve the HG38 assembly in the five repetitive regions to a QAST-based quality comparable to the T2T assembly using HG002, HG00733, or HG01346 HiFi reads. In the second set, we show that if RAmbler was given the same set of HiFi

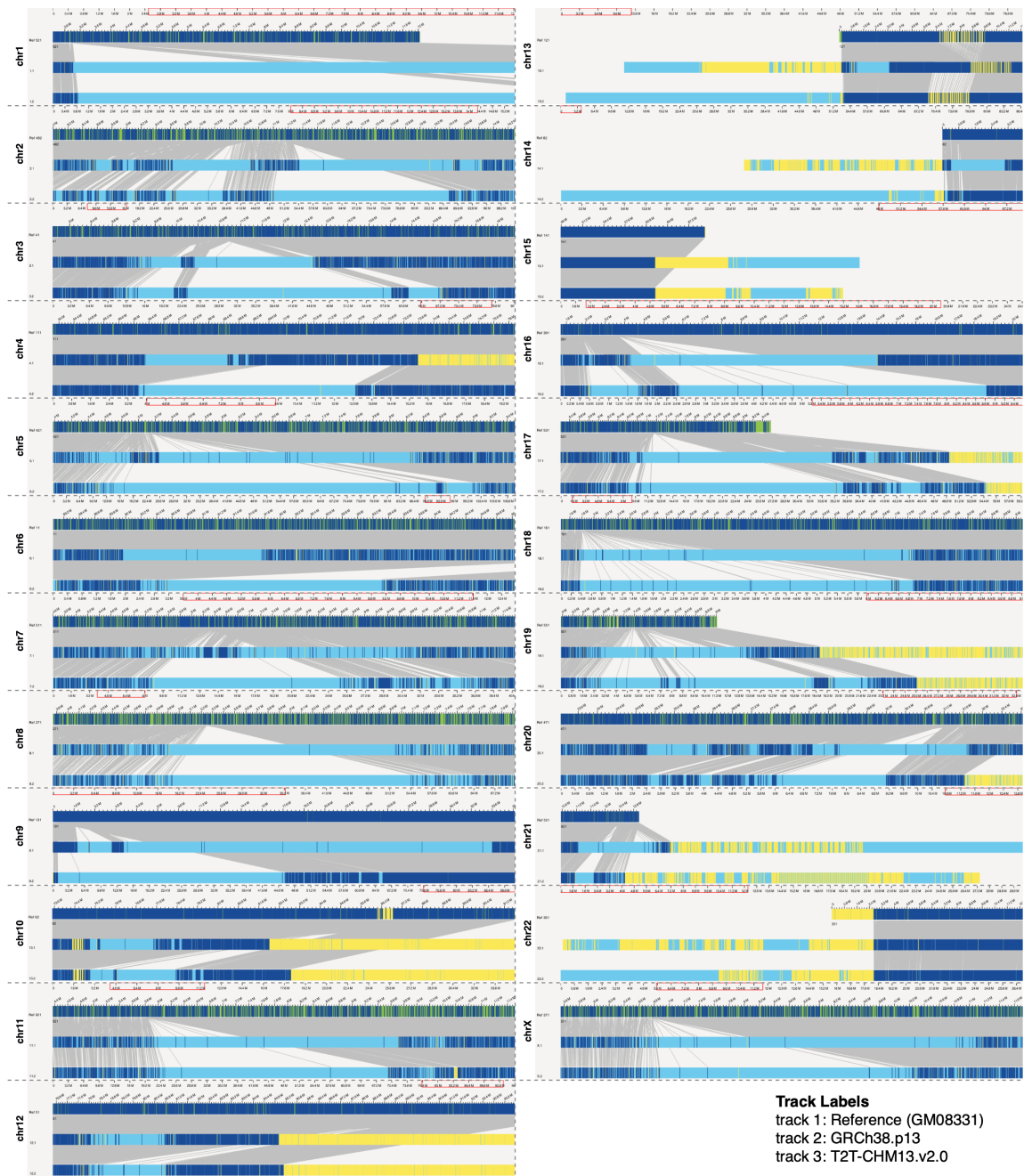


Figure 3.4: Alignment of the centromeres of GRCh38.p13 (HG38) and T2T-CHM13.v2.0 (T2T) against the reference optical map GM08331

reads used for the T2T or the MAT002 assemblies, it would produce an assembly of the five repetitive regions that matches, if not exceeds, the quality of the corresponding reference assemblies.

Figure 3.5A-B summarizes the results of one run from the first set of experiments using HG01346 HiFi reads. Figure 3.5A shows the HiFi coverage. The coverage depth in Figure 3.5A is clipped at $60\times$ for better visualization, whereas Figure 3.2 shows full-scale plots for HG38 and the RAmbler assembly RA.HG38.HG01346. Observe that on Chromosome 19, RA.HG38.HG01346 has a more uniform coverage across the three repetitive regions compared to HG38, in particular for regions (i) and (ii). Also observe that (1) while the coverage on region (iv) of Chromosome 8 of HG38 is $\sim 600\times$, it reduces to $\sim 100\times$ in RA.HG38.HG01346 indicating a partial resolution of the repeat; (2) while the coverage on region (v) of Chromosome X of HG38 is $\sim 800\times$, it reduces to $\sim 30\times$ in RA.HG38.HG01346 indicating a full resolution of the repeat. From a purely coverage viewpoint, these results indicate that RA.HG38.HG01346 is significantly improved compared to HG38. Figure 3.6A illustrates the NucFlag plots for the five regions analyzed. Observe that NucFlag indicates much fewer errors in the RA.HG38.HG01346 assembly compared to HG38.

Figure 3.5B shows the results of the CRAQ analysis [76]. We used HG01346 HiFi reads and HG01346 Illumina reads ($\sim 10\times$ coverage) to evaluate the quality of HG38 and RA.HG38.HG01346. The Circos plots [72] in Figure 3.5B illustrate the (a) assembly quality indicators (AQI) score (blue curve, higher is better, see Section “Performance Metrics for Real Data”, (b) base errors (green bands, fewer is better), and (c) misjoins (red bands, fewer is better). Cones (i)-(v) highlight the repetitive regions. On Chromosome 19, observe that (1) the assembly quality index (AQI) score for RA.HG38.HG01346 is higher than the AQI score for HG38 on all three regions; (2) RA.HG38.HG01346’s base errors are significantly better than HG38; (3) CRAQ reports four mis-

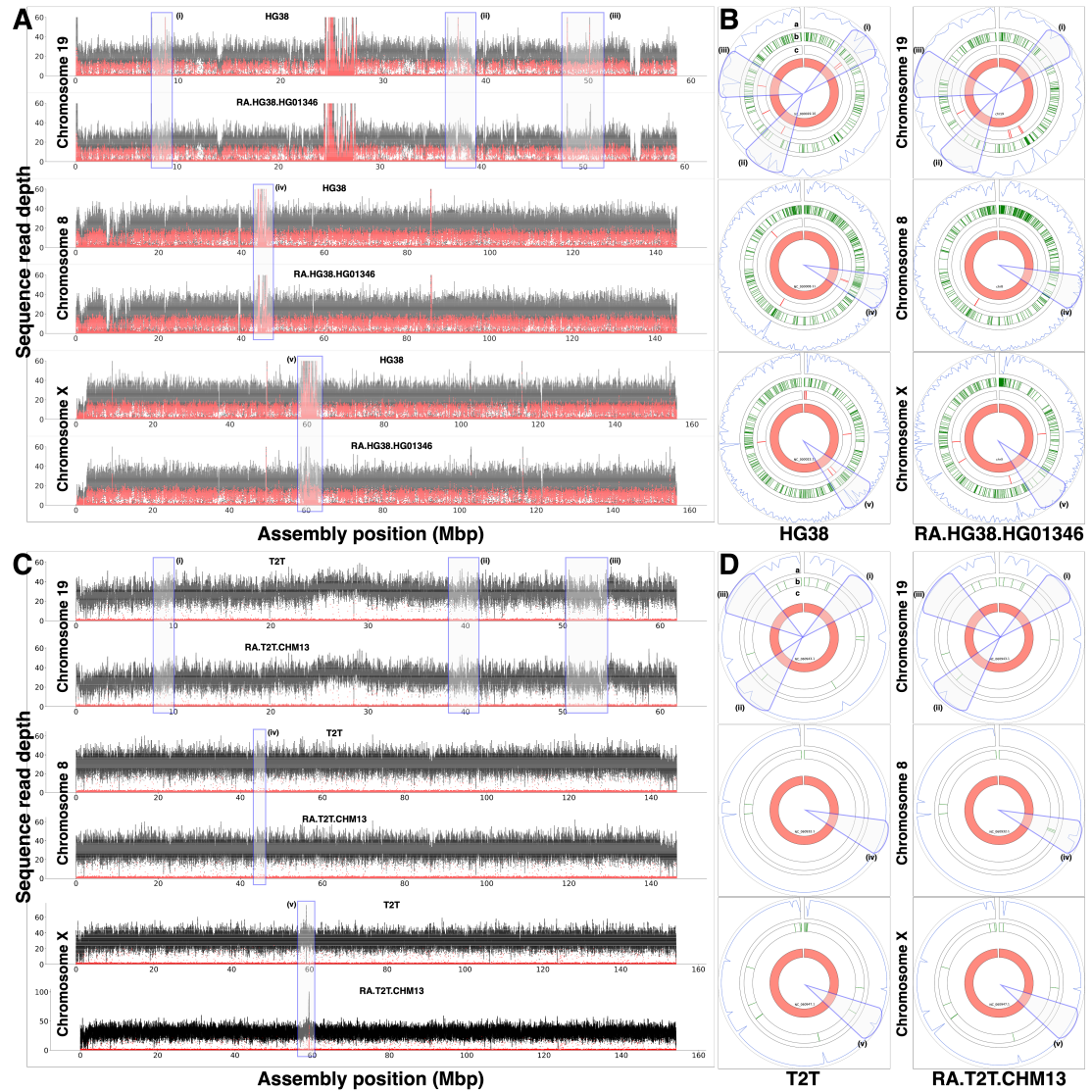


Figure 3.5: Comparing Rambler's assemblies (RA.HG38.HG01346 and RA.T2T.CHM13) against (A, B) the GRCh38.p13 assembly (HG38) and (C, D) the T2T-CHM13.v2.0 assembly (T2T) of human Chromosomes 8, 19, and X; in all plots, blocks (i), (ii), and (iii) are non-centromeric repeats in Chromosome 19; blocks (iv) and (v) are the centromeric regions of Chromosomes 8 and X, respectively; (A, C) NucFreq plots illustrating HiFi read mapping coverage (clipped at 60x); (B, D) CRAQ Circos plots illustrating (a) assembly quality index (AQI) score (higher is better), (b) base errors (fewer is better), and (c) misjoins (fewer is better)

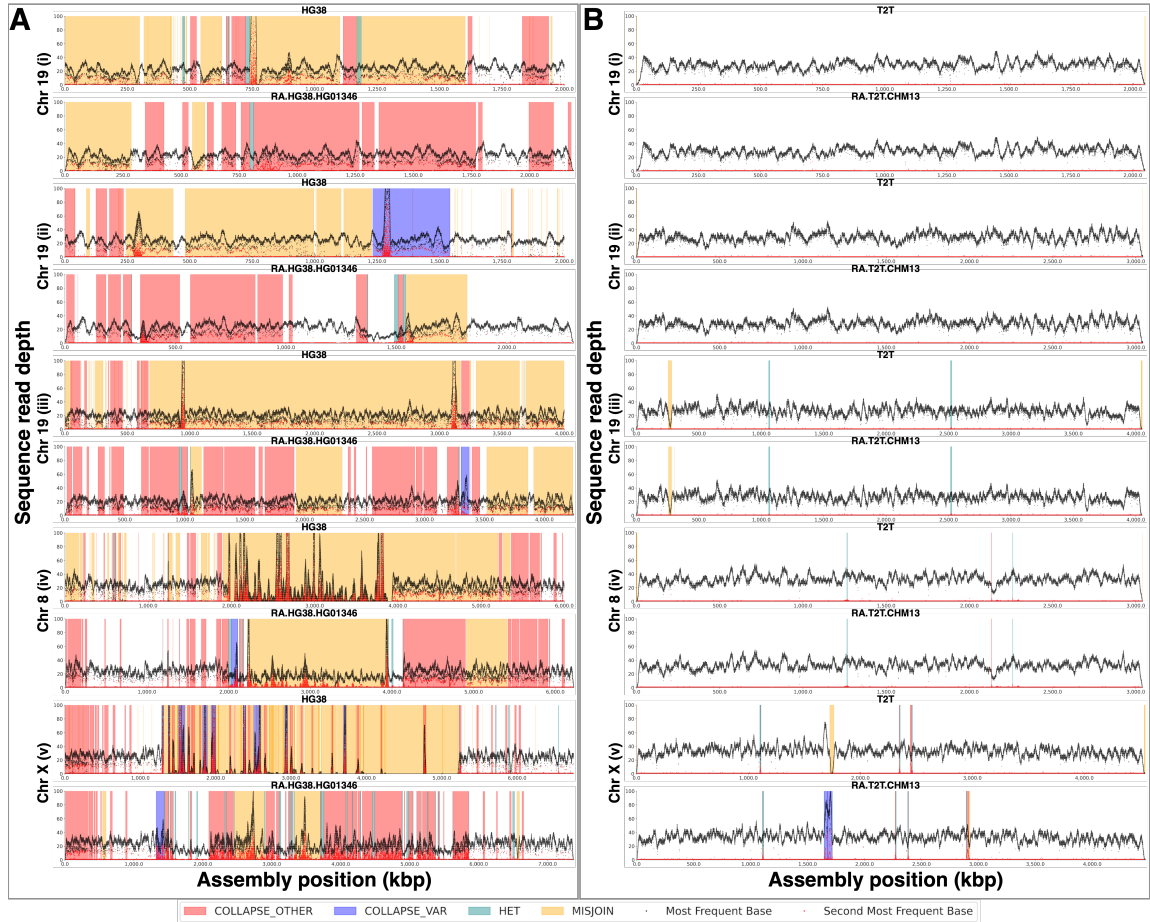


Figure 3.6: NucFlag plots illustrating HiFi read mapping coverage for the comparison of (A) RA.HG38.HG01346 assembly against GRCh38.p13 (HG38) and (B) RA.T2T.CHM13 assembly against T2T-CHM13.v2.0 (T2T) of human Chromosomes 8, 19, and X; in all plots, blocks (i), (ii), and (iii) are non-centromeric repeats in Chromosome 19; blocks (iv) and (v) are the centromeric regions of Chromosomes 8 and X, respectively

joins in HG38, and zero in RA.HG38.HG01346. Also observe that on the centromeres of Chromosome 8 and Chromosome X, the AQI scores for RA.HG38.HG01346 are higher than the AQI scores for HG38. On Chromosome 8, HG38 has two misjoins, and RA.HG38.HG01346 has none. RA.HG38.HG01346 also has fewer base errors. On Chromosome X, RA.HG38.HG01346 has no misjoins (HG38 has two) with fewer base errors than HG38. The NucFreq, the Circos/CRAQ and the NucFlag plots for HiFi reads HG002 and HG00733 are shown in Figures 3.7 and 3.8. The coverage plots and the Circos plots from both of these runs demonstrate similar results as shown in Figure 3.5A-B. Table 3.8 reports all the statistics produced by CRAQ, which clearly indicates that RA.HG38.HG002, RA.HG38.HG00733 and RA.HG38.HG01346 are improved compared to HG38.

Table 3.9 reports the QUAST metrics for HG38, RA.HG38.HG002, RA.HG38.HG00733 and RA.HG38.HG01346, using the T2T assembly as the ground truth and for HG38 and RA.HG38.HG002, using the MAT002 assembly as the ground truth. When T2T was used as the reference, (1) the number of misassemblies on Chromosome 8 decreased from 371 in HG38 to 239-278 in the RAmbler assemblies; on Chromosome 19, they decreased from 101 in HG38 to 92-100; on Chromosome X, they decreased from 374 in HG38 to 205-250, (2) the genome fraction for Chromosome 19 increased from about 90.7% to about 91.5%; on Chromosome X, it increased from about 98.6% to about 99%; on Chromosome 8, it remained the same around 98%. When MAT002 was used as the reference, (1) the number of misassemblies on Chromosome 8 decreased from 382 in HG38 to 205 in RA.HG38.HG002; on Chromosome 19, they reduced to 102 from 119; on Chromosome X, they decreased from 382 in HG38 to 169, (2) the genome fraction for Chromosome 8 increased from 97.5% to 98.4%; on Chromosome 19, it improved from 91.2% to 92.3%; on Chromosome X, it in-

Chromosome	Metric	HG002		HG00733		HG01346	
		HG38	RA.HG38.HG002	HG38	RA.HG38.HG00733	HG38	RA.HG38.HG01346
19	Coverage (%)	90.57	94.25	89.95	94.16	90.32	94.05
	R-AQI	77.33	78.03	75.96	76.45	67.66	68.92
	S-AQI	81.29	82.12	92.69	89.81	89.29	91.41
8	Coverage (%)	92.96	98.19	92.47	97.34	92.41	97.72
	R-AQI	82.11	83.79	80.90	83.81	79.16	79.22
	S-AQI	89.48	92.60	94.21	96.54	98.52	98.60
X	Coverage (%)	94.02	98.31	93.91	97.95	94.06	97.71
	R-AQI	81.52	81.76	86.66	86.17	83.60	82.85
	S-AQI	92.15	93.66	94.69	94.86	96.32	97.42

Table 3.8: Comparing Rambler’s assemblies (RA.HG38.HG002, RA.HG38.HG00733, and RA.HG38.HG01346) for selected regions of human Chromosomes 19, 8, and X against the GRCh38.p13 assembly (HG38) using CRAQ; CRAQ reports assembly coverage rate (i.e., the fraction of the genome assembled), Regional Assembly Quality Indicator (R-AQI) score (higher is better) and Structural Assembly Quality Indicator (S-AQI) score (higher is better); CRAQ was provided Illumina reads ($\sim 13.5\times$ coverage) and long PacBio HiFi reads ($\sim 66\times$ coverage) for HG002; Illumina reads ($\sim 12\times$ coverage) and long PacBio HiFi reads ($\sim 33\times$ coverage) for HG00733; Illumina reads ($\sim 10\times$ coverage) and long PacBio HiFi reads ($\sim 26\times$ coverage) for HG01346; numbers in bold indicate the best scores

Chromosome	Reference	Query	Metric		
			Genome fraction	Normalized NG50 (η)	# of misassemblies
8	T2T	RA.T2T.CHM13	100.000	1.000	0
		HG38	97.971	0.992	371
		RA.HG38.HG002	97.779	0.996	239
		RA.HG38.HG00733	98.303	0.998	278
		RA.HG38.HG01346	97.813	0.994	239
	MAT002	RA.MAT002.HG002	99.482	0.998	49
		HG38	97.532	0.989	382
		RA.HG38.HG002	98.379	0.993	205
19	T2T	RA.T2T.CHM13	100.000	1.000	0
		HG38	90.706	0.950	101
		RA.HG38.HG002	91.480	0.960	92
		RA.HG38.HG00733	91.458	0.961	98
		RA.HG38.HG01346	91.561	0.960	100
	MAT002	RA.MAT002.HG002	99.940	1.000	3
		HG38	91.233	0.956	119
		RA.HG38.HG002	92.275	0.966	102
X	T2T	RA.T2T.CHM13	99.950	1.000	2
		HG38	98.611	1.012	374
		RA.HG38.HG002	99.126	1.006	205
		RA.HG38.HG00733	98.849	1.004	213
		RA.HG38.HG01346	98.960	1.015	250
	MAT002	RA.MAT002.HG002	100.000	1.000	2
		HG38	98.576	1.011	382
		RA.HG38.HG002	99.595	1.005	169

Table 3.9: QUAST scores for human Chromosomes 8, 19, and X from GRCh38.p13 (HG38) and RAmbler (RA.T2T.CHM13, RA.HG38.HG002, RA.HG38.HG00733 and RA.HG38.HG01346) using T2T-CHM13.v2.0 (T2T) as the ground truth; the length of the chromosomes are 146,259,331 bp, 61,707,364 bp, and 154,259,566 bp, respectively, and from GRCh38.p13 (HG38) and RAmbler (RA.MAT002.HG002 and RA.HG38.HG002) using the maternal strand of HG002 (MAT002) as the ground truth; the length of the chromosomes are 146,740,790 bp, 61,317,360 bp, and 154,341,406 bp respectively

creased from 98.6% to 99.6%. Figure 3.9 shows the synteny analysis based on SyRI [45]. Observe the much stronger synteny between the three RAmbler assemblies and T2T, compared to HG38.

Figure 3.5C-D summarizes the results of one of the runs from the second set of experiments obtained by RAmbler using the PacBio HiFi reads that were used for the T2T project ($\sim 32.4\times$ coverage). Hereafter, this RAmbler assembly is called RA.T2T.CHM13. Observe in Figure 3.5C that T2T and RA.T2T.CHM13 have nearly identical HiFi coverage across all five regions, which is also reflected in the NucFlag analysis in Figure 3.6B. Figure 3.5D illustrates the CRAQ assessments based on the alignment of CHM13 HiFi reads and CHM13 Illumina reads. Observe that both T2T and RA.T2T.CHM13 have similar AQI scores across all three chromosomes with no misjoins, except for three base errors introduced in the centromere of Chromosome 8 and one base error corrected in the Chromosome X by RAmbler (CRAQ numerical scores are reported in Table 3.10). We also used ultra-long ONT reads (longer than 100 kb, $\sim 17\times$ coverage) to further evaluate and compare these assemblies. The UL-ONT coverage and CRAQ plots in Figure 3.10, and the CRAQ numerical scores in Table 3.11 indicate that the RA.T2T.CHM13 has the same quality as T2T. The NucFreq plots, the Circos plots based on CRAQ (numerical scores are reported in Table 3.12) and the NucFlag plots for the other run from this set of experiments are shown in Figures 3.11 and 3.12. We used the MAT002 assembly as input assembly with the HG002 HiFi reads. The coverage plots and the Circos plots from this run demonstrate similar results as shown in Figure 3.5C-D. However, the NucFlag analysis reveals that the RA.MAT002.HG002 assembly had fewer errors than MAT002 in three out of the five repetitive regions. Figure 3.13A indicates a perfect synteny between RA.T2T.CHM13 and T2T across Chromosomes 8, 19, and X. Similarly,

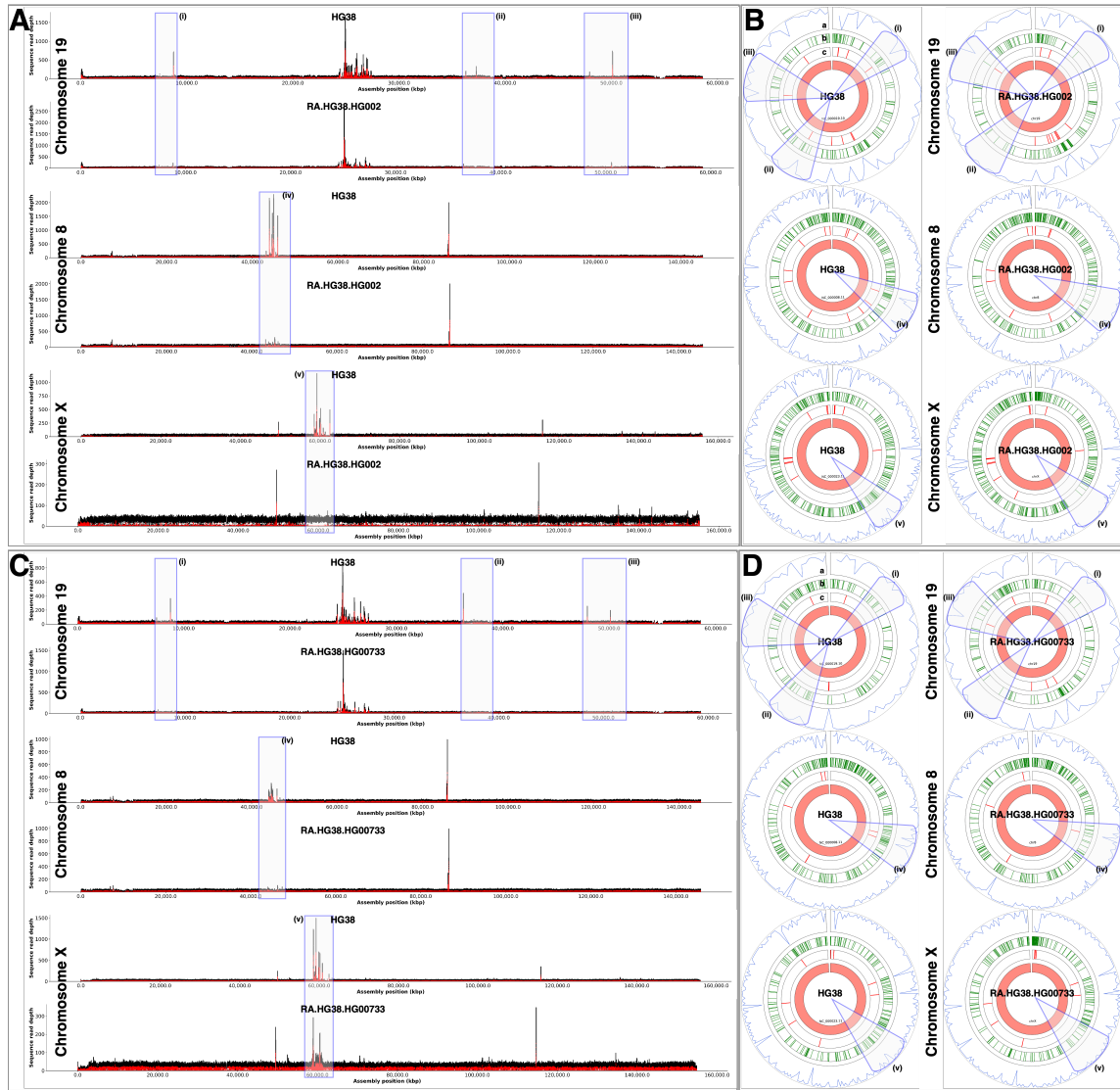


Figure 3.7: Comparing assemblies RA.HG38.HG002 and RA.HG38.HG00733 against GRCh38.p13 (HG38) of human Chromosomes 8, 19, and X; in all plots, blocks (i), (ii), and (iii) are non-centromeric repeats in Chromosome 19; blocks (iv) and (v) are the centromeric regions of Chromosomes 8 and X, respectively; (A, C) NucFreq plots illustrating HiFi read mapping coverage; (B, D) CRAQ Circos plots illustrating (a) assembly quality index (AQI) score (higher is better), (b) base errors (lower is better), and (c) misjoins (lower is better)

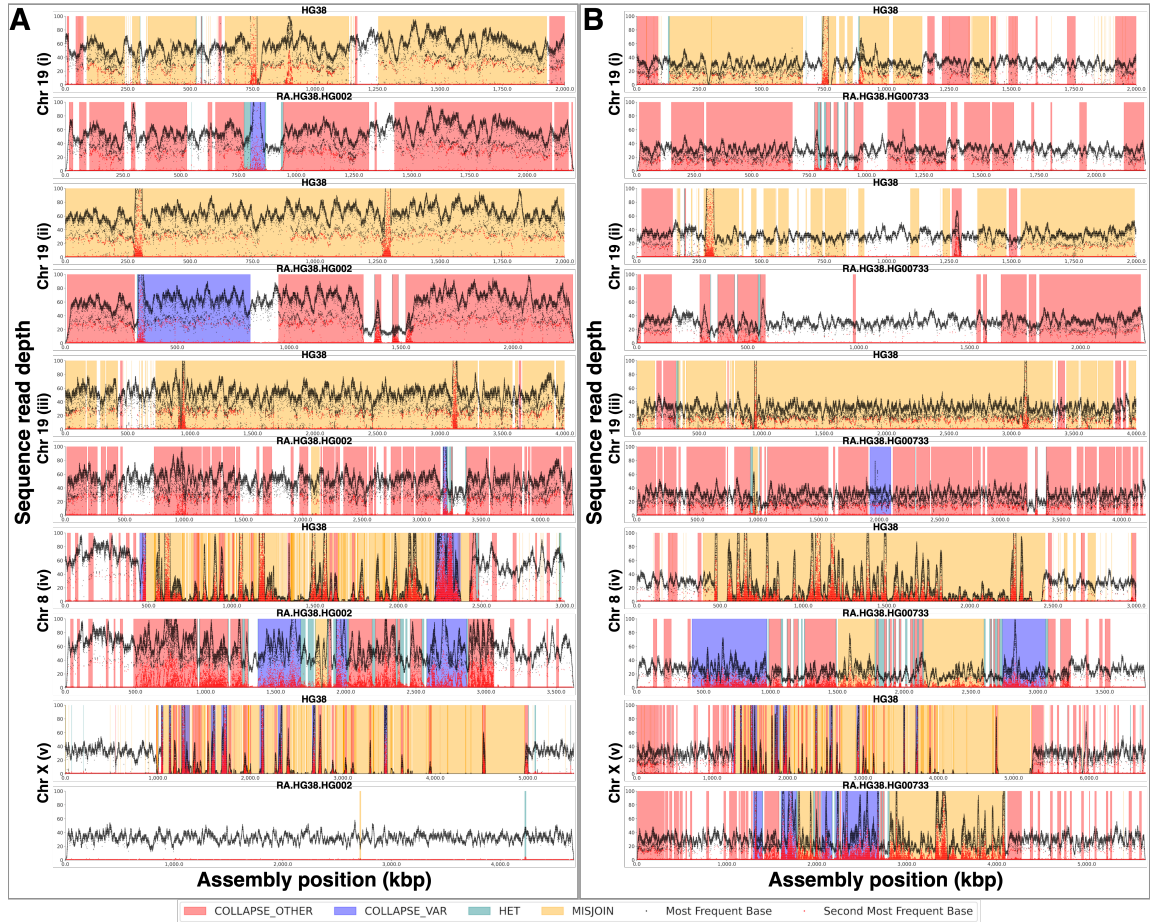


Figure 3.8: NucFlag plots illustrating HiFi read mapping coverage for the comparison of (A) RA.HG38.HG002 assembly and (B) RA.HG38.HG00733 assembly against GRCh38.p13 (HG38) of human Chromosomes 8, 19, and X; in all plots, blocks (i), (ii), and (iii) are non-centromeric repeats in Chromosome 19; blocks (iv) and (v) are the centromeric regions of Chromosomes 8 and X, respectively

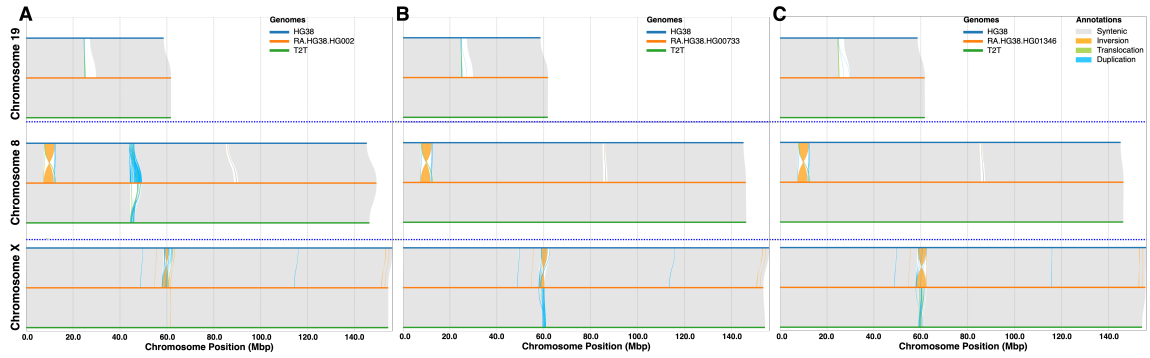


Figure 3.9: Synteny plots of Rambler's assemblies against GRCh38.p13 assembly (HG38) and the T2T-CHM13.v2.0 assembly (T2T) on human Chromosomes 19, 8, and X using SyRI; (A) Synteny plots between HG38 (blue), RA.HG38.HG002 (orange) and T2T (green); (B) Synteny plots between HG38 (blue), RA.HG38.HG00733 (orange) and T2T (green); (C) Synteny plots between HG38 (blue), RA.HG38.HG01346 (orange) and T2T (green)

Figure 3.13B illustrates a perfect synteny between RA.MAT002.HG002 and MAT002 across Chromosomes 8 and 19, and a near perfect synteny with a small inversion in Chromosome X.

