UNIVERSITY OF CALIFORNIA

Los Angeles

Models, Algorithms, and Downstream Applications of Nanopore Sequencing

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical and Computer Engineering

by

Dhaivat Janmejay Joshi

2024

ABSTRACT OF THE DISSERTATION

Models, Algorithms, and Downstream Applications of Nanopore Sequencing

by

Dhaivat Janmejay Joshi
Doctor of Philosophy in Electrical and Computer Engineering
University of California, Los Angeles, 2024
Professor Suhas N. Diggavi, Chair

The advent of nanopore sequencing technology represents a significant leap forward in the ability to read long fragments of DNA, up to 4M bases, surpassing the capabilities of traditional short-read sequencing methods that can read a few hundred bases. Despite its potential, nanopore sequencing is challenged by high error rates ($5\% - 15\%$). In this dissertation, we presents a comprehensive examination of various computational approaches to address these challenges and enhance the utility of nanopore sequencing technology in genomic analysis by using an underlying physics-based model of nanopore sequencers to guide our methods.

First, we describe a mathematical model that describes the “nanopore channel” which takes a DNA sequence as input and outputs observed current variations in a nanopore sequencer. This model accounts for impairments such as inter-symbol interference, insertions-deletions, channel fading, and random responses. Moreover, the model also provides insights for the error profiles in the nanopore sequencer that can be utilized to develop algorithms for downstream applications. We further study the bounds on the information extraction capacity of nanopore sequencers that provides benchmarks for existing base-calling algorithms and guidelines for designing improved nanopores.

Our first main algorithmic work introduces QAlign, a preprocessing tool that improves the accuracy and efficiency of long-read aligners by converting nucleotide reads into discretized current levels. This transformation captures the error characteristics of nanopore sequencers studied in the previous work, enhancing alignment rates of nanopore reads to reference from around 80% to 90%, improving overlap quality for read-to-read alignments, and read-to-transcriptome alignment rates significantly across multiple datasets.

Our second main algorithmic work focuses on the detection of structural variants (SVs) using nanopore sequenced reads. We present HQAlign, an aligner designed to leverage the physics of nanopore sequencing and SV-specific modifications to enhance alignment accuracy. HQAlign demonstrates a $4\% - 6\%$ improvement in detecting complementary SVs compared to the minimap2 aligner, along with substantial improvements in breakpoint accuracy and overall alignment rates for read to reference alignments as compared to QAlign and minimap2.

The final algorithmic work addresses the challenge of identifying heterozygous variants using the highly erroneous nanopore reads data for developing algorithms for diploid genome assembly. We propose an algorithm that identifies heterozygous variants with a recall of 90% and precision of 70%, facilitating the reconstruction of diploid genomes without additional reference information or preliminary draft assemblies.

Collectively, these studies advance the understanding and application of nanopore sequencing technology, offering novel computational methods to mitigate high error rates and improve genomic analyses, including alignment, structural variant detection, and diploid genome assembly.

The dissertation of Dhaivat Janmejay Joshi is approved.

Mark J.P. Chaisson

Sreeram Kannan

Sriram Sanakararaman

Christina P. Fragouli

Suhas N. Diggavi, Committee Chair

University of California, Los Angeles

2024

To my mumma & papa . . .

for always encouraging and motivating me

to face the many challenges of the journey, and for so much more. . .

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

As I wrap up my long journey through graduate school, spanning about seven years for both my Masters and Ph.D. at UCLA, I want to take a moment to thank everyone who helped and supported me along the way. These wonderful people have shaped my career and also helped me grow personally and emotionally.

First and foremost, I want to express my heartfelt gratitude to my advisor, Prof. Suhas Diggavi. His continuous support and guidance throughout my Masters and later as his Ph.D. student have been invaluable. I initially joined UCLA as a Masters student without any intention of pursuing a Ph.D., but working with Prof. Diggavi in the final year of my MS, learning from his insights on problem-solving, how quickly he grasps the ideas I am explaining during our weekly meetings, his depth of technical knowledge always amazes me and sparked my interest in continuing my Ph.D. at UCLA under his supervision. His support, guidance, patience, and genuine concern for his students' best interests have been crucial in helping me reach the finish line and start my career in research.

I would also like to extend my gratitude to Prof. Mark Chaisson at University of Southern California, Los Angeles. He has been an incredibly supportive collaborator and a fantastic person to work with during the last two years of my graduate program. His insights in identifying and solving problems, and guidance have been very valuable in my Ph.D. His dedication to help his students collaborators with their best interests at heart makes me blessed to have two advisors by the end of my graduate program.

I would also like to thank Prof. Sreeram Kannan at University of Washington, Seattle, for his continuous support and guidance in my research during the first three years of my Ph.D. I would also like to thank Prof. Sriram Sankararaman and Prof. Christina Fragouli at UCLA for their guidance and for being on my doctoral committee. Additionally, I'm grateful to Dr. Swati Kaushik, my internship manager at Gilead Sciences, for the opportunity to learn and explore new problems in Bioinformatics, and for collaborating with the amazing research team at Gilead.

I've been incredibly fortunate to have amazing friends from the day I arrived in the US. Special thanks to Farsak Palia, Niket Patel, Aditya Kamble, Naman Parikh, Himanshu Dogra, and Akshay Bapat for being like family away from home. I also want to thank my fellow labmates in the UCLA, UW, and USC groups: Navjot, Deepesh, Sundar, Antonious, Ruida, SJ, Shunfu, Debraj, Kaan, Rida, Ufuk, Bruce, Yagiz, Osama, Mine, Merve, Xinlin, and Chaorui, for all the technical and non-technical conversations and support.

I'm also grateful to my friends here, back in India, and my cricket team members: Akshay, Tapan, Rutvi, Kalpan, Utsavi, Gaurav, Yash, Siddharth, Dhruv, Mihir, Ravi, Darshan, Ranvir, Mayank, Sameer, Ameya, Chintan, Sagar, Shardul, Shaan, Santosh, GB, Vivek, Ashok, Jason, Varun, and Hardik. Thank you for being an integral part of my journey and for all the amazing times we shared.

Last but not least, I want to thank my Mumma and Papa for their constant support, encouragement, and motivation. I couldn't have completed my degree without their love and support. I also thank my family for their blessings, and my cousins for always being there for me.

VITA

2017 Bachelor of Technology in Electrical Engineering, Indian Institute of Technology, Indore, Madhya Pradesh, India.

2018 Teaching Assistant, Physics Department, University of California, Los Angeles.

2019 Master of Science in Electrical and Computer Engineering, University of California, Los Angeles.

2019–2020 Teaching Associate, Electrical and Computer Engineering Department, University of California, Los Angeles.

2021 Summer Mentored Research Fellowship.

2021 Teaching Fellow, Electrical and Computer Engineering Department, University of California, Los Angeles.

2022 Research Intern – Bioinformatics, Gilead Sciences, Foster city, California.

PUBLICATIONS

Dhaivat Joshi, Suhas Diggavi, Mark Chaisson and Sreeram Kannan. HQAlign: Aligning nanopore reads for SV detection using current-level modeling. *Bioinformatics*, 39, (2023).

Dhaivat Joshi, Shunfu Mao, Sreeram Kannan and Suhas Diggavi. QAlign: Aligning nanopore reads accurately using current-level modeling. *Bioinformatics*, 37, 625–633 (2021).

CHAPTER 1

Introduction

Deoxyribonucleic acid (DNA) is the molecule that encodes the instructions for life. This complex molecule, found in nearly all living organisms, stores genetic information in the form of a sequence of four chemical building blocks: adenine (A), guanine (G), cytosine (C), and thymine (T). Understanding the order of these bases, or DNA sequencing, is fundamental to deciphering the secrets of biology and medicine. Since its inception, DNA sequencing has evolved significantly, revolutionizing fields such as genetics, medicine, evolutionary biology, computer science, and bioinformatics. This chapter introduces the problems we study in this dissertation and also gives a historical perspective of the topic. We also outline and summarize the contributions of this dissertation and its organization.

1.1 Historical Development

The quest to read the genetic code began in the early 20th century. Pioneering work by Frederick Sanger in the 1970s led to the development of the Sanger sequencing method, the first widely used technique. Sanger sequencing relied on enzymatic chain termination reactions to generate fragments of DNA with defined lengths. These fragments were then separated by size using gel electrophoresis, allowing researchers to determine the base sequence. This revolutionary method paved the way for the Human Genome Project, a massive undertaking that successfully sequenced majority of the human genome in 2003. Various sequencing techniques developed later employ different methodologies to achieve this, including Illumina sequencing, nanopore sequencing, and single-molecule real-time (SMRT) sequencing. Despite differences in approach, all sequencing methods involve the generation of DNA fragments, sequencing of these fragments, and subsequent assembly of the sequence data to reconstruct the original DNA molecule.

1.2 Significance of DNA Sequencing

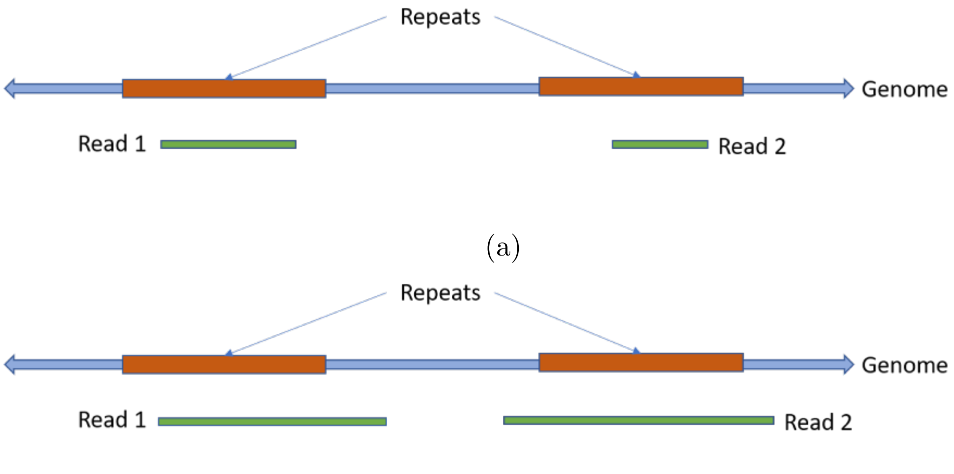
DNA sequencing has transformed biological research in numerous ways:

1. **Genomic Analysis:** By deciphering the complete DNA sequence of an organism (its genome), scientists can gain insights into its genetic makeup, evolutionary history, and potential functions of genes.
2. **Medical Applications:** DNA sequencing plays a crucial role in medical diagnostics, personalized medicine, and genetic counseling. It enables the identification of disease-causing mutations, genetic predispositions, and personalized treatment strategies.
3. **Biotechnology:** DNA sequencing facilitates the engineering of novel proteins, genetic modification of organisms, and development of biopharmaceuticals.
4. **Evolutionary Biology:** Comparative genomics, enabled by DNA sequencing, allows researchers to study evolutionary relationships between species, trace evolutionary events, and understand genomic adaptations.
5. **Forensic Analysis:** DNA sequencing is utilized in forensic science for identifying individuals, determining familial relationships, and solving criminal cases.

1.3 Evolution of DNA Sequencing Technology

Over the past few decades, DNA sequencing technology has undergone remarkable evolution, characterized by improvements in speed, accuracy, scalability, and cost-effectiveness.

The advent of high-throughput DNA sequencing methods has greatly accelerated research and discovery in genomics. Much of this improvement was made possible with the second-generation sequencing technology that not only plummeted the cost of sequencing by nearly six orders of magnitude, but also sequenced the entire genome at once by utilizing massive parallelization. This is accomplished by fragmenting the genome into small pieces, and then sequencing it. Therefore, these devices can only read short fragments of DNA that are typically a few hundred bases long. Utilizing the algorithms that exploits the overlaps between these short reads, they can be stitched together to assemble them into long DNA sequence. However, this assembly is challenging because of several reasons:



(a)

(b)

Figure 1.1: (a) Short reads sequenced a long approximate repeat on genome (highlighted in red) are very similar and therefore it is difficult to resolve the long repeats in the genome using such short reads. (b) Long reads can span the entire approximate repeat region on genome and therefore can resolve such regions well in the downstream task such as assembly or variant detection in such repeats.

1. **Repetitive regions:** Genome contains repetitive sequences that can be difficult to assemble accurately from short reads as shown in Fig. 1.1(a). The short reads are about few hundred basepair long and are insufficient to resolve long repetitive regions that are more than few thousand basepair long such as segmental duplications which are

low-copy repeats of length ranging from 1kb to 400kb with sequence identity more than 90% [4]. This make the problem of finding the exact region of origin on the genome for the short reads very difficult and therefore, assembly with these reads becomes quite challenging and leads to incomplete characterization of the variants within these long repeats. For example, assembly of two long segmental duplication repeats using short reads can interchange the variants within these repeats as the short reads does not span the entire repeat regions.

2. **Variant detection:** Identifying long structural variations in DNA sequence which are crucial for disease studies becomes more challenging with shorter reads. Structural variants (SVs) are block of genomic variations of length at least 50 basepairs that are rearranged in the genome. While short reads can capture SVs that span for a few hundred bases, they miss to detect the long SVs that are a few thousand basepairs long. Moreover, repeat regions on the genome are hotspots for SVs and finding the exact origin and the exact length of the SV within these repetitive regions is a difficult problem using short reads data. For example, finding the exact length and location of a deletion SV in a tandem repeat region of a few thousand bases long is a difficult problem using short reads that cannot span the entire region.

Emerging long read sequencing technologies, particularly, nanopore sequencing [5, 6] offers a potential solution to the limitations of the short read sequencing mentioned above by providing long reads (with average read length more than 15-kb and the longest read sequenced so far 2-Mb) that can span these repetitive regions. However, these long reads are riddled with a high error rate, thus, making alignment of low accuracy [7] and the downstream task difficult. For example, while nanopore sequencing has enabled fully automated assembly of some bacterial genomes, the assembly of human genome still produces many contigs that have to be scaffolded manually [8]. Another important downstream task is structural variant calling, where long reads can play an important role. However, present structural variant calling algorithms have low precision and recall due to noise in the reads [9] which fails to provide sufficient consensus from different reads in a highly repetitive regions of the genome to identify an SV by the variant caller. The assembly of long segmental duplications presents another important problem where long reads can bridge repeated regions but again becomes complicated due to read errors [10]. These challenges in nanopore sequencing reads because of the high error rates motivate for the development of algorithms that can utilize the error profiles of the nanopore sequencers for its downstream applications. The high-level contributions are highlighted in the following section.

1.4 Contributions

In nanopore sequencing, a nanopore is embedded into a membrane that permits the flow of ionic current through it when a voltage is applied across the membrane. When a DNA molecule migrates though the nanopore, different bases offer different resistance to this ionic current and therefore, result in variation of the current signals which are recorded. These current signal variations are decoded back to basepair sequence using base-calling algorithms to estimate the original sequence passing through the nanopore. Ideally, if only one nucleotide affect the current signal at a time and the rate at which the DNA migrates through the pore is constant, the DNA sequence migrating through the nanopore can be decoded accurately with high probability. However, several non-idealities of the nanopore sequencing technology introduces errors in the estimated DNA sequence by the base-calling algorithm. These non-idealities are discussed in more details in chapter 2 of this dissertation and they provide insights to incorporating the current-level modeling of the nanopore sequencer into its downstream applications such as alignment of read to reference genome for Structural Variant calling, read-to-read alignment for determining read overlaps for the assembly problem, or read-to-transcriptome alignments.

These errors by the base-calling algorithm in estimating the DNA sequence given the input raw current signals from nanopore sequencer occurs when there are more than one likely DNA sequence for the same observed current signal and the base-caller makes a hard decision to output a single DNA sequence from the list of ambiguity. However, if we view the hard decoded DNA sequences in nucleotide space from the lens of the current-level space, we maintain a list of all ambiguous DNA sequence that are likely for the given current signal, and therefore, rectifying the errors introduced by the base-caller in inferring the correct DNA sequence in the current-level space. This approach requires a current-level model of the nanopore sequencer to translate the decoded DNA sequences from the nucleotide space to the current-level space. Oxford Nanopore Technologies (ONT) has provided the current-level models from their base-calling algorithm for their nanopore flowcell except their current R10.4 flowcell. Moreover, Mao et. al. in [2] provides a nanopore model that incorporate the non-idealities in the sequencing technology and can be used in the absence of a nanopore model from the sequencing technology. Utilizing these insights about the error profiles in nanopore sequencing, we have contributed in developing algorithms for the following problems:

1. **QAlign:** We developed an algorithm that translates the DNA sequences to sequences in current-levels using the appropriate current-level model of the nanopore sequencer. These current-level sequences are then quantized to two or three levels to enable the use of state-of-the-art long read alignment algorithms developed for DNA sequences (or up to quantization level four). We further make a comparison in the improvement of alignment quality by our algorithm to the state-of-the-art methods in terms of contiguity measured by the length of the alignment as well as accuracy measured by the edit distance. We show that QAlign improves the alignment rate of the real nanopore reads to the reference from 80% with minimap2 (v2.18) on nucleotide sequences to 90% with quantized current-level sequences. Further, QAlign improves the average read-to-read overlap quality by up to 10.8%, and improves read-to-transcriptome alignment rates from 51.6% to 75.4%.

2. **HQAlign:** We developed an alignment algorithm specifically for SV detection using

the idea of current-level modeling of nanopore sequencers from QAlign. HQAlign improves the alignment method of QAlign and incorporates SV-specific changes to the alignment algorithm for efficient alignments. We show that HQAlign captures $4\% - 6\%$ complementary SVs across different datasets which are missed using the alignments from minimap2 (v2.24). Moreover, HQAlign improves the alignment rate with from 85.64% minimap2 (v2.24) to 89.35%.

Finally, we develop algorithm for detecting heterozygous variants using the nanopore reads data only which is a primitive step towards developing an algorithm for haplotype-resolved assembly of diploid human genome. A diploid genome has two copies of each chromosome – one inherited from each parent. Within the diploid genome, each gene exits in two copies or alleles – one from each parent. If the two alleles at a specific genetic locus are different then these alleles are referred to as heterozygous variants. The high error rates in the nanopore reads pose a significant challenge in accurately detecting these heterozygous variants which are "signature" of each haplotype or copy of the chromosome. The existing approaches towards haplotype-resolved assembly utilizes additional high accurate sequencing data of the same sample genome such as short Illumina reads or HiFi reads that has $< 1\%$ error rates to phase the assembly into two haplotypes based on the signature on the accurate reads. On comparison with the known heterozygous variants for the sample genome data in the study, our algorithm shows a precision of 0.70, recall of 0.90, and F1 score 0.78, for a minimum confidence of 50% on the heterozygous region determined by our algorithm. There is a trade-off in precision vs recall as we increase the confidence and a detailed analysis on this is provided in chapter 5 of this dissertation.

1.5 Outline of the dissertation

This dissertation is divided into six chapters including this introduction chapter. In **Chapter 2**, we study the mathematical model of the nanopore sequencer that incorporates the non-idealities of the sequencing technology and the error profile introduced due to these non-idealities in the output hard decoded nucleotide reads. **Chapter 3** develops a novel alignment algorithm of nanopore DNA reads to the reference, the read-to-read alignments, cDNA reads-to-transcriptome alignments that utilizes the error profile studied in chapter 2, and demonstrate the study of detailed analyses and improvements in the alignment algorithm performance using the current-level model of the nanopore sequencer. **Chapter 4** further improves the alignment algorithm specifically for accurate structural variant calling using the current-level model of the nanopore sequencer.

In **chapter 5** of this dissertation, we develop models and algorithms that identifies heterozygous variants from the whole genome nanopore data without using a reference genome or a draft assembly. This method is the primitive step for developing algorithms for haplotype-resolved genome assembly of human diploid genome using the long and noisy nanopore reads data only. Finally, **chapter 6** summarizes the dissertation and provides future directions to be explored based on the insights from this dissertation.

Parts of this dissertation are presented in [2, 1, 11].

CHAPTER 2

Mathematical model for Nanopore Sequencer

In nanopore sequencing technology, a DNA is transmigrated through a nanopore. There is an ionic current established across the nanopore and as the DNA migrates through the nanopore, the variations in the ionic current signals are recorded. The underlying DNA sequence migrating through the nanopore is then inferred by a base-calling algorithm which decodes the observed ionic current signal variations to DNA sequence. The performance of the base-calling algorithms is measured in terms of the number of plausible input DNA sequences given the observed current signals from the nanopore. In an ideal case, the design of the base-calling algorithm is to have a unique map between input DNA sequence and the observed current signals, therefore, resulting in the plausible number of DNA sequences to be exactly one given the observations, which corresponds to the true sequence. However, because of the noise and several non-idealities of the nanopore sequencers, there can be more than one number of plausible sequences for a given observation of current signals. This introduces a structured error profile, in contrast to the random independent and identically distributed (i.i.d.) error patterns, in the hard decoded nanopore DNA sequences which cannot be eliminated by consensus calling or increasing the coverage depth of sequencing.

In this chapter, we summarize the mathematical model of the nanopore sequencer from [2], which lays the foundations for the algorithms developed for alignment in chapter 3 and chapter 4 of this dissertation.

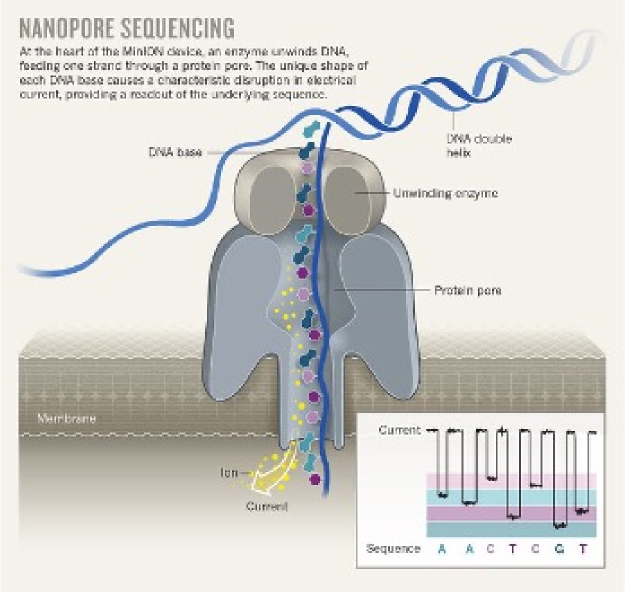

NANOPORE SEQUENCING
At the heart of the MinION device, an enzyme unwinds DNA, feeding one strand through a protein pore. The unique shape of each DNA base causes a characteristic disruption in electrical current, providing a readout of the underlying sequence.
DNA base
DNA double helix
Unwinding enzyme
Protein pore
Membrane
Ion
Current
Current
Sequence A A C T C G T

Figure 2.1: A simplified schematic of nanopore sequencer. The enzyme unwinds the double helix DNA as one of the strands migrates through the nanopore. The ionic current signal variations are recorded corresponding to the DNA sequence in the nanopore.

2.1 Nanopore model

2.1.1 Nanopore sequencer

A simplified schematic of the nanopore sequencer is shown in Fig. 2.1. A nanopore is inserted into a membrane that permits the flow of ionic current when a voltage is applied across the membrane. A strand of DNA is passed through the nanopore which causes modulations in the ionic current corresponding to different DNA protein in the nanopore. An enzyme unwinds the DNA and also moderates the rate at which DNA passes through the nanopore so that the current variations are recorded accurately.

In an ideal case for nanopore sequencing, the rate at which DNA migrates through the nanopore would be constant so that only one base of DNA affect the current at a time. This would enable unambiguous decoding of the DNA sequence with high probability. However, there are several non-idealities of the nanopore sequencer due to the physics of the nanopore and the enzyme as mentioned below:

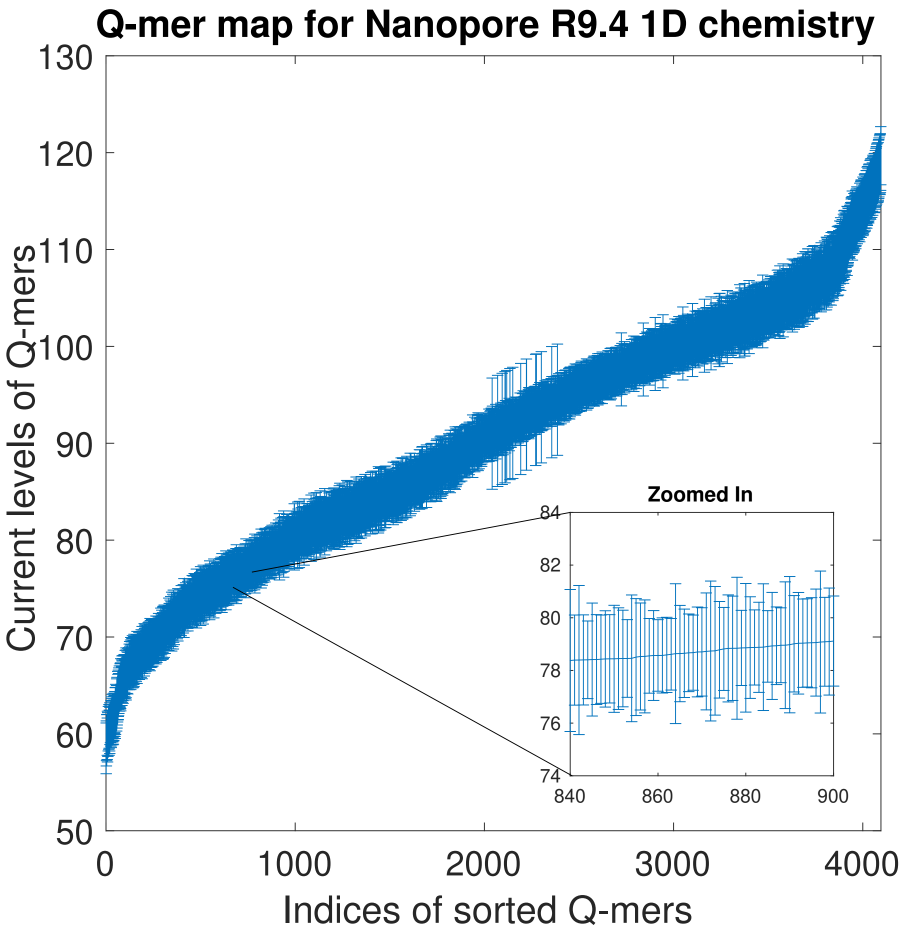


Figure 2.2: Q-mer map for Nanopore R9.4 1D flowcell ($Q = 6$). The Q-mers are sorted in the ascending order of the median current levels.

1. *Inter-symbol interference:* Ideally, we want a design of the nanopore where only a single nucleotide affects the current observation at a given time, however, the constriction of the nanopore is thicker than a single nucleotide. This results in the neighboring nucleotides also influencing the observed current signal at a given time. Let a Q-mer represents Q consecutive bases affecting the current signals at a given time, and statistically, $Q = 5$ or $Q = 6$ is modeled to decipher the current signals by the base-calling algorithm. There are 4 different type of nucleotides in DNA which results in $4^6 = 4096$ different Q-mers.

2. *Random dwelling time:* The amount of time spent by each Q-mer of the DNA sequence in the nanopore depends on the rate at which the enzyme unwinds the DNA and is therefore, not constant. Therefore, the rate at which the DNA sequence migrates through the nanopore is a stochastic process.

3. *Segment insertion and deletion:* The enzyme that controls DNA migration through the nanopore has some backtracking and significant skipping. Therefore, some Q-mers registers multiple repeats while some pass through the nanopore without recording any variations in the current signals. These effects induce insertion and deletion errors in the output of nanopore reads.

4. *Q-mer map fading:* A current level c_i is measured for each Q-mer q_i dwelling in the nanopore at time i, such that $c_i = f(q_i)$. However, this function f is not deterministic, and it can produce different current levels corresponding to the same Q-mer at different times which is similar to the *fading* effect in communication channels, *i.e.*, $c_i \sim \mathcal{D}(q_i)$. Fig. 2.4 shows the median current value for each Q-mer and its standard deviation as error-bars for $Q = 6$.

5. *Noisy samples:* Each observed current level is also subject to a random noise [6].

2.1.2 Mathematical model for nanopore

A nanopore sequencer can be modeled as a communication channel, with input being the DNA sequence to be measured and the output of the channel is the noisy current signals from the nanopore. The decoder then predicts the input sequence given the observed current signals. The mathematical model for nanopore sequencer is shown in Fig. 2.3 from [2]. Consider the DNA sequence represented as $X_1, X_2, \ldots, X_n$ of length n and $X_i \in \{A, C, G, T\}$ represents a nucleotide in DNA. It migrates through the nanopore channel whose mathematical modeling can be considered in the following steps:

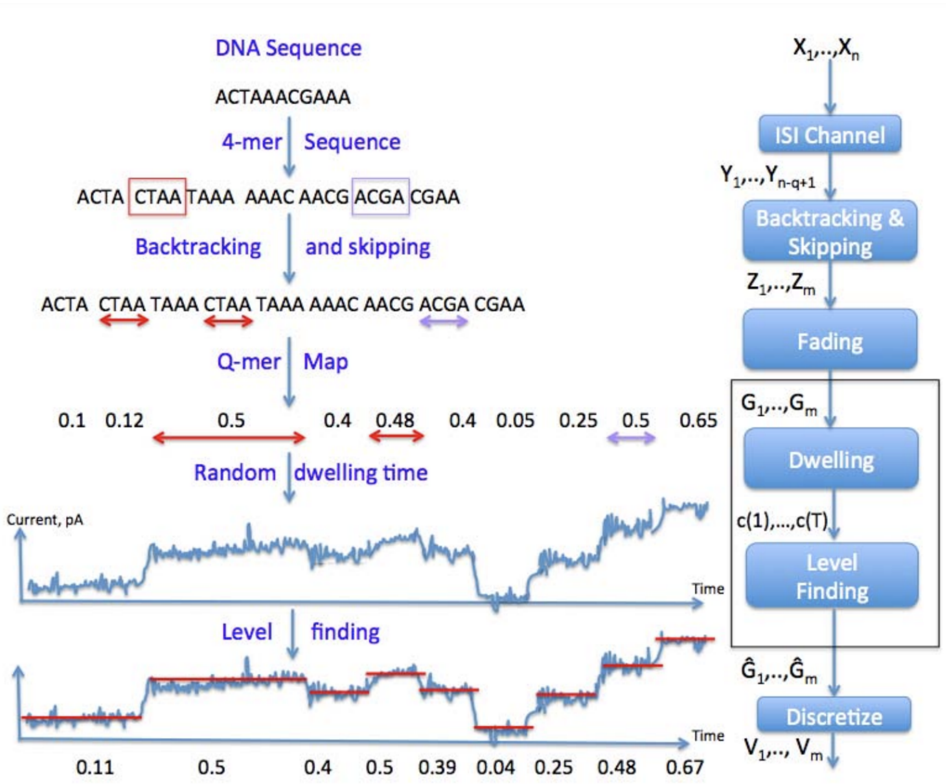


Figure 2.3: Mathematical model for nanopore sequencer for a toy DNA sequence from [2].

1. **Inter-symbol interference channel:** The nanopore observes input DNA sequence $X_1, X_2, \ldots, X_n$ as a sequence of symbols influenced by q neighboring nucleotides represented as $Y_1, Y_2, \ldots, Y_{n-q+1}$, where $Y_i = X_i, \ldots, X_{i+q-1}$, and length of Y_i is q.

2. **Segment insertion and deletion:** Some symbols in the sequence $Y_1, Y_2, \ldots, Y_{n-q+1}$ are missed or repeated because of the skipping and backtracking effects resulting in a sequence $Z_1, Z_2, \ldots, Z_m$ of length m, and each Z_i is a q-mer of length q. As shown in Fig. 2.3, the q-mer highlighted in red is backtracked and this results in insertion errors.

3. **Fading:** The sequence $Z_1, Z_2, \ldots, Z_m$ is measured by nanopore sequencer through variations in the current signals by translating the sequence $Z_1, Z_2, \ldots, Z_m$ from the lens of Q-mer map (shown in Fig. 2.4) and fading effects to a sequence of median current levels $G_1, G_2, \ldots, G_m$, where $G_i = f(Z_i)$. The function f represents the mean

value of the distribution for a given q-mer, *i.e.*, if $g_i \sim \mathcal{D}(Z_i)$, g_i represents a current level sampled from the distribution for q-mer Z_i, therefore, $G_i = \mathbb{E}_{g_i \sim \mathcal{D}(Z_i)}[g_i]$. As shown in Fig. 2.3, the q-mers highlighted in red and blue have similar current levels which results in ambiguity for the decoder to infer the correct input DNA sequence.

4. **Dwelling:** Each current level G_i dwells for a random amount of time in the nanopore and results in a current signal sequence $c(1), c(2), \ldots, c(T)$ of length T based on the sampling rate of the nanopore sequencer. This current signal $c(1), c(2), \ldots, c(T)$ is the raw output of the nanopore sequencer model.

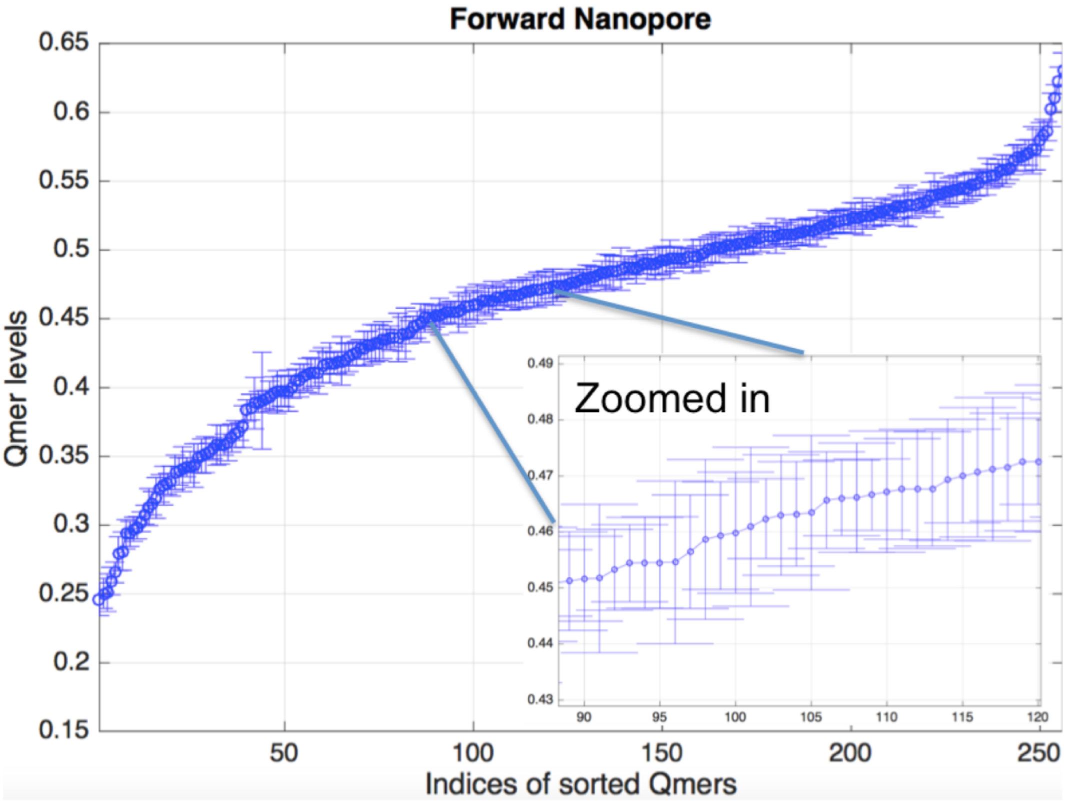
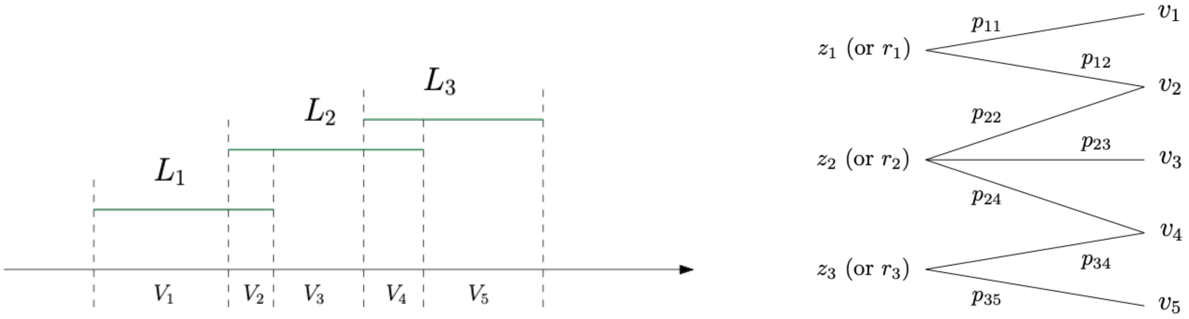


Figure 2.4: Mean and standard deviation of 256 Q-mer maps obtained from nanopore in [2]. There is significant overlaps in range of Q-mer maps.

However, decoding the DNA sequence from the raw current samples with random dwelling time is a harder problem for base-calling since it requires channels with infinite memory.

Therefore, the current-levels are first estimated $\hat{G}_1, \hat{G}_2, \ldots, \hat{G}_m$ from the raw current signals $c(1), c(2), \ldots, c(T)$ using change-point detection algorithms to find the transitions in the raw current signal and the mean value of the current samples is estimated within the transition to estimate the current-levels at the decoder. These estimated current-levels $\hat{G}_1, \hat{G}_2, \ldots, \hat{G}_m$ are given to the decoder to infer the input DNA sequence as $\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_{n'}$, where n' may or may not be equal to n. In an ideal case where the decoding does not have any ambiguity, we have the decoded DNA sequence $\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_{n'}$ to be equal to the input DNA sequence $X_1, X_2, \ldots, X_n$.

This mathematical model assumes level-finding algorithm to be perfect such that $\hat{G}_i = G_i$ making the channel in the box in Fig. 2.3 to be identity. Further, there are two more simplifications made to facilitate the analysis of the models, (i) the frequency of deletions due to skipping is higher than insertions due to backtracking, thus, the model ignores the backtracking effects and models skipping as an i.i.d. deletion process. (ii) Fading effects are modeled as independent random process with each Q-mer having its own uniform fading distribution (as shown in Fig. 2.4) with center being the mean value and the width being a constant multiple of standard deviation.



(a) (b)

Figure 2.5: (a) Partition in intersection of Q-mer map ranges from [2]. (b) DMC arising from the partition in (a) with transition probability $p_{ij} = l(v_j)/l(r_i)$ from [2].

The overlap between the current level range of different Q-mers, say range L_1, L_2, L_3 for Q-mers z_1, z_2, z_3, induce a partition on the set of possible current level values $\{v_1, v_2, v_3, v_4, v_5\}$

shown in Fig. 2.5 which causes the list size of plausible input DNA sequences given the observed current signal to be more than one. Since the fading process is assumed to be memoryless, the partition on the set of current levels for a Q-mer can be modeled as a Discrete Memoryless channel (DMC) with channel transition probability determined by $p(v|z) = \frac{l(v)}{l(z)}$, where $l(\cdot)$ denotes length of an interval and r_i denotes range of current level for Q-mer z_i. This gives a simplified nanopore channel model shown in Fig. 2.6.

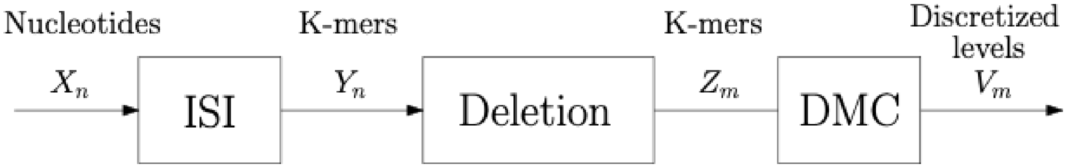


Figure 2.6: Simplified nanopore channel model from [2].

2.2 Insights to nanopore sequencing from information theoretical bounds derived in [2]

The information-theoretic analysis of nanopore sequencing based on the model design in [2] aims at understanding the fundamental limits of nanopore sequencing and providing insights into the design of better sequencing algorithms and nanopore devices. The theoretical results and its interpretation are summarized below (refer to [2] for proofs of the theorems below and detailed definition of the notations):

1. The achievable rates for a cascade of deletion channel and DMC channel without the ISI in Figure 2.6 gives a lower bound on nanopore channel capacity in the following theorem:

   Theorem 1. *For the cascade of a deletion channel with a DMC, the following is an achievable rate for each irreducible Markov transition matrix P on the input alphabet*

$\mathcal{Y}$:

$$C(P) = -\inf_{\gamma > 0} \frac{[(1-\theta)\gamma + \theta E(\gamma)]}{\log 2} - \theta H(\mathcal{Z}|\mathcal{V}) \tag{2.1}$$

where $\theta = 1 - p_d$, p_d is the deletion probability of the deletion channel, $\mathcal{X}$ is the finite alphabet of DNA symbol, $\mathcal{Y} = \mathcal{X}^k$ is the alphabet for k-mers, and $\mathcal{V}$ is the alphabet of DMC channel output in Figure 2.6 corresponding to discretized levels.

Interpretation: The achievable rate that reflects the sequencing capabilities of nanopore sequencer is obtained for i.i.d. uniform distribution on nucleotides to generate the k-mer transition matrix P. If we optimize the transition matrix P, then we obtain the achievable rate that can be used to measure the capability of the nanopore sequencer as a reader for a DNA storage system.

2. The achievable rates for the nanopore channels for different deletion probability p_d and for different ISI length k are computed. The larger k results in more correlation between the input symbols which increases the protection against the deletions. On the other hand, larger k increases the noise in DMC since there are more intersections in the range of Q-mer map in Figure 2.4, and therefore decreasing the achievable rates. This demonstrates the trade-off effect with increasing k in the design of nanopore. Therefore, a nanopore with small deletion probability p_d, achieves higher rates with the design with smaller k, whereas a nanopore with higher deletion probability requires larger k to achieve higher rates for which the intersection between k-mer intervals in the Q-mer map is required to be reduced to improve DMC quality.

