CHARACTERIZATION OF STRUCTURAL VARIATION THROUGH ASSEMBLY-TO-ASSEMBLY COMPARISON

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

By
Muhammet Rafi Çoktalaş
September 2024

Characterization of Structural Variation through Assembly-to-Assembly Comparison
By Muhammet Rafi Çoktalaş
September 2024

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Can Alkan(Advisor)

Eray Tüzün

Aybar Can Acar

Approved for the Graduate School of Engineering and Science:

Orhan Arıkan
Director of the Graduate School

ABSTRACT

CHARACTERIZATION OF STRUCTURAL VARIATION THROUGH ASSEMBLY-TO-ASSEMBLY COMPARISON

Muhammet Rafi Çoktalaş
M.S. in Computer Engineering
Advisor: Can Alkan
September 2024

Structural variations (SVs) are genomic variations affecting more than 50 nucleotides of DNA. SVs play a crucial role in evolution and have critical phenotypic effects on organisms, such as genetic diseases in humans like autism, schizophrenia, epilepsy, and cancer. Thus, SV characterization is of great significance. In the past, read-based methodologies were utilized due to the infeasibility of constructing genome assemblies. However, with technological advancements, assembling genomes has become significantly more feasible, and complete assemblies of human and other primate genomes have been constructed. Despite the high-quality assemblies, SV discovery in human genomes remains challenging due to the genome's repetitive nature and complex rearrangements caused by a combination of SVs. Most existing SV discovery tools operating on genome assemblies require whole genome alignments, leading to high preprocessing times and memory usage. Therefore, new algorithms are still needed to efficiently discover SVs.

Here, we propose Strive, a linear time algorithm that operates on genome assembly sketches instead of whole genome alignments to characterize insertions, deletions, and inversions.

We evaluated the performance Strive with two experiments: simulated data from the human reference genome (GRCh38.p14 / hg38) and real data using a full genome assembly from the Telomere to Telomere Consortium (CHM13). Strive is able to accurately detect insertions, deletions, and inversions in 11 to 12 seconds in addition to preprocessing times ranging from 50 to 55 seconds. Strive achieved over 95% precision and recall values in the simulations without duplications. In the simulations that included segmental duplications and SNPs

and in the experiment with CHM13 assembly, although still maintaining over 95% recall in inversion discovery, the precision and recall for insertions and deletions were lower, suggesting a need for increased robustness to duplications.

Keywords: structural variation, genome assembly, sketching.

ÖZET

TÜM GENOM KARŞILAŞTIRMASI İLE YAPISAL VARYASYONLARIN KARAKTERİZASYONU

Muhammet Rafi Çoktalaş
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Danışmanı: Can Alkan
Eylül 2024

Yapısal varyasyonlar (SV'ler), DNA'nın 50 nükleotitten fazlasını etkileyen genomik varyasyonlardır. SV'ler evrimde önemli bir rol oynar ve insanlarda otizm, şizofreni, epilepsi ve kanser gibi genetik hastalıklar da dahil olmak üzere organizmalar üzerinde kritik fenotipik etkilere sebep olur. Bu nedenle, SV karakterizasyonu büyük önem taşır. Geçmişte, genom birleşimlerinin maliyeti nedeniyle küçük DNA sekansları tabanlı metodolojiler kullanıldı. Ancak, teknoloji alanındaki ilerlemelerle birlikte genom birleşimleri önemli ölçüde daha erişilebilir hale geldi. İnsan ve diğer primat genomlarının dahi eksiksiz birleşimleri oluşturuldu. Yüksek kaliteli birleşimlere rağmen, insan genomlarında SV keşfi, genomun tekrarlı yapısı ve SV'lerin kombinasyonu nedeniyle ortaya çıkan karmaşık yeniden düzenlemeler nedeniyle hala zorlu bir görev olarak kalmaktadır. Mevcut SV keşif araçlarının çoğu, genom birleşimleri üzerinde çalışırken tüm genom hizalamaları gerektirmekte olup, bu da yüksek ön işleme sürelerine ve bellek kullanımına yol açmaktadır. Bu nedenle, SV'leri verimli bir şekilde keşfetmek için yeni algoritmalara hala ihtiyaç vardır.

Burada, tüm genom hizalamaları yerine genom birleşim taslakları üzerinde çalışan ve ekleme, silme ve ters çevirmeleri karakterize eden doğrusal zamanlı bir algoritma olan Strive'i sunuyoruz.

Strive'in performansını, GRCh38.p14 (hg38) birleşimi üzerinde simüle edilmiş verilerle gerçekleştirdiğimiz iki deneyle ve CHM13 birleşiminde tespit edilen SV'ler ile değerlendirdik. Strive, 50 ila 55 saniye arasında değişen ön işleme süreleriyle 11 ila 12 saniye içinde ekleme, silme ve ters çevirmeleri doğru bir şekilde tespit edebilmiştir. Strive, duplikasyonların olmadığı simülasyonlarda %95'in üzerinde kesinlik ve duyarlılık değerleri elde etti. Duplikasyonlar ve SNP'ler içeren simülasyonlarda, ters çevirmeleri keşfetmede hala %95'in üzerinde

duyarlılığı korumasına rağmen, ve CHM13 birleşimi ile yapılan deneyde ekleme ve silme işlemleri için kesinlik ve duyarlılık daha düşüktü. Bu durum, duplikasyonlara karşı daha fazla dayanıklılığa ihtiyaç olduğunu göstermektedir.

Anahtar sözcükler: yapısal varyasyon, genom birleşimi, taslak.

Acknowledgement

I am forever grateful to my advisor, Can Alkan. His boundless patience, which I have tested to its limits, and his profound wisdom have taught me countless invaluable lessons. Without him, this thesis would not have been possible. I cannot even dream of a better advisor.

I thank Eray Tüzün and Aybar Can Acar for accepting to be the committee members for this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Genomic variations represent the differences in the DNA sequence between individual genomes. These variations can range from single nucleotide polymorphisms (SNPs) to larger structural variations (SVs) encompassing tens to thousands or even millions of nucleotides. Such genomic diversity causes phenotypic differences among individuals, influencing traits like physical appearance, behavior, and susceptibility to diseases, making each individual unique. Understanding these phenotypic differences requires a comprehensive analysis of genomic variations. Since the first results of the Human Genome Project [3], the volume of genomic data veritably increased, setting the stage for extensive projects on genomic variations, such as the 1000 Genomes Project [4], and leading to catalogs and maps of genomic variations [5, 6, 7, 8, 9].

SVs include deletions, insertions, inversions, duplications, and translocations of DNA segments, each of which can significantly alter gene function or regulation [1]. These alterations can lead to traits that range from simple phenotypic differences to serious genetic disorders [10]. The study of SV discovery is, therefore, crucial. However, due to their size and complexity, SVs are more difficult to detect and analyze than smaller-scale variations. This challenge has driven the development of methodologies and tools for SV discovery.

The discovery of SVs relies heavily on read-based approaches. These methods involve sequencing short fragments of DNA (reads) and aligning them to a reference genome to identify discrepancies that suggest the presence of SVs [1]. Aligning reads is a complex procedure, primarily due to the repetitive nature of genomes. For example, nearly half of the human genome comprises repeated regions [11]. Repetitive regions lead to ambiguous alignments, making it difficult to accurately map reads from these regions to their original positions. Additionally, mismatches in alignments further complicate the read alignment procedure. Mismatches stem from sequencing errors and small mutations, such as SNPs and INDELs. Unlike issues caused by repetitiveness, mismatches can be addressed by utilizing mapping quality information to detect erroneous alignments caused by them [12]. Techniques like paired-end and split-read mapping have been instrumental in detecting SVs, from small insertions and deletions to large inversions and translocations. Yet, these techniques suffer from the inherent challenges and limitations of read alignment.

Genome assembly involves "stitching" these short reads into longer contiguous sequences (contigs), ultimately reconstructing the entire genome. The feasibility of constructing high-quality genome assemblies has dramatically increased, paving the way for utilizing assembly-based methods for SV discovery [13, 14].

Assembly-based methods allow for a more comprehensive and precise characterization of SVs. By working directly with genome assemblies, many of the limitations associated with read alignment, such as the challenges posed by repetitive sequences, can be bypassed [14].

1.1 Motivation

Current SV discovery tools operating on genome assemblies typically require whole genome alignments. Whole genome alignments are time-consuming and resource-intensive, especially for large genomes like the human genome [15]. An

algorithm that operates on sketches of genomes and/or their alignments can reduce these costs. Such an algorithm that also provides satisfactory results would have the potential to become a pioneer for future tools. Furthermore, a rise in the number of tools focusing on genome assemblies can be anticipated as high-quality and complete genome assemblies are becoming increasingly accessible [16, 14, 17, 18].

1.2 Organization of the Thesis

This thesis consists of four chapters. The second chapter includes the literature review and background knowledge required to explain the problem and the methodology. A detailed explanation of the methodology is in the third chapter. The fourth chapter concludes the discussions with an analysis of experimental results and comparisons with other tools that use similar methodologies.

Chapter 2

Background

All organisms' genetic information is encased in double-stranded DNA molecules. Each strand consists of nucleotides categorized by their type of bases: adenine, cytosine, guanine, and thymine. Computationally, a genome is a string (or a set of strings) over an alphabet A, C, G, T. Human DNA is packed into 23 chromosome pairs, and the set of this information is called the human genome.

2.1 Genome Assembly

Sequences generated by modern sequencing instruments are considerably shorter in length (hundreds to thousands of base pairs) compared to the length of the genomes that are being studied (tens of thousands to billions of base pairs) [19]. Genome assembly is the construction of the whole original DNA genome from smaller sequenced fragments (contigs). Assemblies can be constructed with or without a reference. Constructing assemblies without a reference is called *de novo* assembly. A full reference for the human genome is available thanks to the Human Genome Project [3].

The latest human genome reference, GRCh38.p14, has an ungapped length of 2.9 Gb, whereas the genome size is 3.1 Gb. Having gaps in the reference

genome makes tasks like variation discovery harder, but thanks to the efforts of the Telomere-to-Telomere (T2T) Consortium, a reference genome assembly (CHM13) with no gapped regions is achieved [16]. T2T Consortium has also paved the way for complete assemblies of six ape species [18]. With near-accurate genome assemblies now feasible, assembly-based analysis of genomes has become increasingly practical.

2.2 Mapping

Genomic mapping is a critical process in genomics that involves determining the position of genes or other significant sequences within a genome. Mapping provides a framework for further analysis, such as variant discovery, genome assembly, and functional annotation.

2.2.1 Read Mapping

Read mapping involves aligning short DNA sequences, known as reads, to a reference genome. This method has been fundamental in sequencing projects, particularly with the advent of high-throughput sequencing technologies. Despite its widespread use, read mapping has limitations, especially in regions of the genome that are highly repetitive or contain SVs. The alignment process compares each read to the reference genome to find the best matching location. Two main approaches are Burrows-Wheeler Transform (BWT) [20] and hash-based seed-and-extend. Tools such as BWA [21], Bowtie 2 [22], and Minimap2 [23] are widely used for read mapping.

Despite its widespread use, read mapping has limitations, especially in regions of the genome that are highly repetitive or contain complex structural variations. The accuracy of mapping can be affected by sequencing errors, the length of the reads, and the quality of the reference genome.

2.2.2 Optical Mapping

Optical mapping is first developed in 1993 [24]. DNA molecules are nicked by restriction enzymes, labeled with fluorescent dyes, then stretched in nanochannels and imaged with high-resolution fluorescence microscopy, providing structural information about DNA molecules and outputting restriction site label positions as optical maps [24, 25]. Optical maps are significantly longer than the reads generated by sequencers, making optical mapping a powerful tool in tasks like SV discovery [25, 26]. Moreover, optical mapping is less affected by the repetitive nature of the genome [26].

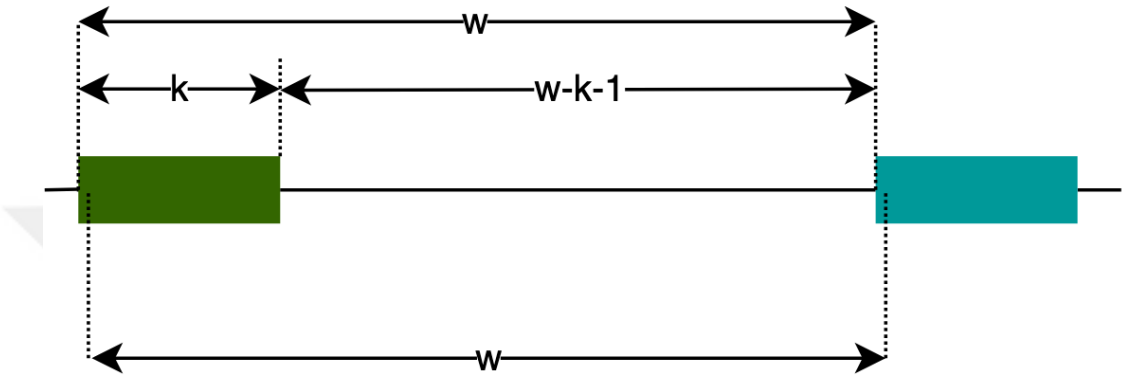
2.3 Sketching

Sketching is widely used for creating approximate, compact summaries, or synopses [27], of large data like genomic data [28]. One of the main uses of sketches in bioinformatics is mapping [29].

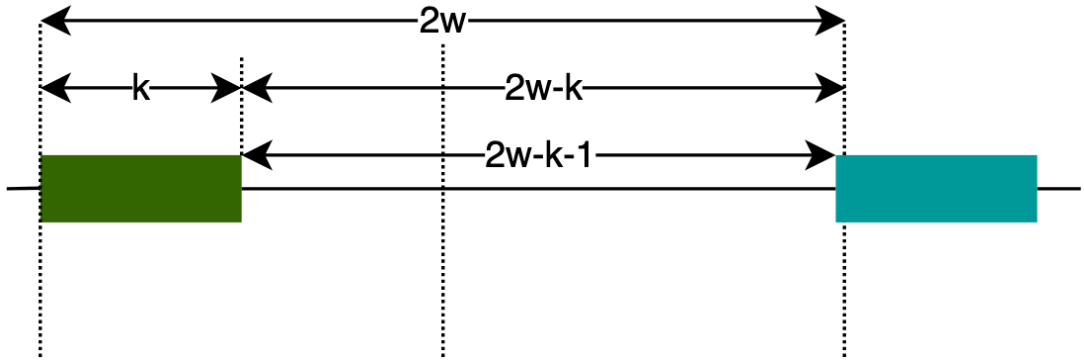
A common approach for sketching large genomic data is using k-mers, which are substrings of fixed length k [30]. k-mers are widely used in high sensitivity tasks like *de novo* genome assembly. Yet, using k-mers can prove computationally expensive for large genomes as for a sequence of length n; there are $n - k + 1$ k-mers that should be generated, stored, and processed.