Table 3.13 shows the runtime and the memory consumption for Rambler to resolve the centromeres of human Chromosomes 8 and X. Currently Rambler takes longer than hifiasm, LJA and Verkko. Rambler's memory consumption is reasonable. The current implementation of Rambler is not multi-threaded, so there is an opportunity to make it faster and more scalable.

Chromosome	Assembly	Coverage Rate (%)	R-AQI	S-AQI
19	T2T	99.73	97.91	100
	RA.T2T.CHM13	99.68	98.23	
8	T2T	99.85	99.69	
	RA.T2T.CHM13	99.77	99.49	
X	T2T	99.56	98.83	
	RA.T2T.CHM13	99.64	99.25	

Table 3.10: CRAQ report on assembly coverage and errors obtained using short Illumina reads (SRA accession SRR10035392, $\sim 17\times$ coverage) and PacBio HiFi reads (SRA accession SRR11292120-SRR11292123, $\sim 32.4\times$ coverage) on human Chromosomes 19, 8, and X from T2T-CHM13.v2.0 (T2T) and RAmbler (RA.T2T.CHM13); numbers in bold indicate the best scores

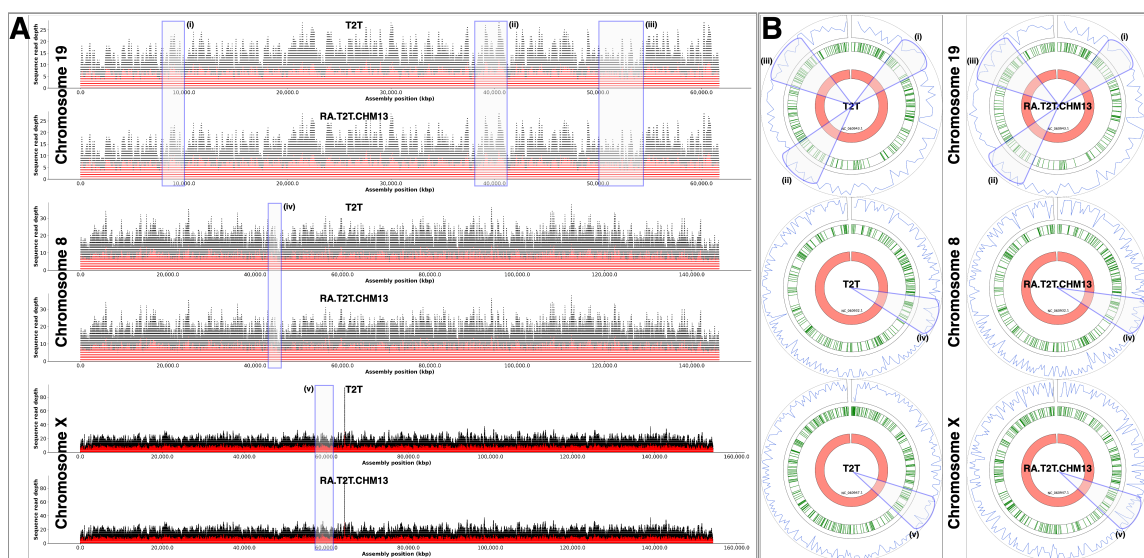


Figure 3.10: Comparing RAMbler's assembly (RA.T2T.CHM13) quality against the T2T-CHM13.v2.0 assembly (T2T) on human Chromosomes 19, 8, and X. Blocks (i), (ii), and (iii) are non-centromeric repeats in Chromosome 19; blocks (iv) and (v) are the centromeric regions of Chromosomes 8 and X; (A) NucFreq plots illustrating ultra-long ONT ([link](#), $\sim 17\times$ coverage (≥ 100 kb)) read mapping coverage for T2T (top), and RAMbler (bottom); (B) CRAQ Circos plots illustrating regional AQI scores, base errors, and misjoins

Chromosome	Assembly	Coverage Rate (%)	R-AQI	S-AQI
19	T2T	99.98	68.60	100
	RA.T2T.CHM13	99.98	72.14	
8	T2T	99.41	84.61	
	RA.T2T.CHM13	99.41	85.57	
X	T2T	99.99	82.03	
	RA.T2T.CHM13	99.99	84.07	

Table 3.11: CRAQ report on assembly coverage and errors obtained using short Illumina reads (SRA accession SRR10035392, $\sim 17\times$ coverage) and ultra-long Oxford Nanopore reads ([link](#), $\sim 17\times$ coverage (≥ 100 kb)) on human Chromosomes 19, 8, and X from T2T-CHM13.v2.0 (T2T), and RAmbler (RA.T2T.CHM13); numbers in bold indicate the best scores

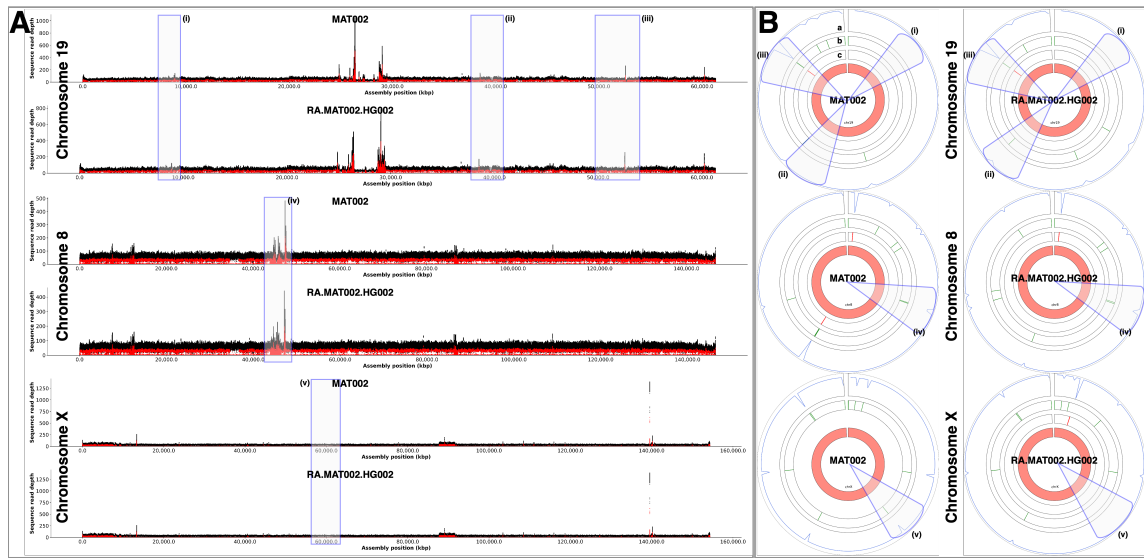


Figure 3.11: Comparing RA.MAT002.HG002 against the maternal HG002 assembly (MAT002) of human Chromosomes 8, 19, and X; in all plots, blocks (i), (ii), and (iii) are non-centromeric repeats in Chromosome 19; blocks (iv) and (v) are the centromeric regions of Chromosomes 8 and X, respectively; (A) NucFreq plots illustrating HiFi read mapping coverage; (B) CRAQ Circos plots illustrating (a) assembly quality index (AQI) score (higher is better), (b) base errors (fewer is better), and (c) misjoins (fewer is better)

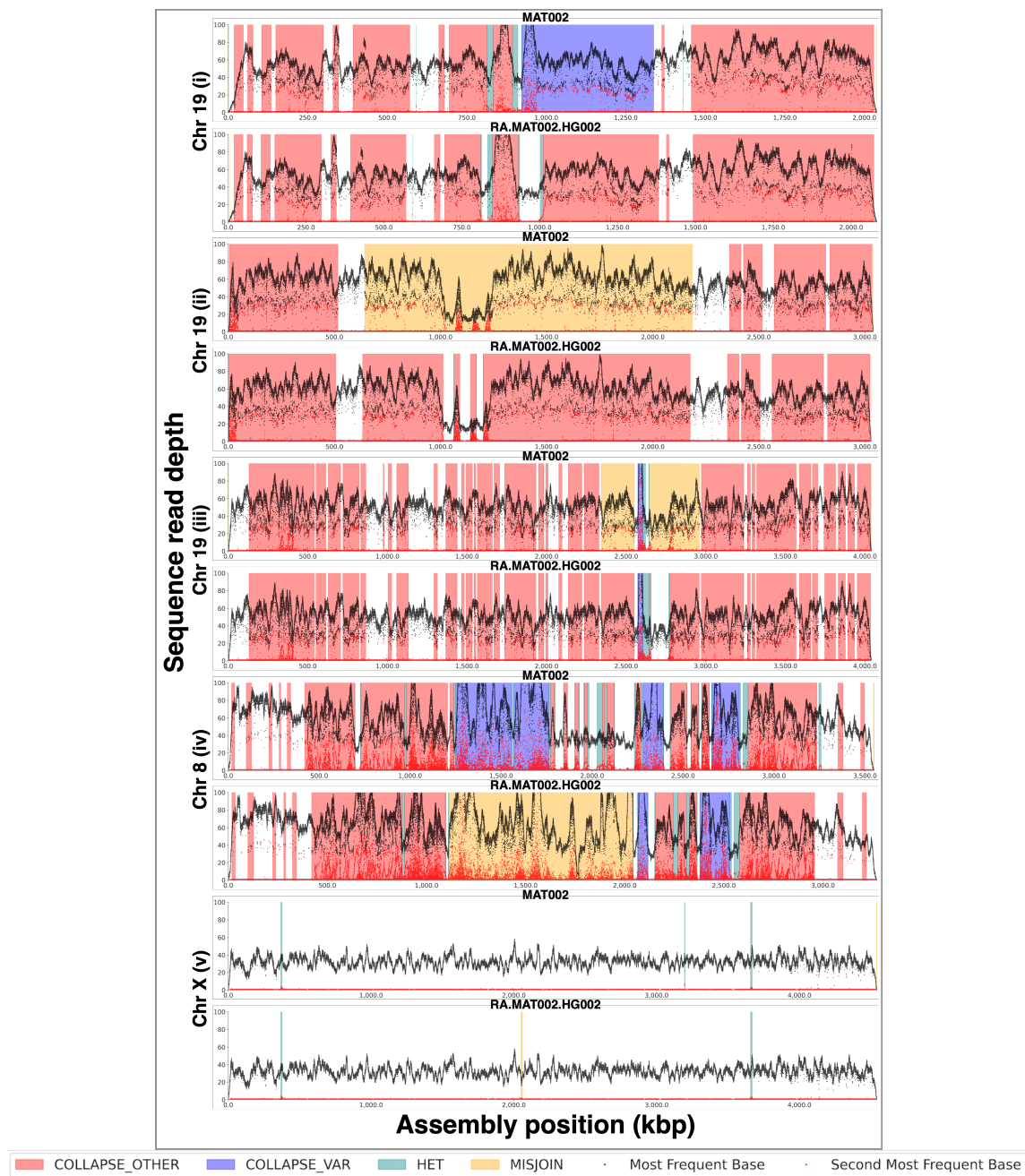


Figure 3.12: NucFlag plots illustrating HiFi read mapping coverage for the comparison of RA.MAT002.HG002 assembly against the maternal HG002 assembly (MAT002) of human Chromosomes 8, 19, and X; in all plots, blocks (i), (ii), and (iii) are non-centromeric repeats in Chromosome 19; blocks (iv) and (v) are the centromeric regions of Chromosomes 8 and X, respectively

Chromosome	Assembly	Coverage Rate (%)	R-AQI	S-AQI
19	MAT002	99.07	98.69	98.37
	RA.MAT002.HG002	99.20	98.86	98.37
8	MAT002	98.97	99.07	98.63
	RA.MAT002.HG002	99.00	99.24	99.31
X	MAT002	99.40	99.29	100
	RA.MAT002.HG002	99.25	99.35	99.35

Table 3.12: CRAQ report on assembly coverage and errors obtained using short Illumina reads ([illuminaHG002](#), $\sim 13.5\times$ coverage) and PacBio HiFi reads ([hifiHG002](#), $\sim 66\times$ coverage) on human Chromosomes 19, 8, and X from the maternal HG002 assembly (MAT002) and RAmbler (RA.MAT002.HG002); numbers in bold indicate the best scores

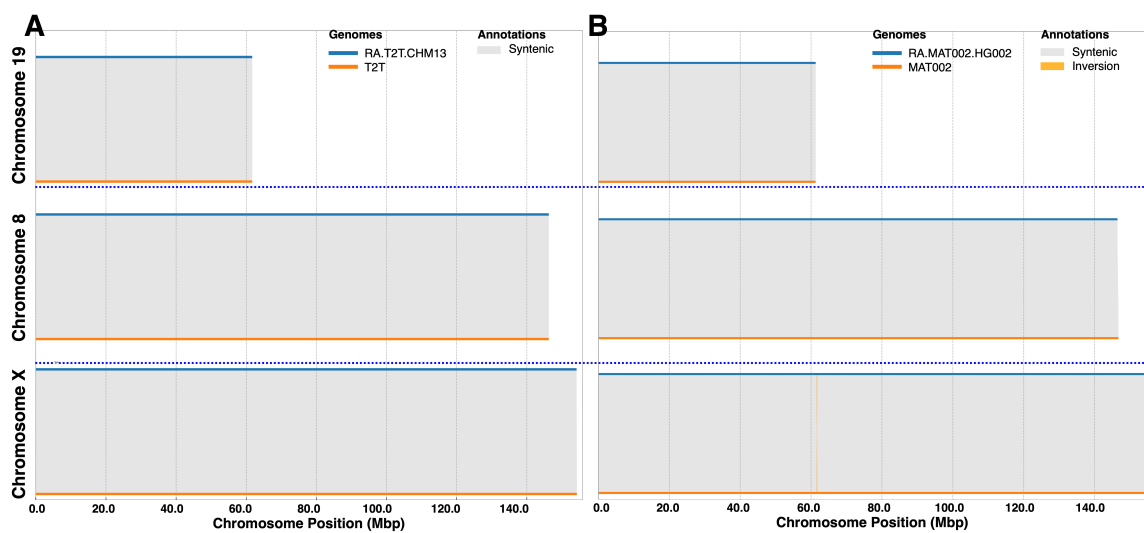


Figure 3.13: Synteny plots of Rambler's assembly against T2T-CHM13.v2.0 (T2T) and the maternal HG002 assembly (MAT002) on human Chromosomes 19, 8, and X using SyRI; (A) Synteny plots between RA.T2T.CHM13 (blue), and T2T (orange); (B) Synteny plots between RA.MAT002.HG002 (blue) and MAT002 (orange)

Centromere	Input Assembly	HiFi Reads Sample	Coverage	Time (hours)	Memory (GB)
8	T2T	CHM13	32.4×	0.84	31.71
	MAT002	HG002	66×	2.42	45.66
	HG38	HG002	67.5×	4.65	57.80
		HG00733	32.8×	1.18	33.61
		HG01346	26×	1.25	43.02
X	T2T	CHM13	34×	2.45	39.67
	MAT002	HG002	33×	2.28	40.47
	HG38	HG002	32.2×	1.88	44.32
		HG00733	31.8×	6.52	58.38
		HG01346	25.5×	4.69	48.46

Table 3.13: Time and memory usage for RAmbler for the reconstruction of the centromeres of Chromosome 8 and X

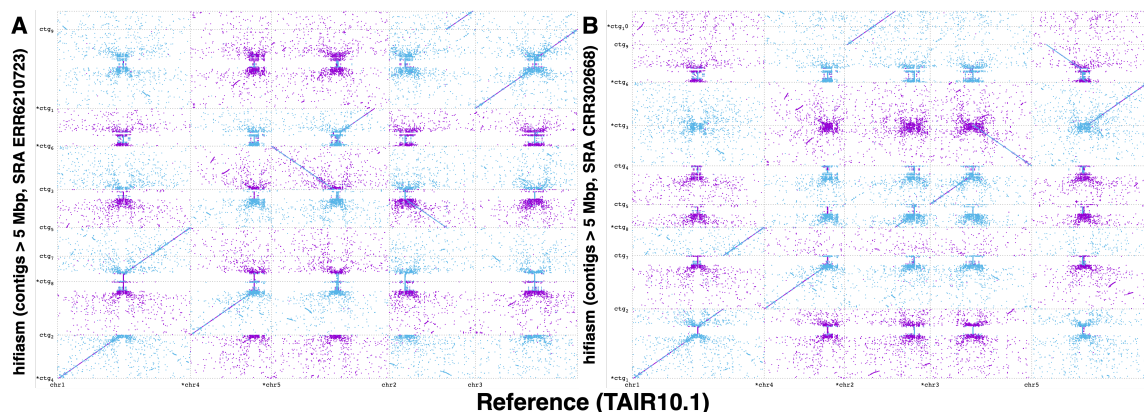


Figure 3.14: Dotplot of contigs longer than 5 Mb produced by hifiasm using PacBio HiFi reads; (A) SRA accession ERR6210723 ($\sim 50\times$ coverage), and (B) SRA accession CRR302668 ($\sim 100\times$ coverage) using the TAIR10.1 assembly as reference

3.5.2 Experimental Results on *Arabidopsis thaliana*

We tested RAMbler on real data for *A. thaliana*, using the TAIR10.1 reference genome and PacBio HiFi reads. RAMbler detected repetitive regions in Chromosomes 2, 3, 4, and 5, which are well-known [24, 130, 52, 92].

When we assembled the whole genome with hifiasm and LJA using two different sets of HiFi reads (SRA accessions ERR6210723 with $\sim 50\times$ coverage and CRR302668 with $\sim 100\times$ coverage) both failed to assemble these repetitive regions (see Figure 3.14).

We focused on three centromeric repetitive regions in Chromosome 5. The three NucFreq plots in Figure 3.15 illustrate the mapping coverage in these repetitive regions. RAMbler estimated the first tandem repeat to be composed of 45 copies of a 4.5 kb unit, the second to be 4 copies of a 11.35 kb unit, and the third to be 75 copies of a 10.45 kb unit (Table 3.15, columns 2 and 3).

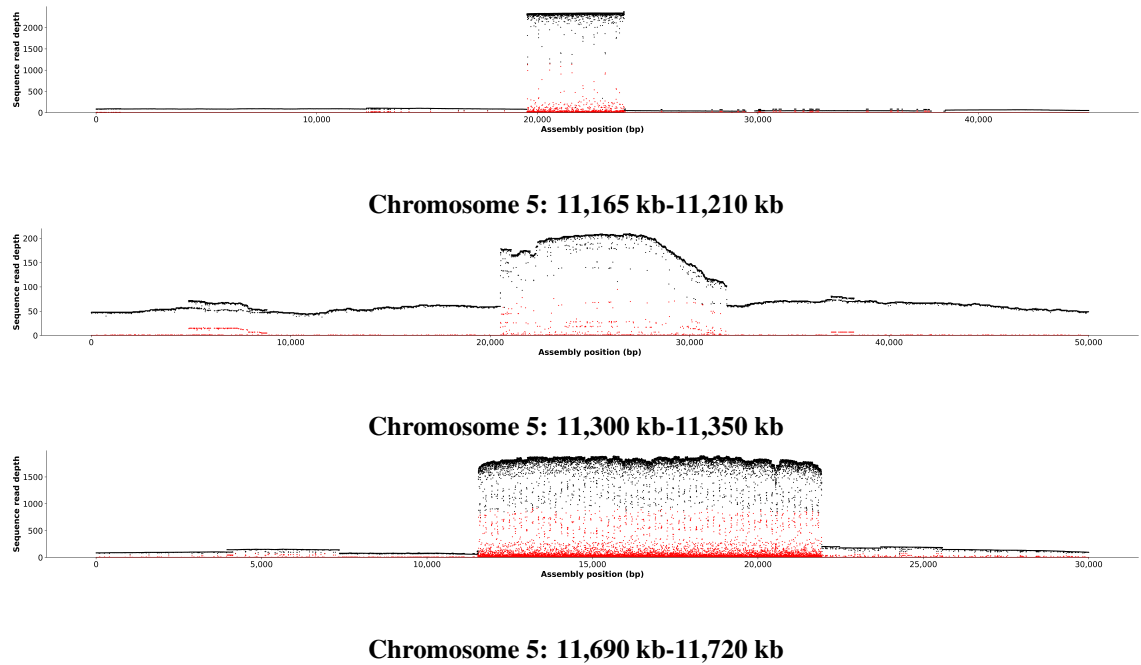


Figure 3.15: PacBio HiFi mapping coverage (SRA accession ERR6210723) for Chromosome 5 in *A. thaliana* illustrated using NucFreq; the coverage spikes indicate the presence of repetitive regions

Reads mapping to these three repetitive regions (including 50 kb upstream and downstream) from both SRA accessions ERR6210723 and CRR302668 were given as inputs to RAMbler, hifiasm, LJA, HiCANU, and Verkko. Table 3.14 reports the number of contigs, total assembly size, and the length of the longest contig for each run of all assemblers. Observe that RAMbler produces assemblies with (i) the fewest contigs in most cases, (ii) the longest contig is longer than those generated by other assemblers, and (iii) the total assembly size closely matches the expected assembly size. hifiasm closely follows RAMbler in these metrics. Although LJA produces the fewest contigs in all cases, it significantly over-compresses the assembly, resulting in a total size that deviates considerably from the expected assembly size. Both HiCANU and Verkko generate very fragmented assemblies and over-inflate the total assembly size. HiCANU failed to complete the assembly even

Accession	Coverage	Repetitive region		# Selected reads	Selected reads sum (Mb)	Expected size (Mb)		Assembler				
		start (Mb)	end (Mb)					RAmbler	hifiasm	LJA	HiCANU	Verkko
ERR6210723	50×	11.165	11.210	2,627	41.642	0.833	# contigs	2	2	1	10	3
							Total (bp)	797,754	801,770	797,803	1,000,811	812,947
							Longest contig	736,535	740,551	797,803	793,849	736,533
		11.300	11.350	379	5.876	0.118	# contigs	5	4	1	5	N/A
							Total (bp)	190,879	155,220	84,893	166,018	
							Longest contig	84,898	84,897	84,893	84,234	
		11.690	11.720	1,531	23.821	0.476	# contigs	7	19	3	20	31
							Total (bp)	508,321	754,782	273,474	736,544	884,083
							Longest contig	125,800	125,800	106,364	124,464	106,199
CRR302668	100×	11.165	11.210	6,524	99.726	0.997	# contigs	1	1	1	-	10
							Total (bp)	802,698	802,698	799,690	-	933,944
							Longest contig	802,698	802,698	799,690	-	712,440
		11.300	11.350	721	10.687	0.107	# contigs	3	5	1	7	13
							Total (bp)	99,045	192,442	89,571	214,294	327,792
							Longest contig	54,704	89,577	89,571	86,657	54,705
		11.690	11.720	5,175	78.302	0.783	# contigs	6	24	7	21	47
							Total (bp)	493,659	1,090,173	341,226	726,310	1,166,414
							Longest contig	196,786	196,787	108,554	105,995	108,546

Table 3.14: Assembly statistics for RAmbler, hifiasm, LJA, HiCANU and Verkko on three centromeric repeats in Chromosome 5 of *A. thaliana* using TAIR10.1 as input assembly; Symbol ‘-’ means that the assembler timed out after 48 hours, ‘N/A’ means that the assembler finished execution with no output

after 48 hours in one instance, while Verkko finished execution without producing any output in another instance.

Table 3.15 shows the number of contigs along with the number of repeat copies reconstructed by RAmbler, hifiasm, LJA, HiCANU, and Verkko using the HiFi reads from SRA accession ERR6210723.

On the first 45-copy tandem repeat, both RAmbler and hifiasm generated two contigs,

Repetitive region	Expected		RAmbler		hifiasm		LJA		HiCANU		Verkko	
	unit size	# copies	# contigs	# copies	# contigs	# copies	# contigs	# copies	# contigs	# copies	# contigs	# copies
11165 kb-11210 kb	4.5 kb	45	2	44	2	0	1	0	10	0	3	0
11300 kb-11350 kb	11.35 kb	4	5	4	4	1	1	1	5	1	N/A	N/A
11690 kb-11720 kb	10.45 kb	75	7	72	19	0	3	0	20	0	31	0

Table 3.15: Repeat resolution statistics for RAmbler, hifiasm, LJA, HiCANU and Verkko on real HiFi reads (SRA accession ERR6210723) for three repeats of *A. thaliana* in Chromosome 5; HiCANU required more than 40 hours to complete all these assemblies and Verkko failed to generate outputs in one case

LJA created a single contig, HiCANU produced ten contigs, and Verkko produced three contigs. We aligned these assemblies against the repeat unit extracted from the reference genome using minimap2. The dotplot confirmed 44 copies for RAmbler (dotplot not shown). The assembly produced by hifiasm, LJA, HiCANU and Verkko did not align well against the repeat unit thus resolving zero copies (dotplots not shown).

On the second centromeric four-copy tandem repeat, both RAmbler and HiCANU generated assemblies with five contigs, LJA produced a single contig and hifiasm created four contigs. Verkko did not generate any output. The evaluation with minimap2 and dot plot revealed that RAmbler resolved all four copies, while hifiasm, LJA, and HiCANU resolved only a single copy of the repeat (dotplots not shown).

On the third, and most challenging, 75-copy tandem repeat, RAmbler, hifiasm, LJA, HiCANU, and Verkko generated assemblies with 7, 19, 3, 20, and 31 contigs, respectively. When we aligned these contigs against the repeat unit using minimap2, the dot plot revealed that RAmbler successfully resolved 72 copies, while all the others failed to resolve any copy (dotplots not shown).

3.6 Discussion

The extensive set of experiments on human Chromosomes 8, 19, and X across centromeric and non-centromeric complex repetitive regions demonstrated RAMbler’s ability to achieve T2T-level assembly quality using PacBio HiFi reads exclusively, without manual intervention. Our comparative experimental results on more than 250 synthetic data sets, and on real data for *H. sapiens* and *A. thaliana*, indicated that RAMbler outperformed hifiasm, LJA, HiCANU, and Verkko on reconstructing these repetitive regions in the majority of cases. RAMbler generated assemblies with the fewest contigs, achieving higher completeness, contiguity, and accuracy in them. Our analysis of synthetic diploid genomes with haplotype-dependent copy number illustrated that RAMbler can produce less fragmented assemblies with fewer haplotype switching errors compared to LJA, HiCANU, and Verkko by relying exclusively on HiFi reads. hifiasm performed comparably to RAMbler on synthetic data sets with fixed-length repeat units and copy number variation between the haplotypes, but its performance declined with variable-length repeats and further deteriorated on real datasets. LJA also performed well, producing assemblies with a low number of contigs, second only to RAMbler on synthetic data. However, LJA performed poorly on contiguity when the region to be assembled was longer than 1 Mb on real data. HiCANU and Verkko generated assemblies with a large number of contigs. Although HiCANU and Verkko rarely produced misassembled contigs, they suffered from poor contiguity, creating many small contigs and inflating the total assembly size. Additionally, there were instances where HiCANU failed to complete the assembly on real datasets. Similarly, Verkko failed to generate an assembly in two separate instances of *H. sapiens* dataset and majority cases of synthetic diploid genomes.

Chapter 4

Towards Optimal Selection of Ultra-Deep Sequencing Reads for *De Novo* Genome Assembly

4.1 Introduction

A practical issue that has been largely unexplored in *de novo* genome assembly is how to deal with too much sequencing data. It is common knowledge that a sequencing depth of 30-50 $\times$ is ideal to obtain a complete and highly contiguous assembly from PacBio HiFi reads [136]. It is also known that a sequencing depth lower than 30 $\times$ can degrade assembly contiguity [96]. It is not well understood, however, what the consequences on the genome assembly quality are when the sequencing depth is much higher than 50 $\times$. This is the subject of this study.

Taking this scenario to the limit, the problem of assembling ultra-deep sequencing data

(i.e., sequencing depth higher than $1,000\times$) has been largely unexplored. Ultra-deep sequencing data is not uncommon. Illumina ultra-deep sequencing has been used for detecting rare mutations causing cancer [71], for studying subclonal structures in cancer [18, 71], for sequencing viruses [18, 11, 42, 84] and organelle genomes [38, 5]. A few studies addressed the problem of genome assembly for ultra-deep Illumina data [80, 89]. Here we address the problem of assembling ultra-deep PacBio HiFi sequencing data for the first time.

Due to its higher cost, ultra-deep PacBio HiFi is not as common as ultra-deep Illumina data. However, it can occur when attempting to assemble small genomes. For instance, in [119, 120] the authors encountered major challenges while assembling the genomes of various *Babesia* species from ultra-deep HiFi sequencing data, which is a tick-borne pathogen responsible for babesiosis in humans and cattle. They observed that in the presence of “excessive sequencing coverage”, current state-of-the-art assemblers like hifiasm [26] and HiCANU [96] produced unsatisfactory assemblies. Their solution to reduce the sequencing depth was to select the longest 150 thousand reads out of the 2.7 M PacBio HiFi reads [120]. We will show here that selecting the longest reads is not necessarily the best strategy to obtain the best assembly. We will also show that selecting the subset of reads that have the best average Phred quality values does not lead to good assemblies. Two tools have been proposed in the literature for this problem. YAK assesses the quality of HiFi reads by comparing their k -mer distribution to that of short reads (if available) [2]. Note that YAK is used within hifiasm for trio-binning when parental short reads are provided [26]. Filtlong scores long reads based on their length and the percentage of their bases that match a reference sequence or a set of high-quality short reads [1]. It can be used for ranking reads by quality and retaining the best-scoring subset. Thus, filtlong attempts at reducing data volume while preserving the most informative reads.