3. The upper bound for nanopore channel capacity is derived using the upper bounds in the form of finite block mutual information below as the starting point:

Theorem 2. *For each n, $\bar{C}_n$ defined below is the capacity upper bound of the nanopore* *channel:*

$$\bar{C_n} = \frac{1}{n}\left[\max_{s_0}\max_{P_{X^n}} I\left(X^n; V^{(n)}|s_0\right) + \log(|\mathcal{S}|)\right] \tag{2.2}$$

where the input DNA symbol at each time n is represented as X_n, and the DNA sequence until time n as X^n with corresponding output of the nanopore channel in Figure 2.6 as $V^{(n)}$, s_0 is the initial state of the ISI channel which is known through the prefix adapters and has the alphabet $\mathcal{S} = \mathcal{X}^{k-1}$. Therefore, $\log(|\mathcal{S}|) = (k-1)\log(|\mathcal{X}|)$. Therefore, as n increases, the nanopore channel capacity is upper bounded by $C = \lim_{n\to\infty} C_n$:

$$C_n = \frac{1}{n}\max_{P_{X^n}} I\left(X^n; V^{(n)}\right) \tag{2.3}$$

with $C_n \leq \bar{C_n}$. For large n, the second term $\log(|\mathcal{S}|)$ is negligible, but the output alphabet size grows exponentially which is a major bottleneck for the computation on the case of nanopore channels.

2.3 Implications of the nanopore model to algorithm development

The work in [2] proposed the mathematical model for the nanopore communication channel described in this chapter, and derives the information-theoretic bounds for the reliable information rate analysis that can be achieved for the proposed model. This chapter proposed a mathematical model for the nanopore channel which provides insights to the physics and error profile inherent in the nanopore channel and motivates the development of algorithms for its downstream applications as shown in chapter 3 and chapter 4 of this dissertaiton, especially, the Q-mer map function f. The base-calling algorithm can make errors in decoding the DNA sequence when the current signals corresponding two or more different sequences are similar. Therefore, a structured error pattern is introduced in the output hard decoded reads by the base-caller which are difficult to resolve in the downstream applications using the consensus from the read coverage depth.

These errors can be taken into account for downstream tasks when we translate back to the current-level space from the hard decoded nucleotide space. The idea behind this change of space to current-level is to maintain a list of all plausible DNA sequences that could be inferred from this current-level sequence and develop the algorithms that can perform the alignments using these current-level sequences by first quantizing them to finite quantized levels, therefore, mitigating the errors introduced by the base-callers in hard decoding of the DNA sequences. This idea is explained in detail in the subsequent chapters with detailed analysis of the performance of the algorithms that take these error profile of the nanopore sequencer into account for the purpose of read alignments and its application in structural variant calling.

CHAPTER 3

QAlign: Aligning nanopore reads accurately with current-level modeling

3.1 Introduction

In genomic data analysis, aligning DNA / RNA-seq reads to a genome / transcriptome is a key primitive, that precedes many downstream tasks, including genome / transcriptome assembly [12, 13] and variant calling [14, 15, 16]. Getting accurate read alignment is difficult especially in repetitive regions of the genome, due to the short length of the reads obtained via high throughput sequencing. Emerging sequencing technologies, particularly, nanopore sequencing [5, 6] offers a potential solution to this problem by providing long reads (with average read length 10-kb and the longest read sequenced so far up to 4Mb) that can span these repetitive regions. However, these long reads are riddled with a high error rate, thus, making alignment of low accuracy [7] and the downstream task difficult. For example, while nanopore sequencing has enabled fully automated assembly of some bacterial genomes, the assembly of human genome still produces many contigs that have to be scaffolded manually [8]. Another important downstream task is structural variant calling, where long reads can play an important role. However, present structural variant calling algorithms have low precision and recall due to noise in the reads [9]. The assembly of long segmental duplications presents another important problem where long reads can bridge repeated regions but again becomes complicated due to read errors [10].

In this chapter, we study a novel method for aligning nanopore reads that takes into

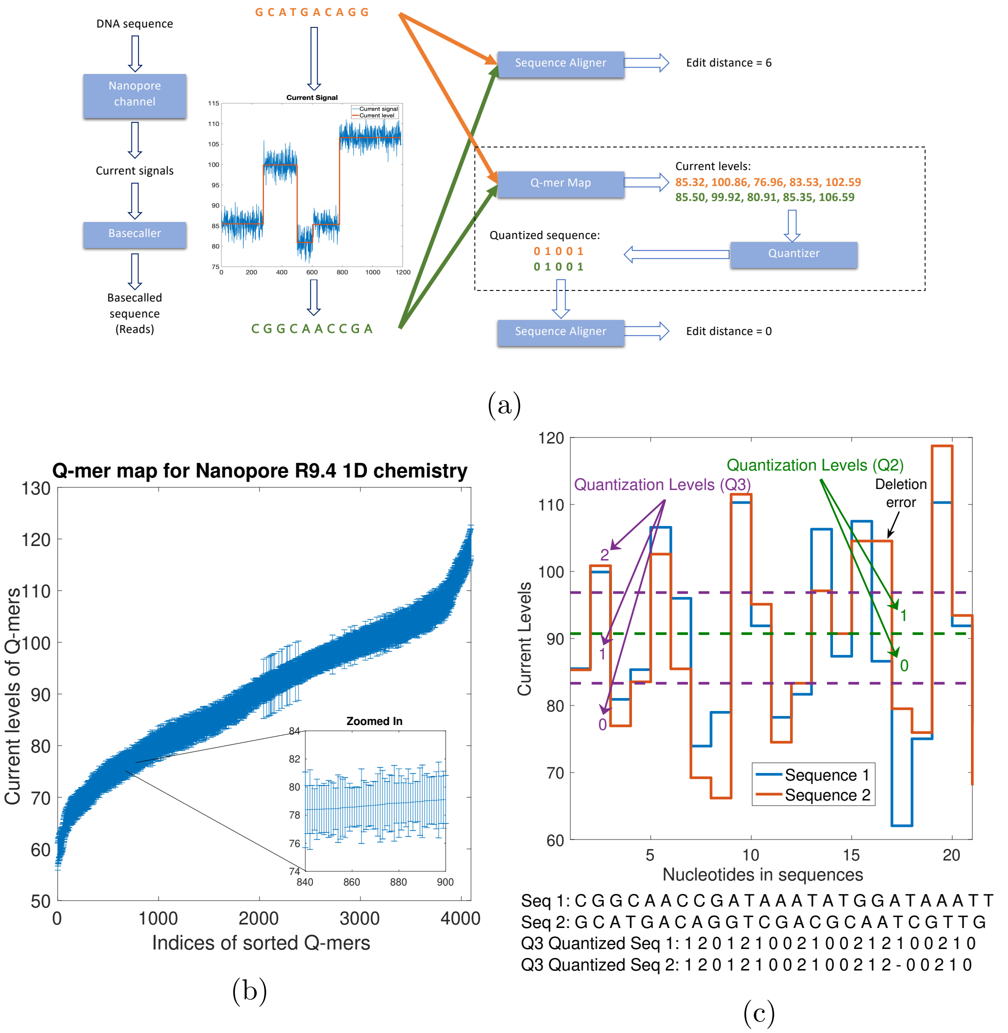


Figure 3.1: (a) An example to illustrate the error-profile in nanopore base-called reads, and the ability of QAlign to perform accurate alignment despite of the errors (the edit distance used here is to demonstrate accuracy of the alignment, however, the *nucleotide edit distance*, which is used as a metric for read-to-genome and read-to-transcriptome alignment, is computed in the nucleotide domain for the quantized alignments as well). (b) Q-mer map for Nanopore R9.4 1D flow cell (for $Q = 6$). It represents the physics of nanopore. The median current value along with the standard deviation (as error bars) are plotted for every 6-mers in the Q-mer map for R9.4 1D nanopore flow cell. Note that the difference between the median current levels of any two consecutive Q-mers is very small. (c) An example showing the two different nucleotide sequences have similar current levels (therefore similar quantized sequences).

account the particular structure of errors that is inherent in the nanopore sequencing process. In many of the long read aligners, the read errors are modeled using insertions, deletions and substitutions which happen at differing rates. However, in nanopore sequencing, many errors induced have structure, which is missed by viewing the errors as independent insertions, deletions and substitutions. In the nanopore sequencer, the current level depends on a Q-mer (a set of Q consecutive nucleotide bases which influence the current measurement in the nanopore). This is due to the physics of the nanopore sequencing, where a set of DNA base-pairs together influence the current output of the nanopore reader [17, 18, 2] (*e.g.,* occupying the nanopore width). Therefore, the output current depends on a set of DNA base-pairs (Q-mer) influencing it. The current reading, which is used by a de-novo base caller for decoding, therefore could cause structured errors, especially between Q-mers that have similar outputs. This confusability between different Q-mers, is captured by the so-called Q-mer map. In Figure 4.1 (b), the median current levels for various Q-mers are plotted and it is clear that there is significant overlap in the current levels observed when different Q-mers are passed through the nanopore. These overlaps are one source of structured errors in the sequencer and can be fundamental since they can be indistinguishable by any de-novo sequencer.

The novel alignment strategy that we study in this chapter takes into account the structure of the Q-mer map in order to perform better alignment. In Figure 4.1 (a), we give an example where a DNA sequence (GCATGACAGG) gets wrongly sequenced as a completely different sequence (CGGCAACCGA) due to this error mode of the nanopore sequencer. Ideally, we would like to maintain the list of “equivalent” Q-mers that could have plausibly caused the observed current readings. However, this is infeasible as this would entail changing the de-novo sequencing process itself to output either multiple possible reads, or give soft information about different possibilities. This is difficult, as sophisticated de-novo sequencing have been developed using artificial neural networks, which have been optimized for read error-rate performance [19]. Moreover, for a modular approach, we would not want to change the de-novo sequencer for different downstream applications. Therefore, we take a *different* approach to resolve this problem, by using the de-novo sequenced read as the *input* to our strategy. We then deterministically *convert* this de-novo sequenced nucleotide read into a current value using the Q-mer median current level of the corresponding Q-mer (i.e. the Q-mer map as in Figure 4.1b). We further *quantize* these resulting current values from continuous values into properly chosen discrete levels. This is illustrated in Figure 4.1c. In this work we use 2 to 3 levels of discrete values for the quantization, which is determined based on the Q-mer map. Now, given this new discrete representation of the de-novo reads, we develop the new alignment algorithm, whose workflow is illustrated in Figure 4.1a.

A natural question is why this should help, since we are processing the de-novo reads which are erroneous, and we are *not* using any additional soft information, such as raw current values from the nanopore reads themselves. The basic insight is that the translation of the nucleotide reads to current levels enables grouping together reads that are confusable given the structure of the Q-mer map of the nanopore sequencer. For example, when we have two reads illustrated in Figure 4.1c, if the de-novo sequencer has chosen one of the two equally-likely sequences as the nucleotide read, it is clear that the alternate read, which has significant edit distance (in the nucleotide domain) is actually quite close when viewed from the lens of the Q-mer map, as captured by our quantized conversion process. Therefore, this process naturally groups together reads that could have been confused, and uses this as the input to our alignment algorithm, QAlign. Therefore, this reduces the effect of the errors by recognizing one structure in the error process. Note that QAlign builds an overlay layer on top of any alignment algorithm in order to align based on current levels implied by the reads instead of directly aligning the reads. Though we illustrate our ideas using the Minimap2 aligner [20], this principle can be implemented with *any* other long-read aligner such as GMAP [21].

We show that QAlign gives rise to significant performance improvements across a variety of alignment tasks including read-to-genome, read-to-read and read-to-transcriptome alignment as well as different datasets spanning from R7 nanopore sequenced data (Figure 3.9) to R9.4 data.

QAlign shows significant improvement in read-to-genome alignment rates for datasets where Minimap2 alignment rate is low (improving up to around 90% for four real datasets). Furthermore, the alignments are also of higher quality: QAlign shows up to around 18% lower normalized edit distance than Minimap2 as well as longer alignments.

For read-to-read alignments, QAlign is able to align around 3.6% more overlaps between read pairs with a high overlap quality (refer to Methods for a description of the overlap quality) where Minimap2 is either unable to align the read overlaps or aligns with a low overlap quality. We show that a hybrid alignment strategy which combines QAlign and Minimap2 can improve the metric even further to around 4.6% (Figure 3.18).

For read-to-transcriptome alignments, our method achieves 90% alignment rate as opposed to 82.6% with mouse 2D reads and 75.4% as opposed to 51.6% with Human 1D reads. Furthermore, the alignments are also of higher quality: QAlign shows 13.27% lower normalized edit distance than Minimap2 as well as longer alignments for Human 1D data.

In this study, we focus on the improvement of long read (in particular the Nanopore long read) alignment. To the best of our knowledge, there is no existing aligner, specifically designed to handle the error modes introduced in nanopore sequencing. There is, however, some work on incorporating the nanopore current levels in downstream tasks including post-processing of assembly by Nanopolish [22]. Nanopolish has demonstrated that utilizing the current levels can reduce assembly errors. The major difference of our work with Nanopolish is the level at which the current-level information is taken into account. Since we take into account current-level information while performing alignment, we are able to get substantially more overlaps which can lead to potentially better assembly of contigs whereas Nanopolish is only able to correct fine errors.

3.2 QAlign Algorithm

The QAlign strategy consists of two steps including the conversion of the nucleotide sequences to quantized (e.g. 2 levels or 3 levels) sequences in the first step. The next step is the alignment of the quantized sequences for various alignment tasks such as read-to-genome, read-to-read, and read-to-transcriptome.

3.2.0.1 Nanopore Sequencer

The Oxford Nanopore Technology is based on DNA transmigrated through a nanopore, which results in the changes in the ionic currents through the pore [6]. The current changes are caused by the different nucleotides that are partially blocking the pore as the (single-strand) DNA sequence to be measured is migrated through the nanopore. An enzyme slows down the motion of the DNA through the pore, so that the variations in the current signals can be measured accurately [6]. Base calling algorithms [23] are developed to infer the nucleotide sequence (A,C,G,T) from the measured current changes.

Ideally, it is desired that the DNA is migrated through the nanopore at a constant rate and the current signal recorded at a given time is only affected by a single nucleotide in the nanopore. Consequently, the DNA sequence can be decoded unambiguously with high probability.

However, in reality, there are several non-idealities due to the physics of the nanopore and the enzyme. (i) *Inter-symbol interference:* Since the nanopore is bigger than the size of a single nucleotide, the observed current at a given time is influenced by multiple (neighbor) bases or Q-mer (where Q could be 4,5, or 6). (ii) *Random dwelling time:* The amount of time spent by each Q-mer of the DNA sequence in the nanopore may vary. So the rate at which the DNA sequence migrates through the nanopore is a stochastic process. (iii) *Segment insertion and deletion:* There are segments that are repeated as well as segments that migrate through the nanopore without registering a current reading. This results in redundant and missing segments. (iv) *Q-mer map fading:* The measured current level is a function of the corresponding Q-mer in the nanopore. However, this function is not deterministic, i.e., for the same Q-mer dwelling in the nanopore at a different time may produces a current level with some variations. This is also reflected in the Q-mer map (Figure 4.1c), where we have plotted the median value of the current levels along with the variances as the error-bar. (v) *Noisy samples:* Each observed current level is also subject to a random noise [6].

3.2.1 Quantization

The nucleotide sequences are inferred from the nanopore current signals by base-callers, therefore, using a Q-mer map to translate the base-called sequences to the current levels implicitly maintains all of the "equivalent" base-called sequences that could be inferred from the observed current levels. These current levels can be quantized to an alphabet of finite size (Figure 4.1a,c).

Mathematically, the quantization process is as follows. Let $\Sigma = \{A, C, G, T\}$ be the alphabet of nucleotide sequences. For a symbol $x \in \Sigma$, let $\bar{x}$ be the Watson-Crick complement of x. A string $s = x_1 x_2 \dots x_n$ over Σ is called a *DNA sequence*, where $|s| = n$ is the string length and the *reverse complement* of s is $\overline{s} = \overline{x_1 x_2 \dots x_n} = \overline{x}_n \overline{x}_{n-1} \dots \overline{x}_1$. Let $p(s)$ be a list of all Q-mers (e.g. Q=6) in the string s, sorted by their occurances. For example, $p(s) = k_1 k_2 \dots k_{n-Q+1}$ and each Q-mer $k_i = x_i x_{i+1} \dots x_{i+Q-1}$ for $i = 1, 2, \dots, n-Q+1$. Now, we define $f : \Sigma^Q \to \mathbb{R}$ as the Q-mer map [1], which is a deterministic function that translates each Q-mer (k_i) to the (median) current level (Figure 4.1b). Now, let $C(s) = c_1 c_2 \dots c_{n-Q+1}$ be the sequence of the current levels, so that $c_i = f(k_i)$ for $i = 1, 2, \dots, n-Q+1$. The current sequence C can be further quantized into $w(s) = q_1 q_2 \dots q_{n-Q+1}$ by applying a thresholding function $q_i = g(c_i)$. The thresholding can be binary ($q_i \in \{0, 1\}$) or ternary ($q_i \in \{0, 1, 2\}$)

[1] Q-mer map is determined by the chemistry of the nanopore flow cell, and is therefore dataset dependent, *i.e.*, the Q-mer map for sequencing using R9 flow cell is different from Q-mer map for sequencing using R9.4.1 flow cell. The Q-mer maps used in this work are generated by [22].

(Figure 4.1c). We define $w(\overline{s})$ as the reverse complementary of a quantized sequence $w(s)$, so $\overline{w}(s) = w(\overline{s})$.

3.2.2 Alignment

We can now use the aligners (e.g. Minimap2) to align the quantized sequences. It is important to note that these aligners inherently performs the alignment of the query sequence (e.g. s_1) to the reference sequence (e.g. s_2) and also aligns the *reverse complement* ($\overline{s}_1$) to the reference (s_2). For the corresponding quantized sequences, aligners need to align the query sequence (e.g. w_1) and its reverse complementary (e.g. $\overline{w}_1$) *explicitly* to the reference (e.g. w_2), in order to take care of the Q-mer map for both w_1 and $\overline{w}_1$ properly.

The performance of such an aligner can be evaluated by comparing the alignments of the nucleotide sequences s_1 onto s_2 to the alignments of their quantized sequences w_1 onto w_2 union with $\bar{w}_1$ onto w_2, respectively, using appropriate performance evaluation metrics.

3.2.2.1 Read-to-Genome Alignment

We apply QAlign to the task of read-to-genome alignment. Given a nucleotide read r and the reference nucleotide genome G, we first obtain r^Q (the quantized *template* strand of the read, $r^Q = w(r)$) and $\overline{r}^Q$ (quantized *reverse complement* strand of the read, $\overline{r}^Q = \overline{w}(r)$) from r, and obtain G^Q (quantized reference genome) from G. We next align r^Q and $\overline{r}^Q$ respectively to G^Q using Minimap2. Lastly, we aggregate the results from both the template and reverse complement outputs to determine the best alignment for each read.

Note that the quantized alignment procedure differs from the direct nucleotide alignment process in two ways. First, the nucleotide alignment does not require Minimap2 to additionally align $\overline{r}$ to G explicitly. Second, the quantized alignment employs a different seed length (e.g. minimizer length `k` in Minimap2) to ensure that the computation time for quantized alignment is similar as nucleotide alignment (see Table 3.4 for the details of computation time versus seed lengths).

We define several terms that are crucial for later performance analyis, mainly including *well-aligned*, *normalized edit distance*, and *normalized alignment length.*

Consider in Figure 4.2a, *Read 1* aligns at location i_1 through j_1 on the genome (we can get these locations from Minimap2 output). We say that the read is *well-aligned*, if at-least 90% of the read is aligned onto the genome (i.e., $j_1 - i_1 \geq 0.9(\text{length}(\textit{Read 1})))$, and has either the *(approximate) normalized edit distance from Minimap2* (i.e. number of unmatched bases, normalized with read length, based on Minimap2 output) is less than a threshold value or the mapping quality from Minimap2 is high (greater than 20). The filtering for the well-aligned reads using this distance and mapping quality is incorporated to eliminate some spurious alignments from Minimap2. Note that the (approximate) normalized edit distance from Minimap2 is specific to nucleotide or quantized alignment. For example, for nucleotide sequences the value returned by Minimap2 is in nucleotide domain; the value returned by Minimap2 is in $Q2$ domain for the $Q2$ sequences. Therefore, different filtering threshold values are used - 0.48 for nucleotide sequence, 0.25 for $Q2$ sequence and 0.35 for $Q3$ sequence (Figure 3.23 and 3.24).

In order to compare the quality of the alignments at fine-grained level, we further define *Normalized edit distance*[2]. The normalized edit distance for nucleotide alignment is $\frac{\text{edit\_distance}\{r;G[i_1:j_1]\}}{\text{length}(r)}$ and for quantized alignment is $\frac{\text{edit\_distance}\{r;G[i_1^q:j_1^q]\}}{\text{length}(r)}$, where i_1, j_1 and i_1^q, j_1^q are locations obtained from nucleotide and quantized alignment respectively. Similarly, we define *Normalized edit distance of aligned read* for nucleotide alignment as $\frac{\text{edit\_distance}\{r[i_1':j_1'],G[i_1:j_1]\}}{\max(j_1-i_1,j_1'-i_1')}$ (Figure 4.2a) and for the quantized alignment as $\frac{\text{edit\_distance}\{r[i_1':j_1'],G[i_1^q:j_1^q]\}}{\max(j_1^q-i_1^q,j_1'-i_1')}$. As evident from the definitions, these metrics for both nucleotide and quantized alignment are calculated all in nucleotide domain (unlike the approximate normalized edit distance from Minimap2, which is domain specific). Specifically, for quantized alignment, we leverage the information about

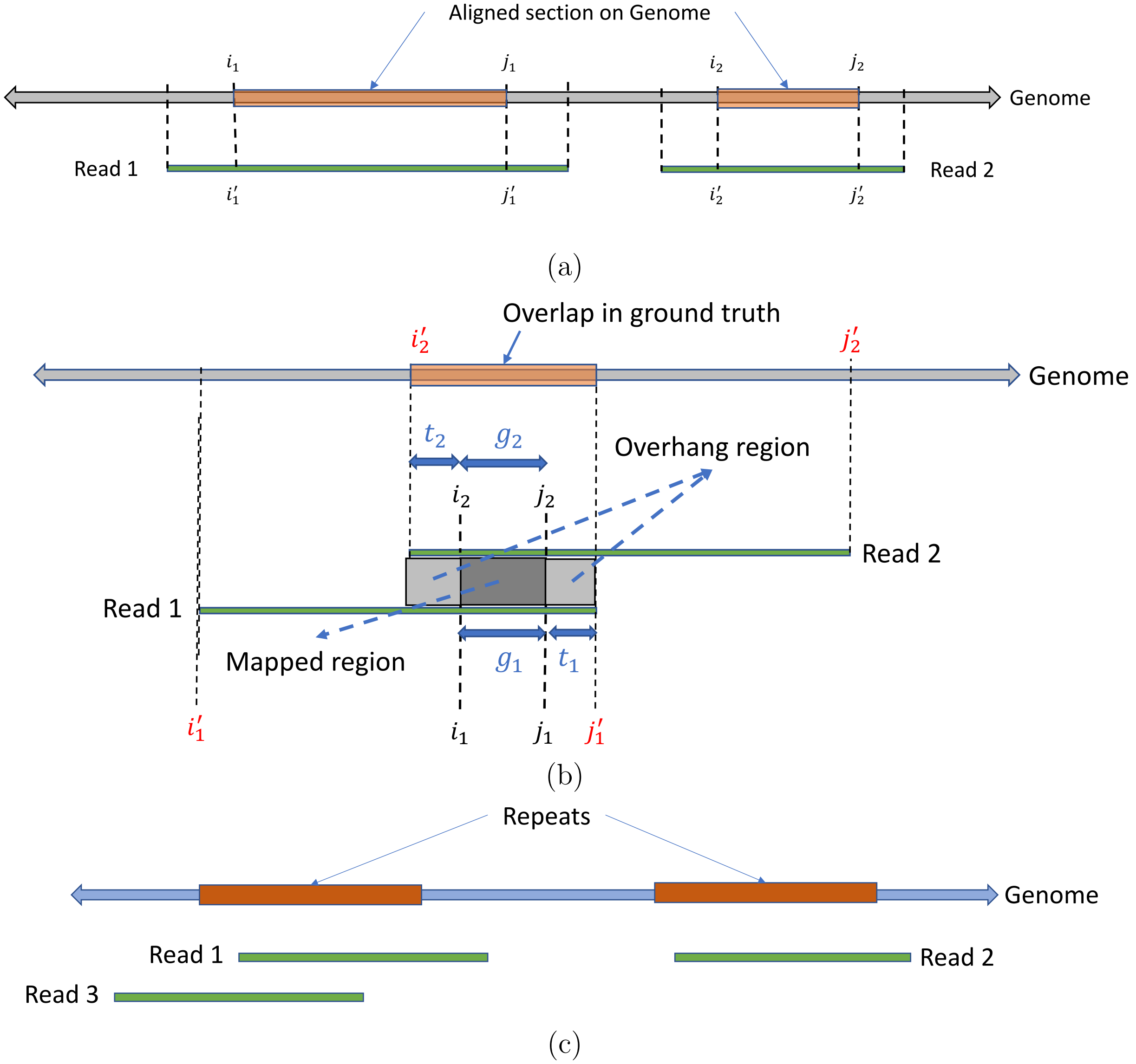
[2] This is different from *approximate normalized edit distance from Minimap2* to filter for *well-aligned* reads.

the alignment location on genome (i.e. i_1^q and j_1^q) to calculate the normalized edit distance between the nucleotide read and the corresponding aligned section on the nucleotide genome.

Another metric at fine-grained level is *normalized alignment length*, which is the ratio of the length of the section on genome where a read aligns to the length of the read. It is $\frac{j_1 - i_1}{len(r)}$ for nucleotide alignment, and $\frac{j_1^q - i_1^q}{len(r^Q)}$ for quantized alignment. A contiguous alignment tends to have this metric as 1.

We have been discussing the nanopore 1D reads for read-to-genome alignment so far. There are also 2D reads (e.g. r), which are the consensus calling using the 1D reads from both the *template* strand (e.g. r_t) and the *complement* strand (e.g. r_c). For the read-to-genome alignment algorithm of the 2D reads using QAlign, the experiment pipeline has been modified so that the error profile introduced in the sequencing of the 1D reads can be mitigated. Specifically, the quantized reads from both the template strand (e.g. r_t^Q) and the complement strand (e.g. r_c^Q) are aligned to the quantized genome (e.g. G^Q) individually using Minimap2. The union of the two alignments[3] is considered as the output of the QAlign algorithm. In case there is no overlap in the alignments of the reads from the individual strands, both the genome sections are given as the output of the QAlign algorithm, as the 2D consensus read might align to either of these sections. Since QAlign yields the genome section as the union of the two alignments, it could be much larger (nearly twice) than the read length. Therefore, the genome section needs to be further refined by the local alignment of the 2D consensus read onto the section. The performance evaluation of QAlign is mainly based on the normalized edit distance between the 2D consensus read and the refined genome section (the results using this method for the 2D read alignment onto genome are discussed in Figure 3.6 and 3.7).

[3] For example, alignment regions are [0, 2] and [1, 3], and union is [0, 3].



(a)

(b)

(c)

Figure 3.2: (a) An example of read-to-genome alignment. (b) An example of read-to-read alignment. (c) An example for head-to-tail alignments between reads.

3.2.2.2 Read-to-Read Alignment

We apply QAlign to read-to-read alignment as the second alignment task, which provides overlaps between reads that are typically necessary for genome assembly.

The alignments between the nucleotide reads r_1 and r_2 (or between the quantized reads

r_1^Q and r_2^Q) are obtained using Minimap2 as well. Similar to read-to-genome alignment, the quantized alignment not only aligns r_1^Q to r_2^Q, but also needs to align $\overline{r_1}^Q$ to r_2^Q. In addition, the quantized alignment employs a seed length (e.g. the minimizer length '`k`' in Minimap2) different from nucleotide alignment so that the computation time in both nucleotide and quantized regimes is maintained to be similar (see Table 3.5 for detailed analysis of computation time versus `k`).

For the algorithm evaluation purpose, we need to have the ground truth, which is unknown. One way to judge the quality is to compute the normalized edit distances of alignment overlaps. However, this is not only computationally expensive but also suffers from false alignments between reads from repeated regions. Instead, we leverage the read-to-genome alignments to build the ground truth for read-to-read alignment. Specifically, all of the reads are firstly aligned to the genome via both the nucleotide alignment and the quantized alignment. The reason behind performing the read-to-genome alignment in both the nucleotide and the quantized domain is to ensure that more read alignments are captured, as there can be some alignments that are captured/contiguous only in quantized alignments and vice-versa. For the experiments, we focus on a section of the genome G (say, G_1=G[1:1000000]) to find all the reads aligning onto G_1. Assume there are n_1 and n_2 number of reads aligned to G_1 in nucleotide domain and in quantized domain, respectively, with normalized edit distance (in nucleotide domain for both methods) less than 0.48, which indicates the found alignment of the reads are better than the alignment of two random DNA sequences (see Figure 3.23). Now, we randomly choose n reads from a union of n_1 and n_2 reads such that $n \approx \frac{N \times d_{cov}}{L}$, where d_{cov} is the required coverage depth, N is the length of the genome section G_1 (i.e. 1000000), and L is the average length of the $n_1 \cup n_2$ reads.

To estimate the ground truth, consider the alignment of *Read 1* (r_1) and *Read 2* (r_2) onto the genome as shown in Figure 4.2b, where the alignment locations of the reads on the genome are (i'_1, j'_1) and (i'_2, j'_2), respectively. We say that the reads are *overlapping in the ground truth* if there is an overlap (denoted as l) of at least 100 bases, where $l = \min(j'_2 - i'_1, j'_1 - i'_2)$. For reads that have overlaps in both nucleotide and quantized alignment, denoted as ($l^{nucleotide}$) and (l^{Q}) respectively, the larger one is chosen as $l = \max(l^{nucleotide}, l^{Q})$).

Provided the ground truth, we can compute the Precision and Recall to make a comparison between the two methods. Precision is defined as the fraction of overlaps in the ground truth among the overlaps determined by the algorithm. Recall (also known as *sensitivity*) is the fraction of overlaps in the ground truth that are determined by the algorithm.

The read-to-read alignment will label two reads to have an overlap (different from the overlap used to find ground truth) if the length of the 'Mapped Region' is at least 90% of the 'Mapped Region' plus the 'Overhang Region' (Figure 4.2b, i.e., $g_1 \geq 0.9(g_1 + t_1 + t_2)$ and $g_2 \geq 0.9(g_2 + t_1 + t_2)$, or equivalently $t_1 + t_2 \leq 0.1(\min(g_1, g_2))$. For evaluation, we define another metric called the *overlap quality* (denoted as X) as $\frac{(g_1-d_1)+(g_2-d_2)}{2l+d_1+d_2}$ where [4] the overlap quality measures how well the reads are aligned with respect to each other, compared to the alignment in ground truth. Ideally, it is desired to have the value of overlap quality close to 1. We also define the *average overlap quality*, which is the expected value of the overlap quality (i.e. $\mathbb{E}[X] = \int \mathbb{P}\{X > x\}dx$), or the area under the complementary CDF of X.

It is possible that two reads will be falsely aligned especially when they are from repetitive regions. To remedy this, we only consider head-to-tail alignment between reads. For example in Figure 4.2c, three reads *Read 1*, *Read2* and *Read 3* have been sequenced where *Read 1* and *Read 2* are from repetitive regions. Consequently after read-to-read alignment, there will be an overlap between *Read 1* and *Read 2* that can be filtered out since it is not a head-to-tail alignment. However, there will also be another false positive overlap between *Read 3* and *Read 2*, which will not be removed as it satisfies the head-to-tail alignment condition. In order to further reduce the number of false positives of read-to-read alignments, the (approximate) normalized edit distance provided by Minimap2 is used for extra filtering (see Figure 3.23 and 3.24).

[4] Empirically d_1 and d_2 tend to be simply zero

In addition to reduce false positives, the read-to-read alignment results can be further improved by implementing an Ensemble model, which is able to capture the best alignment (e.g. longer length of 'Mapped Region') from both methods of the nucleotide alignment and the quantized alignment, as well as to incorporate the alignments that are complementary in either method (see Figure 3.18).

3.2.2.3 Read-to-Transcriptome Alignment

Applying QAlign strategy to the third task of the RNA read-to-transcriptome alignment is analogous to the DNA read-to-genome alignment. This is not the spliced alignment of the reads to the genome; instead all of the RNA reads are aligned to the transcriptome. Since the ground truth is unknown for the alignments, we use *normalized edit distance*, and *normalized alignment length* as the evaluation metric.

3.3 Results

In this section, we demonstrate and discuss the results for (i) DNA Read-to-Genome alignment, (ii) DNA Read-to-Read alignment, and (iii) RNA Read-to-Transcriptome alignment using QAlign and Minimap2.

3.3.1 Datasets

Datasets for DNA-seq alignments: We use five datasets for DNA read-to-genome and read-to-read alignment: (1) MinION sequencing of K. Pneumoniae DNA using R9.4 1D flow cell [24], (2) MinION sequencing of E. Coli DNA using R9 2D flow cell [25], (3) MinION sequencing of E. Coli DNA using R9.4 1D flow cell [26], (4) MinION sequencing of Human genome using R9.4.1 flow cell [27], and (5) Simulated read data from GRCh38 chromosome 1 using Deep Simulator [28] for benchmarking the performance of QAlign.

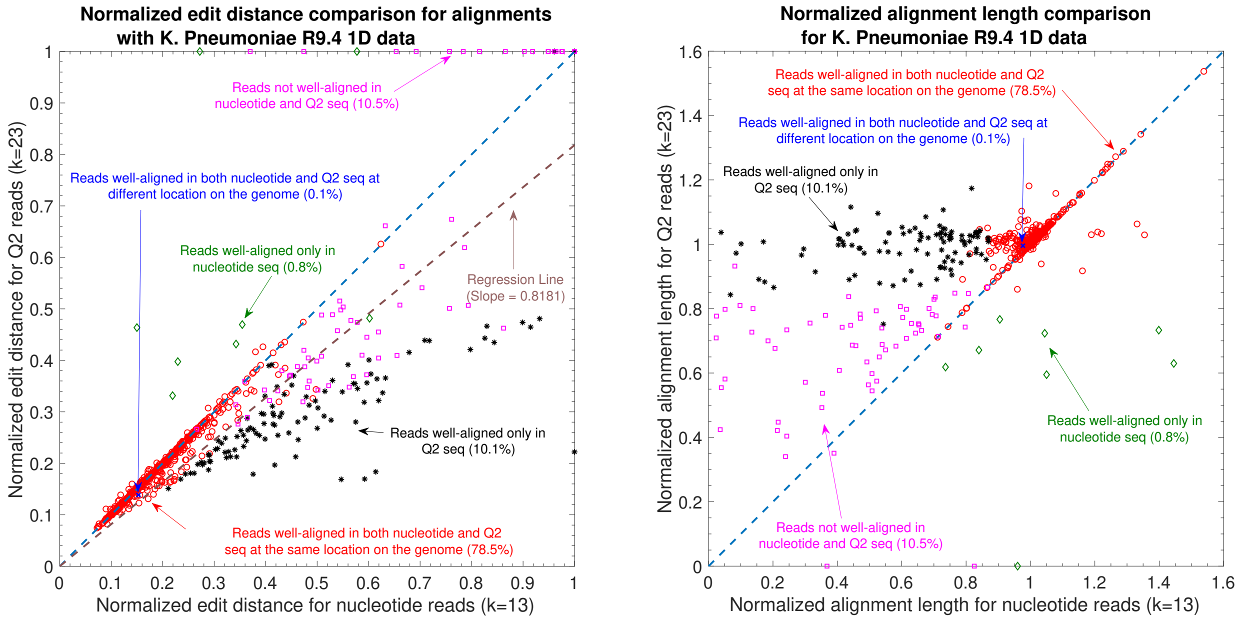
Datasets for RNA-seq alignments: The experiments are based on the RNA reads obtained by MinION sequencing of human cDNA using R9.4 1D flow cell [29], and MinION sequencing of mouse cDNA using R9.4 2D flow cell [30] (SRA access No. SRR5286961).

Compared to DNA-seq datasets, RNA-seq datasets carried out using nanopore sequencing are relatively rare. We select these datasets because they have relatively complete reference transcriptome (e.g. for human there are 200,401 annotated transcripts [31] and for mouse 46,415 annotated transcripts from UCSC Genome Browser [32]), and the corresponding Q-mer map models [33] are available for quantization.

3.3.2 DNA Read-to-Genome Alignment

The alignment of DNA reads to the genome is a task with wide-ranging applications in sequencing experiments. It is a required step in variant calling pipelines [34], in particular structural variant calling can benefit significantly from long reads offered by the nanopore sequencing platform [9]. It is also useful in calling variants in long segmental duplications [10], where long duplications necessitate long reads to resolve the repeat ambiguity. Another application for DNA read-to-genome alignment appears in reference matching - for example, in meta-genomics, in estimating which reference species is present in the sample.

The results are illustrated in the Figure 3.3 and Table 3.1. At a coarse level, the performance is measured by the fraction of the reads that have been well-aligned by the algorithm. A read is said to be well-aligned if at-least 90% of the read is aligned to genome and has either the (approximate) normalized edit distance from Minimap2 (i.e. number of unmatched bases, normalized with the read length) below a threshold value or the mapping quality from Minimap2 is high (see Methods). QAlign is shown to significantly improve the fraction of well-aligned reads - in particular, in the K.Pneumoniae R9.4 1D dataset, this metric improves to 88.7% from 79.4%. In the E. Coli R9.4 1D dataset, it improves to 84.2% from 79.2%; in the E. Coli R9 2D dataset, the numerics improves to 91.8% from 82.6%, and for the human R9.4.1 dataset, it improves to 87.95% from 85.70%. For the benchmark with the simulated



(a) (b)

Figure 3.3: **Nanopore long DNA reads alignment onto Genome.** (a) Comparison of normalized edit distance for K. Pneumoniae R9.4 1D reads data. Smaller values for *normalized edit distance* is desirable as it represents better alignment. The slope of the regression line is < 1, therefore, representing better alignments with $Q2$ than nucleotide alignments for same reads on average. (b) Comparison of normalized align-length on genome for K. Pneumoniae R9.4 1D reads data. Normalized alignment length of 1 is desirable as it represents that entire read is aligned. Majority of the reads are above $y = x$ line representing longer alignment length in $Q2$ than nucleotide alignment.

data, the numerics improves to 84.35% from 69.04% (refer to Table 3.1).

The results in Figure 3.3a-b compares the quality of the alignments using Minimap2 and QAlign at a fine-grained level for the K. Pneumoniae dataset (plots for other datasets are available in Figure 3.15-3.13).

Specifically, Figure 3.3a compares the normalized edit distance for QAlign and Minimap2. The normalized edit distance is the edit distance between the entire read and the aligned section on the genome normalized with the length of the read, in nucleotide domain for *both* nucleotide alignment and quantized alignment (Q2). In case of Q2, the information of the location of the alignment on the genome is leveraged from the alignment between the quantized read and the quantized genome first, and the edit distance is computed between the corresponding nucleotide read and the aligned section on the nucleotide genome (see Methods for details). Intuitively, the normalized edit distance gives a measure of how close the two sequences are. Therefore, the smaller the normalized edit distance, better is the alignment. In addition, the *normalized edit distance* for the reads that have *normalized edit distance of aligned reads* more than 0.48 is set to 1 (We noticed that the normalized edit distance between a pair of random DNA sequences is above 0.48, refer to Figure 3.23). Therefore, the figure represents only those alignments that are better than alignment of random DNA sequences.

To better visualize the results, we group alignments with different colors and marks for different conditions. The red circles in Figure 3.3a-b represent the reads that are well-aligned in both nucleotide and $Q2$ alignments and at nearly the same location on the genome. The blue cross represent well-aligned reads in both $Q2$ and nucleotide alignments but at different location on the genome or on a different chromosome. The black asterisks are the reads that are well-aligned in $Q2$ only, *i.e.*, in nucleotide alignments, the alignment of these reads are either missing or does not satisfy the definition of the well-aligned reads. The green diamonds are the reads that are well-aligned in nucleotide alignments only. The pink square points are the reads that are not well-aligned in both $Q2$ and nucleotide alignments. For each read, there could be multiple alignments on the genome because of the repeats in the genome, but we consider the alignment that has the minimum edit distance amongst all of them for the evaluation in these plots.

Figure 3.3a shows that the normalized edit distance is overall smaller for $Q2$ alignments than nucleotide alignments. The better alignment in $Q2$ is also evident from the slope of the regression line in Figure 3.3a. It shows that on average $Q2$ alignments has 18.19% improvement in terms of the normalized edit distance than the nucleotide alignments.

The results for another fine-grained metric are shown in Figure 3.3b, which compares the normalized alignment length on genome in $Q2$ to the normalized alignment length on genome in nucleotide alignments. The normalized alignment length is the ratio of the length of the section on genome where a read aligns to the length of the read. There are 10.1% reads that are well-aligned in $Q2$ only (the black asterisks), and the normalized alignment length is close to 1 in $Q2$ but it is much less than 1 in nucleotide alignments, therefore representing several non-contiguous alignments in nucleotide domain that are captured in $Q2$. The normalized edit distance for such reads in $Q2$ is much less than the normalized edit distance for the same reads in nucleotide alignments. Similar results are observed across different datasets as evident from the slope of the regression line for normalized edit distance comparison between $Q2$ and nucleotide alignments shown in Table 3.1.

Table 3.1: Comparison for the percentage of well-aligned reads onto genome, and slope of the regression line (for normalized edit distance comparison plot of $Q2$ vs nucleotide alignments) with randomly sampled reads for each datasets. The slope of the regression line shows the average gain in the normalized edit distance.

