HyINDEL - A Hybrid approach for Detection of Insertions and Deletions in Next Generation Sequencing Data

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science in Computational Natural Sciences by Research

by

Alok Thatikunta
201264197
alok.t@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
June 2024

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "**HyINDEL - A Hybrid approach for Detection of Insertions and Deletions in Next Generation Sequencing Data**" by **Alok Thatikunta**, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Prof. Nita Parekh

To my loving parents and sister

Acknowledgments

I would like to express my deepest gratitude to my supervisor Dr. Nita Parekh for her constant support and thorough guidance. She gave me the opportunity to pursue research and motivated me from time to time. The regular detailed discussions helped me in building concepts and improving the understanding of the subject. I would also like to thank her for encouraging and helping me to shape my interest and ideas.

I want to thank the Center for Computational Natural Sciences and Bioinformatics at the International Institute of Information Technology, Hyderabad for giving me support in various forms throughout my research. I would also like to thank the faculty of the centre for encouraging and helping me to enhance my knowledge in various fields.

I wish to thank Sree Harsha and Prashanthi for creating a healthy environment in the workspace and giving their honest inputs whenever needed. Special thanks to Anirudh and Aditya for being by my side all the time.

Lastly, I would like to take this opportunity to thank my parents for everything they have done for me. They made selfless sacrifices throughout their life so that I can get the best possible opportunity for growth. This work is a result of their blessings and good wishes.

Abstract

Insertion and deletion (INDELs) mutations, the most common type of structural variation in the human genome, has been implicated in numerous human traits and diseases including rare genetic disorders and cancer. Next generation sequencing (NGS) technologies have drastically reduced the cost of sequencing whole genomes, greatly contributing to the detection of structural variants. However, due to large variations in INDEL sizes and presence of low complexity and repeat regions, their detection remains a challenge. Here we present a hybrid approach, HyINDEL, for the detection of INDELs from paired-end NGS data which integrates clustering, split-mapping and assembly-based approaches. The method starts with identifying clusters of discordant and soft-clip reads which are validated by depth-of-coverage and alignment of soft-clip reads to identify candidate INDELs, while the assembly-based approach is used in identifying the insertion sequence. Performance of HyINDEL is evaluated on both simulated and real datasets and compared with the state-of-the-art tools. A significant improvement in recall and F-score metrics as well as in breakpoint support is observed on using soft-clip alignments. HyINDEL detects INDELs of all sizes (from small to large) and also identifies the insertion sequences. It is freely available at https://github.com/alok123t/HyINDEL.

CONTENTS

LIST OF FIGURES **2**

LIST OF TABLES **4**

CHAPTER 1 INTRODUCTION **5**

1.1. TYPES OF VARIATIONS 5

1.2. MECHANISMS FOR FORMATION OF STRUCTURAL VARIATIONS 7

1.3. EFFECTS OF STRUCTURAL VARIATIONS 8

1.4. APPROACHES FOR DETECTION 9

1.5. DNA SEQUENCING TECHNOLOGIES 10

1.6. ORGANIZATION OF THESIS 17

CHAPTER 2 MATERIALS AND METHODS **18**

2.1 DETECTION OF INDELS USING NGS 18

2.2 HYINDEL ALGORITHM 24

2.2.1 Preprocessing *24*

2.2.2 Deletions detection write your heading properly either it should be :detection of deletions", or "detecting deletions" *25*

2.2.3 Insertions detection *29*

2.2.4 Postprocessing *31*

2.3 IMPLEMENTATION OVERVIEW OF HYINDEL 32

CHAPTER 3 RESULTS AND DISCUSSION **36**

3.1. EXPERIMENTS ON SIMULATED DATA 36

3.1.1. Generation of simulated data *36*

3.1.2. Performance of HyINDEL *37*

3.1.3. Comparison of HyINDEL with other tools *40*

3.2. COMPARISON ON REAL DATA 45

3.2.1. Benchmarking Dataset *45*

3.2.2. Performance of HyINDEL *48*

3.2.3. Comparison of HyINDEL with other tools *50*

CHAPTER 4 CONCLUSION **52**

RELATED PUBLICATIONS **54**

BIBLIOGRAPHY **55**

LIST OF FIGURES

FIGURE 1.1: VARIOUS TYPE OF SIMPLE STRUCTURAL VARIATIONS, UNBALANCED SVS (DELETION, INSERTION, DUPLICATION) ARE SHOWN IN FIRST TWO ROWS, BALANCED SVS (INVERSION, TRANSLOCATION) ARE SHOWN IN THE THIRD ROW (REPRODUCED FROM [2]) 6

FIGURE 1.2: ILLUSTRATION OF SHOTGUN PROTOCOL FOR WHOLE GENOME SEQUENCING (REPRODUCED FROM [3]) 10

FIGURE 1.3: EXAMPLE OF MAXAM-GILBERT SEQUENCING REACTION (REPRODUCED FROM [5]) 12

FIGURE 1.4: EXAMPLE OF SANGER SEQUENCING REACTION (REPRODUCED FROM [6]) 13

FIGURE 2.1: (A) A DELETION EVENT SHOWN. PAIRED-END READS (1 AND 8) SHOWN IN BLACK, ARE THE CONCORDANT READS THAT CORRECTLY ALIGN TO THE REFERENCE GENOME, WHILE LARGE INSERT SIZE IS OBSERVED IN THE CASE OF DISCORDANT READS (MARKED IN RED). PARTIAL ENDS OF READS (SHOWN IN BLUE FOR READS 2 AND 7) DENOTE THE ALIGNED PART, WHILE THE UNALIGNED PARTS (SHOWN IN ORANGE) ARE THE SOFT-CLIPPED PART OF THE READS. THE READ SHOWED IN PURPLE IS A SPLIT READ (3 AND 6), WHEREIN BOTH THE PARTIAL ENDS OF THE READ ARE ALIGNED TO THE REFERENCE 20

FIGURE 2.2: EXAMPLE OF REGION FROM CHR2:92,325,000-92,326,000 (1000BP) WITH ABNORMALLY HIGH COVERAGE (16974) PRESENT IN TELOMERIC REGION OF NA12878 REAL SAMPLE WITH 52X MEDIAN COVERAGE (COVERAGE OF REGION IS HIGHLIGHTED IN RED OVAL) 25