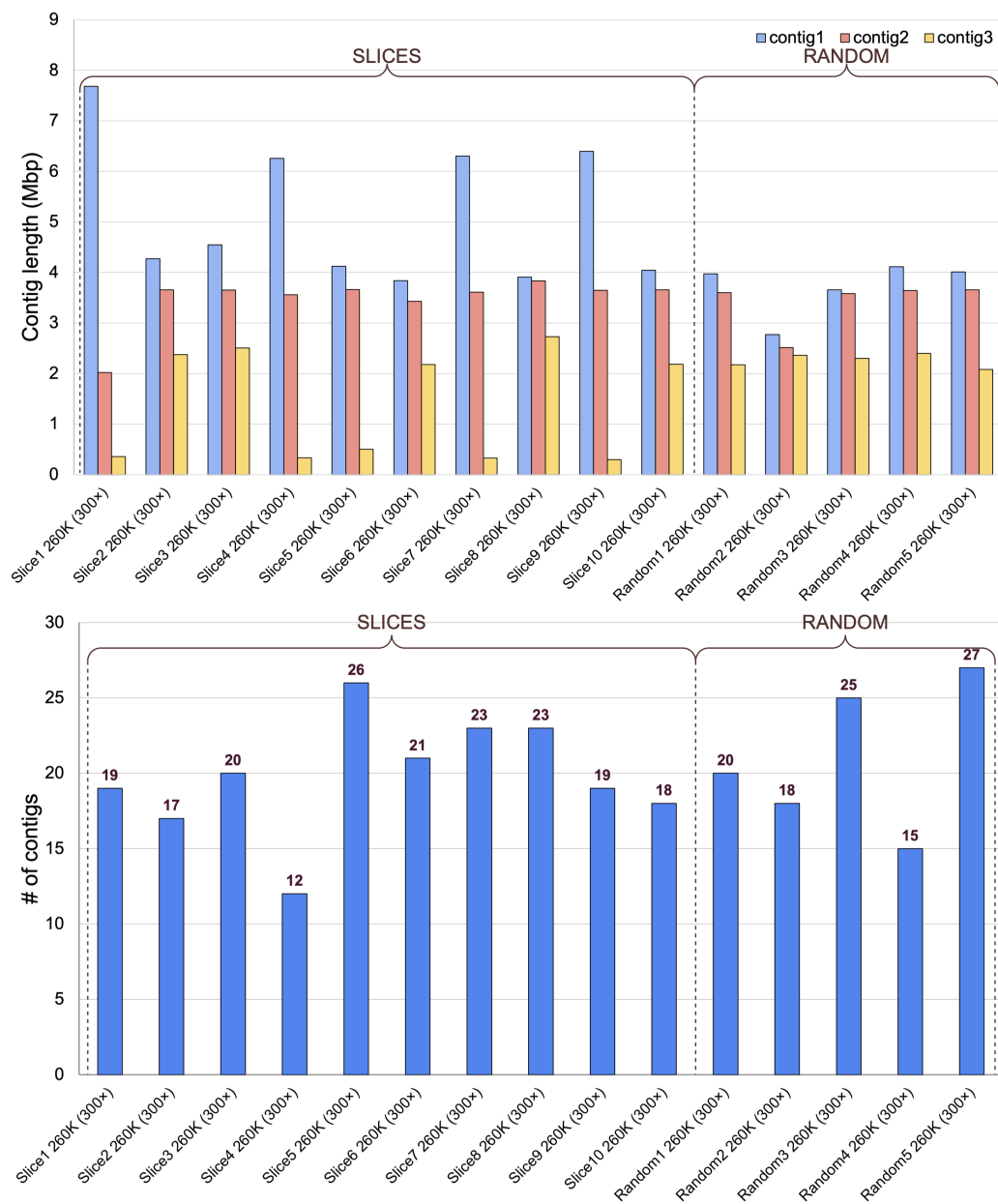


Figure 4.1: Running hifiasm on (1) subset of reads selected at sequencing depth of $300\times$ with uniform SLICES and (2) RANDOM sets at $300\times$ (from a set of real HiFi read with sequencing depth of $3000\times$) for the genome of *B. divergens MOI*; (top) length of the three longest assembled contigs produced by hifiasm; (bottom) number of assembled contigs produced by hifiasm

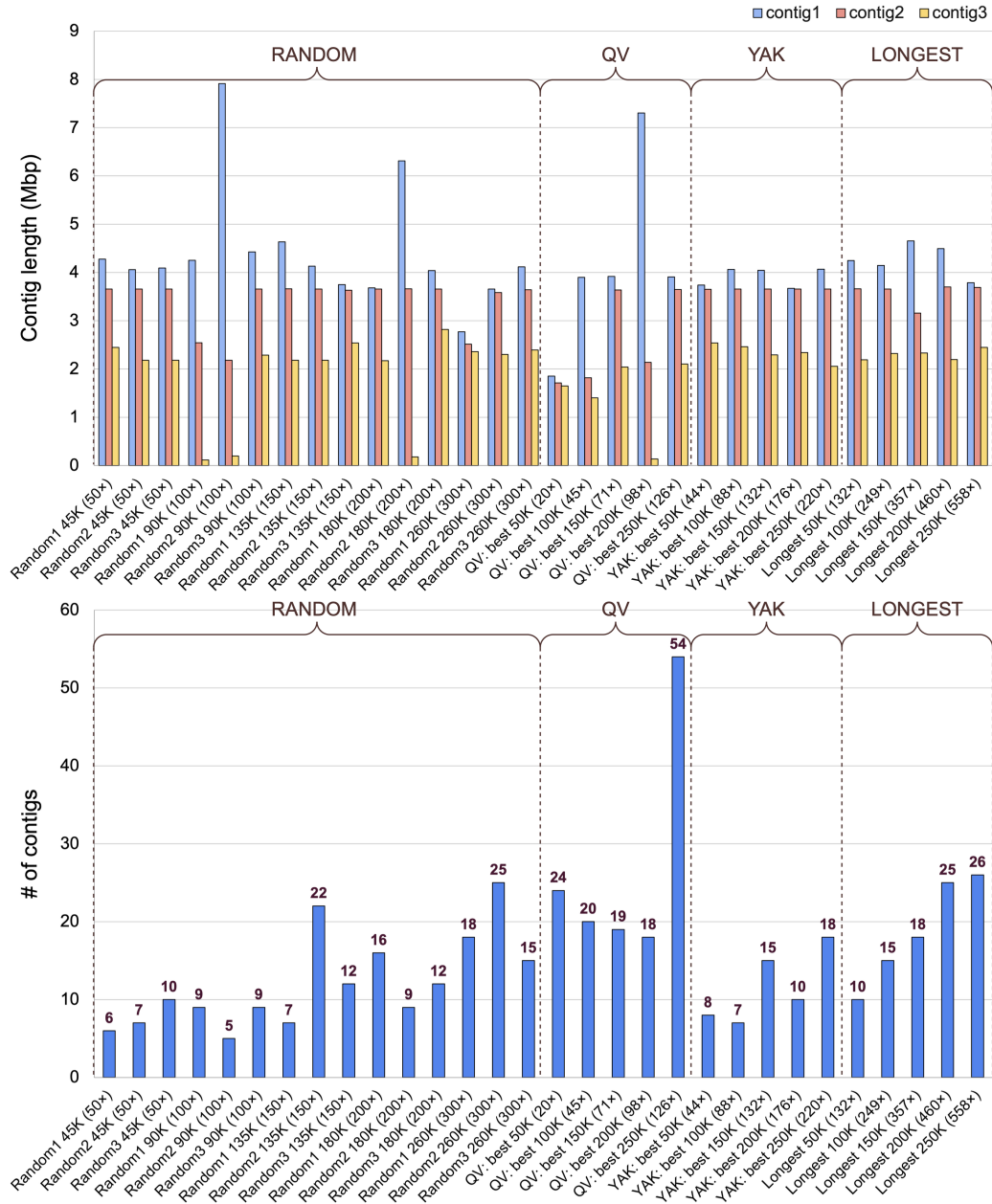


Figure 4.2: Running hifiasm on subset of reads selected using various strategies and sequencing depths from a set of real HiFi read with sequencing depth of $3000\times$ for the genome of *B. divergens* *MOI*; (top) length of the three longest assembled contigs produced by hifiasm; (bottom) number of assembled contigs produced by hifiasm

Here we propose a read selection strategy based on the frequency of rare k -mers in the reads. In the last 15 years, several studies have used single-copy or rare k -mers for the assembly or the alignment of repetitive genomic regions such as centromeres and segmental duplications. The first use of single-copy k -mers dates to 2010 [122]. In their paper, Sudmant et al. define the notion of a *singly unique nucleotide* (SUN) and used it to provide accurate estimates of copy number variation and multi-copy genes in the human genome. Ten years later, the same group developed a method to assemble human chromosome 8 by exploiting the use of single-copy k -mers to resolve complex repetitive regions in its centromere [79]. A recent study resolved several repeats in human chromosomes 8, 19, and X by utilizing single-copy k -mers [22]. CentroFlye [16] also used rare k -mers to classify reads that belong to the repetitive regions. The same group used the idea of using rare k -mers for the problem of mapping long reads accurately in the presence of repeats [88, 15, 17]. To the best of our knowledge, single-copy k -mers have not been used for read selection in the context of *de novo* genome assembly in the presence of ultra-deep sequencing data.

As said, in this thesis we address the *optimal read selection problem*, that is the problem of selecting the subset of ultra-deep sequencing reads that can generate the best assembly (in terms of contiguity and completeness). We propose a novel technique called AWinK (Adaptive **W**indow-based read selection using Uni**K**mers) to optimally select reads using single-copy k -mers from ultra-deep sequencing data that can produce a highly contiguous and complete assembly [23]. In AWinK, we (i) extract single-copy k -mers from the ultra-deep sequencing reads, (ii) barcode the reads based on the presence of single-copy k -mers, (iii) bin the barcoded reads based on the frequency of single-copy k -mers, (iv) select reads from each bin to meet the desired target coverage, and (v) assemble the selected reads.

4.2 Preliminary Analysis

First, we demonstrate the difficulty of assembling a genome from ultra-deep sequencing reads. We used real HiFi reads for the genome of *B. divergens* *MOI* that have a sequencing coverage of about $3000\times$ (assuming a genome of 10 Mb) [120]. We first attempted to assemble this dataset with both hifiasm [26] and HiCANU [96]. HiCANU warned us of “excessive coverage” and selected reads arbitrarily to reduce the amount of data to $200\times$. After running for three days, hifiasm crashed due to a lack of memory, despite having 512 Gb on the server. We should note that hifiasm can easily complete the assembly of a $300\times$ subset of the data in a few minutes. The author of hifiasm, Dr. Haoyu Cheng, confirmed that in the presence of ultra-deep data, hifiasm is likely to get stuck in the error-correction phase.

Since running hifiasm and HiCANU on the entire dataset was not an option, the next step was to evaluate how they would perform on various samples. We explored several options to sample the reads: (i) break the dataset into equal-sized slices by maintaining the order of the reads in the original file (called SLICES in Figure 4.1), (ii) select a random sample (RANDOM at $300\times$ in Figure 4.1), (iii) select random samples (RANDOM) at variable coverage depths in Figure 4.2, (iv) select reads with the highest average Phred quality scores (QV in Figure 4.2), (v) select reads with the highest YAK scores [2] (YAK in Figure 4.2), (vi) select the longest reads (LONGEST in Figure 4.2). Then, we assembled each subset of reads with hifiasm using the default parameters. The x -axis in both histograms in Figures 4.1 and 4.2 reports the strategy used to select the reads and the corresponding sequencing depth in parentheses. The top histograms show the length of the longest assembled contig, the second longest contig, and the third longest contig. The bottom

histograms show the number of contigs produced by hifiasm on the same set of reads.

The results in Figures 4.1 and 4.2 show highly variable assemblies, even though all of them had sufficient coverage to produce a “good” assembly. In particular, in Figure 4.1, (i) on the ten consecutive slices, the length of the longest contig ranged from 3.8 Mb to 7.6 Mb, while the number of contigs ranged from 12 to 26; (ii) random samples of $300\times$ showed a similar variance in the number of contigs; in Figure 4.2, (iii) on random samples at different coverage depths $\{50\times, 100\times, 150\times, 200\times\}$, the length of the longest contig ranged from 2.8 Mb to 7.9 Mb, (iv) selecting reads based on quality values also gave unpredictable results; for instance, the assembly for the top 250K reads had 54 contigs, while the assembly for the top 200K reads had 18 contigs; (v) YAK-score based read selection also showed similar variability in their assemblies; (vi) the number of contigs increased (instead of decreasing) as we provided more and more of the longest reads to hifiasm. The counter-intuitive message is that more HiFi data does not lead to better assemblies. Similar observations were reported many years ago on the assembly of ultra-deep Illumina data using Velvet [139] and SPAdes [9] in [80]. In that latter study, the authors also observed that the assembly quality improved as the sequencing depth increased, up to a certain point (about $800\times$ for Illumina reads using Velvet), then started to degrade.

4.3 Methods

4.3.1 Problem Formulation

Given a set T of ultra-deep sequencing reads sequenced for an unknown G , the objective is to select $T' \subset T$ such that the genome G' generated by assembling T' is as close as to G with respect to



Figure 4.3: Problem formulation. Given a dataset of ultra-deep sequencing reads (top), AWinK selects a subset of reads that when given to hifiasm, it will produce a contiguous and complete *de novo* assembly (bottom); if the positions of the reads were known, the minimum tiling path (MTP) is the smallest subset of the reads that can be assembled to a complete genome (middle)

accuracy, contiguity, and completeness. Observe that if we knew the location of each read $r \in T$ in the genome G , we could select the optimal read set $T_{opt} \subset T$ by finding the *minimum tiling path* (MTP) of the reads (more details in Section 4.3.2). However, since G is unknown, we do not know the genomic location of the reads. Below we propose a heuristic that can approximate T_{opt} using single-copy k -mers (henceforth called *unikmers*). Figure 4.3 illustrates the problem, where T_{opt} is in the middle and the sub-optimal T' is at the bottom.

As said, the objective is to obtain the highest quality genome G' from the selected reads T' . On synthetic reads (i.e., when the ground truth G is known), one can measure the differences between G' and G in terms of contiguity or completeness. On real data sets (i.e., when the ground truth G is unavailable), an assessment of the contiguity and completeness of G' can be obtained only by comparing G' to a genome map, e.g., a genetic, physical, or optical map for G .

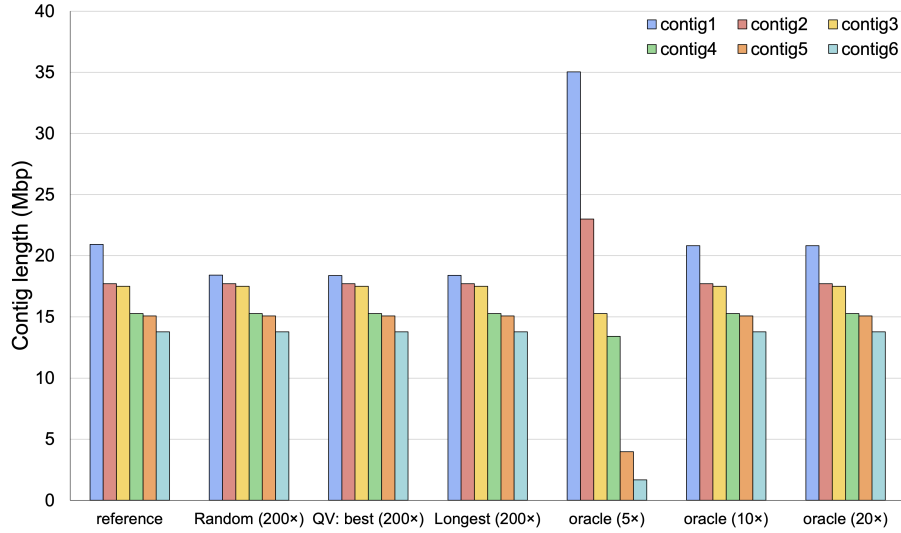


Figure 4.4: The length of the six longest contigs produced by hifiasm on a subset of reads selected (i) at random (200 $\times$ coverage), (ii) with best average QV (200 $\times$), (iii) with longest (200 $\times$), and (iv) produced by the “oracle” at 5 $\times$, 10 $\times$, and 20 $\times$ coverage selected from a set of PBSim synthetic HiFi reads with a sequencing depth of 1000 $\times$ for the genome of *C. elegans* (contig1 is chromosome V); numerical values are available in Table 4.1

4.3.2 Optimal Selection

In this section, we attempt to address the following question: what is the quality of the best possible assembly if one had access to the position of the reads in the unknown genome? Here we assume that there exists an “oracle” that has that knowledge. If the oracle existed, the optimal selection problem could be solved by the following steps. Let D be the desired sequencing depth for T_{opt} . First, (i) represent each read by an interval $[i, j]$, where i is the start position and j is the end position of that read in the genome; (ii) remove any interval that is completely contained in other intervals; (iii) build

Assembly	contig1	contig2	contig3	contig4	contig5	contig6	# of contigs	Total length (bp)
Reference	20,924,180	17,718,942	17,493,829	15,279,421	15,072,434	13,783,801	7	100,286,401
Random (200×)	18,416,632	17,718,798	17,493,796	15,279,214	15,072,421	13,783,719	9	100,285,249
QV: best (200×)	18,379,014	17,718,881	17,493,792	15,279,304	15,072,395	13,783,687	9	100,239,133
Longest (200×)	18,389,640	17,718,888	17,493,256	15,279,172	15,072,425	13,783,562	8	100,216,292
oracle (5×)	35,032,850	23,002,036	15,273,332	13,406,979	3,983,934	1,667,497	8	92,427,632
oracle (10×)	20,815,879	17,718,653	17,493,341	15,279,162	15,072,238	13,775,179	7	100,188,613
oracle (20×)	20,816,019	17,718,829	17,493,548	15,279,162	15,072,238	13,783,620	7	100,198,975

Table 4.1: The length of the six longest contigs, number of assembled contigs and the total assembly length produced by hifiasm on a subset of reads selected (i) at random (200× coverage), (ii) with best average QV (200×), (iii) with longest (200×), and (iv) produced by the “oracle” at 5×, 10×, and 20× coverage selected from a set of PBSim synthetic HiFi reads with a sequencing depth of 1000× for the genome of *C. elegans* (contig1 is chromosome V)

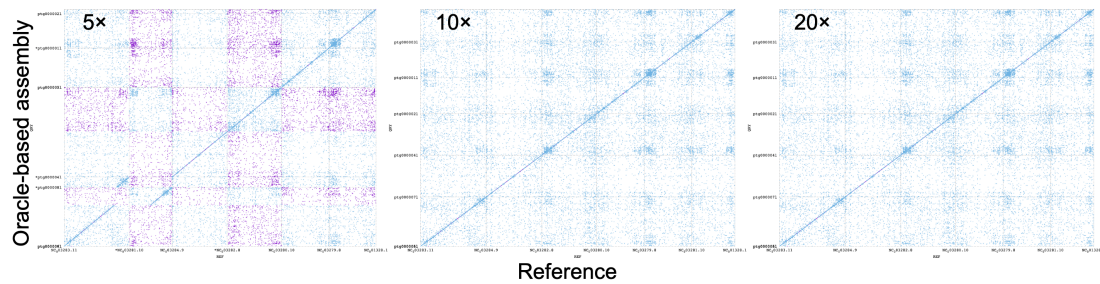


Figure 4.5: Dot plots comparing three hifiasm assemblies to the *C. elegans* reference genome using a subset of reads produced by the “oracle” at 5×, 10×, and 20× coverage selected from a set of PBSim synthetic HiFi reads with a sequencing depth of 1000×

an interval graph, in which each node is an interval and there is an edge between two nodes if the corresponding intervals overlap by at least mo (here we have used $mo = 100$ bp); (iv) find a shortest path from the first interval (which corresponds to the 5' end of a chromosome) to the last interval (which corresponds to the 3' end of a chromosome) of this unweighted interval graph. Observe that each shortest path will provide about $1\times$ of sequencing depth using the minimum number of reads (thus implicitly favoring longer reads). The reads on the shortest path, which represent a *minimum tiling path* of the chromosome, can be (v) added to the set T_{opt} (initially empty), (vi) then removed from the interval graph. By repeating steps (iv-vi) D times, T_{opt} will end up with a sequencing depth approximately equal to D .

We have tested this strategy on the *C. elegans* genome, which is about 100 Mb and has five chromosomes, one X chromosome, and one mitochondrial genome. We generated $1000\times$ HiFi reads with PBSim [97] with the CCS model, then produced assemblies from subsets of reads obtained by the following selection strategies: (i) randomly selected reads – $200\times$ coverage, (ii) highest average QV – $200\times$ coverage, (iii) longest reads – $200\times$ coverage and (iv) oracle-based at $5\times$, $10\times$, and $20\times$ coverage. As shown in Figure 4.4, even at $10\times$ coverage of reads, the hifiasm assembly of oracle-selected reads matched almost perfectly the reference genome (see the dot plots in Figure 4.5), while the assemblies obtained from selection strategies (i-iii) produced incorrect assemblies (e.g., observe the shorter length of contig1 for Random, QV, and Longest in Figure 4.4). This experiment seems to indicate that the oracle-based selection strategy can produce subsets of reads that will lead to very high quality assemblies. However, the oracle does not exist when the genome is unknown. Our objective is to find a way to “approximate” the interval graph without knowing the locations of the reads.

4.3.3 Analysis of Unikmer Distribution

The key idea of our approach is to take advantage of *unikmers* to select subsets of ultra-deep sequencing reads. Since *unikmers* occur only once in the genome, it would seem that selecting the subset of reads that cover all the *unikmers* could approximate the minimum tiling path. This would be true if the distribution of *unikmers* was somewhat uniform and sufficiently high across the genome. However, when we analyzed the frequency of *unikmers* across the genome of *C. elegans* we discovered that it was not the case. Figure 4.6 illustrates such frequency distribution (*y*-axis) in 20 kbp non-overlapping sliding windows (*x*-axis) across the six chromosomes of *C. elegans*. While the average density is about 9,200 *unikmers* per 20 kbp, it peaks at 16,117 *unikmers* per 20 kbp but bottoms out with a minimum of zero *unikmers*. For example, there is a 140 kbp region in chromosome V of *C. elegans* at coordinates [2.34,2.48] Mbp that has no *unikmers*. This indicates that selecting the subset of reads that cover all *unikmers* would not cover the entire genome.

Based on these observations, we formulated the following refined selection criterion: select a subset of reads in a way that would maintain the overall distribution of *unikmers*. To test our selection criterion, we analyzed the *unikmer* distribution on a dataset of synthetic HiFi reads generated by PBSim [97] for *C. elegans* at 1000 $\times$ coverage. Observe in Figure 4.7 that (i) the frequency of reads with a given number of *unikmers* follows a positively-skewed normal distribution (its mean is approximately 9,727), and (ii) there is an out-of-distribution spike at the beginning of the plot where there are a large number of reads with very few *unikmers* (≤ 5). For instance, we counted about 4,000 reads with no *unikmers*. Obviously, one expects that the distribution of *unikmer* in the reads matches the distribution of *unikmer* in the genome, only “amplified” by the depth of coverage.

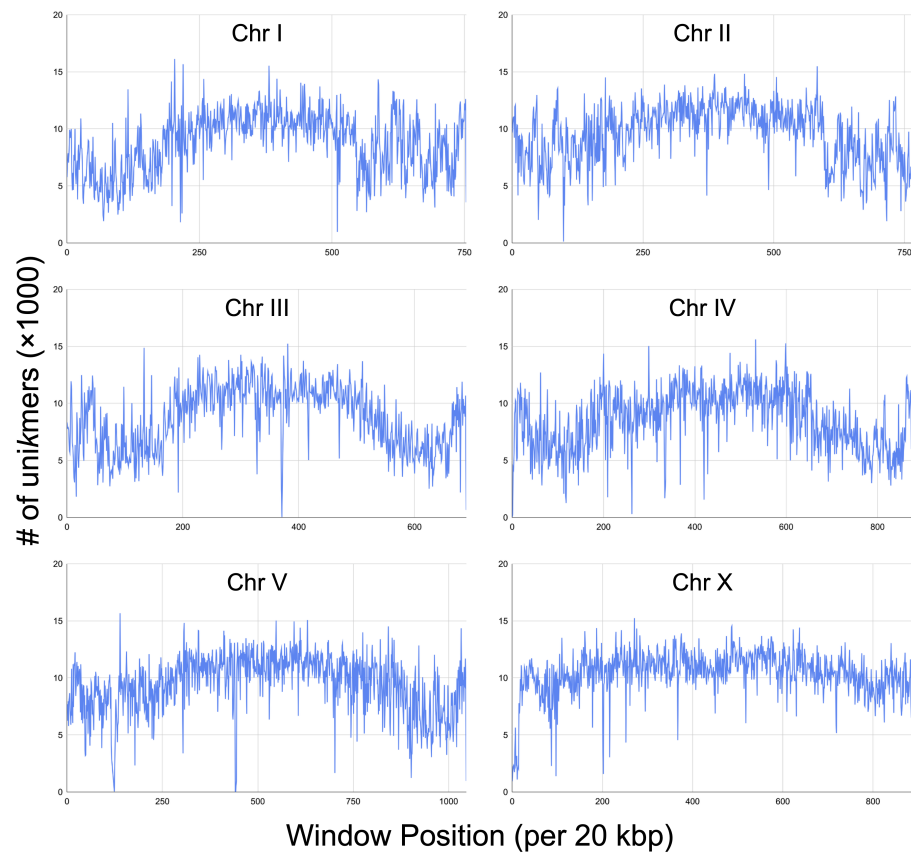


Figure 4.6: The frequency distribution of *unikmers* in 20 kbp non-overlapping sliding windows across the six chromosomes of *C. elegans*

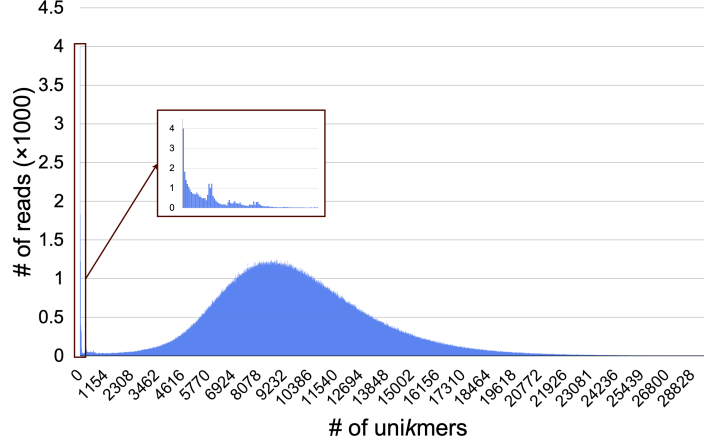


Figure 4.7: The frequency distribution of *unikmers* in a data set of synthetic HiFi reads at $1000\times$ coverage for *C. elegans*

4.3.4 AWinK's Algorithm

Based on the analysis of the *unikmer* distribution in *C. elegans* and the synthetic ultra-deep sequencing reads in the previous section, we propose to approximate the optimal selection problem by choosing reads in a way so their *unikmers* frequency distribution mirrors the distribution of the entire ultra-deep dataset (as shown in Figure 4.7). The selected reads are intended to capture genomic regions with all possible *unikmer* density, including regions with no or very few *unikmers*, thus maximizing the overall genome coverage. Our adaptive-window selection technique called AWinK is illustrated in Figure 4.8 and explained next. Users are expected to choose the desired target coverage depth D_t for the selected reads $T' \subset T$.

A. Extract *unikmers*. AWinK uses Jellyfish [85] on the reads T to obtain the count distribution of k -mers genome-wide. Next, it calculates the mean μ_u and standard deviation σ_u of the distribution.

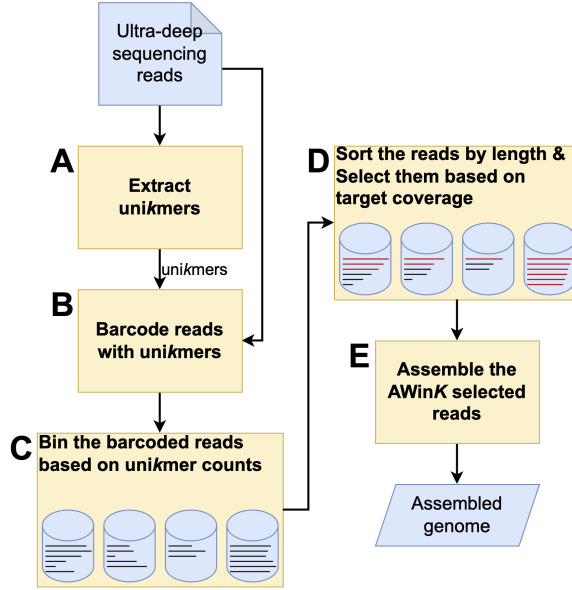


Figure 4.8: The algorithmic pipeline used in AWinK

AWinK excludes k -mers that appear less than five times because these are most likely induced by sequencing errors. Then, it selects the k -mers that have a frequency within the interval $[\mu_u - t\sigma_u, \mu_u + t\sigma_u]$. According to [21, 22], if $k = 21$ and $t = 3$, the k -mers within that interval are single-copy k -mers with high probability (Figure 4.8, step A).

B. Barcode reads. AWinK searches for exact occurrences of the *unikmers* in the reads T or their reverse complement. We call the set of *unikmers* in a read r (and their location) the *barcode* of r . The barcode is stored as a list of pairs (u, j) , where u is a *unikmer* in r and j is the location of u within r (see Figure 4.8, step B).