Minimizers are introduced as a method of k-mer selection to reduce the computational and storage cost of k-mers [31]. By use of minimizers, the number of k-mers can be significantly reduced, resulting in savings in computational power and memory [31]. A minimizer is a substring that minimizes a k-mer hash function $\phi : \Sigma^k \to \mathbb{Z}$ in a window of w consecutive overlapping k-mers where Σ^k denotes the set of strings of length k over the alphabet of the genome $\Sigma = \{A, C, G, T\}$ [32]. The distance between two consecutive minimizers is $w - k - 1$ if windows can overlap and $2w - k - 1$ if windows are non-overlapping, where the first minimizer is selected at the beginning of its window and the second is selected at the

end of its window as illustrated in Figure 2.1. This inherent limitation provides uniformity as minimizers provide window guarantee [28].



(a) Furthest consecutive minimizers in overlapping windows



(b) Furthest consecutive minimizers in non-overlapping windows

Figure 2.1: Maximum distances between consecutive minimizers

Both k-mers and minimizers are sensitive to mutations [33, 34]. To overcome this limitation, syncmers [34] and strobemers [33] are proposed.

Syncmers select k-mers based on a specific pattern (a submer rule) within the sequence. For example, one rule could be to select k-mers in a window where a certain position within the k-mer (like the middle character) meets a specific criterion, such as being the smallest or largest character in that k-mer. As the sliding window moves across the sequence, the rule is applied to each window. Therefore, unlike minimizers, syncmer selection does not depend on all the characters of k-mers. This results in a consistent selection of k-mers across sequences, even when small mutations exist. While syncmers are less

prone to mutation effects, they do not provide as strong of a window guarantee as minimizers, which are more uniformly distributed across a sequence [28].

A strobemer of order n in sequence is a subsequence of n ordered non-contiguous substrings (strobes) of equal length l. For a strobemer containing strobes $m_1, m_2, ..., m_n$, if the first strobe, m_1, begins at position i, the second strobe, m_2, is selected from the interval $[i + w_{min} : i + w_{max}]$, where w_{min} and w_{max} define the minimum and maximum allowed spacing between strobes. Allowing flexibility in the spacing of strobes, strobemers make producing matching strobemers between two sequences in a region with indels possible [33]. For example, a strobemer of order 2 for the sequence AGCTTACGGA is formed using two strobes. For $l = 3$, $w_{min} = 1$ and $w_{min} = 4$, the first (anchor) strobe can be selected as AGC (starting at position 0), leaving the second strobe to be chosen in the range $[1, 4]$. The second strobe can, therefore, be GCT, CTT, TTA, or TAC. Each of which would form a different strobemer combined with the first strobe. Many approaches can be adapted for selecting strobes, such as minstrobes, randstrobes, and hybrid strobes [33].

2.4 Genomic Variation

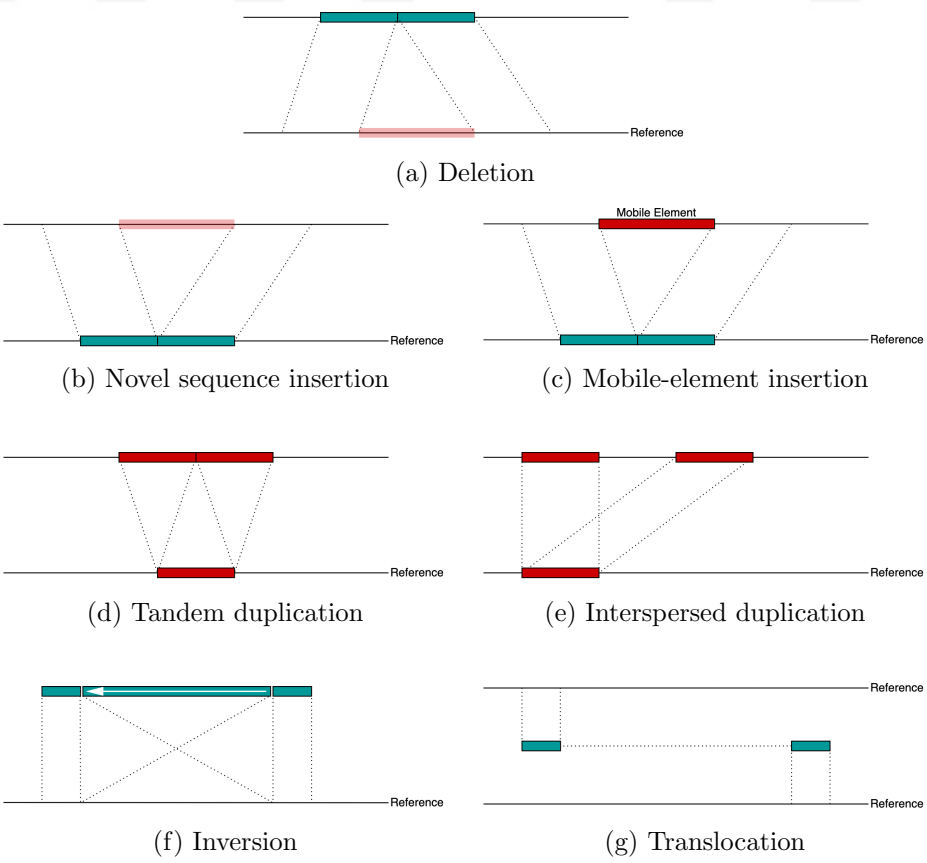
2.4.1 SNP, INDEL, STR

Single nucleotide polymorphism (SNP) is change of a single nucleotide in the genome. Over 3.1 million SNPs are revealded by the International HapMap Consortium [35]. INDELs are insertions deletion of size smaller than 50 nucleotides and corresponds to 16% to 25% of the genomic variation in humans [36]. A short tandem repeat (STR) is the repetition of a small segment in the genome. STRs play an important role to gene expression and are used in forensic DNA profiling [37, 38].

2.4.2 Structural Variation

Variations between genomes spanning more than 50 base pairs (up to several millions) are called structural variations (SVs) [1]. Types of SVs are deletion, insertion (novel and mobile-element), duplications (tandem and interspersed), inversions and translocations (Figure 2.2) [39, 40, 41]. SVs that are affecting the size of the genome are called copy number variations (CNVs) or imbalanced SVs [1]. Deletions, insertions and duplications are imbalanced, while inversions and translocations are balanced [1].

Numerous studies have demonstrated the relationship between SVs and genetic diseases like autism, schizophrenia, epilepsy and cancer [42, 43, 44, 45]. SVs are also a key factor in evolution [46]. Consequently, the discovery of SVs are of great significance.



(a) Deletion

(b) Novel sequence insertion

(c) Mobile-element insertion

(d) Tandem duplication

(e) Interspersed duplication

(f) Inversion

(g) Translocation

Figure 2.2: Structural variation types (adapted from [1])

2.5 Structural Variation Discovery

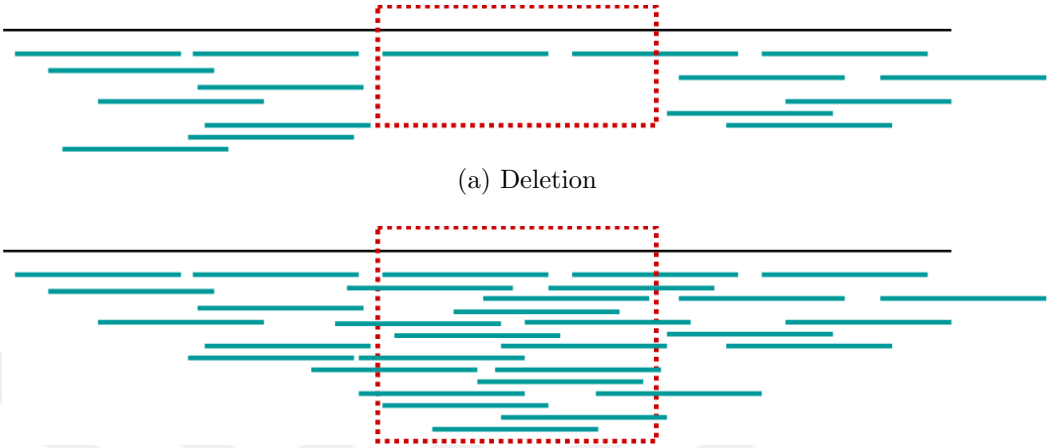
Due to limitations in constructing assemblies in the past, SV discovery tools had relied on reads. Mainly, three approaches has been followed: read-pair, read-depth and split-read. However, constructing assemblies has become more feasible. Consequently, SV discovery tools based on genome assembly comparisons have been developed as well. Another approach for SV discovery is using optical maps.

2.5.1 Read-pair

Read-pair method is the most widely followed approach for SV discovery [1]. Read-pairs are aligned to a reference genome, allowing to analyze the distance between the pairs. Two thresholds are used to identify insertions and deletions, with the threshold for insertions being smaller than the one for deletions. Distances below the insertion threshold indicate insertions, while distances above the deletion threshold indicate deletions. Taking the orientation of the reads into account, inversions and duplications can also be detected using read-pairs [39, 47].

2.5.2 Read-depth

The number of reads mapping to a region is called read-depth. Assuming that read-depths follow a Poisson distribution, read-depth methods are used to detect deletions and duplications by analyzing the regions diverging from this distribution [1, 48]. Regions of lower read-depth are denoted as deletions and of higher read-depth are denoted as duplications. These signatures are visualized in Figure 2.3. The downsides of this approach are poor breakpoint resolution, susceptibility to sequencing bias and the dependency on the coverage of the dataset.



(a) Deletion

(b) Duplication

Figure 2.3: Read-depth signatures of deletion and duplication

2.5.3 Split-read

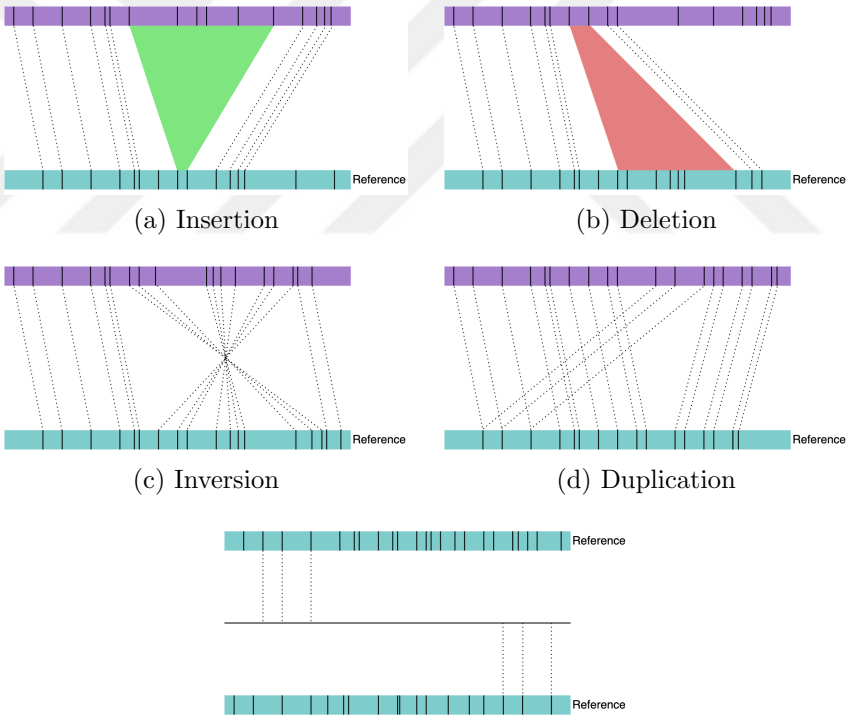
Read mappings can include portions that cannot be mapped correctly. These portions are called soft clips. Soft clips mostly are at beginning or at the and of the reads. Split-read approaches map these reads to reference genome, allowing gaps. Analysis of the distance and the orientations of these remapped reads can indicate SVs. Split-read approaches can produce single base-pair resolution in breakpoints [1]. The downside of this approach is the reliance on read lengths as remapping will be more accurate with reads of longer lengths.

2.5.4 Genome Assembly Comparison

As high quality assemblies become more accessible, SV discovery tools focused on genome assembly comparison are developed in an effort to overcome the limitations posed by read based approaches. This approach is based on the comparative analyses of whole genome alignments of two assemblies. The downside of this approach is the computational cost of such alignments, especially working with large genomic data.

2.5.5 Optical Maps

This approach is based on the analysis of the distances between restriction sites in optical maps. Differences in distances and relative positions of restriction site labels indicate SVs. Signatures are illustrated in Figure 2.4. The downsides of this approach are difficult *de novo* assembly of optical maps or construction of consensus and false positive restriction size labels [25]. In addition, optical maps do not provide any sequence information and they are limited to large SV discovery with poor breakpoint resolution.



(a) Insertion (b) Deletion

(c) Inversion (d) Duplication

(e) Translocation

Figure 2.4: Optical map signatures

Chapter 3

Methods

STRIVE discovers insertions, deletions, and inversions by comparing the positions of generated minimizers from the reference genome against their alignment to the query genome. The methodology resembles the analysis of restriction site labels in optical mapping-based approaches discussed in Section 2.5.5.

3.1 Sketching Genomes via Minimizers

STRIVE is not dependent on any sketching technique, but using minimizers without overlapping windows is preferred because of the window guarantee, hence the uniformity they provide as mentioned in Section 2.3. Minimizers are generated from the reference genome FASTA file and stored in the order found in the FASTA file. The name and the position of the minimizers are also stored in a tab-delimited file. This tab-delimited file serves as an alignment file for the reference genome. Minimizer sequences are named in a way that can be sorted when used with other genomes, i.e., it is possible to sort minimizers by their positions on the reference genome, even with different positions, by using their sequence names. To achieve this, minimizer sequences are named using their position in the reference genome. Sketching outputs are exemplified in Figure 3.1.

```
>0000010100     10098
AACCCTAACCCAACCCTAACCCTAACCCTAACCCTAACCCTAACCCCTAACCCTAACC
>0000010200     10104
AACCCAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCCTAACCCTAACCCTAACC
>0000010300     10248
AAACCCTAAACCCTAACCCTAACCCTAACCCTAACCCTAACCCCAACCCCAACCCCAACCCCAA
>0000010400     10304
AACCCCAACCCCAACCCCAACCCTAACCCCTAACCCTAACCCTAACCCTACCCTAACCCTAACC
>0000010500     10438
AACCCCTAACCCTAACCCTAACCCTAACCCTCGCGGTACCCTCAGCCGGCCCGCCCGCCCGGGT
```

(a) Minimizer FASTA file

```
0000010100      10098
0000010200      10104
0000010300      10248
0000010400      10304
0000010500      10438
```

(b) Reference locations file

Figure 3.1: Example sections from FASTA for generated minimizers with window size 100 and minimizer size 64 and reference locations file. Minimizer sequences are named by the window start position prepended with zeros to ensure lexicographical sorting. The starting position of the minimizer sequence is also added.

3.2 Aligning Minimizers

Once minimizers are generated from the reference genome, an alignment of them to the query genome is constructed by using aligners like Minimap2 [23], BWA [49], or NGLMR [50]. Strive does not depend on any aligner, but it requires the primary alignments of the minimizer sequences to the query genome in SAM format [51]. Moreover, without secondary alignments, the number of alignments equals the number of minimizer sequences. Therefore, Strive operates on primary alignments.