Dataset (No. of sampled reads)	**Method of alignment**	**Percentage well-aligned reads**	**Slope of Regression line**
K. Pneumoniae R9.4 1D (1k)	nucleotide	79.4	0.8181
	$Q2$	88.70	
E. Coli R9.4 1D (1k)	nucleotide	79.2	0.9584
	$Q2$	84.20	
E. Coli R9 2D (1k)	nucleotide	82.6	0.9627
	$Q2$	91.8	
Human R9.4 1D (50k)	nucleotide	85.70	0.9696
	$Q2$	87.95	
Simulated Human with Deep Simulator [28] (10k)	nucleotide	69.04	0.8527
	$Q2$	84.35	

Methodology and performance metrics: The ability of QAlign to align DNA reads to a reference genome is discussed here. In each experiment, we take the reads and align them to the genome both using QAlign and Minimap2. There are two different types of performance metrics: (1) Coarse performance is measured by the fraction of reads that have been *well-aligned* by the algorithm. A read is said to be well-aligned if at least 90% of the read is aligned to the genome and the *(approximate) normalized edit distance* from Minimap2 is below a threshold value (threshold values are 0.48 for nucleotide alignments, 0.25 for $Q2$, and 0.35 for $Q3$ - these values are chosen based on the empirical results shown in Figure 3.23 and 3.24) or the mapping quality is high (greater than 20). (2) Fine-grained performance is measured in our experiments by two metrics: the *normalized alignment length* and the *normalized edit distance* between the read and the genome region that the read aligns to. We note that we do not use the particular alignment returned by the different aligners since this is not directly comparable. The normalized-edit distance is defined as the ratio of the edit distance between the genomic region aligned by the read and the entire read to the length of the read - thus, normalized edit distance of 0 corresponds to perfect alignment, and an upper bound is 1 if the read does not return an alignment. The normalized edit distance is computed in the nucleotide domain for both the nucleotide and quantized alignments. The information about the start and the end location on the genome is leveraged to compute the edit distance in the nucleotide domain for the quantized alignments. We note that the normalized edit distance between a random sequence of genome and a given read of the same length is 0.48 (Refer to Figure 3.23 for the details on the normalized edit distance between random DNA sequences).

Read-to-genome alignment for the 2D reads: An alternate pipeline for the alignment of the 2D consensus reads is using the corresponding 1D reads from both the template and the complement strand and estimate the alignment location of the 2D consensus read onto the genome using the QAlign algorithm. This can be done by exploiting the nanopore physics with the 1D reads from each strands. The experiment pipeline is explained in detailed in the QAlign algorithm section of this chapter. The results in Figure 3.5 illustrates that the alignment length on the genome, which is the output of the QAlign algorithm is atleast twice the length aligned by the local alignment algorithm for the same read at no additional cost of the normalized edit distance (as shown in Figure 3.6 and 3.7).

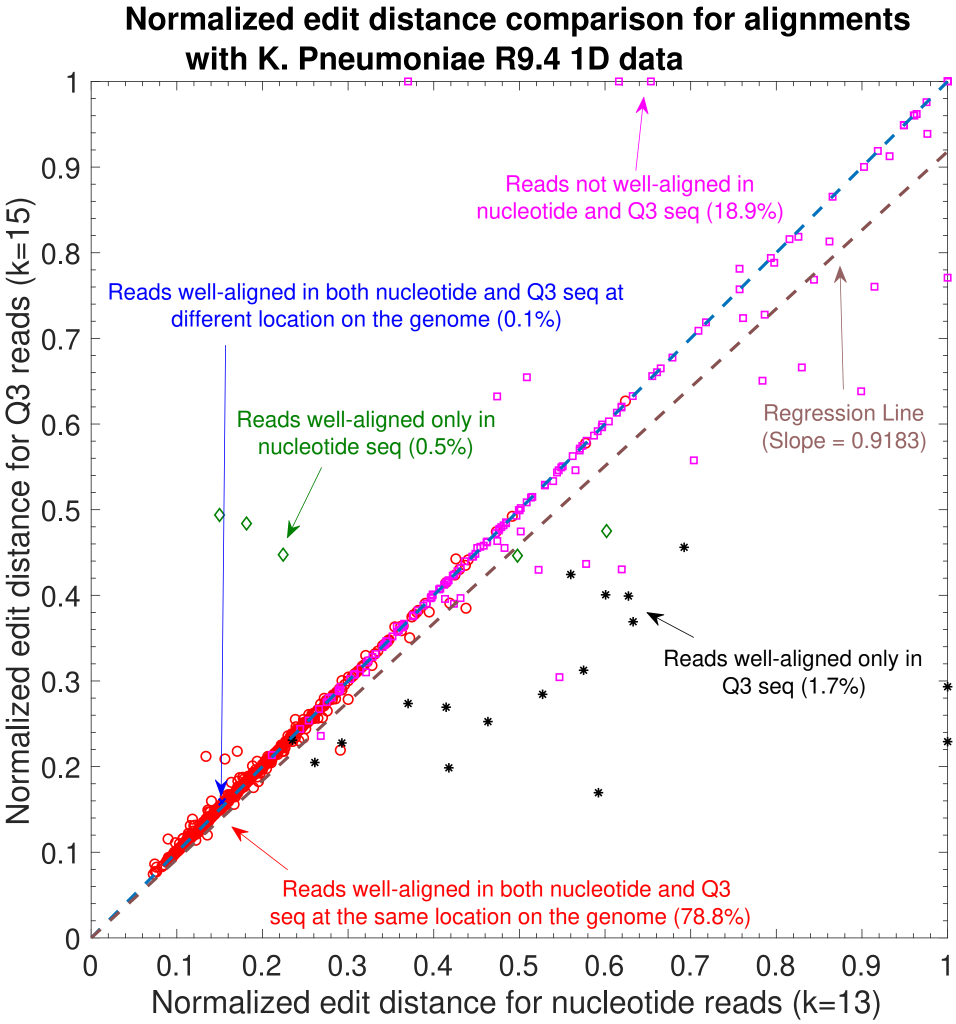


Figure 3.4: Normalized Edit distance comaprison for nucleotide vs $Q3$ for K. Pneumoniae R9.4 1D reads. The slope of regression line is 0.9183, indicating 8.17% reads have a smaller distance in Q3 than in nucleotide. 1.7% reads are well aligned to genome in $Q3$ only, whereas there are 0.5% reads that are well aligned in nucleotide domain only.

3.3.3 Read-to-Read alignment

Alignment of genomic reads to other reads is a basic primitive useful in many settings. For example, this is a first step in many overlap-layout-consensus (OLC) assemblers [12]. A key challenge in read-to-read assembly is the increased error rate that the aligner has to deal with. For example, if two reads R_1, R_2 are sampled from the same region of the genome,

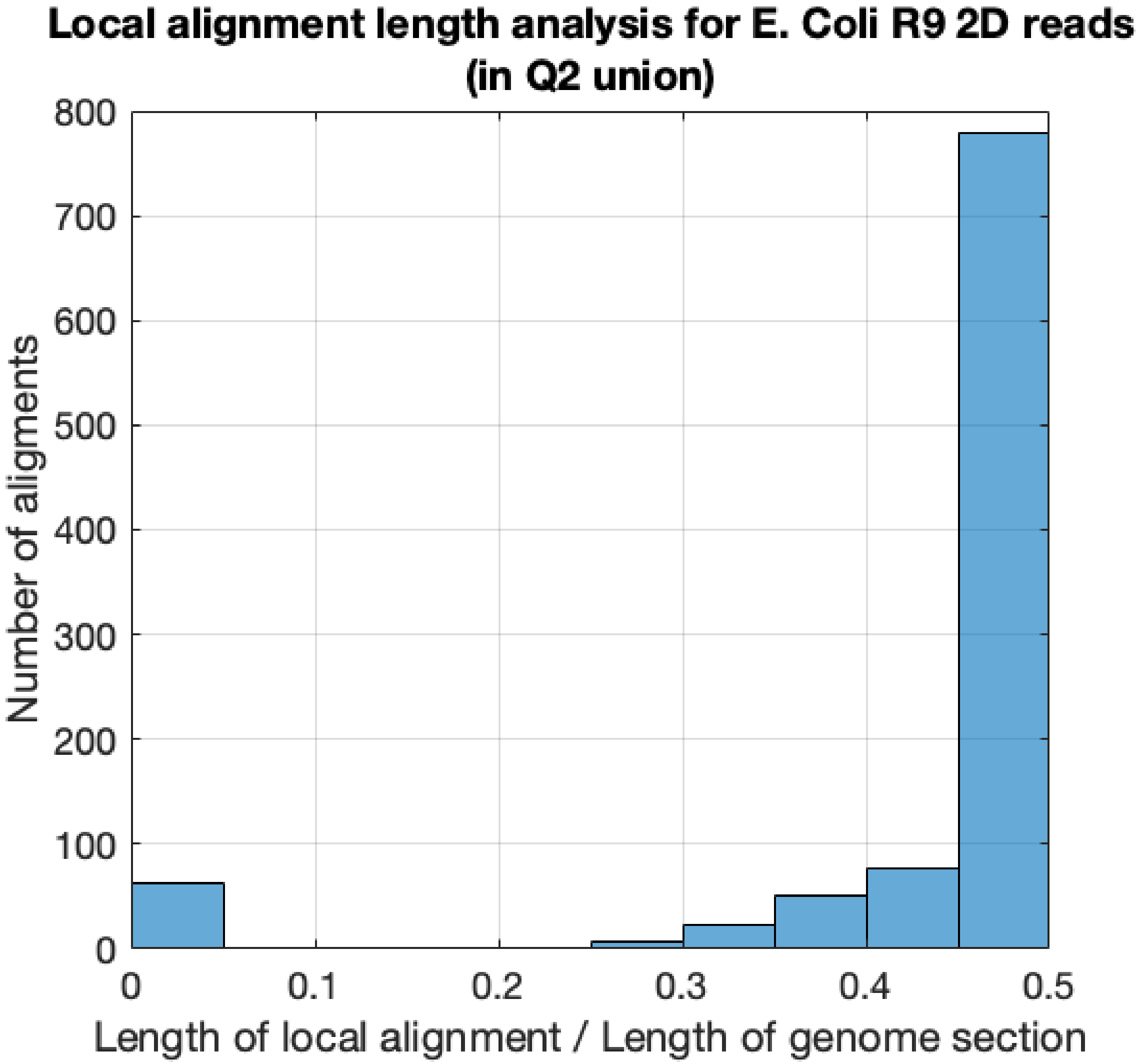
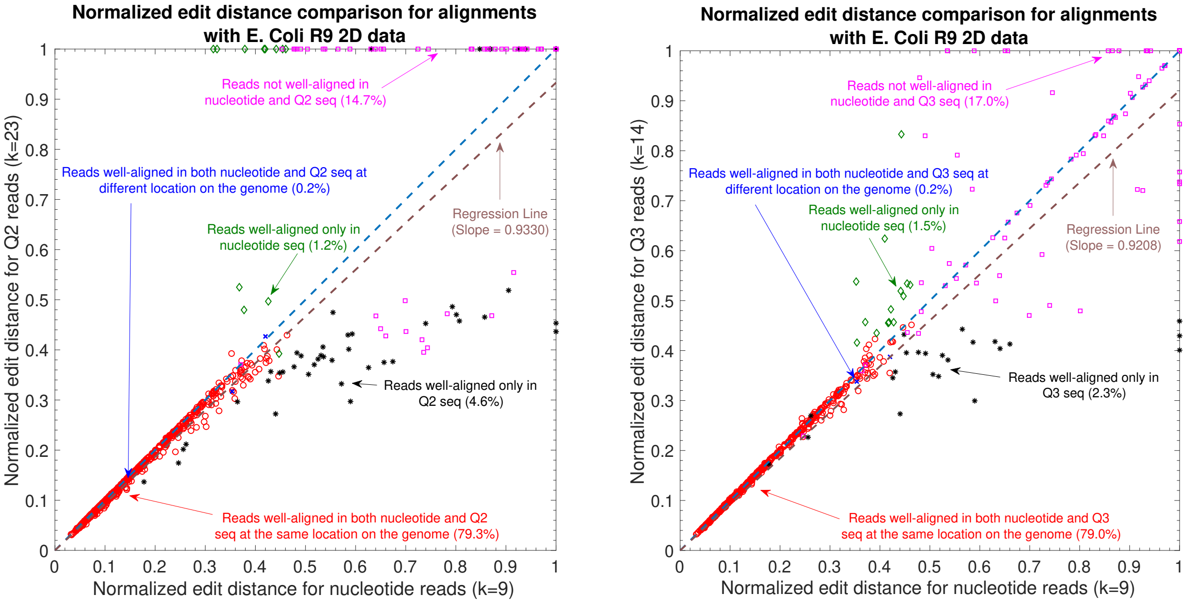


Figure 3.5: Hybrid approach for the 2D reads using the alignments from the 1D reads of each strand of the DNA. Length of the genome section is the output of QAlign using the alternate approach for 2D reads and it is the union of the region where the 1D reads from both the template and the complement strand aligns to on the genome. Length of local alignment is the region on this genome section where the 2D consensus reads aligns to (using a local alignment algorithm). Therefore, the output of QAlign algorithm is atleast twice the length of the local alignment.

each may be within 15% edit-distance of the reference genome (assuming a 15% error-rate), however, the edit distance between R_1 and R_2 can be up to 30% leading to an effective doubling of the error-rate. Long-reads hold the promise of fully-automated assembly but is



(a) (b)

Figure 3.6: The alignment of long nanopore DNA-Seq R9 2D reads onto E. Coli reference genome using the same pipeline for 2D reads, i.e., the 2D reads are translated to the quantized current sequences without any knowledge of the 1D reads. (a) The plot shows the normalized edit distance comparison for nucleotide vs $Q2$ in the 2D reads experiment setting. The slope of the regression line is 0.9330, indicating 6.7% well aligned reads have a better normalized edit distance in $Q2$ than in nucleotide. (b) The plot shows the normalized edit distance comparison for nucleotide vs $Q3$ in the 2D reads experiment setting. The slope of the regression line is 0.9208, indicating 7.92% well aligned reads have a better normalized edit distance in $Q3$ than in nucleotide.

currently feasible only when for bacterial genomes [35]. For complex mammalian genomes, long repeats fragment assembly [36] and more accurate alignment can help alleviate this problem.

The results for read-to-read alignment are illustrated in the Figure 3.14a-b, and Table 3.2. Table 3.2 summarizes the precision, recall, and average overlap quality for different methods used (namely, nucleotide, $Q2$, and $Q3$) to find the alignments between the overlapping reads across different datasets. It is evident from the table that $Q2$ provides higher recall and average overlap quality than nucleotide alignments at the cost of a bit lower precision. $Q3$, on the other hand, shows better recall and average overlap quality than nucleotide alignments

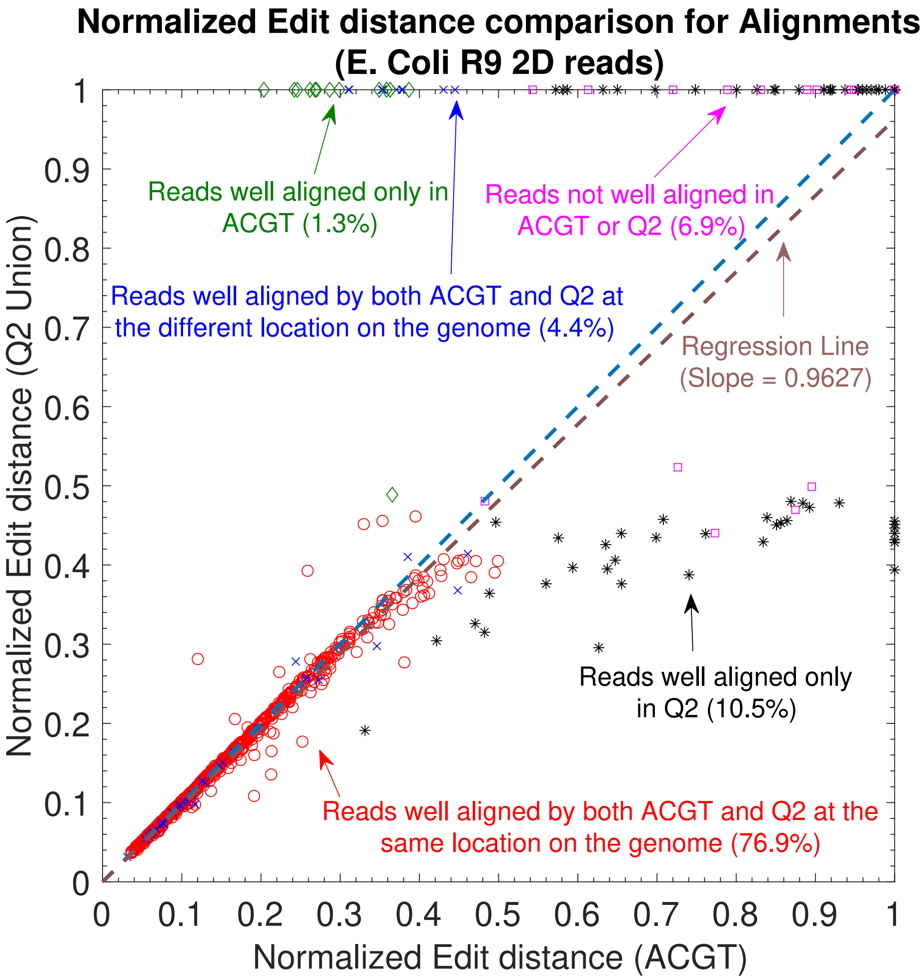
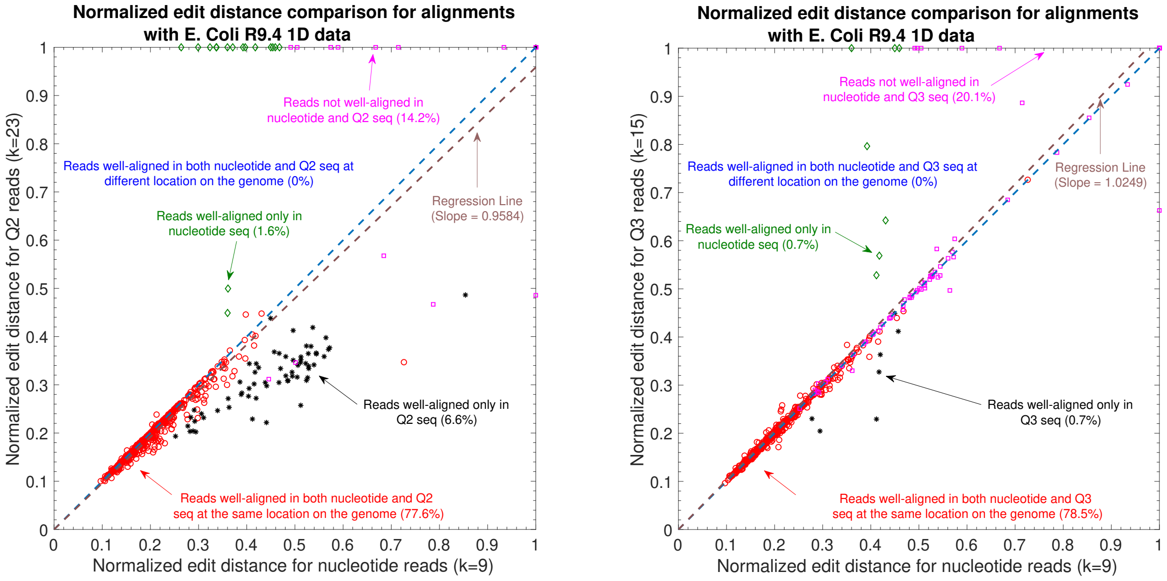


Figure 3.7: The alignment of long nanopore DNA-Seq R9 2D read onto E. Coli reference genome using the alternate approach that uses the information of the alignments of the corresponding 1D reads. The plot shows the normalized edit distance comparison for nucleotide vs $Q2$. The slope of the regression line is 0.9627, indicating 3.73% well aligned reads have a better normalized edit distance in $Q2$ than in nucleotide.

at similar precision.

For a fine-grained evaluation, Figure 3.14a shows overlap quality comparison for the quantized ($Q2$) alignments versus nucleotide alignments using the K. Pneumoniae dataset. The blue circles in the figure represent the overlaps that are aligned [5] in both QAlign and

[5] An overlap between a pair of reads is said to be aligned by the algorithm if the Mapped region by the algorithm is at least 90% of the Mapped region plus the Overhang region (refer to the Methods section for more details).



(a)

(b)

Figure 3.8: The alignment of long nanopore DNA-Seq R9.4 1D read onto E. Coli reference genome (a) Normalized edit distance comparison for nucleotide vs $Q2$ (b) Normalized edit distance comparison for nucleotide vs $Q3$.

Minimap2. The black asterisks (along the line x=0) represent the overlaps that are aligned only in $Q2$ and not aligned in nucleotide, whereas the green diamonds (along the line y=0) represent the overlaps that are aligned only in nucleotide and not aligned in $Q2$. In Figure 3.14a, the read overlaps that are aligned only in $Q2$ is 7.3%, whereas the read overlaps that are aligned only in nucleotide is 2.3%. Therefore, QAlign demonstrates a net gain of 5.0% in terms of the number of reads aligned by the algorithm. For the read overlaps that are aligned in both $Q2$ and nucleotide, 4.62% of the read overlaps have overlap quality more than 0.9 in QAlign but not in Minimap2 whereas the opposite holds true in only 1.0% of the read overlaps. Thus QAlign gives a net performance improvement of 3.62% over Minimap2. In addition to that, the slope of the regression line in the figure is 1.0089, therefore also illustrating better overlap quality with QAlign than Minimap2.

Figure 3.14b shows the fraction of reads which have overlap quality greater than x for the two aligners - the performance gain is seen to hold across a wide range of threshold

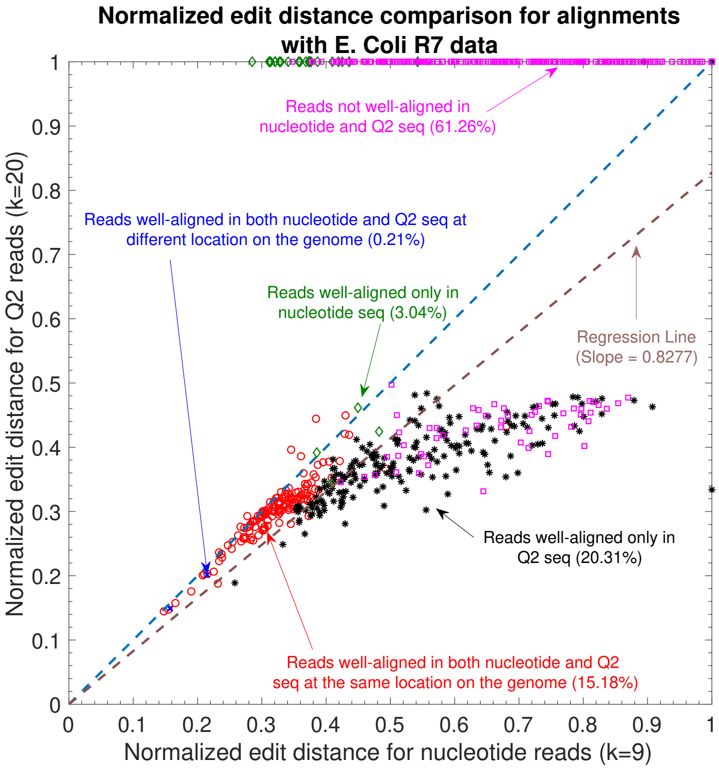
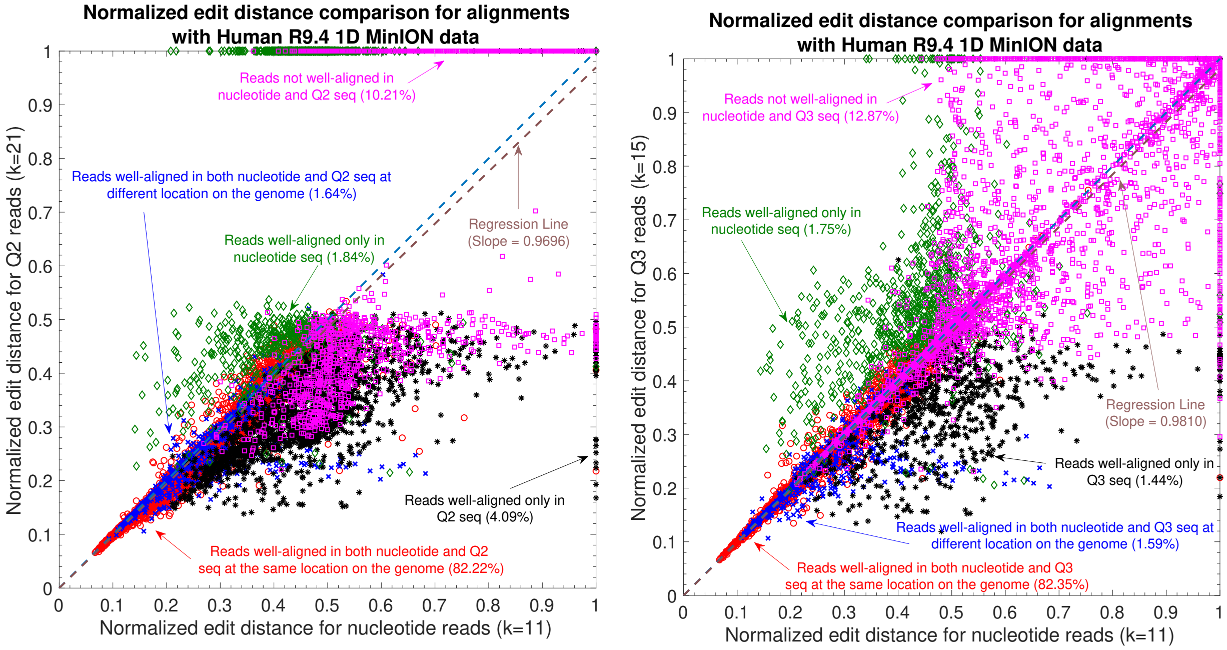


Figure 3.9: Normalized edit distance comaprison for nucleotide vs $Q2$ for E. Coli R7 1D reads. Majority of the reads are well-aligned by QAlign (35.7% of the 955 total reads are well-aligned in $Q2$, whereas only 18.43% reads are well-aligned in nucleotide). Moreover, the slope of the regression line is 0.8277 indicating an average gain of 17.23% in terms of the normalized edit distance.

values x. The area under the curve (which equals the average overlap quality) is computed for nucleotide, $Q2$, and $Q3$ alignments across all the datasets and is demonstrated in Table 3.2. The gain in the average overlap quality is observed using QAlign across all the datasets as evident from Figure 3.14b. Specifically, there is a gain of 9.2% in K. Pneumoniae dataset, when we compute it as the ratio of the average overlap quality of $Q2$ to average overlap quality of nucleotide alignments. Similarly, there is a gain of 2.5%, 10.8%, and 31.2% in the average overlap quality for the E. Coli R9.4 1D, E. Coli R9 2D dataset, and simulated human dataset, respectively.

Methodology and Performance Metrics: The ability of QAlign to align DNA reads to other DNA reads is discussed here. In each experiment, the DNA reads are aligned to other DNA



(a) (b)

Figure 3.10: (a) Normalized edit distance comparison for read-to-genome alignments with $Q2$ and nucleotide Human R9.4 1D MinION data. The slope of the regression line is 0.9696, therefore, representing more accurate alignments in $Q2$. (b) Normalized edit distance for read-to-genome alignment with $Q3$ and nucleotide Human R9.4 1D MinION data. The slope of the regression line is 0.9810, therefore, representing more accurate alignments in $Q3$ as well.

reads using both QAlign and Minimap2. The read datasets from the same organisms are used for the experiments that we have used for read-to-genome alignment. We say that the algorithm is able to align the pair of reads (*i.e.*, the pair of reads has an overlap), if the 'Mapped Region' of overlap by the algorithm is atleast 90% of the estimated overlap between the same pair of reads in the ground truth (Please refer to the algorithm section for more details). Since we do not have a ground truth specifying what pairs of reads are aligning with each other with a head-to-tail alignment, therefore, in our evaluation we have used the read-to-genome alignments to leverage this information to estimate the ground truth for the read-to-read alignment. The overlap length between a pair of overlapping reads can be determined from the location of the alignments of the reads on the genome. We define

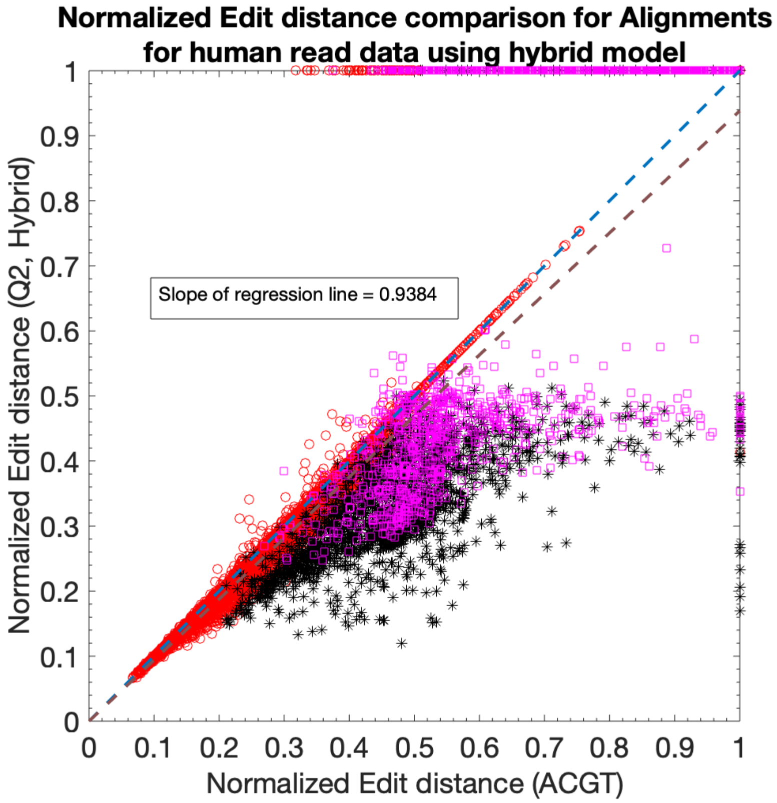
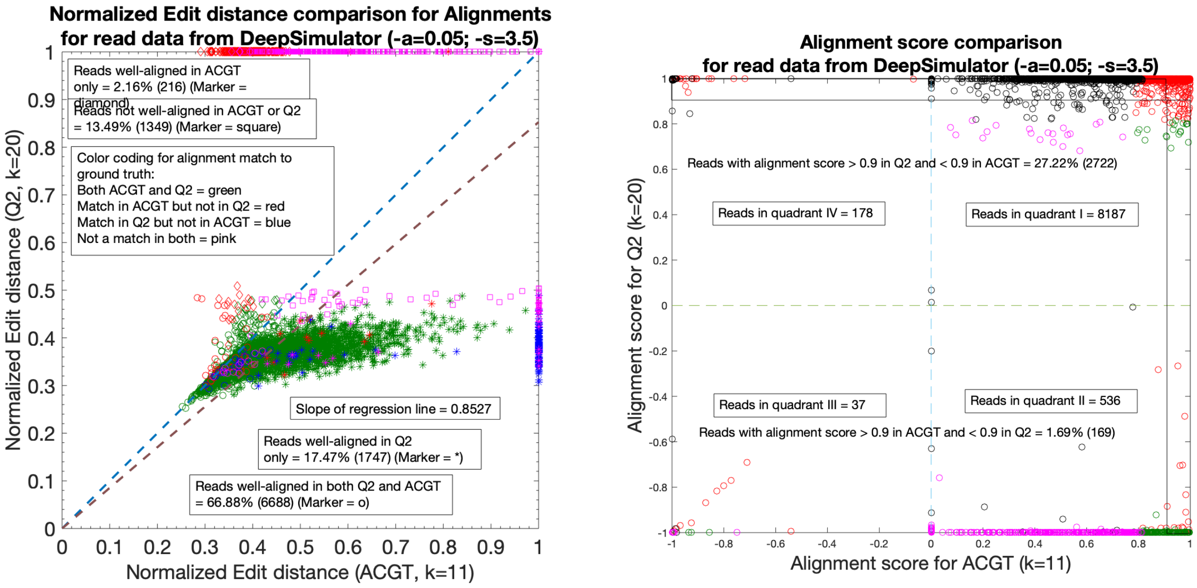


Figure 3.11: Normalized edit distance comparison for alignments using the hybrid model and the nucleotide alignments using the Human R9.4 1D MinION data. The hybrid model enhances the overall alignment accuracy by using nucleotide alignments when the $Q2$ alignments are missing and vice-versa. It is evident from the slope of the regression line as well as it decreases to 0.9384 from 0.9696.

the overlap quality as the ratio of the size of the intersection between the overlap region estimate (from the alignment algorithm) and the true overlap (from the ground truth), to the size of the union between the overlap region estimate and the true overlap. Thus, the overlap quality is equal to 1 if and only if the overlap region estimate is the same as overlap in the ground truth. The expected overlap quality is the area under the complementary cumulative distribution curve of overlap quality. This expected value is also referred to as average overlap quality. Moreover, an ensemble model chooses the best overlap between a given pair of reads using the information of the longest overlap in nucleotide and in $Q2$.

Ensemble model for read-to-read alignment: In case of the read-to-read alignment, there

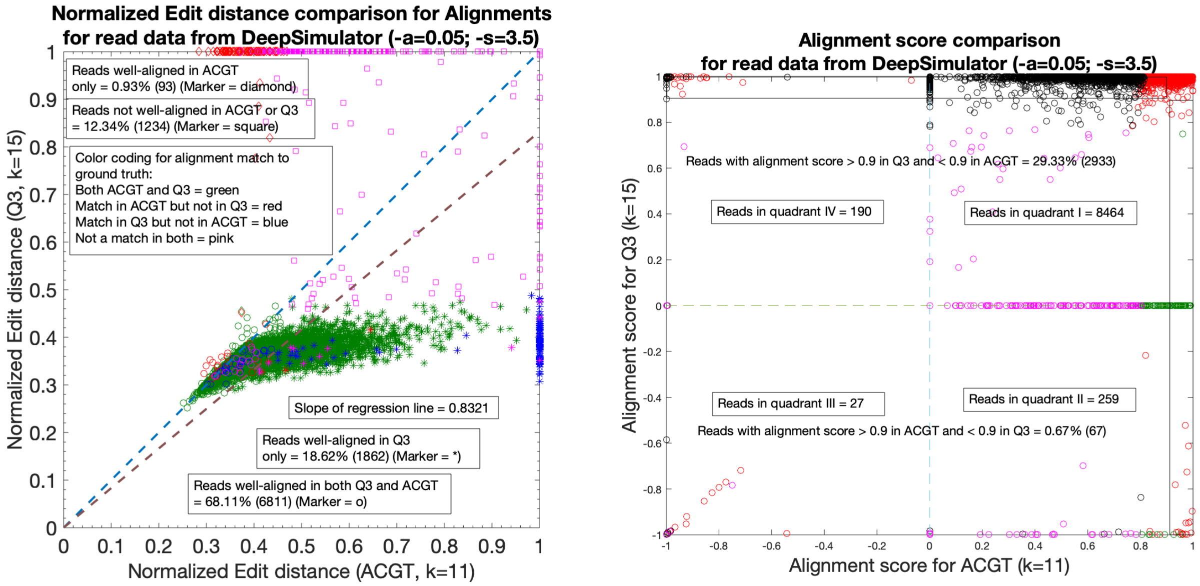


(a)

(b)

Figure 3.12: Benchmark for read-to-genome alignment using the simulated human reads from chromosome 1 of GRCh38 using Deep Simulator. (a) Normalized edit distance comparison for $Q2$ and nucleotide alignments using 10000 reads with an average read length of 12000. The Deep Simulator parameters used are corresponding to *low accuracy* reads, i.e., $a = 0.05$ and $s = 3.5$ using the context-dependent pore model. The color coding represents whether an alignment matches to the location in the ground truth - green represents both $Q2$ and nucleotide alignments matching in ground truth; red represents only nucleotide alignments matching in ground truth; blue represents only $Q2$ alignments matching in ground truth; pink represents neither of them matching the ground truth. The different markers represent if the reads are well-aligned in nucleotide or $Q2$ alignments as shown in the plot. (b) Alignment score comparison plot for $Q2$ and nucleotide alignments. For a known ground truth, we define alignment score as the ratio of the intersection of the alignment in the ground truth and the algorithm to the union of the alignment in the ground truth and the algorithm. It is computed similarly as *overlap quality* for read-to-read alignment. The negative score represents that the alignment by the algorithm has zero overlap with the alignment in the ground truth (mis-alignment).

are several read overlaps that have better overlap quality in QAlign than in nucleotide alignment and several other read overlaps that have better overlap quality in nucleotide alignment than in QAlign. Therefore, in order to increase the *sensitivity*, an ensemble model is used, and the overlap quality comparison for the ensemble model against the nucleotide read overlaps is shown in Figure 3.18.

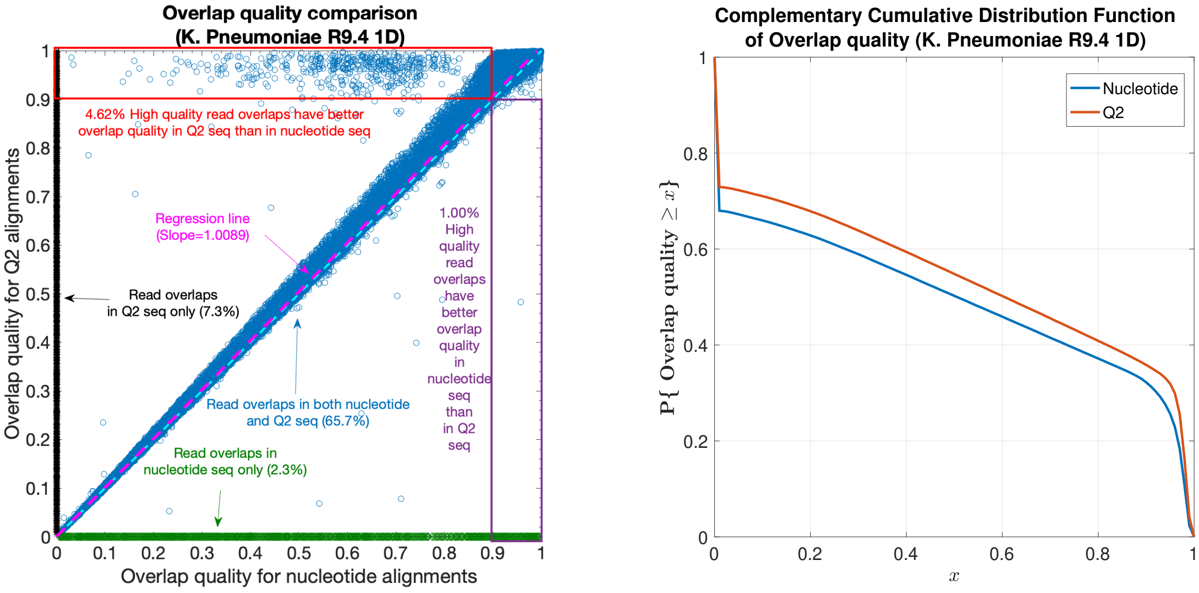


(a) (b)

Figure 3.13: Benchmark for read-to-genome alignment using the simulated human reads from chromosome 1 of GRCh38 using Deep Simulator. (a) Normalized edit distance comparison for $Q2$ and nucleotide alignments using 10000 reads with an average read length of 12000. The Deep Simulator parameters used are corresponding to *low accuracy* reads, i.e., $a = 0.05$ and $s = 3.5$ using the context-dependent pore model. The color coding represents whether an alignment matches to the location in the ground truth - green represents both $Q3$ and nucleotide alignments matching in ground truth; red represents only nucleotide alignments matching in ground truth; blue represents only $Q3$ alignments matching in ground truth; pink represents neither of them matching the ground truth. The different markers represent if the reads are well-aligned in nucleotide or $Q3$ alignments as shown in the plot. (b) Alignment score comparison plot for $Q3$ and nucleotide alignments. For a known ground truth, we define alignment score as the ratio of the intersection of the alignment in the ground truth and the algorithm to the union of the alignment in the ground truth and the algorithm. It is computed similarly as *overlap quality* for read-to-read alignment. The negative score represents that the alignment by the algorithm has zero overlap with the alignment in the ground truth (mis-alignment).

3.3.4 Read-to-transcriptome alignment

RNA-seq is a popular sequencing technology with emerging applications including single-cell RNA-seq [37]. While short high-throughput reads may suffice to assess rough gene expression estimates, isoform level analysis is better facilitated by long nanopore reads that



(a) (b)

Figure 3.14: **Nanopore long DNA read-to-read alignment.** (a) Comparison of overlap quality for K. Pneumoniae R9.4 1D reads dataset ($Q2$ vs nucleotide). Overlap quality of 1 is desirable as it represents the alignment of the algorithm matched the alignment in the ground truth exactly. Therefore, slope of the regression line > 1 represents better overlap quality of $Q2$ alignments than nucleotide alignments on average. (b) Complementary CDF of overlap quality for K. Pneumoniae R9.4 1D reads dataset. $Q2$ curve is strictly above the curve for nucleotide, therefore, demonstrating better overlap quality for $Q2$. Area under the curve gives an average overlap quality which is higher for $Q2$.

can straddle several exons simultaneously [6]. Here we perform the alignment of cDNA reads (complementary DNA reads extracted from reverse transcription of RNA) onto a reference transcriptome.

The results for read-to-transcriptome alignment are illustrated in Figure 3.20a-b, and Table 3.3. At a coarse level, QAlign improves the fraction of the well-aligned reads significantly. For the Human R9.4 1D dataset, the metric improves to 75.40% from 51.60%, and for the Mouse R9.4 2D dataset, it improves to 90.00% from 82.60%, as shown in Table 3.3.

At a fine-grained level, Figure 3.20a compares the normalized edit distance for Human R9.4 1D dataset. Note that the *normalized edit distance* is set to 1 for the reads that have

Table 3.2: Comparison for precision, recall and and average overlap quality for read-to-read alignment for four different datasets. Average overlap quality is computed as the area under the complementary CDF curve of overlap quality.