FIGURE 2.3: LARGE DELETION IDENTIFIED USING HYINDEL ON NA12878, REAL DATASET FROM CHR1: 63,705,366-63,708,146 (2781 BP). RED LINES DENOTE DISCORDANT READS, SOFTCLIP PART OF READ (UNALIGNED BASES) ARE HIGHLIGHTED AT THE BREAKPOINTS 27

FIGURE 2.4: SMALL DELETION IDENTIFIED USING HYINDEL ON NA12878, REAL DATASET FROM CHR1: 74,755,170- 74,755,289 (120 BP). SOFTCLIP PART OF READ (UNALIGNED BASES) ARE HIGHLIGHTED AT THE BREAKPOINTS 28

FIGURE 2.5: SMALL INSERTION IDENTIFIED USING HYINDEL ON NA12878, REAL DATASET AT CHR1: 80,271,156 (65 BP). SOFTCLIP PART OF READ (UNALIGNED BASES) ARE HIGHLIGHTED AT THE BREAKPOINTS 30

FIGURE 2.6: SMALL INSERTION IDENTIFIED USING HYINDEL ON NA12878, REAL DATASET AT CHR2: 26937337 (2388 BP). SOFTCLIP PART OF READ (UNALIGNED BASES) ARE HIGHLIGHTED AT THE BREAKPOINTS, ONE END ANCHOR READS ARE HIGHLIGHTED IN RED 31

FIGURE 2.7: WORKFLOW OF HYINDEL FOR DETECTION OF INDELS (YELLOW: INPUT/OUTPUT FILES, PURPLE: METHOD, GREEN: TEMPORARY FILES, ORANGE: IN-MEMORY TEMPORARY FILES) 32

FIGURE 3.1: RECALL VALUES FOR HOMOZYGOUS AND HETEROZYGOUS DELETIONS FOR VARYING SEQUENCE COVERAGE ON SIMULATED DATA 38

FIGURE 3.2: RECALL VALUES FOR HOMOZYGOUS AND HETEROZYGOUS INSERTIONS FOR VARYING SEQUENCE COVERAGE ON SIMULATED DATA 39

FIGURE 3.3: ERROR IN INSERTION LENGTH FOR SIMULATED SAMPLE AT 30X USING OUR TOOL, HYINDEL. 40

FIGURE 3.4: COMPARISON OF F-SCORES FOR DELETIONS OF OUR TOOL HYINDEL WITH OTHER TOOLS ON SIMULATED DATA AT VARYING SEQUENCE COVERAGE 42
FIGURE 3.5: COMPARISON OF F-SCORES FOR INSERTIONS OF OUR TOOL HYINDEL WITH OTHER TOOLS ON SIMULATED DATA AT VARYING SEQUENCE COVERAGE 42
FIGURE 3.6: COMPARISON OF ERROR IN BREAKPOINT PREDICTIONS FOR DELETIONS USING OUR TOOL, HYINDEL WITH OTHER TOOLS ON SIMULATED DATA FOR 30X SEQUENCE COVERAGE 43
FIGURE 3.7: COMPARISON OF ERROR IN BREAKPOINT PREDICTIONS FOR INSERTIONS USING OUR TOOL, HYINDEL WITH OTHER TOOLS ON SIMULATED DATA FOR 30X SEQUENCE COVERAGE 43
FIGURE 3.8: COMPARISON OF NUMBER OF BREAKPOINT SUPPORTING (SOFTCLIP/SPLIT) READS USING OUR TOOL, HYINDEL WITH OTHER TOOLS FOR DELETIONS ON SIMULATED DATA FOR 30X SEQUENCE COVERAGE 44
FIGURE 3.9: SIZE DISTRIBUTION OF DELETIONS IDENTIFIED ON REAL DATA, NA12878 USING OUR TOOL, HYINDEL 49
FIGURE 3.10: SIZE DISTRIBUTION OF INSERTIONS IDENTIFIED ON REAL DATA, NA12878 USING OUR TOOL, HYINDEL 49

LIST OF TABLES

TABLE 1.1: COMPARISON OF SECOND-GENERATION SEQUENCING SYSTEMS (REPRODUCED FROM [7]) 15
TABLE 1.2: COMPARISON OF THREE GENERATIONS SEQUENCING SYSTEMS (REPRODUCED FROM [1]) 17
TABLE 2.1: TYPE OF DISCORDANT READS FOR EACH VARIANT TYPE 22
TABLE 2.2: DIFFERENT APPROACHES USED BY METHODS FOR DETECTION OF INDELS 24
TABLE 2.3: DESCRIPTION OF COMMAND-LINE INPUT ARGUMENTS FOR HYINDEL 33
TABLE 2.4: DEFAULT VALUES OF VARIOUS PARAMETERS IN EACH MODULE OF HYINDEL 34
TABLE 3.1: PRECISION, RECALL AND F-SCORE METRICS FOR PREDICTING DELETIONS USING HYINDEL ON SIMULATED DATA AT VARYING SEQUENCE COVERAGES 38
TABLE 3.2: PRECISION, RECALL AND F-SCORE METRICS FOR PREDICTING INSERTIONS USING HYINDEL, ON SIMULATED DATA AT VARYING SEQUENCE COVERAGE 39
TABLE 3.3: PRECISION, RECALL AND F-SCORE METRICS FOR PREDICTING DELETIONS USING HYINDEL WITH OTHER TOOLS ON SIMULATED DATA AT VARYING SEQUENCE COVERAG 41
TABLE 3.4: PRECISION, RECALL AND F-SCORE METRICS FOR PREDICTING INSERTIONS USING HYINDEL WITH OTHER TOOLS ON SIMULATED DATA AT VARYING SEQUENCE COVERAGE 41
TABLE 3.5: COMPARISON OF TIME AND MEMORY UTILIZATION FOR VARYING SEQUENCE COVERAGE 45
TABLE 3.6: PRECISION, RECALL AND F-SCORE METRICS FOR PREDICTING DELETIONS USING HYINDEL WITH OTHER TOOLS ON REAL DATA (NA12878) USING GIAB BENCHMARK (N=2676) 47
TABLE 3.7: PRECISION, RECALL AND F-SCORE METRICS FOR PREDICTING DELETIONS USING HYINDEL WITH OTHER TOOLS ON REAL DATA (NA12878) USING DGV BENCHMARK (N=973) 47
TABLE 3.8: PRECISION, RECALL AND F-SCORE METRICS FOR PREDICTING DELETIONS USING HYINDEL WITH OTHER TOOLS ON REAL DATA (NA12878) USING PACBIO-MERGED BENCHMARK (N=9241) 47
TABLE 3.9: NUMBER OF TRUE INSERTION PREDICTIONS USING HYINDEL WITH OTHER TOOLS ON REAL DATA (NA12878) USING DGV (N=108), GIAB (N=68) AND PACBIO (N=13669) BENCHMARKS 48
TABLE 3.10: NUMBER OF TRUE INSERTION PREDICTIONS USING HYINDEL WITH OTHER TOOLS ON REAL DATA (NA12878) USING DGV (N=108), GIAB (N=68) AND PACBIO (N=13669) BENCHMARKS 48