Mapping errors and coincidental alignments reduce the effectiveness of Strive. These stem from the inherent errors in mapping tools, the presence of SNPs, and repetitive regions in the genome. Disregarding secondary and supplementary alignments helps mitigate the negative impact. Further solutions are introduced in Section 3.4.2.

3.3 STRIVE Algorithm

STRIVE compares the alignments of minimizers to both the reference and the query genomes and reports the structural variances. Differences between the positions indicate insertions and deletions. Changes in the ordering indicate inversions. The locality is achieved by utilizing the minimizer sequence names, as the minimizer sequences are sorted by their positions in the reference genome.

Reference position file and query alignment file are processed line by line. At the same line count, the reference file contains the position, and the query alignment file contains all the alignment details for the same minimizer. Each line pair is used to create an event point. An event point consists of the position of the minimizer sequence in the reference genome, the position of the minimizer sequence in the query genome, the CIGAR string, and the reverse complement bit of the flag from the alignment of the minimizer to the query genome. Unmapped minimizer sequences are excluded while processing the lines.

As mentioned in Section 3.2, some alignments can negatively impact the algorithm. Such lines are tried to be ignored for event point creation by utilizing the number of mismatches and mapping quality information. Furthermore, the *closeness* of the positions are also checked. The closeness is calculated as $\frac{min(r,q)}{max(r,q)}$ where r is the position of the minimizer sequence in the reference genome and q is the position of the minimizer sequence alignment to the query genome.

Event Points are processed three by three as they are created. Insertions and deletions do not span over more than two event points. Both can be reported only by comparing the positions kept in the event points. If the difference between the positions of the minimizer sequence in the reference genome and between the positions of the minimizer sequence alignment to the query genome is not the same, an insertion or deletion is reported. Specifically, if the difference in the reference genome is greater, a deletion is reported between the corresponding positions; if the difference in the query genome is greater, an insertion is reported. Inversions can span over more than two event points; therefore, a state is introduced to the

processing of the points, inversion handling state (IHS). IHS starts when an event point with the reverse complement bit set to 1 is encountered. The start is kept when the method is called within this state. In IHS, encountering an event point with the reverse complement bit set to 0 results in reporting an inversion and exit from IHS. This process is visualized in Figure 3.2.

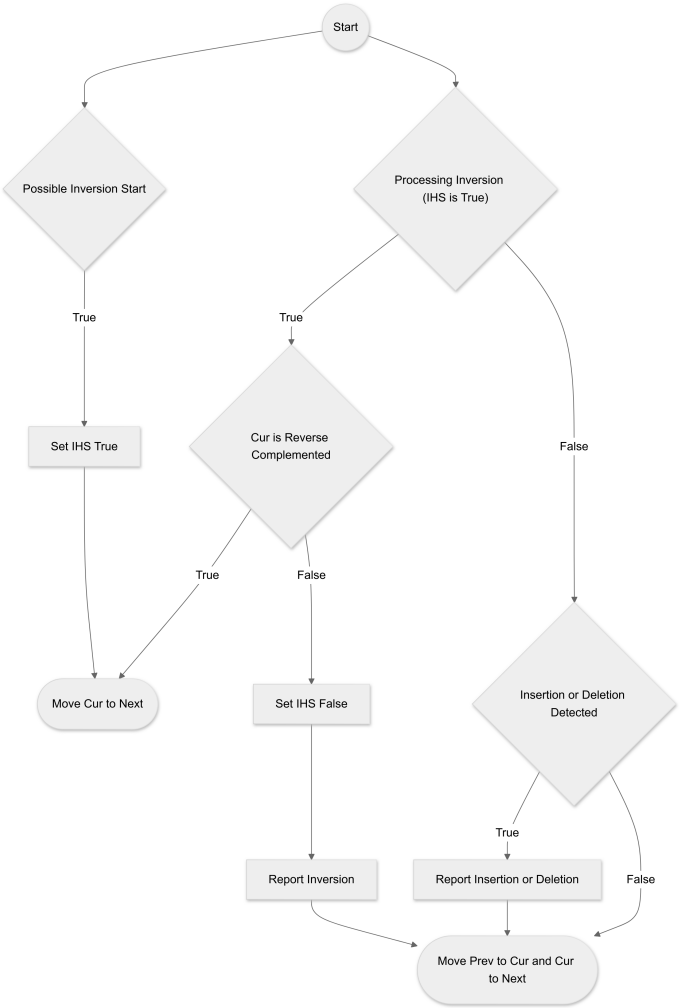


Figure 3.2: Overview of Process Points

CIGAR strings are utilized if an SV's interval endpoints coincide with minimizer sequences. Soft-clipped regions are taken into account. For insertions and deletions, if the CIGAR string of the event point corresponds to the

- start of the SV starts with hard matches, e.g., 52M12S, 64M; the number

of hard matches is added to the start endpoint of the interval.

- end of the SV starts with soft clippings, e.g., 12S52M; the number of soft clippings is added to the endpoint of the interval.

For inversions, if the CIGAR string of the event point corresponds to the

- start of the inversion starts with hard matches, e.g., 52M12S, 64M; the number of soft clippings is added to the endpoint of the interval.
- end of the inversion starts with soft clippings, e.g., 12S52M; the number of hard matches is added to the start endpoint of the interval.

Algorithm 1 Create Event Point

Input: tab separated line with minimizer name and location on the reference genome, a line of the alignment file of minimizer alignments to query genome, minimum mapping quality Q_{min}, maximum number of mismatches m_{max}, minimum closeness C_{min}

Output: Event Point

$Q \leftarrow$ mapping quality from MAPQ field
if $Q \leq Q_{min}$ **then**
    return null
end if
$m \leftarrow$ number of mismatches from NM:i tag
if $m > m_{max}$ **then**
    return null
end if
$r \leftarrow$ location on the reference genome
$q \leftarrow$ location on the query genome
$C \leftarrow \frac{min(r,q)}{max(r,q)}$
$UM \leftarrow$ unmapped bit of the flag
$SA \leftarrow$ supplementary alignment bit of the flag
if $UM \neq 1$ **and** $SA \neq 1$ **and** $C \geq C_{min}$ **then**
    $RC \leftarrow$ reverse complement bit of the flag
    $CIGAR \leftarrow$ CIGAR string
    return Event Point of (r, q, $CIGAR$, RC)
else
    return null
end if

Algorithm 2 STRIVE

Input: tab separated file with lines of minimizer name and location on the reference genome, alignment file of minimizer alignments to query genome, minimum mapping quality Q_{min}, maximum number of mismatches m_{max}, minimum closeness C_{min}, minimum SV size s_{min}, maximum SV size s_{max}

Output: 3 BED files denoting chromosome, start, end, for discovered insertions, deletions and inversions.

$prev \leftarrow$ null
while $prev = null$ **do**
    $l_r \leftarrow$ line from the reference locations file
    $l_q \leftarrow$ line from the query alignment file
    $prev \leftarrow$ Create Event Point(l_r,l_q)
end while
$cur \leftarrow$ null
while $cur = null$ **do**
    $l_r \leftarrow$ line from the reference locations file
    $l_q \leftarrow$ line from the query alignment file
    $cur \leftarrow$ Create Event Point(l_r,l_q)
end while
$IHS \leftarrow$ false {Inversion handling state}
while Files have unprocessed lines **do**
    $l_r \leftarrow$ line from the reference locations file
    $l_q \leftarrow$ line from the query alignment file
    $next \leftarrow$ Create Event Point(l_r,l_q)
    if $next \neq null$ **then**
        Process Points($prev$, cur, $next$, BED files, IHS, s_{min}, s_{max})
    end if
end while

3.4 Parameters

3.4.1 Performance Related Parameters

The time complexity of STRIVE is defined by the number of iterations of the main loop and is affected by only one parameter: sliding window size w. The precision for the breakpoint positions is affected by w and the minimizer size k. The preprocessing for STRIVE is discussed in Section 3.2 consists only of aligning the minimizer sequences to the query genome and is affected by both w and k.

Conversely, an increase in k increases the complexity of the alignment of the sequences.

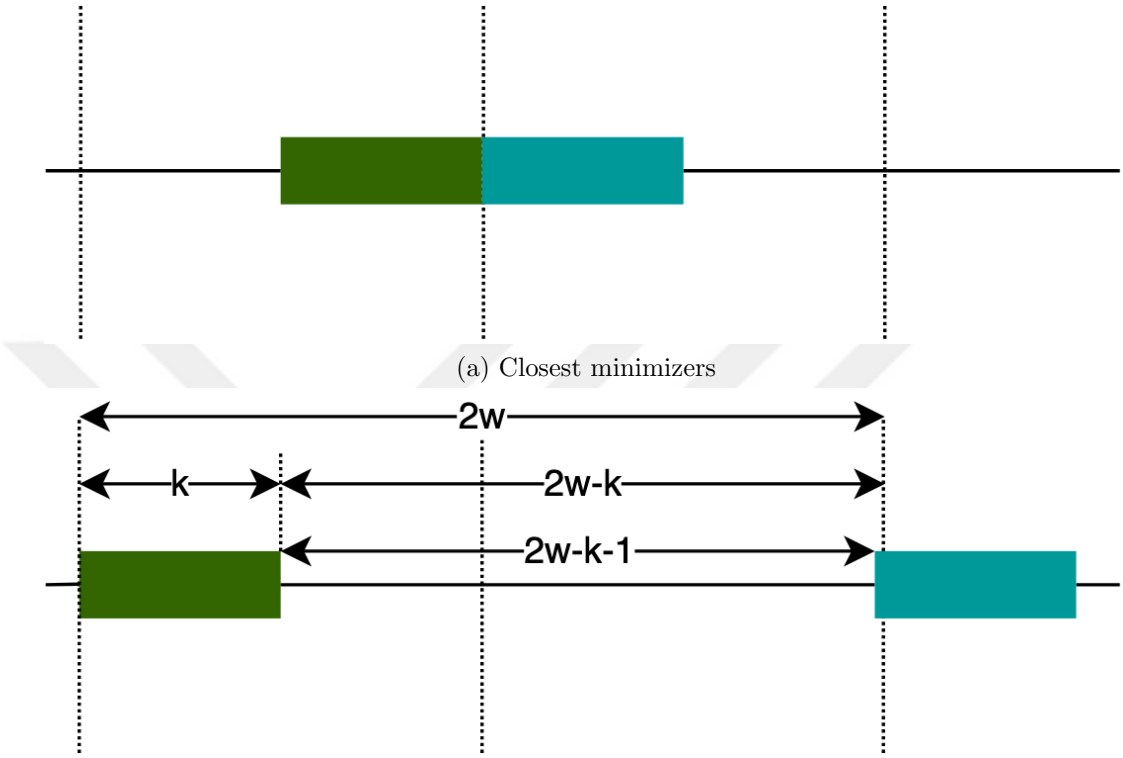
3.4.1.1 Window size w

The sliding window size is an indirect parameter for STRIVE. It is a parameter for minimizer generation. Its effect on STRIVE is in two aspects: time and precision.

The number of generated minimizers is equal to $\frac{Genome\ size}{w}$. Consequently, the number of iterations in the main loop of STRIVE equals $\frac{Genome\ size}{w}$, since each line pair corresponding to a minimizer is processed only once, and processing is confined in a single iteration.

Breakpoints of reported SVs can occur between a minimizer's start and end positions. In other cases, SVs may be entirely in between minimizers. Thus, the breakpoints of the reported SV can be as off as the distance between two consecutive minimizers. In the reference genome, the distance between minimizer positions is in the set $\{n \in \mathbb{Z} \mid 0 \leq n < 2w - k\}$ as illustrated in Figure 3.3 where dotted lines denote sliding windows. However, in the alignment to the query genome, this distance can exceed the upper limit due to unmapped or disregarded minimizer sequence alignments. This is illustrated in Figure 3.4 where faded colored minimizers are disregarded by STRIVE and dotted lines are sliding windows. Disregarded minimizers lead the distance between consecutive minimizers to exceed $2w-k$ in the query genome alignment. Moreover, an increase in w reduces the number of generated minimizers, hence reducing the number of alignments, easing the alignment task.

In summary, as w increases, the time complexity for STRIVE and the preprocessing decreases, but the precision for the reported regions also decreases.



(a) Closest minimizers

(b) Furthest minimizers

Figure 3.3: Minimum and maximum distances between consecutive minimizers when windows are non-overlapping

3.4.1.2 Minimizer size k

The minimizer size is also an indirect parameter for STRIVE. Similarly, it is also a parameter for minimizer generation. While it does not affect time complexity, it plays a significant role in the accuracy of breakpoint refinement. CIGAR strings of minimizer alignments to the query genome are used in breakpoint refinement as discussed in Section 3.3 and are affected by k.

Increase in k increases the complexity for alignment of the sequences. Moreover, increasing minimizer size k results in more unique alignments to the query genome, decreasing the number of coincidental alignments.

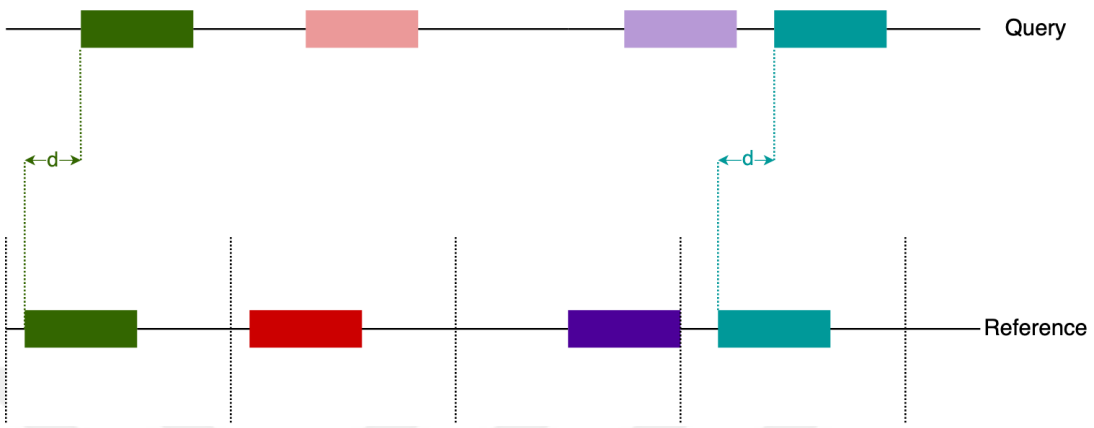


Figure 3.4: Unmapped or disregarded minimizer alignments

3.4.2 Adjustment Parameters

Mapping errors and coincidental alignments of minimizer sequences to the query genome negatively affect STRIVE as mentioned in Section 3.2. The following parameters are used to mitigate these negative effects. There is a tradeoff to consider, particularly with parameters disregarding certain alignments, as excluding alignments may lead to the loss of critical information.

3.4.2.1 Minimum Mapping Quality Q_{min}

In SAM formatted alignment files, mapping quality is represented by the MAPQ field [51]. Low MAPQ scores can indicate coincidental or erroneous alignments. Therefore, STRIVE disregards minimizer sequence alignments with mapping qualities below the specified threshold Q_{min}.