Dataset	**Method of alignment**	**Precision (%)**	**Recall (%)**	**Avg. Overlap quality**
K. Pneumoniae R9.4 1D	nucleotide	97.47	67.99	0.4908
	$Q2$	96.93	72.92	0.5360
	$Q3$	97.49	69.52	0.5053
E. Coli R9.4 1D	nucleotide	99.20	62.27	0.4688
	$Q2$	99.06	62.60	0.4803
	$Q3$	99.23	63.87	0.4811
E. Coli R9 2D	nucleotide	98.94	59.42	0.5339
	$Q2$	96.97	65.47	0.5914
	$Q3$	98.99	62.46	0.5615
Simulated Human with Deep Simulator [28]	nucleotide	75.72	41.91	0.3888
	$Q2$	75.34	53.10	0.5100
	$Q3$	76.08	54.19	0.5174

normalized edit distance of aligned reads greater than 0.48. Therefore, the figure represents the alignments that are not "equivalent" to the alignment of random nucleotide sequences. This figure clearly demonstrates the gain of quantized alignment. Specifically, $Q2$ is able to align 27.00% more reads with 8.75% better quality than nucleotide alignments (from the slope of the regression line; a similar trend of slope of regression line using Mouse R9.4 2D dataset is shown in Table 3.3). In Figure 3.20b, the lengths of aligned chunks are compared between nucleotide and $Q2$ domain. Most of the reads gets larger aligned chunks using $Q2$ quantization. Moreover, we observe a similar trend in the alignment using the Mouse R9.4 2D dataset as shown by the slope of the regression line in Table 3.3.

Methodology and Performance Metrics: The methodology and the performance metrics are similar to the DNA read-to-genome alignment experiments: we compare the fraction of well-aligned reads as well as the normalized alignment length and the normalized edit distance for each alignment algorithm.

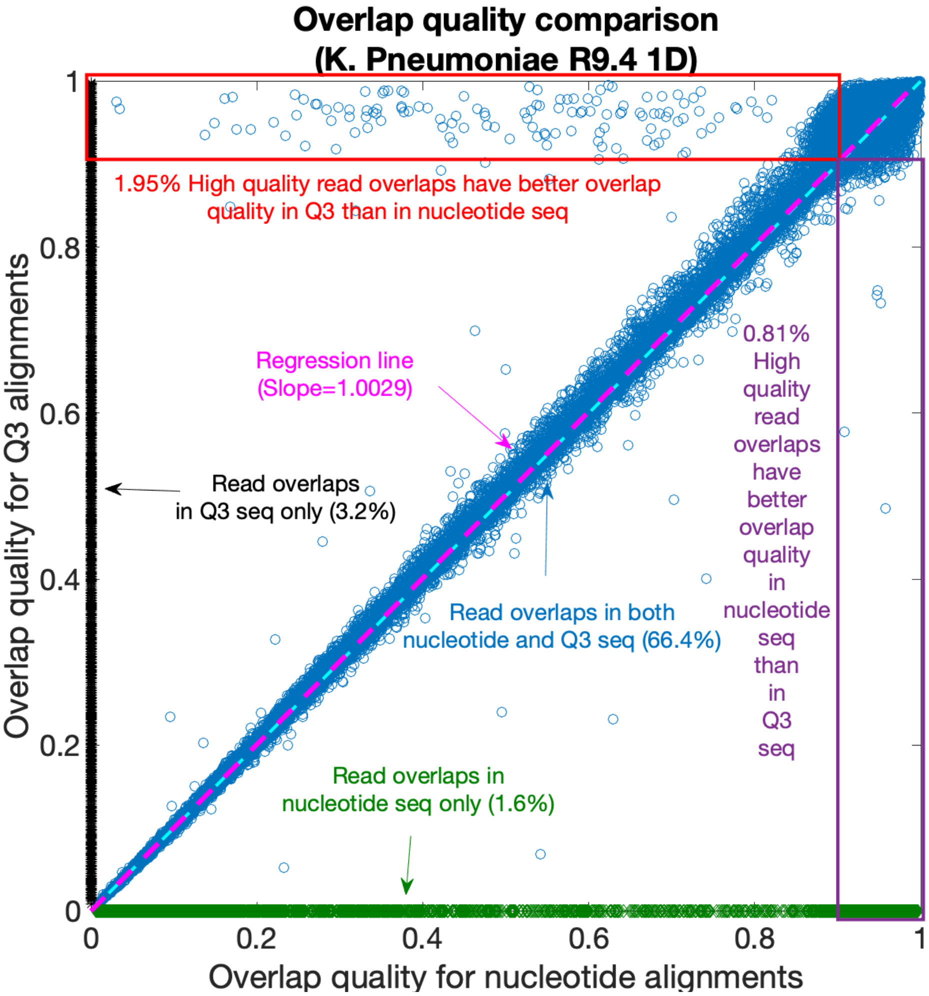
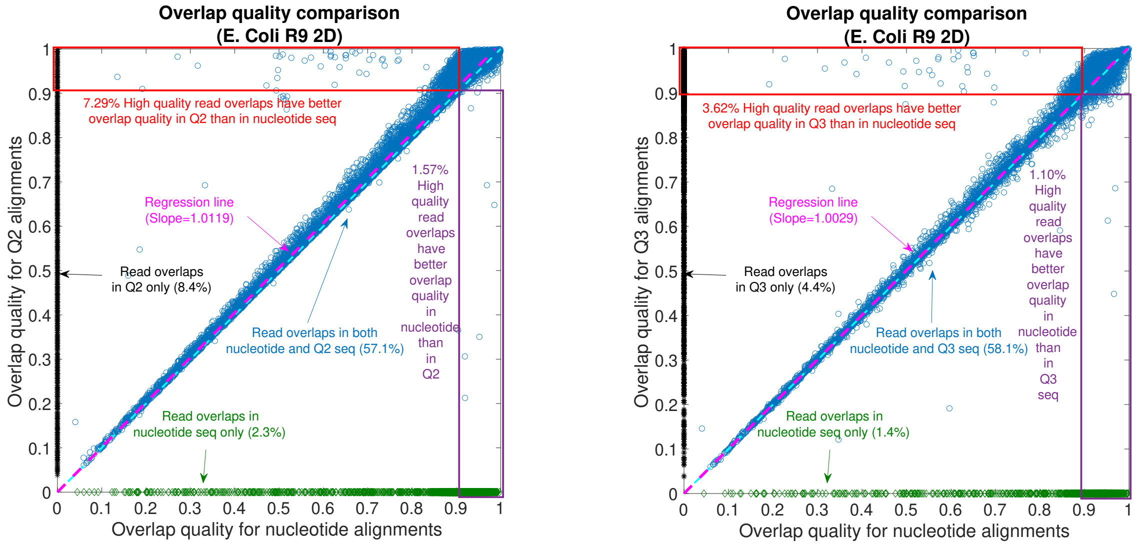


Figure 3.15: The read-to-read alignment of long nanorepore DNA-Seq reads (Overlap quality comaprison for nucleotide vs $Q3$ for K. Pneumoniae R9.4 1D dataset).

3.4 Conclusion

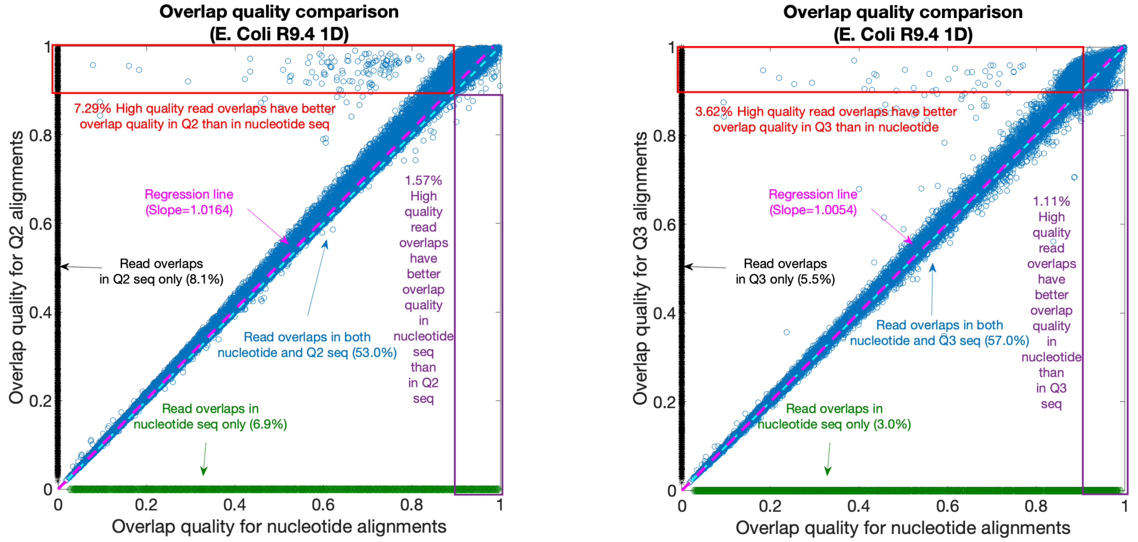
QAlign is a pre-processor that can be used with any long read aligner for a nanopore sequencer. It can be used for aligning reads onto genome or as a long-read overlapper or for aligning RNA-seq reads onto transcriptome. QAlign provides alignments that outperforms other aligners that uses nucleotide sequences in terms of the accuracy of the alignment at the cost of a similar computation time.

The reason for this performance improvement is because it takes into account the un-



(a) (b)

Figure 3.16: The read-to-read alignment of long nanorepore DNA-Seq reads of E. Coli sequenced from MinION R9 2D flow cell. (a) Overlap quality comaprison for nucleotide vs $Q2$ alignments (b) Overlap quality comparison for nucleotide vs $Q3$ alignments.



(a) (b)

Figure 3.17: The read-to-read alignment of long nanorepore DNA-Seq reads of E. Coli sequenced from MinION R9.4 1D flow cell. (a) Overlap quality comparison for nucleotide vs $Q2$ alignments (b) Overlap quality comparison for nucleotide vs $Q3$ alignments.

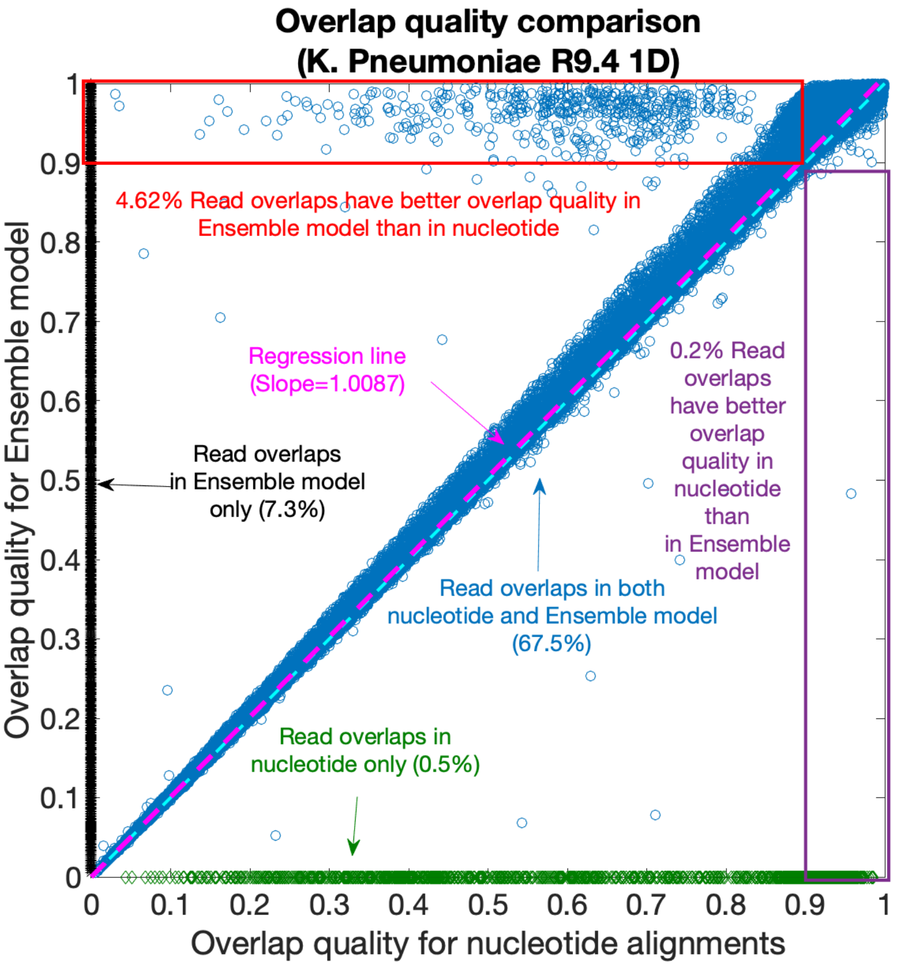
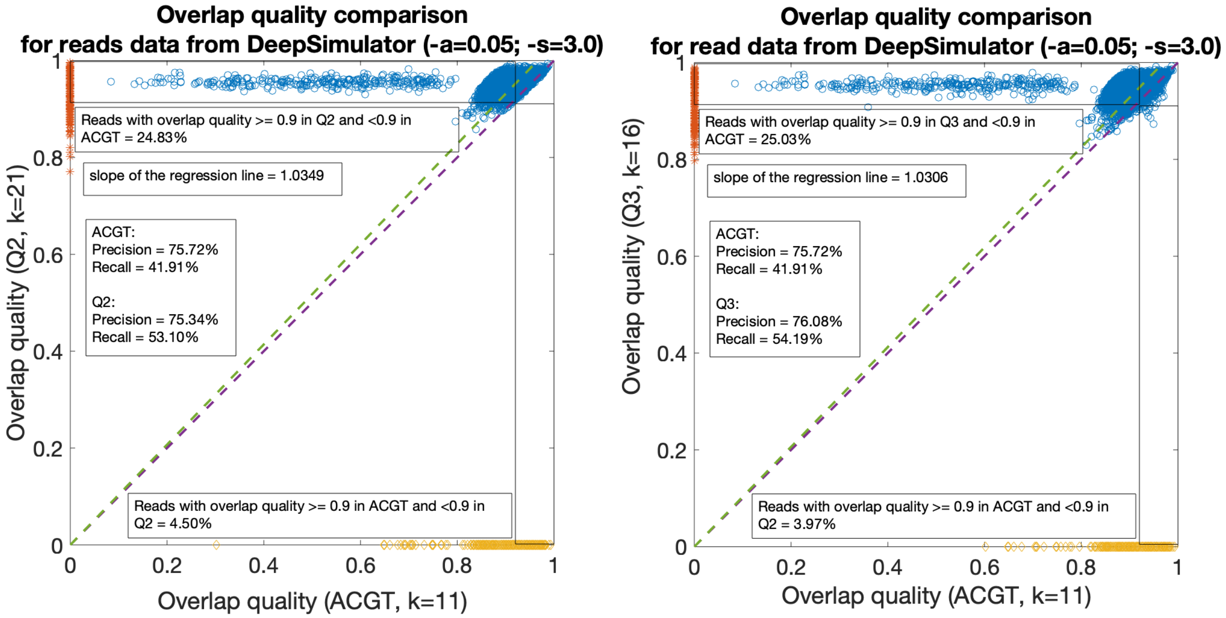


Figure 3.18: Ensemble model for read-to-read alignment for K. Pneumoniae R9.4 1D reads. We increase the *sensitivity* by capturing the alignments that are detected by either QAlign and nucleotide reads. In case of the common overlaps that are detected by both the algorithms, the one that has a longer overlap is chosen for the evaluation in this plot.

derlying physics of the nanopore sequencer through its Q-mer mapping, which could be the pre-dominant cause of the error behavior in nanopore sequencing. We demonstrated how the structure of the Q-mer map can be used even with only nucleotide read outputs, and without access to the current-level output of the sequencer. In particular, QAlign converts the nucleotide reads to quantized current levels (of finite alphabet size) which are then aligned using any state-of-the-art aligner. This improvement in the alignment of the long nanopore reads can be useful in several downstream applications such as structural variant calling,



(a) (b)

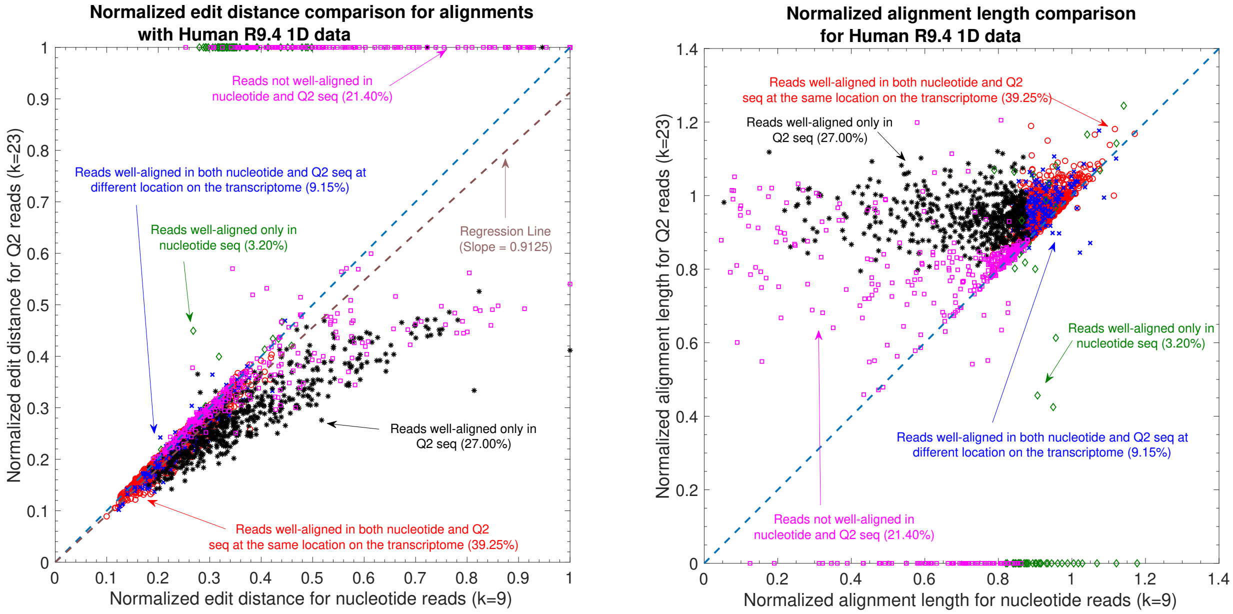
Figure 3.19: (a) Overlap quality comparison for $Q2$ and nucleotide read-to-read alignment using simulated Human reads from Deep Simulator. (b) Overlap quality comparison for $Q3$ and nucleotide read-to-read alignment using simulated Human reads from Deep Simulator.

Table 3.3: Comparison for the percentage of well-aligned reads onto transcriptome, and slope of the regression line (for normalized edit distance comparison plot for $Q2$ vs nucleotide) for two different dataset for randomly sampled reads for each dataset.

Dataset (No. of sampled reads)	**Method of alignment**	**Percentage well-aligned reads**	**Slope of the Regression line**
Human R9.4 1D (2k)	nucleotide	51.60	0.9125
	$Q2$	75.40	
Mouse R9.4 2D (2k)	nucleotide	82.60	0.8455
	$Q2$	90.00	

assembly - where the QAlign can benefit in the discovery of SVs and read overlaps that are difficult to capture because of the high error rate of nanopore reads.

The current limitation of QAlign is that it works well when we have long contiguity in the alignments. Therefore it does not perform as well in doing the spliced alignments of the



(a) (b)

Figure 3.20: **Nanopore long RNA read to transcriptome alignment.** (a) Comparison of normalized edit distance for Human R9.4 1D dataset. A small *normalized edit distance* is desirable as it represents better alignment. The slope of the regression line is < 1, therefore, representing better alignments with $Q2$ than nucleotide alignments for same reads. (b) Comparison of normalized alignment length of the aligned sections on the transcriptome for Human R9.4 1D dataset. Normalized alignment length of 1 is desirable as it represents that entire read is aligned. Majority of the reads are above $y = x$ line, representing longer alignment length in $Q2$ than nucleotide alignment.

RNA-seq reads onto genome while maintaining a similar computation time cost (as shown using empirical results in Figure 3.25). Part of ongoing extensions is to build a deep hybrid aligner which brings together the advantages of the nucleotide alignments and QAlign.

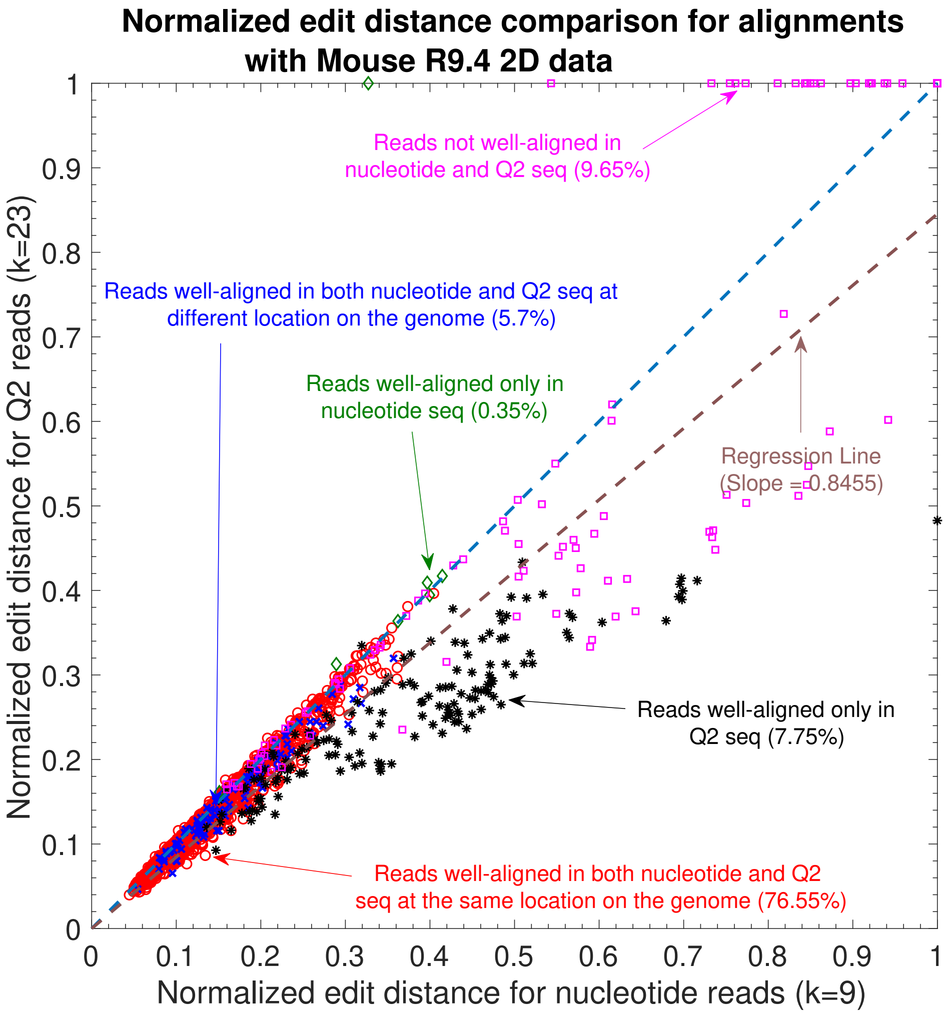


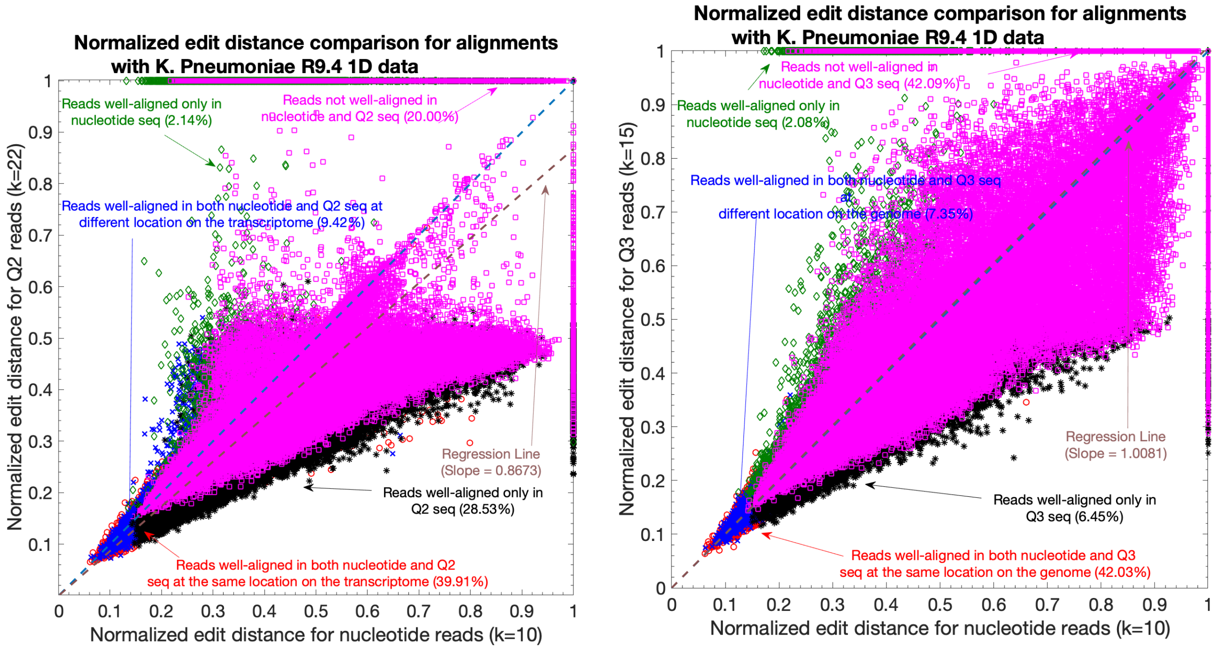
Figure 3.21: Normalized edit distance comparison for nucleotide vs $Q2$ alignments for Mouse R9.4 2D reads.

3.5 Appendix

3.5.1 Tool commands for Read-to-Genome Alignment

The Minimap2 command for the read to reference alignment with nucleotide sequence is:

```
minimap2 -c -k 9 reference.fasta reads.fasta
```



(a)

(b)

Figure 3.22: (a) Normalized edit distance comparison for $Q2$ and nucleotide read-to-transcriptome alignment using 1 Million Human cDNA reads data. (b) Normalized edit distance comparison for $Q3$ and nucleotide read-to-transcriptome alignment using 1 Million Human cDNA reads data.

The Minimap2 command for the alignment with the quantized sequences is:

```
minimap2 -c -k 23 --for-only reference_q2.fasta reads_q2.fasta

minimap2 -c -k 23 --for-only reference_q2.fasta rc_reads_q2.fasta
```

The two separate commands for the alignment with the quantized sequences are for the alignment of the quantized *template* strand of the reads to the quantized genome and the alignment of the quantized *reverse complement* strand of the reads to the quantized reference genome, respectively. We can then aggregate the results from both the outputs to determine the best alignment for each read.

3.5.2 Tool commands for Read-to-Read alignment

The Minimap2 command used for the alignment with nucleotide sequences is:

```
minimap2 -cx ava-ont -k 10 reads.fasta reads.fasta
```

The Minimap2 command used for the alignments with the quantized sequences is:

```
minimap2 -cx ava-ont -k 20 --for-only reads_q2.fasta reads_q2.fasta

minimap2 -cx ava-ont -k 20 --for-only reads_q2.fasta rc_reads_q2.fasta
```

The two separate commands used for the alignment with the quantized sequence also aims to *explicitly* account for the *template* and *reverse complement* strands of the reads, and the best alignment between a pair of reads is determined by the maximum overlap length.

3.5.3 Normalized Edit Distance between Random DNA Sequences

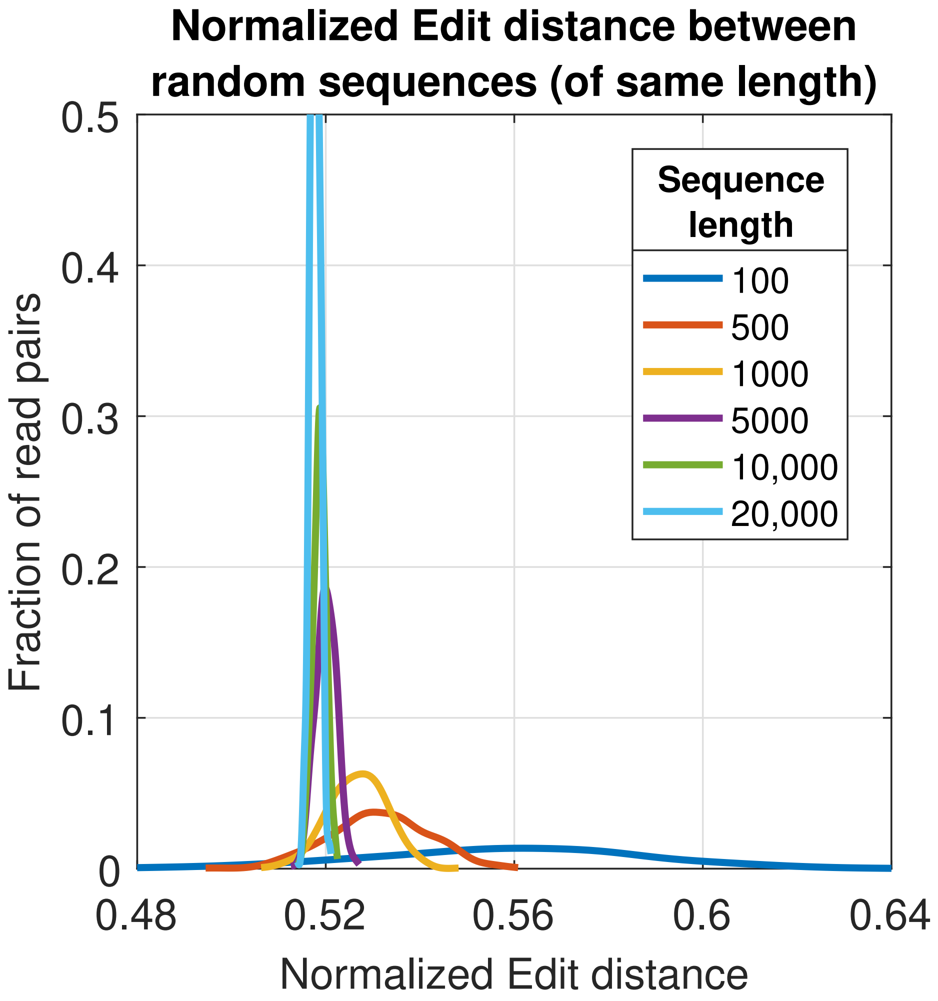
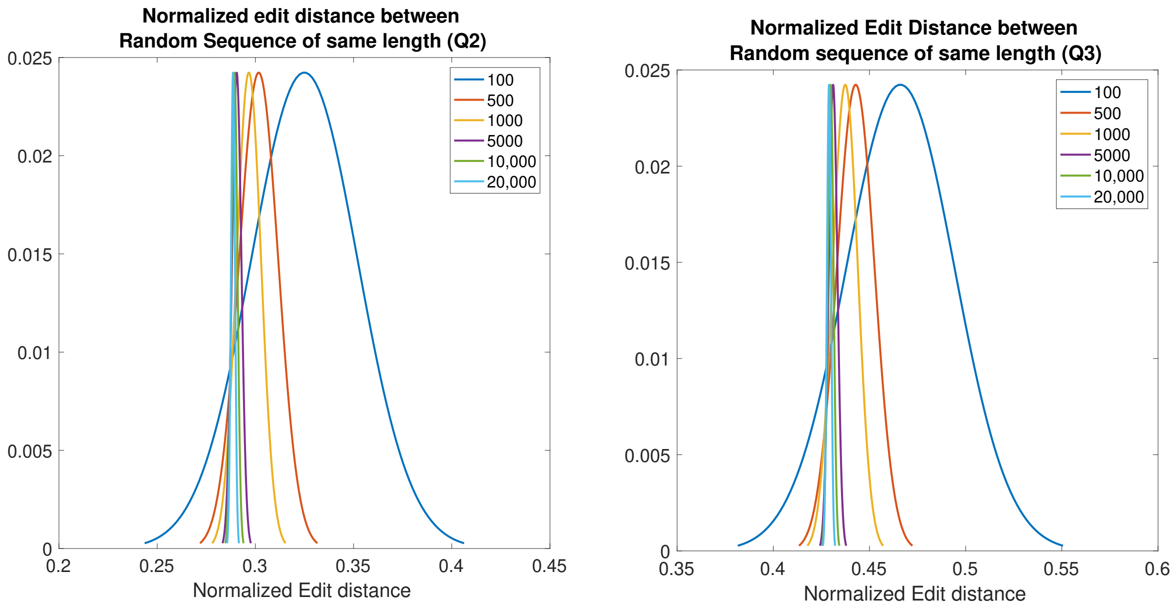


Figure 3.23: Empirical distribution for the normalized edit distance between pair of random DNA sequences. A random sequence has its symbols drawn i.i.d. from an alphabet e.g. $\Sigma = \{A, C, G, T\}$. We are interested in determining the minimum value of the normalized edit distance between two random sequences of same length, which can be used as a threshold to distinguish the alignment between a pair of random DNA sequences to pair of sequences that actually aligns to each other. We observe that the minimum value is 0.48, which we have used to comment if an alignment between a pair of DNA sequences is better (if the normalized edit distance is less than 0.48) than the alignment between a pair of random DNA sequences.



(a)

(b)

Figure 3.24: (a) Empirical distribution for the normalized edit distance between pair of random binary sequences. A random binary sequence has its symbols drawn i.i.d. from an alphabet e.g. $\Sigma = \{0, 1\}$. We are interested in determining the minimum value of the normalized edit distance between two random binary sequences of same length, which can be used as a threshold to differentiate the alignment between a pair of random binary sequences. We observe that the minimum value is 0.25, which we have used to comment if an alignment between a pair of binary (or $Q2$) sequences is better (if the normalized edit distance computed by Minimap2 is less than 0.25) than the alignment between a pair of random binary sequences. (b) Empirical distribution for the normalized edit distance between pair of random ternary sequences. A random ternary sequence has its symbols drawn iid from an alphabet e.g. $\Sigma = \{0, 1, 2\}$. We observe that the minimum value is 0.35, which we have used to comment if an alignment between a pair of ternary (or $Q3$) sequences is better (if the normalized edit distance computed by Minimap2 is less than 0.35) than the alignment between a pair of random ternary sequences.

Table 3.4: This table shows the computation time trade-off with different choice of the minimizer length `k` for the read-to-genome alignment

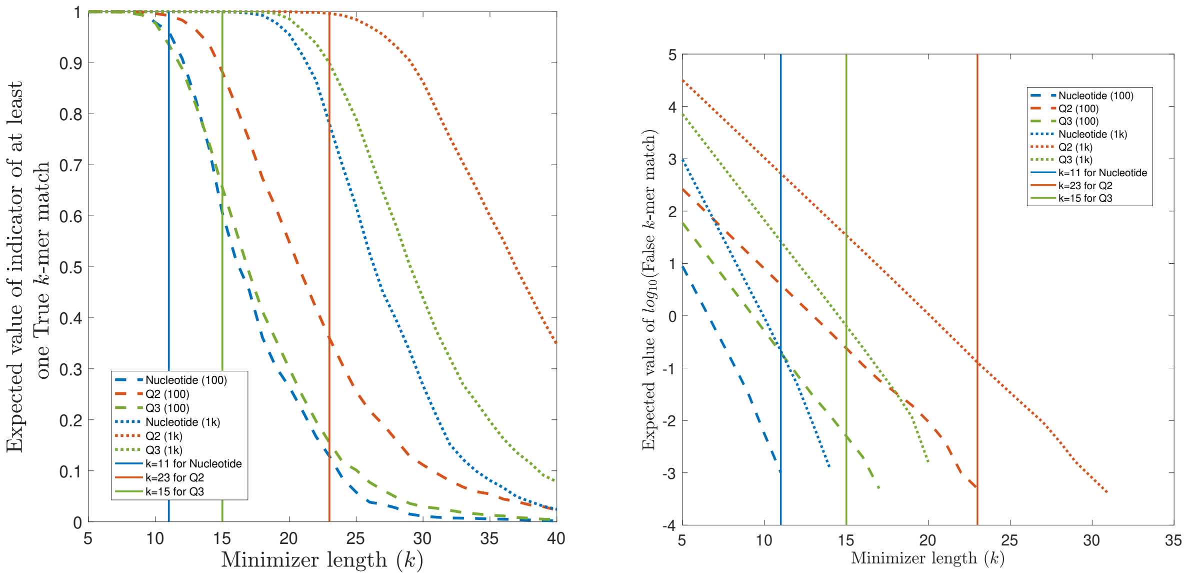
Dataset	**Method of alignment**	**Minimizer length (`k`)**	**Computation Time (in seconds)**	**Percentage of well-aligned reads**
K. Pneumoniae R9.4 1D	Nucleotide	9	164.75	76.20
		10	22.91	78.00
		11	7.97	79.00
	Q2	22	93.92	88.50
		23	43.01	88.70
		24	22.99	88.50
	Q3	14	67.54	79.20
		15	24.37	80.70
		16	12.35	81.00
E. Coli R9.4 1D	Nucleotide	9	23.02	79.20
		10	5.11	79.00
		11	3.11	78.70
	Q2	22	29.55	84.50
		23	13.2	84.20
		24	6.79	83.60
	Q3	14	17.57	79.10
		15	6.02	79.20
		16	4.42	78.90
E. Coli R9 2D	Nucleotide	9	25.17	80.70
		10	6.25	80.10
		11	3.64	79.30
	Q2	22	34.39	84.40
		23	16.65	84.10
		24	10.76	82.70
	Q3	14	23.86	81.50
		15	8.81	80.90
		16	6.01	80.10

Table 3.5: This table shows the computation time trade-off with different choice of the minimizer length `k` for the read-to-read alignment

Dataset	**Method of alignment**	**Minimizer length (`k`)**	**Computation Time (in seconds)**	**Percentage of well-aligned reads**
K. Pneumoniae R9.4 1D	Nucleotide	10	4118.63	68.07
		11	2388.37	67.00
		12	2263.17	66.00
	Q2	21	4864.79	71.98
		22	4477.58	70.92
		23	4425.29	69.73
	Q3	14	3715.9	69.60
		15	2854.07	68.73
		16	2768.73	67.77
E. Coli R9.4 1D	Nucleotide	11	882.78	62.27
		12	824.00	60.49
		13	808.74	58.72
	Q2	21	1169.15	62.60
		22	1134.86	60.82
		23	1115.41	58.87
	Q3	14	985.57	63.87
		15	914.89	62.53
		16	903.19	60.92
E. Coli R9 2D	Nucleotide	10	727.56	59.99
		11	560.08	58.78
		12	510.38	57.78
	Q2	20	1129.05	65.47
		21	1074.90	64.82
		22	1048.20	64.14
	Q3	14	705.86	62.46
		15	640.17	62.00
		16	621.43	61.44

Table 3.6: This table shows the computation time trade-off with different choice of the minimizer length `k` for the read-to-genome alignment of 50000 Human R9.4 1D reads.

Method of alignment	**Minimizer length (`k`)**	**Computation Time (in seconds)**	**Percentage of well-aligned reads**
Nucleotide	11	326,270	85.69
	12	38,446	85.11
	13	12,030	84.36
	14	4100	83.69
	15	2202	82.89
Q2	21	268,864	87.94
	22	149,614	87.45
	23	68,403	86.87
	24	50,913	86.19
	25	44,028	85.49
Q3	15	203,926	85.37
	16	63,994	85.10
	17	29,428	84.54
	18	16,182	84.07
	19	10,615	83.45



(a) (b)

Figure 3.25: **Why short alignments are difficult with QAlign?** In this experiment, a random DNA sequence s_1 of length 100 and 1000 are passed through an i.i.d. substitution channel with an error rate or 15% and the output sequence from the channel is s_2. The corresponding quantized sequences are s_1^Q and s_2^Q, respectively. A “True k-mer match” is a sub-sequence of length k that matches exactly at the same position on s_1 and s_2 (or s_1^Q and s_2^Q). A “False k-mer match is when there is an exact match of length k but at different location on s_1 and s_2 (or s_1^Q and s_2^Q). The indicator function returns 1 in case there is at least one “True k-mer match”. The expected value of the indicator function gives an estimate of the probability of finding at least one “True k-mer match”. It is evident from figure (a) that the Expected value for $Q2$ for a choice of minimizer length $k = 23$ is smaller than the expected value for Nucleotide for $k = 11$ when small sequence of length 100 is considered. Whereas the expected value for $Q2$ is nearly 1 for same choice of $k = 23$ when sequence of length 1000 is considered. On the other hand, choosing a smaller k for $Q2$ (for example, $k = 11$ for $Q2$ which gives the expected value for True k-mer match to be nearly 1) leads to more “False k-mer matches”, and hence, it requires more computation time as compared to Nucleotide alignments. Therefore, the alignments of small chunks of reads is difficult in QAlign, which is likely the case in spliced-alignments of RNA reads to genome or read-to-read alignment when the overlaps is as small as a few hundreds of bases.