C. Assign reads to bins. In this step, AWinK assigns barcoded reads to bins based on their number of *unikmers*. First, it calculates the mean μ_c and standard deviation σ_c of the distribution of *unikmers* (reads containing up to 50 *unikmers* are excluded to avoid the out-of-distribution spike

which would bias the estimate for μ_c and σ_c). Given a read r that has c *unikmers*, (i) when $0 \leq c \leq 5$, r is assigned to an individual c -th bin, (ii) if $c > \mu_c + 3\sigma_c$, r is assigned to a single high-frequency bin, (iii) if $5 < c \leq \mu_c + 3\sigma_c$, r is assigned to a bin using the method described next. The *capacity* of each bin is lower bounded by a target coverage ratio D/s where D is the coverage depth of T and s is a *scaling factor*. First, AWinK creates a new bin and it assigns reads with $c > 5$ to it until the target coverage ratio D/s is reached. If the target coverage is reached before all reads with the current *unikmer* count are added to the current bin, the remaining reads with that *unikmer* count are assigned to the current bin as well. This ensures that reads with the same *unikmer* count c are always placed in the same bin and never split across different bins. AWinK then creates a new bin and assigns reads with the next higher *unikmer* count to it until all the remaining reads are assigned to their respective bins. (Figure 4.8, step C)

D. Select reads. AWinK constructs the set T' as follows. It first selects all reads from the high-frequency bin, defined as those with *unikmer* counts $c > \mu_c + 3\sigma_c$. Assuming that these reads provide a coverage depth D' , AWinK takes additional reads to account for $(D_t - D')/D$ bases from each of the remaining bins. Within each bin, reads are chosen in descending order of length. This phase begins with the 0-th bin which contains reads with no *unikmers* and proceeds sequentially through bins with progressively higher *unikmer* counts, excluding the high-frequency bin. (Figure 4.8, step D)

E. Assemble reads. The selected reads T' with coverage depth D_t are assembled using hifi-asm [26]. (Figure 4.8, step E)

Parameter	Value	Parameter	Value
Mean read length	11,500	Mean read accuracy	0.999
Standard deviation of read length	3,500	Standard deviation of read accuracy	0.001
Minimum read length	100	Minimum read accuracy	0.990
Maximum read length	50,000	Maximum read accuracy	1.000

Table 4.2: PBSim’s parameters and their optimal values

4.4 Experimental Setup

4.4.1 Synthetic Data Generation

We used PBSim [97] to generate synthetic HiFi reads. PBSim depends on eight user-defined parameters. To find the values that best simulate real HiFi reads, we used the YAK scores and the read lengths as evaluation metrics. We also used real HiFi reads with coverage depth $3,000\times$ for the genome of *B. divergens MOI* as the “ground truth.” For each set of parameter values, we generated synthetic HiFi reads using PBSim on the CCS model with read coverage of $3,000\times$ for *B. divergens MOI* and $1,000\times$ for *C. elegans*. PBSim synthetic HiFi reads had the highest similarity with the real HiFi reads (on the basis of YAK scores and read lengths) for the parameter values provided in Table 4.2. These were the parameters used in the experiments in Section 4.5.1.

4.4.2 Performance Metrics

When the reference genome was available, we measured the quality of the assembly G' by comparing its contiguity and completeness against the reference genome G . We recorded the total

assembly size, and the number and the size of each contig in G' . We also used QUAST [48] and minimap2 [75] to compare G' to G , and recorded the genome fraction, NG50 and NGA50 reported by QUAST and the sequence identity reported by minimap2. In the ideal case, the contig sizes of G' should be equal to the size of the chromosomes in G , the number of contigs should be equal to the number of chromosomes in G , the genome fraction and the sequence identity should be 100%. We used these metrics to compare the hifiasm assemblies produced from reads selected by various strategies, namely AWinK, highest QV, best YAK score, longest, and best Filtlong score.

When the reference genome was unavailable (i.e., when assembling the genome of a new species), we assessed contiguity and completeness by comparing G' to a genome map. For our experiments on the genome of *B. divergens MOI*, we evaluated the assembly using an optical map.

4.5 Result Analysis

4.5.1 Experimental Results on Synthetic Data

To generate synthetic data, we chose the assembly WBcel235 for *C. elegans* as the ground truth. We recall that the genome of *C. elegans* is about 100 Mb and has six chromosomes (five autosomes and one sex chromosome) and one mitochondrial genome.

We carried out the following experiment. (i) We generated a set of ultra-deep HiFi reads at coverage depth $D = 1000\times$ from WBcel235 using PBSim, (ii) we selected subsets of reads at coverage depth $\{20\times, 25\times, 30\times, 35\times, 40\times, 45\times, 50\times, 100\times, 150\times, 200\times\}$ using AWinK (by setting the scaling factor s to 10,000 we ensured that the lower bound on the capacity of each bin was $D/s = 0.1\times$, as explained in Section 4.3.4 Step C), (iii) we assembled each subset of selected reads using hifiasm with the default parameters. Figure 4.9 shows the size of the six longest contigs

on the top, and the number of contigs produced by each assembly on the bottom. The leftmost set of bars in Figure 4.9 (top) are the sizes of the six chromosomes of *C. elegans*. Observe that when the input reads had $20\times$ - $50\times$ coverage, the contigs were shorter than the reference. The contigs reached the correct sizes at $100\times$ coverage. Adding more reads, however, seemed to degrade the quality of the assembly. Similar observations were made on the number of contigs in Figure 4.9 (bottom). The reference genome has seven contigs (six chromosomes and one mitochondrial genome). Between $20\times$ and $30\times$ coverage, hifiasm produces fragmented assemblies. The smallest number of contigs was achieved at $50\times$ coverage (for assembly statistics including coverage depths between $50\times$ and $100\times$, see Table 4.3). However, by adding more reads, the number of contigs started to increase quickly and reached a very high level of fragmentation at $200\times$ coverage.

Figure 4.10 illustrates the genome fraction and the sequence identity of the assemblies produced from the AWinK selected reads at various coverage depths. Since assemblies produced from reads selected by AWinK at higher coverage depth ($\geq 100\times$) were found to be highly fragmented, we analyzed the length distribution of contigs shorter than 50 kb and shorter than 100 kb. Table 4.4 shows that over 90% of these short contigs are indeed unassembled HiFi reads. To avoid over-inflating the genome fraction and sequence identity statistics, in the following analyses we have considered only contigs longer than 100 kb. Observe that at $20\times$ coverage, both genome fraction and sequence identity were not good. The genome fraction, however, was almost perfect already at $25\times$ coverage and remained that high throughout. The sequence identity value kept improving until $60\times$ coverage and stayed that way until $150\times$ coverage. The assembly at $100\times$ coverage achieved the highest sequence identity of 100%. Adding more reads started to degrade the sequence iden-

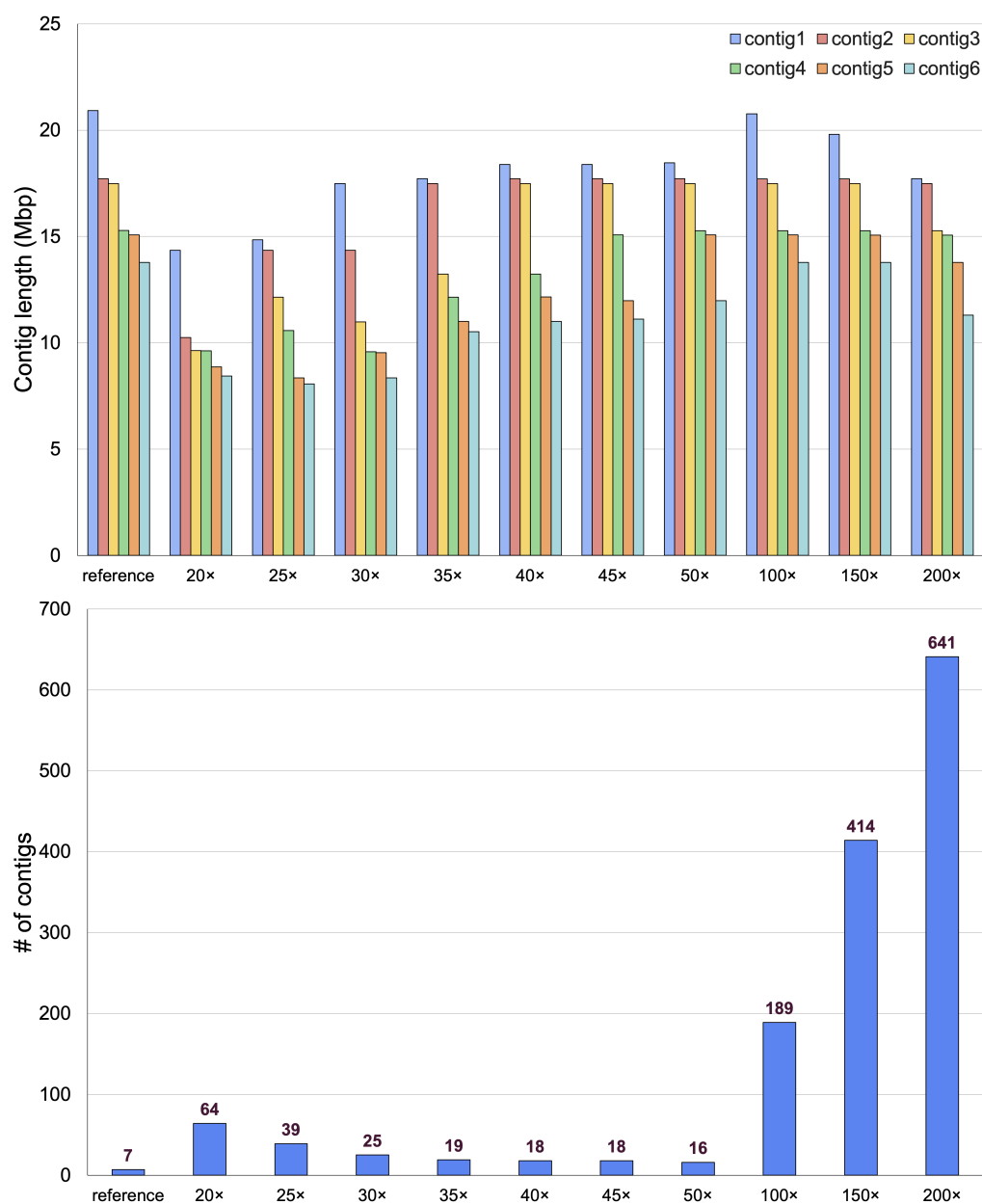


Figure 4.9: Length of the six longest assembled contigs (top) and number of assembled contigs (bottom) produced by hifiasm on subset of reads selected by AWinK at different sequencing depths from synthetic HiFi reads with sequencing depth of 1,000 $\times$ for the genome of *C. elegans*; numerical values are available in Table 4.3

Assembly	contig1	contig2	contig3	contig4	contig5	contig6	# of contigs	Total length (bp)
Reference	20,924,180	17,718,942	17,493,829	15,279,421	15,072,434	13,783,801	7	100,286,401
20×	14,353,301	10,245,828	9,636,687	9,618,046	8,864,646	8,439,682	64	94,317,956
25×	14,842,596	14,353,669	12,142,845	10,579,969	8,345,395	8,066,091	39	100,062,442
30×	17,492,666	14,353,670	10,983,157	9,582,624	9,535,291	8,345,394	25	100,130,912
35×	17,718,886	17,492,967	13,232,487	12,142,865	11,008,444	10,522,223	19	100,166,017
40×	18,389,194	17,718,886	17,492,965	13,232,438	12,154,690	11,009,585	18	100,244,671
45×	18,389,193	17,718,886	17,493,058	15,071,354	11,977,560	11,113,608	18	100,435,205
50×	18,460,696	17,718,888	17,493,086	15,276,212	15,071,363	11,977,763	16	100,481,717
60×	18,389,436	17,701,132	17,493,086	15,276,206	15,071,359	13,781,138	37	101,063,716
70×	18,389,436	17,710,003	17,493,057	15,276,213	15,071,359	13,783,520	93	102,553,110
80×	18,389,641	17,718,525	17,493,057	15,252,239	15,071,357	13,783,727	109	102,668,652
90×	18,389,641	17,708,807	17,490,415	15,276,211	15,071,357	13,783,727	98	102,153,158
100×	20,774,037	17,707,649	17,493,695	15,276,312	15,071,358	13,783,791	189	104,474,733
150×	19,800,194	17,711,149	17,488,195	15,276,393	15,062,976	13,783,793	414	109,417,601
200×	17,718,525	17,488,195	15,276,394	15,063,184	13,783,791	11,299,277	641	115,281,081

Table 4.3: Length of the six longest assembled contigs, number of assembled contigs and the total assembly length produced by hifiasm on subset of reads selected by AWinK at different sequencing depths from synthetic HiFi reads with sequencing depth of 1000× for the genome of *C. elegans*

Depth	# of contigs	Length < 50 kb		Length < 100 kb	
		# of contigs (%)	avg. length (bp)	# of contigs (%)	avg. length (bp)
100×	189	175 (92%)	21,730.80	183 (97%)	23,868.30
150×	414	390 (94%)	21,213.00	407 (98%)	22,919.10
200×	641	601 (94%)	21,588.30	630 (98%)	23,516.70

Table 4.4: Average length of short contigs produced by hifiasm on subset of reads selected by AWinK at sequencing depths = $\{100\times, 150\times, 200\times\}$ from synthetic HiFi reads with sequencing depth of $1000\times$ for the genome of *C. elegans*

tity. Figure 4.11 illustrates the NG50 and NGA50 scores for the same set of assemblies where we observed a similar trend in assembly quality with respect to the read coverage depth.

Figure 4.12 compares the performance of AWinK to the other selection strategies, namely best average QV scores, best YAK scores, longest, and best Filtlong scores at coverage depths $\{20\times, 30\times, 40\times, 50\times, 100\times\}$. Observe that the contig sizes varied unpredictably with increasing coverage, similar to what we have presented in Section 4.2. The assemblies that best matched the reference were QV $20\times$, QV $40\times$, Filtlong $20\times$, and AWinK $100\times$.

Table 4.5 shows the genome fraction, sequence identity, NG50 and NGA50 of the various selection strategies. Observe that even though AWinK produced a higher number of contigs, it achieved the highest sequence identity among all while maintaining a comparable genome fraction.

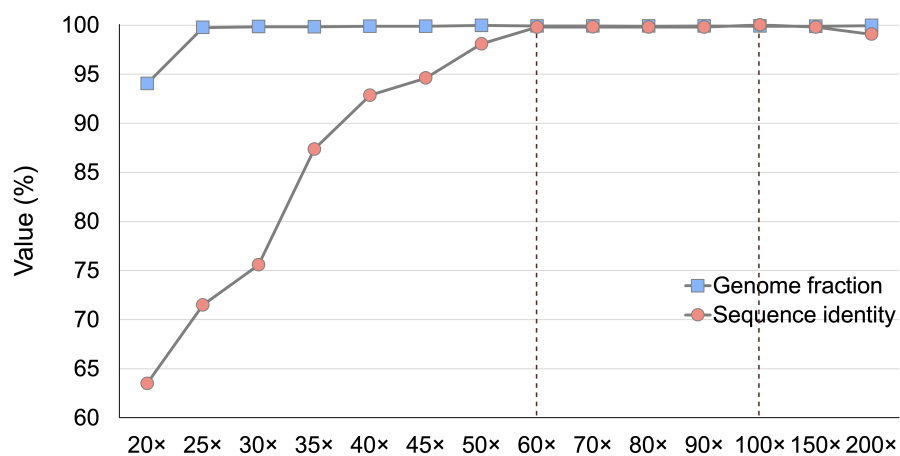


Figure 4.10: The genome fraction and the sequence identity of the assemblies produced by running hifiasm on various subset of reads selected by AWinK at different coverage depths from a set of PBSim simulated HiFi data with sequencing depth of 1,000 $\times$ for the genome of *C. elegans*

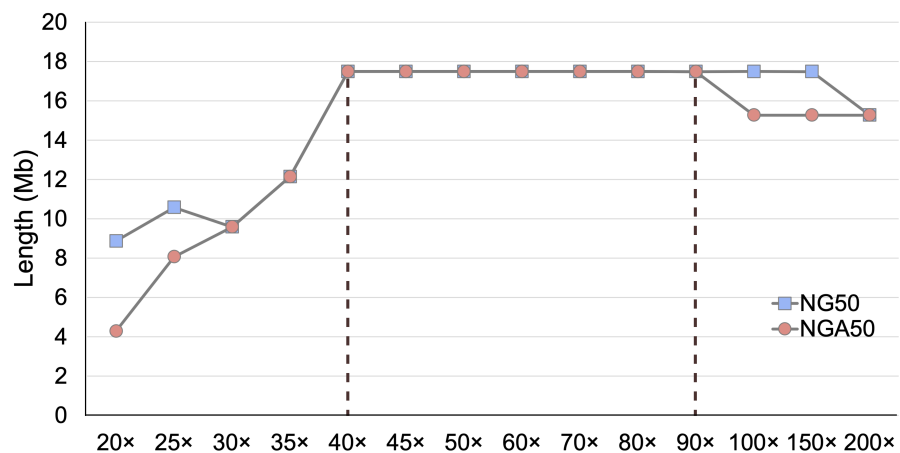


Figure 4.11: NG50 and NGA50 of the assemblies produced by running hifiasm on various subset of reads selected by AWinK at different coverage depths from a set of PBSim simulated HiFi data with sequencing depth of 1,000 $\times$ for the genome of *C. elegans*

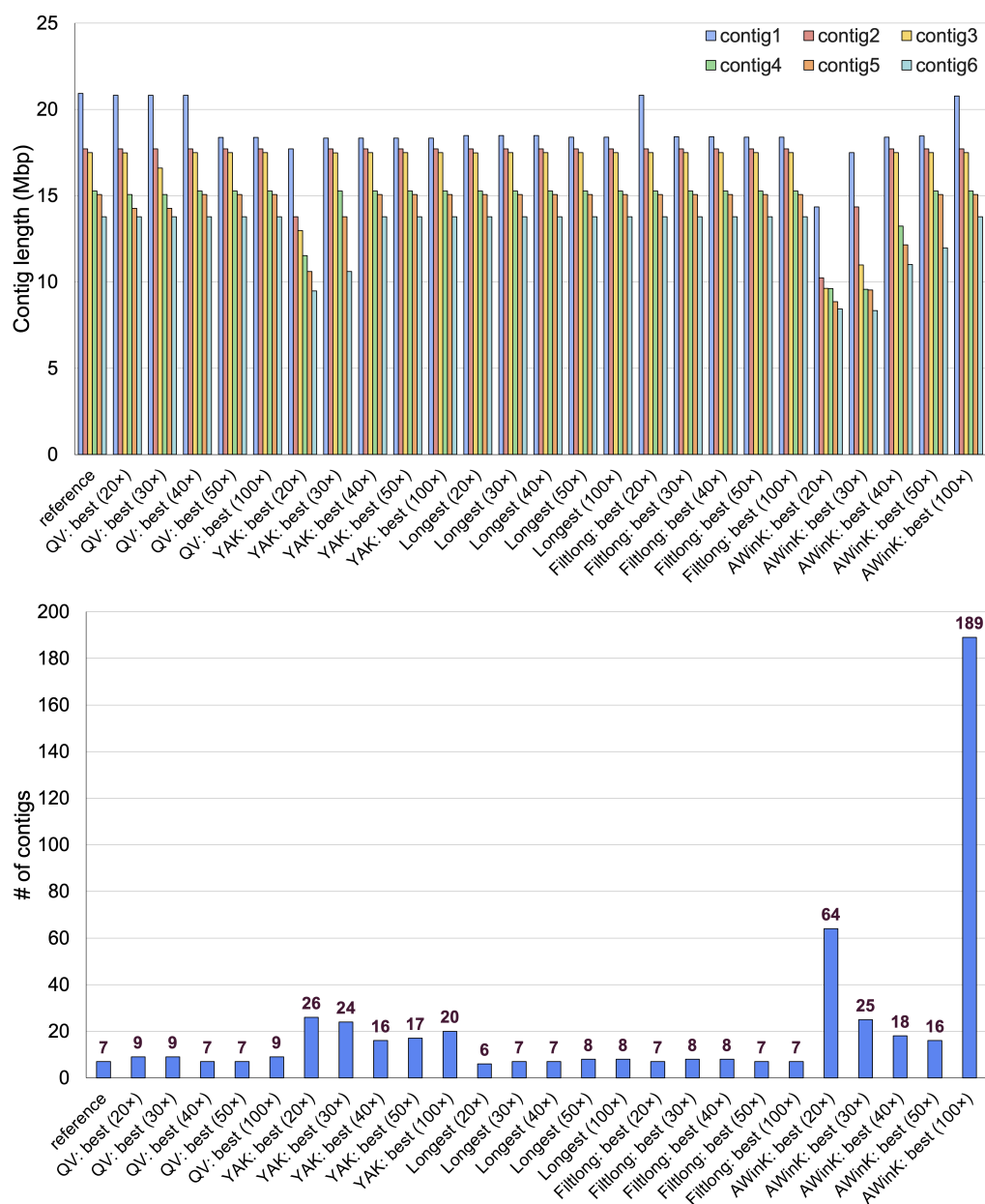


Figure 4.12: Length of the six longest assembled contigs (top) and number of assembled contigs (bottom) produced by hifiasm on subset of reads selected by different strategies and coverage depths from a set of PBSim simulated HiFi data with sequencing depth of 1,000 $\times$ for the genome of *C. elegans*

Method	Depth	Performance metrics			
		Genome fraction (%)	Seq. identity (%)	NG50	NGA50
SLICES	50×	99.915	99.806	17,493,698	17,493,698
	100×	99.941	99.806	17,493,661	17,493,649
RANDOM	50×	99.859	99.906	17,493,493	17,493,493
	100×	99.910	99.764	17,493,731	15,279,061
QV	50×	99.904	99.904	17,492,865	17,492,865
	100×	99.914	99.807	17,493,621	17,493,621
YAK	50×	99.876	99.753	17,493,202	15,274,813
	100×	99.895	99.761	17,493,202	15,276,768
Longest	50×	99.903	99.818	17,492,955	17,492,955
	100×	99.906	99.821	17,492,985	17,492,985
Filtlong	50×	99.924	99.924	17,492,983	17,492,983
	100×	99.926	99.926	17,493,188	17,493,188
AWinK	50×	99.975	98.088	17,493,086	17,493,086
	100×	99.905	99.994	17,493,695	15,276,312

Table 4.5: Comparing the metrics for hifiasm assemblies produced by various read selection strategies for *C. elegans*; numbers in bold indicate the best metric

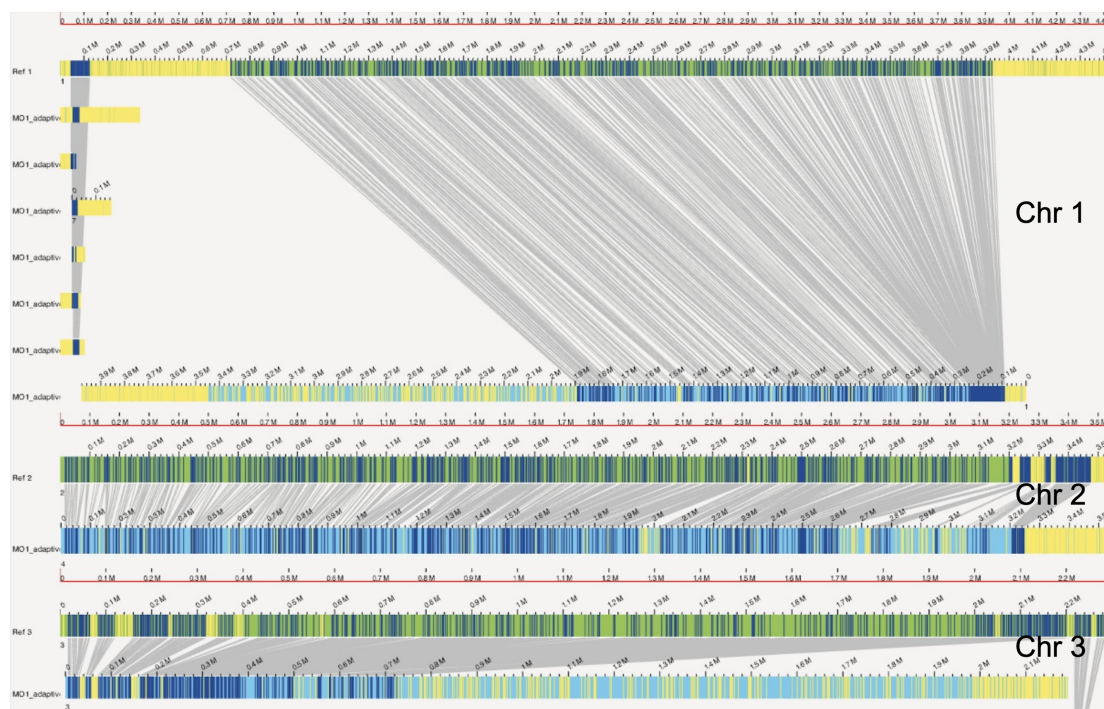


Figure 4.13: A visualization of the alignment between the three longest contigs of the hifiasm assembly from AWinK selected reads at $50\times$ coverage and the optical map for *B. divergens* MO1

4.5.2 Experimental Results on Real Data

On this last set of experiments, we used real ultra-deep HiFi reads at $D = 3000\times$ coverage for the genome of *B. divergens MOI*, obtained from [120]. The authors of [120] claim that *B. divergens MOI* has three chromosomes, but they are unsure about their sizes.

We carried out the experiments as follows. We used AWinK to select subsets of reads at various coverage depths, namely $\{20\times, 25\times, 30\times, 35\times, 40\times, 45\times, 50\times, 100\times, 150\times, 200\times\}$ (we set the scaling factor s to 1000 to obtain a lower bound on the capacity of each bin to $D/s = 3\times$ as explained in Section 4.3.4 Step C). Then, we assembled each subset of reads using hifiasm with its default parameters.

To measure the quality of the assembly, we aligned the three longest contigs of each assembly to the optical map molecules for *B. divergens MOI*, obtained from [120]. Figure 4.13 shows the alignment obtained using RefAligner. Observe that there is strong agreement between the second hifiasm contig and the second optical molecule (labeled Chr2). However, the other two contigs show lower levels of agreement with the optical molecules.

4.5.3 Runtime and Memory Usage

Table 4.6 summarizes the runtime and peak memory usage for assembling the *B. divergens MOI* and *C. elegans* genomes from ultra-deep HiFi sequencing reads with coverage depths of $3000\times$ and $1000\times$, respectively, using AWinK to select subsets of reads at target coverages of $50\times$ and $100\times$.

For *B. divergens MOI* (genome size ~ 10 Mb), assembling the AWinK-selected reads with hifiasm at $50\times$ and $100\times$ coverage takes approximately 3.5 hours and uses a peak of 30.62 GB of memory. For *C. elegans* (genome size ~ 100 Mb, $\sim 10\times$ larger), the same process takes

Organism		<i>B. divergens MOI</i>		<i>C. elegans</i>	
Input HiFi read depth		3000×		1000×	
Genome size		~10 Mb		~100 Mb	
Target depth		50×	100×	50×	100×
Runtime (minutes)	Step A	3.06		56.12	
	Step B	169.15		650.4	
	Step C + D	18		62.56	
	Step E	10.32	25.97	47.7	134.45
	Total	200.52	216.17	816.78	903.53
Peak memory usage (GB)		30.62		117.22	

Table 4.6: Runtime and Peak memory usage for hifiasm assemblies produced by AWinK selected reads of *B. divergens MOI* and *C. elegans* for the target coverage depths of 50× and 100×

14–15 hours with a peak memory usage of 117.22 GB. Both runtime and memory for *C. elegans* are about 3.5–4× higher than for *B. divergens MOI*, matching the ratio of the input ultra-deep HiFi read volumes, indicating that AWinK scales linearly with input data size.

Notably, most of the runtime is consumed by Step B of AWinK, which performs read barcoding. This step is currently single-threaded, and we are actively working to parallelize it to improve runtime scalability.

4.6 Discussion

To the best of our knowledge, this is the first study that addresses the problem of *de novo* genome assembly for ultra-deep sequencing HiFi reads. We are the first to report that current state-of-the-art assemblers for HiFi reads (e.g., HiCANU and hifiasm) struggle when given in input ultra-deep

sequencing data. This serious limitation motivates the study of the selection problem, i.e., how to select a subset of the reads that would produce the most contiguous and complete assembly.

We have also shown here that if there was an “oracle” that knew the location of every read, it would be possible to solve the selection problem optimally by choosing the reads on a minimum tiling path that achieved the desired coverage. When we fed the reads on the minimum tiling path into hifiasm, it produced an almost perfect assembly with a read coverage as low as $10\times$. Since the oracle does not exist, we proposed to exploit single-copy k -mers (*unikmers*) to approximate the minimum tiling path. Since *unikmers* occur only once in the genome, it would seem that selecting the subset of reads that cover all the *unikmers* could approximate the minimum tiling path. We discovered, however, that *unikmers* are not uniformly distributed across the genome, likely due to the repetitive regions in the genome.

Our proposed heuristic called AWin*K* selects a subset of reads so that distribution of *unikmer* frequency in the subset is as similar as possible as the distribution of *unikmer* to the whole ultra-deep dataset. Our experiments on synthetic HiFi reads showed that the reads selected by AWin*K* produced assemblies with good contiguity and completeness when the coverage was $50\times$ - $100\times$. However, increasing the coverage beyond $100\times$ degraded the assembly quality. Compared to six alternative selection strategies, AWin*K* produced assemblies that had the highest sequence identity and genome fraction when aligned to the reference genome. AWin*K* is the first attempt at solving the read selection problem. The experimental results on synthetic reads show that there is still a lot of work to do to better approximate the minimum tiling path.

4.7 Software Availability

AWinK is available from GitHub (<https://github.com/sakshar/AWinK>) and as Supplemental Code in the Appendix [B](#).

Chapter 5

Collaborative Projects

5.1 Prediction of DNA Methylation with Long-Range State-Space Models

5.1.1 Introduction

DNA methylation is an epigenetic marker which involves the addition of a methyl group to cytosine bases. It plays a critical role in regulating gene expression and other important cellular molecular processes [61]. In vertebrates, DNA methylation is mostly found at cytosines that are followed by a guanine, which is referred to as the CG context. In plants instead, methylation can occur at cytosines in any context, i.e., CG, CHG, and CHH, where $H=\{A,T,C\}$ [123, 73, 35]. Methylation patterns influence critical cellular processes such as stress responses, phenotypic plasticity, and trait inheritance [140, 12]. The traditional method of identifying cytosine methylation involves sequencing DNA after being treated with sodium bisulfite (BS-Seq). BS-Seq can be costly for large eukaryotic genomes because the sequencing coverage has to be sufficiently deep to enable

one to determine the methylation levels with statistical confidence. Typically, 5-10 reads covering a cytosine are necessary to reliably determine the methylation levels. Due to the non-uniformity of Illumina sequence coverage, even if the average sequencing depth is higher than $10\times$, a small but significant fraction of the cytosines will not have sufficient read coverage. Thus, there is a need to develop computational methods to predict the methylation levels for some of the cytosines either from the surrounding DNA sequence, or the methylation levels of nearby cytosines, or other local epigenetic signals. The prediction of methylation from the primary DNA sequence is the most convenient because it does not require any additional data or wet lab experiment, but it is also the most computationally challenging due to complex sequence-context dependencies and long-range interactions across the genome.