3.4.2.2 Maximum Number of Mismatches m_{max}

Coincidental alignments with high MAPQ scores can occur due to the presence of SNPs and repetitive regions in the genome. To address this, it is also important to control the number of mismatches. The number of mismatches in a mapping is indicated by the NM:i tag in SAM formatted alignment files [51]. STRIVE

filters out minimizer sequence alignments that exceed the maximum number of mismatches, m_{max}.

3.4.2.3 Minimum Closeness C_{min}

Due to the repetitiveness of the genome, coincidental alignments with high mapping quality and a low number of mismatches can occur, potentially being marked as the primary alignment. STRIVE uses the concept of closeness, as introduced in Section 3.3, to disqualify potentially coincidental alignments. Alignments are disregarded if the closeness of the positions falls below the specified threshold C_{min} to ensure that only relevant alignments are retained.

3.4.2.4 SV Size Bounds

STRIVE reports SVs of size that fall within a specified range. This range is defined by user-specified lower and upper bounds, ensuring that only SVs of interest are reported. Size bounds affect only the reporting and have no impact on the functioning or the efficiency of the algorithm.

3.5 Output

STRIVE outputs insertions, deletions, and inversions in three separate BED files, each containing chromosome name, start and end positions of the SV in the reference genome, and the size of the SV.

Chapter 4

Results

STRIVE is tested with two simulations to analyze the impacts of parameters and robustness to the presence of other mutations. A third simulation is used to compare STRIVE with other tools. Lastly, real biological data is used to test STRIVE's performance in a real scenario comparatively to other tools.

4.1 Simulations

In this section, there are two simulation experiments to test STRIVE. The first consists of insertions, deletions, and inversions. The second one introduces SNPs and duplications, creating a more realistic experiment. In both experiments, the first chromosome from GRCh38.p14 (hg38) assembly is used. Simulation results are created utilizing the overlap method of BEDTools [52].

4.1.1 Insertions, Deletions and Inversions

A simulation on the first chromosome of GRCh38.p14 assembly is generated using RSVSim [53]. The simulation includes 100 insertions, deletions, and inversions.

The size of insertions and deletions ranges from 200 bp to 100 Kbp. The size of inversions ranges from 600 bp to 1 Mbp. The histograms of the SV sizes are shown in Figure 4.1.

4.1.1.1 Effects of w and k

To understand the effects of w and k on STRIVE, combinations of several values are tested while other parameters are set to their default values. The default values for the other parameters are given in Table 4.1.

Table 4.1: Default parameters for the evaluation of w and k in the first simulation

Parameter	**Default Value**
Q_{min}	10
m_{max}	1
C_{min}	0.75

Generation of minimizers from the reference genome and the position file is done only once for a reference genome. Thus, preprocessing is considered only to align the minimizer sequences to the query genome. Using minimap2 [23], generated minimizers are aligned to the first chromosome of the human genome. As w increases, the number of generated minimizers decreases linearly, as STRIVE prefers non-overlapping windows, as discussed in Section 3.1, resulting in less time and memory usage for preprocessing. The effect of an increase in k has a similar relation with preprocessing time and memory usage, especially on time. These relations are demonstrated in Table 4.2 and 4.3.

Table 4.2: Preprocessing times (in seconds) for different w and k values.

	w = 50	**w = 100**	**w = 250**	**w = 1000**
k = 64	99.741	31.464	17.114	9.943
k = 128	182.035	53.494	26.549	12.281
k = 256	235.440	104.979	49.114	17.832

The execution time and memory usage of STRIVE are influenced by w and k as discussed in Section 3.4.1. Specifically, the number of iterations of the main loop of the algorithm is $\frac{Genome\ size}{w}$, resulting in an inverse linear relationship

Table 4.3: Memory usage (in GB) in preprocessing for different w and k values.

	w = 50	**w = 100**	**w = 250**	**w = 1000**
k = 64	3.014	2.481	2.053	1.971
k = 128	3.316	2.537	1.988	1.971
k = 256	2.888	2.776	1.995	1.971

between w and the execution time. A similar relation is also seen between w and memory usage of Strive. The time complexity and memory usage are increasing as k increases. This can be explained by the decrease in coincidental alignments, hence processing more information. The impacts of w and k are shown in Tables 4.4 and 4.5.

Table 4.4: Execution times (in seconds) of Strive for different w and k values.

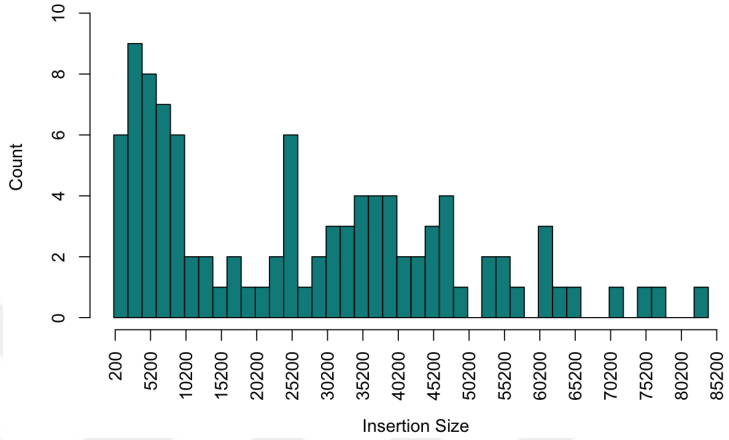
	w = 50	**w = 100**	**w = 250**	**w = 1000**
k = 64	20.54	10.22	4.31	1.18
k = 128	22.26	11.23	4.76	1.62
k = 256	24.45	12.53	5.29	1.56

Table 4.5: Memory usage (in GB) of Strive for different w and k values.

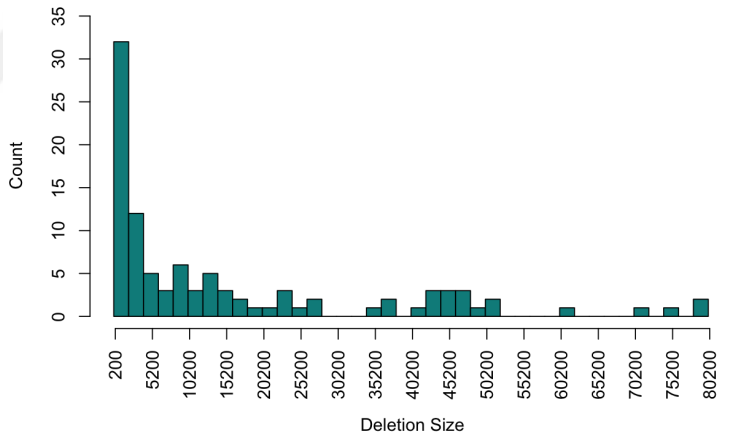
	w = 50	**w = 100**	**w = 250**	**w = 1000**
k = 64	0.222	0.113	0.048	0.015
k = 128	0.254	0.129	0.054	0.016
k = 256	0.265	0.134	0.056	0.017

The effects of w and k on the effectiveness of Strive is evaluated focused on precision, recall, and f1-score values of insertion, deletion, and inversion discoveries. A discovery is accepted as a true positive when the reciprocal overlap between the reported and truth interval exceeds a threshold. To evaluate this simulation, lower bounds for reciprocal overlaps are decided as 50%, 75%, and 90%. The breakpoints of the reported SVs can be as off as the distance between two consecutive minimizers and are affected directly by w as mentioned in Section 3.4.1.1. Increasing w results in reporting larger intervals hence less number of true positives. This effect is demonstrated empirically by this simulation and can be seen in Figures 4.2, 4.3, 4.4.

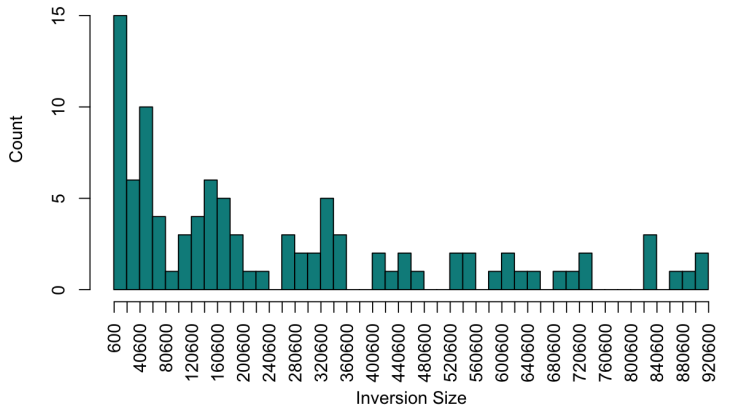
Increase in k results in more unique alignments, hence less coincidental alignments as discussed in Section 3.4.1.2. However, the simulation indicates that having relatively too small w compared to k results in an increase of false positives that can be seen in Figures 4.2(i), 4.3(i), 4.4(i).



(a) Histogram of insertion sizes in the first simulation

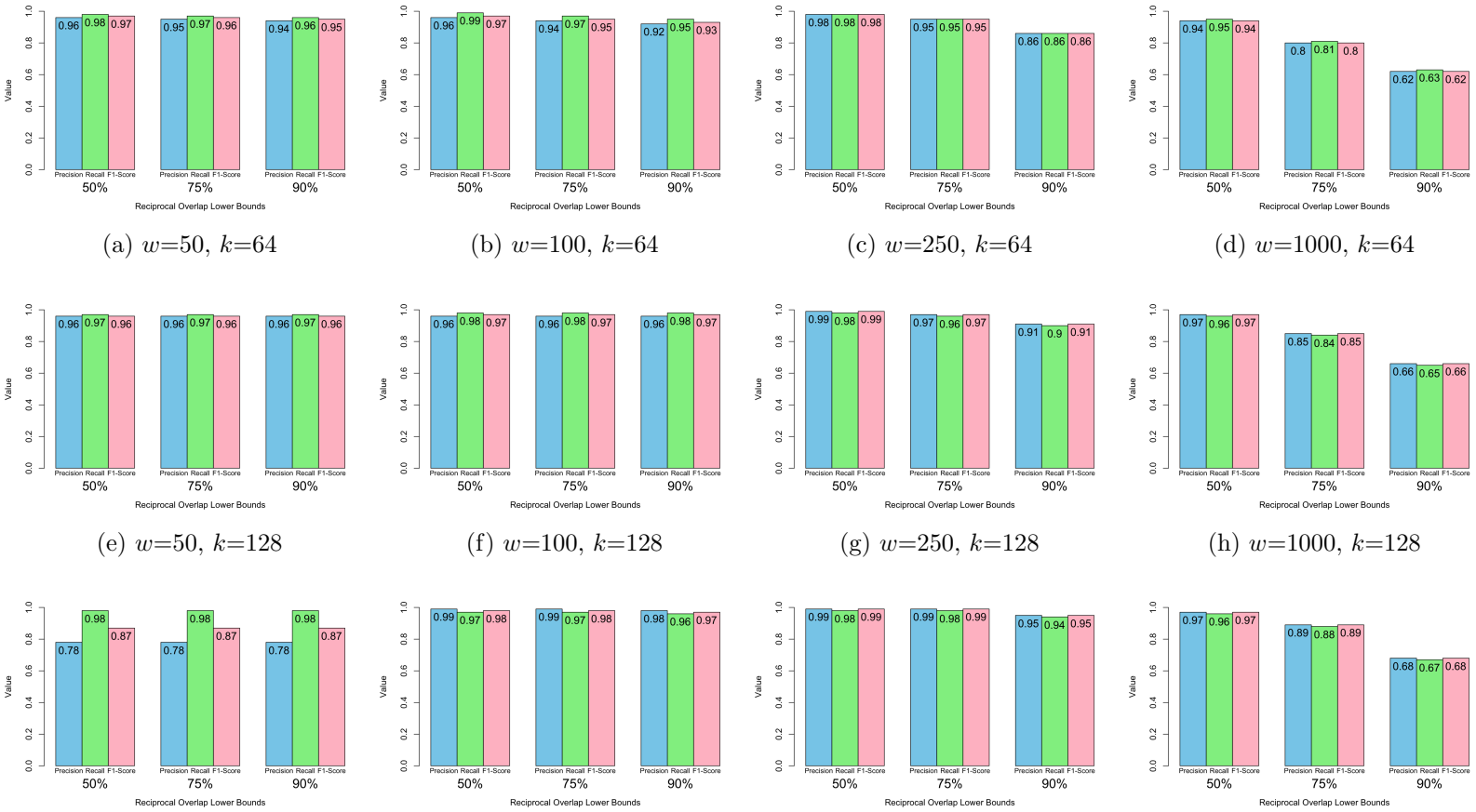


(b) Histogram of deletion sizes in the first simulation



(c) Histogram of inversion sizes in the first simulation

Figure 4.1: Histograms of the SV sizes in the first simulation

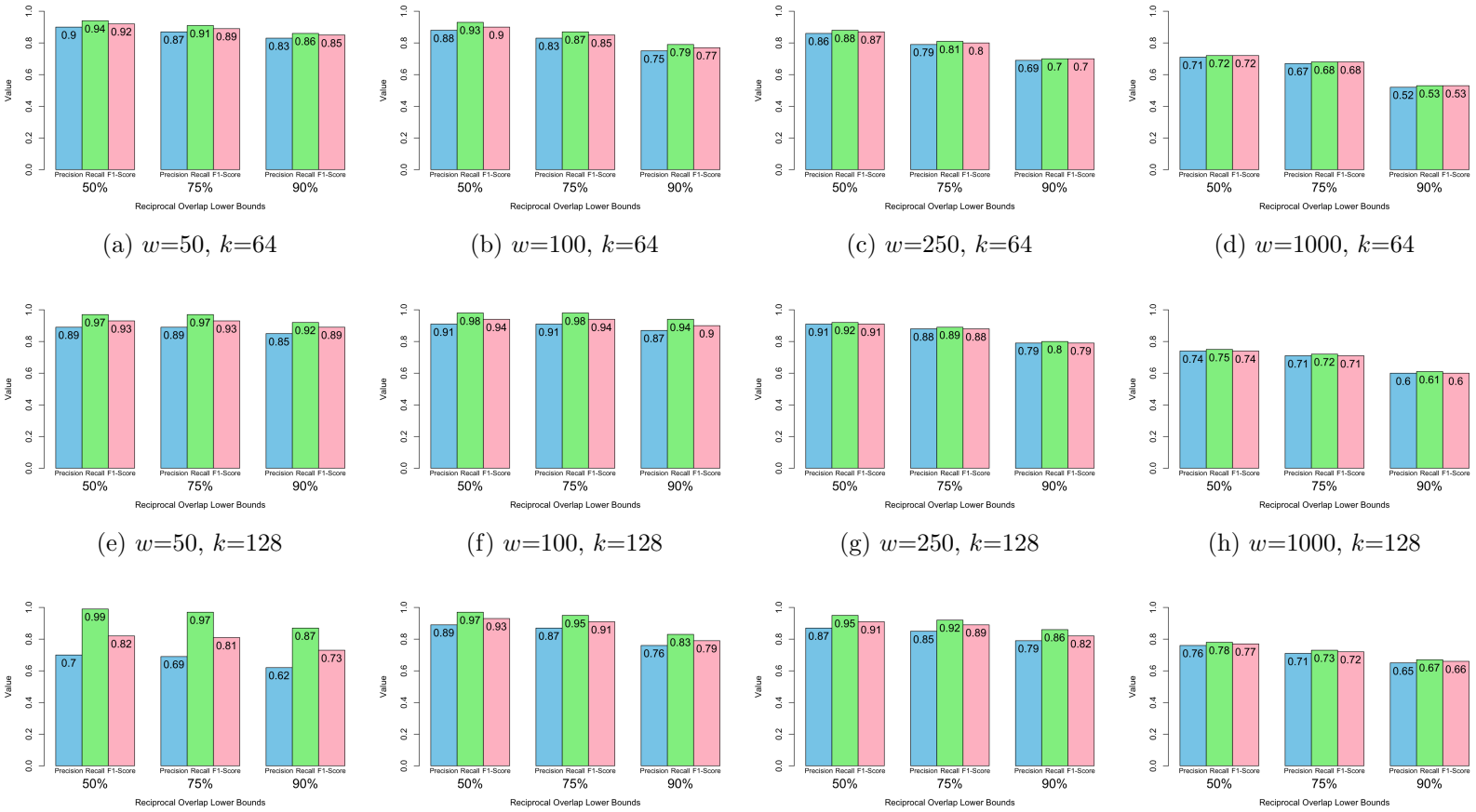


(a) w=50, k=64 (b) w=100, k=64 (c) w=250, k=64 (d) w=1000, k=64

(e) w=50, k=128 (f) w=100, k=128 (g) w=250, k=128 (h) w=1000, k=128

(i) w=50, k=256 (j) w=100, k=256 (k) w=250, k=256 (l) w=1000, k=256

Figure 4.2: Precision, Recall, and F1-Score values for insertion discovery in the first simulation