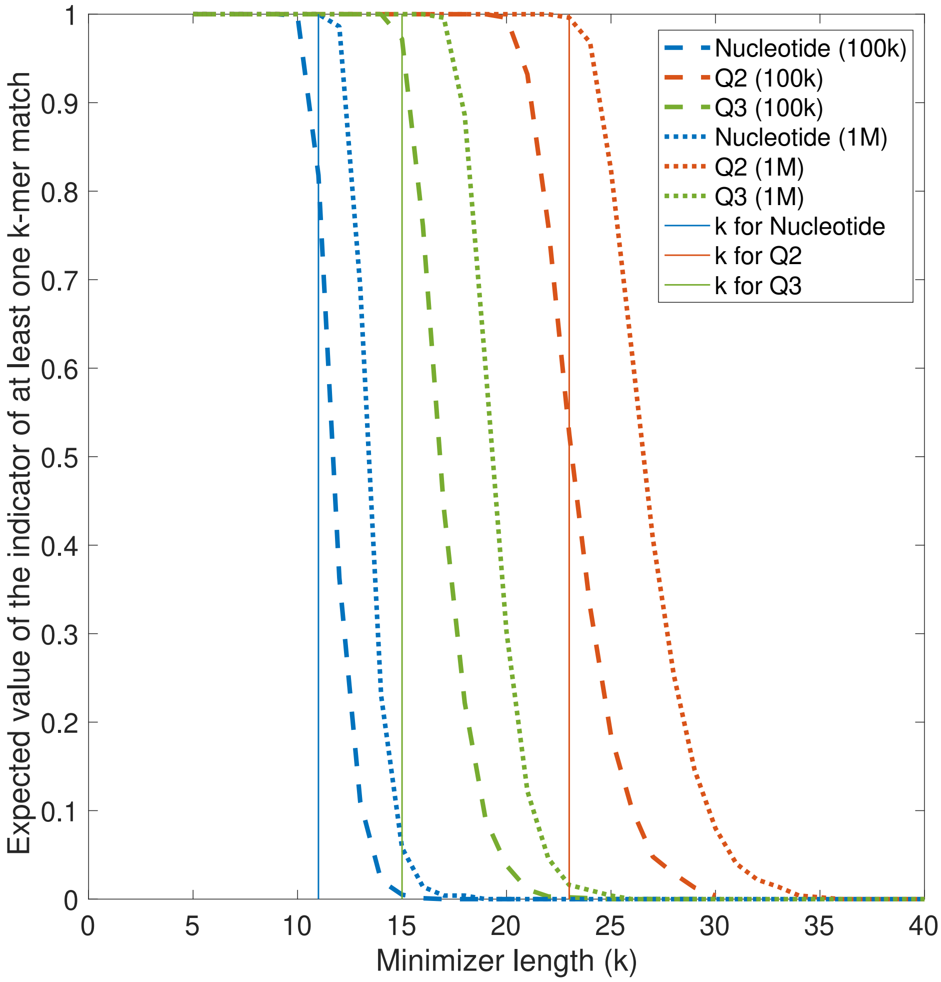


Figure 3.26: The plot shows the expected value of the indicator of at least one k-mer match when all the k-mers from a random DNA Genome (of length 100000 and 1000000) are matched against all the k-mers from a random DNA read (of length 1000). The expected value also represent the probability of at least one false k-mer match since both the genome and the read sequence are randomly generated. It shows that as the length of the genome increases the probability of a random k-mer match also increases for the same value of k.

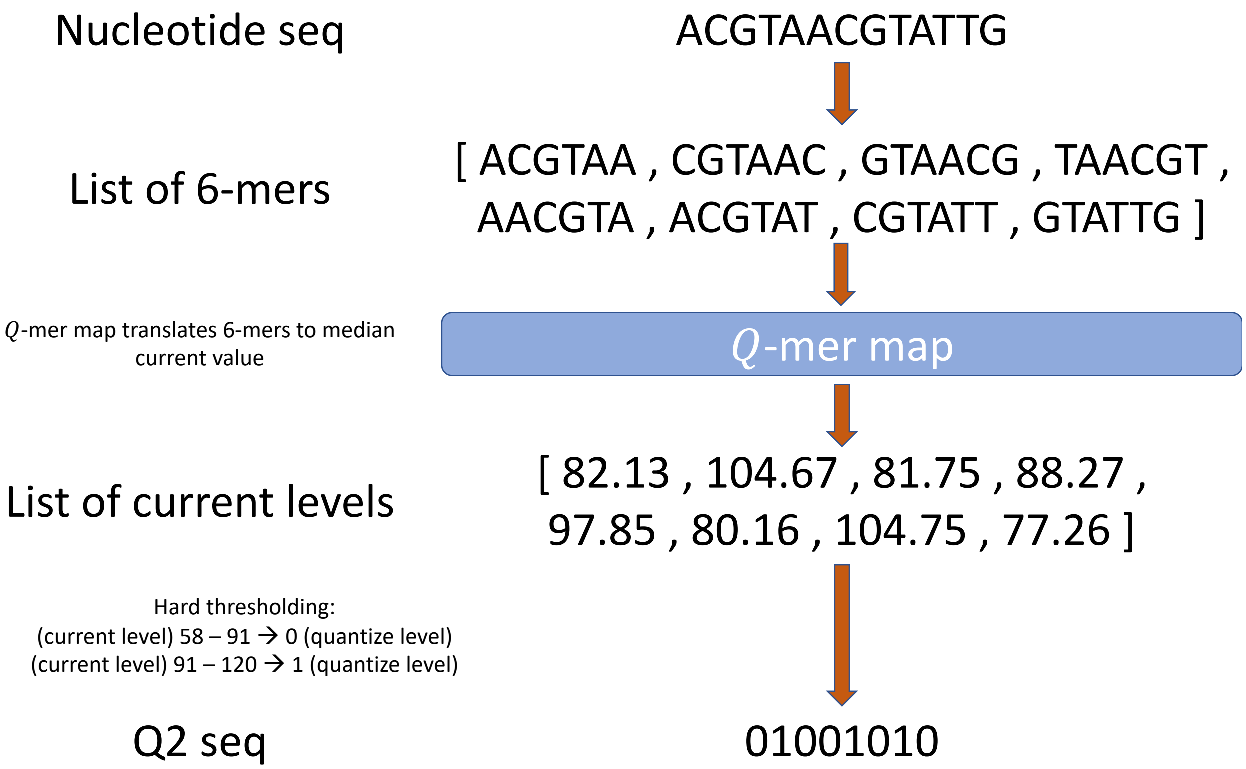
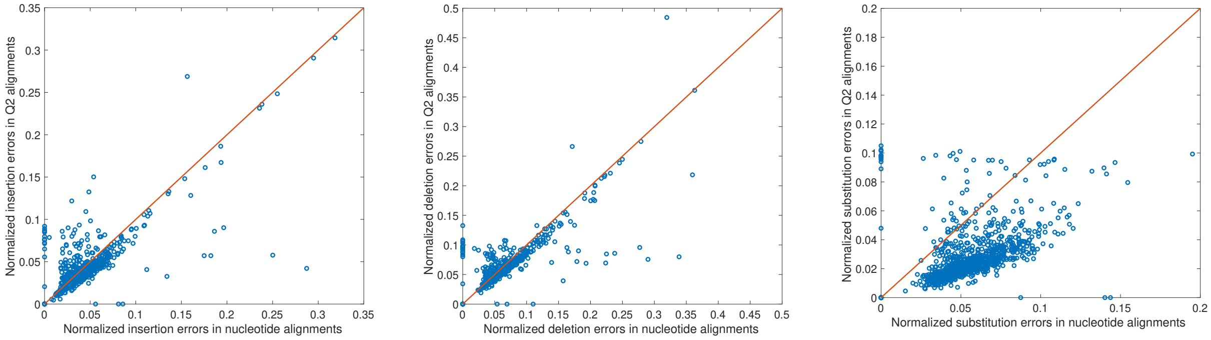
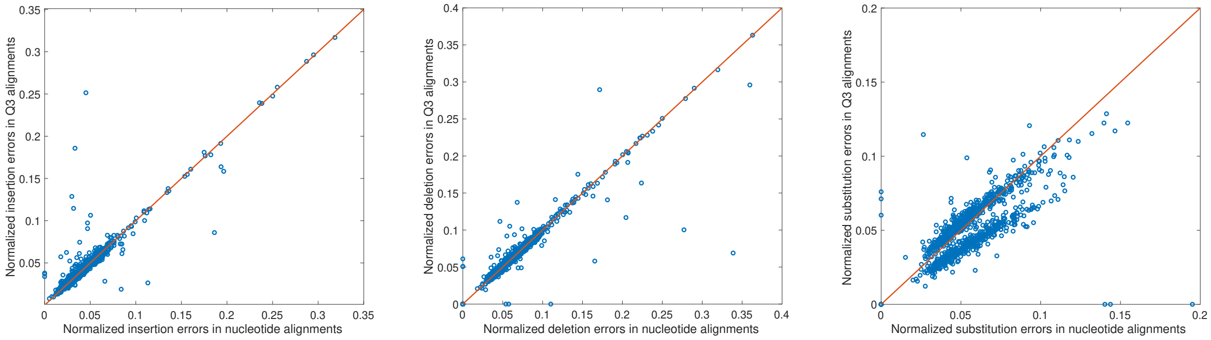

Nucleotide seq
ACGTAACGTATTG
List of 6-mers
[ACGTAA , CGTAAC , GTAACG , TAACGT , AACGTA , ACGTAT , CGTATT , GTATTG]
Q-mer map translates 6-mers to median current value
Q-mer map
List of current levels
[82.13 , 104.67 , 81.75 , 88.27 , 97.85 , 80.16 , 104.75 , 77.26]
Hard thresholding:
(current level) 58 – 91 → 0 (quantize level)
(current level) 91 – 120 → 1 (quantize level)
Q2 seq
01001010

Figure 3.27: An example for the Quantization method for QAlign.



(a) (b) (c)

Figure 3.28: A comparison for the insertion, deletion, and substitution errors for nucleotide and $Q2$ alignments for K. Pneumoniae R9.4 1D reads dataset. The normalized errors are defined as number of errors normalized by the length of the alignment. For example, normalized insertion errors is computed as the ratio of total number of insertions in the alignment of a read to the length of the alignment. The comparison is provided for the 1000 reads used for the read-to-genome alignment. We observe that the normalized insertion and normalized deletion errors in $Q2$ are nearly same as that of nucleotide alignments. However, the normalized substitution errors in $Q2$ alignments is much low than that of nucleotide alignments across the 1000 reads.



(a) (b) (c)

Figure 3.29: A comparison for the insertion, deletion, and substitution errors for nucleotide and $Q3$ alignments for K. Pneumoniae R9.4 1D reads dataset. The normalized errors are defined as number of errors normalized by the length of the alignment. For example, normalized insertion errors is computed as the ratio of total number of insertions in the alignment of a read to the length of the alignment. The comparison is provided for the 1000 reads used for the read-to-genome alignment. We observe that the normalized insertion and normalized deletion errors in $Q3$ are nearly same as that of nucleotide alignments. However, the normalized substitution errors in $Q3$ alignments is slightly lower than that of nucleotide alignments across the 1000 reads.

CHAPTER 4

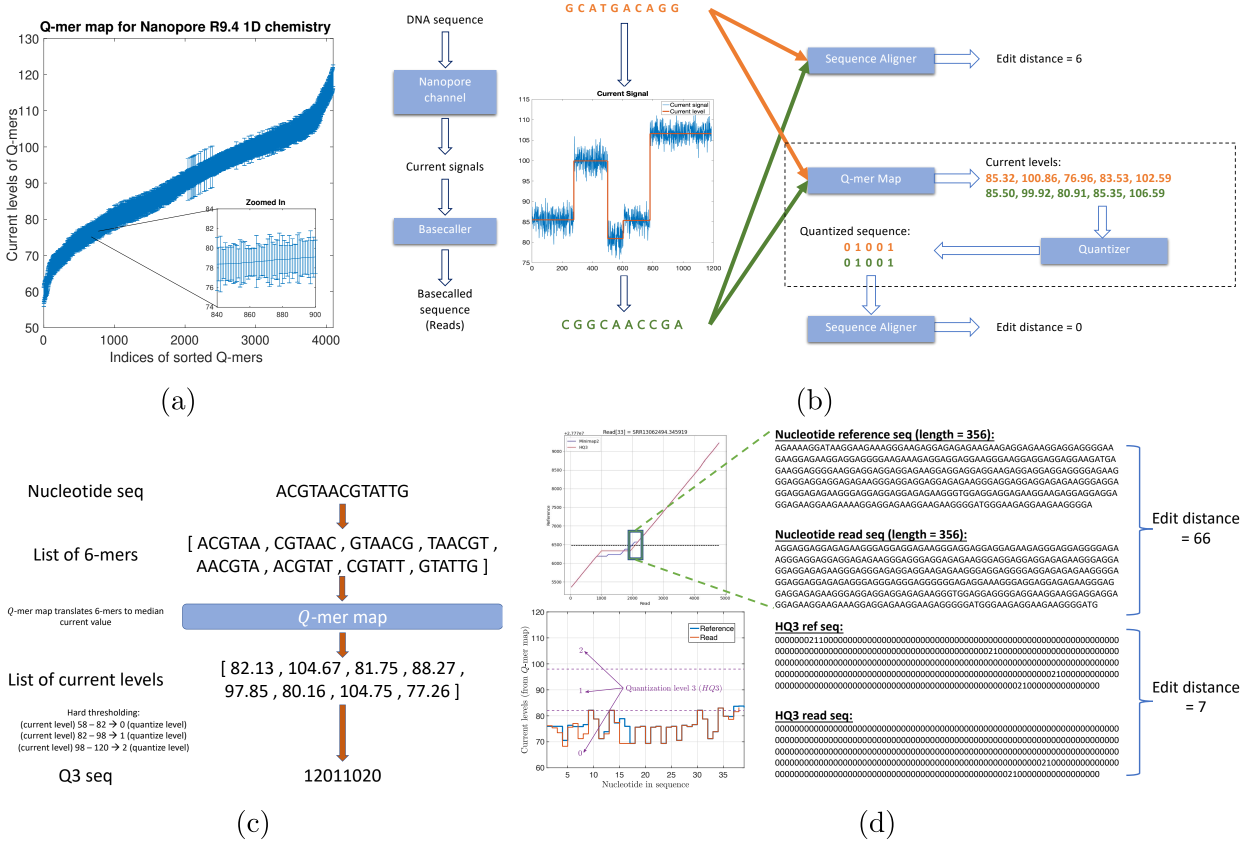
HQAlign: Aligning nanopore reads accurately for SV detection using current-level modeling

4.1 Introduction

Structural variations (SVs) are genomic alterations of size at least 50 bp long, including insertions, deletions, inversions, duplications, translocations, or a combination of these types ([38]). The study of these genetic variations has an important role in understanding human diseases, including cancer ([39]), and begins with the alignment of reads sequenced from the sample genome back to the reference genome. Accurate alignment of short reads from high throughput sequencing poses a challenge, especially, in the repetitive regions of the genome which are also the hotspots of nearly 70% of the observed structural variations ([40]).

Long-read sequencing technologies have addressed this problem by producing reads that are longer than the repeat regions, therefore, enabling the detection of variants in the repeat regions at the cost of higher error rates than short-read sequencing technologies. These high error rates in the long reads lead to non-contiguous alignment which poses a challenge in variant detection problems, especially, in the repeat regions.

Nanopore sequencing ([5, 6]) is a long read sequencing technology that provides reads (with average read length more than 10-kb and the longest read sequenced more than 4-Mb long) that can span these repetitive regions but it has a high error rate of (average) 10%. These high error rates result in low accuracy alignments ([7]) using state-of-the-art methods including minimap2 (v2.24) ([41]) which is a fast method designed for the compu-



(a)

(b)

(c)

(d)

Figure 4.1: (a) Q-mer map for Nanopore R9.4 1D flow cell (for $Q = 6$). It represents the physics of nanopore. The median current value along with the standard deviation (as error bars) are plotted for all 6-mers in the Q-mer map for R9.4 1D nanopore flow cell (the Q-mers are sorted in increasing median current levels). Note that the difference between the median current levels of any two consecutive Q-mers is very small, therefore, resulting in large overlaps. (b) An example to illustrate the error biases in nanopore base-called reads which can be resolved through the Q-mer map ability of HQAlign to perform accurate alignment despite the errors (the edit distance used here is domain-specific and is used to demonstrate the accuracy of the alignment). (c) An example of quantization method for translating the nucleotide sequences to the current level sequences using Q-mer map, and then quantizing the (real-valued) current level sequences to finite quantized sequences (*e.g.* three levels for $HQ3$). (d) An example from PromethION R9.4.1 ONT data in the neighborhood of an SV in repeat region shows the two different nucleotide sequences have similar current levels and therefore, the edit distance as observed through the lens of quantized sequence is significantly lower in $HQ3$.

tationally challenging task of long sequence alignment. This problem is further amplified in the repetitive regions such as variable-number tandem repeats (VNTR) region that accounts for a significant fraction of SVs ([42, 43]). However, these errors in nanopore sequencing have a bias induced by nanopore physics which is missed by many long-read aligners since they consider the errors as independent insertions, deletions, and substitutions. In nanopore sequencing, a DNA strand migrates through the nanopore, and an ionic current that is established in the nanopore changes according to the nucleotide sequence in or near the nanopore. However, because of the physics and non-idealities of the nanopore sequencing, each current level recorded depends on a Q-mer [1] (a set of Q consecutive nucleotide bases which influence the measurement in the nanopore) ([17, 2]). These current readings are translated back to nucleotide sequences by base-calling algorithms. Therefore, the error biases could be introduced in base-calling, especially, between different Q-mers that have similar current levels. This similarity in the current levels for different Q-mers is captured by the Q-mer map as shown in Figure 4.1a. A Q-mer map represents the median current level and the standard deviation (as error bars) for different Q-mers ($Q = 6$) for the nanopore flow cell. It is evident from this figure that there is a significant overlap between the current levels observed for different Q-mers migrating the nanopore. We propose a new alignment method, HQAlign (Hybrid-QAlign, which is based on QAlign ([1])) and is designed specifically for detecting SVs while incorporating the error biases inherent in the nanopore sequencing process.

HQAlign takes the dependence of Q-mer map into account to perform accurate alignment with modifications specifically for the discovery of SVs. Figure 4.1b gives an example where a DNA sequence (GCATGACAGG) is sequenced incorrectly as (CGGCAACCGA) due to the error bias in the nanopore sequencer. As shown in the figure, the sequences are different in the nucleotide space but they are identical in the Q-mer map space. It is important to note that *no additional soft information is used* to establish this identity such as raw nanopore current values for the nanopore reads. Instead, the nucleotide sequences that have indistinguishable current levels from the lens of the Q-mer map are mapped to a common quantized sequence. A nucleotide sequence is converted to a quantized sequence by first

[1] `https://nanoporetech.com/support/how-it-works`.

converting the nucleotide sequence to a sequence of current levels using the Q-mer map and then converting the sequence of (real-valued) current levels to a (finite level) quantized sequence by hard thresholding the current levels as shown in Figure 4.1c. Therefore, the additional information about the raw current signals is not used in the quantization process but only the Q-mer map is utilized. Further, the quantization of current levels to finite discrete levels (*e.g.* three levels) enables the use of existing software pipelines for long-read aligners such as minimap2 as the core seed and extend algorithm for the alignment of quantized sequences.

HQAlign is a hybrid mechanism with two steps of alignment. In the initial alignment step, the reads are aligned onto the genome in the nucleotide space using minimap2 to determine the region of interest where a read can possibly align to. In the hybrid step, the read is re-aligned to the region of interest on the genome (determined from the initial alignment) in the quantized space. The narrow focus on the region of interest on target for the hybrid step leads to an accurate alignment of the read-to-genome without dropping the frequently occurring seed matches from the chain in minimap2 algorithm while taking the error biases of nanopore sequencing into account through quantized sequences. The new modifications in minimap2 ([41]) address this issue with a heuristic that adds additional low occurrence minimizers to the chain if the adjacent anchors in the chain are far apart, whereas the implementation of HQAlign has a different and complementary approach by focusing on a narrow region on target to enable alignments with an improvement in chaining score over new minimap2. Moreover, HQAlign pipeline enables the detection of inversion variants unlike QAlign pipeline. In QAlign, the quantized forward read and the quantized reverse complement read are aligned separately to the quantized genome, therefore, an inversion within a read alignment is not detected in QAlign (refer to section 4.2 for details). The separate alignment of the forward and the reverse complement reads in QAlign also results in a high false discovery rate for SV detection (as explained in section 4.2.5). It is because QAlign was not optimized for downstream structural variant callers such as Sniffles2 ([45]). However, in

HQAlign, we have modified the minimap2 pipeline to enable simultaneous alignment of the quantized reverse complement read along with the quantized forward read sequence to the quantized genome. This helps not only in resolving the high false discovery rate in QAlign but also in detecting inversions within a read alignment. Further, HQAlign is optimized for the downstream structural variant callers, and is more than 2.5x faster than QAlign (as shown in Table 4.4) as the seed search domain for the alignment of quantized sequences is reduced to a region of interest determined in the initial step of the algorithm.

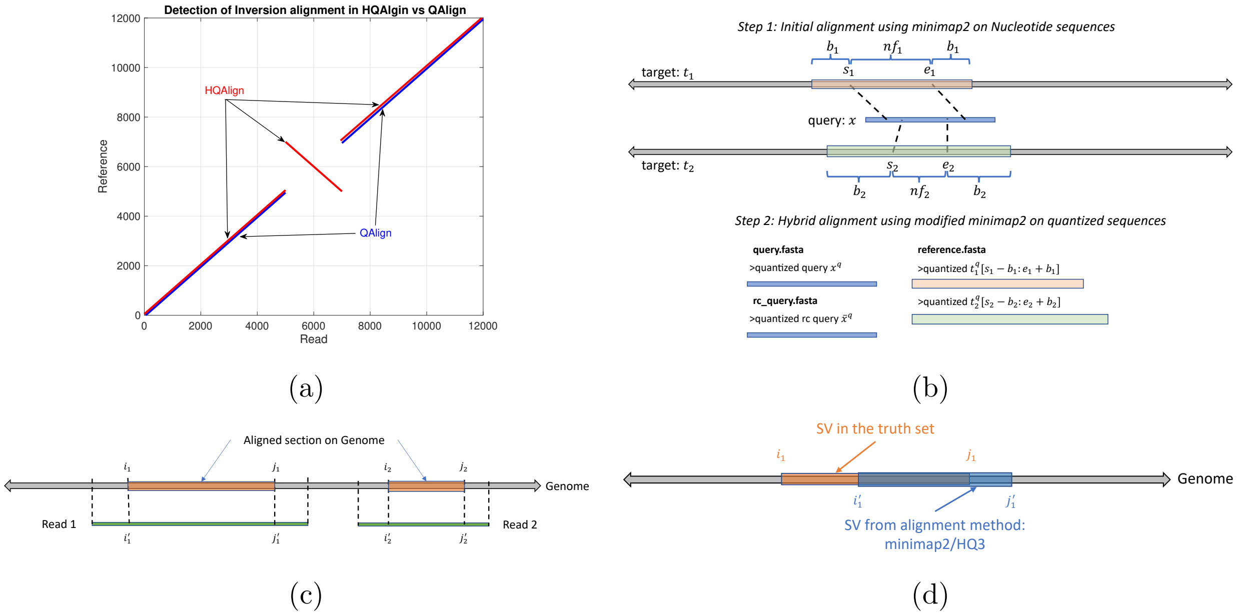
An example of the performance of HQAlign against minimap2 (v2.24) in detecting an insertion SV in a repeat region is demonstrated in Figure 4.1d. It shows alignment of a real ONT read in a repeat region (note that a pattern of a few consecutive nucleotide bases is repeated in the example) that is flanking around an insertion structural variant. Minimap2 alignment of nucleotide reference and read (both of length 356 from the region highlighted with a box) have an edit distance of 66 whereas the $HQ3$ alignment ($HQ3$ is an alignment from HQAlign pipeline where the nucleotide sequences are translated to three level quantized sequences, as shown in the example in Figure 4.1c) of quantized reference and read sequences from the same region have a significantly smaller edit distance of 7. This is because the current level sequence (by converting the nucleotide sequences using the Q-mer map in Figure 4.1a) for the reference and the read sequences is very similar. Therefore, the sequences that are far apart in nucleotide space are inherently very similar in $HQ3$ space in terms of the edit distance in the transformed space.

We show that HQAlign gives significant performance improvements in the quality of read alignment across real and simulated data. The well-aligned reads (a read is defined as well-aligned if at least 90% of the read is aligned on the genome with a mapping quality more than 20) improve to 86.65% with $HQ3$ from 83.48% with minimap2 (v2.24) for the alignment of ONT reads from HG002 sample to GRCh37 human genome. The metric improves to 89.35% from 85.64% for HG002 reads alignment to T2T CHM13 assembly ([44]), and improves to 81.57% from 81.01% for the simulated reads data. These results are presented in the results section 3.1.2 Table 4.5.

In terms of SV detection, HQAlign has F1 score at par with minimap2 (v2.24) with Sniffles2 ([45]) as the variant calling algorithm across both real and simulated datasets (Table 4.7). However, both HQAlign and minimap2 capture many complementary calls ($4-6\%$) which are missed by the other method (as shown in Figure 4.10, 4.12, 4.13, 4.14, and 4.15). For instance, the complementary HQAlign calls are SVs that are uniquely called by HQAlign or labeled missed in minimap2 due to breaking in the SV and vice-versa for the complementary calls in minimap2. Further, the analysis of common true positive SV calls in HQAlign and minimap2 against the truth set shows that HQAlign has on average a significant improvement ($10-50\%$, from the slope of the regression line in Figure 4.11, and Figures 4.16, 4.17, 4.18, and weighted average improvement across all datasets by 39%) in the breakpoint accuracy than minimap2 for the calls with a difference in breakpoint greater than 50 bp (breakpoint accuracy is determined from the difference in the start and end breakpoints of an SV with respect to the match SV in truth set, therefore, lower the difference higher is the breakpoint accuracy, refer to section 2.3 for a precise definition). Moreover, for the common true positive calls, HQAlign has (on average) better SV length similarity than minimap2 (when SV length similarity is less than 0.95, SV length similarity is a measure of how similar is the length of SV from an alignment method relative to the match SV in truth set; refer to section 2.3 for a precise definition) as shown in Figure 4.11, 4.16, 4.17, and 4.18.

4.2 HQAlign Algorithm

The HQAlign strategy consists of two steps: (1) the initial alignment of the standard base-called query sequence x to a target sequence t using Minimap2. This initial step identifies the regions of interest on the target where x aligns. (2) the hybrid step is re-aligning the query x only to the regions of interest on the genome determined in the first step in the quantized current-level space of the nanopore flow cell (refer to section 4.2.1 for more details


Detection of Inversion alignment in HQAlgin vs QAlign
Reference
Read
0
2000
4000
6000
8000
10000
12000
HQAlign
QAlign
Step 1: Initial alignment using minimap2 on Nucleotide sequences
b_1
nf_1
b_1
s_1
e_1
target: t_1
query: x
target: t_2
s_2
e_2
b_2
nf_2
b_2
Step 2: Hybrid alignment using modified minimap2 on quantized sequences
query.fasta
>quantized query x^q
rc_query.fasta
>quantized rc query $\bar{x}^q$
reference.fasta
>quantized $t_1^q[s_1 - b_1 : e_1 + b_1]$
>quantized $t_2^q[s_2 - b_2 : e_2 + b_2]$
Aligned section on Genome
i_1
j_1
i_2
j_2
Genome
Read 1
Read 2
i_1'
j_1'
i_2'
j_2'
SV in the truth set
i_1
j_1
Genome
i_1'
j_1'
SV from alignment method:
minimap2/HQ3

(a)

(b)

(c)

(d)

Figure 4.2: (a) An example to demonstrate the ability of HQAlign pipeline to align inverted sequences where QAlign fails (b) An example of HQAlign pipeline. (c) An example of read-to-genome alignment. (d) Comparison of SV in truth set to SV determined by the method: minimap2/$HQ3$.

on the quantization method and the choice of quantization level). However, HQAlign differs from QAlign method because of three key reasons: (1) In the original implementation of minimap2 ([20]), it only uses the low occurrence minimizers during read mapping which leads to misalignments, especially, in the highly repititive regions. The new modifications in minimap2 ([41]) address this problem with a heuristic that adds additional lowest occurrence minimizers to the chain if the two adjacent anchors in the original chain are far apart. The implementation of HQAlign also addresses this issue in a different (and complementary manner) by focusing on a narrow region on the reference to enable alignments with a better chaining score, and the statistics to quantify the improvement in HQAlign over the modifications in minimap2 ([41]) are in section 4.2.5. (2) QAlign has a high false discovery rate for SV detection because it is not optimized for the downstream structural variant callers (such as Sniffles), and was solely designed for the purpose of accurate alignments of the nanopore reads using the current-level modeling (refer to section 4.2.5 for statistics and more details).

(3) Further, we have modified the minimap2 (v2.24) pipeline for the simultaneous alignment of the quantized forward and the quantized reverse complement read sequences to the quantized region of interest on the genome in the hybrid step. This enables resolving the high false discovery rate for SVs in QAlign and the detection of the inversion variants within the alignment using the quantized sequences. This is explained in detail in section 2.2. Further, HQAlign is more than 2.5x faster than QAlign standalone as it narrows down the seed search domain for lower alphabet size (*e.g.* three levels) in QAlign. This strategy is explained in Figure 4.2b, and mathematically in the following sections.

4.2.1 Quantization method from QAlign [1]

The nucleotide sequences are inferred from the nanopore current signals by basecallers, therefore, using a Q-mer map to translate the basecalled sequences to the current levels implicitly maintains all of the "equivalent" basecalled sequences that could be inferred from the observed current levels. These current levels can be quantized to an alphabet of finite size.

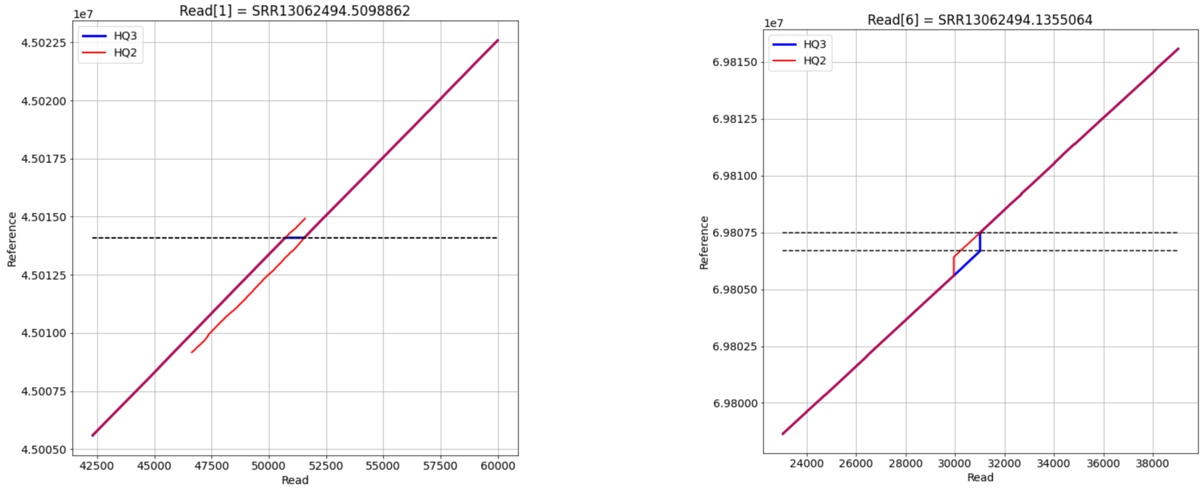
Mathematically, the quantization process is as follows. Let $\Sigma = \{A, C, G, T\}$ be the alphabet of nucleotide sequences. For a symbol $s \in \Sigma$, let $\bar{s}$ be the Watson-Crick complement of s. A string $x = s_1 s_2 \ldots s_n$ over Σ is called a nucleotide sequence, where $|x| = n$ is the string length and the reverse complement of x is $\overline{x} = \overline{s_1 s_2 \ldots s_n} = \bar{s}_n \bar{s}_{n-1} \ldots \bar{s}_1$. Let $p(x)$ be a list of all Q-mers (e.g. Q=6) in the string x, sorted by their occurrences. For example, $p(x) = k_1 k_2 \ldots k_{n-Q+1}$ and each Q-mer $k_i = s_i s_{i+1} \ldots s_{i+Q-1}$ for $i = 1, 2, \ldots, n-Q+1$. Now, we define $f : \Sigma^Q \to \mathbb{R}$ as the Q-mer map [2], which is a deterministic function that translates each Q-mer (k_i) to the (median) current level (Figure 4.1a). Now, let $C(x) = c_1 c_2 \ldots c_{n-Q+1}$ be the sequence of the current levels, such that $c_i = f(k_i)$ for $i = 1, 2, \ldots, n - Q + 1$. The

[2] Q-mer map is determined by the chemistry of the nanopore flow cell, and is therefore dataset dependent, *i.e.*, the Q-mer map for sequencing using R9 flow cell is different from Q-mer map for sequencing using R9.4.1 flow cell. The Q-mer maps used in this work are generated by Nanopolish (https://github.com/jts/nanopolish).

current sequence $C(x)$ can be further quantized into $w(x) = q_1 q_2 \ldots q_{n-Q+1}$ by applying hard thresholding function $q_i = g(c_i)$. For ternary quantization, in $HQ3$, $q_i \in \{0, 1, 2\}$ (as shown in an example in Figure 4.1c). We define $w(\overline{x})$ as the quantized reverse complement sequence of x, $\overline{w}(x) = w(\overline{x})$.

4.2.2 Choice of Quantization level

$HQ2$ and $HQ4$ does not perform as well as $HQ3$ in detecting the SVs because of the following reason.



(a)

(b)

Figure 4.3: Comparison of $HQ2$ and $HQ3$ alignments around an SV region. (a) Captures the alignment around an insertion SV in a repeat region (the location of the SV on the reference is shown as dotted black line). $HQ2$ (in red) tends to extend the alignment around the SV due to higher probability of false positive matches in low sequence complexity. (b) Captures the alignment around a deletion SV (the location of SV on the reference is shown in dotted black lines). $HQ2$ has a higher variance in SV breakpoints.

In case of $HQ2$, although the overall alignment is correct (in terms of the start and the end location of the read mapping to genome) but because of the coarse quantization (to only two levels), it becomes hard to determine an accurate alignment of $Q2$ sequences in a repeat region containing an SV. This results in a high variance of the SV breakpoints amongst the reads that overlaps around the SV in repeat regions. A few examples comparing the alignment of $HQ2$ and $HQ3$ in are shown in Figure 4.3. It is evident from Figure 4.3a that it is difficult to accurately align $Q2$ sequences around an SV in the repeat region, and the breakpoint of SV captured has a higher variance than $HQ3$ in Figure 4.3. Therefore, it becomes difficult to make a confident call on the SV as observed by the variant caller such as Sniffles, and leads to a lower recall and precision rate with $HQ2$. Further, $HQ2$ is computationally more expensive (as shown in Table 4.1 below) since there could be many false positive seed matches in the chaining algorithm of minimap2 and it takes more computation to identify the correct chain due to low sequence complexity.

Table 4.1: The evaluation of SV detection for $HQ3$, $HQ2$, and $HQ4$ in terms of Recall, Precision and F1 score. The computation time is benchmarked for each method with 45 CPU cores. $HQ3$ has better performance compared to other quantization levels.

alignment method (minimizer length)	**Computation time (in sec)**	**Ground truth SV set**	**Recall**	**Precision**	**F1 score**
$HQ3$ (k=18)	$50,069$	GIAB Tier 1	0.94	0.94	0.94
		HG002 assembly	0.75	0.79	0.77
$HQ2$ (k=21)	$145,325$	GIAB Tier 1	0.90	0.92	0.91
		HG002 assembly	0.70	0.77	0.73
$HQ4$ (k=15)	$51,064$	GIAB Tier 1	0.93	0.93	0.93
		HG002 assembly	0.73	0.78	0.76

In case of $HQ4$, it might be intuitive to expect that higher levels of quantization leads to better performance. However, this is not true for HQAlign since once the current levels are quantized at finer levels, we call the quantized levels to be matched (in the alignment) only if they match at a finer level. As we observe from the Q-mer map of R9.4 nanopore flow cell (shown in Figure 4.1a), the difference between the median current values of the consecutive 6-mers is small. Therefore, a finer quantization leads to a higher rate of implied substitution errors due to quantization. In this chapter, we show that the coarse three level quantization ($HQ3$) is sufficient to obtain good accuracy for SV calling as well as fast alignment.

4.2.3 Initial alignment

The nucleotide query x is aligned to a nucleotide target sequence t using minimap2. This is similar to aligning a read to a genome with one chromosome. Here we consider only one chromosome in target t for simplicity but the method generalizes to multiple chromosomes in t such as $t = (t_1, t_2, \ldots, t_m)$. This step identifies the regions of interest on the target t, say, $t[s_i : e_i]$, where $i \in \{1, 2, 3, \ldots\}$ represent one or more alignments on t and s_i and e_i are the corresponding start and end location of each alignment i on target t, respectively.

Generalization of HQAlign method: The nucleotide query x is aligned to a set of nucleotide target sequences $t = (t_1, t_2, \ldots, t_m)$ using Minimap2. This is similar to aligning a read to a genome which has several chromosome sequences. This step identifies the region of interest on the target t, say, $t_j[s_i : e_i]$, where t_j, $j \in \{1, 2, \ldots, m\}$ represent alignment to one or more target chromosomes that x aligns to, $i \in \{1, 2, 3, \ldots\}$ represent represent one or more alignments to chromosome j, s_i and e_i are the corresponding start and end location of each alignment i on the target t_j, respectively.

4.2.4 Hybrid alignment

In this step, the query x is re-aligned to an extended region of interest on the target $t[s_i^q : e_i^q]$ in the quantized current-level space, where $s_i^q = s_i - b_i$ and $e_i^q = e_i + b_i$, $b_i = (1 - f_i + 0.25)n$ is an appended extension of the region of interest on target, $f_i = (e_i - s_i)/n$ is the fraction of read aligned in the initial step, and n is the length of the query x. The nucleotide query x and the nucleotide extended target $t[s_i^q : e_i^q]$ are converted to the quantized query x^q and quantized extended target $t^q[s_i^q : e_i^q]$, respectively, using the quantization method demonstrated in QAlign (refer to section 4.2.1 for more details on quantization process).

Generalization of HQAlign method: In this step, the query x is re-aligned to an extended region of interest on the target $t_j[s_i^q : e_i^q]$ using the quantized sequences, where $s_i^q = s_i - b_i$ and $e_i^q = e_i + b_i$, $b_i = (1 - f_i + 0.25)n$ is an appended extension of the region of interest on target, $f_i = (e_i - s_i)/n$ is the fraction of read aligned in initial step, and n is the length of the query x. The nucleotide query x and the nucleotide extended target $t_j[s_i^q : e_i^q]$ are converted to the quantized query x^q, quantized reverse complement query $\overline{x}^q$ and quantized extended target $t_j^q[s_i^q : e_i^q]$, respectively, using the quantization method demonstrated in QAlign. These quantized sequences are then aligned using modified minimap2 pipeline that performs alignment using both the quantized forward and the quantized reverse complement query, simultaneously.

It is important to note that we do not use any additional soft information such as raw current signals from nanopore sequencing in the quantization process, instead, we translate the base-called nucleotide reads to current levels using the Q-mer map (in Figure 4.1a) and then hard threshold the current levels to finite (*e.g.* three) levels to get the quantized ($HQ3$) reads (in Figure 4.1c). The choice on three levels of quantization is because of the following reasons: (1) using a coarser quantization to only two levels makes it difficult to accurately align low complexity sequence in a repetitive region containing an SV, therefore, it becomes hard for the variant caller to make a confident call for such SVs due to high variance of the SV breakpoints. (2) using higher level of quantization requires a match in the current levels at a finer level for the alignment of sequences. As we observe from the Q-mer map of R9.4 nanopore flow cell (shown in Figure 4.1a), the difference between the median current values of the consecutive 6-mers is very small. Therefore, a finer quantization leads to a higher rate of implied substitution errors due to quantization. In this study, we show that the coarse three level quantization (HQ3) is sufficient to obtain good accuracy for SV calling as well as fast alignment.

These quantized sequences are then aligned using a modified pipeline of minimap2 (v2.24). We have modified minimap2 pipeline for this hybrid step to take the quantized reverse complement query $\bar{x}^q$ as an input which helps in identifying the inversion SVs within the contiguous alignment of quantized sequences which was not possible with the earlier QAlign method as shown in Figure 4.2a. QAlign uses the default minimap2 pipeline for the alignment of quantized sequences. While minimap2 can inherently compute and align the reverse complement of a read in the nucleotide domain, the quantized reverse complement sequence cannot be computed given only the forward quantized sequence, therefore, QAlign separately aligns both quantized forward and quantized reverse complement sequence. This method, however, fails to identify an inverted alignment as shown in Figure 4.2a, and also results in a high false discovery rate for SV calling since it is not optimized for the downstream structural variant callers such as Sniffles2. Therefore, in HQAlign, we have modified the minimap2 pipeline to enable alignment using both quantized forward and quantized reverse complement sequences, simultaneously. Note that the quantized alignment employs a different minimizer length `k` $= 18$ in minimap2 for ternary ($HQ3$) quantization.

4.2.5 How HQAlign distinguishes from QAlign?

There are two ideas that distinguishes HQAlign from QAlign summarized below:

1. In the original implementation of minimap2 [20], it only uses the low occurrence minimizers during read mapping. The new modifications in minimap2 [41] address this problem with a heuristic that adds additional lowest occurrence minimizers to the chain if the two adjacent anchors in the original chain are far apart. The implementation of HQAlign addresses this issue in a different (and complementary manner) by focusing on a narrow region on the reference to enable alignments with a better chaining score, and we have provided statistics to quantify the improvement in HQAlign over the modifications in minimap2 [41]. Further, HQAlign does the necessary fix to QAlign algorithm to enable accurate structural variant calling with the quantized reverse complement sequences.

2. It is not feasible to use QAlign for the detection of structural variants because of the high false discovery rate of SVs with QAlign as shown in Table 4.3. The reason for the high false discovery rate of SVs in QAlign is because it was not optimized for the down-

stream structural variant callers (such as Sniffles). Therefore, we designed HQAlign that enables SV detection using the current-level modeling of nanopore sequencers.