To address the need to impute the methylation levels for the cytosines with low sequencing coverage, several deep learning methods have been proposed in the literature to predict the methylation status from the sequence or the methylation levels of nearby cytosines [137, 3, 41, 141]. These methods include (i) CpGenie, which is a classifier based on a convolutional neural network (CNN) [138], (ii) DeepCpG, which uses a CNN and recurrent neural network [6], (iii) AMPS, which is another CNN-based classifier oriented to predict methylation in plants [113], (iv) scGPT, which is a model based on transformers and trained on single-cell data [31], (v) iDNA-ABF, which uses a bidirectional transformer (BERT) [60, 32, 62], and (vi) BERT-5mC, which also uses a BERT architecture [131]. Transformer-based methods have also been employed for predicting RNA methylation from sequence data [121]. We should note that, with the exception of AMPS, all of these DNA methylation prediction methods have focused on CG methylation in humans or other vertebrates. To the best of our knowledge, only AMPS can predict methylation in all CG, CHG, and

CHH contexts, for the benefit of non-vertebrate species like plants. Based on the results in these studies, transformer-based models appear to produce the most accurate predictions from the DNA sequence, which we attribute to their powerful self-attention mechanisms. However, the attention mechanism relies on computing the relationship between every pair of tokens in a sequence; thus it incurs quadratic time complexity with respect to sequence length during both training and evaluation, making it inefficient for long DNA sequences [127, 65]. Since methylation potentially involves interactions across thousands of base pairs, there is a need for models that can efficiently capture long-range dependencies without excessive computational overhead [67, 10]. State-space models (SSMs), traditionally used for time series modeling, have recently been explored as an alternative to transformers for sequence-based tasks [47]. Unlike transformers, which use attention to explicitly model interactions between all sequence positions, SSMs use state representations to propagate information through structured recurrence. As shown by the Structured State Space sequence model and Hungry Hungry Hippos, it is possible to achieve competitive performance compared to transformers while demanding lower computation and memory costs [46, 43]. HyenaDNA adapts the SSM-based Hyena architecture to achieve subquadratic complexity in genomic sequence modeling [94, 99]. HyenaDNA employs long convolutions and data-controlled gating mechanisms to implicitly capture long-range dependencies as a replacement for the more computationally expensive attention mechanism of transformers. This architecture is particularly beneficial for genomic tasks, where regulatory elements can influence gene expression over thousands of base pairs. By leveraging memory-efficient computation over long context lengths, HyenaDNA has shown strong performance in long-range predictive tasks, e.g., chromatin profile prediction, species classification, among others.

In this work, we focus our attention on plant genomes due to their more diverse type of methylation contexts compared to mammalian genomes [125, 39]. Since plants have significantly more non-CG methylation than mammals, predictive methods trained on plants are more general compared to methods trained only on mammalian genomes. For the same reason, we used AMPS [113] as our performance benchmark because it has demonstrated strong predictive performance across all CG, CHG, and CHH contexts.

In this section, we investigate for the first time the performance of HyenaDNA in predicting DNA methylation from the primary DNA sequence, taking advantage of its ability to model long-range genomic interactions while maintaining computational scalability. We train the Hyena architecture using the DNA sequence surrounding methylated and unmethylated cytosines in six plant species. As it was done in AMPS, we predict cytosine methylation in all contexts, i.e., CG, CHG, and CHH. However, our architecture uses the DNA sequence exclusively, while AMPS also takes advantage of genomic annotations. We investigate the performance of three model sizes for HyenaDNA by pretraining the HyenaDNA framework on the genomes of the six plant species. Finally, we compare the predictive accuracy of HyenaDNA against other published methods, and we also examine the performance when predicting methylation across contexts and across species. The ability of HyenaDNA to “transfer” predictive knowledge from one species to another would be very valuable, because obtaining high-quality methylation data for a new species through bisulfite sequencing is laborious and expensive.

5.1.2 Problem Formulation

Let a genome be represented as a sequence G over the alphabet $\{A, C, G, T\}$. Given a cytosine site $c \in G$, we denote by $S(c)$ the subsequence of length L centered at c . We define $M(c) \in \{0, 1\}$ as

the methylation state of cytosine c , where $M(c) = 1$ if c is methylated, and $M(c) = 0$ otherwise. The problem can thus be expressed as predicting the mapping $f : S(c) \rightarrow M(c)$. In other words, the objective is to design a system which predicts the methylation state of a cytosine given the DNA sequence in the window containing the selected cytosine at its center.

5.1.3 Methods

Our approach follows a two-stage framework of pre-training and fine-tuning using HyenaDNA. The pre-training input consists of genomic DNA sequences. During pre-training, we train a HyenaDNA backbone of two Hyena layers using a masked language modeling objective. In the fine-tuning stage, we pair the genomic DNA sequences surrounding cytosines with binary methylation labels for CG, CHG, or CHH contexts as inputs and use a decoder layer connected to the backbone to perform the binary classification of the methylation status. We use balanced sampling of methylated and unmethylated sites and an 80:10:10 train/validation/test split.

Since the original HyenaDNA foundation model was trained on the human genome, we retrained our instances of HyenaDNA from scratch, and then fine-tuned them for the methylation prediction task. Our HyenaDNA models were trained on the genomes of tomato (*Solanum lycopersicum*), thale cress (*Arabidopsis thaliana*), rice (*Oryza sativa*), cowpea (*Vigna unguiculata*), cucumber (*Cucumis sativus*), and common liverwort (*Marchantia polymorpha*).

For each species, we tested three model sizes for HyenaDNA, namely 128k, 436k, and 940k parameters. These model sizes correspond to two Hyena layers with a width of 64, 128, and 192, respectively (as shown in Table 5.1). The original HyenaDNA study describes experiments using up to eight Hyena layers, but we chose a smaller depth for our models due to the relatively

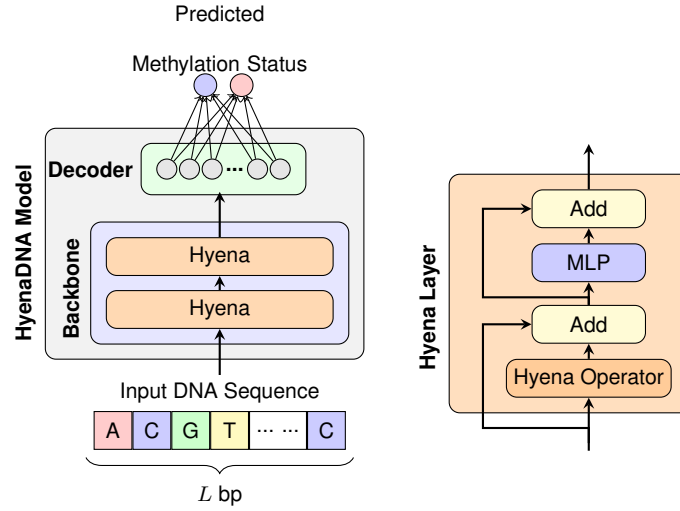


Figure 5.1: The HyenaDNA model structure; two Hyena layers constitute the architecture backbone (pre-training); a downstream decoder layer is used for classification (fine-tuning); input sequences are processed using a single-nucleotide tokenizer; the right side shows the components of the Hyena layer: the Hyena operator combines long convolutions with element-wise gating, and the long convolutions are implicitly parameterized by an MLP [94].

small context range used for our tasks.

For the training dataset, we fragmented the genomes into sequences of 6,400 bp, twice the length that would be used for the downstream classification task, to ensure that the models are provided adequate contextual information. We used a single nucleotide tokenizer (where each nucleotide A, T, C, G is treated as a discrete token, along with special tokens for sequence boundaries and padding) as this approach performed best on the classification tasks in [94, 99].

For the downstream task, we fine-tuned the base species-specific models to predict the methylation status in the contexts of CG, CHG, and CHH. We first extracted genomic sequences of length 3,200 bp around each methylated/unmethylated cytosine for the selected context (1,600 bp

	HyenaDNA-128k	HyenaDNA-436k	HyenaDNA-940k
Model size	128k	436k	940k
Model width	64	128	192
Decoder head size	130	258	386
Architecture	Two Hyena layers + MLP decoder layer		
Optimizer	AdamW		
Batch size	128		
Tokenization	single nucleotide		

Table 5.1: Hyperparameters of HyenaDNA

upstream and 1,600 bp downstream of the cytosine site). The window size of 3,200 bp was chosen because it was shown to be optimal in [113]. A linear classifier downstream of the HyenaDNA model was trained via supervised learning on the sequence dataset and their associated methylation labels. The architecture of our classifier is illustrated in Figure 5.1. We applied global average pooling over all sequence positions before passing the output to the decoder classification layer, as opposed to using the last token embedding. In both the pre-training and fine-tuning steps, we applied dropout with a probability of 0.1 to help mitigate overfitting and increase robustness against unseen sequences. In total, we produced 54 fine-tuned HyenaDNA models (6 species $\times$ 3 model sizes $\times$ 3 contexts).

Each dataset was curated to maintain an equal 50:50 ratio of methylated to unmethylated cytosines, with a total size of 500,000 samples for each dataset. The size of the dataset was chosen to be consistent with [113]. We extracted 250,000 DNA windows (3,200 bp long) centered

at a methylated cytosine in the genome and 250,000 DNA windows (3,200 bp long) centered at an unmethylated cytosine in the genome. If the number of methylated cytosines or the number of unmethylated cytosines was lower than 250,000, we took an equal number of samples from both classes. The data was split into an 80:10:10 ratio for the purposes of training, validation, and testing. We trained our models with a batch size of 128. Detailed statistics for each dataset are shown in Table 5.2. We trained using an adaptive learning rate schedule, with performance monitored on cross-entropy validation loss for early stopping with a patience of three epochs. We set a maximum limit of 100 epochs for both the pre-training and fine-tuning steps. For the largest models (HyenaDNA-940k) we used gradient checkpointing during training to reduce memory usage.

5.1.4 Experimental Results

We collected experimental results on three classification problems, namely (i) species and context-specific prediction, where the test data matches the species and context of the trained model, (ii) cross-context prediction, where the test data matches the species of, but differs in context from the trained model, (iii) cross-species prediction, where the test data is of a different species, but from the same context as the trained model. The experiments were performed for the six species listed in Table 5.3, and for the CG, CHG, and CHH contexts. We compared our experimental results with [113]. We did not compare any other published method, because in [113] it was shown that AMPS outperformed iDNA-ABF [62], MRCNN [124], CpGenie [138], and SMEP [132]. We compared our HyenaDNA architecture against AMPS without annotations (i.e., sequence only) and against AMPS with annotations.

species	genome size (Mb)	context	methyalted	unmethyalted	dataset size	dataset size (capped)	meth% (total)	acc (c/s-spec, 436k)
tomato	827.4	CG	2,101,929	236,762	2,338,691	473,524	89.9	0.87671
		CHG	1,610,667	962,623	2,573,290	500,000	62.6	0.83864
		CHH	403,972	16,969,766	17,373,738	500,000	2.3	0.69984
arabidopsis	119.1	CG	1,136,640	2,994,338	4,130,978	500,000	27.5	0.89672
		CHG	323,748	4,079,995	4,403,743	500,000	7.4	0.94078
		CHH	106,395	18,045,571	18,151,966	212,790	0.6	0.91245
rice	373.8	CG	7,036,195	5,978,404	13,014,599	500,000	54.1	0.88936
		CHG	3,082,917	8,597,826	11,680,743	500,000	26.4	0.84884
		CHH	900,875	35,690,939	36,591,814	500,000	2.5	0.88032
cowpea	518.6	CG	4,778,734	3,260,285	8,039,019	500,000	59.4	0.93168
		CHG	4,710,302	5,415,359	10,125,661	500,000	46.5	0.94468
		CHH	2,200,934	53,919,224	56,120,158	500,000	3.9	0.70364
cucumber	224.8	CG	1,267,547	974,108	2,241,655	500,000	56.5	0.89482
		CHG	581,242	1,805,491	2,386,733	500,000	24.4	0.94952
		CHH	557,331	11,313,108	11,870,439	500,000	4.7	0.84100
marchantia	225.8	CG	1,557,172	3,871,960	5,429,132	500,000	28.7	0.97476
		CHG	514,728	4,893,529	5,408,257	500,000	9.5	0.92930
		CHH	42,551	23,584,656	23,627,207	85,102	0.2	0.88365

Table 5.2: Data sets used for training the models in this study; data sets were capped to a maximum size of 500,000 with a 50:50 split between methylated and unmethylated samples; the right-most column shows the predictive accuracy of HyenaDNA-436k on each context and species

Common name	Accession number
Tomato (<i>Solanum lycopersicum</i>)	GCA_000188115.4 (<i>SL3.1</i>)
Thale cress (<i>Arabidopsis thaliana</i>)	GCA_000001735.2 (<i>TAIR10.1</i>)
Rice (<i>Oryza sativa</i>)	GCA_001433935.1 (<i>IRGSP-1.0</i>)
Cowpea (<i>Vigna unguiculata</i>)	GCA_004118075.2 (<i>ASM411807v2</i>)
Cucumber (<i>Cucumis sativus</i>)	GCA_000004075.3 (<i>Cucumber_9930.V3</i>)
Liverwort (<i>Marchantia polymorpha</i>)	GCA_003032435.1 (<i>Marchantia_polymorpha_v1</i>)

Table 5.3: Plant genomes used in this study

A. Species and Context-specific Performance. We first evaluated each of our fine-tuned HyenaDNA models on the test data set matching both the species and context on which the models were trained. Figure 5.2 shows that our HyenaDNA models (irrespective of their sizes) outperformed both sequence-only AMPS and sequence+annotation AMPS in 15 contexts/species combinations out of 18 (83.4%). The highest improvement over AMPS with annotations was achieved by HyenaDNA-940k on the rice genome for the CG context, with an accuracy increase of 10.5%. The highest accuracy was obtained on the marchantia data set (CG context) by the HyenaDNA-940k model with an accuracy of 97.7%. We observed that, in general, HyenaDNA models with a higher number of parameters yielded higher accuracies in most cases. In 17 contexts/species combinations out of 18, HyenaDNA-436k performed better than HyenaDNA-128k. In 12 contexts/species combinations out of 18, HyenaDNA-940k performed better than HyenaDNA-436k. There were a few cases in which the HyenaDNA models yielded a lower accuracy compared to AMPS, namely tomato (CHG and CHH context), and cowpea (CHH context). In the CHG context for tomato, HyenaDNA-436k

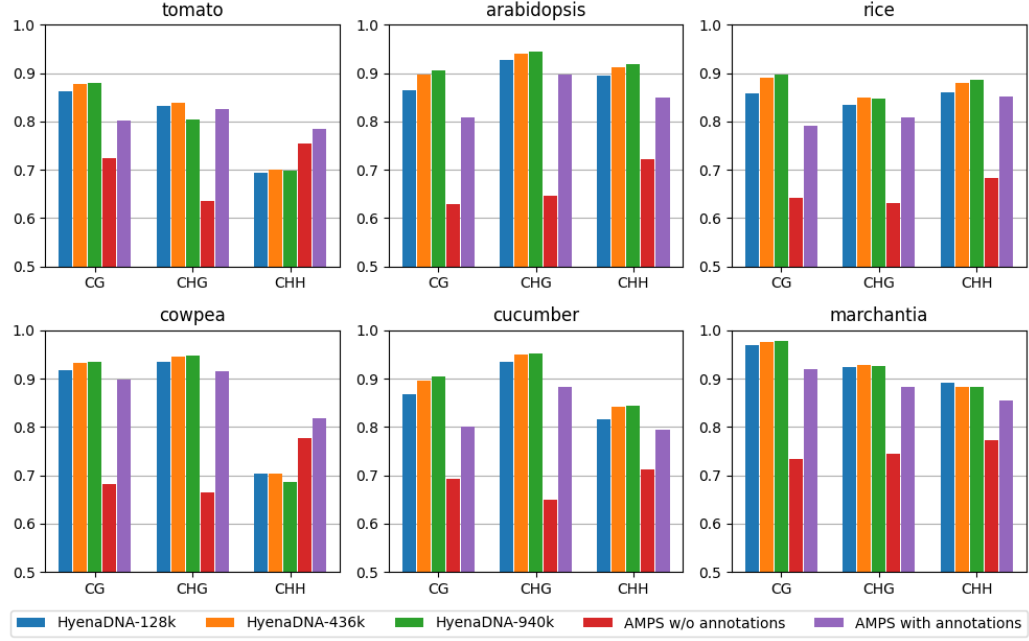


Figure 5.2: Context and species-specific accuracy comparisons between HyenaDNA and AMPS (sequence-only and sequence+annotation). In 15 contexts/species combinations out of 18 (83.4%), HyenaDNA models (irrespective of their sizes) outperformed AMPS with and without annotations.

performed similarly to AMPS with annotations, while HyenaDNA-128k performed marginally better and HyenaDNA-940k performed worse. In the context of CHH in tomato and CHH in cowpea, both AMPS models outperformed HyenaDNA by a significant margin. In terms of precision and recall, we observed a similar performance by HyenaDNA in these contexts as shown in Figures 5.3 and 5.4. We do not have a good explanation for the lower predictive performance of HyenaDNA in these three specific cases.

B. Cross-context Performance. For the cross-context problem, we tested our fine-tuned HyenaDNA models on data from the same species but in different contexts. In total, we evaluate 36

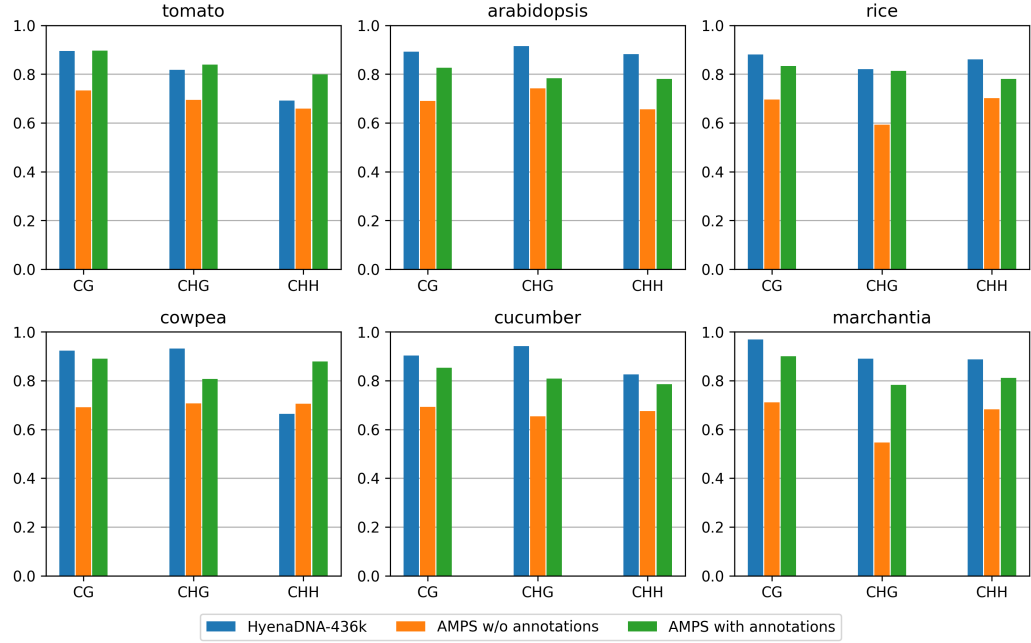


Figure 5.3: Comparing the precision of HyenaDNA-436k and AMPS for several contexts and species

cross-context cases (two test contexts for each of three training contexts across six species). The test accuracy is shown in Figure 5.5. We observed that the HyenaDNA-940k models outperformed AMPS with annotations in 23 out of 36 cases (64%) and AMPS without annotations in 35 out of 36 cases (97%). The accuracy improvements of HyenaDNA were particularly consistent in cross-context cases between CG and CHG, where the HyenaDNA-940k models outperformed both AMPS classifiers in 12 out of 12 cases (100%). The improvement of HyenaDNA models for the CHH context was less consistent. In particular, the cross-context accuracy for the CHH context in rice for all HyenaDNA models was lower than AMPS with annotations. Among the HyenaDNA models, the large models achieved the highest accuracy in 16 out of 36 cases (44%), the medium models in 13 out of 36 cases (36%), and the small models in 7 out of 36 cases (19%).

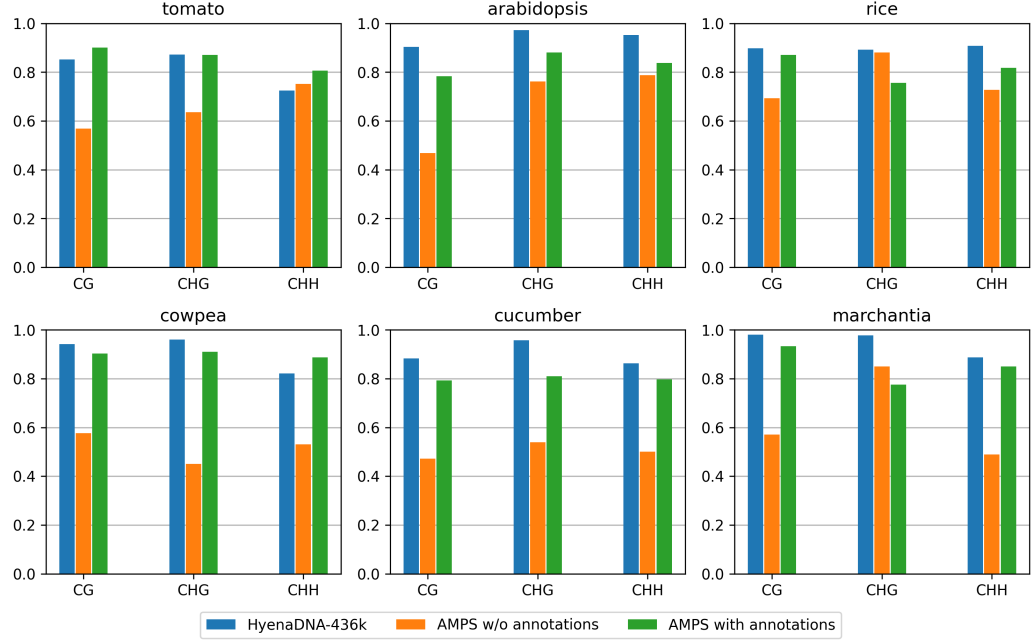


Figure 5.4: Comparing the recall of HyenaDNA-436k and AMPS for several contexts and species

C. Cross-species Performance. To investigate the ability of HyenaDNA to predict methylation across species, we evaluated our fine-tuned models on test datasets of the same context but in different species. We measured the accuracy across 90 cross-species cases (five test species, six training species across three contexts). Figure 5.6 shows that the performance of HyenaDNA on this problem was somewhat inconsistent. For instance, HyenaDNA-128k trained on the rice CHG context and tested on the cowpea CHG context yielded 82.0% accuracy; however, the same model trained on the marchantia CHH context and tested on the tomato CHH only achieved 48.8% accuracy. Surprisingly, some of these results were asymmetrical; for instance, HyenaDNA-128k trained on the tomato CHH context and tested on the marchantia CHH context set yielded a much higher accuracy of 62.8%. We observed that (i) the HyenaDNA-128k models outperformed AMPS with annotations in only 15 of 90 cases (17%), (ii) the HyenaDNA-436k models outperformed AMPS with anno-

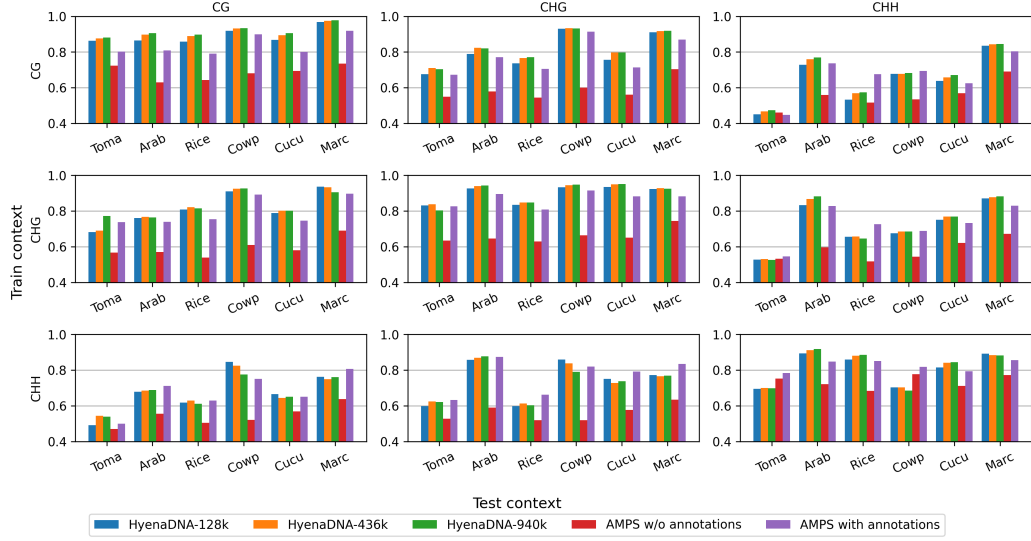


Figure 5.5: Cross-context testing accuracy comparisons between HyenaDNA and AMPS. HyenaDNA shows consistent improvement in accuracy scores over AMPS, especially in cross-context cases between CG and CHG, with the 940k model outperforming both classes of AMPS in 12 of 12 such cases (100%).

tations in 11 of 90 cases (12%), and (iii) the HyenaDNA-940k models outperformed AMPS with annotations in 12 of 90 cases (13%). Compared to AMPS without annotations, we observed an improved performance in 68 of 90 cases (76%) for the HyenaDNA-128k models, 75 of 90 cases (83%) for the HyenaDNA-436k models, and 75 of 90 (83%) for the HyenaDNA-940k models.

D. Runtime Analysis. To determine the runtime advantage of Hyena with respect to a transformer-based model, we carried out an additional set of experiments. First, we collected the runtime of HyenaDNA-436k for the inference of the methylation status on the cucumber genome (CG context) over various input sequences, ranging from 1,600 bp to 12,800 bp. Then, we ran another inference over the same inputs using a fine-tuned DNABERT-2 architecture. Both models were fine-tuned

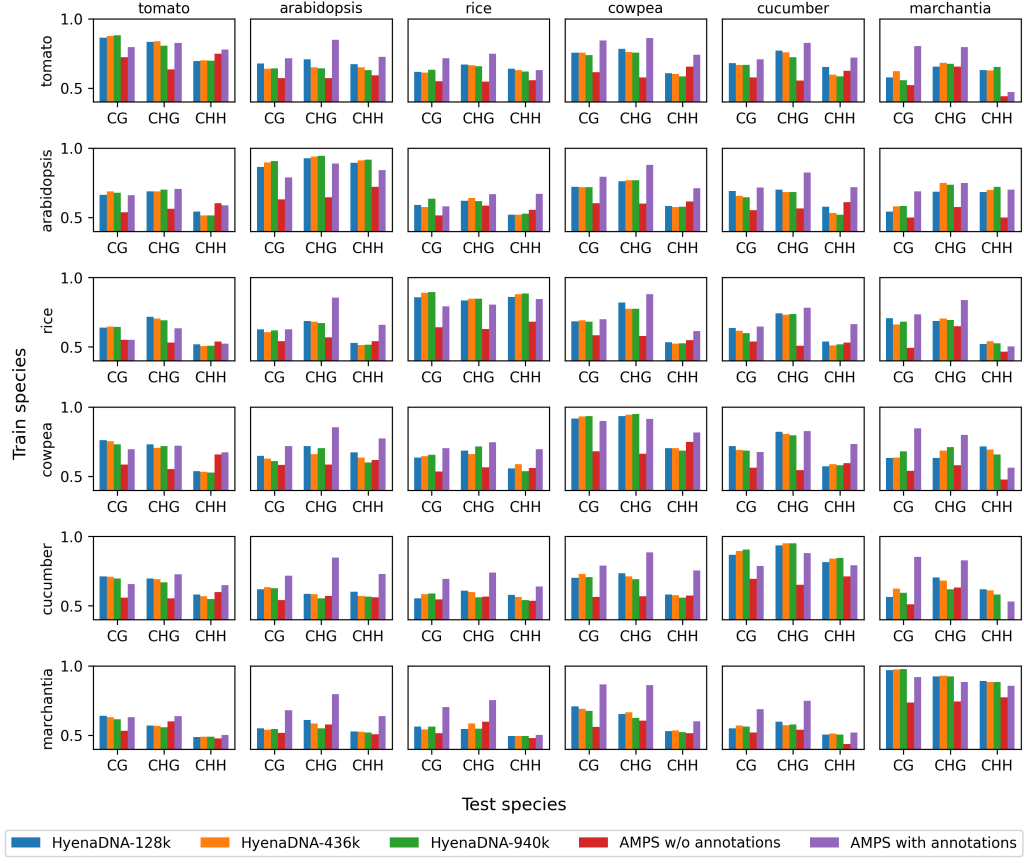


Figure 5.6: Cross-species testing accuracy comparisons between HyenaDNA and AMPS. HyenaDNA-940k outperforms AMPS without annotations in a majority of cases, but falls behind AMPS with annotations.

with input samples of 3,200 bp. Figure 5.7 shows that HyenaDNA scales significantly better than the DNABERT-2. When the input was 1,600 bp, HyenaDNA was $7.2\times$ faster than DNABERT-2. At 3,200 bp, HyenaDNA was $9.0\times$ faster. At 12,800 bp, HyenaDNA was $17.5\times$ faster. To complete this analysis, Figure 5.8 shows the predictive accuracy of DNABERT-2 and HyenaDNA on this data set. Observe that the predictive accuracy in both models peaks with inputs of 3-4 thousand base

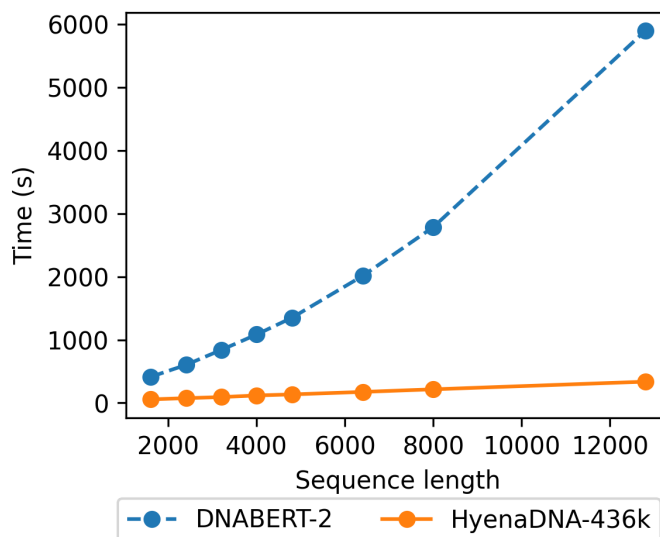


Figure 5.7: Comparing DNABERT-2 and HyenaDNA-436k’s runtime for different input sequence lengths.

pairs, which is expected since both models were fine-tuned on 3,200 bp long sequences.