Chapter 1 Introduction

The differences in genome between individuals are known as genome variations. They can vary in terms of size, type and the position of the variation. These can be broadly classified into three categories based on the number of bases affected by the variation as, Single Nucleotide Variations, Insertions and Deletions (small indels), Structural variations (variations > 50bp). Genome variations are known to cause cancer and are linked to various neurological diseases. Therefore, it is important to understand and study the variations. The different types of variations and mechanisms for their formation are discussed in brief.

1.1. Types of variations

1) Single Nucleotide Variations

A point mutation that occurs at some location in the genome is called a single nucleotide variation (SNV). SNVs are the most frequently occurring type of genome variation. A human individual has approximately 3×10^6 SNVs. Variations can occur in protein coding or non-coding regions of the genome. A sequence of triplets of nucleotides are called codons, and each codon determines a single amino acid. A single SNV can change the amino acid, which can significantly alter the structure and function of protein. While non-coding SNVs can affect the expression of genes if they occur on functional sites. SNVs are different types and can have different impact

Example: SNVs in TP53, CTNNB1 genes have been shown to recurrently occur in hepatocellular carcinoma [1]. Small Indels

A small indels is defined as an insertion/deletion of a short segment of at most 50bp from the genome. Small indels are the second most frequently occurring type of genome variations. Indels occurring in protein-coding regions, can lead to frameshift variations. If the size of indel is a multiple of 3, it will cause an insertion/deletion of few codons, which may or may not affect the property of the gene. In the case of when size of indels is not a multiple of 3, a frameshift variation occurs leading to change/destruction of the whole protein. While indels in non-coding regions can affect the gene expression if they are present in functional sites.

Example: Deletion in intron 2 of BIM gene has been associated with resistance to tyrosine kinase inhibitors in CML patients.

2) Structural Variations

Large scale variations in the genome affecting more than 50bp are known as structural variations (SVs). Based on the type of SV, they are classified as simple and complex SVs. Simple SVs include insertion, deletion, reversal, duplication, transposition and translocation. These can be grouped as balanced and unbalanced variations. Balanced variations involve reversal, transposition and translocation. Unbalanced variations change the number of copies of DNA segments and include insertion, deletion and duplication. The different types of SVs are shown in Figure 1.1. Complex SVs involve more than one SV. Each human individual is expected to have tens of thousands of SVs. The number of SVs is far lower than the number of SNVs and small indels. But, the number of bases in the genome affected by SV are more. SVs have been linked to various diseases and phenotypic variations. The different mechanisms which lead to formation of SVs are described in the next section.

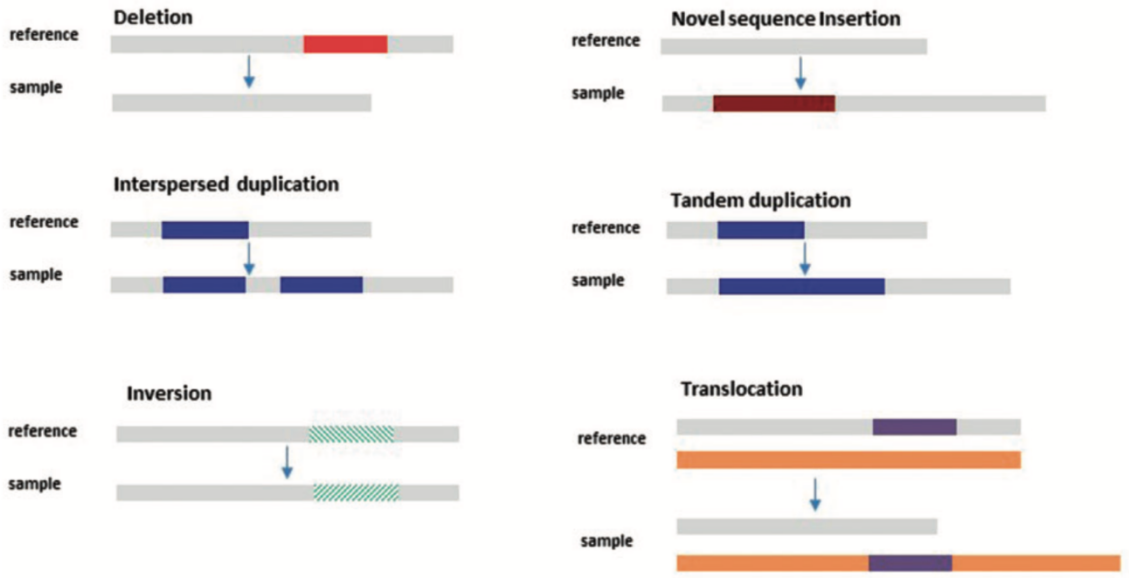

Figure 1.1: Various type of structural variations, unbalanced SVs (deletion, insertion, duplication) are shown in first two rows, balanced SVs (inversion, translocation) are shown in the third row (reproduced from [2])

1.2. Mechanisms for formation of Structural Variations

Variations which are observed to occur frequently across individuals are known as recurrent variations. SVs mainly occur due to inaccurate repair and errors during replication of DNA. The mechanisms are described below. Non allelic homologous recombination (NAHR)

DNA damaging occurs when exposed to smoking, UV radiation, chemotherapy and also due to various environmental factors. The double stranded DNA is broken into two pieces, creating double strand breaks (DSBs). Homologous recombination is one of the major repairing pathway. In this process, an allelic homologous chromosome is used as a template to rejoin the DSBs. During the repair mechanisms the DSBs may be misaligned to other homologous regions, which leads to the formation of an SV. NAHR occurs in the majority of segmental duplications, repeat elements like SINEs, LINEs, LTRs. NAHR can generate translocation, deletion, duplication and inversions. Many SVs formed by NAHR are known to be recurrent.