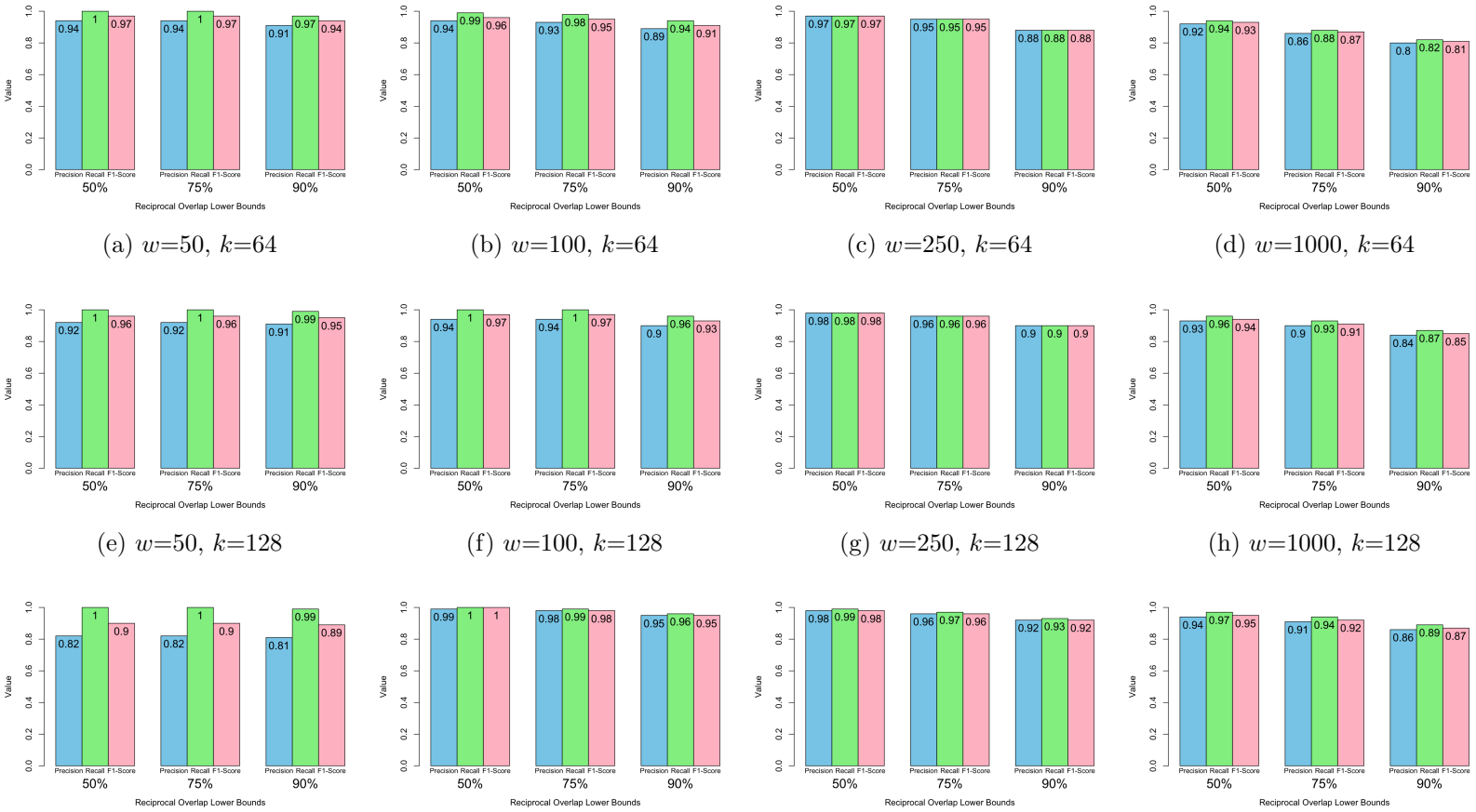


(a) w=50, k=64 (b) w=100, k=64 (c) w=250, k=64 (d) w=1000, k=64

(e) w=50, k=128 (f) w=100, k=128 (g) w=250, k=128 (h) w=1000, k=128

(i) w=50, k=256 (j) w=100, k=256 (k) w=250, k=256 (l) w=1000, k=256

Figure 4.3: Precision, Recall, and F1-Score values for deletion discovery in the first simulation



(a) w=50, k=64 (b) w=100, k=64 (c) w=250, k=64 (d) w=1000, k=64

(e) w=50, k=128 (f) w=100, k=128 (g) w=250, k=128 (h) w=1000, k=128

(i) w=50, k=256 (j) w=100, k=256 (k) w=250, k=256 (l) w=1000, k=256

Figure 4.4: Precision, Recall, and F1-Score values for inversion discovery in the first simulation

4.1.1.2 Effects of the other parameters

Based on the results of the previous section, the choices for w and k are 100 and 128, respectively. The following parameters have no influence on preprocessing; therefore, assessments are only done on the execution of STRIVE. Simulations on a parameter are done while setting the other parameters to default values provided in Table 4.1.

1. Q_{min}: The simulation results indicate that even a modest threshold, such as 6—allowing alignments with a probability of being false as low as 25%—on the mapping quality of minimizer sequence alignments to the query genome yields satisfactory outcomes. However, imposing a high threshold on mapping quality, i.e. too large Q_{min}, can slightly degrade performance, as illustrated in Figure 4.6. This observation supports the tradeoff about disregarding alignments discussed in Section 3.4.2. Furthermore, there is no evident correlation between Q_{min} and the execution time of STRIVE. Nevertheless, as shown in Figure 4.5, the memory usage tends to decrease as Q_{min} increases.

2. m_{max}: The simulation results suggest that permitting mismatches either has a negligible impact on the effectiveness of STRIVE or leads to a slight reduction, as illustrated in Figure 4.7. This observation is expected as no SNPs are introduced in the simulation. Yet, this result also validates the choice for the value k since any abundance in coincidental alignments would come to the surface as they are more likely to include mismatches.

3. C_{min} This parameter is introduced to overcome issues with coincidental alignments. The results show that the effectiveness of STRIVE is not impacted by the closeness parameter in this simulation. Like the observations about m_{max}, the observations illustrated in Figure 4.8 also validate the choice for the value of k.

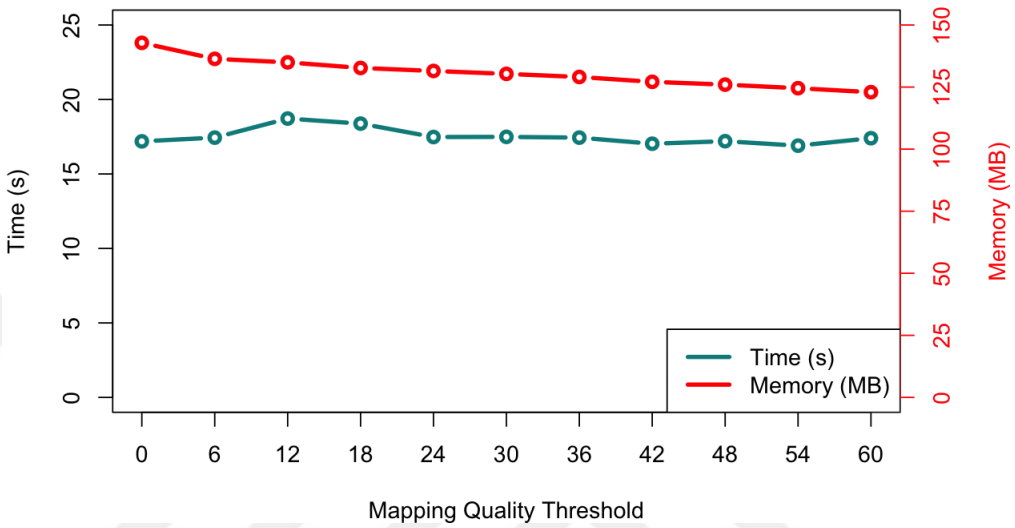


Figure 4.5: Time and memory usage of STRIVE with different Q_{min} values

4.1.2 Presence of SNPs and Duplications

A simulation on the first chromosome of GRCh38.p14 assembly is generated using SURVIVOR [54]. The simulation includes insertions, deletions, inversions, duplications, and SNPs. The count for each type is provided in Table 4.6, and the histogram for the SV sizes is provided in Figure 4.9. The number of SNPs is around 0.1% of the size of the first chromosome. The parameters for STRIVE are set based on the discussions in **??** and are provided in Table 4.7.

Table 4.6: The count of mutations by type in the second simulation

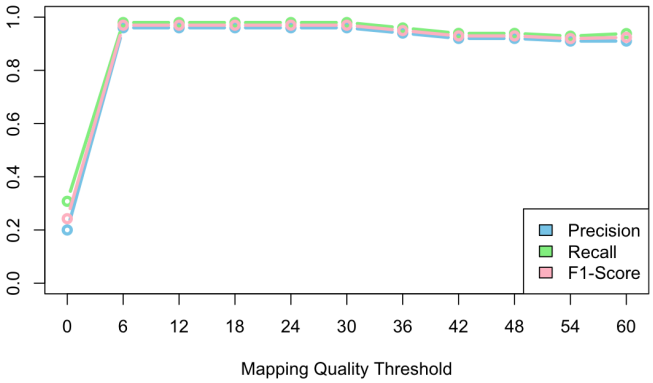
Type	Count
Insertion	52
Deletion	48
Inversion	104
Duplication	302
SNP	276841

SNPs can cause coincidental alignments, as discussed in Section 3.2. Its negative effect is observed in the discovery results provided in Figure 4.10. More significantly, the presence of duplications severely increases the number of false

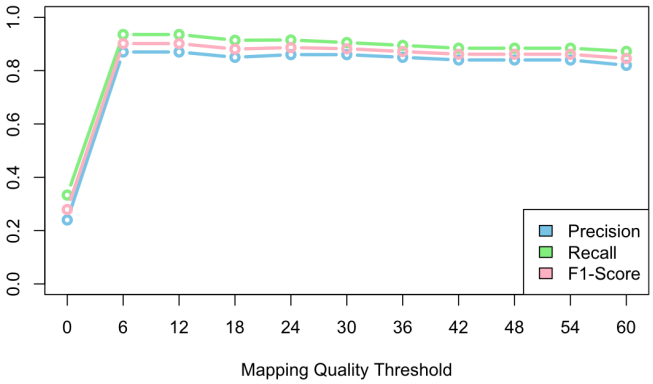
Table 4.7: Parameters for the second simulation

Parameter	**Default Value**
w	100
k	128
Q_{min}	10
m_{max}	1
C_{min}	0.75

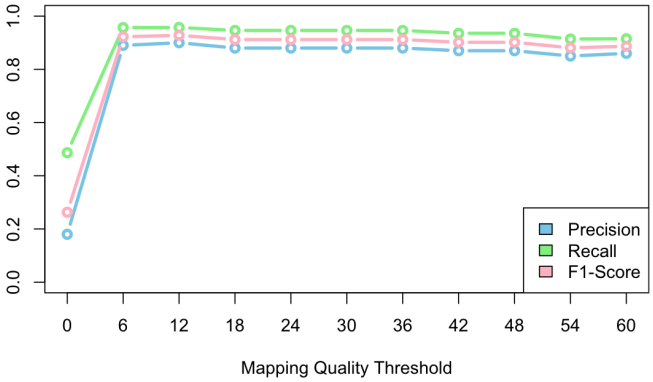
negatives for insertions, hence great reductions in recall values compared to the first simulation. STRIVE is unable to differentiate duplications from insertions; therefore, it reports them as insertions. Figure 4.10d demonstrates the statistics under this circumstance. It is observed that with higher reciprocal overlap thresholds, the precision and recall values decrease drastically. The reason behind these results is that STRIVE operates only on primary alignments. With duplicated regions, the aligner used in preprocessing decides on the primary alignment. If the last repeated alignment is selected as the primary alignment, STRIVE reports the interval correctly. In other cases, STRIVE reports a larger interval, hence the increase in true negatives and false positives in higher reciprocal overlap thresholds.