We elaborate on the arguments made above in detail here:

1. **Improving identification of minimizers in the chain:** In HQAlign, we first determine the region of interest on the reference through the alignment of a read in the nucleotide space and then re-align the quantized read to the quantized region of interest on the reference. The idea is to determine an accurate alignment of the read on the reference in the quantized space without dropping frequently occurring minimizers from the mapping chain in minimap2. However, in QAlign (coupled with the original minimap2 algorithm), since the minimizers are indexed from the entire reference the heuristic of dropping frequently occurring minimizers could result in shorter alignments or false alignments.

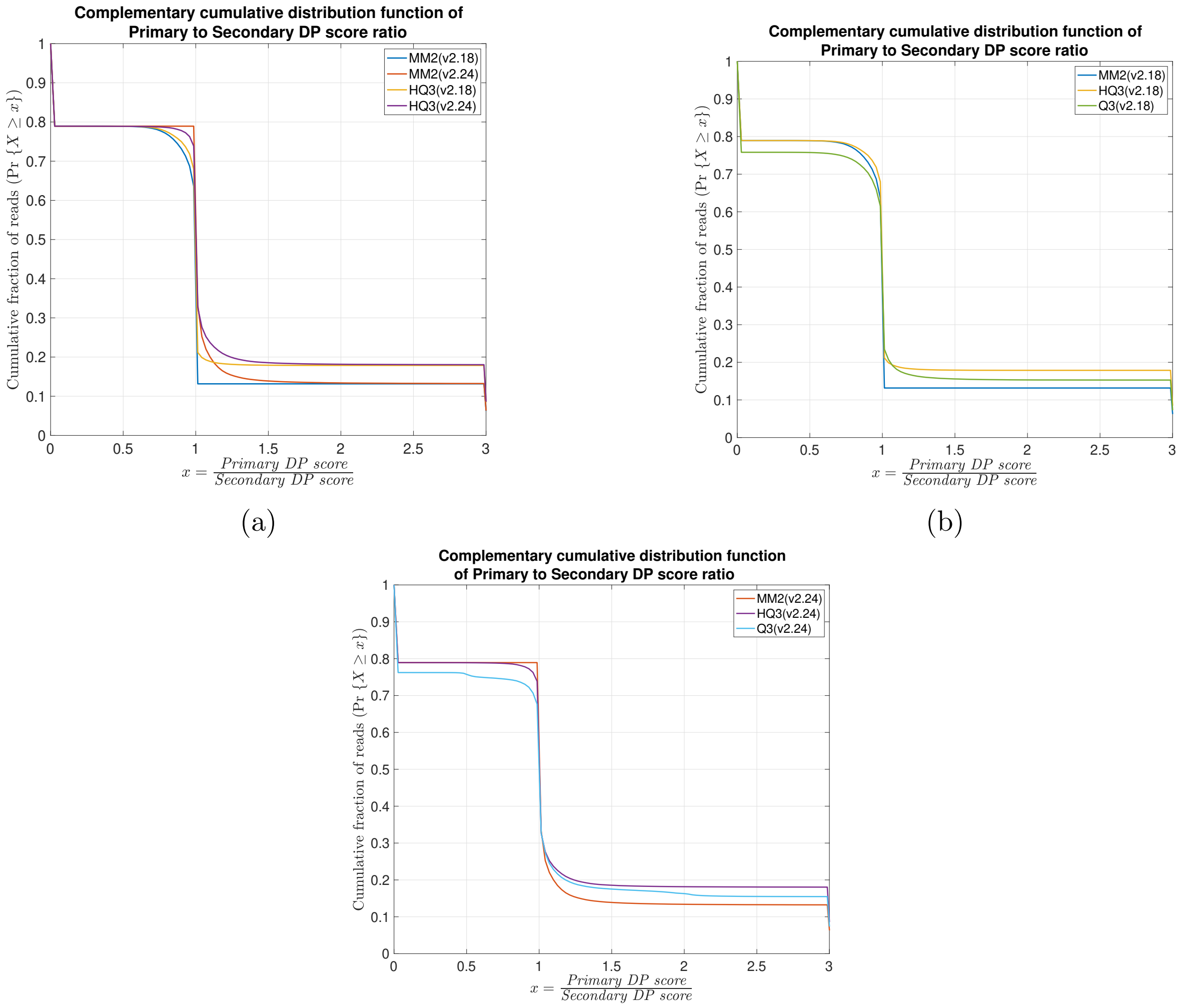
   This can be analyzed statistically by comparing the dynamic programming (DP) score of the primary and the secondary alignments in minimap2 algorithm. There are $152,237$ reads out of 10.36 million reads dataset for which the primary label to alignments in minimap2 (v2.24, the modified minimap2 [41]) does not match with the primary label to alignments in minimap2 (v2.18, the old minimap2 implementation [20]), however, the alignments from old and new version have an overlap of at least 90% on the reference (here the reads are aligned in the nucleotide space using the standard minimap2 pipeline). This change in label could be because of improved chaining heuristic for low occurrence minimizers, and it is analyzed by comparing the ratio of the DP score of the primary alignment to DP score of the secondary alignment for different alignment methods (such as minimap2, QAlign and HQAlign). For instance, consider the example shown in Table 4.2 for a single read case - the read aligns to chromosome 14 and a contig in the human reference genome GRCh37 with the DP score and the mapping quality[3] of the alignment as shown in the table. Considering the primary alignment to chromosome 14 and the secondary alignment to contig from minimap2 (v2.24) in this example as a ground truth label, we compute the ratio of the primary DP score to secondary DP score for all the alignment methods listed in the table with respect to the ground truth label determined by minimap2 (v2.24). The results shows that $HQ3$ (with minimap2 v2.24) has a better DP score ratio and mapping quality than other alignment methods.

Table 4.2: In this example, a single read aligns to chromosome 14 and a contig of the human reference GRCh37, and the DP score of both the alignments from different methods are shown in the first two rows in this table (minimap2, Q3, and $HQ3$ alignments using minimap2 (v2.18), and minimap2, Q3, and $HQ3$ alignments using minimap2 (v2.24)). The mapping quality of the alignments is in the third row. Since the absolute values of the DP score are not comparable due to the change in the chaining algorithm of minimap2 from v2.18 to v2.24, we compute the ratio of the primary DP score to secondary DP score (the ground truth of primary and secondary alignment label is defined based on the alignment of minimap2 (v2.24)). $HQ3$ (with minimap2 v2.24) has a better alignment in terms of the mapping quality and the DP score ratio.

alignment to	**v2.18**			**v2.24**		
	mm2	**Q3**	**HQ3**	**mm2**	**Q3**	**HQ3**
chromosome 14	51880	50480	54618	28137	27566	29123
contig	51910	50370	54468	27079	26314	27411
mapping quality	0	17	20	35	39	47
DP score ratio	0.999	1.002	1.003	1.039	1.048	1.062

Next, we analyze the metric of DP score ratio comparison across different alignment methods for the $152,237$ reads is shown in Figure 4.4. The ground truth of the primary and the secondary alignment label of a read is defined by the alignments of minimap2 (v2.24), and the ratio of the DP score for alignments from other methods are computed based on these ground truth labels (similar to example in Table 4.2). The average alignment performance of a method is measured by the complementary cumulative

[3] The mapping quality is a measure of the ratio of the primary to secondary chaining score and it is strictly positive for a primary alignment and zero for any secondary alignments for a given query alignment to reference.



(a)

(b)

(c)

Figure 4.4: This figure compares the complementary cumulative density function for the ratio of the primary to secondary DP score for different alignment methods for $152,237$ reads that have a change in the primary/secondary labels between minimap2 alignments from v2.24 and v2.18. The plots are divided into three subplots to make them easier to read.

distribution function $\mathbb{P}[X \geq x]$, where X is the random variable that represents the ratio of the DP score, and the probability measure gives the fraction of reads that satisfies the inequality $X \geq x$. Therefore, higher the probability for any x represents better alignment.

It is evident from Figure 4.4(a) that $HQ3$ (with minimap2 v2.24) has a better score ratio, although the ground truth is biased for minimap2 (v2.24). Further, it is important to note that $HQ3$ (with minimap2 v2.18) (represents the performance of aligning only onto the region of interest with the original algorithm of minimap2) has better alignment for $x > 1.1$ than minimap2 (v2.24) and Q3 (v2.18).

Further, we have analyzed the average performance improvement in mapping quality of alignments (that represents the improvement in chaining scores) using the entire read dataset. It is measured in terms of the complementary cumulative distribution function of the mapping quality of alignments $\mathbb{P}[X \geq x]$, where X is a random variable that represents the mapping quality, and the probability measure gives the fraction of reads that satisfies the inequality $X \geq x$, for the entire read dataset (with 10.36 million reads) that are aligned across different methods as shown in Figure 4.5. It is evident from the figure that the optimization of aligning onto the region of interest has a better performance in $HQ3$ (with both minimap2 v2.18 and minimap2 v2.24).

2. **False discovery rate of SVs using QAlign:** It is not feasible to use QAlign for the detection of structural variants because of the high false discovery rate of SVs with QAlign as shown in Table 4.3. The reason for the high false positive rate in QAlign is because it is not optimized for the downstream structural variant callers (such as Sniffles), and was solely designed for the purpose of accurate alignments of the nanopore reads using the current-level modeling. In QAlign, since the strand of the read sequenced is not known, QAlign performs the alignment of a read to the reference separately for both the forward and the reverse strand of the reads and then aggregates the alignments from each strand in the final output. This results in false labeling of primary alignments from both the strands and therefore, Sniffles make false positive SV calls from these primary alignments from both strands. For example, if a read is sequenced from reverse complement strand of an approximate repeat region on a chromosome, it might also have a forward strand alignment to a similar approximate repeat sequence on another chromosome. However, only one of the two alignments

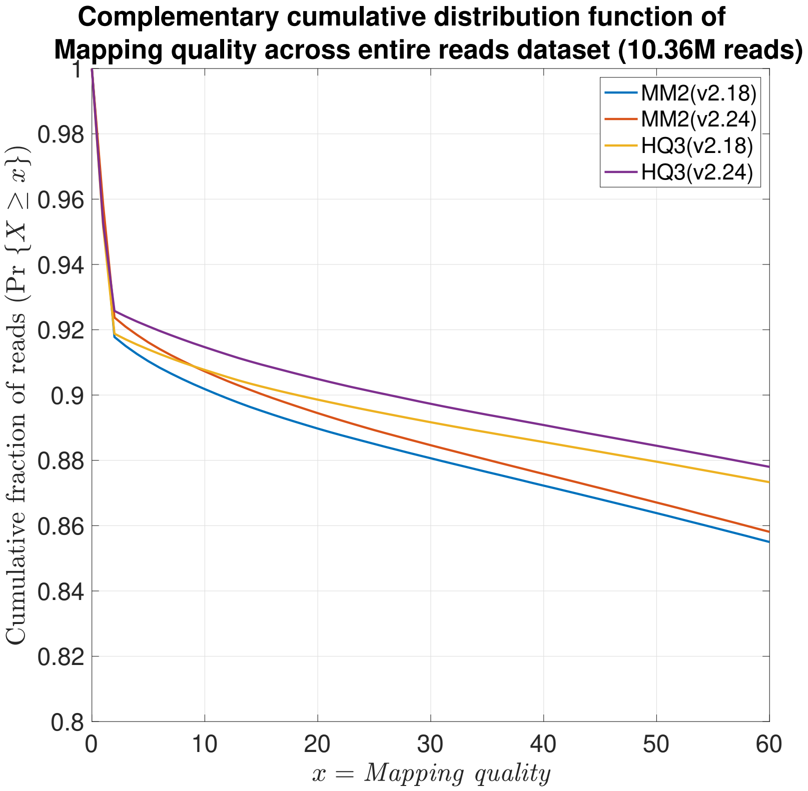


Figure 4.5: Complementary cumulative distribution function of the mapping quality of alignments for entire read dataset (with 10.36 million ONT reads).

is more likely based on the chaining and DP scores, and should be labeled primary accordingly. In QAlign, since each strand of the read is aligned separately to the entire genome, it results in labeling both the alignments to different chromosomes as primary and therefore, contributing to the false SV calls. This is fixed in HQAlign by performing the initial alignment of the reads in the nucleotide space which not only determines the region of interest but also provides the information about the strand of the aligned read, and later in the hybrid step, the quantized read of only the corresponding strand is re-aligned to the quantized region of interest. Therefore, resolving the high false discovery rate for SVs in QAlign.

Table 4.3: SV calls for QAlign (Q3 (with minimizer length $k = 18$)) and it comparison with ground truth sets: (i) GIAB Tier 1 SV calls and (ii) SV calls from HG002 haplotype resolved assembly comparison to GRCh37 reference.

alignment method	**Ground truth SV set**	**True positive calls**	**False positive calls**
$HQ3$ (v2.24)	GIAB Tier 1	8949	584
$HQ3$ (v2.24)	HG002 assembly	16470	4510
Q3 (v2.18)	GIAB Tier 1	8743	6853
Q3 (v2.18)	HG002 assembly	16034	24020
Q3 (v2.24)	GIAB Tier 1	8899	6562
Q3 (v2.24)	HG002 assembly	16384	23170

4.2.6 Performance metrics

We define several metrics that are used for the performance evaluation of HQAlign against minimap2 (some of these metrics are used from the earlier QAlign method ([1])).

(i) **well-aligned:** Consider in Figure 4.2c, *Read 1* aligns at location i_1 through j_1 on the genome determined using nucleotide alignment. We say that the read is well-aligned if at least 90% of the read is aligned onto the genome (i.e., $j_1 - i_1 \geq 0.9(\text{length}(\textit{Read 1}))$), and has high mapping quality (greater than 20). This metric quantifies the reads that are mapped almost entirely to the reference.

(ii) **normalized edit distance:** In order to compare the quality of the alignments at a fine-grained level, we further define normalized edit distance. The normalized edit distance for nucleotide alignment is defined as

$$\frac{\text{edit\_distance}\{r; G[i_1 : j_1]\}}{\text{length}(r)} \tag{4.1}$$

and for quantized alignment is

$$\frac{\text{edit\_distance}\{r; G[i_1^q : j_1^q]\}}{\text{length}(r)} \tag{4.2}$$

where i_1, j_1 are the start and end location of alignment on the genome in nucleotide space and i_1^q, j_1^q are the start and end location of alignment on the genome in the quantized space, r is the entire read and G is the genome. It is important to note that for computing the normalized edit distance for alignments in the quantized space, we only leverage the information of the location of the alignment on genome from quantized space, *i.e.*, i_1^q and j_1^q, but the edit distance between read and the aligned section on the genome is computed on the nucleotide sequences. This metric gives a measure of the distance similarity between two sequences, especially, used for the real data where the truth of sequence sampling location is not known.

(iii) **normalized alignment length:** Another metric at the fine-grained level is normalized alignment length, which is the ratio of the length of the section on the genome where a read aligns to the length of the read. It is

$$\frac{j_1 - i_1}{\text{len}(r)} \tag{4.3}$$

for nucleotide alignment, and

$$\frac{j_1^q - i_1^q}{\text{len}(r^Q)} \tag{4.4}$$

for quantized alignment. A contiguous alignment tends to have this metric as 1. This metric gives a measure of the contiguity of the alignment.

4.2.7 SV calling

The alignments from HQAlign and minimap2 in sorted *bam* format are used to detect structural variants using Sniffles2. These calls are benchmarked against a truth set using Truvari

([46]). We have used the F1 score, precision, and recall as the metric to analyze the performance of HQAlign and compare them with minimap2. Precision (P) is defined as the fraction of SVs detected by the algorithm in the truth set among the total SVs detected by the algorithm. Recall (R) is the fraction of SVs detected by the algorithm in the truth set among the total SVs in the truth set. F1 score is the harmonic mean of precision and recall ($= \frac{2P \cdot R}{P+R}$). Further, we have observed that there are many complementary SV calls made by both minimap2 and $HQ3$ that are missed by the other method. Therefore, we have defined a union model which takes a union of the SV calls from both minimap2 and $HQ3$. The precision, recall, and F1 score of the union model are also computed and reported in Table 4.7.

Further, the quality of the SVs for the common calls in minimap2 and HQAlign is evaluated by comparing the following metrics w.r.t. the SVs in truth set

(i) **breakpoint accuracy:** Breakpoint accuracy is measured by taking an average of the difference in the start and end breakpoint of the SV w.r.t. the SV in truth set. For instance, as shown in Figure 4.2d, i_1 and j_1 are the start and the end point on the genome of SV in the truth set, and $i_1^{'}$ and $j_1^{'}$ are the start and the end point of the same SV determined by any alignment method (minimap2/$HQ3$), then breakpoint score is calculated as

$$\frac{|i_1^{'} - i_1| + |j_1^{'} - j_1|}{2} \tag{4.5}$$

where $|\cdot|$ is absolute value function. Therefore, the lower the score higher the breakpoint accuracy of the SV determined by the alignment method.

(ii) **SV length similarity:** SV length similarity is measured as the ratio of minimum SV length in the truth set and from the algorithm to the maximum of two values. Mathematically, it is

$$\frac{\min(j_1 - i_1, j_1^{'} - i_1^{'})}{\max(j_1 - i_1, j_1^{'} - i_1^{'})} \tag{4.6}$$

for the example shown in Figure 4.2d.

4.3 Results

In this section, we demonstrate the results for (1) comparison of alignments from $HQ3$ and minimap2 on real as well as simulated data, and (2) comparison of SV calls from $HQ3$ and minimap2 alignments using Sniffles2 as the variant caller on real and simulated data.

4.3.1 DNA read-to-genome alignment

4.3.1.1 Datasets

We have used the publicly available R9.4.1 ONT PromethION reads dataset from HG002 sample ([47]). These reads are aligned to the recent telomere-to-telomere assembly CHM13 and the human reference genome GRCh37. GRCh37 is used as the reference build to map the real data so that the curated variants can be used for accuracy analysis ([48]). Further, we have also benchmarked the performance of HQAlign and minimap2 on simulated data for both alignments and SV calling.

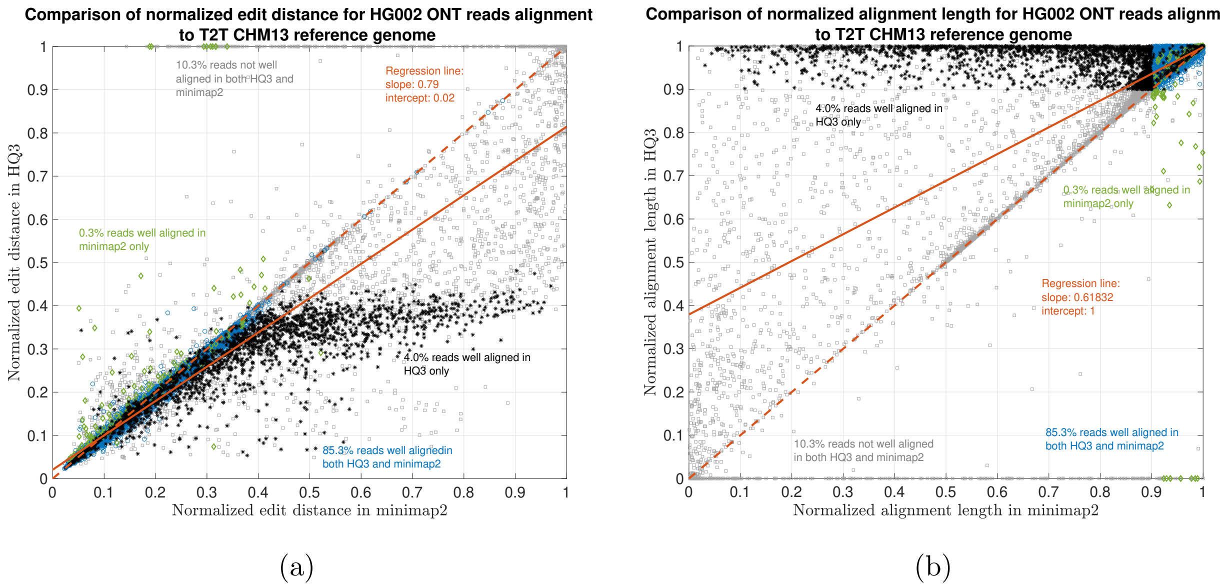
4.3.2 Computation time

We have benchmarked the computation time for minimap2, QAlign, and HQAlign (total time including both the initial step and the hybrid step) with 45 CPU cores in Table 4.4 using the entire reads dataset (with 10.36M reads instead of 500k randomly sampled reads as in the earlier submission).

Table 4.4: Computation time (wall clock time) for different alignment method on the entire dataset with 10.36M reads.

alignment method (minimizer length)	**Computation time (in sec)**
Minimap2 (k=15)	$24,479$
$HQ3$ (k=18)	$50,069$
Q3 (k=18)	$141,253$

4.3.2.1 Alignment results



(a)

(b)

Figure 4.6: **HG002 nanopore long DNA reads alignment onto T2T CHM13 genome.** (a) Comparison of normalized edit distance for HG002 R9.4.1 PromethION reads data. Smaller values for normalized edit distance are desirable as it represents better alignment. The slope of the regression line is $0.79 < 1$, therefore, representing better alignments with $HQ3$ than minimap2 alignments for the same reads on average. (b) Comparison of normalized alignment length for HG002 R9.4.1 PromethION reads data. Normalized alignment length of 1 is desirable as it represents that the entire read is aligned. The majority of the reads are above $y = x$ line representing longer alignment length in $HQ3$ than minimap2 alignment.

The alignment of DNA reads to the genome is a primitive step in structural variant calling pipelines ([34]). $HQ3$ alignments show an improvement over minimap2 alignments in terms of contiguity measured by normalized alignment length and alignment quality measured by normalized edit distance.

The results are illustrated in Figure 4.6, 4.8, 4.9, and Table 4.5. At a coarse level, the performance is measured by the fraction of the reads that are well-aligned by the algorithm. A read is well-aligned if at least 90% of the read is aligned to the genome and has a high mapping quality (see HQAlign algorithm section). HQAlign improves the fraction of well-aligned reads than minimap2 - in particular, in the HG002 R9.4.1 reads alignment to T2T

Table 4.5: Comparison for the percentage of well-aligned reads onto genome, and slope of the regression line from normalized edit distance comparison plot of $HQ3$ vs minimap2 alignments with randomly sampled reads for each dataset (the reads are randomly sampled to reduce the amount of edit distance computations). The slope of the regression line shows the average gain in the normalized edit distance over the subsampled reads in each dataset.

Dataset (No. of sampled reads)	**Method of alignment**	**well-aligned reads (%)**	**Slope of regression line**
HG002 R9.4.1 reads to CHM13 (50k)	minimap2	85.64	0.7940
	$HQ3$	89.35	
HG002 R9.4.1 reads to GRCh37 (50k)	minimap2	83.48	0.8301
	$HQ3$	86.65	
Simulated reads from chr 8 & X of CHM13 assembly (50k)	minimap2	81.01	0.9860
	$HQ3$	81.57	

CHM13 reference, this metric improves to 89.35% from 85.64%, and for the alignments to GRCh37 reference, this metric improves to 86.65% from 83.48%. Furthermore, there are $310,036$ reads (from the entire dataset with 10.36M reads) with at-least 1kb additional bases aligned using HQAlign compared to minimap2 alignments for T2T CHM13 reference, and there are $299,896$ reads with at least 1kb additional bases aligned using HQAlign compared to minimap2 for GRCh37 reference.

The results in Figure 4.6 and Figure 4.8 compare the quality of the alignments using minimap2 and HQAlign at a fine-grained level for HG002 ONT reads alignment to T2T CHM13 genome and GRCh37 genome, respectively. Figure 4.6a and Figure 4.8a compare the normalized edit distance for HQAlign and minimap2. The normalized edit distance is the edit distance between the entire read and the aligned section on the genome normalized by the length of the read, in nucleotide domain for *both* minimap2 alignment and quantized alignment ($HQ3$). In the case of $HQ3$, the information of the location of the alignment on the genome is leveraged from the quantized read and the quantized genome alignment, and the edit distance is computed between the corresponding nucleotide read and the aligned region on the nucleotide genome (see Methods for details). Intuitively, the normalized edit distance gives a measure of how close the two sequences are. Therefore, the smaller the normalized edit distance, the better the alignment.

Figure 4.6a shows that for alignments of the reads to T2T CHM13 reference, the normalized edit distance is on average smaller for $HQ3$ alignments than minimap2 alignments. The better alignment in $HQ3$ is also evident from the slope of the regression line in Figure 4.6a. It shows that on average $HQ3$ alignments have 21% improvement in terms of the normalized edit distance than the minimap2 alignments. Well-aligned reads in both $HQ3$ and minimap2 are represented by blue circles in Figure 4.6, well-aligned reads in $HQ3$ only are represented in black asterisks, well-aligned in minimap2 only are represented in green diamonds and reads that are not well-aligned in both are represented in grey squares. Further, it is important to note that for normalized edit distance less than 0.1, the alignments are marginally better in the DNA space, but for normalized edit distance higher than 0.1, the alignments are significantly better in $HQ3$ space, especially, the 4% reads that are well aligned in $HQ3$ and not well aligned in minimap2. This is because of the higher contiguity of alignments in $HQ3$ space and signifies the improvement by $HQ3$ when the error rates are higher. For alignments to GRCh37 reference, $HQ3$ has an average improvement of 17%, as shown in Figure 4.8a.

The results for another fine-grained metric are shown in Figure 4.6b and Figure 4.8b, which compares the normalized alignment length in $HQ3$ to the normalized alignment length in minimap2 alignments. The normalized alignment length is the ratio of the length of the section on the genome where a read aligns to the length of the read. In Figure 4.6b, there are 4% reads that are well-aligned in $HQ3$ only, and the normalized alignment length is close to 1 in $HQ3$ but it is much less than 1 in minimap2, therefore representing several non-contiguous alignments in nucleotide domain that are captured as contiguous alignment in $HQ3$. In Figure 4.8b, there are 3.7% that are well-aligned in $HQ3$ only.

We have also benchmarked the performance of HQAlign with the simulated reads data and compared its alignment performance with minimap2 in Figure 4.9. The ONT reads are simulated from chromosome 8 and X of CHM13 T2T assembly using nanosim ([49]) with coverage of 40x, median and mean read length 4.5 kb, and 14 kb, respectively. The results show that the alignment performance of both HQAlign and minimap2 are at par with each other.

4.3.3 Alignment performance

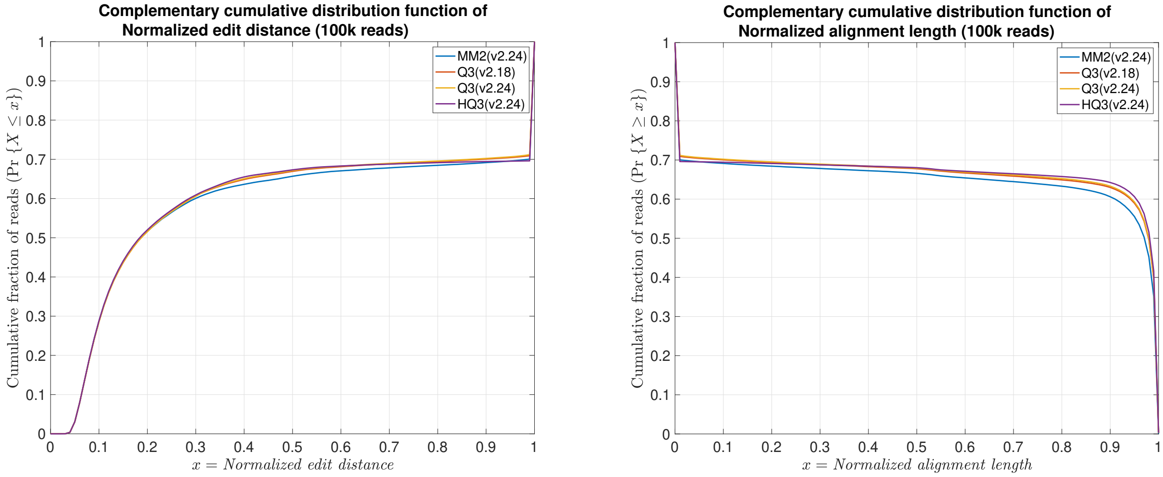
We have compared the alignment quality in HQAlign to the best alignment in QAlign in Figure 4.7 (the best alignment in QAlign is the alignment with minimum normalized edit distance from the aggregated alignments from both strands). It is evident from the figure that $HQ3$ performs better than Q3 and minimap2 in terms of both the normalized edit distance and the normalized alignment length metric. The expected value of normalized edit distance (computed as $\mathbb{E}[X] = \int \mathbb{P}[X > x]dx$ for reads that are aligned by any method) is 0.1613 for $HQ3$ (v2.24), 0.1673 for Q3 (v2.24), 0.1678 for Q3 (v2.18), and 0.1742 for minimap2 (v2.24). Recall that the normalized edit distance for all alignment methods are computed in nucleotide space by leveraging the information about the location of the alignment in the quantized space.

$$\text{normalized edit distance} = \frac{\text{edit distance}(Genome[start : end]; read)}{\text{length of the read}} \tag{4.7}$$

The expected value of normalized alignment length (computed as $\mathbb{E}[X] = \int \mathbb{P}[X > x]dx$ for reads that are aligned by any method) is 0.9576 for $HQ3$ (v2.24), 0.9481 for Q3 (v2.24), 0.9459 for Q3 (v2.18), and 0.9320 for minimap2 (v2.24).

$$\text{normalized alignment length} = \frac{end - start}{\text{length of the read}} \tag{4.8}$$

where *start* and *end* are the start and the end location of alignment on the reference.



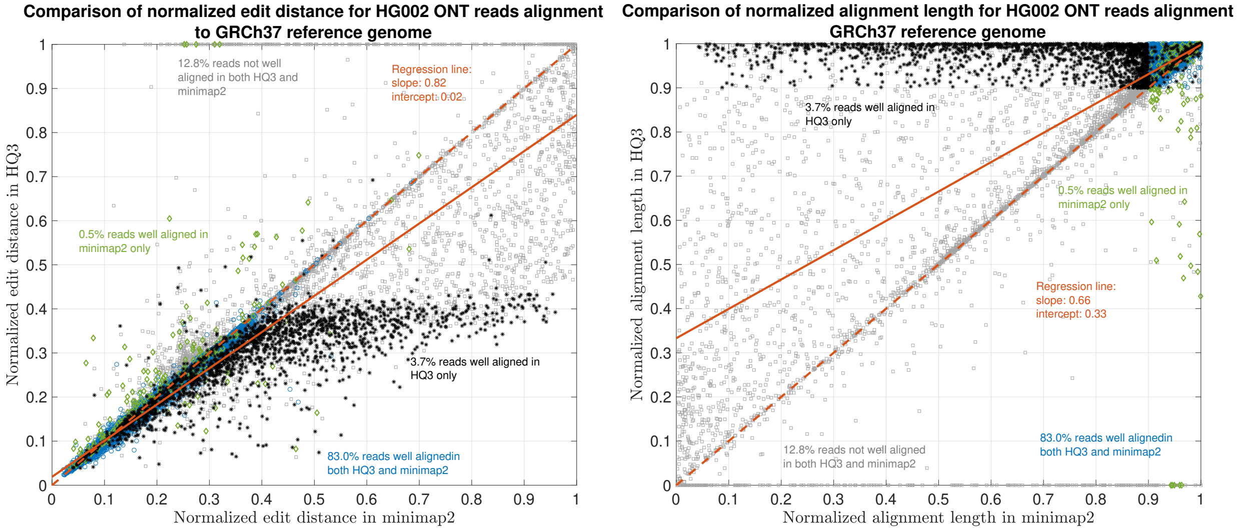
(a) (b)

Figure 4.7: Alignment quality comparison between HQAlign ($HQ3$ ($k = 18$)), QAlign (Q3 ($k = 18$)), and minimap2 ($k = 15$), where k is the minimizer length. (a) Comparison of normalized edit distance for 100k randomly sampled reads from the entire dataset. (b) Comparison of normalized alignment length for the same 100k randomly sampled read in (a). For QAlign, the best alignment in terms of normalized edit distance from both the strands is chosen for analysis.

4.3.4 SV calling

4.3.4.1 Dataset

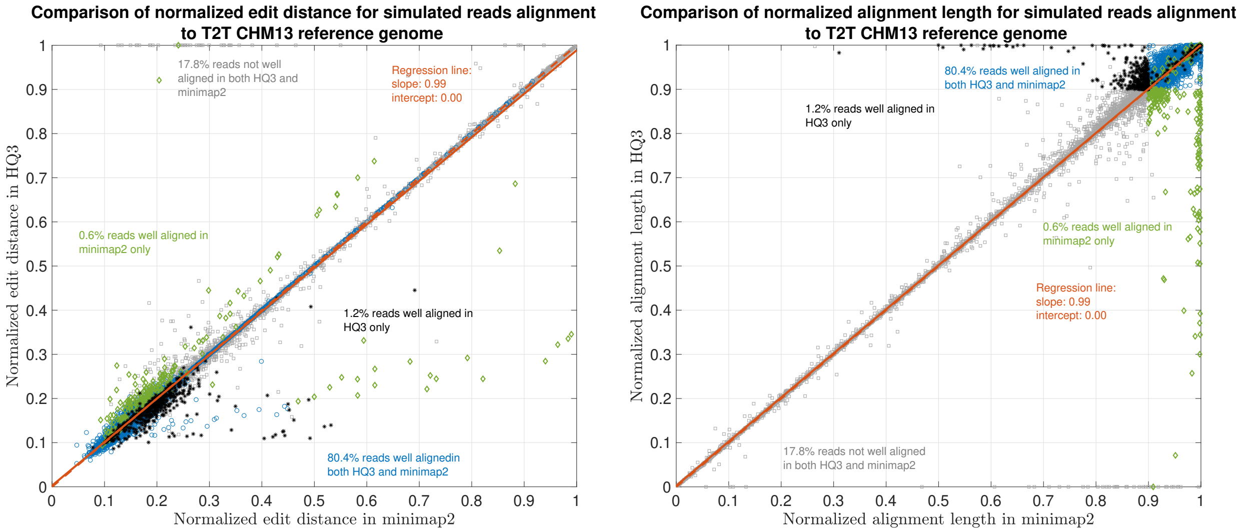
Long-read sequencing plays an important role in detecting structural variations. We evaluated SV detection using minimap2 and HQAlign with Sniffles2 as the variant calling algorithm on both real and simulated data. We simulated 2000 INDELS and 200 Inversion SVs on chromosome 8 and X of T2T CHM13 reference genome using SURVIVOR ([50]) with SV length uniformly distributed between 50 and 10000, and the ONT reads are simulated using nanosim with an average length of 14k, median length of 4.5k and maximum length 2.5Mbp at coverage of 40x. We have used Truvari to benchmark the calls against the truth set. For real data alignment with GRCh37 as the reference genome, the SV calls are compared against the ground truth sets from (1) Genome In A Bottle (GIAB) Tier 1 calls ([48]) and (2) another truth set is constructed by comparing the haplotype-resolved assembly of



(a) (b)

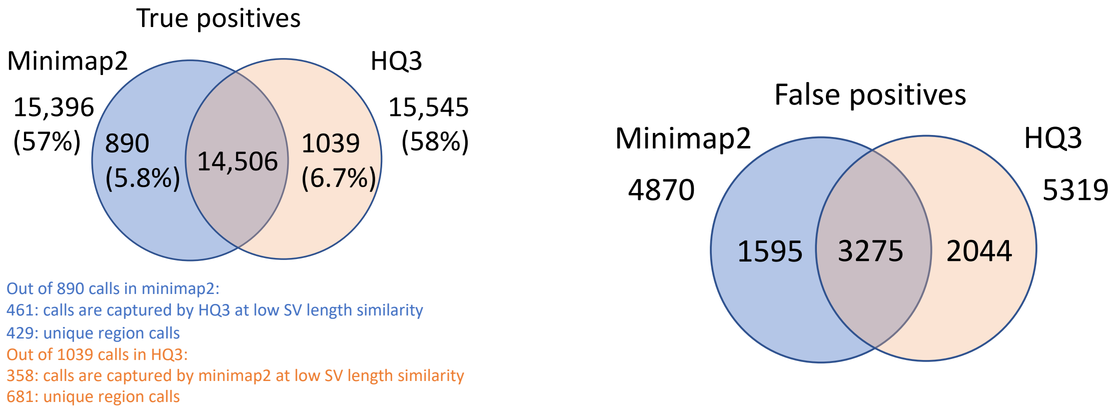
Figure 4.8: **HG002 nanopore long DNA reads alignment onto GRCh37 genome.** (a) Comparison of normalized edit distance for HG002 R9.4.1 PromethION reads data. Smaller values for normalized edit distance is desirable as it represents better alignment. The slope of the regression line is $0.82 < 1$, therefore, representing better alignments with $HQ3$ than minimap2 alignments for same reads on average. (b) Comparison of normalized alignment length for HG002 R9.4.1 PromethION reads data. Normalized alignment length of 1 is desirable as it represents that entire read is aligned. The majority of the reads are above $y = x$ line representing longer alignment length in $HQ3$ than minimap2 alignment.

HG002 against GRCh37 reference genome using dipcall ([51]). For T2T CHM13 reference genome, since the ground truth for SVs is not available, we have constructed the truth set by comparing the haplotype-resolved assembly of HG002 against CHM13 reference using dipcall. However, it is hard to establish ground truth for the SV calls that are made in the centromere regions, even though the assembly is likely to be correct. Therefore, we have provided both the analysis including the SV calls in centromere regions (in Figures 4.10 and 4.11) and the analysis for SV calls excluding the centromere regions (in Figures 4.15 and 4.18).



(a) (b)

Figure 4.9: **Simulated nanopore reads alignment onto T2T CHM13 genome.** (a) Comparison of normalized edit distance for simulated nanopore reads data. Smaller values for normalized edit distance is desirable as it represents better alignment. The slope of the regression line is $0.99 < 1$, therefore, representing marginally better alignments with $HQ3$ than minimap2 alignments for same reads on average. (b) Comparison of normalized alignment length for simulated nanopore reads data. Normalized alignment length of 1 is desirable as it represents a contiguous alignment of the entire read.



Out of 890 calls in minimap2:
461: calls are captured by HQ3 at low SV length similarity
429: unique region calls
Out of 1039 calls in HQ3:
358: calls are captured by minimap2 at low SV length similarity
681: unique region calls

(a) (b)

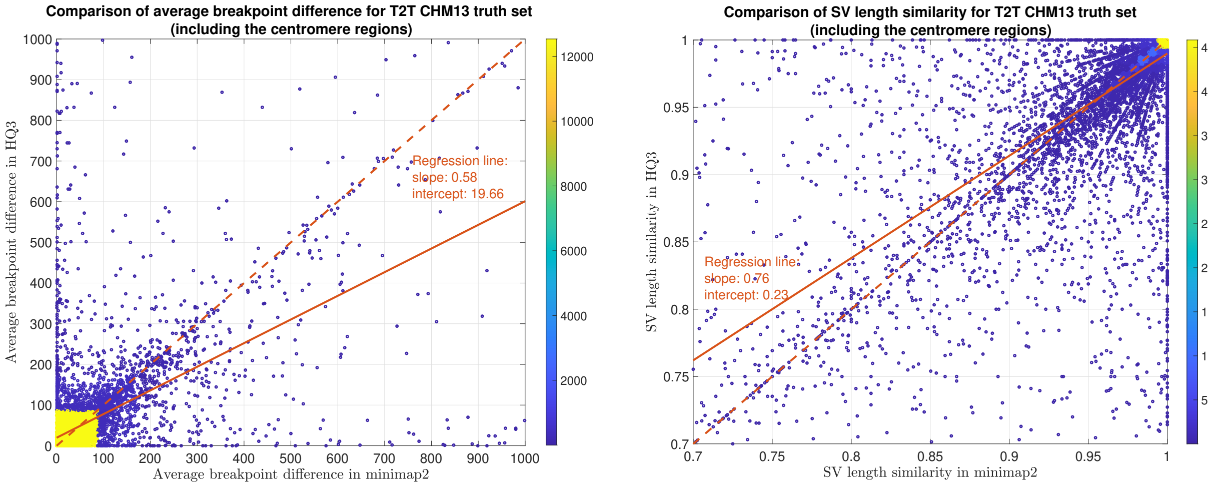
Figure 4.10: **Comparison of SV calls from $HQ3$ and minimap2 with HG002-to-CHM13 dipcall as truth set.** (a) Comparison of true positive calls. (b) Comparison of false positive calls.

4.3.4.2 SV calling results

The standalone performance of both $HQ3$ and minimap2 is at par with each other across different references and truth set used in this study for real data as well as for the simulated data in terms of the F1 score. However, both $HQ3$ and minimap2 detect complementary SV calls most likely in the repeat regions where accurate alignment is difficult and therefore, leads to many broken calls.

The analysis with comparison of SV calls from $HQ3$ and minimap2 with GIAB Tier 1 truth set gives a precision, recall, and F1 score of 0.94, 0.94, and 0.94, respectively for both minimap2 and $HQ3$. A union model of minimap2 and $HQ3$ can improve the recall rate at the same F1 score, and the union model has a precision, recall, and F1 score of 0.93, 0.95, and 0.94, respectively. Moreover, out of 103 SV calls that are made by $HQ3$ only (Figure 4.12), 41 calls are made by minimap2 alignments at a lower SV length similarity, and 62 calls are unique region calls. Out of 105 SV calls made by minimap2 only, 51 are captured by $HQ3$ at a lower SV length similarity and 54 are unique region calls. $HQ3$ improves the breakpoint accuracy by 14.11% for calls that have difference in breakpoints higher than 50 and it improves the length similarity by 19.97% that have SV length similarity lower than 0.95 (Figure 4.16).