1) Non homologous end joining (NHEJ)

Double strand breaks can also be repaired by non-homologous end joining and microhomology-mediated end joining (MMEJ) mechanisms. These mechanisms do not require a homologous region as template for rejoining the DSBs. First, the overhanging ends of the DSBs are joined. This joining process is guided by short homologous sequences known as microhomology. Next, the mismatched nucleotides are removed/modified and the gaps are filled by synthesis. After the ligation step, SV are formed. This mechanism is known to create non-recurrent SVs.

2) Replication based

SVs are also generated due to errors during replication process of DNA. During replication, the double stranded DNA is first separated and a replication fork is formed. The lagging strand is synthesized in a direction opposite to that of the growth direction. During the synthesis process, two types of mistakes can happen, described below.

Polymerase slippage: The template forms a secondary structure and the synthesis of the lagging strand might skip the DNA segment. This process results in a deletion.

Template switching: The lagging strand may disengage from the template and switch to another template in a nearby replication fork.

SVs formed due to errors in replication are known to be non-recurrent.

3) Mobile element insertion

Transposable elements (Alu, L1 sequences) change their positions in the genome leading to mobile element insertions. This movement is mediated by retrotransposons, DNA transposons and retroviruses.

4) Chromothripsis

The genome is shattered at multiple breakpoints, followed by an error prone DNA repair by NHEJ. This phenomenon is known as Chromothripsis. This results in complex genome rearrangements. It is observed in bone and liver cancers, and other type of cancers.

The effects observed due to the presence of SVs are described in the next section.

1.3. Effects of Structural variations

SVs have been linked to various diseases and also drug resistance. The phenotypic changes due to SVs are described below.

1) Loss/gain of gene

Unbalanced SVs can lead to gain or loss of genes. This can also affect the dosage sensitive genes.

Example: Variation in number of copies of 1.4 Mb in 17p12 leads to change in dosage of gene PMP22. A tandem duplication of this region, increases the copies of this gene from 2 to 3, causing Charcot-Marie-Tooth disease (CMT1A). While, a deletion reduces the gene copies from 2 to 1, causing hereditary neuropathy with liability to pressure palsies (HNPP).

2) Loss/gain of part of gene

Unbalanced SVs can duplicate or delete only a part of gene. This can lead to gain or loss of a functional part of gene or change expression levels.

Example: Deletion of intron 2 of BIM is known to be associated with TKI resistance treatment in CML patients.

3) Fusion of genes

Balanced SVs can link two genes together, causing fusion genes.

Example: Translocation of chr9, chr22 causes BCR-ABL1 gene fusion leading to chronic myelogenous leukemias (CMLs). Integration of foreign DNA

Foreign DNA can also be integrated in the genomes by SVs.

Example: Hepatitis B virus (HBV) integration into genome is observed in liver cancer.

Next, we discuss various approaches for detection of SVs.

1.4. Approaches for detection

Traditional approaches like qPCR, FISH, comparative genome hybridization (CGH) can be used to verify the presence of a single SV. Development of Array-CGH technology has led to the detection of unbalanced SVs. Array-CGH is based on the analysis of intensity ratios of hybridization of two differentially dyed DNAs against the same target oligonucleotides. Array-CGH methods are limited to detection of only unbalanced SVs and also cannot give the absolute copy number of the detected variants. Further, due to the resolution of array-CGH, small variants couldn't be detected.

The above-mentioned approaches are limited in detection of SVs. With a rapid increase in throughput, a comprehensive detection of SVs can be achieved using second-generation sequencing systems. Initial methods involved short single-end reads for detection of SVs. Technological advancements have led to the development of paired-end reads, which can be more reliably aligned to the reference genome. This has led to the detection of variants of all types across the whole genome. The basic steps involved in second-generation sequencing methods are briefly summarized below.

Second-generation sequencing methods involve a wet-lab phase and dry-lab phase. In the wet-lab phase, shotgun sequencing is done to obtain a set of paired-end reads covering the whole genome. The protocol for whole-genome sequencing involves three steps. First step involves randomly dividing the genome into smaller DNA fragments, known as sonication. In the second step, DNA fragments of a fixed size, called the insert size are selected. Single-end or paired-end reads are sequenced from the DNA fragments in the third step. In single-end sequencing, reads are sequenced from one end of each DNA fragment. The read is obtained from the 5′ end of the forward template of the DNA fragment. In case of paired-end sequencing, reads are sequenced from both ends of the DNA fragment. 5′ reads are obtained from both forward and reverse template of the DNA fragment in inward orientation. This process is illustrated in Figure 1.2.