E. Hyperparameter Analysis. Here we investigated the effects of some important hyper-parameters on the performance of HyenaDNA, namely (a) residual dropout, (b) training set size, and (c) window size.

To evaluate the effects of changes to the residual dropout, we chose the HyenaDNA-436k model pre-trained on the cucumber genome, and fine-tuned using a residual dropout of 0.1, in addition to the 0.1 embedding dropout used in the previous experiments. Figure 5.9 shows that in all cases there is a small drop in predictive accuracy when a residual dropout is incorporated.

To test the effects of changes to the training set size, we used the HyenaDNA-436k model pre-trained on the cucumber genome, and we fine-tuned using samples of 25%, 50%, 75%, and

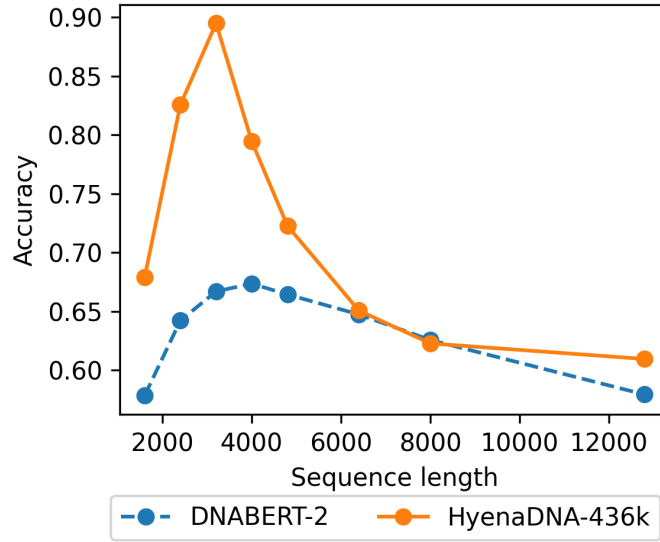


Figure 5.8: Comparing the predictive accuracy of DNABERT-2 and HyenaDNA-436k on the cucumber CG context with respect to different input sequence length (both models were fine-tuned using a window size of 3,200 bp)

100% of the original training set. As expected, Figure 5.10 shows a positive correlation between HyenaDNA’s accuracy and the training data set size across all contexts on the cucumber genome.

To investigate the effect of using a longer input window, we fine-tuned HyenaDNA-436k on all contexts and species using a 6,400 bp window size (double the size of the default window used in all previous experiments). Figure 5.11 shows that the 6,400 bp models performed worse than the 3,200 bp HyenaDNA models in 17 out of 18 cases, with the only exception of marchantia CHH (which shows a minimal improvement).

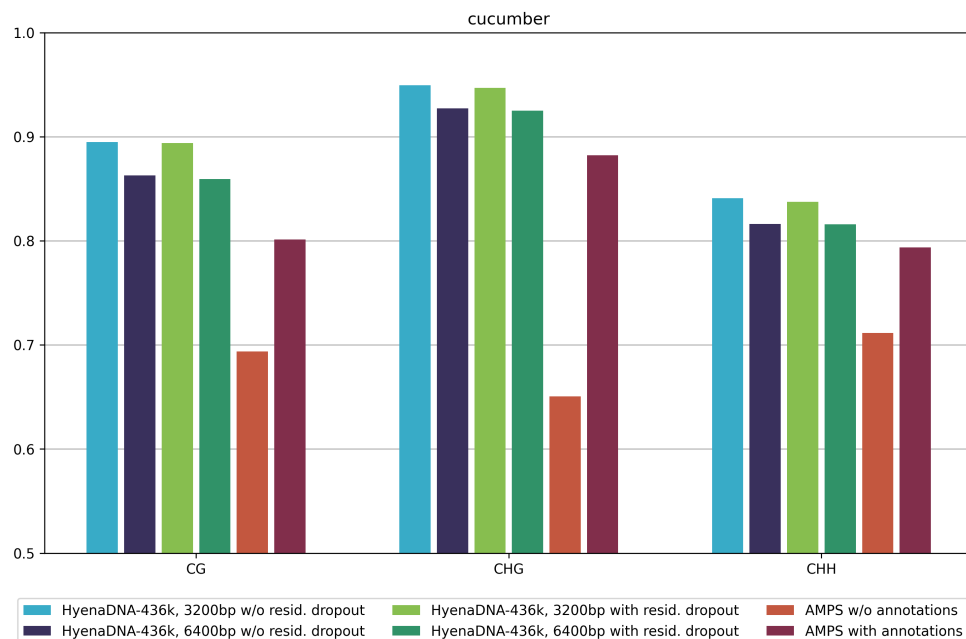


Figure 5.9: Comparing the accuracy of HyenaDNA-436k trained with embedding dropout only versus HyenaDNA-436k trained with both embedding and residual dropouts; the model was pre-trained on the cucumber genome and then fine-tuned using residual dropout of 0.1, in addition to 0.1 embedding dropout

5.1.5 Discussion

Our experimental results showed that the Hyena architecture was very accurate in predicting the methylation status in plants using exclusively the primary DNA sequence. Our Hyena-based models demonstrated significant accuracy improvements compared to the sequence-only AMPS classifiers for species/context-specific and cross-context problems. In some cases, we observed that larger Hyena-based models achieved a lower accuracy compared to smaller models, indicating possible issues with overfitting. An example of this behavior is the species/context-specific case for tomato in the CHG context, where the experimental results indicate that the optimal model size may be 128k

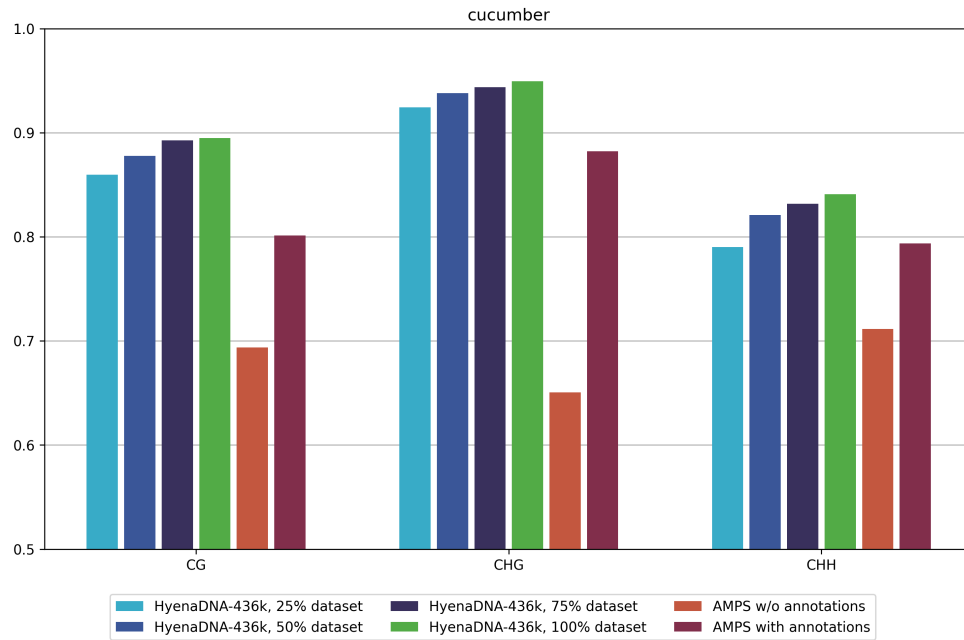


Figure 5.10: Comparing HyenaDNA-436k’s accuracy for various choices of the training dataset size (25%, 50%, 75%, and 100%) on the cucumber genome

or even lower. The large gap in prediction accuracy between AMPS without annotations and AMPS with annotations for this particular context suggests that the methylation state of a cytosine strongly depends on its genomic localization (i.e., within a gene or a repeat), which our HyenaDNA models do not use. We have considered the possibility that smaller training set sizes in certain contexts (i.e., fewer than 500,000 samples) may have contributed to differences in the classifier’s performance. However, we did not find a correlation between training set size and performance across our experiments. In cross-species situations, HyenaDNA performed on par with AMPS without annotations, but frequently underperformed compared to AMPS with annotations. This suggests that gene and repeat annotations could be key to accurately predicting methylation across different species. In cross-context situations, we found that HyenaDNA performed better than AMPS, even with anno-

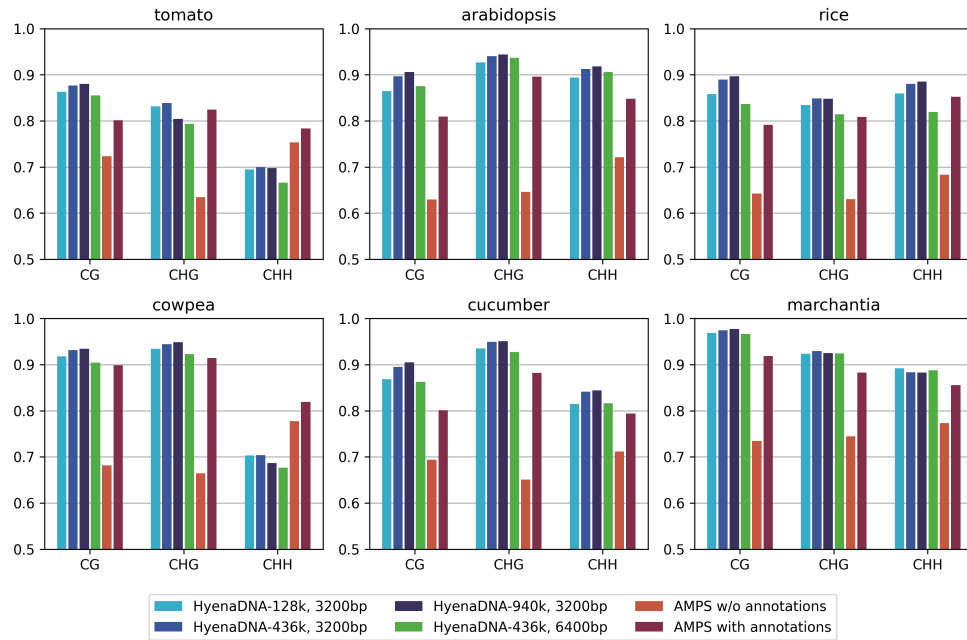


Figure 5.11: Comparing the accuracy of HyenaDNA-436k using a window of 3,200 bp vs. 6,400 bp for all contexts and species; AMPS' accuracy with and without annotations is also included as a baseline

tations, in most cases. Our findings support the observation reported in [113] that cross-context predictions between CG and CHG overall yield good accuracies, while this is not the case with cross-context predictions involving CHH.

While our study demonstrates the effectiveness of HyenaDNA for DNA methylation prediction, we acknowledge our method's performance is limited in cross-species contexts. This suggests that in general, models trained on one plant species do not readily transfer to others, and new species will likely require retraining with their own methylation data. However, we hypothesize that species which are evolutionarily closely related and share a high-level of sequence similarity may show methylation patterns similar enough to the extent that retraining would not be necessary.

Additionally, several questions remain open for further exploration. A primary concern about deep learning models is their black-box nature, which makes it difficult to explain the reasons behind the predictions of these models [83, 20]. In this study, understanding the workings of the models can provide deeper insights into the regulatory mechanisms that control DNA methylation in the cell. Therefore, priority should be given to the interpretability analysis of the HyenaDNA models, focusing on understanding the features and sequence motifs that contribute most to their predictions. To interpret the weights of the CNN-based AMPS classifiers, the authors used Grad-CAM [112] to identify motifs within the input sequence that are most informative with respect to the classification of methylation status. To the best of our knowledge, there are no equivalent tools for HyenaDNA architectures. Future work should involve the analysis of the weights of the HyenaDNA models and study their layer activations. Otherwise, model-agnostic approaches to interpretability analysis, such as SHAP [82] and LIME [107], could be used for this purpose.

The authors of AMPS have shown that incorporating gene and repeat annotations significantly increased the prediction accuracy. It is possible that adding gene and repeat annotations could increase the overall performance of HyenaDNA as well. Possible improvements in performance could be achieved by carrying out additional experiments on the hyperparameters and/or the training datasets. For instance, it could be worth investigating the effect of other sequence window sizes on the performance of the HyenaDNA classifier. This could help determine whether local sequence motifs or long-range interactions are more relevant for specific methylation contexts. It may also be interesting to explore the effects of pretraining on all six plant species, to test the possibility of creating a single foundational model. Along these lines, the recent Evo foundation model, trained on millions of prokaryotic and phage genomes, demonstrated impressive performance in predictive

and generative tasks [93, 100]. We believe that further studies on HyenaDNA may pave the way for similar results on plant species as well.

5.1.6 Software Availability

The code and its execution instructions are available from GitHub (<https://github.com/hfeng95/HyenaDNA-methyl>).

5.2 Visualizing the localization of multigene families with GFViewer

5.2.1 Background

A *gene family* is a set of genes which share a common evolutionary origin and exhibit structural and functional similarities. Gene families arise through gene duplication events, followed by divergence that allows them to acquire specialized or redundant functions. The study of gene families is crucial for various biological, medical, and computational research areas, as they provide insights into gene function, evolution, and potential therapeutic targets.

Additionally, analyzing the genomic localization of genes within a specific gene family is essential for downstream analyses. The position of genes within the genome can influence their regulation, evolutionary history, and functional interactions. For example, finding genomic localization of genes belonging to the same gene family may help in identifying gene clusters and understanding their regulation. It also facilitates synteny analysis of conserved gene order across species for evolutionary insights. These analyses can potentially reveal cis-regulatory elements influencing gene expression.

In this section, we introduce a tool called GFViewer for visualizing the localization of multi-gene families across the genome of a given organism. It requires a gene-family data file (.xlsx/.csv/.tsv) containing gene family and location information for each gene and a genome sequence file (.fasta/.fna/.fa) or a chromosome-lengths file (.txt) for the same organism as inputs. As output, it generates a plot illustrating the precise locations of genes belonging to the same gene-family using unique colors specific to the corresponding gene-family in a PDF file. The initial prototype of this software has already been used in studying the localization of multi-gene fami-

lies of different *Babesia* species [119, 120] for understanding their evolution, speciation, virulence factors, and drug targets.

5.2.2 Description

A. Installation. GFViewer can be installed on a local machine using Linux, Mac, or Windows, either from its [GitHub repository](#) or as [conda package](#), where installing locally from the GitHub source offers users maximum freedom of usage. GFViewer is also available as a [web application](#) with limited functionality. Figure 5.12 illustrates the web application’s homepage.

B. Input Files and Parameters. Table 5.5 summarizes the input files required for GFViewer along with the parameters and options to configure it. The detailed specification of the input files (three example sets are provided in the Appendix) is as follows:

- **Gene-family data file:** contains gene family and location information for each gene in the genome and is expected to be in .xlsx/.csv/.tsv format. The header row must contain *gene_id*, *gene_family*, *chromosome*, *start*, *end*, and *strand*. This naming convention for the header row must be followed strictly. Centromere data is optional and required only when plotting them. To plot the centromeres along with genes belonging to different gene-families, always put *centromere* under *gene_family* and 0 under *strand* columns. A toy example for gene-family data file is illustrated in Table 5.4.
- **Genome or chromosome-lengths file:** contains the genome sequence of the given organism in FASTA (.fasta/.fna/.fa) format or the chromosome ids with their lengths in TXT (.txt) format; *seq_id*, *seq_length* per line.

[Home](#) [Help](#) [About us](#)



Gene Family Data (.xlsx/.csv/.tsv) *:

Choose File
no file selected

Telomere Length (bp):

10000
⬆ ⬇ ⬆

Genome/Chromosome-Lengths File (.txt/.fasta/.fna/.fa) *:

Choose File
no file selected

Chromosomes Per Page:

3
⬆ ⬇ ⬆

Color Map File (.txt):

Choose File
no file selected

Legend Rows:

2
⬆ ⬇ ⬆

Plot Centromeres: ☐

Legend Location:

Lower
⬆ ⬇ ⬆

Submit

Example run (with default set-up):

test 1

Run

Quick links to example test data: [test1.zip](#), [test2.zip](#), [test3.zip](#)



2025 [Lonardi Lab](#)

Figure 5.12: The homepage of GFViewer web application

- Color map file:** contains the gene family (gf) ids with their color codes in TXT (.txt) format. The colors can be chosen from the provided color palette (see Supplementary Material) or provided with specific RGB values. If assigning colors based on provided color palette, structure the color map file as *gf_id, color_code* per line. For example, *gf_1, 1* will assign *red* to *gf_1*. If assigning custom colors, provide the corresponding RGB values for each gene family as *gf_id, r, g, b* per line where $r, g, b \in [0, 1]$. Note that GFViewer can handle plotting a maximum of 19 different gene-families with the default color palette where *color_code*: 20

gene_id	gene_family	chromosome	start	end	strand
gene_1	family_1	chr_1	4031	15104	-
gene_2	family_2	chr_2	24310	45159	+
...	...	...	...	...	...
centro.1	centromere	chr_1	82125	98476	0

Table 5.4: A toy example for gene-family data file

(grey) is reserved for centromeres. When a color map file is not provided, each gene family is assigned a color from the default palette in sequential order.

Among other parameters, a valid output directory must be provided via “-o” while executing locally. GFViewer offers some configurable features for the organization, orientation and placement location of legends. By default, legends are plotted as a separate file, but setting “-lpp” integrates them within the main output file. While plotting separately, “-or” can either be horizontal or vertical, horizontal being the default. When plotting the legends horizontally, the number of rows to arrange them can be specified via “-r” (default: 2). While plotting within the main file, legends can be placed at the bottom/top/left/right via “-l” parameter, bottom being the default. GFViewer also allows users to specify the number of chromosomes to be displayed per page through “-p” (default: 3) and setting “-conc” flag concatenates the pages into a single PDF file otherwise the plots are generated in separate PDF files by default. Moreover, users can specify the average length of telomeres in bp (default: 10000) for scaling purposes via “-t”. Lastly, setting “-cen” plots centromeres on the chromosomes provided the centromere locations through the gene-family data file.

Parameters/Options	Code	Description
gene family data file*	-d	a file (.xlsx/.csv/.tsv) containing gene family and location information for each gene
genome/chr-lengths file*	-g	a fasta file containing the genome sequence or a text file containing the chr ids with their lengths
output directory*	-o	path to the output directory
color map file	-c	a text file containing the gene family (gf) ids with their color codes or RGB values
legend location	-l	the location (upper/lower/left/right) of legends only when adding them to each page (default: lower)
legend orientation	-or	the orientation (horizontal/vertical) of legends only when plotting legends separately (default: horizontal)
telomere length	-t	the length of telomeres in bp (default: 10000)
# of chrs per page	-p	number of chromosomes to be plotted per page (default: 3)
# of rows in legends	-r	number of rows in the legends (default: 2)
centromeres	-cen	by default centromeres of the chromosomes are skipped
legends per page	-lpp	by default legends are plotted separately in a pdf file (legends.pdf)
concatenate pages	-conc	by default plots spanning multiple pages are generated in separate pdf files

Table 5.5: Summary of parameters and options available in GFViewer; (* indicates the required parameters)

C. Example. Figure 5.13 illustrates an example plot generated by GFViewer for the top 6 gene families of *Babesia divergens* MOI across its three chromosomes [120]. Observe that genes belonging to the same family tend to cluster together and the gene families are mostly placed near the telomeres.

D. Implementation. GFViewer is OS independent. It has been developed in Python 3 and distributed under MIT license. Flask web framework is used to make GFViewer accessible via a web interface. The web application is hosted on a Google Cloud VM instance, ensuring seamless deployment and accessibility. This setup allows users to interact with GFViewer's python library through a user-friendly web interface, facilitating efficient data processing and visualization. To

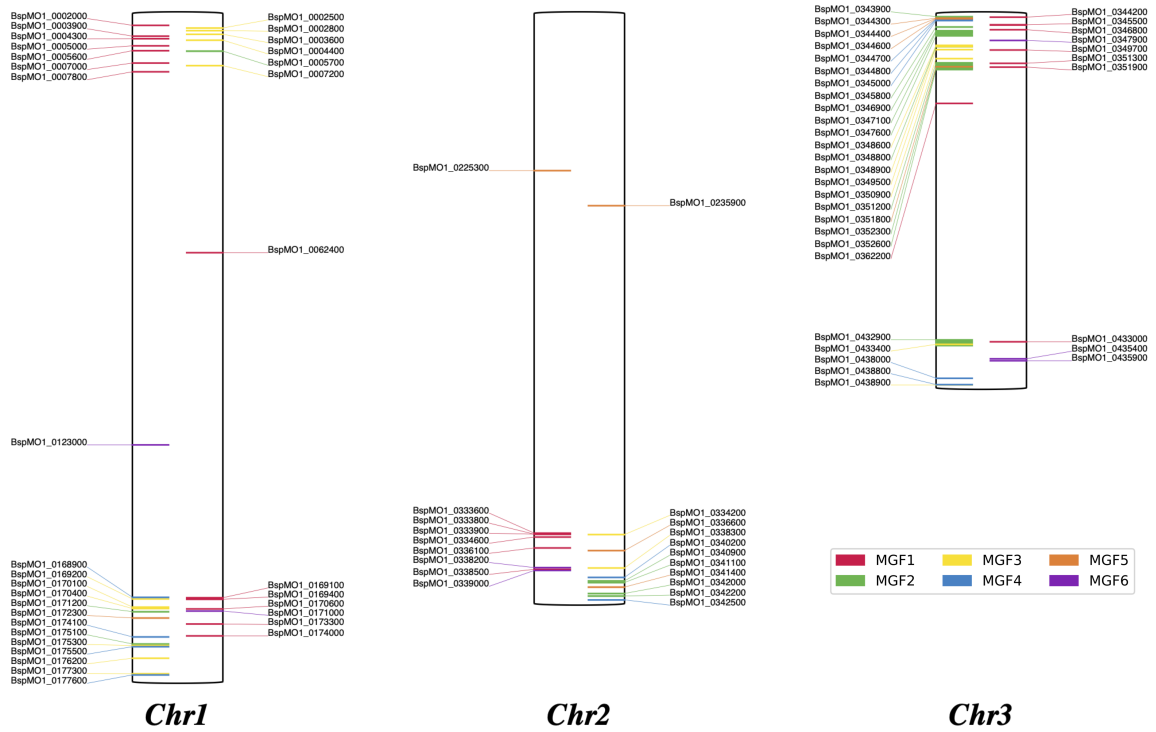


Figure 5.13: Multi-gene family localization plot for top-6 gene families of *Babesia divergens* MO1 across its chromosomes

ensure the security of the server machine, input validation is performed at both front and backend of the web application. GFViewer doesn't require any user registration step and anyone with access to a web browser can get its benefits. We have tested our web application on some of the most popular browsers including Chrome, Firefox, Safari and Edge. It ran smoothly in all of them. Moreover, to ensure the safety and privacy of data uploaded by users, GFViewer generates encrypted unique session tokens at each job request and processes all data under that token. On top of that, it runs a data clean up once every 24 hours.

Note that apart from generating the localization plots of multi-gene families, GFViewer outputs genbank (.gb) files for each chromosome utilizing the user provided gene-family data file.

Users running GFViewer locally on their machine, can access these intermediate data from the “tmp” folder inside the output directory.

5.2.3 Software Availability

- The source code and the usage instructions are available at <https://github.com/sakshar/GFViewer>.
- The conda package is available at [bioconda/gfviewer](https://anaconda.org/bioconda/gfviewer).
- The web application is available at <http://gfviewer.cs.ucr.edu>.

Chapter 6

Conclusions

In this dissertation, we have focused on developing algorithmic techniques to address two of the most challenging and least explored problems in genome assembly: (i) the assembly of highly repetitive genomic regions, and (ii) genome assembly in the presence of ultra-deep sequencing data (i.e., sequencing depth exceeding $1000\times$). We first reviewed the current state of sequencing technologies and computational genome assembly methods, highlighting their limitations with respect to these challenges. We then discussed how single-copy k -mers (*unikmers*), or rare k -mers, have been previously used for read-to-genome alignment and overlap detection among reads [122, 79, 88, 15, 17]. Building on these ideas, we designed novel algorithms that leverage *unikmers* in new ways to address the aforementioned assembly problems. In addition, we investigated computational modeling of DNA methylation across both CG and non-CG contexts. Finally, we introduced a visualization tool for analyzing the genomic localization of multigene families.

In Chapter 2, published in the Proceedings of the 14th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics (BCB'23) [21], we proposed

a reliable statistical framework for identifying *unikmers* from long-read sequencing data in the absence of a reference genome. Using PacBio HiFi reads, we demonstrated that for several values of k , selecting k -mers whose frequencies fall within three standard deviations of the mean coverage yields *unikmers* with approximately 99% precision and recall. We also introduced a new metric, called the *assembly score*, to jointly assess contiguity, completeness, and accuracy of an assembly when a reference genome is available. Furthermore, we presented RAMbler, a repeat assembler that exploits *unikmers* for reconstructing complex repetitive regions. Extensive experiments on synthetic datasets showed that RAMbler outperforms state-of-the-art assemblers on both fixed-length and variable-length repeats, while achieving performance comparable to hifiasm on repeats with copy number variation in synthetic diploid genomes.

In Chapter 3, published in *Genome Research* [22], we extended RAMbler to handle cycles in overlap graphs and analyzed the effects of reference bias and HiFi sequencing coverage bias on its performance in repetitive regions. We applied RAMbler to five selected repetitive regions (both centromeric and non-centromeric) from human Chromosomes 8, 19, and X, achieving telomere-to-telomere (T2T)-level assembly quality using HiFi data exclusively. We also evaluated RAMbler on three repetitive regions from *A. thaliana* Chromosome 5. Comparisons with other assemblers on real datasets from both human and *Arabidopsis* showed that RAMbler consistently outperforms existing methods by a clear margin. While RAMbler is currently the most effective specialized assembler for complex repeats, the current implementation is haplotype-unaware and performs best when the mutation rate p is at least $1/1000$. A promising direction for future work is haplotype-resolved repeat reconstruction using trio and/or Hi-C data in combination with HiFi reads.

In Chapter 4, published in the Proceedings of the 16th ACM International Conference

on Bioinformatics, Computational Biology, and Health Informatics (BCB'25) [23], we addressed the problem of *de novo* genome assembly in the presence of ultra-deep HiFi sequencing data for the first time. We introduced AWinK, an adaptive window-based read selection algorithm that leverages *unikmers*. Through extensive experiments on both synthetic and real ultra-deep datasets, we compared AWinK with six existing read selection strategies. Our results showed that AWinK produces assemblies with high contiguity and completeness for coverage depths between $50\times$ and $100\times$, while consistently achieving the highest genome fraction and sequence identity among the methods evaluated. Although our current experiments are limited to the model organism *C. elegans*, future work will extend this analysis to multiple organisms with varying genome sizes, complexities, and repeat contents.

In Chapter 5, we presented collaborative studies spanning multiple research directions. In Section 5.1, published in *IEEE Transactions on Computational Biology and Bioinformatics (TCBB)* [40], we developed a sequence-only HyenaDNA model for DNA methylation prediction. Our state-space model outperformed state-of-the-art methods across all methylation contexts and demonstrated improved efficiency over transformer-based models for long input sequences. While it outperformed competing approaches in cross-context prediction, its performance in cross-species prediction was comparable to models incorporating both sequence and annotation information. Finally, in Section 5.2, we introduced GFViewer, a visualization tool for analyzing the chromosomal localization of multigene families using gene family and genome sequence data. GFViewer has been successfully applied to the study of gene families across multiple *Babesia* species, contributing to research on evolution, speciation, virulence factors, and drug target discovery [119, 120].

Bibliography

- [1] Filtlong. <https://github.com/rrwick/Filtlong>.
- [2] Yet another k-mer analyzer (yak). <https://github.com/lh3/yak>.
- [3] Wardah S. Alharbi and Mamoon Rashid. A review of deep learning applications in human genomics using next-generation sequencing data. *Human Genomics*, 16(1):26, July 2022.
- [4] Nicolas Altemose, Glennis A. Logsdon, Andrey V. Bzikadze, Pragya Sidhwani, Sasha A. Langley, Gina V. Caldas, Savannah J. Hoyt, Lev Uralsky, Fedor D. Ryabov, Colin J. Shew, Michael E. G. Sauria, Matthew Borchers, Ariel Gershman, Alla Mikheenko, Valery A. Shepelev, Tatiana Dvorkina, Olga Kunyavskaya, Mitchell R. Vollger, Arang Rhie, Ann M. McCartney, Mobin Asri, Ryan Lorig-Roach, Kishwar Shafin, Julian K. Lucas, Sergey Aganezov, Daniel Olson, Leonardo Gomes de Lima, Tamara Potapova, Gabrielle A. Hartley, Marina Haukness, Peter Kerpedjiev, Fedor Gusev, Kristof Tigyi, Shelise Brooks, Alice Young, Sergey Nurk, Sergey Koren, Sofie R. Salama, Benedict Paten, Evgeny I. Rogaev, Aaron Streets, Gary H. Karpen, Abby F. Dernburg, Beth A. Sullivan, Aaron F. Straight, Travis J. Wheeler, Jennifer L. Gerton, Evan E. Eichler, Adam M. Phillippy, Winston Timp, Megan Y. Dennis, Rachel J. O'Neill, Justin M. Zook, Michael C. Schatz, Pavel A. Pevzner, Mark Diekhans, Charles H. Langley, Ivan A. Alexandrov, and Karen H. Miga. Complete genomic and epigenetic maps of human centromeres. *Science*, 376(6588):eabl4178, 2022.
- [5] Adam Ameur, James B Stewart, Christoph Freyer, Erik Hagström, Max Ingman, Nils-Göran Larsson, and Ulf Gyllenstein. Ultra-deep sequencing of mouse mitochondrial DNA: mutational patterns and their origins. *PLoS Genetics*, 7(3):1–9, 03 2011.
- [6] Christof Angermueller, Heather J. Lee, Wolf Reik, and Oliver Stegle. DeepCpG: accurate prediction of single-cell DNA methylation states using deep learning. *Genome Biology*, 18(1):67, April 2017.
- [7] Philip M Ashton, Satheesh Nair, Tim Dallman, Salvatore Rubino, Wolfgang Rabsch, Solomon Mwaigwisya, John Wain, and Justin O'Grady. MinION nanopore sequencing identifies the position and structure of a bacterial antibiotic resistance island. *Nature Biotechnology*, 33(3):296–300, 03 2015.