We have compared the SV calls made by HG002 reads against T2T CHM13 reference genome using both minimap2 and $HQ3$ and benchmarked them against the truth set generated by comparing HG002 haplotype-resolved assembly to T2T CHM13 assembly. The standalone performance have precision, recall and F1 score of 0.77, 0.57 and 0.66, respectively for minimap2 and 0.75, 0.58 and 0.65, respectively for $HQ3$. However, because of the high number of complementary true positive calls in minimap2 and $HQ3$, the union model has a significantly improved recall at the same F1 score with precision, recall, and F1 score of 0.71, 0.61, and 0.66, respectively. Out of 1039 (6.7%) calls that are made in $HQ3$ only, 358 are captured by minimap2 at a lower SV length similarity threshold and 681 are unique



(a) (b)

Figure 4.11: **SV quality comparison for common true positive calls in $HQ3$ and minimap2 against HG002-to-CHM13 dipcall truth set.** (a) Comparison of SV breakpoint accuracy in $HQ3$ and minimap2 for common true positive calls. The difference in SV breakpoint is compared to the truth set generated from comparing HG002 haplotype-resolved assembly to T2T CHM13 build. A smaller difference represents better breakpoint accuracy. Therefore, the slope of the regression line $0.58 < 1$ represents better accuracy of $HQ3$ than minimap2 on average. (b) Comparison of SV length similarity in $HQ3$ and minimap2 for common true positive calls. The slope of the regression line $0.76 < 1$ represents better SV length in minimap2 than $HQ3$ on average, but the intercept is high (0.23). However, this is due to a large density of SVs with length similarity ≥ 0.95 in both minimap2 and $HQ3$. For length similarity less than 0.95, $HQ3$ has better performance than minimap2.

calls, whereas out of 890 (5.8%) calls that are made by minimap2 only, 461 are captured by $HQ3$ at a lower SV length similarity threshold and 429 are unique (as shown in Figure 4.10a). Further, for the common true positive calls in both minimap2 and $HQ3$, we observe a similar pattern as the other datasets in the improvement of breakpoint accuracy with $HQ3$ by 18.66% for calls that have a difference in breakpoint greater than 50, and improvement in SV length similarity by 19.76% for calls with similarity less than 0.95 (Figure 4.11a-b).

SV calls from HG002 reads alignment to GRCh37 are benchmarked against the truth set generated by comparing HG002 haplotype-resolved assembly to GRCh37. Minimap2 has precision 0.78, recall 0.76, and F1 score 0.77 while $HQ3$ has precision 0.79, recall 0.75, and F1 score 0.77. Out of 16462 true positive calls in $HQ3$, 703 (4.27%) are made only in $HQ3$

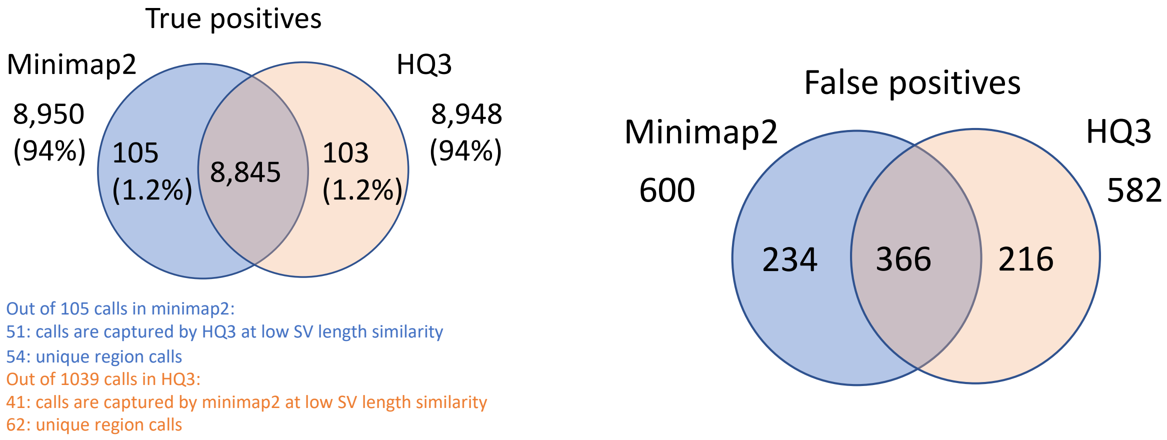
with SV length similarity to the truth set greater than 0.7 (default parameter in Truvari). However, 376/703 calls that are captured by minimap2 with SV length similarity less than 0.7 and 327/703 calls that are uniquely made by $HQ3$. Likewise, out of 16620 true positive calls in minimap2, 861 (5.18%) are made only in minimap2 with SV length similarity greater than 0.7. However, 524/861 are captured by $HQ3$ with SV length similarity less than 0.7, and 337/861 are uniquely made by minimap2. A fine-grain analysis of the common true positive calls by minimap2 and $HQ3$ in Figure 4.17a, shows that a major density of SV calls (81.85%) have a difference in breakpoint below 50 in both minimap2 and $HQ3$, and minimap2 has marginally better performance in terms of a lower difference in breakpoint of SVs that have a value below 50. Whereas, for a large difference in the SV breakpoint (greater than 50), $HQ3$ is better in terms of the breakpoint accuracy of the SV calls (on average across all SV calls). Therefore, $HQ3$ improves the SV breakpoint for the rest 18.15% calls that have high differences in breakpoints. Further, Figure 4.17b demonstrates that $HQ3$ has better SV length similarity when the length similarity is below 0.95 which corresponds to 21.82% calls.

Table 4.6: The evaluation of SV detection in terms of Recall, Precision and F1 score for different number of extended base pairs on the target region of interest. The number of bases extended on the target region in HQAlign is defined as $(1 - f + b) *$ read length, where f is the fraction of read aligned in the initial alignment and b is the buffer fraction.

buffer fraction	**Ground truth SV set**	**Recall**	**Precision**	**F1 score**
0.25	GIAB Tier 1	0.94	0.94	0.94
	HG002 assembly	0.75	0.79	0.77
0.1	GIAB Tier 1	0.94	0.94	0.94
	HG002 assembly	0.74	0.78	0.76
0.3	GIAB Tier 1	0.94	0.94	0.94
	HG002 assembly	0.75	0.79	0.77
0.4	GIAB Tier 1	0.94	0.94	0.94
	HG002 assembly	0.74	0.78	0.76

Table 4.7: Comparison for precision, recall and and F1 score for SV calls made by *HQ*3, minimap2, and the Union model.

Dataset	**Truth set**	**Method of alignment**	**Precision**	**Recall**	**F1 score**
HG002 reads to GRCh37	GIAB Tier 1 calls	minimap2	0.94	0.94	0.94
		*HQ*3	0.94	0.94	0.94
		Union	0.93	0.95	0.94
HG002 reads to CHM13	HG002 assembly to CHM13 (including centromere calls)	minimap2	0.77	0.57	0.66
		*HQ*3	0.75	0.58	0.65
		Union	0.71	0.61	0.66
HG002 reads to CHM13	HG002 assembly to CHM13 (excluding centromere calls)	minimap2	0.82	0.75	0.78
		*HQ*3	0.82	0.74	0.78
		Union	0.78	0.79	0.78
HG002 reads to GRCh37	comparing HG002 assembly to GRCh37	minimap2	0.78	0.76	0.77
		*HQ*3	0.79	0.74	0.77
		Union	0.74	0.79	0.77
Simulated reads to CHM13	Simulated SVs on chr 8 and X	minimap2	0.99	0.97	0.98
		*HQ*3	0.99	0.97	0.98
		Union	0.99	0.98	0.98



Out of 105 calls in minimap2:
51: calls are captured by HQ3 at low SV length similarity
54: unique region calls
Out of 1039 calls in HQ3:
41: calls are captured by minimap2 at low SV length similarity
62: unique region calls

(a)

(b)

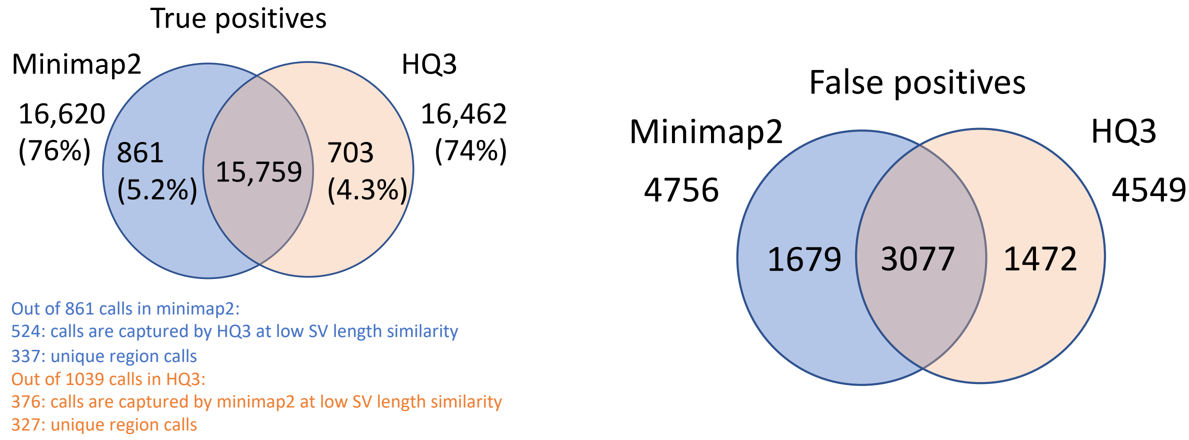
Figure 4.12: **Comparison of SV calls from *HQ*3 and minimap2 with Genome in a Bottle (GIAB) Tier 1 truth set for GRCh37 build.** (a) Comparison of true positive calls. (b) Comparison of false positive calls.

Table 4.8: Comparison for precision, recall and and F1 score for SV calls made by HQ3, minimap2 (with parameter tuning), and the Union model. The computation time is the wall clock time using 45 CPU cores for each method.

Ground Truth (Total SV calls in the truth set)	**Method of alignment**	**Computation time (in sec)**	**Precision**	**Recall**	**F1 score**
GIAB Tier 1 calls (9,641)	minimap2 (default: k=15)	24,479	0.94	0.94	0.94
	HQ3	50,069	0.94	0.94	0.94
	minimap2 (k=14)	67,315	0.94	0.94	0.94
	HACGT	50,443	0.94	0.87	0.91
	Union (mm2 (k=15) & HQ3)		0.93	**0.95**	0.94
	Union (mm2 (k=15) & mm2(k=14))		0.93	0.94	0.94
	Union (mm2 (k=15) & HACGT)		0.93	0.94	0.94
comparing HG002 assembly to GRCh37 (23,781)	minimap2 (default: k=15)	24,479	0.78	0.76	0.77
	HQ3	50,069	0.79	0.75	0.77
	minimap2 (k=14)	67,315	0.78	0.75	0.77
	HACGT	50,443	0.81	0.69	0.74
	Union (mm2 (k=15) & HQ3)		0.75	**0.79**	0.77
	Union (mm2 (k=15) & mm2(k=14))		0.77	0.76	0.77
	Union (mm2 (k=15) & HACGT)		0.78	0.76	0.77

4.4 Conclusion

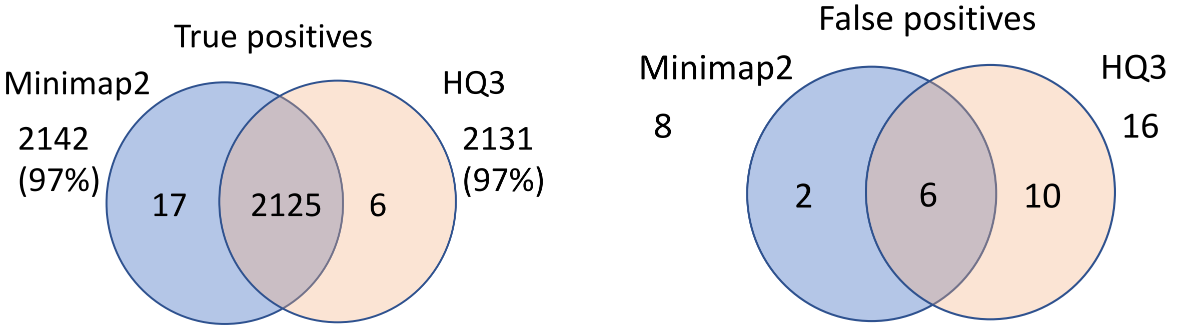
In this chapter, we studied HQAlign as an alignment algorithm designed for the detection of structural variants for nanopore sequencing reads. HQAlign provides alignment that outperforms the recent minimap2 aligner in terms of the accuracy and quality of the alignments. The SV calling from HQAlign is also at par with minimap2 in terms of F1 score and it outperforms minimap2 SV calls in terms of the quality of SVs measured in breakpoint accu-



Out of 861 calls in minimap2:
524: calls are captured by HQ3 at low SV length similarity
337: unique region calls
Out of 1039 calls in HQ3:
376: calls are captured by minimap2 at low SV length similarity
327: unique region calls

(a) (b)

Figure 4.13: **Comparison of SV calls from** $HQ3$ **and minimap2 with HG002-to-GRCh37 dipcall as truth set.** (a) Comparison of true positive calls. (b) Comparison of false positive calls.

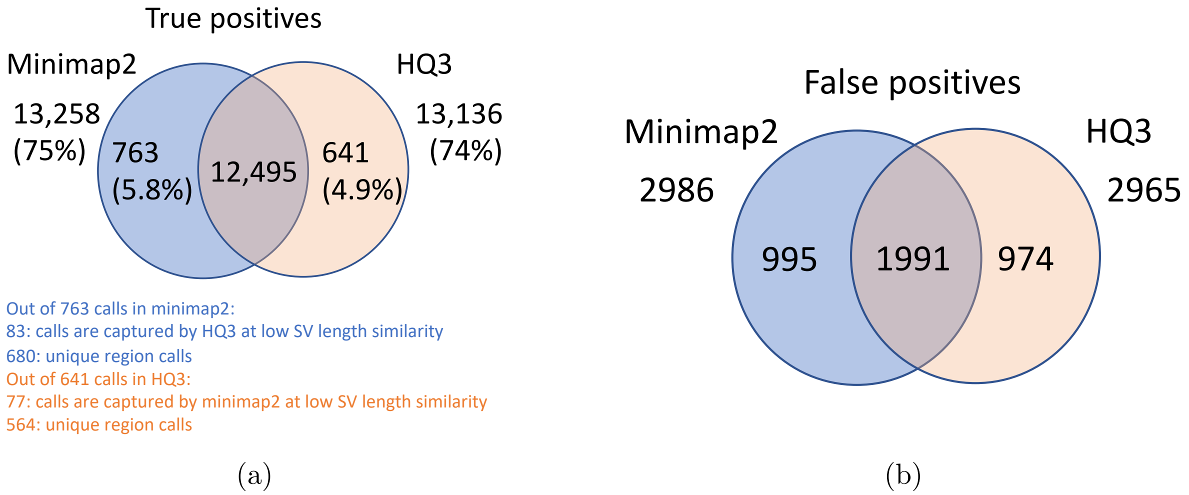


(a) (b)

Figure 4.14: **Comparison of SV calls from** $HQ3$ **and minimap2 with simulated data.** (a) Comparison of true positive calls. (b) Comparison of false positive calls.

racy and SV length similarity. Moreover, there are many complementary SVs captured by HQAlign that are missed by minimap2 alignments.

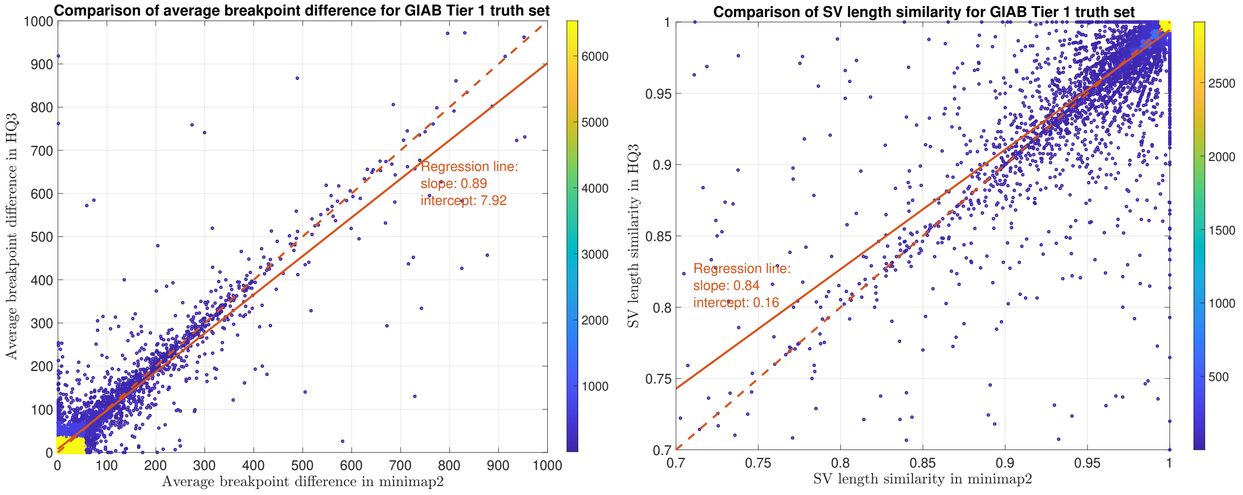
The reason for this improvement in the performance of alignment and SV calling with HQAlign is that it takes into account the underlying physics of nanopore sequencer through the Q-mer map, which could be one of the major causes of the high error rates in nanopore



Out of 763 calls in minimap2:
83: calls are captured by HQ3 at low SV length similarity
680: unique region calls
Out of 641 calls in HQ3:
77: calls are captured by minimap2 at low SV length similarity
564: unique region calls

Figure 4.15: **Comparison of SV calls from $HQ3$ and minimap2 with HG002-to-CHM13 dipcall as truth set and excluding the calls from the centromere regions.** (a) Comparison of true positive calls. (b) Comparison of false positive calls.

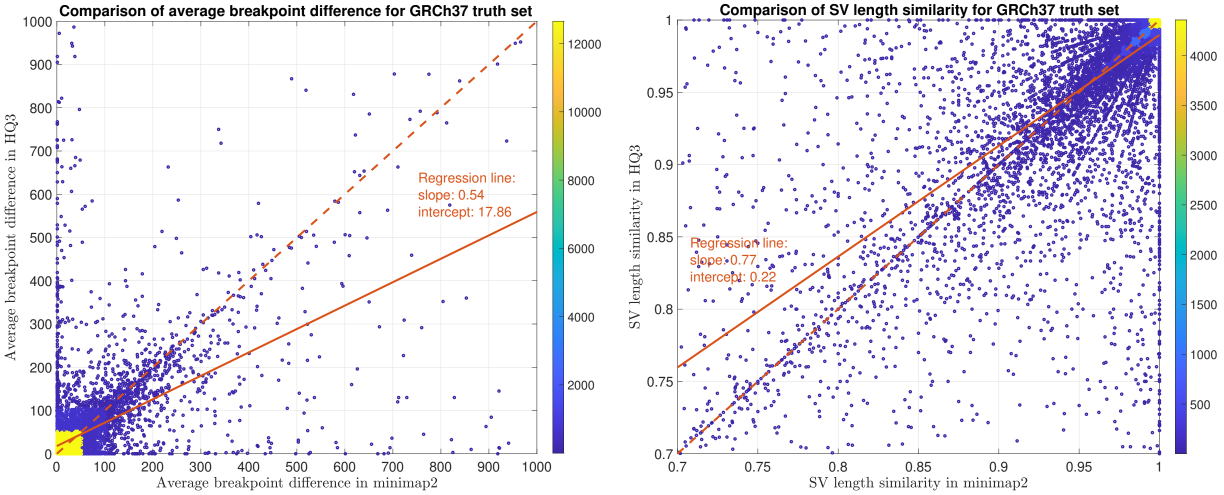
sequencing, and also it focuses on a narrow region of the genome (where the read aligns in nucleotide domain) for alignment with quantized sequences. Further, this pipeline is adapted specifically for the detection of SVs. We demonstrated how HQAlign utilizes the bias of Q-mer map without accessing the raw current signal of nanopore sequencer by translating the basecalled nucleotide sequences to quantized current level (of finite alphabet size) sequences. This improvement helps in detecting several SVs that are missed by minimap2 due to high error rates in the nanopore reads. Further, the recall rate for SV detection can be improved by combining the complementary calls from both $HQ3$ and minimap2 in the union model at the same F1 score.



(a)

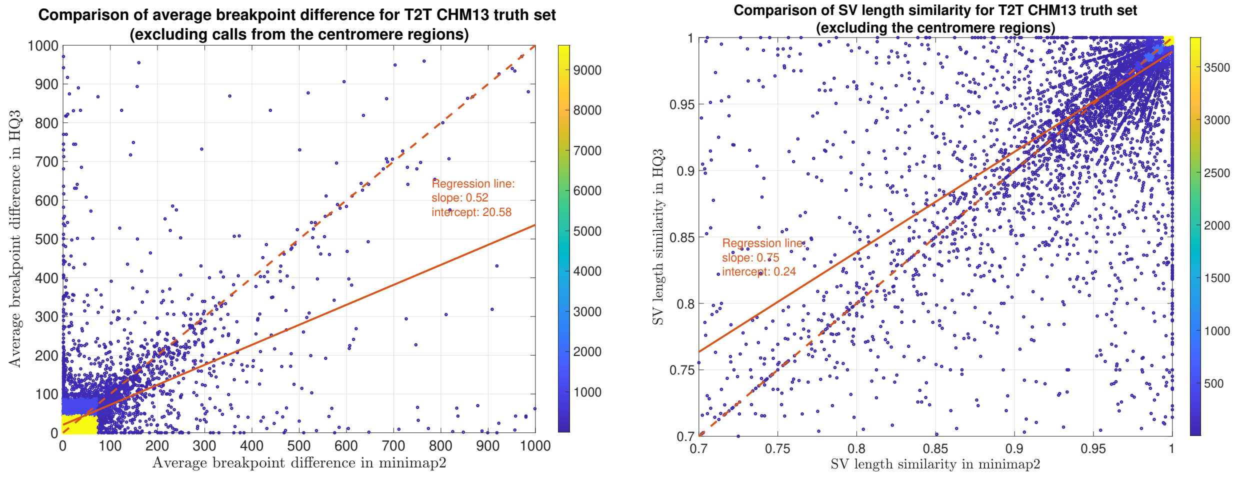
(b)

Figure 4.16: **SV quality comparison for common true positive calls in $HQ3$ and minimap2 against GIAB Tier 1 truth set.** (a) Comparison of SV breakpoint accuracy in $HQ3$ and minimap2 for common true positive calls. The difference of SV breakpoint is compared to the GIAB Tier 1 truth set. A smaller difference represents better breakpoint accuracy. Therefore, slope of the regression line < 1 represents better accuracy of $HQ3$ than minimap2 on average. (b) Comparison of SV length similarity in $HQ3$ and minimap2 for common true positive calls. The slope of the regression line < 1 represents better SV length in minimap2 than $HQ3$ on average. However, this is due to a large density of SVs with length similarity ≥ 0.95 in both minimap2 and $HQ3$. For length similarity less than 0.95, $HQ3$ has better performance than minimap2.



(a) (b)

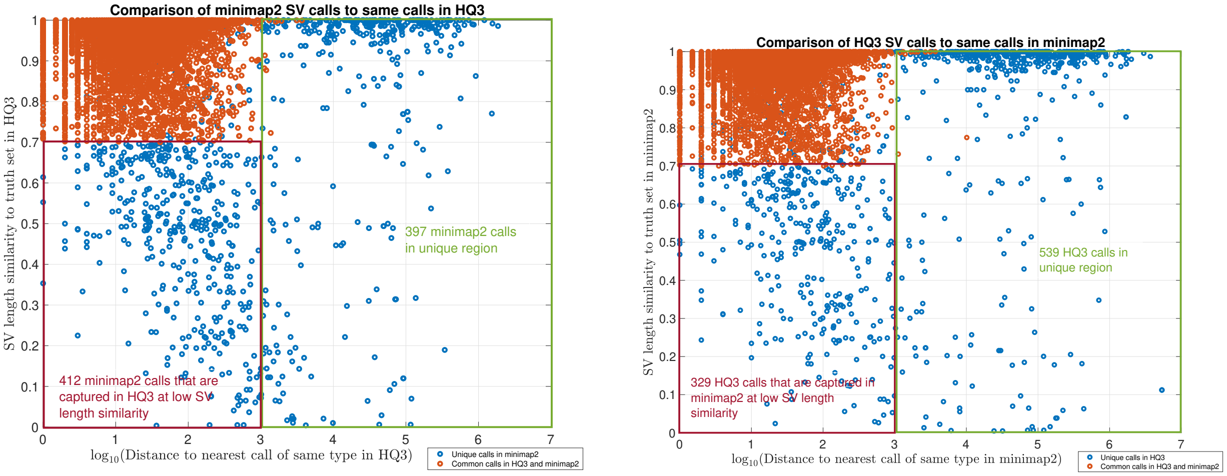
Figure 4.17: **SV quality comparison for common true positive calls in *HQ*3 and minimap2 against HG002-to-GRCh37 dipcall truth set.** (a) Comparison of SV breakpoint accuracy in *HQ*3 and minimap2 for common true positive calls. The difference of SV breakpoint is compared to the truth set generated from comparing HG002 haplotype-resolved assembly to GRCh37 build. A smaller difference represents better breakpoint accuracy. Therefore, slope of the regression line < 1 represents better accuracy of *HQ*3 than minimap2 on average. (b) Comparison of SV length similarity in *HQ*3 and minimap2 for common true positive calls. The slope of the regression line < 1 represents better SV length in minimap2 than *HQ*3 on average. However, this is due to a large density of SVs with length similarity ≥ 0.95 in both minimap2 and *HQ*3. For length similarity less than 0.95, *HQ*3 has better performance than minimap2.



(a)

(b)

Figure 4.18: **SV quality comparison for common true positive calls in $HQ3$ and minimap2 against HG002-to-CHM13 dipcall truth set (excluding the centromere region).** (a) Comparison of SV breakpoint accuracy in $HQ3$ and minimap2 for common true positive calls. The difference of SV breakpoint is compared to the truth set generated from comparing HG002 haplotype-resolved assembly to T2T CHM13 build and the SV calls in the centromere region are excluded in this truth set. A smaller difference represents better breakpoint accuracy. Therefore, slope of the regression line < 1 represents better accuracy of $HQ3$ than minimap2 on average. (b) Comparison of SV length similarity in $HQ3$ and minimap2 for common true positive calls. The slope of the regression line < 1 represents better SV length in minimap2 than $HQ3$ on average. However, this is due to a large density of SVs with length similarity ≥ 0.95 in both minimap2 and $HQ3$. For length similarity less than 0.95, $HQ3$ has better performance than minimap2.



(a) (b)

Figure 4.19: **Comparison of SV calls made by minimap2 and $HQ3$ to other method.** (a) For the complementary calls (in blue) and common calls (in red) made by minimap2, this figure compares SV length similarity and distance to nearest SV in $HQ3$ of the same type. 397 complementary calls made by minimap2 are in unique region, whereas 412 complementary calls in minimap2 are captured in neighboring region (within 1000 bp) in $HQ3$ but with a low SV length similarity. (b) For the complementary calls (in blue) and common calls (in red) made by $HQ3$, this figure compares SV length similarity and distance to nearest SV in minimap2 of the same type. 539 complementary calls made by $HQ3$ are in unique region, whereas 329 complementary calls in $HQ3$ are captured in neighboring region (within 1000 bp) in minimap2 but with a low SV length similarity.

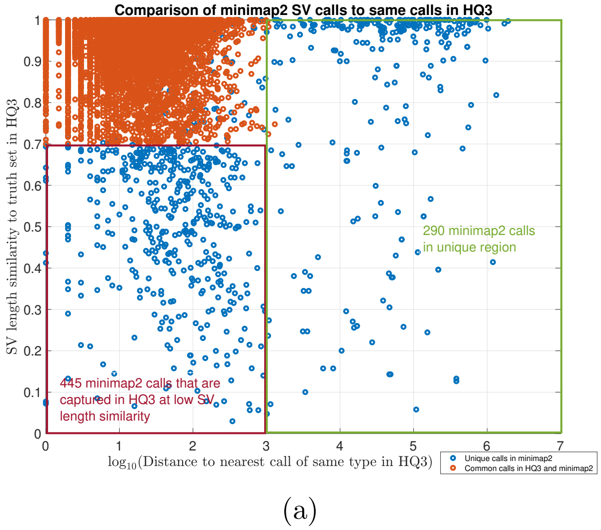


Figure 4.20: **Comparison of SV calls to HG002 truth set.** (a) For the complementary calls (in blue) and common calls (in red) made by minimap2, this figure compares SV length similarity and distance to nearest SV in $HQ3$ of the same type. 290 complementary calls made by minimap2 are in unique region, whereas 445 complementary calls in minimap2 are captured in neighboring region (within 1000 bp) in $HQ3$ but with a low SV length similarity. (b) For the complementary calls (in blue) and common calls (in red) made by $HQ3$, this figure compares SV length similarity and distance to nearest SV in minimap2 of the same type. 179 complementary calls made by $HQ3$ are in unique region, whereas 347 complementary calls in $HQ3$ are captured in neighboring region (within 1000 bp) in minimap2 but with a low SV length similarity.

4.5 Appendix

4.5.1 Accessing HQAlign on github

HQAlign requires python 3, and the installation guideline can be found on github.
The software is available at: `https://github.com/joshidhaivat/HQAlign.git`

```
usage:  python hqalign.py [-h] -r REF -i READS -o OUTPUT [-t THREADS] [-k
KMER]
```

```
arguments:
 -h, --help                        show this help message and exit
 -r REF, --ref REF                 reference genome filename in fasta format
 -i READS, --reads READS           directory location of read files in fasta format (with
                                   file extension .fasta)
 -o OUTPUT, --output OUTPUT        location of directory of output files
 -t THREADS, --threads THREADS     maximum number of parallel threads (default=4)
 -k KMER, --kmer KMER              minimizer length for hybrid step (default=18)
```

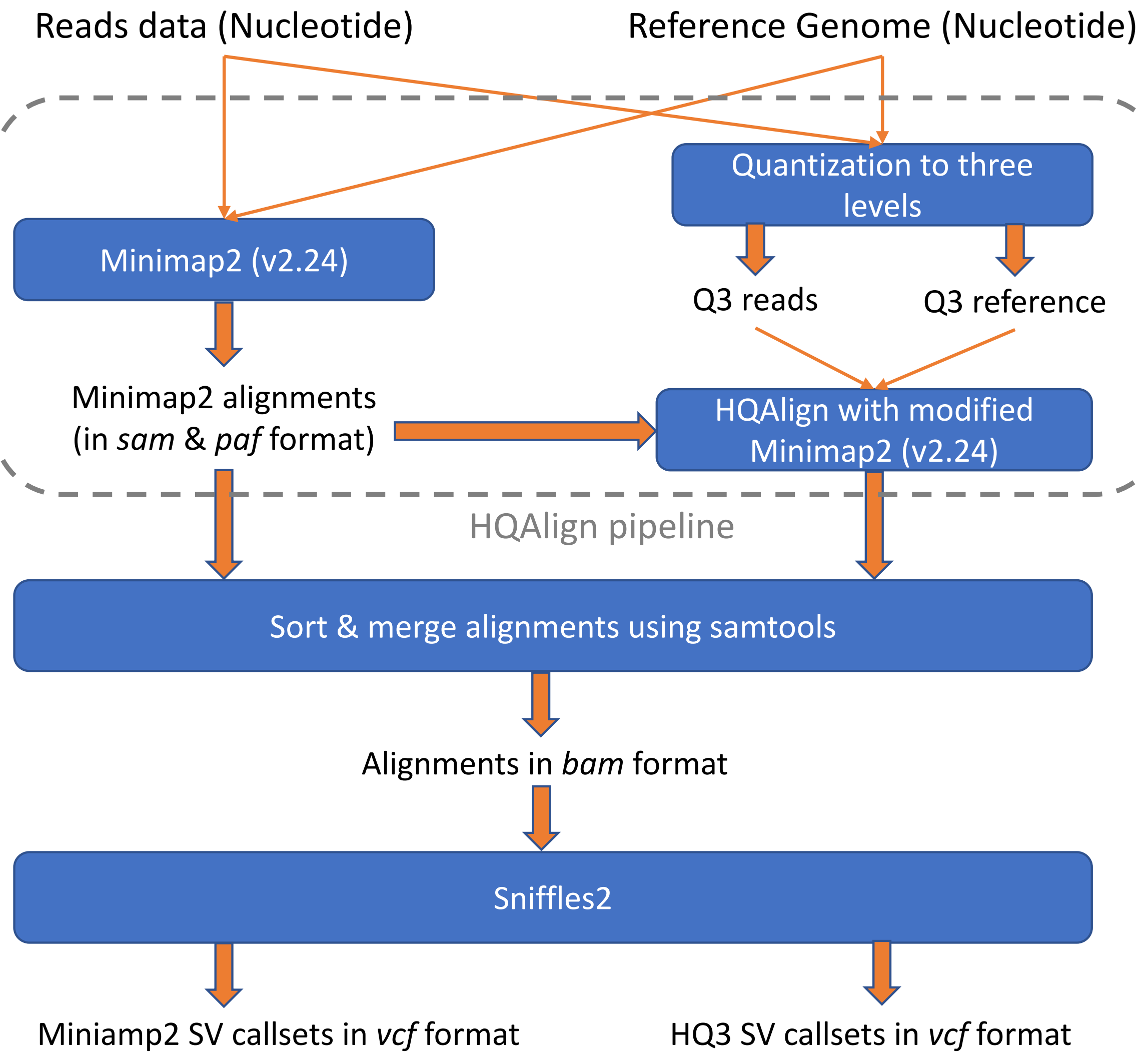

Reads data (Nucleotide)
Reference Genome (Nucleotide)
Quantization to three levels
Minimap2 (v2.24)
Q3 reads
Q3 reference
Minimap2 alignments (in *sam* & *paf* format)
HQAlign with modified Minimap2 (v2.24)
HQAlign pipeline
Sort & merge alignments using samtools
Alignments in *bam* format
Sniffles2
Miniamp2 SV callsets in *vcf* format
HQ3 SV callsets in *vcf* format

Figure 4.21: Complete pipeline for SV calling using minimap2 and HQAlign.

CHAPTER 5

Heterozygous variant detection for haplotype-resolved assembly using long nanopore reads

5.1 Introduction

De novo genome assembly is a crucial step in understanding the genetic makeup of an organism. By piecing the sequenced reads together, the entire genome sequence can be reconstructed. However, for diploid organisms, which have two copies of each chromosome with variations in one or both copies w.r.t. the reference genome, traditional assembly methods struggle to differentiate between the two haplotypes (versions) of each chromosome. This can lead to a loss of valuable genetic information as shown in Fig. 5.1 (a).

Nanopore sequencing technology offers the ability to generate long reads, which can be beneficial for genome assembly. However, these reads also come with a higher error rate compared to other sequencing methods. The high error rates pose a challenge for haplotype resolved assembly, where the goal is to reconstruct both haplotypes of a diploid genome. Current approaches to haplotype resolved assembly with nanopore reads often rely on additional sequencing data, such as highly accurate short reads from Illumina, highly accurate long HiFi reads data or data from parental genomes. While this can improve assembly accuracy, it also increases the cost and complexity of the process. The standard approach of these methods is to label each read to one of the two haplotypes for a diploid genome using the highly accurate short reads data or long HiFi reads data and then assemble reads labeled to a single haplotype to produce haplotype-resolved assembly as shown in Fig. 5.1

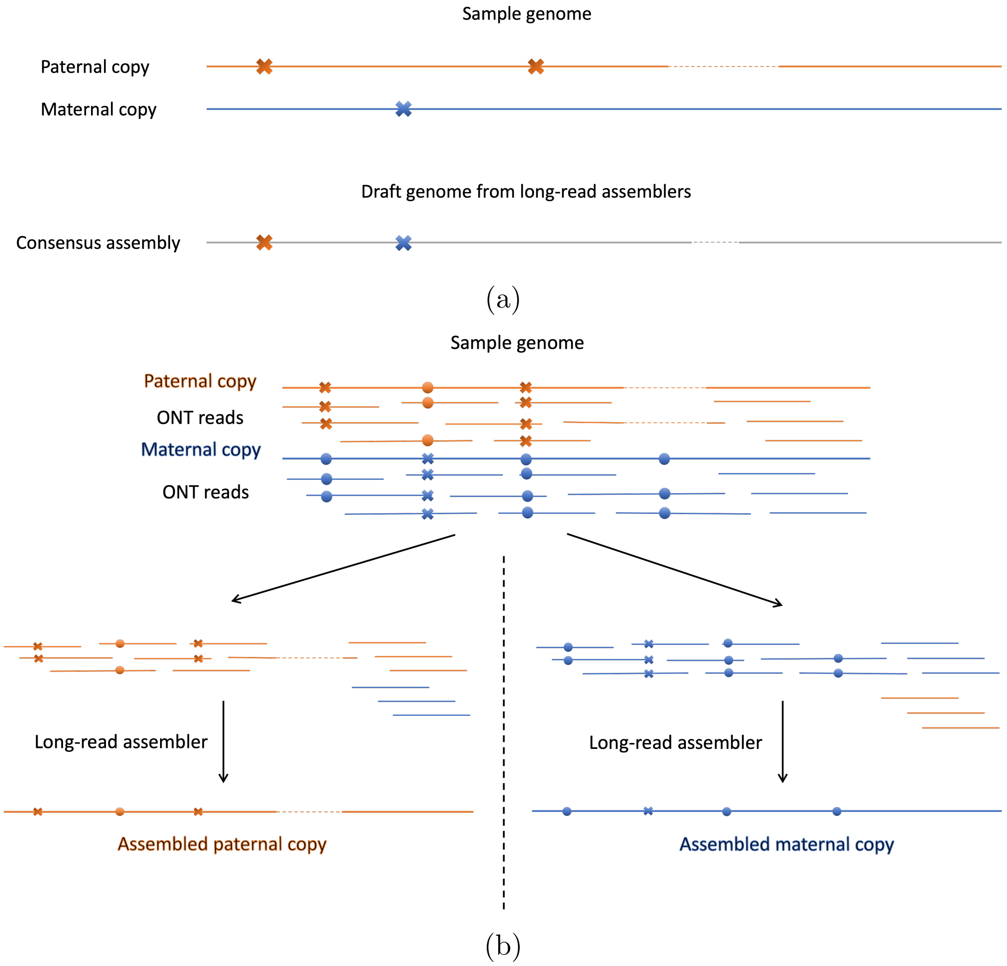


Figure 5.1: (a) The sequencing reads data from sample diploid genome (with two haplotypes represented in red and blue) is assembled using long-read assemblers such as Flye and the resulting draft genome represents a consensus assembly with loss of genetic information. (b) High level idea for haplotype-resolved genome assembly for diploid genome where each read is phased and labeled to the haplotype it is sequenced from and then assembled using the long-read assemblers to maintain each copy of the genome without any loss of genetic information. The cross and the circles represent heterozygous variants specific to each haplotype.

(b). In these approaches, phasing each read to its correct haplotype is the bottleneck for the accuracy of the haploid assembly, and therefore, highly accurate short reads from Illumina or HiFi reads are used in addition to long nanopore reads to clearly make a distinction in read errors and variations in the sequence specific to one of the two haplotypes also known as heterozygous variants. Moreover, utilizing highly accurate HiFi reads data ($< 1\%$ error rate) only for accurate identification of heterozygous variants leads to accurate haplotype contigs post assembly, but the lower average read length of the HiFi reads (around 10-25 kb) leads to lower contiguity of the assembled contigs as compared to the assembly from long nanopore reads (with the longest read about 4Mb). Therefore, it is useful to develop assemblers that can take advantage of the long but noisy nanopore reads for higher contiguity of the assembly and accurately identify the heterozygous variants despite the high error rates of the nanopore reads.

This thesis chapter focuses on the primitive step of the haplotype-resolved genome assembly which is developing a method for identifying heterozygous variants from the nanopore sequencing reads data accurately.

5.2 Algorithm for detection of heterozygous variants

The high-level idea of the algorithm to detect the heterozygous variants from the read data is shown in Fig. 5.2. Given only the long and noisy nanopore reads from the entire diploid genome, the algorithm determines the regions on the reads that overlap heterozygous alleles which are "signature" of the haplotypes without using any side information such as the reference genome or a draft assembly from the reads data. The details of the model and algorithm developed to detect the heterozygous variants are discussed in this section.

Let λ be the average coverage depth of the reads data, where coverage depth is defined as number of times a nucleotide is read during the sequencing. Therefore, a heterozygous allele has a coverage depth of $\lambda/2$. This is observed in the Fig. 5.3 which represents the

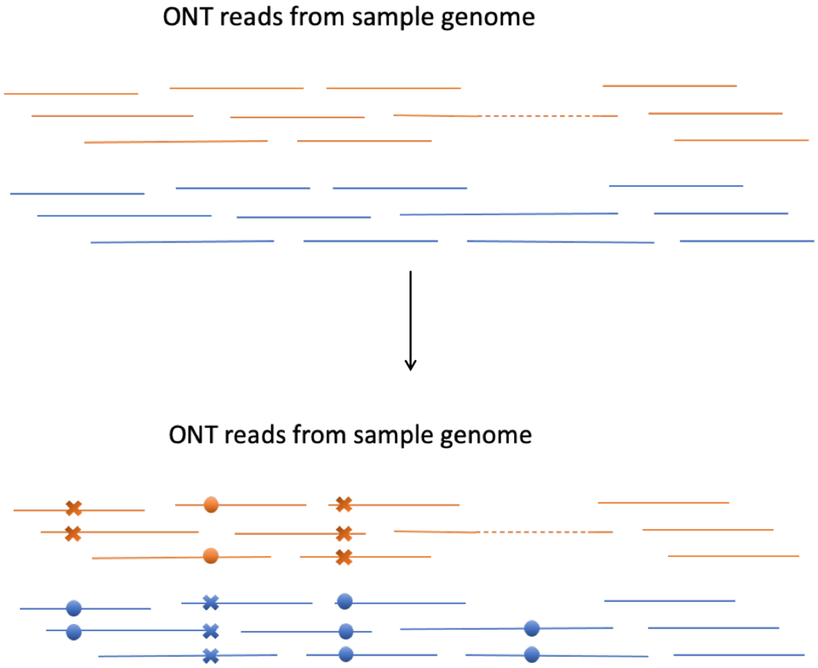


Figure 5.2: A high-level idea of the algorithm to detect the heterozygous variants from the long noisy nanopore reads data.

k-mer frequency or the number of times a k consecutive nucleotide sequence occurs across the R10.4 nanopore reads data for HG00733 sample diploid genome. The reads data have an average coverage depth around $\lambda = 40$ which is observed as a peak in k-mer frequency at 40. These k-mers represent sequence from autozygous regions between two haplotypes, while the small bump around $\lambda/2 = 20$ in the figure represents k-mers overlapping heterozygous alleles. This motivates the development of a Hidden Markov Model (HMM) for predict the regions on the reads likely to be overlapping the heterozygous alleles given the sequence of k-mer frequency through the read computed using jellyfish[1] [3].