(a) Insertions

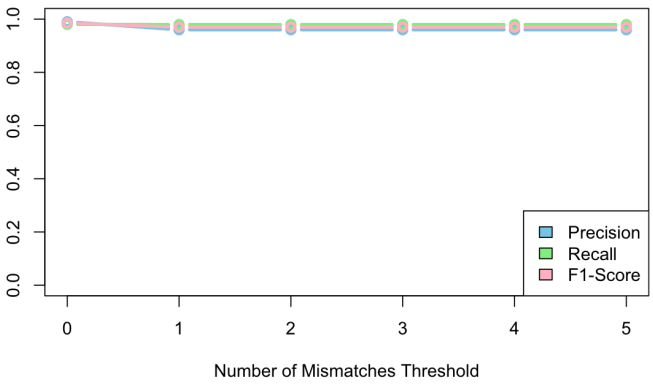


(b) Deletions

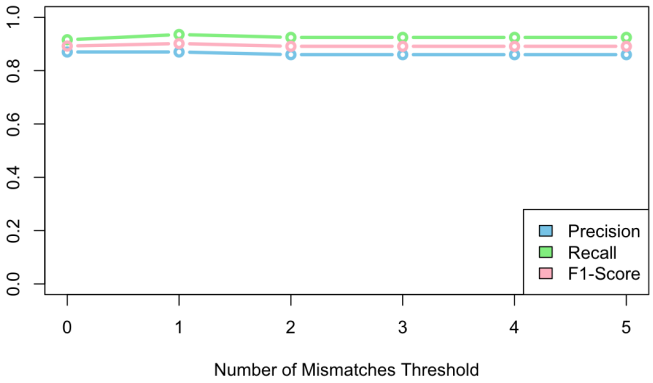


(c) Inversions

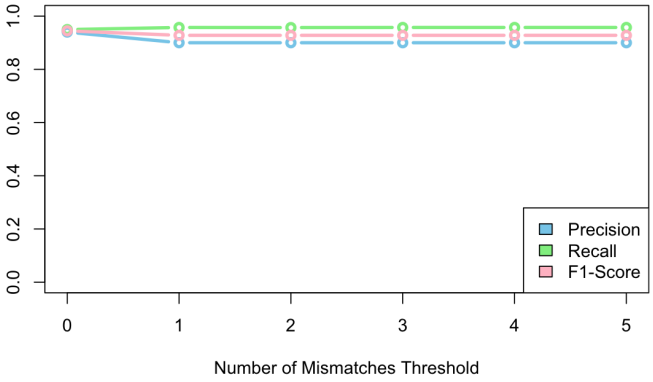
Figure 4.6: Precision, recall and F1-Score values with different Q_{min} values



(a) Insertions

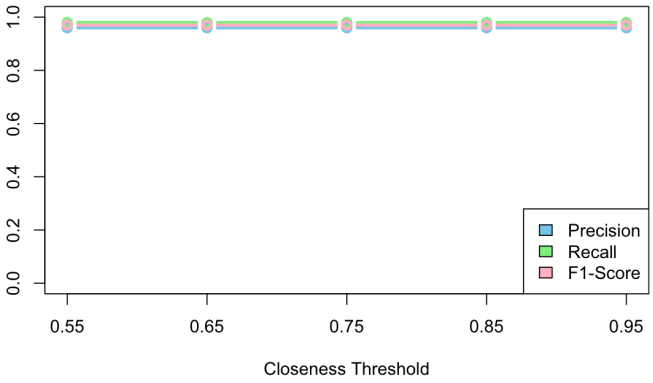


(b) Deletions

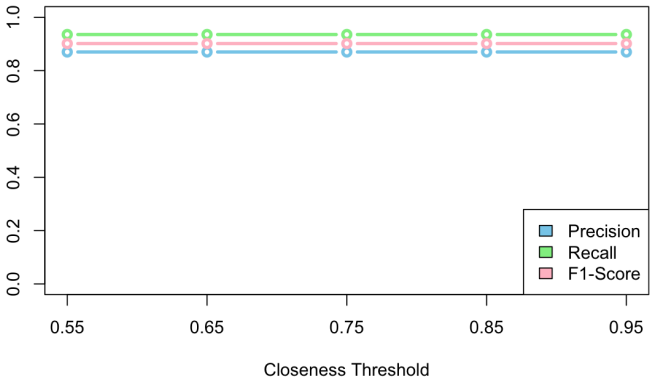


(c) Inversions

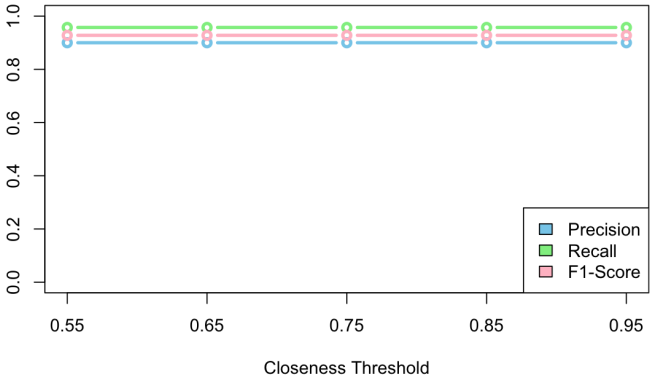
Figure 4.7: Precision, recall and F1-Score values with different m_{max} values



(a) Insertions

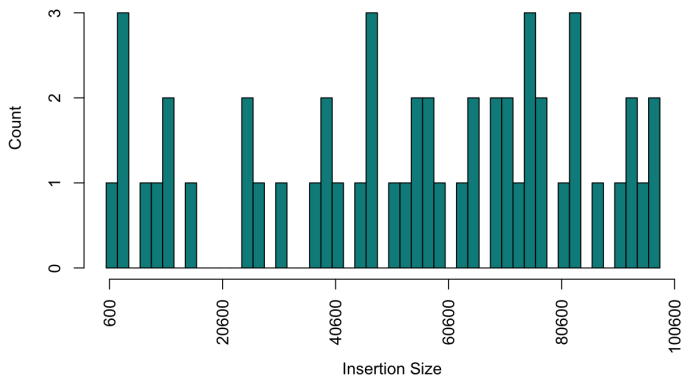


(b) Deletions

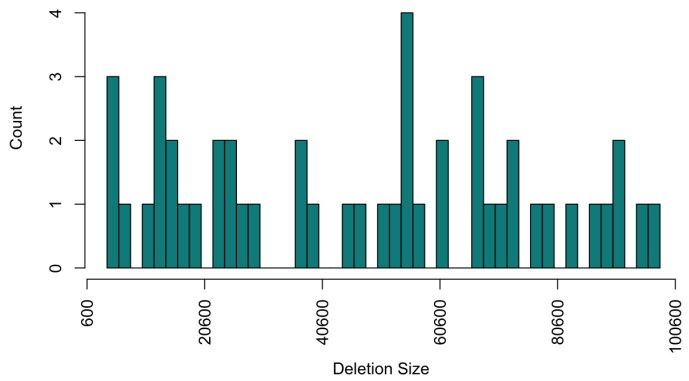


(c) Inversions

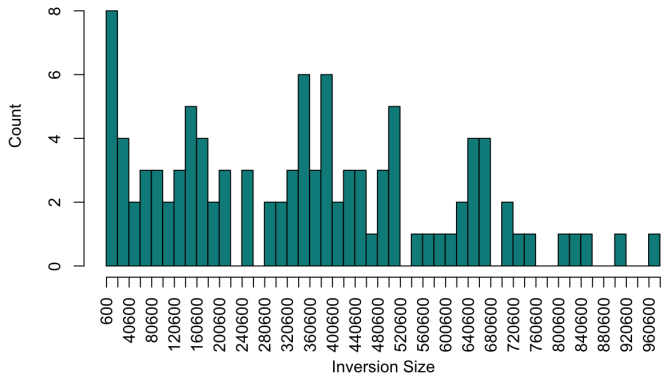
Figure 4.8: Precision, recall and F1-Score values with different C_{min} values



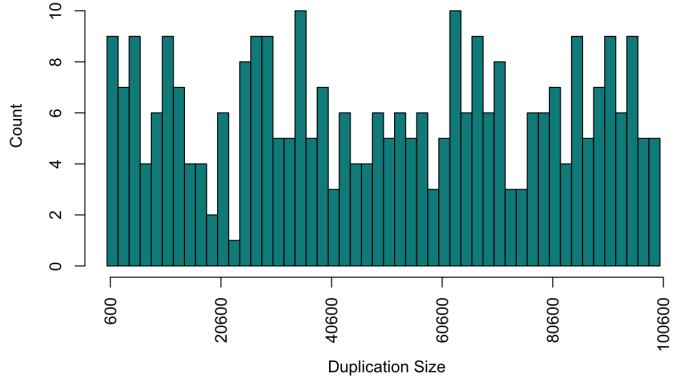
(a) Histogram of insertion sizes in the second simulation



(b) Histogram of deletion sizes in the second simulation

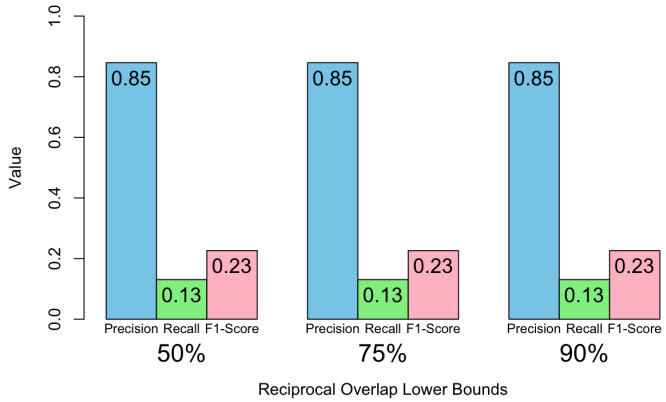


(c) Histogram of inversion sizes in the second simulation

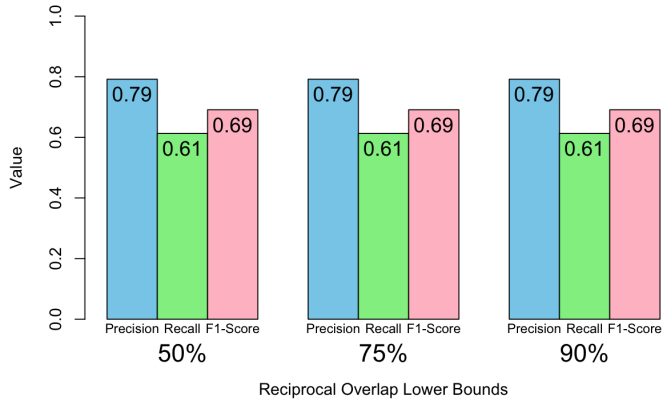


(d) Histogram of duplication sizes in the second simulation

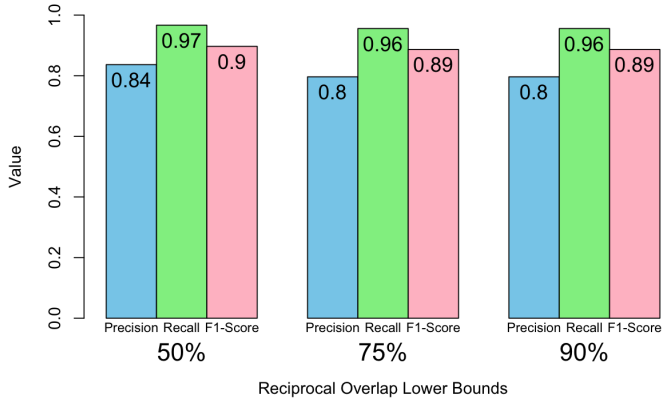
Figure 4.9: Histograms of the SV sizes in the second simulation



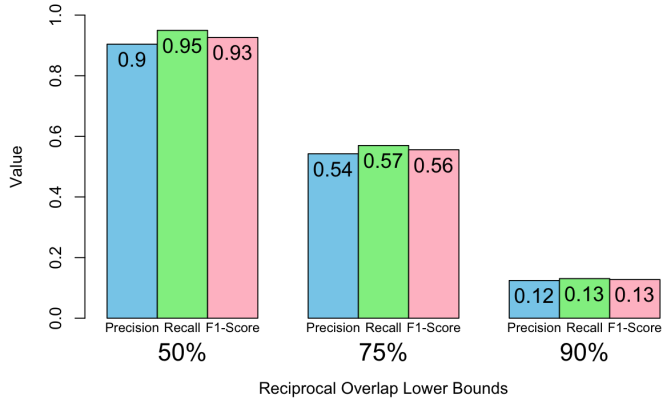
(a) Insertions



(b) Deletions



(c) Inversions



(d) Insertions with duplications as insertions

Figure 4.10: Precision, Recall, and F1-Score values for SV discovery in the second simulation

4.2 Comparisons

4.2.1 Simulation

A simulation on the chromosome 21 of CHM13 assembly is generated using SURVIVOR [54] to compare Strive against SVIM-asm [55] and SyRI [56] in terms of time complexity, memory usage and effectiveness. The simulation includes insertions, deletions, inversions, duplications, and SNPs. The counts and sizes are provided in Table 4.8. The parameters for Strive are given in Table 4.9.

Table 4.8: Mutation counts and sizes in the third simulation

Type	**Count**	**Sizes**
Insertion	14	3146, 6649, 11218, 25664, 26390, 34270, 40988, 45029, 55820, 71059, 78153, 80250, 87404, 89532
Deletion	16	2821, 6742, 6853, 13495, 22239, 24745, 29770, 38193, 41231, 43893, 50957, 55214, 65190, 76182, 87696, 94896
Inversion	5	125299, 139868, 145958, 357021, 590730
Duplication	3	6918, 20770, 23944
SNP	54210	-

Table 4.9: Parameters of Strive for the third simulation

Parameter	**Value**
w	100
k	128
Q_{min}	60
m_{max}	1
C_{min}	0.75

In preprocessing, both SVIM-asm and SyRI require whole genome alignment. The costliness of this process is discussed in Section 2.5.4 and observed in the simulation results in Table 4.10. Strive requires significantly less time and memory in preprocessing.

Strive produced results in less time and with less memory usage than SVIM-asm and SyRI as demonstrated in 4.11.

Table 4.10: Preprocessing times and memory usage of Strive and the other tools in the third simulation

Tool	Time(s)	Memory (GB)
Strive	17.24	0.532
SVIM-asm	108.35	2.861
SyRI	106.93	2.407

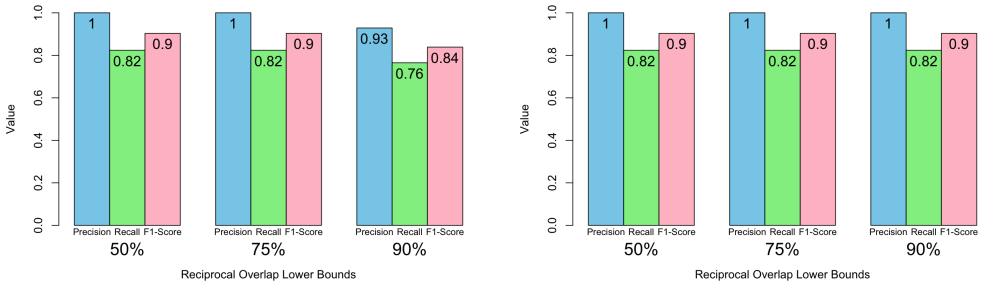
Table 4.11: Execution times and memory usage of Strive and the other tools in the third simulation

Tool	Time(s)	Memory (GB)
Strive	3.73	0.024
SVIM-asm	4.49	0.307
SyRI	9.07	0.356

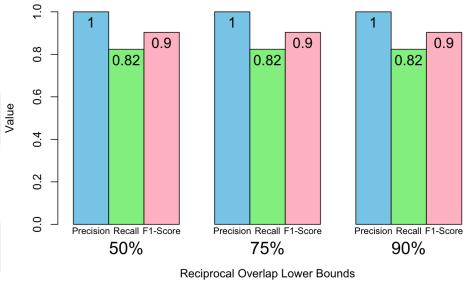
In insertion discovery, all tools performed similarly as shown in Figure 4.11. However, Strive underperformed the others regarding breakpoint refinement, leading to lower precision and recall at the 90% reciprocal overlap threshold. In deletion discovery, SVIM-asm slightly outperformed Strive and SyRI, as shown in Figure 4.12. The same issue with breakpoint refinement in Strive was observed again at the 90% reciprocal overlap threshold. Notably, SVIM-asm did not detect any inversions, whereas Strive and SyRI successfully identified all inversions, as shown in Figure 4.13.