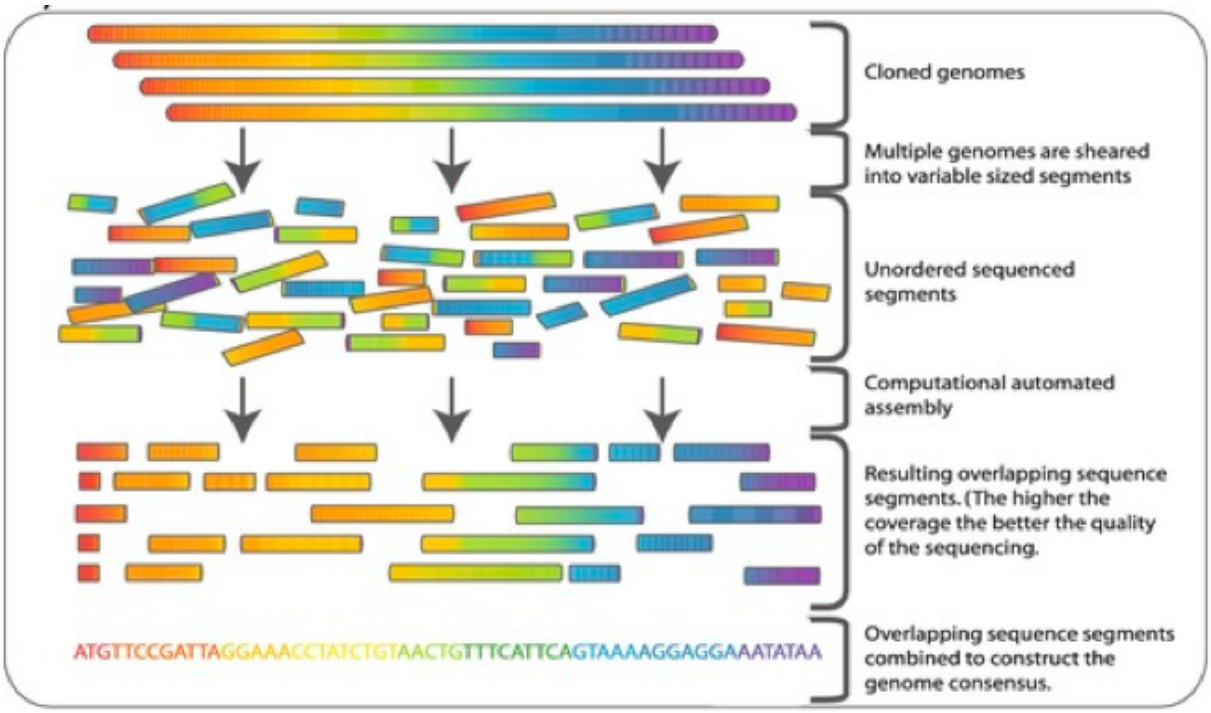

Figure 1.2: Illustration of shotgun protocol for whole genome sequencing (reproduced from [3])

Using second-generation sequencing methods, we can detect both unbalanced and balanced SVs. In this thesis, we focus on the detection of Insertions and Deletions, which are the most commons type of SVs in the human genome. They have been implicated in numerous human traits and diseases including rare genetic disorders and cancer. However, due to large variations in size and presence of low complexity and repeat regions, their detection remains a challenge. We further discuss more about the detection of variations and challenges involved in Chapter-2.

Next, we briefly discuss about DNA and the various methods for sequencing DNA.

1.5. DNA Sequencing technologies

Deoxyribonucleic acid (DNA) is a hereditary material present in all living organisms carrying genetic code essential for life. It was first isolated by Friedrich Miescher in 1869. The double helical molecular structure of DNA was identified by Francis Crick and James Watson in 1953. In 1977, Sanger had sequenced the first complete DNA sequence of a viral genome.

Each DNA strand is composed of monomer units called nucleotides. Each nucleotide is composed of a nitrogen containing base, a sugar molecule and a phosphate group. There are four different nucleotides namely Adenine (A), Guanine (G), Cytosine (C) and Thymine (T).

The nucleotides in a DNA strand are linked by covalent bonds between the sugar of one nucleotide and phosphate of the next nucleotide. The other strand of DNA is antiparallel to the first strand. The two strands of DNA are linked by hydrogen bonds and two nucleotides from opposing strands obey the Watson-Crick rule of base pairing. The nucleotide A from one strand is paired with T from the other strand, while G is paired with C.

Cell division is the process in which a single cell is divided into two daughter cells. In this process the double stranded DNA is separated into single strands and the DNA polymerase enzyme uses the single strand as a template to replicate into two identical double helixes. Due to this duplication process, all the cells within an individual have the same genome. Errors in copying, lead to variations in few cells which can lead to diseases. Genome variations among individuals lead to different phenotype attributes and diseases. For this it is important to determine the order of nucleotides in the DNA sequence. The various methods for sequencing of DNA is described next.

First generation sequencing

Based on distances from a radioactive label to positions occupied by each base along a DNA molecule, two methods were developed.

a) Chain termination procedure by Sanger and Coulson
b) Chemical cleavage procedure by Maxam and Gilbert

The general steps involved in first generation sequencing methods are mentioned below.

1) Amplification of DNA template

In this step, the DNA template is amplified to produce multiple copies of the input DNA. First, the DNA template is inserted into a plasmid vector. Then the plasmid vector is inserted into host cells. These host cells are cloned to result in multiple copies of the original DNA template.

2) Generating all possible prefixes of DNA template

Two different approaches were used for generating prefixes.

a) Maxam-Gilbert method

Radioactive labelling at 5′ end of DNA fragment was performed by a kinase reaction using gamma ^{32}P. Next, DNA strand is cleaved at specific positions using chemical reactions. Purines (A, G) are de-purinated using dimethyl sulphate, while pyrimidines (C, T) are hydrolysed using hydrazine. Chemical treatments cleave G, A+G, C, C+T. A+G

refers that both A and G can be cleaved by the same reaction. Four reactions are performed separately, corresponding to above cleave patterns. This results in differently sized DNA strands with radioactive labels and mixture separated by prefixes ending with the above cleavage patterns in different test tubes. An example is illustrating the different cleavage patterns is shown in Figure 1.3.

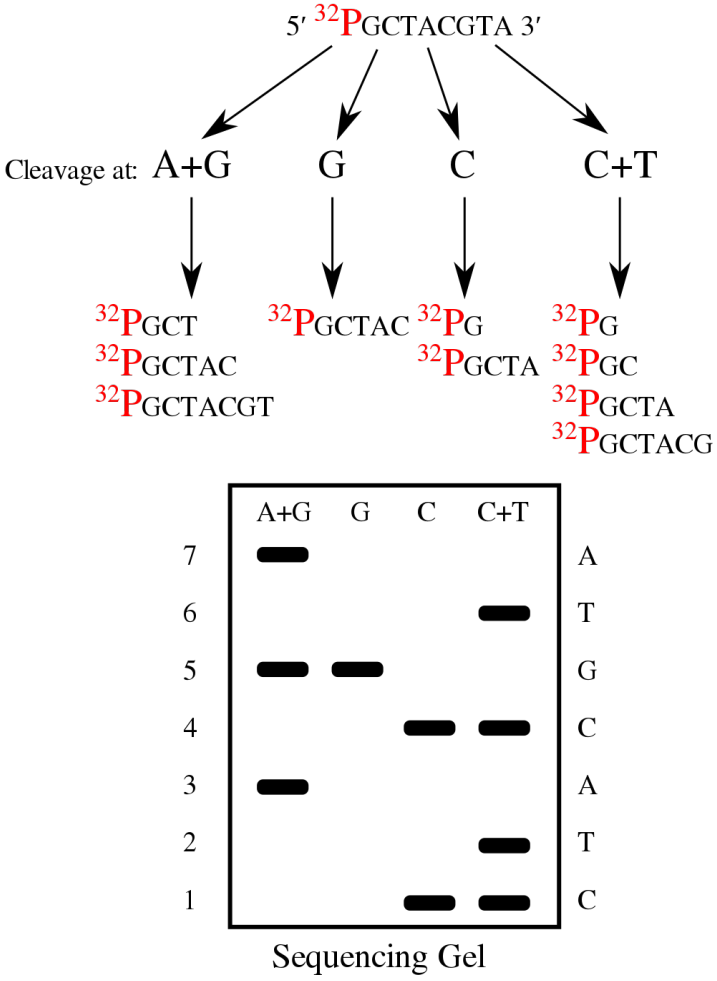

Figure 1.3: Example of Maxam-Gilbert sequencing reaction (reproduced from [5])