A nucleotide read sequence of length n is represented as $S_i = s_{i,1}, s_{i,2}, \ldots, s_{i,n}$, where $s_{i,j} \in \{A, C, G, T\}$. A k-mer from this sequence is a subsequence of k consecutive nucleotide, e.g., $s^k_{i,j} = s_{i,j}, s_{i,j+1}, \ldots, s_{i,j+k-1}$ of length k. Therefore, there are $n - k + 1$ k-mers from

[1] Jellyfish is a fast and memory efficient algorithm that creates a database of kmer frequency across the whole genome sequencing data and can also return the frequency of a query kmer.

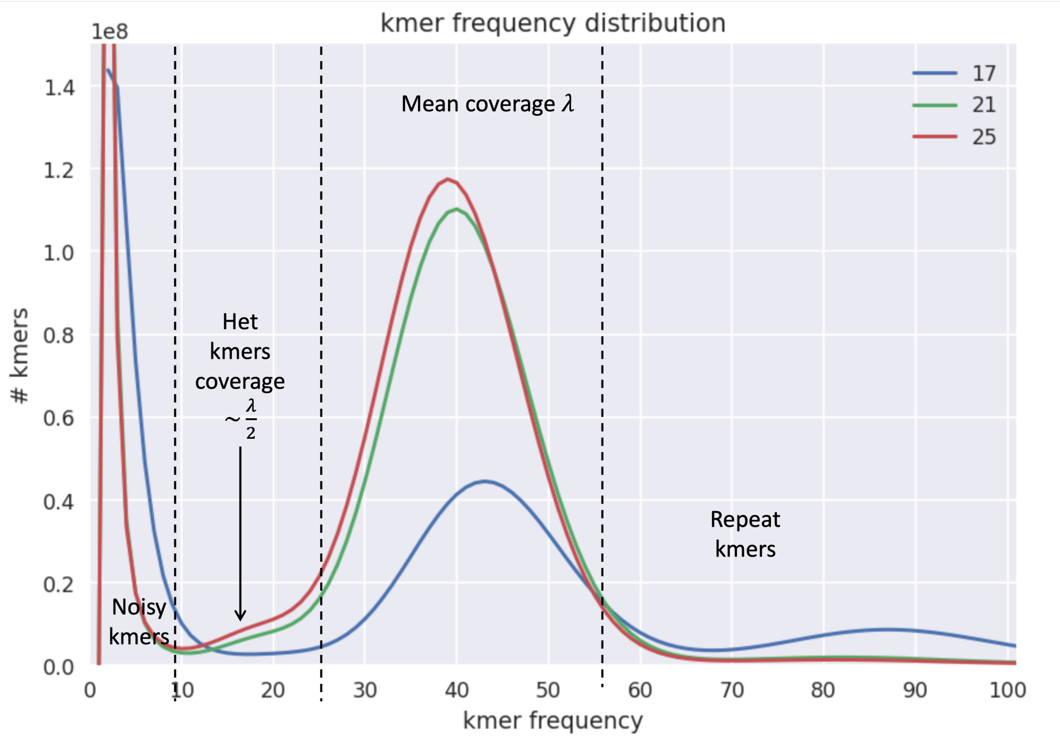


Figure 5.3: Distribution of k-mer frequency from R10.4 nanopore reads data for HG00733 diploid genome sample (for $k = 17, 21, 25$). The k-mer frequency from the entire data is computed using jellyfish [3]. The peak around low kmer frequency (< 5) represent the kmers from noisy regions of the read. The major peak around 40 represents the average coverage of the nanopore reads data.

the read S_i of length n. The number of occurrence of each k-mer encountered in the reads data from the entire genome is computed and stored in the database by jellyfish algorithm. The sequence of k-mer frequency for read S_i is represented as $X_i = x_{i,1}, x_{i,2}, \ldots, x_{i,n-k+1}$, where $x_{i,j} \in \mathbb{N}$ is the k-mer frequency of $s^k_{i,j}$. The sequence of k-mer frequency X_i for each read i is considered to be observed based on the hidden underlying state sequence $Z_i = z_{i,1}, z_{i,2}, \ldots, z_{i,n-k+1}$ of the model which captures if the k-mer overlaps a region of sequencing noise, heterozygous allele, autozygous region or a low-count or high-count repeat region on the genome, therefore, $z_{i,j} \in \{0, 1, 2, 3, 4\}$ represent the mentioned hidden states, respectively. The developed HMM shown in Fig 5.4 predicts the hidden state sequence Z_i given only the observed sequence of k-mer frequency X_i for all reads. The model parameters are initial hidden state probability π, transition probability A, and emission probability B, where $\mathbb{P}[\cdot]$ is the probability measure of an event, are learned using expectation-maximization (EM) algorithm.

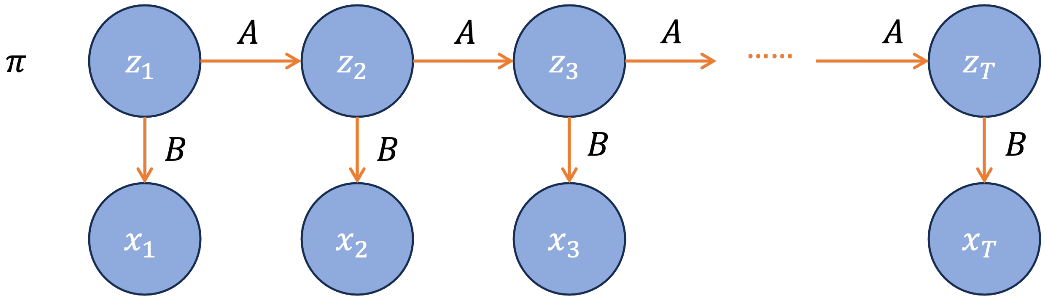


Figure 5.4: Hidden Markov Model to predict heterozygous regions on the reads with $x_1, x_2, \ldots, x_T$ as observed k-mer frequency and $z_1, z_2, \ldots, z_T$ as the hidden states.

5.2.1 Notation of HMM parameters

Let $\mathbf{X} = [x_1, \ldots, x_T]$ where $x_t \in \mathbb{N}$ be the sequence of observed k-mer frequency and $\mathbf{Z} = [z_1, \ldots, z_T]$ where $z_t \in \{0, 1, 2, 3, 4\}$ be the sequence of hidden states. The model parameters are $\boldsymbol{\theta} = (\boldsymbol{\pi}, \mathbf{A}, \mathbf{B})$ where $\boldsymbol{\pi}$ be the initial hidden state probability with $\pi_j = \mathbb{P}[z_1 = j]$ is the probability of initial hidden state being j; $\mathbf{A}$ be the transition probability matrix where an entry $A_{i,j} = \mathbb{P}[z_t = j | z_{t-1} = i]$ representing the probability of transiting to hidden state j from hidden state i; and $\mathbf{B}$ be the emission probability where $B_j(x_t) = \mathbb{P}[x_t | z_t = j]$ is the probability of emitting x_t at time t when the hidden state is j which is modeled as gaussian distribution with mean μ_j and variance σ_j^2, *i.e.*, $x_t | (z_t{=}j) \sim \mathcal{N}(\mu_j, \sigma_j^2)$.

5.2.2 Forward-backward algorithm

The forward algorithm is defined as $\alpha_i(t) = \mathbb{P}[x_1, \ldots, x_t, z_t = i]$ with $\alpha_i(1) = \pi_i B_i(x_1)$ and the dynamic programming algorithm is:

$$
\begin{aligned}
\alpha_i(t) &= \mathbb{P}[x_1, \ldots, x_t, z_t = i] \\
&= \sum_j \mathbb{P}[x_1, \ldots, x_t, z_{t-1} = j, z_t = i] \\
&= \sum_j \mathbb{P}[x_1, \ldots, x_{t-1}, z_{t-1} = j] \cdot \mathbb{P}[x_t, z_t = i | z_{t-1} = j] \\
&= \sum_j \alpha_j(t-1) \cdot \mathbb{P}[z_t = i | z_{t-1} = j] \cdot \mathbb{P}[x_t | z_t = i] \\
&= \sum_j \alpha_j(t-1) \cdot A_{j,i} \cdot B_i(x_t) \\
&= B_i(x_t) \cdot \sum_j \alpha_j(t-1) \cdot A_{j,i} \qquad (5.1)
\end{aligned}
$$

The backward algorithm is defined as $\beta_i(t) = \mathbb{P}[x_{t+1}, \ldots, x_T | z_t = i]$ with $\beta_i(T) = 1$ and the dynamic programming algorithm is:

$$
\begin{aligned}
\beta_i(t) &= \mathbb{P}[x_{t+1}, \ldots, x_T | z_t = i] \\
&= \sum_j \mathbb{P}[x_{t+1}, \ldots, x_T, z_{t+1} = j | z_t = i] \\
&= \sum_j \mathbb{P}[x_{t+2}, \ldots, x_T | z_{t+1} = j] \cdot \mathbb{P}[x_{t+1}, z_{t+1} = j | z_t = i] \\
&= \sum_j \beta_j(t+1) \cdot \mathbb{P}[z_{t+1} = j | z_t = i] \cdot \mathbb{P}[x_{t+1} | z_{t+1} = j] \\
&= \sum_j \beta_j(t+1) \cdot A_{i,j} \cdot B_j(x_{t+1}) \qquad (5.2)
\end{aligned}
$$

Further, we also define the following quantities that are useful for the EM algorithm:

$$\gamma_j(t) = \frac{\alpha_j(t)\beta_j(t)}{\sum_i \alpha_i(t)\beta_i(t)} = \mathbb{P}[z_t = j|x_1, x_2, \ldots, x_T] \tag{5.3}$$

$$\xi_{i,j}(t) = \frac{\alpha_i(t)A_{i,j}B_j(x_{t+1})\beta_j(t+1)}{\sum_l \sum_m \alpha_l(t)A_{l,m}B_m(x_{t+1})\beta_m(t+1)} = \mathbb{P}[z_t = i, z_{t+1} = j|x_1, x_2, \ldots, x_T] \tag{5.4}$$

5.2.3 EM algorithm to learn HMM parameters

The EM algorithm seeks to learn the maximum likelihood estimate (MLE) of the unknown model parameters by iteratively applying the following steps:

1. **Expectation step:** Define $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)})$ as the expected value of the log-likelihood function of $\boldsymbol{\theta}$ defined as $\mathcal{L}(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z}) = p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$ with respect to the current conditional distribution of hidden variable $\mathbf{Z}$ given the observations $\mathbf{X}$ and the estimate of the model parameters at the current time step s, $\boldsymbol{\theta}^{(s)}$:

   $$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) = \mathbb{E}_{\mathbf{Z}\sim p(\cdot|\mathbf{X},\boldsymbol{\theta}^{(s)})}\left[\log\left(p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})\right)\right] \tag{5.5}$$

   Now, $\log(p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})) = \log(\pi_{z_1}) + \sum_{t=1}^{T}\log(B_{z_t}(x_t)) + \sum_{t=2}^{T}\log(A_{z_{t-1},z_t})$ and $\mathcal{Z}$ represent a set of all possible hidden state sequence of length T, then

   $$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) = \sum_{\mathbf{Z}\in\mathcal{Z}} \mathbb{P}\left[\mathbf{Z}|\mathbf{X}; \boldsymbol{\theta}^{(s)}\right] \cdot \left[\log(\pi_{z_1}) + \sum_{t=1}^{T}\log(B_{z_t}(x_t)) + \sum_{t=2}^{T}\log(A_{z_{t-1},z_t})\right] \tag{5.6}$$

2. **Maximization step:** This step finds the optimal model parameters that maximizes the expected value $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)})$ in Eqn. 5.6:

$$
\begin{aligned}
\boldsymbol{\theta}^{(s+1)} &= \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{Z}\sim p(\cdot|\mathbf{X},\boldsymbol{\theta}^{(s)})}\left[\log\left(p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta})\right)\right] \\
&= \arg\max_{\boldsymbol{\theta}} \sum_{\mathbf{Z}\in\mathcal{Z}} \mathbb{P}\left[\mathbf{Z}|\mathbf{X};\boldsymbol{\theta}^{(s)}\right]\cdot\left[\log(\pi_{z_1}) + \sum_{t=1}^{T}\log(B_{z_t}(x_t)) + \sum_{t=2}^{T}\log(A_{z_{t-1},z_t})\right]
\end{aligned}
\tag{5.7}
$$

On maximizing the equation in Eqn. 5.7 w.r.t. the model parameters $\boldsymbol{\theta} = (\boldsymbol{\pi}, \mathbf{A}, \mathbf{B})$, we get:

$$
\begin{aligned}
\pi_j^{(s+1)} &= \mathbb{P}[z_1 = j|\mathbf{X};\boldsymbol{\theta}^{(s)}] \\
&= \gamma_j^{(s)}(1)
\end{aligned}
\tag{5.8}
$$

$$
\begin{aligned}
A_{i,j}^{(s+1)} &= \frac{\sum_{t=2}^{T}\mathbb{P}\left[z_{t-1}=i, z_t=j|\mathbf{X};\boldsymbol{\theta}^{(s)}\right]}{\sum_{t=2}^{T}\mathbb{P}\left[z_{t-1}=i|\mathbf{X};\boldsymbol{\theta}^{(s)}\right]} \\
&= \frac{\sum_{t=2}^{T}\xi_{i,j}^{(s)}(t-1)}{\sum_{t=2}^{T}\gamma_i^{(s)}(t-1)}
\end{aligned}
\tag{5.9}
$$

$$
\begin{aligned}
\mu_i^{(s+1)} &= \frac{\sum_{t=1}^{T} x_t\mathbb{P}[z_t=i|\mathbf{X};\boldsymbol{\theta}^{(s)}]}{\sum_{t=1}^{T}\mathbb{P}[z_t=i|\mathbf{X};\boldsymbol{\theta}^{(s)}]} \\
&= \frac{\sum_{t=1}^{T} x_t\cdot\gamma_i^{(s)}(t)}{\sum_{t=1}^{T}\gamma_i^{(s)}(t)}
\end{aligned}
\tag{5.10}
$$

$$
\begin{aligned}
(\sigma_i^2)^{(s+1)} &= \frac{\sum_{t=1}^{T}(x_t-\mu_i^{(s+1)})^2\mathbb{P}[z_t=i|\mathbf{X};\boldsymbol{\theta}^{(s)}]}{\sum_{t=1}^{T}\mathbb{P}[z_t=i|\mathbf{X};\boldsymbol{\theta}^{(s)}]} \\
&= \frac{\sum_{t=1}^{T}(x_t-\mu_i^{(s+1)})^2\gamma_i^{(s)}(t)}{\sum_{t=1}^{T}\gamma_i^{(s)}(t)}
\end{aligned}
\tag{5.11}
$$

Further, since there is an imbalance in the data for the numbers of k-mers from the autozygous region and the number of k-mers from heterozygous region, the mean of the gaussian distribution for emission probability for the heterozygous state is set to half of the updated mean if the gaussian distribution of the emission probability for the autozygous state instead of learning the parameter from the data itself from Eqn. 5.10, *i.e.*, $\mu^{(s+1)}_{\text{heterozygous}} = \mu^{(s+1)}_{\text{autozygous}}/2$.

5.2.4 Viterbi algorithm to predict heterozygous regions in the reads

The Viterbi algorithm is a dynamic programming algorithm that predicts the most likely sequence of the hidden states by computing the maximum a posteriori probability (MAP) estimate. Mathematically, it computes $V_i(t) = \max_{z_1,\dots,z_{t-1}} \mathbb{P}[x_1, x_2, \dots, x_t, z_1, z_2, \dots, z_{t-1}, z_t{=}i]$, and $V_i(t{+}1) = B_i(x_{t+1}) \cdot \max_j\{A_{j,i} V_j(t)\}$ which computes the maximum probability to reach hidden state i after t steps across all possible paths. The decoded hidden state sequence $\hat{\mathbf{Z}} = [\hat{z}_1, \dots, \hat{z}_T]$ is obtained by decoding the last hidden state first, *i.e*, $\hat{z}_T = \arg\max_i V_i(T)$ followed by backtracking the path that maximizes the joint probability in the previous steps.

5.2.5 Performance metrics

The ground truth of the heterozygous variants for the genome sample HG00733 is developed using multiple sequencing technologies such as highly accurate long HiFi reads and short Illumina reads to identify the heterozygous alleles and their position on the reference human genome. Using this ground truth variants, we have computed metrics such as Precision, Recall, and F1 score as metric to evaluate the performance of our algorithm to detect the heterozygous variants from the reads data only.

5.3 Results

5.3.1 EM algorithm to learn HMM parameters

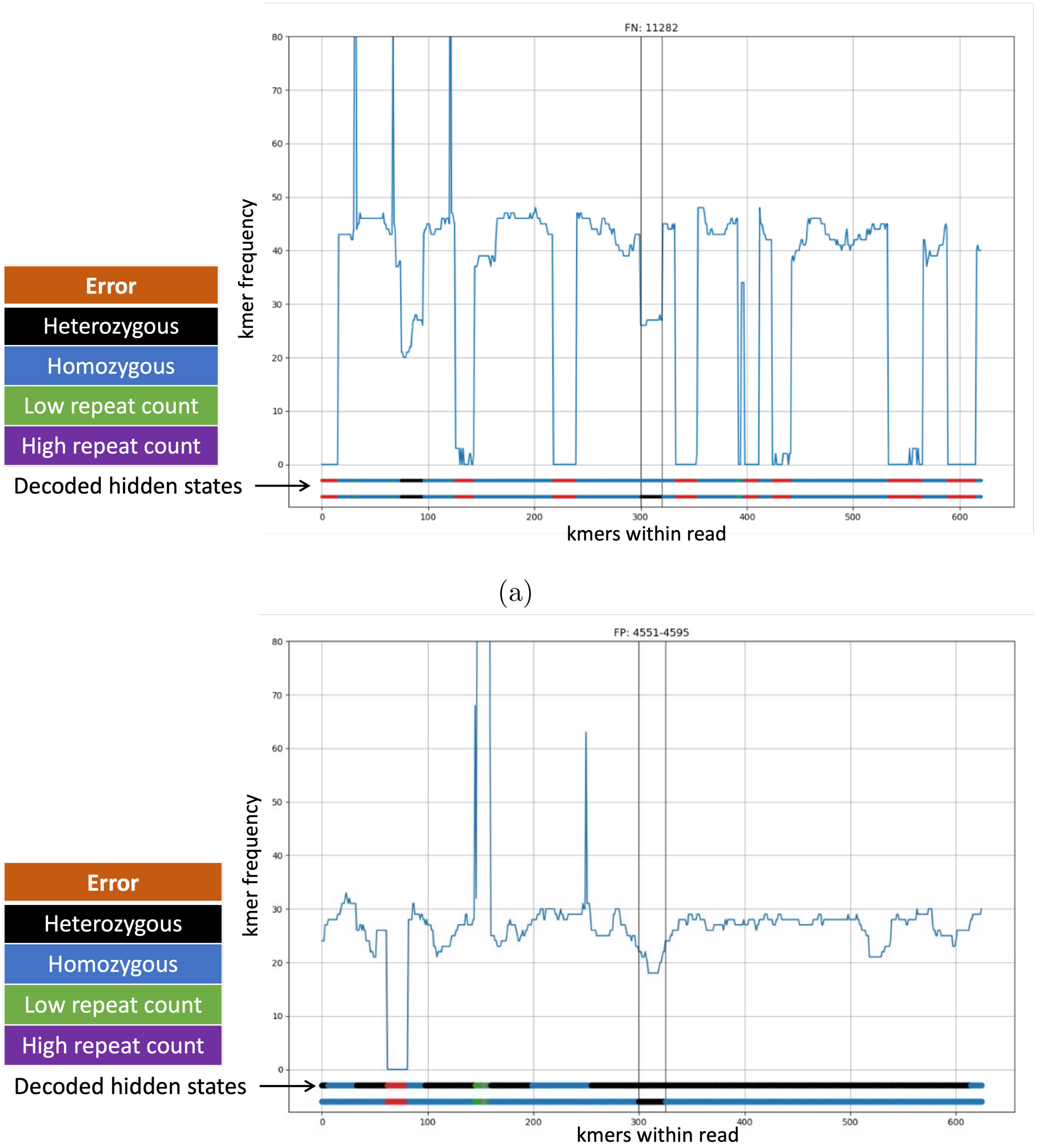
The model parameters for HMM are initialized randomly and learned using the EM algorithm in Eqn. 5.8–5.11. The initial probabilities π_i and the transition probabilities $A_{i,j}$ are initialized randomly from a uniform distribution. The mean μ_i of the emission gaussian distribution for the autozygous state is initialized to the coverage depth value from the reads data, the mean for the error state is initialized to 0, the mean for the repeat state and the variance of all hidden states are initialized to arbitrary high values.

Dataset: HMM parameters are trained iteratively using 5000 randomly sampled nanopore reads each of length at least 50k basepairs. The model parameters converges in 13 epochs where the convergence is defined as $||\boldsymbol{\theta}^{(s+1)} - \boldsymbol{\theta}^{(s)}|| < \epsilon$.

5.3.2 Detecting heterozygous variants on reads using Viterbi decoding

Given the observation of k-mer frequency as an input to the HMM with parameters trained using EM algorithm described in section 5.3.1, we decode the hidden state sequence using Viterbi algorithm described in section 5.2.4. This model has an underlying assumption of gaussian distribution for the emission probability and it estimates the mean μ_i for the distribution of the hidden state i which represents the average global coverage. However, the coverage depth is not constant throughout the genome. There are sections of the genome that has a higher local coverage than average coverage as well as the sections of the genome that has a lower local coverage than the average coverage.

Fig. 5.5 (a) shows a read region from a section of genome that has higher coverage than average. The heterozygous variant of interest is highlighted by vertical lines on the plot of k-mer frequency vs read index around the region of interest. The decoded hidden state sequence is represented at the bottom of the plot with color coding for each state in the legend key. The top line of decoded hidden states fails to decode the hidden state as heterozygous within the region of interest as the local coverage is higher than average. Therefore, leading to an increase in false negative calls in such regions. Similarly, Fig. 5.5 (b) shows a read region from a section that has lower coverage than average. It is evident from the decoded hidden state on the top that a long stretch is falsely identified as a heterozygous variant, therefore, leading to an increase in false positive calls in such regions.



(a)

(b)

Figure 5.5: k**-mer frequency vs read index plot around the interested heterozygous variant region on the read.** (a) A region on read with coverage depth higher than the average coverage. This leads to a false negative call after Viterbi decoding without taking the local coverage into account. (b) A region on read with coverage depth lower than the average coverage. This leads to a false positive calls after Viterbi decoding without taking the local coverage into account.

This can be resolved by computing the local coverage from a window of k-mer frequency that are emitted from heterozygous or autozygous hidden states and using this local coverage as the mean of the gaussian distribution for computing the emission probability in the second pass of Viterbi decoding as described in Eqn. 5.12, where x_i is the k-mer frequency, $z_i = 1$ is heterozygous hidden state, and $z_i = 2$ is autozygous hidden state. This two-pass Viterbi decoding algorithm resolves the higher false negative rates in the high coverage regions as well as higher false positive rates in the low coverage regions. The hidden state sequence after resolving with two-pass Viterbi decoding algorithm is represented in Fig. 5.5 in the bottom of the plot.

$$
\begin{aligned}
\lambda_{local} &= \frac{1}{\sum_{i-300}^{i+300} \mathbb{I}\{z_i = 1 \text{ or } z_i = 2\}} \sum_{i-300}^{i+300} x_i \mathbb{I}\{z_i = 1 \text{ or } z_i = 2\} \\
B_1(x_t) &= \mathbb{P}\left[x_t | z_t = 1\right] \sim \mathcal{N}\left(\frac{\lambda_{local}}{2}, \sigma_1^2\right) \\
B_2(x_t) &= \mathbb{P}\left[x_t | z_t = 2\right] \sim \mathcal{N}\left(\lambda_{local}, \sigma_2^2\right)
\end{aligned}
\tag{5.12}
$$

5.3.3 Evaluating heterozygous variants on the reads data

The decoded heterozygous states from two-pass Viterbi decoding algorithm above are used to identify the region on read overlapping the heterozygous variant. For a given choice of k, there are at most k consecutive k-mers that overlap a heterozygous SNP. If all of these k consecutive k-mers are correctly decoded as heterozygous hidden state from HMM, we define a confidence of het call from HMM as 1. Similarly, the confidence is computed for all heterozygous states decoded by HMM as

$$
\text{confidence} = \frac{\text{number of het states in a window of length } k}{k} \tag{5.13}
$$

Fig. 5.6 shows the trade-off between Precision, Recall and F1 score for different choice of

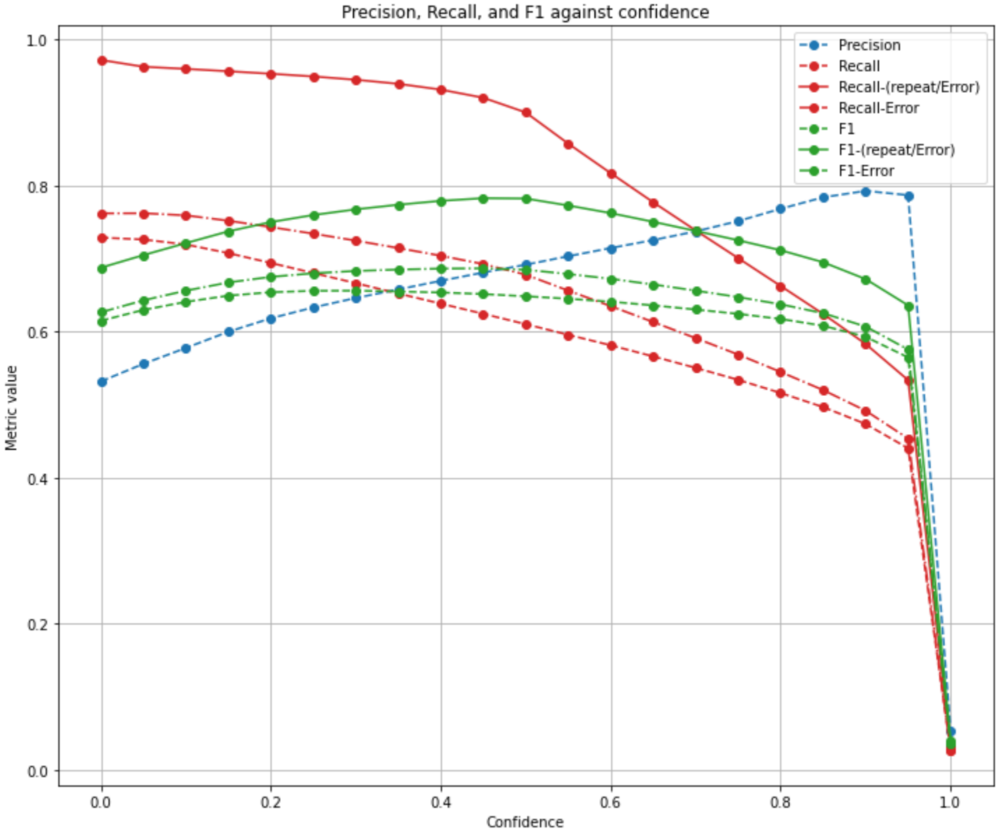


Figure 5.6: Precision, Recall and F1 score for different choice of minimum confidence level for heterozygous calls made by HMM

minimum confidence level for heterozygous calls made by HMM. There are many heterozygous variants loci on reads which are dominated by sequencing errors, therefore, making it impossible for the HMM to identify heterozygous state from those loci since the k-mer frequency is too low. On removing such regions from the analysis of the recall rate, the updated metric for recall and F1 score without error overlapping het regions are shown as in dotted lines. Likewise, if majority of k-mers overlapping a het region have very high frequency since they are same as k-mers from the repeat regions, then HMM has limitation in identifying heterozygous calls from such regions. On removing both the error and repeat majority het calls from false negative rate, the updated recall and F1 score are shown in solid line in the Fig. 5.6. The operating confidence of 0.5 is chosen for the downstream task of haplotype-resolved assembly algorithm which gives a precision of 0.70, recall of 0.90, and F1 score 0.78.

5.4 Conclusion

In conclusion, the challenge of accurately assembling diploid genomes using nanopore sequencing technology lies in the high error rates of the long reads, which complicate the differentiation of heterozygous variants specific to each haplotype. Current methods mitigate this by integrating additional high-accuracy sequencing data, but this increases both cost and complexity. Therefore, there is a critical need for innovative assembly algorithms that can leverage the advantages of long nanopore reads while accurately identifying heterozygous variants without relying on supplementary data. This chapter aims to address this gap by focusing on the development of a novel method for heterozygous variant identification from nanopore sequencing reads only without the reference genome or a draft assembly from the reads data, thereby facilitating development of an accurate and contiguous haplotype-resolved genome assembly.

CHAPTER 6

Conclusion and Open questions

The rapid advancements in nanopore sequencing technology have ushered in new possibilities for genomic research, allowing the reading of extraordinarily long DNA fragments. However, the high error rates associated with nanopore sequencing present significant technical challenges. This thesis addressed these challenges through a series of computational approaches designed to enhance the accuracy, efficiency, and overall utility of nanopore sequencing in various genomic applications.

Chapter 2: Mathematical Model for Nanopore Sequencer

In this chapter, we studied a comprehensive mathematical model to characterize the sequencing process of nanopore devices. This model incorporates key several non-idealities such as non intersymbol interference, insertions-deletions, channel fading, and random noise, which are inherent in the nanopore sequencing process. By understanding bounds on the information extraction capacity, we studied benchmarks for current base-calling algorithms and guidelines for designing improved nanopores. Additionally, this model offers a method to quantify the storage capacity of nanopore sequencers when employed in DNA-based storage systems. The findings from this study lay the groundwork for developing algorithms that utilizes the error profiles in the nanopore sequencer for its downstream applications, enhancing base-calling accuracy, and optimizing nanopore design for more efficient data extraction and storage.

Chapter 3: QAlign

The chapter introduced QAlign, a preprocessing tool designed to improve the alignment of long nanopore reads to reference genomes or other reads. QAlign converts nucleotide reads into discretized current levels that takes the error modes of nanopore sequencing into account while performing read alignments. This space transformation significantly boosts alignment accuracy and efficiency, raising alignment rates from approximately 80% to 90% and enhancing read overlap quality in multiple real datasets. The application of QAlign demonstrates a robust approach to mitigating the high error rates of nanopore reads, making it a valuable addition to the toolkit for genomic analysis.

Chapter 4: HQAlign

The chapter focused on the detection of structural variants (SVs) using nanopore sequenced reads. We developed HQAlign, an aligner that leverages the error characteristics of nanopore sequencing and incorporates SV-specific modifications in the alignment algorithm. HQAlign improves the detection of complementary SVs by $4\% - 6\%$ compared to the state-of-the-art aligner minimap2 and significantly enhances breakpoint accuracy across real nanopore data. Furthermore, HQAlign achieves superior alignment rates for nanopore reads against recent alignment algorithms such as minimap2 and QAlign. These improvements highlight HQAlign's potential in accurately identifying structural variants, which is crucial for understanding complex genomic rearrangements and their implications in human diseases.

Chapter 5: Heterozygous Variant detection for haplotype-resolved assembly

The final chapter addressed the challenge of assembling diploid genomes from long noisy reads produced by nanopore sequencers. We proposed an algorithm to identify heterozygous variants with high recall (90%) and precision (70%) without relying on additional reference information or preliminary draft assemblies. This algorithm is a crucial step towards reconstructing diploid genomes accurately, overcoming the limitations posed by high sequencing errors. By effectively distinguishing between heterozygotes and sequencing errors, this approach facilitates development of algorithms for diploid genome assembly, enabling more accurate representation of genetic diversity.

6.1 Open Questions and Future Directions

While the contributions of this thesis significantly advance the capabilities of nanopore sequencing, several open questions and areas for future research remain.

Haplotype-resolved genome assembly

The detection of heterozygous variants using the nanopore read data only is the primitive step of our on going work towards developing algorithm for haploty-resolved assembly for diploid human genome. Using the identified variants as "signature" for each haplotype, the goal is to develop an efficient algorithm that can phase the nanopore reads into two separate bins, one corresponding to each haplotype, and then perform the assemble the reads in each bin using the existing assembly algorithms to produce haplotype-resolved assembly.

Improving Base-Calling Accuracy

Despite the advancements in base-calling algorithms and mathematical modeling, achieving higher accuracy in base-calling remains a critical challenge. Future research could explore more sophisticated models that account for additional noise sources and sequencing errors. Integrating machine learning techniques, particularly transformer models, could also offer new avenues for improving base-calling accuracy by learning complex patterns in the sequencing data.

DNA Storage

Using nanopore sequencer as a reader for DNA storage based systems should focus on enhancing the efficiency and reliability of both encoding and decoding processes. This includes developing advanced error correction algorithms to mitigate the high error rates inherent in nanopore sequencing and exploring novel encoding schemes that maximize storage density and retrieval accuracy.

Ethical and Regulatory Considerations

As nanopore sequencing technology becomes more widespread, ethical and regulatory considerations will become increasingly important. Research in this area should focus on ensuring data privacy, addressing ethical concerns related to genetic information, and developing regulatory frameworks to oversee the use of nanopore sequencing in clinical and research settings.

6.2 Conclusion

The studies presented in this dissertation demonstrate significant progress in addressing the challenges posed by high error rates in nanopore sequencing. By developing novel computational approaches, we have improved the accuracy and efficiency of genomic analysis, paving the way for broader applications of nanopore sequencing technology. As research in this field continues to evolve, the potential for nanopore sequencing to revolutionize genomics and related disciplines remains vast. Future work will undoubtedly build upon these foundations, further enhancing the capabilities and applications of this transformative technology.

REFERENCES

[1] Joshi, D., Mao, S., Kannan, S. & Diggavi, S. Qalign: aligning nanopore reads accurately using current-level modeling. *Bioinformatics* **37**, 625–633 (2021).

[2] Mao, W., Diggavi, S. N. & Kannan, S. Models and information-theoretic bounds for nanopore sequencing. *IEEE Transactions on Information Theory* **64**, 3216–3236 (2018).

[3] Marçais, G. & Kingsford, C. A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics* **27**, 764–770 (2011).

[4] Eichler, E. E. Box 1. models of genomic duplication. *Trends in Genetics* **11**, 661–669 (2001).

[5] Mikheyev, A. S. & Tin, M. M. A first look at the oxford nanopore minion sequencer. *Molecular ecology resources* **14**, 1097–1102 (2014).

[6] Deamer, D., Akeson, M. & Branton, D. Three decades of nanopore sequencing. *Nature biotechnology* **34**, 518–524 (2016).

[7] Križanović, K., Echchiki, A., Roux, J. & Šikić, M. Evaluation of tools for long read rna-seq splice-aware alignment. *Bioinformatics* **34**, 748–754 (2018).

[8] Jain, M. *et al.* Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nature biotechnology* **36**, 338–345 (2018).

[9] Stancu, M. C. *et al.* Mapping and phasing of structural variation in patient genomes using nanopore sequencing. *Nature communications* **8**, 1–13 (2017).

[10] Chaisson, M. J., Mukherjee, S., Kannan, S. & Eichler, E. E. Resolving multicopy duplications de novo using polyploid phasing. In *International Conference on Research in Computational Molecular Biology*, 117–133 (Springer, 2017).

[11] Joshi, D., Diggavi, S., Chaisson, M. J. & Kannan, S. Hqalign: aligning nanopore reads for sv detection using current-level modeling. *Bioinformatics* **39**, btad580 (2023).

[12] Pevzner, P. A., Tang, H. & Waterman, M. S. An eulerian path approach to dna fragment assembly. *Proceedings of the national academy of sciences* **98**, 9748–9753 (2001).

[13] Haas, B. J. *et al.* De novo transcript sequence reconstruction from rna-seq using the trinity platform for reference generation and analysis. *Nature protocols* **8**, 1494–1512 (2013).

[14] DePristo, M. A. *et al.* A framework for variation discovery and genotyping using next-generation dna sequencing data. *Nature genetics* **43**, 491 (2011).

[15] Li, H. A statistical framework for snp calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics* **27**, 2987–2993 (2011).

[16] Mao, S., Mohajer, S., Ramachandran, K., Tse, D. & Kannan, S. Absnp: Rna-seq snp calling in repetitive regions via abundance estimation. In *17th International Workshop on Algorithms in Bioinformatics (WABI 2017)* (Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017).

[17] Laszlo, A. H. *et al.* Decoding long nanopore sequencing reads of natural dna. *Nature biotechnology* **32**, 829 (2014).

[18] Nanopore - how it works. `https://nanoporetech.com/how-it-works`.

[19] Tran, N. H., Zhang, X., Xin, L., Shan, B. & Li, M. De novo peptide sequencing by deep learning. *Proceedings of the National Academy of Sciences* **114**, 8247–8252 (2017).

[20] Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094–3100 (2018).

[21] Wu, T. D. & Watanabe, C. K. Gmap: a genomic mapping and alignment program for mrna and est sequences. *Bioinformatics* **21**, 1859–1875 (2005).

[22] Nanopolish. URL `\url{https://github.com/jts/nanopolish}`.

[23] Wick, R. R., Judd, L. M. & Holt, K. E. Performance of neural network basecalling tools for oxford nanopore sequencing. *Genome biology* **20**, 129 (2019).

[24] Wick, R. R., Judd, L. M., Gorrie, C. L. & Holt, K. E. Completing bacterial genome assemblies with multiplex minion sequencing. *Microbial genomics* **3** (2017).

[25] Nanopore r9 rapid run data release. `http://lab.loman.net/2016/07/30/nanopore-r9-data-release/` (2016).

[26] Whole genome sequencing (oxford nanopore r9.4) of escherichia coli uti89. `https://www.ncbi.nlm.nih.gov/sra/SRX4387499[accn]`.

[27] De Coster, W. *et al.* Structural variants identified by oxford nanopore promethion sequencing of the human genome. *Genome research* **29**, 1178–1187 (2019).

[28] Li, Y. *et al.* Deepsimulator: a deep simulator for nanopore sequencing. *Bioinformatics* **34**, 2899–2908 (2018).

[29] Oxford nanopore human reference datasets. `https://github.com/nanopore-wgs-consortium/NA12878`.

[30] Byrne, A. *et al.* Nanopore long-read rnaseq reveals widespread transcriptional variation among the surface receptors of individual b cells. *Nature communications* **8**, 1–11 (2017).

[31] Gencode v27. `https://www.gencodegenes.org/human/release_27.html`.

[32] Haeussler, M. *et al.* The ucsc genome browser database: 2019 update. *Nucleic acids research* **47**, D853–D858 (2019).

[33] Oxford nanopore kmer models. `https://github.com/nanoporetech/kmer_models`.

[34] DePristo, M. A. *et al.* A framework for variation discovery and genotyping using next-generation dna sequencing data. *Nature genetics* **43**, 491 (2011).

[35] Loman, N. J., Quick, J. & Simpson, J. T. A complete bacterial genome assembled de novo using only nanopore sequencing data. *Nature methods* **12**, 733–735 (2015).

[36] Pevzner, P. A., Tang, H. & Tesler, G. De novo repeat classification and fragment assembly. *Genome research* **14**, 1786–1796 (2004).

[37] Tang, F. *et al.* mrna-seq whole-transcriptome analysis of a single cell. *Nature methods* **6**, 377–382 (2009).

[38] Alkan, C., Coe, B. P. & Eichler, E. E. Genome structural variation discovery and genotyping. *Nature reviews genetics* **12**, 363–376 (2011).

[39] Pan-cancer analysis of whole genomes. *Nature* **578**, 82–93 (2020).

[40] Rowell, W. J. *et al.* Comprehensive variant detection in a human genome with highly accurate long reads. *EUROPEAN JOURNAL OF HUMAN GENETICS* **27**, 1723–1723 (2019).

[41] Li, H. New strategies to improve minimap2 alignment accuracy. *Bioinformatics* **37**, 4572–4574 (2021).

[42] Chaisson, M. J. *et al.* Multi-platform discovery of haplotype-resolved structural variation in human genomes. *Nature communications* **10**, 1–16 (2019).

[43] Ebert, P. *et al.* Haplotype-resolved diverse human genomes and integrated analysis of structural variation. *Science* **372**, eabf7117 (2021).

[44] Rhie, A. *et al.* The complete sequence of a human y chromosome. *bioRxiv* (2022).

[45] Smolka, M. *et al.* Comprehensive structural variant detection: from mosaic to population-level. *BioRxiv* (2022).

[46] English, A. C., Menon, V. K., Gibbs, R., Metcalf, G. A. & Sedlazeck, F. J. Truvari: Refined structural variant comparison preserves allelic diversity. *bioRxiv* (2022).

[47] Ren, J. & Chaisson, M. J. lra: A long read aligner for sequences and contigs. *PLOS Computational Biology* **17**, e1009078 (2021).

[48] Zook, J. M. *et al.* A robust benchmark for detection of germline large deletions and insertions. *Nature biotechnology* **38**, 1347–1355 (2020).

[49] Yang, C., Chu, J., Warren, R. L. & Birol, I. Nanosim: nanopore sequence read simulator based on statistical characterization. *GigaScience* **6**, gix010 (2017).

[50] Jeffares, D. C. *et al.* Transient structural variations have strong effects on quantitative traits and reproductive isolation in fission yeast. *Nature communications* **8**, 1–11 (2017).

[51] Li, H. *et al.* A synthetic-diploid benchmark for accurate variant-calling evaluation. *Nature methods* **15**, 595–597 (2018).