4.2.2 Biological Data

Strive, SVIM-asm [55] and SyRI [56] are used to find the SVs between the first chromosomes of CHM13 and GRCh38.p14 (hg38) assemblies. The performance is evaluated using the validated 15 insertions, 5 deletions, and 6 inversions provided in [2]. The counts and sizes are provided in Table 4.12. The parameters for Strive are given in Table 4.13. The parameters are selected considering the possible complexity of the real assembly data, i.e. all the measures to overcome coincidental alignments and mapping errors explained in Section 3.4.2 are taken.



(a) STRIVE (b) SVIM-asm

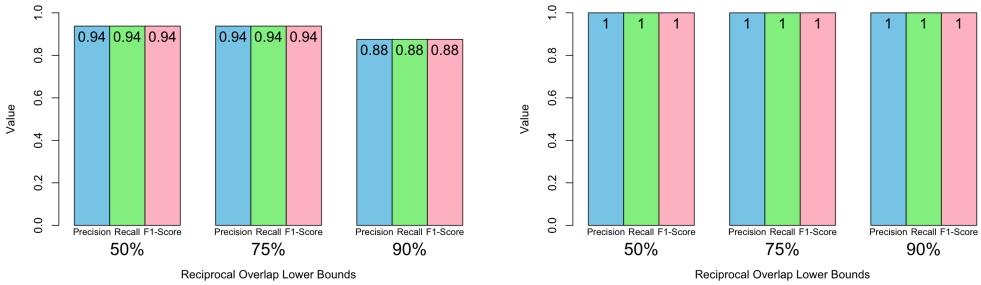


(c) SyRI

Figure 4.11: Insertion performances in the third simulation

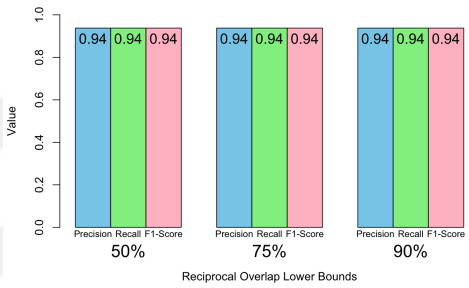
STRIVE required significantly less time and memory for preprocessing. Similar to the results in Section 4.2.1, the discussion about the costliness of whole genome alignment in Section 2.5.4 is empirically shown in Table 4.14. In terms of execution time, SVIM-asm outperformed STRIVE and SyRI. However, STRIVE required significantly less memory in execution. Execution times and memory usage are given in Table 4.15.

Since only the validated SVs are provided in [2], only true positive discoveries are considered. A discovery is considered a true positive if it has a reciprocal overlap of more than 75%. In insertion discovery, SVIM-asm and SyRI outperformed STRIVE as shown in Table 4.16. Tools performed similarly in deletion discovery as shown in Table 4.17. STRIVE performed better than SyRI and SVIM-asm in inversion discovery as shown in Table 4.18.



(a) STRIVE

(b) SVIM-asm



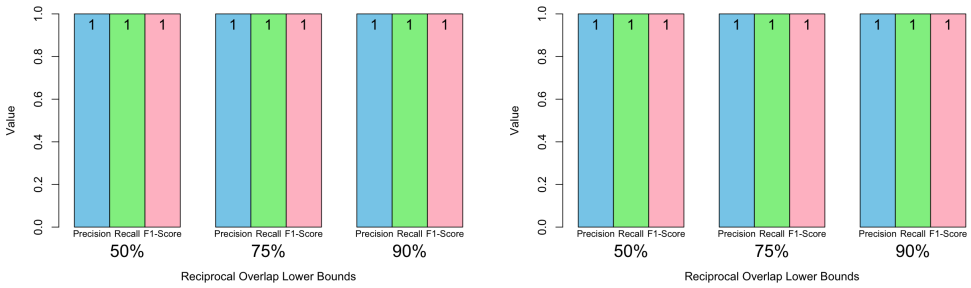
(c) SyRI

Figure 4.12: Deletion performances in the third simulation

4.3 Discussion

STRIVE demonstrates comparable effectiveness to other whole-genome assembly-based tools like SVIM-asm and SyRI but without the disadvantage of requiring whole-genome alignment. Therefore, the preprocessing requires significantly less time and memory. For relatively small genomes like chromosome 21 of the human genome, which is of length approximately 45 Mbp, offering execution times better than SVIM-asm and SyRI, whereas for larger genomes like chromosome 1 of the human genome, which is of length approximately 248 Mbp, offering competitive execution times to SyRI while SVIM-asm outperforms the both. Even for large genomes, STRIVE uses 90% less memory than the other tools.

The insertion discovery results of the experiments showed that STRIVE is in dire need of robustness to the presence of duplications. Furthermore, the future for STRIVE should include characterizing other SV types.



(a) Strive (b) SyRI

Figure 4.13: Inversion performances in the third simulation

Table 4.12: Validated insertion, deletion, and inversion SV counts and sizes between CHM13 and GRCh38.p14 (hg38) assemblies [2]

Type	**Count**	**Sizes**
Insertion	15	10600, 10667, 11286, 20424, 24019, 29570, 81762, 86400, 88092, 89522, 92112, 95967, 98513, 109052, 165367
Deletion	5	18444, 18555, 45515, 49841, 84863
Inversion	6	4947, 241972, 253970, 269345, 575584, 2465027

Table 4.13: Parameters of Strive for the experiment with biological data

Parameter	**Value**
w	50
k	128
Q_{min}	60
m_{max}	0
C_{min}	0.99

Table 4.14: Preprocessing times and memory usage of Strive and the other tools in the experiment with biological data

Tool	**Time(s)**	**Memory (GB)**
Strive	92.7	3.239
SVIM-asm	739.99	11.431
SyRI	693.19	8.342

Table 4.15: Execution times and memory usage of STRIVE and the other tools in the experiment with biological data

Tool	**Time(s)**	**Memory (GB)**
STRIVE	37.11	0.194
SVIM-asm	7.63	1.283
SyRI	38.07	1.366

Table 4.16: Characterized validated insertions in Chromosome 1 between CHM13 and GRCh38.p14

	Characterized by		
Size	STRIVE	SVIM-asm	SyRI
10600	✓	✓	✓
10667	✓	✓	✓
11286	✓	✓	
20424		✓	✓
24019	✓	✓	✓
29570		✓	✓
81762	✓	✓	✓
86400		✓	
88092		✓	
89522		✓	✓
92112			✓
95967	✓	✓	✓
98513		✓	✓
109052			✓
165367			
Total	6	12	11

Table 4.17: Characterized validated deletions in Chromosome 1 between CHM13 and GRCh38.p14

	Characterized by		
Size	STRIVE	SVIM-asm	SyRI
18444	✓	✓	✓
18555	✓	✓	✓
45515	✓	✓	✓
49841	✓		
84683		✓	✓
Total	4	4	4

Table 4.18: Characterized validated inversions in Chromosome 1 between CHM13 and GRCh38.p14

	Characterized by		
Size	STRIVE	SVIM-asm	SyRI
4947	✓	✓	✓
241972	✓		✓
253970	✓		
269345			
575584	✓		✓
2465027	✓		✓
Total	5	1	4

Chapter 5

Future Work

The following aspects can be targeted for enhancing Strive.

1. **Characterization of other SV types**
   Strive is proficient in detecting three types of SVs: insertions, deletions, and inversions. These variants are essential for understanding the structural alterations in a genome. However, the algorithm can be extended to include other types of SVs, such as duplications and translocations, which would broaden its applicability.

2. **Increased robustness to duplications in insertion discovery**
   Strive has limitations in distinguishing duplications from insertions, which are often confused due to their similar nature of structural changes. Both duplications and insertions involve the presence of additional sequences in the genome, but duplications refer to repeated segments of an existing sequence, whereas insertions involve new sequences being introduced into the genome. To improve the differentiation between duplications and insertions, Strive can be enhanced by incorporating secondary alignments into its input. Secondary alignments provide information on multiple possible alignments of sketches, particularly useful in repetitive regions of the genome.

3. **Support for diploid assemblies**

   STRIVE operates effectively in the context of haploid genome assemblies, where a single set of chromosomes is used for variant detection. However, most complex organisms, such as humans, possess diploid genomes, meaning they have two sets of chromosomes—one inherited from each parent. The presence of these two sets introduces genetic variations between the two chromosome copies, known as heterozygous variants. Incorporating diploid assembly capabilities would allow STRIVE to handle heterozygous variants. Moreover, this improvement would increase the accuracy of identifying SVs in diploid genomes by preserving both sets of genetic information.

4. **Characterization of interchromosomal SVs**

   SVs that span across different chromosomes (interchromosomal SVs), such as translocations, are often harder to detect due to their complexity. Adding characterization of such SVs to STRIVE's capacity would provide a more detailed understanding of chromosomal rearrangements, especially in cases like cancer genomics where interchromosomal SVs are crucial [57].

5. **Concurrency**

   Introducing concurrency support may enhance STRIVE's performance in terms of execution time, especially when analyzing large datasets such as whole-genome assemblies. By implementing parallelization techniques and optimizing multi-threaded execution, the algorithm can scale effectively.

Bibliography

[1] C. Alkan, B. P. Coe, and E. E. Eichler, "Genome structural variation discovery and genotyping," *Nature Reviews Genetics*, vol. 12, pp. 363–376, May 2011.

[2] X. Yang, X. Wang, Y. Zou, S. Zhang, M. Xia, L. Fu, M. R. Vollger, N.-C. Chen, D. J. Taylor, W. T. Harvey, G. A. Logsdon, D. Meng, J. Shi, R. C. McCoy, M. C. Schatz, W. Li, E. E. Eichler, Q. Lu, and Y. Mao, "Characterization of large-scale genomic differences in the first complete human genome," *Genome Biology*, vol. 24, p. 157, July 2023.

[3] International Human Genome Sequencing Consortium, "Initial sequencing and analysis of the human genome," *Nature*, vol. 409, pp. 860–921, Feb. 2001.

[4] The 1000 Genomes Project Consortium, "A global reference for human genetic variation," *Nature*, vol. 526, pp. 68–74, Sept. 2015.

[5] The International SNP Map Working Group, "A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms," *Nature*, vol. 409, pp. 928–933, Feb. 2001.

[6] The International HapMap Consortium, "The international HapMap project," *Nature*, vol. 426, pp. 789–796, Dec. 2003.

[7] T. . G. P. Consortium, "An integrated map of genetic variation from 1,092 human genomes," *Nature*, vol. 491, pp. 56–65, Nov. 2012.

[8] C. Alkan, P. Kavak, M. Somel, O. Gokcumen, S. Ugurlu, C. Saygi, E. Dal, K. Bugra, T. Güngör, S. C. Sahinalp, N. Ozören, and C. Bekpen, "Whole genome sequencing of turkish genomes reveals functional private alleles and impact of genetic interactions with Europe, Asia and Africa," *BMC Genomics*, vol. 15, no. 1, p. 963, 2014.

[9] P. H. Sudmant, T. Rausch, E. J. Gardner, R. E. Handsaker, A. Abyzov, J. Huddleston, Y. Zhang, K. Ye, G. Jun, M. Hsi-Yang Fritz, M. K. Konkel, A. Malhotra, A. M. Stütz, X. Shi, F. Paolo Casale, J. Chen, F. Hormozdiari, G. Dayama, K. Chen, M. Malig, M. J. P. Chaisson, K. Walter, S. Meiers, S. Kashin, E. Garrison, A. Auton, H. Y. K. Lam, X. Jasmine Mu, C. Alkan, D. Antaki, T. Bae, E. Cerveira, P. Chines, Z. Chong, L. Clarke, E. Dal, L. Ding, S. Emery, X. Fan, M. Gujral, F. Kahveci, J. M. Kidd, Y. Kong, E.-W. Lameijer, S. McCarthy, P. Flicek, R. A. Gibbs, G. Marth, C. E. Mason, A. Menelaou, D. M. Muzny, B. J. Nelson, A. Noor, N. F. Parrish, M. Pendleton, A. Quitadamo, B. Raeder, E. E. Schadt, M. Romanovitch, A. Schlattl, R. Sebra, A. A. Shabalin, A. Untergasser, J. A. Walker, M. Wang, F. Yu, C. Zhang, J. Zhang, X. Zheng-Bradley, W. Zhou, T. Zichner, J. Sebat, M. A. Batzer, S. A. McCarroll, . G. P. C. , R. E. Mills, M. B. Gerstein, A. Bashir, O. Stegle, S. E. Devine, C. Lee, E. E. Eichler, and J. O. Korbel, "An integrated map of structural variation in 2,504 human genomes," *Nature*, vol. 526, pp. 75–81, Sept. 2015.

[10] J. Weischenfeldt, O. Symmons, F. Spitz, and J. O. Korbel, "Phenotypic impact of genomic structural variation: insights from and for human disease," *Nature Reviews Genetics*, vol. 14, pp. 125–138, Feb. 2013.

[11] T. J. Treangen and S. L. Salzberg, "Repetitive DNA and next-generation sequencing: computational challenges and solutions," *Nature Reviews Genetics*, vol. 13, pp. 36–46, Jan. 2012.

[12] H. Li, J. Ruan, and R. Durbin, "Mapping short DNA sequencing reads and calling variants using mapping quality scores," *Genome Research*, vol. 18, pp. 1851–1858, Nov. 2008.

[13] M. J. P. Chaisson, R. K. Wilson, and E. E. Eichler, "Genetic variation and the de novo assembly of human genomes.," *Nature Reviews Genetics*, vol. 16, pp. 627–640, Nov. 2015.

[14] C. Alkan, L. Carbone, M. Dennis, J. Ernst, G. Evrony, S. Girirajan, D. C. Y. Leung, C. C. Cheng, D. MacAlpine, T. Ni, M. Ramsay, H. Rowe, P. Gould, R. Enriquez-Gasca, and B. Sullivan, "Implications of the first complete human genome assembly.," *Genome research*, vol. 32, pp. 595–598, Apr. 2022.

[15] B. Saada, T. Zhang, E. Siga, J. Zhang, and M. M. Magalhães Muniz, "Whole-genome alignment: Methods, challenges, and future directions," *Applied Sciences*, vol. 14, no. 11, 2024.