b) Sanger method

A synthetic oligonucleotide is annealed, which acts as a binding site for primer and to provide initiation for DNA synthesis. DNA polymerase synthesis is performed in presence of dNTP, ddNTPs (here, N refers to any of A, C, T, G). Four reaction vials, each containing all dNTPs, DNA polymerases, and each reaction vial containing one type of ddNTP are made. DNA synthesis occurs in each vial resulting in a set of single stranded DNA

molecules of different lengths with prefixes ending in N in vial corresponding to ddNTP. An example illustrating Sanger sequencing is shown in Figure 1.4.

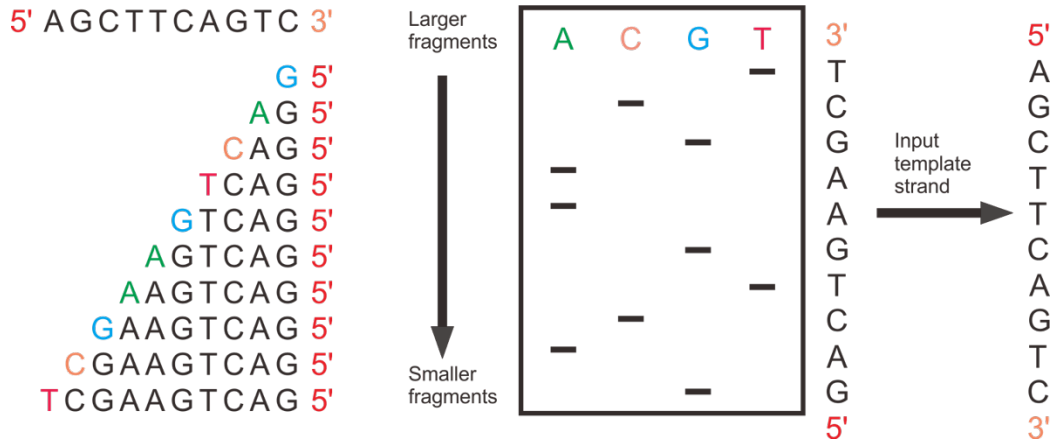

Figure 1.4: Example of Sanger sequencing reaction (reproduced from [6])

3) Separating by Electrophoresis

The mixture of DNA fragments of varying length is separated using gel electrophoresis. An electric field is applied to the DNA mixture. DNA being negatively charged moves to the positive pole with shorter fragments moving faster due to friction. Gel electrophoresis separates the DNA mixture into DNA fragments by size with single base resolution.

4) Readout with Fluorescent tags

The gels are out on an X-ray film, resulting in a ladder image. The DNA fragment has a fluorescent tag attached to the terminal ddNTP. Based on the light emitted from different bands the DNA sequence can be read.

Due to the usage of toxic chemicals in Maxam-Gilbert technique, the Sanger sequencing approach was more preferred. Using, Sanger sequencing reads of length about 800bp could be sequenced. One of the major drawbacks of first-generation sequencing methods was the cost involved and the time taken to sequence.

Second generation sequencing

Key changes from Sanger sequencing has led to the development of massively parallel DNA sequencing. Few of them are mentioned below.

a) In vitro amplification

b) Multiplexing

The general steps involved in second-generation sequencing are template preparation and base calling. They are explained in brief below.

1) Template preparation

In this step, for the given set of DNA fragments, DNA templates are generated by ligating adaptor sequences to the ends of each fragment. These templates are amplified using PCR. This can be done in two different techniques, summarized below.

a) Emulsion PCR: Each DNA template is amplified using a bead. The surface of bead contains a primer corresponding to one adaptor. Then the DNA template hybridizes with one primer.
b) Bridge PCR: Two types of primers corresponding to different adaptors are coated on a flat surface. Each DNA template is hybridized to one primer. One end of each bridge is tethered to the surface and the amplification is repeated in cycles.

2) Base calling

In this step, the DNA sequences are read from the amplified templates. There are different approaches for sequencing, sequencing by synthesis and sequencing by ligase. These sequencing methods are explained in brief along with a technology which is based on it. Various metrics are compared for each technology in Table 1.1.

a) Polymerase mediated using reversible terminator nucleotides

First, the primer is hybridized on the adaptor of template. Using DNA polymerase, a reversible terminator nucleotide is incorporated to the template. Using imaging, the signal corresponding to the dye of reversible terminator nucleotide can be scanned. Next, the termination is reversed by cleaving the dye-nucleotide and the above steps are repeated.

Illumina: Using Bridge PCR, the DNA templates are amplified. Four color cyclic reversible termination is used for sequencing in parallel. One of the drawbacks is that the accuracy of sequence decreases with increase in number of nucleotides added.

b) Polymerase mediated using unmodified nucleotides

During the incorporation of dNTP into a growing DNA strand, a pyrophosphate and a positively charged hydrogen ion are released. By detecting the change in concentration of pyrophosphate, hydrogen ion, the template DNA is sequenced. This method is also known as pyrosequencing.

Example technology Roche 454: Emulsion PCR is used for amplification of templates. Each DNA template along with one bead is loaded into a well. In each iteration, a different dNTP flows across the well. Polymerase is extended by one base if the dNTP is complementary to the template and a pyrophosphate is released. Using enzymes, this pyrophosphate is converted into visual light and a CDC camera detects the light signal from the well. The intensity of light is recorded as a flowgram, which is interpreted to get the DNA sequence. A drawback of this approach is that, when long homopolymers present in template, a higher rate of errors in indels is observed.

c) Ligase mediated

Polymerase mediated methods extend the template base by base using polymerase. In contrast ligase mediated method used probes to identify the bases on the template.