- [8] Anton Bankevich, Andrey V. Bzikadze, Mikhail Kolmogorov, Dmitry Antipov, and Pavel A. Pevzner. Multiplex de bruijn graphs enable genome assembly from long, high-fidelity reads. *Nature Biotechnology*, 40(7):1075–1081, 07 2022.
- [9] Anton Bankevich, Sergey Nurk, Dmitry Antipov, Alexey A Gurevich, Mikhail Dvorkin, Alexander S Kulikov, Valery M Lesin, Sergey I Nikolenko, Son Pham, Andrey D Prjibelski, Alexey V Pyshkin, Alexander V Sirotkin, Nikolay Vyahhi, Glenn Tesler, Max A Alekseyev, and Pavel A Pevzner. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *Journal of Computational Biology*, 19(5):455–477, 04 2012.
- [10] Marek Bartkuhn and Rainer Renkawitz. Long range chromatin interactions involved in gene regulation. *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research*, 1783(11):2161–2166, 2008. Functional Architecture of the Cell Nucleus.
- [11] Niko Beerenwinkel and Osvaldo Zagordi. Ultra-deep sequencing for the analysis of viral populations. *Current Opinion in Virology*, 1(5):413–418, 08 2011.
- [12] Samuel N Bogan and Soojin V Yi. Potential role of dna methylation as a driver of plastic responses to the environment across cells, organisms, and populations. *Genome Biology and Evolution*, 16(2):evae022, 02 2024.
- [13] Sébastien Boisvert, François Laviolette, and Jacques Corbeil. Ray: simultaneous assembly of reads from a mix of high-throughput sequencing technologies. *Journal of Computational Biology*, 17(11):1519–1533, 10 2010.
- [14] Bernabe I Bustos, Kimberley Billingsley, Cornelis Blauwendraat, J Raphael Gibbs, Ziv Gan-Or, Dimitri Krainc, Andrew B Singleton, Steven J Lubbe, and International Parkinson’s Disease Genomics Consortium (IPDGC). Genome-wide contribution of common short-tandem repeats to Parkinson’s disease genetic risk. *Brain*, 146(1):65–74, 11 2022.
- [15] Andrey V Bzikadze, Alla Mikheenko, and Pavel A Pevzner. Fast and accurate mapping of long reads to complete genome assemblies with VerityMap. *Genome Research*, 32(11-12):2107–2118, 11 2022.
- [16] Andrey V. Bzikadze and Pavel A. Pevzner. Automated assembly of centromeres from ultra-long error-prone reads. *Nature Biotechnology*, 38(11):1309–1316, 11 2020.
- [17] Andrey V Bzikadze and Pavel A Pevzner. UniAligner: a parameter-free framework for fast sequence alignment. *Nature Methods*, 20(9):1346–1354, 09 2023.
- [18] Peter J Campbell, Erin D Pleasance, Philip J Stephens, Ed Dicks, Richard Rance, Ian Goodhead, George A Follows, Anthony R Green, P Andy Futreal, and Michael R Stratton. Sub-clonal phylogenetic structures in cancer revealed by ultra-deep sequencing. *Proceedings of the National Academy of Sciences*, 105(35):13081–13086, 08 2008.
- [19] Mark J Chaisson, Sudipto Mukherjee, Sreeram Kannan, and Evan E Eichler. Resolving multicopy duplications de novo using polyploid phasing. *Proceedings of RECOMB*, 10229:117–133, 05 2017.

- [20] Supriyo Chakraborty, Richard Tomsett, Ramya Raghavendra, Daniel Harborne, Moustafa Alzantot, Federico Cerutti, Mani Srivastava, Alun Preece, Simon Julier, Raghuveer M. Rao, Troy D. Kelley, Dave Braines, Murat Sensoy, Christopher J. Willis, and Prudhvi Gurram. Interpretability of deep learning models: A survey of results. In *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pages 1–6, 2017.
- [21] Sakshar Chakravarty, Glennis Logsdon, and Stefano Lonardi. RAMbler: de novo genome assembly of complex repetitive regions. In *Proceedings of the 14th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, BCB '23*, New York, NY, USA, 2023. Association for Computing Machinery.
- [22] Sakshar Chakravarty, Glennis Logsdon, and Stefano Lonardi. RAMbler resolves complex repeats in human Chromosomes 8, 19, and X. *Genome Research*, 03 2025.
- [23] Sakshar Chakravarty and Stefano Lonardi. Towards optimal selection of ultra-deep sequencing reads for de novo genome assembly. In *Proceedings of the 16th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, BCB '25*, New York, NY, USA, 2025. Association for Computing Machinery.
- [24] S Chan, W Wang, B ten Hallers, SA Peters, P Gaiero, H de Jong, GS Perez, A Hastie, and H Cao. *Detection, Characterization, and Biological Analysis of Long Tandem Repeats Using Nanochannel Technology*. 2015.
- [25] Ying Chen, Fan Nie, Shang-Qian Xie, Ying-Feng Zheng, Qi Dai, Thomas Bray, Yao-Xin Wang, Jian-Feng Xing, Zhi-Jian Huang, De-Peng Wang, Li-Juan He, Feng Luo, Jian-Xin Wang, Yi-Zhi Liu, and Chuan-Le Xiao. Efficient assembly of nanopore reads via highly accurate and intact error correction. *Nature Communications*, 12(1):60, 01 2021.
- [26] Haoyu Cheng, Gregory T. Concepcion, Xiaowen Feng, Haowen Zhang, and Heng Li. Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm. *Nature Methods*, 18(2):170–175, 02 2021.
- [27] Haoyu Cheng, Erich D Jarvis, Olivier Fedrigo, Klaus-Peter Koepfli, Lara Urban, Neil J Gemmell, and Heng Li. Haplotype-resolved assembly of diploid genomes without parental data. *Nature Biotechnology*, 40(9):1332–1335, 09 2022.
- [28] Chen-Shan Chin and Asif Khalak. Human genome assembly in 100 minutes. *bioRxiv*, 2019.
- [29] Chen-Shan Chin, Paul Peluso, Fritz J Sedlazeck, Maria Nattestad, Gregory T Concepcion, Alicia Clum, Christopher Dunn, Ronan O’Malley, Rosa Figueroa-Balderas, Abraham Morales-Cruz, Grant R Cramer, Massimo Delledonne, Chongyuan Luo, Joseph R Ecker, Dario Cantu, David R Rank, and Michael C Schatz. Phased diploid genome assembly with single-molecule real-time sequencing. *Nature Methods*, 13(12):1050–1054, 12 2016.
- [30] James Clarke, Hai-Chen Wu, Lakmal Jayasinghe, Alpesh Patel, Stuart Reid, and Hagan Bayley. Continuous base identification for single-molecule nanopore DNA sequencing. *Nature Nanotechnology*, 4(4):265–270, 04 2009.

- [31] Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. scGPT: toward building a foundation model for single-cell multi-omics using generative AI. *Nat. Methods*, 21(8):1470–1480, August 2024.
- [32] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [33] DNA sequencing. https://en.wikipedia.org/wiki/dna_sequencing. accessed: 2025-03-08.
- [34] Huilong Du and Chengzhi Liang. Assembly of chromosome-scale contigs by efficiently resolving repetitive sequences with long reads. *Nature Communications*, 10(1):5360, 11 2019.
- [35] Jiamu Du, Lianna M Johnson, Steven E Jacobsen, and Dinshaw J Patel. DNA methylation pathways and their crosstalk with histone methylation. *Nat. Rev. Mol. Cell Biol.*, 16(9):519–532, September 2015.
- [36] Olga Dudchenko, Sanjit S Batra, Arina D Omer, Sarah K Nyquist, Marie Hoeger, Neva C Durand, Muhammad S Shamim, Ido Machol, Eric S Lander, Aviva Presser Aiden, and Erez Lieberman Aiden. De novo assembly of the aedes aegypti genome using Hi-C yields chromosome-length scaffolds. *Science*, 356(6333):92–95, 03 2017.
- [37] John Eid, Adrian Fehr, Jeremy Gray, Khai Luong, John Lyle, Geoff Otto, Paul Peluso, David Rank, Primo Baybayan, Brad Bettman, Arkadiusz Bibillo, Keith Bjornson, Bidhan Chaudhuri, Frederick Christians, Ronald Cicero, Sonya Clark, Ravindra Dalal, Alex Dewinter, John Dixon, Mathieu Foquet, Alfred Gaertner, Paul Hardenbol, Cheryl Heiner, Kevin Hester, David Holden, Gregory Kearns, Xiangxu Kong, Ronald Kuse, Yves Lacroix, Steven Lin, Paul Lundquist, Congcong Ma, Patrick Marks, Mark Maxham, Devon Murphy, Insil Park, Thang Pham, Michael Phillips, Joy Roy, Robert Sebra, Gene Shen, Jon Sorenson, Austin Tomaney, Kevin Travers, Mark Trulson, John Vieceli, Jeffrey Wegener, Dawn Wu, Alicia Yang, Denis Zaccarin, Peter Zhao, Frank Zhong, Jonas Korlach, and Stephen Turner. Real-time DNA sequencing from single polymerase molecules. *Science*, 323(5910):133–138, 11 2008.
- [38] Robert Ekblom, Linnéa Smeds, and Hans Ellegren. Patterns of sequencing coverage bias revealed by ultra-deep sequencing of vertebrate mitochondria. *BMC Genomics*, 15(1):467, 06 2014.
- [39] Jian Fang, Jianjun Jiang, Sarah M. Leichter, Jie Liu, Mahamaya Biswal, Nelli Khudaverdyan, Xuehua Zhong, and Jikui Song. Mechanistic basis for maintenance of CHG DNA methylation in plants. *Nature Communications*, 13(1):3877, July 2022.
- [40] Hao Feng, Sakshar Chakravarty, and Stefano Lonardi. Prediction of dna methylation with long-range state-space models. *IEEE Transactions on Computational Biology and Bioinformatics*, 22(6):3584–3590, 2025.

- [41] Haonan Feng, Lang Wu, Bingxin Zhao, Chad Huff, Jianjun Zhang, Jia Wu, Lifeng Lin, Peng Wei, and Chong Wu. Benchmarking DNA foundation models for genomic sequence classification. August 2024.
- [42] Will Fischer, Vitaly V Ganusov, Elena E Giorgi, Peter T Hraber, Brandon F Keele, Thomas Leitner, Cliff S Han, Cheryl D Gleasner, Lance Green, Chien-Chi Lo, Ambarish Nag, Timothy C Wallstrom, Shuyi Wang, Andrew J McMichael, Barton F Haynes, Beatrice H Hahn, Alan S Perelson, Persephone Borrow, George M Shaw, Tanmoy Bhattacharya, and Bette T Korber. Transmission of single HIV-1 genomes and dynamics of early immune escape revealed by ultra-deep sequencing. *PLoS One*, 5(8):e12303, 08 2010.
- [43] Daniel Y. Fu, Tri Dao, Khaled K. Saab, Armin W. Thomas, Atri Rudra, and Christopher Ré. Hungry Hungry Hippos: Towards language modeling with state space models. In *International Conference on Learning Representations*, 2023.
- [44] Shilpa Garg, Arkarachai Fungtammasan, Andrew Carroll, Mike Chou, Anthony Schmitt, Xiang Zhou, Stephen Mac, Paul Peluso, Emily Hatas, Jay Ghurye, Jared Maguire, Medhat Mahmoud, Haoyu Cheng, David Heller, Justin M Zook, Tobias Moemke, Tobias Marschall, Fritz J Sedlazeck, John Aach, Chen-Shan Chin, George M Church, and Heng Li. Chromosome-scale, haplotype-resolved assembly of human genomes. *Nature Biotechnology*, 39(3):309–312, 03 2021.
- [45] Manish Goel, Hequan Sun, Wen-Biao Jiao, and Korbinian Schneeberger. SyRI: finding genomic rearrangements and local sequence differences from whole-genome assemblies. *Genome Biology*, 20(1):277, 12 2019.
- [46] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *The International Conference on Learning Representations (ICLR)*, 2022.
- [47] Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state-space layers. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS ’21, Red Hook, NY, USA, 2021. Curran Associates Inc.
- [48] Alexey Gurevich, Vladislav Saveliev, Nikolay Vyahhi, and Glenn Tesler. QUASt: quality assessment tool for genome assemblies. *Bioinformatics*, 29(8):1072–1075, 02 2013.
- [49] Anthony J Hannan. Tandem repeats mediating genetic plasticity in health and disease. *Nature Reviews Genetics*, 19(5):286–298, 02 2018.
- [50] David Hernandez, Patrice François, Laurent Farinelli, Magne Osterås, and Jacques Schrenzel. De novo bacterial genome sequencing: millions of very short reads assembled on a desktop computer. *Genome Research*, 18(5):802–809, 03 2008.
- [51] Ting Hon, Kristin Mars, Greg Young, Yu-Chih Tsai, Joseph W Karalius, Jane M Landolin, Nicholas Maurer, David Kudrna, Michael A Hardigan, Cynthia C Steiner, Steven J Knapp, Doreen Ware, Beth Shapiro, Paul Peluso, and David R Rank. Highly accurate long-read HiFi sequencing data for five complex genomes. *Scientific Data*, 7(1):399, 11 2020.

- [52] Xueren Hou, Depeng Wang, Zhukuan Cheng, Ying Wang, and Yuling Jiao. A near-complete assembly of an arabidopsis thaliana genome. *Molecular Plant*, 15(8):1247–1250, 06 2022.
- [53] Savannah J. Hoyt, Jessica M. Storer, Gabrielle A. Hartley, Patrick G. S. Grady, Ariel Gershman, Leonardo G. de Lima, Charles Limouse, Reza Halabian, Luke Wojenski, Matias Rodriguez, Nicolas Altemose, Arang Rhie, Leighton J. Core, Jennifer L. Gerton, Wojciech Makalowski, Daniel Olson, Jeb Rosen, Arian F. A. Smit, Aaron F. Straight, Mitchell R. Vollger, Travis J. Wheeler, Michael C. Schatz, Evan E. Eichler, Adam M. Phillippy, Winston Timp, Karen H. Miga, and Rachel J. O’Neill. From telomere to telomere: The transcriptional and epigenetic state of human repeat elements. *Science*, 376(6588):eabk3112, 2022.
- [54] Jiang Hu, Junpeng Fan, Zongyi Sun, and Shanlin Liu. Nextpolish: a fast and efficient genome polishing tool for long-read assembly. *Bioinformatics*, 36(7):2253–2255, 11 2019.
- [55] J Huddleston, S Ranade, M Malig, F Antonacci, M Chaisson, L Hon, P H Sudmant, T A Graves, C Alkan, M Y Dennis, R K Wilson, S W Turner, J Korlach, and E E Eichler. Reconstructing complex regions of genomes using long-read sequencing technology. *Genome Research*, 24(4):688–696, 04 2014.
- [56] International Human Genome Sequencing Consortium. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, 02 2001.
- [57] Hiroyuki Ishiura, Shota Shibata, Jun Yoshimura, Yuta Suzuki, Wei Qu, Koichiro Doi, M Asem Almansour, Junko Kanda Kikuchi, Makiko Taira, Jun Mitsui, Yuji Takahashi, Yaeko Ichikawa, Tatsuo Mano, Atsushi Iwata, Yasuo Harigaya, Miho Kawabe Matsukawa, Takashi Matsukawa, Masaki Tanaka, Yuichiro Shirota, Ryo Ohtomo, Hisatomo Kowa, Hidetoshi Date, Aki Mitsue, Hiroyuki Hatsuta, Satoru Morimoto, Shigeo Murayama, Yasushi Shiio, Yuko Saito, Akihiko Mitsutake, Mizuho Kawai, Takuya Sasaki, Yusuke Sugiyama, Masashi Hamada, Gaku Ohtomo, Yasuo Terao, Yoshihiko Nakazato, Akitoshi Takeda, Yoshio Sakiyama, Yumi Umeda-Kameyama, Jun Shinmi, Katsuhisa Ogata, Yutaka Kohno, Shen-Yang Lim, Ai Huey Tan, Jun Shimizu, Jun Goto, Ichizo Nishino, Tatsushi Toda, Shinichi Morishita, and Shoji Tsuji. Noncoding CGG repeat expansions in neuronal intranuclear inclusion disease, oculopharyngodistal myopathy and an overlapping disease. *Nature Genetics*, 51(8):1222–1232, 08 2019.
- [58] Shaun D Jackman, Benjamin P Vandervalk, Hamid Mohamadi, Justin Chu, Sarah Yeo, S Austin Hammond, Golnaz Jahesh, Hamza Khan, Lauren Coombe, Rene L Warren, and Inanc Birol. ABySS 2.0: resource-efficient assembly of large genomes using a bloom filter. *Genome Research*, 27(5):768–777, 02 2017.
- [59] Erich D Jarvis, Giulio Formenti, Arang Rhie, Andrea Guarracino, Chentao Yang, Jonathan Wood, Alan Tracey, Francoise Thibaud-Nissen, Mitchell R Vollger, David Porubsky, Haoyu Cheng, Mobin Asri, Glennis A Logsdon, Paolo Carnevali, Mark J P Chaisson, Chen-Shan Chin, Sarah Cody, Joanna Collins, Peter Ebert, Merly Escalona, Olivier Fedrigo, Robert S Fulton, Lucinda L Fulton, Shilpa Garg, Jennifer L Gerton, Jay Ghurye, Anastasiya Granat, Richard E Green, William Harvey, Patrick Hasenfeld, Alex Hastie, Marina Haukness, Erich B Jaeger, Miten Jain, Melanie Kirsche, Mikhail Kolmogorov, Jan O Korbel, Sergey Koren,

- Jonas Korlach, Joyce Lee, Daofeng Li, Tina Lindsay, Julian Lucas, Feng Luo, Tobias Marschall, Matthew W Mitchell, Jennifer McDaniel, Fan Nie, Hugh E Olsen, Nathan D Olson, Trevor Pesout, Tamara Potapova, Daniela Puiu, Allison Regier, Jue Ruan, Steven L Salzberg, Ashley D Sanders, Michael C Schatz, Anthony Schmitt, Valerie A Schneider, Siddarth Selvaraj, Kishwar Shafin, Alaina Shumate, Nathan O Stitzel, Catherine Stober, James Torrance, Justin Wagner, Jianxin Wang, Aaron Wenger, Chuanle Xiao, Aleksey V Zimin, Guojie Zhang, Ting Wang, Heng Li, Erik Garrison, David Haussler, Ira Hall, Justin M Zook, Evan E Eichler, Adam M Phillippy, Benedict Paten, Kerstin Howe, Karen H Miga, and Human Pangenome Reference Consortium. Semi-automated assembly of high-quality diploid human reference genomes. *Nature*, 611(7936):519–531, 11 2022.
- [60] Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. DNABERT: pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. *Bioinformatics*, 37(15):2112–2120, August 2021.
- [61] Bilian Jin, Yajun Li, and Keith D Robertson. DNA methylation: superior or subordinate in the epigenetic hierarchy? *Genes Cancer*, 2(6):607–617, June 2011.
- [62] Junru Jin, Yingying Yu, Ruheng Wang, Xin Zeng, Chao Pang, Yi Jiang, Zhongshen Li, Yutong Dai, Ran Su, Quan Zou, Kenta Nakai, and Leyi Wei. iDNA-ABF: multi-scale deep biological language learning model for the interpretable prediction of DNA methylations. *Genome Biology*, 23(1):219, October 2022.
- [63] M Johnston, L Hillier, L Riles, K Albermann, B André, W Ansorge, V Benes, M Brückner, H Delius, E Dubois, A Düsterhöft, K D Entian, M Floeth, A Goffeau, U Hebling, K Heumann, D Heuss-Neitzel, H Hilbert, F Hilger, K Kleine, P Kötter, E J Louis, F Messenguy, H W Mewes, and J D Hoheisel. The nucleotide sequence of *Saccharomyces cerevisiae* chromosome XII. *Nature*, 387(6632 Suppl):87–90, 05 1997.
- [64] Jerzy Jurka, Vladimir V Kapitonov, Oleksiy Kohany, and Michael V Jurka. Repetitive sequences in complex genomes: structure and evolution. *Annual Review of Genomics and Human Genetics*, 8:241–259, 2007.
- [65] Feyza Duman Keles, Pruthuvi Mahesakya Wijewardena, and Chinmay Hegde. On the computational complexity of self-attention. In *International conference on algorithmic learning theory*, pages 597–619. PMLR, 2023.
- [66] Kristi E Kim, Paul Peluso, Primo Babayan, P Jane Yeadon, Charles Yu, William W Fisher, Chen-Shan Chin, Nicole A Rappavoli, David R Rank, Joachim Li, David E A Catcheside, Susan E Celniker, Adam M Phillippy, Casey M Bergman, and Jane M Landolin. Long-read, whole-genome shotgun sequence data for five model organisms. *Scientific Data*, 1:140045, 11 2014.
- [67] Soyeon Kim, Hyun Jung Park, Xiangqin Cui, and Degui Zhi. Collective effects of long-range DNA methylations predict gene expressions and estimate phenotypes in cancer. *Scientific Reports*, 10(1):3920, March 2020.

- [68] Yeon-Hee Kim, Daisuke Ishikawa, Ho Phu Ha, Minetaka Sugiyama, Yoshinobu Kaneko, and Satoshi Harashima. Chromosome XII context is important for rDNA function in yeast. *Nucleic Acids Research*, 34(10):2914–2924, 05 2006.
- [69] Mikhail Kolmogorov, Jeffrey Yuan, Yu Lin, and Pavel A Pevzner. Assembly of long, error-prone reads using repeat graphs. *Nature Biotechnology*, 37(5):540–546, May 2019.
- [70] Sergey Koren, Brian P Walenz, Konstantin Berlin, Jason R Miller, Nicholas H Bergman, and Adam M Phillippy. Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *Genome Research*, 27(5):722–736, 03 2017.
- [71] Jeffrey D Krimmel, Michael W Schmitt, Maria I Harrell, Kathy J Agnew, Scott R Kennedy, Mary J Emond, Lawrence A Loeb, Elizabeth M Swisher, and Rosa Ana Risques. Ultra-deep sequencing detects ovarian cancer cells in peritoneal fluid and reveals somatic TP53 mutations in noncancerous tissues. *Proceedings of the National Academy of Sciences*, 113(21):6005–6010, 05 2016.
- [72] Martin Krzywinski, Jacqueline Schein, Inanc Birol, Joseph Connors, Randy Gascoyne, Doug Horsman, Steven J Jones, and Marco A Marra. Circos: an information aesthetic for comparative genomics. *Genome Research*, 19(9):1639–1645, 2009.
- [73] Julie A Law and Steven E Jacobsen. Establishing, maintaining and modifying DNA methylation patterns in plants and animals. *Nat. Rev. Genet.*, 11(3):204–220, March 2010.
- [74] Heng Li. Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. *Bioinformatics*, 32(14):2103–2110, 03 2016.
- [75] Heng Li. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100, 05 2018.
- [76] Kunpeng Li, Peng Xu, Jinpeng Wang, Xin Yi, and Yuannian Jiao. Identification of errors in draft genome assemblies at single-nucleotide resolution for quality assessment and improvement. *Nature Communications*, 14(1):6556, 10 2023.
- [77] Wen-Wei Liao, Mobin Asri, Jana Ebler, Daniel Doerr, Marina Haukness, Glenn Hickey, Shuangjia Lu, Julian K Lucas, Jean Monlong, Haley J Abel, Silvia Buonaiuto, Xian H Chang, Haoyu Cheng, Justin Chu, Vincenza Colonna, Jordan M Eizenga, Xiaowen Feng, Christian Fischer, Robert S Fulton, Shilpa Garg, Cristian Groza, Andrea Guarracino, William T Harvey, Simon Heumos, Kerstin Howe, Miten Jain, Tsung-Yu Lu, Charles Markello, Fergal J Martin, Matthew W Mitchell, Katherine M Munson, Moses Njagi Mwaniki, Adam M Novak, Hugh E Olsen, Trevor Pesout, David Porubsky, Pjotr Prins, Jonas A Sibbesen, Jouni Sirén, Chad Tomlinson, Flavia Villani, Mitchell R Vollger, Lucinda L Antonacci-Fulton, Gunjan Baid, Carl A Baker, Anastasiya Belyaeva, Konstantinos Billis, Andrew Carroll, Pi-Chuan Chang, Sarah Cody, Daniel E Cook, Robert M Cook-Deegan, Omar E Cornejo, Mark Diekhans, Peter Ebert, Susan Fairley, Olivier Fedrigo, Adam L Felsenfeld, Giulio Formenti, Adam Frankish, Yan Gao, Nanibaa’ A Garrison, Carlos Garcia Giron, Richard E Green, Leanne Haggerty, Kendra Hoekzema, Thibaut Hourlier, Hanlee P Ji, Eimear E Kenny, Barbara A Koenig, Alexey Kolesnikov, Jan O Korbel, Jennifer Kordosky, Sergey Koren,

- Hojoon Lee, Alexandra P Lewis, Hugo Magalhães, Santiago Marco-Sola, Pierre Marijon, Ann McCartney, Jennifer McDaniel, Jacquelyn Mountcastle, Maria Nattestad, Sergey Nurk, Nathan D Olson, Alice B Popejoy, Daniela Puiu, Mikko Rautiainen, Allison A Regier, Arang Rhie, Samuel Sacco, Ashley D Sanders, Valerie A Schneider, Baergen I Schultz, Kishwar Shafin, Michael W Smith, Heidi J Sofia, Ahmad N Abou Tayoun, Françoise Thibaud-Nissen, Francesca Floriana Tricomi, Justin Wagner, Brian Walenz, Jonathan M D Wood, Aleksey V Zimin, Guillaume Bourque, Mark J P Chaisson, Paul Flicek, Adam M Phillippy, Justin M Zook, Evan E Eichler, David Haussler, Ting Wang, Erich D Jarvis, Karen H Miga, Erik Garrison, Tobias Marschall, Ira M Hall, Heng Li, and Benedict Paten. A draft human pangenome reference. *Nature*, 617(7960):312–324, 05 2023.
- [78] Xingyu Liao, Wufei Zhu, Juexiao Zhou, Haoyang Li, Xiaopeng Xu, Bin Zhang, and Xin Gao. Repetitive dna sequence detection and its role in the human genome. *Communications Biology*, 6(1):954, 09 2023.
- [79] Glennis A Logsdon, Mitchell R Vollger, Pinghsun Hsieh, Yafei Mao, Mikhail A Liskovych, Sergey Koren, Sergey Nurk, Ludovica Mercuri, Philip C Dishuck, Arang Rhie, Leonardo G de Lima, Tatiana Dvorkina, David Porubsky, William T Harvey, Alla Mikheenko, Andrey V Bzikadze, Milinn Kremitzki, Tina A Graves-Lindsay, Chirag Jain, Kendra Hoekzema, Shwetha C Murali, Katherine M Munson, Carl Baker, Melanie Sorensen, Alexandra M Lewis, Urvashi Surti, Jennifer L Gerton, Vladimir Larionov, Mario Ventura, Karen H Miga, Adam M Phillippy, and Evan E Eichler. The structure, function and evolution of a complete human chromosome 8. *Nature*, 593(7857):101–107, 05 2021.
- [80] Stefano Lonardi, Hamid Mirebrahim, Steve Wanamaker, Matthew Alpert, Gianfranco Ciardo, Denisa Duma, and Timothy J. Close. When less is more: ‘slicing’ sequencing data improves read decoding accuracy and de novo assembly quality. *Bioinformatics*, 31(18):2972–2980, 05 2015.
- [81] Matthew Loose, Sunir Malla, and Michael Stout. Real-time selective sequencing using nanopore technology. *Nature Methods*, 13(9):751–754, 09 2016.
- [82] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc.
- [83] Andreas Madsen, Siva Reddy, and Sarath Chandar. Post-hoc interpretability for neural nlp: A survey. *ACM Comput. Surv.*, 55(8), December 2022.
- [84] Serghei Mangul, Nicholas C. Wu, Nicholas Mancuso, Alex Zelikovsky, Ren Sun, and Eleazar Eskin. Accurate viral population assembly from ultra-deep sequencing data. *Bioinformatics*, 30(12):i329–i337, 06 2014.
- [85] Guillaume Marçais and Carl Kingsford. A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics*, 27(6):764–770, 01 2011.
- [86] Scott McGinnis and Thomas L Madden. BLAST: at the core of a powerful and diverse set of sequence analysis tools. *Nucleic Acids Research*, 32(Web Server issue):W20–5, 07 2004.