[16] S. Nurk, S. Koren, A. Rhie, M. Rautiainen, A. V. Bzikadze, A. Mikheenko, M. R. Vollger, N. Altemose, L. Uralsky, A. Gershman, S. Aganezov, S. J. Hoyt, M. Diekhans, G. A. Logsdon, M. Alonge, S. E. Antonarakis, M. Borchers, G. G. Bouffard, S. Y. Brooks, G. V. Caldas, N.-C. Chen, H. Cheng, C.-S. Chin, W. Chow, L. G. de Lima, P. C. Dishuck, R. Durbin, T. Dvorkina, I. T. Fiddes, G. Formenti, R. S. Fulton, A. Fungtammasan, E. Garrison, P. G. S. Grady, T. A. Graves-Lindsay, I. M. Hall, N. F. Hansen, G. A. Hartley, M. Haukness, K. Howe, M. W. Hunkapiller, C. Jain, M. Jain, E. D. Jarvis, P. Kerpedjiev, M. Kirsche, M. Kolmogorov, J. Korlach, M. Kremitzki, H. Li, V. V. Maduro, T. Marschall, A. M. McCartney, J. McDaniel, D. E. Miller, J. C. Mullikin, E. W. Myers, N. D. Olson, B. Paten, P. Peluso, P. A. Pevzner, D. Porubsky, T. Potapova, E. I. Rogaev, J. A. Rosenfeld, S. L. Salzberg, V. A. Schneider, F. J. Sedlazeck, K. Shafin, C. J. Shew, A. Shumate, Y. Sims, A. F. A. Smit, D. C. Soto, I. Sović, J. M. Storer, A. Streets, B. A. Sullivan, F. Thibaud-Nissen, J. Torrance, J. Wagner, B. P. Walenz, A. Wenger, J. M. D. Wood, C. Xiao, S. M. Yan, A. C. Young, S. Zarate, U. Surti, R. C. McCoy, M. Y. Dennis, I. A. Alexandrov, J. L. Gerton, R. J. O'Neill, W. Timp, J. M. Zook, M. C. Schatz, E. E. Eichler, K. H. Miga, and A. M. Phillippy, "The complete sequence of a human genome.," *Science*, vol. 376, pp. 44–53, Apr. 2022.

[17] M. Rautiainen, S. Nurk, B. P. Walenz, G. A. Logsdon, D. Porubsky, A. Rhie, E. E. Eichler, A. M. Phillippy, and S. Koren, "Telomere-to-telomere assembly of diploid chromosomes with Verkko.," *Nature Biotechnology*, Feb. 2023.

[18] D. Yoo, A. Rhie, P. Hebbar, F. Antonacci, G. A. Logsdon, S. J. Solar, D. Antipov, B. D. Pickett, Y. Safonova, F. Montinaro, Y. Luo, J. Malukiewicz, J. M. Storer, J. Lin, A. N. Sequeira, R. J. Mangan, G. Hickey, G. M. Anez, P. Balachandran, A. Bankevich, C. R. Beck, A. Biddanda, M. Borchers, G. G. Bouffard, E. Brannan, S. Y. Brooks, L. Carbone, L. Carrel, A. P. Chan, J. Crawford, M. Diekhans, E. Engelbrecht, C. Feschotte, G. Formenti, G. H. Garcia, L. d. Gennaro, D. Gilbert, R. E. Green, A. Guarracino, I. Gupta, D. Haddad, J. Han, R. S. Harris, G. A. Hartley, W. T. Harvey, M. Hiller, K. Hoekzema, M. L. Houck, H. Jeong, K. Kamali, M. Kellis, B. Kille, C. Lee, Y. Lee, W. Lees, A. P. Lewis, Q. Li, M. Loftus, Y. H. E. Loh, H. Loucks, J. Ma, Y. Mao, J. F. I. Martinez, P. Masterson, R. C. McCoy, B. McGrath, S. McKinney, B. S. Meyer, K. H. Miga, S. K. Mohanty, K. M. Munson, K. Pal, M. Pennell, P. A. Pevzner, D. Porubsky, T. Potapova, F. R. Ringeling, J. L. Rocha, O. A. Ryder, S. Sacco, S. Saha, T. Sasaki, M. C. Schatz, N. J. Schork, C. Shanks, L. Smeds, D. R. Son, C. Steiner, A. P. Sweeten, M. G. Tassia, F. Thibaud-Nissen, E. Torres-González, M. Trivedi, W. Wei, J. Wertz, M. Yang, P. Zhang, S. Zhang, Y. Zhang, Z. Zhang, S. A. Zhao, Y. Zhu, E. D. Jarvis, J. L. Gerton, I. Rivas-González, B. Paten, Z. A. Szpiech, C. D. Huber, T. L. Lenz, M. K. Konkel, S. V. Yi, S. Canzar, C. T. Watson, P. H. Sudmant, E. Molloy, E. Garrison, C. B. Lowe, M. Ventura, R. J. O'Neill, S. Koren, K. D. Makova, A. M. Phillippy, and E. E. Eichler, "Complete sequencing of ape genomes," *bioRxiv*, 2024.

[19] N. Nagarajan and M. Pop, "Sequence assembly demystified," *Nature Reviews Genetics*, vol. 14, pp. 157–167, Mar. 2013.

[20] M. Burrows and D. J. Wheeler, "A block sorting lossless data compression algorithm," tech. rep., Digital Equipment Corporation, 1994.

[21] R. Li, Y. Li, X. Fang, H. Yang, J. Wang, K. Kristiansen, and J. Wang, "SNP detection for massively parallel whole-genome resequencing," *Genome*

Research, vol. 19, pp. 1124–1132, June 2009.

[22] B. Langmead and S. L. Salzberg, "Fast gapped-read alignment with Bowtie 2," *Nature Methods*, vol. 9, pp. 357–359, Mar. 2012.

[23] H. Li, "Minimap2: pairwise alignment for nucleotide sequences.," *Bioinformatics*, vol. 34, pp. 3094–3100, Sept. 2018.

[24] D. C. Schwartz, X. Li, L. I. Hernandez, S. P. Ramnarain, E. J. Huff, and Y.-K. Wang, "Ordered restriction maps of saccharomyces cerevisiae chromosomes constructed by optical mapping," *Science*, vol. 262, no. 5130, pp. 110–114, 1993.

[25] L. Li, A. K.-Y. Leung, T.-P. Kwok, Y. Y. Y. Lai, I. K. Pang, G. T.-Y. Chung, A. C. Y. Mak, A. Poon, C. Chu, M. Li, J. J. K. Wu, E. T. Lam, H. Cao, C. Lin, J. Sibert, S.-M. Yiu, M. Xiao, K.-W. Lo, P.-Y. Kwok, T.-F. Chan, and K. Y. Yip, "OMSV enables accurate and comprehensive identification of large structural variations from nanochannel-based single-molecule optical maps," *Genome Biology*, vol. 18, p. 230, Dec. 2017.

[26] P. Dremsek, T. Schwarz, B. Weil, A. Malashka, F. Laccone, and J. Neesen, "Optical genome mapping in routine human genetic Diagnostics-Its advantages and limitations," *Genes (Basel)*, vol. 12, Dec. 2021.

[27] G. Cormode, M. Garofalakis, P. J. Haas, and C. Jermaine, "Synopses for massive data: Samples, histograms, wavelets, sketches," *Foundations and Trends® in Databases*, vol. 4, no. 1–3, pp. 1–294, 2011.

[28] W. P. M. Rowe, "When the levee breaks: a practical guide to sketching algorithms for processing the flood of genomic data," *Genome Biology*, vol. 20, p. 199, Sep 2019.

[29] C. Jain, S. Koren, A. Dilthey, A. M. Phillippy, and S. Aluru, "A fast adaptive algorithm for computing whole-genome homology maps.," *Bioinformatics*, vol. 34, pp. i748–i756, Sept. 2018.

[30] R. Wittler, "General encoding of canonical k-mers," *bioRxiv*, 2024.

[31] M. Roberts, W. Hayes, B. R. Hunt, S. M. Mount, and J. A. Yorke, "Reducing storage requirements for biological sequence comparison.," *Bioinformatics*, vol. 20, pp. 3363–3369, Dec. 2004.

[32] H. Li, "Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences.," *Bioinformatics*, vol. 32, pp. 2103–2110, July 2016.

[33] K. Sahlin, "Effective sequence similarity detection with strobemers.," *Genome research*, vol. 31, pp. 2080–2094, Nov. 2021.

[34] R. Edgar, "Syncmers are more sensitive than minimizers for selecting conserved k -mers in biological sequences.," *PeerJ*, vol. 9, p. e10805, 2021.

[35] The International HapMap Consortium, "A second generation human haplotype map of over 3.1 million SNPs," *Nature*, vol. 449, pp. 851–861, Oct. 2007.

[36] R. E. Mills, C. T. Luttig, C. E. Larkins, A. Beauchamp, C. Tsui, W. S. Pittard, and S. E. Devine, "An initial map of insertion and deletion (INDEL) variation in the human genome," *Genome Research*, vol. 16, pp. 1182–1190, Sept. 2006.

[37] M. Gymrek, T. Willems, A. Guilmatre, H. Zeng, B. Markus, S. Georgiev, M. J. Daly, A. L. Price, J. K. Pritchard, A. J. Sharp, and Y. Erlich, "Abundant contribution of short tandem repeats to gene expression variation in humans," *Nature Genetics*, vol. 48, pp. 22–29, Jan. 2016.

[38] N. Wyner, M. Barash, and D. McNevin, "Forensic autosomal short tandem repeats and their potential association with phenotype," *Front Genet*, vol. 11, p. 884, Aug. 2020.

[39] E. Tuzun, A. J. Sharp, J. A. Bailey, R. Kaul, V. A. Morrison, L. M. Pertz, E. Haugen, H. Hayden, D. Albertson, D. Pinkel, M. V. Olson, and E. E. Eichler, "Fine-scale structural variation of the human genome," *Nature Genetics*, vol. 37, pp. 727–732, July 2005.

[40] A. J. Sharp, D. P. Locke, S. D. McGrath, Z. Cheng, J. A. Bailey, R. U. Vallente, L. M. Pertz, R. A. Clark, S. Schwartz, R. Segraves, V. V. Oseroff,

D. G. Albertson, D. Pinkel, and E. E. Eichler, "Segmental duplications and copy-number variation in the human genome," *American Journal of Human Genetics*, vol. 77, pp. 78–88, July 2005.

[41] K. K. Wong, R. J. deLeeuw, N. S. Dosanjh, L. R. Kimm, Z. Cheng, D. E. Horsman, C. MacAulay, R. T. Ng, C. J. Brown, E. E. Eichler, and W. L. Lam, "A comprehensive analysis of common copy-number variations in the human genome," *Am J Hum Genet*, vol. 80, pp. 91–104, Dec. 2006.

[42] J. Sebat, B. Lakshmi, D. Malhotra, J. Troge, C. Lese-Martin, T. Walsh, B. Yamrom, S. Yoon, A. Krasnitz, J. Kendall, A. Leotta, D. Pai, R. Zhang, Y.-H. Lee, J. Hicks, S. J. Spence, A. T. Lee, K. Puura, T. Lehtimäki, D. Ledbetter, P. K. Gregersen, J. Bregman, J. S. Sutcliffe, V. Jobanputra, W. Chung, D. Warburton, M.-C. King, D. Skuse, D. H. Geschwind, T. C. Gilliam, K. Ye, and M. Wigler, "Strong association of de novo copy number mutations with autism," *Science*, vol. 316, pp. 445–449, Mar. 2007.

[43] A. Guilmatre, C. Dubourg, A.-L. Mosca, S. Legallic, A. Goldenberg, V. Drouin-Garraud, V. Layet, A. Rosier, S. Briault, F. Bonnet-Brilhault, F. Laumonnier, S. Odent, G. Le Vacon, G. Joly-Helas, V. David, C. Bendavid, J.-M. Pinoit, C. Henry, C. Impallomeni, E. Germano, G. Tortorella, G. Di Rosa, C. Barthelemy, C. Andres, L. Faivre, T. Frébourg, P. Saugier Veber, and D. Campion, "Recurrent rearrangements in synaptic and neurodevelopmental genes and shared biologic pathways in schizophrenia, autism, and mental retardation," *Arch Gen Psychiatry*, vol. 66, pp. 947–956, Sept. 2009.

[44] M. Poot, J. J. van der Smagt, E. H. Brilstra, and T. Bourgeron, "Disentangling the myriad genomics of complex disorders, specifically focusing on autism, epilepsy, and schizophrenia," *Cytogenet Genome Res*, vol. 135, pp. 228–240, Nov. 2011.

[45] F. Mitelman, B. Johansson, and F. Mertens, "The impact of translocations and gene fusions on cancer causation," *Nature Reviews Cancer*, vol. 7, pp. 233–245, Apr 2007.

[46] E. J. Hollox, L. W. Zuccherato, and S. Tucci, "Genome structural variation in human evolution," *Trends Genet*, vol. 38, pp. 45–58, July 2021.

[47] J. O. Korbel, A. E. Urban, J. P. Affourtit, B. Godwin, F. Grubert, J. F. Simons, P. M. Kim, D. Palejev, N. J. Carriero, L. Du, B. E. Taillon, Z. Chen, A. Tanzer, A. C. E. Saunders, J. Chi, F. Yang, N. P. Carter, M. E. Hurles, S. M. Weissman, T. T. Harkins, M. B. Gerstein, M. Egholm, and M. Snyder, "Paired-end mapping reveals extensive structural variation in the human genome," *Science*, vol. 318, pp. 420–426, Oct. 2007.

[48] J. A. Bailey, Z. Gu, R. A. Clark, K. Reinert, R. V. Samonte, S. Schwartz, M. D. Adams, E. W. Myers, P. W. Li, and E. E. Eichler, "Recent segmental duplications in the human genome," *Science*, vol. 297, pp. 1003–1007, Aug. 2002.

[49] H. Li and R. Durbin, "Fast and accurate short read alignment with Burrows-Wheeler transform," *Bioinformatics*, vol. 25, pp. 1754–1760, July 2009.

[50] F. J. Sedlazeck, P. Rescheneder, M. Smolka, H. Fang, M. Nattestad, A. von Haeseler, and M. C. Schatz, "Accurate detection of complex structural variations using single-molecule sequencing.," *Nature Methods*, vol. 15, pp. 461–468, June 2018.

[51] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin, and . G. P. D. P. Subgroup, "The sequence alignment/map format and SAMtools," *Bioinformatics*, vol. 25, pp. 2078–2079, Aug. 2009.

[52] A. R. Quinlan and I. M. Hall, "BEDTools: a flexible suite of utilities for comparing genomic features," *Bioinformatics*, vol. 26, pp. 841–842, Mar. 2010.

[53] C. Bartenhagen and M. Dugas, "RSVSim: an R/Bioconductor package for the simulation of structural variations," *Bioinformatics*, vol. 29, pp. 1679–1681, 04 2013.

[54] D. C. Jeffares, C. Jolly, M. Hoti, D. Speed, L. Shaw, C. Rallis, F. Balloux, C. Dessimoz, J. Bähler, and F. J. Sedlazeck, "Transient structural variations have strong effects on quantitative traits and reproductive isolation in fission yeast," *Nature Communications*, vol. 8, p. 14061, Jan. 2017.

[55] D. Heller and M. Vingron, "SVIM-asm: structural variant detection from haploid and diploid genome assemblies," *Bioinformatics*, vol. 36, pp. 5519–5521, 12 2020.

[56] M. Goel, H. Sun, W.-B. Jiao, and K. Schneeberger, "Syri: finding genomic rearrangements and local sequence differences from whole-genome assemblies," *Genome Biology*, vol. 20, p. 277, Dec 2019.

[57] S. F. Bunting and A. Nussenzweig, "End-joining, translocations and cancer," *Nat Rev Cancer*, vol. 13, pp. 443–454, June 2013.