SOLiD: The template sequences are amplified using emulsion PCR. Next, these templates are placed on a plate and the bases of each templated are checked using probes. For a template, in each iteration SOLiD probes two adjacent bases, resulting in a two-base color encoding. The DNA sequence is decoded from the color encoding. As each base in the template is covered by two probes, the error rate in detection of single nucleotide variations is lower.

Table 1.1: Comparison of second-generation sequencing systems (reproduced from [7])

Sequencer	**Roche 454 GS FLX**	**Illumina HiSeq 2000**	**SOLiD v4**
Sequencing mechanism	Pyrosequencing	Sequencing by synthesis	Ligase mediated, two-base coding
Read length	700bp	101bp (paired-end)	50 + 50bp
Accuracy	99.9%	98%	99.94%
Number of reads	1 M	3 G	1200-1400 M
Time/run	24 hours	3-10 days	14 days
Advantage	Read length, fast	High throughput	Accuracy
Disadvantage	Error rate for long homopolymers, cost	Short read assembly	Short read assembly
Cost/million bases	$10	$0.07	$0.13

Third generation sequencing

First and second-generation sequencing methods involve template amplification, due to which copying errors and sequence biases arise. Third generation sequencing methods do not involve PCR, which reduces the preparation time. The signal for reading DNA sequence is captured in real time. The DNA sequences from third generation methods have longer read lengths as compared to first and second-generation sequencing methods. Two different approaches are described below.

1) PacBio SMRT sequencing

It is based on optically observing the polymerase mediated synthesis in real time. It utilizes a zero-mode waveguide, a nanophotonic structure consisting of a circular hole. Each of the DNA bases is attached to a different fluorescent dye. Inside the ZMW, a single active DNA polymerase with a single molecule of single stranded DNA template is immobilized. Light illuminated in this structure is monitored. When a nucleotide is incorporated by DNA polymerase, the fluorescent tag is cleaved off, which then emits signals for a sufficient time to be detected. Using this technology reads of length > 10kb are obtained. Further, sequence methylation status is also detected by this approach. A drawback of this approach is the error rate in sequencing is high (about 10%) and the errors in sequencing are randomly distributed across the read.

2) Nanopore sequencing

It is based on observing the pattern in the flow of ions, when a single stranded DNA molecule passes through a narrow channel. A pore of size in nanoscale in a thin membrane is called a nanopore. When a constant electric field is applied, an electric current can be observed in the system. A positive charge draws the DNA strand across the two chambers flowing through the nanopore. The DNA sequence is decoded by detecting the difference in electrical conductivity. One of the major advantages of this approach is the portability of the sequencing device, due to the use of electrical signals as compared to optic signals. A drawback of this approach is the high error rate in sequencing.

The cost of sequencing per genome has drastically reduced from first to second to third generation sequencing techniques. Further, the lengths of reads sequenced has drastically increased from first and second generation as compared to third generation sequencing. The three generations of sequencing methods are summarized in Table 1.2.

Table 1.2: Comparison of three generations sequencing systems (reproduced from [1])

	First generation	**Second generation**	**Third generation**
Amplification	In-vivo cloning and amplification	In-vitro PCR	Single molecule
Sequencing	Electrophoresis	Cyclic array sequencing	Real-time monitoring of PCR, Nanopore
Starting material	More	Less (<1 μg)	Very less
Cost	Expensive	Cheap	Very cheap
Time	Very slow	Fast	Very fast
Read length	About 800bp	Short	Very long
Accuracy	< 1% error	< 1% error	High error rate

1.6. Organization of thesis

In this thesis, we propose a pipeline for detection of Insertions and Deletions (HyINDEL) in second-generation sequencing data. In Chapter-2, Section-2.2 we describe the proposed approach based on clustering and assembly of reads. Chapter-3 describes the methods involved in construction of data sample, details about benchmarks used and metrics used for comparison. Our method is first evaluated on simulated data and compared to state-of-the-art tools in Chapter-3, Section-3.1. Next, we have compared the performance of our method on real data NA12878 sample using various benchmarks in Chapter-3, Section-3.2.

Chapter 2 Materials and Methods

Overview

In this chapter, we first describe the classification of reads observed in the vicinity of INDELs, which help us identify candidate sites for variant detection. Next, we discuss the various approaches used for detection of INDELs. Finally, we present the algorithm developed by us, HyINDEL for detection of INDELs involving clustering of reads at a locus and identification of variants of different sizes in Section-2.2. And finally summarize the implementation details of HyINDEL in Section-2.3Chapter 0.

2.1 Detection of INDELs using NGS

A typical pipeline for detection of INDELs in NGS data first involves alignment of reads to the reference. Next, we estimate insert size and coverage parameters, which help us in the classification of reads. Each of these steps are briefly explained below. Based on the distribution of read signatures observed, we identify candidate sites for INDELs which are later used for variant detection. These steps are discussed in detail below.

Read alignment

The first step of INDEL detection is the alignment of reads to the reference genome. Read alignment algorithms can be classified based on their method into two types as hash table indexing and burrow-wheeler transform (BWT) based methods [8]. Hash table-based algorithms use seed and extend strategy. In the seed phase a small subset of possible locations for the alignment to the reference are identified by detecting common *k-mers* between the read and the reference using hash tables. In the extend phase, the exact location of the read alignment is identified using dynamic programming algorithm. Example of hash table methods are Novoalign [9]. BWT based methods align the entire read to the reference. Burrow wheeler transform is a reversible transformation of a string into runs of similar characters, which can be easily compressed. Using FM-index, one can efficiently identify the alignment locations. BWT based methods have lower memory requirements as compared to hash table methods. Example of BWT methods are Bowtie [10], BWA [11].

Insert size estimation

During sequencing of Illumina paired-end reads, the size selection step involves extracting DNA fragments of a certain fixed length, Insert size. Insert size is defined as the outer distance between the two reads of a pair. It can be computationally calculated using insert size information from the alignment file. Reads of a pair that do not cross any breakpoint of a variation are expected to align to the same chromosome with inward orientation and insert size within a range ($span_{min}$, $span_{max}$).

$$(span_{min}, span_{max}) = (I_{median} - k\delta, I_{median} + k\delta)$$

$$\delta = \sqrt{\sum_{i=1}^{n} \frac{(I_i - I_{median})^2}{n}}$$

Here, I_{median} and δ are the median and standard deviation of insert size respectively.

The values of I_{median} and δ are calculated using the Picard [12] tool's CollectInsertSizeMetric module. The input to Picard is an alignment file and the output is a text file containing various insert size metrics.