- [87] Karen H Miga, Sergey Koren, Arang Rhie, Mitchell R Vollger, Ariel Gershman, Andrey Bzikadze, Shelise Brooks, Edmund Howe, David Porubsky, Glennis A Logsdon, Valerie A Schneider, Tamara Potapova, Jonathan Wood, William Chow, Joel Armstrong, Jeanne Fredrickson, Evgenia Pak, Kristof Tigyi, Milinn Kremitzki, Christopher Markovic, Valerie Maduro, Amalia Dutra, Gerard G Bouffard, Alexander M Chang, Nancy F Hansen, Amy B Wilfert, Françoise Thibaud-Nissen, Anthony D Schmitt, Jon-Matthew Belton, Siddarth Selvaraj, Megan Y Dennis, Daniela C Soto, Ruta Sahasrabudhe, Gulhan Kaya, Josh Quick, Nicholas J Loman, Nadine Holmes, Matthew Loose, Urvashi Surti, Rosa Ana Risques, Tina A Graves Lindsay, Robert Fulton, Ira Hall, Benedict Paten, Kerstin Howe, Winston Timp, Alice Young, James C Mullikin, Pavel A Pevzner, Jennifer L Gerton, Beth A Sullivan, Evan E Eichler, and Adam M Phillippy. Telomere-to-telomere assembly of a complete human X chromosome. *Nature*, 585(7823):79–84, 09 2020.
- [88] Alla Mikheenko, Andrey V Bzikadze, Alexey Gurevich, Karen H Miga, and Pavel A Pevzner. TandemTools: mapping long reads and assessing/improving assembly quality in extra-long tandem repeats. *Bioinformatics*, 36(Suppl_1):i75–i83, 07 2020.
- [89] Hamid Mirebrahim, Timothy J Close, and Stefano Lonardi. De novo meta-assembly of ultra-deep sequencing data. *Bioinformatics*, 31(12):i9–i16, 06 2015.
- [90] Camilo Mora, Derek P Tittensor, Sina Adl, Alastair G B Simpson, and Boris Worm. How many species are there on earth and in the ocean? *PLoS Biology*, 9(8):e1001127, 08 2011.
- [91] Jan Mrázek, Xiangxue Guo, and Apurva Shah. Simple sequence repeats in prokaryotic genomes. *Proceedings of the National Academy of Sciences*, 104(20):8472–8477, 2007.
- [92] Matthew Naish, Michael Alonge, Piotr Wlodzimierz, Andrew J Tock, Bradley W Abramson, Anna Schmücker, Terezie Mandáková, Bhagyshree Jamge, Christophe Lambing, Pallas Kuo, Natasha Yelina, Nolan Hartwick, Kelly Colt, Lisa M Smith, Jurriaan Ton, Tetsuji Kakutani, Robert A Martienssen, Korbinian Schneeberger, Martin A Lysak, Frédéric Berger, Alexandros Bousios, Todd P Michael, Michael C Schatz, and Ian R Henderson. The genetic and epigenetic landscape of the arabidopsis centromeres. *Science*, 374(6569):eabi7489, 11 2021.
- [93] Eric Nguyen, Michael Poli, Matthew G. Durrant, Brian Kang, Dhruva Katrekar, David B. Li, Liam J. Bartie, Armin W. Thomas, Samuel H. King, Garyk Brix, Jeremy Sullivan, Madelena Y. Ng, Ashley Lewis, Aaron Lou, Stefano Ermon, Stephen A. Baccus, Tina Hernandez-Boussard, Christopher Ré, Patrick D. Hsu, and Brian L. Hie. Sequence modeling and design from molecular to genome scale with evo. *Science*, 386(6723):eado9336, 2024.
- [94] Eric Nguyen, Michael Poli, Marjan Faizi, Armin W. Thomas, Callum Birch Sykes, Michael Wornow, Aman Patel, Clayton Rabideau, Stefano Massaroli, Yoshua Bengio, Stefano Ermon, Stephen A. Baccus, and Christopher Ré. Hyenadna: long-range genomic sequence modeling at single nucleotide resolution. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23*, Red Hook, NY, USA, 2023. Curran Associates Inc.
- [95] Sergey Nurk, Sergey Koren, Arang Rhie, Mikko Rautiainen, Andrey V. Bzikadze, Alla Mikheenko, Mitchell R. Vollger, Nicolas Altemose, Lev Uralsky, Ariel Gershman, Sergey

- Aganezov, Savannah J. Hoyt, Mark Diekhans, Glennis A. Logsdon, Michael Alonge, Stylianos E. Antonarakis, Matthew Borchers, Gerard G. Bouffard, Shelise Y. Brooks, Gina V. Caldas, Nae-Chyun Chen, Haoyu Cheng, Chen-Shan Chin, William Chow, Leonardo G. de Lima, Philip C. Dishuck, Richard Durbin, Tatiana Dvorkina, Ian T. Fiddes, Giulio Formenti, Robert S. Fulton, Arkarachai Functammasan, Erik Garrison, Patrick G. S. Grady, Tina A. Graves-Lindsay, Ira M. Hall, Nancy F. Hansen, Gabrielle A. Hartley, Marina Haukness, Kerstin Howe, Michael W. Hunkapiller, Chirag Jain, Miten Jain, Erich D. Jarvis, Peter Kerpeldjiev, Melanie Kirsche, Mikhail Kolmogorov, Jonas Korlach, Milinn Kremitzki, Heng Li, Valerie V. Maduro, Tobias Marschall, Ann M. McCartney, Jennifer McDaniel, Danny E. Miller, James C. Mullikin, Eugene W. Myers, Nathan D. Olson, Benedict Paten, Paul Peluso, Pavel A. Pevzner, David Porubsky, Tamara Potapova, Evgeny I. Rogaev, Jeffrey A. Rosenfeld, Steven L. Salzberg, Valerie A. Schneider, Fritz J. Sedlazeck, Kishwar Shafin, Colin J. Shew, Alaina Shumate, Ying Sims, Arian F. A. Smit, Daniela C. Soto, Ivan Sović, Jessica M. Storer, Aaron Streets, Beth A. Sullivan, Françoise Thibaud-Nissen, James Torrance, Justin Wagner, Brian P. Walenz, Aaron Wenger, Jonathan M. D. Wood, Chunlin Xiao, Stephanie M. Yan, Alice C. Young, Samantha Zarate, Urvashi Surti, Rajiv C. McCoy, Megan Y. Dennis, Ivan A. Alexandrov, Jennifer L. Gerton, Rachel J. O'Neill, Winston Timp, Justin M. Zook, Michael C. Schatz, Evan E. Eichler, Karen H. Miga, and Adam M. Phillippy. The complete sequence of a human genome. *Science*, 376(6588):44–53, 2022.
- [96] Sergey Nurk, Brian P Walenz, Arang Rhie, Mitchell R Vollger, Glennis A Logsdon, Robert Grothe, Karen H Miga, Evan E Eichler, Adam M Phillippy, and Sergey Koren. HiCanu: accurate assembly of segmental duplications, satellites, and allelic variants from high-fidelity long reads. *Genome Research*, 30(9):1291–1305, 09 2020.
- [97] Yukiteru Ono, Kiyoshi Asai, and Michiaki Hamada. PBSIM: PacBio reads simulator—toward accurate genome assembly. *Bioinformatics*, 29(1):119–121, 11 2012.
- [98] Nick Orr and Stephen Chanock. *Chapter 1 Common Genetic Variation and Human Disease*, volume 62, pages 1–32. Academic Press, 2008.
- [99] Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y. Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: towards larger convolutional language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.
- [100] Michael Poli, Jue Wang, Stefano Massaroli, Jeffrey Quesnelle, Ryan Carlow, Eric Nguyen, and Armin Thomas. StripedHyena: Moving Beyond Transformers with Hybrid Signal Processing Models, 12 2023.
- [101] David Porubsky, Peter Ebert, Peter A Audano, Mitchell R Vollger, William T Harvey, Pierre Marijon, Jana Ebler, Katherine M Munson, Melanie Sorensen, Arvis Sulovari, Marina Haukness, Maryam Ghareghani, Peter M Lansdorp, Benedict Paten, Scott E Devine, Ashley D Sanders, Charles Lee, Mark J P Chaisson, Jan O Korbel, Evan E Eichler, Tobias Marschall, and Human Genome Structural Variation Consortium. Fully phased human genome assembly without parental data using single-cell strand sequencing and long reads. *Nature Biotechnology*, 39(3):302–308, 03 2021.

- [102] Junjie Qin, Yingrui Li, Zhiming Cai, Shenghui Li, Jianfeng Zhu, Fan Zhang, Suisha Liang, Wenwei Zhang, Yuanlin Guan, Dongqian Shen, Yangqing Peng, Dongya Zhang, Zhuye Jie, Wenxian Wu, Youwen Qin, Wenbin Xue, Junhua Li, Lingchuan Han, Donghui Lu, Peixian Wu, Yali Dai, Xiaojuan Sun, Zesong Li, Aifa Tang, Shilong Zhong, Xiaoping Li, Weineng Chen, Ran Xu, Mingbang Wang, Qiang Feng, Meihua Gong, Jing Yu, Yanyan Zhang, Ming Zhang, Torben Hansen, Gaston Sanchez, Jeroen Raes, Gwen Falony, Shujiro Okuda, Mathieu Almeida, Emmanuelle LeChatelier, Pierre Renault, Nicolas Pons, Jean-Michel Batto, Zhaoxi Zhang, Hua Chen, Ruifu Yang, Weimou Zheng, Songgang Li, Huanming Yang, Jian Wang, S Dusko Ehrlich, Rasmus Nielsen, Oluf Pedersen, Karsten Kristiansen, and Jun Wang. A metagenome-wide association study of gut microbiota in type 2 diabetes. *Nature*, 490(7418):55–60, 10 2012.
- [103] Joshua Quick, Aaron R Quinlan, and Nicholas J Loman. A reference bacterial genome dataset generated on the MinION portable single-molecule nanopore sequencer. *Gigascience*, 3(1):1–6, 2014.
- [104] T. Rhyker Ranallo-Benavidez, Kamil S. Jaron, and Michael C. Schatz. Genomescope 2.0 and smudgeplot for reference-free profiling of polyploid genomes. *Nature Communications*, 11(1):1432, 03 2020.
- [105] Mikko Rautiainen, Sergey Nurk, Brian P Walenz, Glennis A Logsdon, David Porubsky, Arang Rhie, Evan E Eichler, Adam M Phillippy, and Sergey Koren. Telomere-to-telomere assembly of diploid chromosomes with verkko. *Nature Biotechnology*, 41(10):1474–1482, 10 2023.
- [106] Sara Reardon. A complete human genome sequence is close: how scientists filled in the gaps. *Nature*, 594(7862):158–159, June 2021.
- [107] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ”why should i trust you?”: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery.
- [108] Neil J. Risch. Searching for genetic determinants in the new millennium. *Nature*, 405(6788):847–856, 06 2000.
- [109] Richard J Roberts, Mauricio O Carneiro, and Michael C Schatz. The advantages of SMRT sequencing. *Genome Biology*, 14(6):405, 01 2013.
- [110] Jue Ruan and Heng Li. Fast and accurate long-read assembly with wtdbg2. *Nature Methods*, 17(2):155–158, 02 2020.
- [111] Sanger sequencing. https://en.wikipedia.org/wiki/sanger_sequencing. accessed: 2025-03-08.
- [112] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, October 2019.

- [113] Saleh Sereshki, Nathan Lee, Michalis Omirou, Dionysia Fasoula, and Stefano Lonardi. On the prediction of non-cg dna methylation using machine learning. *NAR Genomics and Bioinformatics*, 5(2):lqad045, 05 2023.
- [114] Kishwar Shafin, Trevor Pesout, Ryan Lorig-Roach, Marina Haukness, Hugh E Olsen, Colleen Bosworth, Joel Armstrong, Kristof Tigyi, Nicholas Maurer, Sergey Koren, Fritz J Sedlazeck, Tobias Marschall, Simon Mayes, Vania Costa, Justin M Zook, Kelvin J Liu, Duncan Kilburn, Melanie Sorensen, Katy M Munson, Mitchell R Vollger, Jean Monlong, Erik Garrison, Evan E Eichler, Sofie Salama, David Haussler, Richard E Green, Mark Akeson, Adam Phillippy, Karen H Miga, Paolo Carnevali, Miten Jain, and Benedict Paten. Nanopore sequencing and the shasta toolkit enable efficient de novo assembly of eleven human genomes. *Nature Biotechnology*, 38(9):1044–1053, 09 2020.
- [115] Nakul M Shah, H Josh Jang, Yonghao Liang, Ju Heon Maeng, Shin-Cheng Tzeng, Angela Wu, Noah L Basri, Xuan Qu, Changxu Fan, Amy Li, Benjamin Katz, Daofeng Li, Xiaoyun Xing, Bradley S Evans, and Ting Wang. Pan-cancer analysis identifies tumor-specific antigens derived from transposable elements. *Nature Genetics*, 55(4):631–639, 04 2023.
- [116] Alaina Shumate, Aleksey V Zimin, Rachel M Sherman, Daniela Puiu, Justin M Wagner, Nathan D Olson, Mihaela Pertea, Marc L Salit, Justin M Zook, and Steven L Salzberg. Assembly and annotation of an Ashkenazi human reference genome. *Genome Biology*, 21(1):129, 06 2020.
- [117] Jared T Simpson and Richard Durbin. Efficient de novo assembly of large genomes using compressed data structures. *Genome Research*, 22(3):549–556, 12 2011.
- [118] Jared T Simpson, Kim Wong, Shaun D Jackman, Jacqueline E Schein, Steven J M Jones, and Inanç Birol. ABySS: a parallel assembler for short read sequence data. *Genome Research*, 19(6):1117–1123, 02 2009.
- [119] Pallavi Singh, Stefano Lonardi, Qihua Liang, Pratap Vydyam, Eleonora Khabirova, Tiffany Fang, Shalev Gihaz, Jose Thekkiniath, Muhammad Munshi, Steven Abel, Loic Ciampossin, Gayani Batugedara, Mohit Gupta, Xueqing Maggie Lu, Todd Lenz, Sakshar Chakravarty, Emmanuel Cornillot, Yangyang Hu, Wenxiu Ma, Luis Miguel Gonzalez, Sergio Sánchez, Karel Estrada, Alejandro Sánchez-Flores, Estrella Montero, Omar S Harb, Karine G Le Roch, and Choukri Ben Mamoun. Babesia duncani multi-omics identifies virulence factors and drug targets. *Nature Microbiology*, 8(5):845–859, 05 2023.
- [120] Pallavi Singh, Pratap Vydyam, Tiffany Fang, Karel Estrada, Luis Miguel Gonzalez, Ricardo Grande, Madelyn Kumar, Sakshar Chakravarty, Vincent Berry, Vincent Ranwez, Bernard Carcy, Delphine Depoix, Sergio Sánchez, Emmanuel Cornillot, Steven Abel, Loic Ciampossin, Todd Lenz, Omar Harb, Alejandro Sanchez-Flores, Estrella Montero, Karine G Le Roch, Stefano Lonardi, and Choukri Ben Mamoun. Insights into the evolution, virulence and speciation of babesia MO1 and babesia divergens through multiomics analyses. *Emerging Microbes & Infections*, 13(1):2386136, 09 2024.

- [121] Necla Nisa Soylu and Emre Sefer. Bert2ome: Prediction of 2'-o-methylation modifications from rna sequence by transformer architecture based on bert. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(3):2177–2189, 2023.
- [122] Peter H Sudmant, Jacob O Kitzman, Francesca Antonacci, Can Alkan, Maika Malig, Anya Tsalenko, Nick Sampas, Laurakay Bruhn, Jay Shendure, 1000 Genomes Project, and Evan E Eichler. Diversity of human copy number variation and multicopy genes. *Science*, 330(6004):641–646, 10 2010.
- [123] Ruixia Sun and Ping Zhu. Advances in measuring DNA methylation. *Blood Sci.*, 4(1):8–15, January 2022.
- [124] Qi Tian, Jianxiao Zou, Jianxiong Tang, Yuan Fang, Zhongli Yu, and Shicai Fan. MRCNN: a deep learning model for regression of genome-wide DNA methylation. *BMC Genomics*, 20(Suppl 2):192, April 2019.
- [125] Louis Tirot, Pauline E Jullien, and Mathieu Ingouff. Evolution of CG methylation maintenance machinery in plants. *Epigenomes*, 5(3):19, September 2021.
- [126] Todd J. Treangen, Anne-Laure Abraham, Marie Touchon, and Eduardo P.C. Rocha. Genesis, effects and fates of repeats in prokaryotic genomes. *FEMS Microbiology Reviews*, 33(3):539–571, 05 2009.
- [127] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [128] Mitchell R. Vollger, Philip C. Dishuck, Melanie Sorensen, AnneMarie E. Welch, Vy Dang, Max L. Dougherty, Tina A. Graves-Lindsay, Richard K. Wilson, Mark J. P. Chaisson, and Evan E. Eichler. Long-read sequence and assembly of segmental duplications. *Nature Methods*, 16(1):88–94, 01 2019.
- [129] Gregory W Vulture, Fritz J Sedlazeck, Maria Nattestad, Charles J Underwood, Han Fang, James Gurtowski, and Michael C Schatz. GenomeScope: fast reference-free genome profiling from short reads. *Bioinformatics*, 33(14):2202–2204, 03 2017.
- [130] Bo Wang, Xiaofei Yang, Yanyan Jia, Yu Xu, Peng Jia, Ningxin Dang, Songbo Wang, Tun Xu, Xixi Zhao, Shenghan Gao, Quanbin Dong, and Kai Ye. High-quality arabidopsis thaliana genome assembly with nanopore and HiFi long reads. *Genomics, Proteomics & Bioinformatics*, 20(1):4–13, 09 2021.
- [131] Shuyu Wang, Yinbo Liu, Yufeng Liu, Yong Zhang, and Xiaolei Zhu. BERT-5mC: an interpretable model for predicting 5-methylcytosine sites of DNA based on BERT. *PeerJ*, 11:e16600, December 2023.
- [132] Yifan Wang, Pingxian Zhang, Weijun Guo, Hanqing Liu, Xiulan Li, Qian Zhang, Zhuoying Du, Guihua Hu, Xiao Han, Li Pu, Jian Tian, and Xiaofeng Gu. A deep learning approach

- to automate whole-genome prediction of diverse epigenomic modifications in plants. *New Phytol*, 232(2):880–897, August 2021.
- [133] Aaron M Wenger, Paul Peluso, William J Rowell, Pi-Chuan Chang, Richard J Hall, Gregory T Concepcion, Jana Ebler, Arkarachai Fungtammasan, Alexey Kolesnikov, Nathan D Olson, Armin Töpfer, Michael Alonge, Medhat Mahmoud, Yufeng Qian, Chen-Shan Chin, Adam M Phillippy, Michael C Schatz, Gene Myers, Mark A DePristo, Jue Ruan, Tobias Marschall, Fritz J Sedlazeck, Justin M Zook, Heng Li, Sergey Koren, Andrew Carroll, David R Rank, and Michael W Hunkapiller. Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome. *Nature Biotechnology*, 37(10):1155–1162, 10 2019.
 - [134] Lingjuan Xie, Xiaojiao Gong, Kun Yang, Yujie Huang, Shiyu Zhang, Leti Shen, Yanqing Sun, Dongya Wu, Chuyu Ye, Qian-Hao Zhu, and Longjiang Fan. Technology-enabled great leap in deciphering plant genomes. *Nature Plants*, 10(4):551–566, 04 2024.
 - [135] Yingxue Yang, Wenjie Du, Yanchun Li, Jiawei Lei, and Weihua Pan. Recent advances and challenges in de novo genome assembly. *Genomics Communications*, 2(gcomm-0025-0015), 2025.
 - [136] Wenjuan Yu, Haohui Luo, Jinbao Yang, Shengchen Zhang, Heling Jiang, Xianjia Zhao, Xingqi Hui, Da Sun, Liang Li, Xiu-Qing Wei, Stefano Lonardi, and Weihua Pan. Comprehensive assessment of 11 de novo HiFi assemblers on complex eukaryotic genomes and metagenomes. *Genome Research*, 34(2):326–340, 03 2024.
 - [137] Tianwei Yue, Yuanxin Wang, Longxiang Zhang, Chunming Gu, Haoru Xue, Wenping Wang, Qi Lyu, and Yujie Dun. Deep learning for genomics: From early neural nets to modern large language models. *International Journal of Molecular Sciences*, 24(21), 2023.
 - [138] Haoyang Zeng and David K Gifford. Predicting the impact of non-coding variants on DNA methylation. *Nucleic Acids Res.*, 45(11):e99, June 2017.
 - [139] Daniel R. Zerbino and Ewan Birney. Velvet: Algorithms for de novo short read assembly using de bruijn graphs. *Genome Research*, 18(5):821–829, 2008.
 - [140] Huiming Zhang, Zhaobo Lang, and Jian-Kang Zhu. Dynamics and function of DNA methylation in plants. *Nature Reviews Molecular Cell Biology*, 19(8):489–506, August 2018.
 - [141] Shuang Zhang, Rui Fan, Yuti Liu, Shuang Chen, Qiao Liu, and Wanwen Zeng. Applications of transformer-based language models in bioinformatics: a survey. *Bioinform. Adv.*, 3(1):vbad001, January 2023.
 - [142] Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana Davuluri, and Han Liu. Dnabert-2: Efficient foundation model and benchmark for multi-species genome, 2023.
 - [143] Aleksey V. Zimin, Guillaume Marçais, Daniela Puiu, Michael Roberts, Steven L. Salzberg, and James A. Yorke. The MaSuRCA genome assembler. *Bioinformatics*, 29(21):2669–2677, 08 2013.

Appendix A

RAmbler

A.1 Command-line listing

All the commands used to execute tools, assemblers, and RAmbler are listed below.

Commands for k -mer analysis

```
jellyfish count -m $k -C -o $count_output -c 3 -s 10000000 \
--disk -t 16 $reads
jellyfish histo -o $histo_output -v $count_output
jellyfish dump -c -t -L $lower -U $upper $count_output | awk \
  '{print $1}' > $dump_output
```

Commands for assemblers

```
# hifiasm
hifiasm -o $output/prefix -t 20 $reads

# LJA
lja -t 20 -o $output --reads $reads

# LJA (for simulated diploid genomes)
lja --diploid -t 20 -o $output --reads $reads

# HiCANU
```

```
canu -p prefix -d $output genomeSize=$ASM_SIZE \
-pacbio-hifi $reads
```

```
# Verkko
verkko -d $output --hifi $reads --no-nano
```

Commands for RAmbler

```
bash run.sh -r read_file -u unikmer_file -o output_directory -s \
assembly_length [-k kmer_length] [-l tolerance] [-h threshold]
```

required:

-r path to the fasta file containing the HiFi reads extracted \
in step A

-u path to the file containing the list of unikmers extracted \
in step B

-o path to the directory where RAmbler will put all the \
generated outputs

-s estimated length of the repeat resolved part of the genome in bp
optional:

-k length of kmers used (default: 21)

-l tolerance parameter (default: 15)

-h threshold parameter (default: 15)

```
bash stitch.sh -g draft_genome_file -n contig_id -b start_position \
-e end_position -o output_directory -s contig_length
```

required:

-g path to the fasta file containing the draft/reference genome

-n id of the contig that is being re-assembled

-b approximate start position of the repeat region in the contig \
sequence from the draft genome

-e approximate end position of the repeat region in the contig \
sequence from the draft genome

-o path to the directory where RAmbler will put all the generated \
outputs (always put the same one as run.sh)

-s estimated length of the repeat resolved contig in bp

Commands for evaluation tools

```
# CRAQ
```

```
perl craq -g reference.fasta -ngs ngs.bam -sms hifi.bam \
-pl T -D output_directory
```

```
# QUAST
```

```
quast <assembly1> <assembly2> ... <assemblyN> -r reference.fasta
```

```
# BLAST
makeblastdb -in assembly.fasta -dbtype nucl -out prefix_db
blastn -query repeat_unit.fasta -db prefix_db \
  -outfmt "6 qseqid sseqid pident length evalue bitscore" \
  -out out_file.tsv
```

A.2 Supplemental Code

Detailed instructions for executing RAMbler are provided in the GitHub repository (<https://github.com/ucrbioinfo/rambler>).

Appendix B

AWin*K*

B.1 Command-line listing

All the commands used to execute tools, assemblers, and AWin*K* are listed below.

Commands for AWin*K*

```
cd ~
git clone https://github.com/sakshar/AWinK.git
cd AWinK/scripts
bash extractUnikmers.sh prefix k
python read2UnikmerMap.py
python awink.py
```

Commands to generate the final assembly using hifiasm

```
hifiasm -o $OUT_DIR/$PREFIX -t $THREADS $READ_FILE
awk '/^S/{print ">"$2;print $3}' $OUT_DIR/$PREFIX.bp.p_ctg.gfa \
> $OUT_DIR/$PREFIX.asm.fasta
```

B.2 Supplemental Code

Detailed instructions for executing *AWinK* are provided in the GitHub repository (<https://github.com/sakshar/AWinK>).

Appendix C

GFViewer

C.1 Test data

Some example test data sets provided with GFViewer are listed below.

An example gene-family data file for *Babesia divergens* *MOI* with six most frequent gene-families

(see Figure [5.13](#))

```
gene_id, gene_family, chromosome, start, end, strand
BspMO1_0002000, MGF1, chr1, 73596, 74168, -
BspMO1_0003900, MGF1, chr1, 140431, 141150, -
BspMO1_0004300, MGF1, chr1, 156414, 157130, -
BspMO1_0005000, MGF1, chr1, 200517, 201089, -
BspMO1_0005600, MGF1, chr1, 230921, 231637, -
BspMO1_0007000, MGF1, chr1, 307437, 308009, -
BspMO1_0007800, MGF1, chr1, 361666, 362238, -
BspMO1_0062400, MGF1, chr1, 1483848, 1484420, +
BspMO1_0169100, MGF1, chr1, 3623067, 3623639, +
BspMO1_0169400, MGF1, chr1, 3633008, 3633580, +
BspMO1_0170600, MGF1, chr1, 3692601, 3693173, +
BspMO1_0173300, MGF1, chr1, 3786067, 3786639, +
BspMO1_0174000, MGF1, chr1, 3859772, 3860344, +
BspMO1_0333600, MGF1, chr2, 3222662, 3223234, -
BspMO1_0333800, MGF1, chr2, 3229213, 3229857, -
BspMO1_0333900, MGF1, chr2, 3229477, 3229857, -
```

BspMO1_0334600, MGF1, chr2, 3246995, 3247567, -
 BspMO1_0336100, MGF1, chr2, 3314519, 3315148, -
 BspMO1_0338500, MGF1, chr2, 3447133, 3447837, -
 BspMO1_0344200, MGF1, chr3, 22823, 23395, +
 BspMO1_0345500, MGF1, chr3, 69838, 72393, +
 BspMO1_0346800, MGF1, chr3, 100739, 101311, +
 BspMO1_0349700, MGF1, chr3, 226569, 227141, +
 BspMO1_0351300, MGF1, chr3, 309054, 309626, +
 BspMO1_0351900, MGF1, chr3, 332615, 333187, +
 BspMO1_0362200, MGF1, chr3, 557573, 558145, -
 BspMO1_0433000, MGF1, chr3, 2036145, 2036861, +
 BspMO1_0005700, MGF2, chr1, 233313, 234613, +
 BspMO1_0171200, MGF2, chr1, 3709819, 3711119, -
 BspMO1_0175100, MGF2, chr1, 3909871, 3911171, -
 BspMO1_0340900, MGF2, chr2, 3518664, 3519964, +
 BspMO1_0341100, MGF2, chr2, 3528937, 3530237, +
 BspMO1_0342000, MGF2, chr2, 3596832, 3598252, +
 BspMO1_0342200, MGF2, chr2, 3612453, 3613753, +
 BspMO1_0343900, MGF2, chr3, 19853, 21153, -
 BspMO1_0345800, MGF2, chr3, 83447, 84783, -
 BspMO1_0346900, MGF2, chr3, 106717, 137895, -
 BspMO1_0347100, MGF2, chr3, 114629, 115929, -
 BspMO1_0347600, MGF2, chr3, 138035, 138817, -
 BspMO1_0348800, MGF2, chr3, 204603, 205903, -
 BspMO1_0351200, MGF2, chr3, 306045, 339993, -
 BspMO1_0352300, MGF2, chr3, 338573, 339993, -
 BspMO1_0352600, MGF2, chr3, 347046, 348466, -
 BspMO1_0432900, MGF2, chr3, 2024264, 2060341, -
 BspMO1_0002500, MGF3, chr1, 89931, 90713, +
 BspMO1_0002800, MGF3, chr1, 106224, 107006, +
 BspMO1_0003600, MGF3, chr1, 129128, 129766, +
 BspMO1_0004400, MGF3, chr1, 165572, 166348, +
 BspMO1_0007200, MGF3, chr1, 323385, 324167, +
 BspMO1_0169200, MGF3, chr1, 3630538, 3631317, -
 BspMO1_0170100, MGF3, chr1, 3681303, 3682082, -
 BspMO1_0170400, MGF3, chr1, 3690170, 3690949, -
 BspMO1_0175300, MGF3, chr1, 3919356, 3920012, -
 BspMO1_0176200, MGF3, chr1, 3998183, 3998965, -
 BspMO1_0177300, MGF3, chr1, 4094362, 4095027, -
 BspMO1_0334200, MGF3, chr2, 3231554, 3232336, +
 BspMO1_0338300, MGF3, chr2, 3438627, 3439409, +
 BspMO1_0348600, MGF3, chr3, 197487, 198269, -
 BspMO1_0348900, MGF3, chr3, 205121, 205903, -
 BspMO1_0349500, MGF3, chr3, 224105, 224887, -

```

BspM01_0350900, MGF3, chr3, 279904, 280686, -
BspM01_0433400, MGF3, chr3, 2051982, 2052764, -
BspM01_0438900, MGF3, chr3, 2303902, 2303939, -
BspM01_0168900, MGF4, chr1, 3620600, 3621382, -
BspM01_0174100, MGF4, chr1, 3866202, 3866867, -
BspM01_0175500, MGF4, chr1, 3926726, 3927508, -
BspM01_0177600, MGF4, chr1, 4101807, 4102514, -
BspM01_0340200, MGF4, chr2, 3496998, 3497780, +
BspM01_0342500, MGF4, chr2, 3636724, 3637389, +
BspM01_0344700, MGF4, chr3, 34343, 35125, -
BspM01_0344800, MGF4, chr3, 36648, 37313, -
BspM01_0345000, MGF4, chr3, 44072, 44737, -
BspM01_0438000, MGF4, chr3, 2262409, 2263191, -
BspM01_0438800, MGF4, chr3, 2301678, 2302460, -
BspM01_0172300, MGF5, chr1, 3748908, 3749615, -
BspM01_0225300, MGF5, chr2, 974330, 975109, -
BspM01_0235900, MGF5, chr2, 1191643, 1192425, +
BspM01_0336600, MGF5, chr2, 3330901, 3331680, +
BspM01_0341400, MGF5, chr2, 3557117, 3558516, +
BspM01_0344300, MGF5, chr3, 29331, 30751, -
BspM01_0344400, MGF5, chr3, 29969, 30751, -
BspM01_0344600, MGF5, chr3, 32156, 32938, -
BspM01_0351800, MGF5, chr3, 330127, 330909, -
BspM01_0123000, MGF6, chr1, 2675381, 2675737, -
BspM01_0171000, MGF6, chr1, 3705866, 3706222, +
BspM01_0338200, MGF6, chr2, 3436412, 3436780, -
BspM01_0339000, MGF6, chr2, 3454335, 3454411, -
BspM01_0347900, MGF6, chr3, 167052, 167420, +
BspM01_0435400, MGF6, chr3, 2141630, 2141995, +
BspM01_0435900, MGF6, chr3, 2153892, 2154254, +

```

The corresponding chromosome-lengths file

```

chr1, 4143664
chr2, 3659128
chr3, 2321341

```

The color map file

- with color codes from the default color palette

```
MGF1, 1  
MGF2, 2  
MGF3, 3  
MGF4, 4  
MGF5, 5  
MGF6, 6
```

- with custom RGB values

```
MGF1, 1, 0, 0  
MGF2, 0, 1, 0  
MGF3, 0, 0, 1  
MGF4, 1, 1, 0  
MGF5, 0, 1, 1  
MGF6, 1, 0, 1
```

C.2 Supplemental Code

Detailed instructions for installing and executing GFViewer are provided in the GitHub repository (<https://github.com/sakshar/GFViewer>).