Classification of reads in vicinity of INDELs

Reads from regions without any variation are expected to align completely to the reference, except for reads from low complexity regions. While reads inside and around the variation region align differently based on the type of variation. First, we define types of read signatures based on which we can associate them to variants. Next, we describe the distribution of reads in deletion and insertion locus.

Concordant read: A paired-end read that does not cross the INDEL breakpoint, for which both reads of the pair aligns to the same chromosome of the reference, with the same inward orientation and insert size within the range ($span_{min}$, $span_{max}$).

Discordant read: Anomalous paired-end reads which cross one of the INDEL breakpoints, do not align concordantly. In case of deletions, the insert size > $span_{max}$, while in case of insertions, the insert size < $span_{min}$. In both cases of insertions and deletions the reads of the pair align to the same chromosome with the same inward orientation.

Split read: Reads spanning the INDEL breakpoint, align partly at the 5′ breakpoint and other part aligns at the 3′ breakpoint. Due to the split nature of the alignment, these reads are known as split reads.

Softclip read: Reads spanning the INDEL breakpoint, for which only one part of the read is aligned, and the other part is unaligned and represented as soft-clip. These partially aligned reads are known as softclip reads.

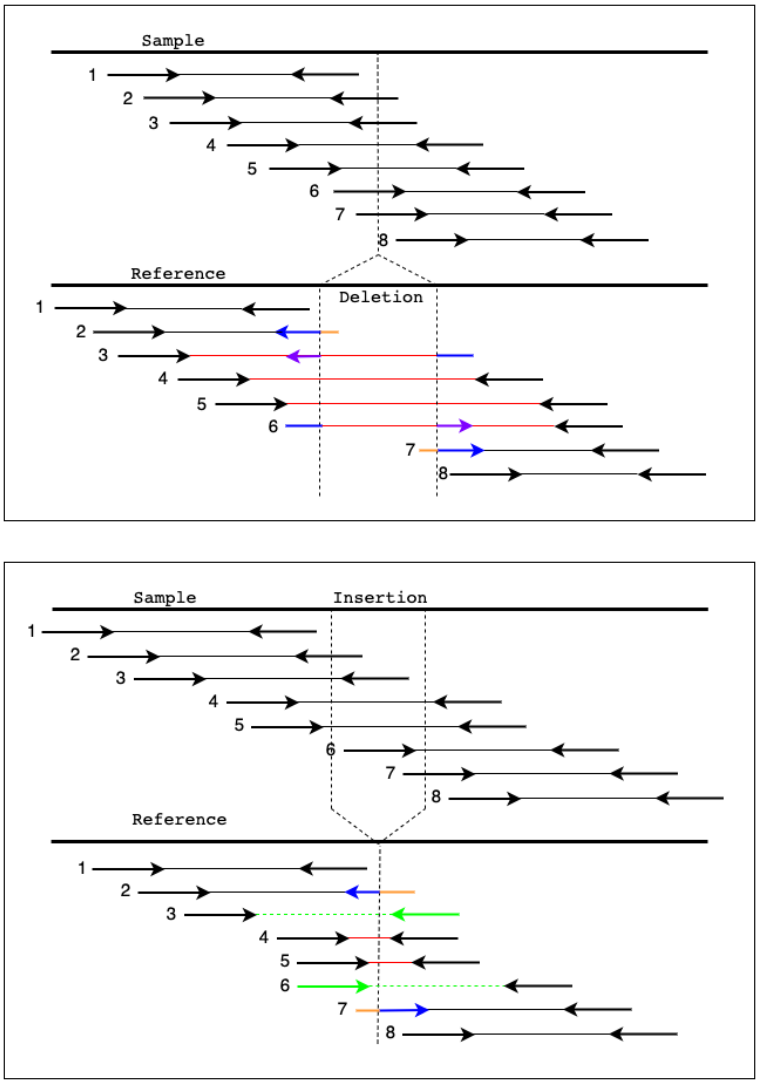

Figure 2.1: (a) A deletion event shown. Paired-end reads (1 and 8) shown in black, are the concordant reads that correctly align to the reference genome, while large insert size is observed in the case of discordant reads (marked in red). Partial ends of reads (shown in blue for reads 2 and 7) denote the aligned part, while the unaligned parts (shown in orange) are the soft-clipped part of the reads. The read showed in purple is a split read (3 and 6), wherein both the partial ends of the read are aligned to the reference

(b) An insertion event shown. Here, black paired-end reads (1 and 8) are the concordant reads that correctly align to the reference genome, while small insert size is observed in the case of discordant reads (shown in red for reads 4 and 5). Partially aligned reads (reads 2 and 7 shown in blue) have their soft-clipped part (shown in orange) unaligned. The reads 3 and 6 marked in green denote the unaligned reads of one-end anchored (OEA) reads

Distribution of Reads at a Deletion locus

A contiguous sequence of bases absent in the sample with respect to the reference is known as a deletion. It is represented by two breakpoints on the reference, referred to as 5′ and 3′ breakpoints, the sequence between which is missing in the sample (Figure 2.1(a)). It may be noted that paired-end reads in the vicinity of a deletion event can be categorized into different types based on the orientation and insert size on mapping to the reference genome. A paired-end read that does not cross the INDEL breakpoint, aligns to the same chromosome of the reference with the same inward orientation and insert size within the range ($span_{min}$, $span_{max}$) where $span_{min} = I_{median} - k\delta$ and $span_{max} = I_{median} + k\delta$, I_{median} and δ are median and standard deviation of the insert size, (default k=3). These are called *concordant* reads (reads 1 and 8 in Figure 2.1(a)). Anomalous paired-end reads are those that cross one of the deletion breakpoints. In this case two reads of a pair do not align concordantly (i.e., insert size > $span_{max}$) and are called *discordant* reads (reads 3, 4, 5 and 6 in Figure 2.1(a)). When a part of the read aligns at 5′ breakpoint and the other part aligns at 3′ breakpoint of the deletion (or vice-versa), because of the split nature of alignment these are called *split* reads (reads 3 and 6 in Figure 2.1(a)). If only one part of the read aligns to the reference while the other part is unmapped, then such reads are called *soft-clip* reads. Reads marked with soft-clip at 3′ end of the alignment (e.g., 70M30S in CIGAR string) provide information of the 5′ breakpoint of the deletion (read 2 in Figure 2.1(a)), while those marked with a soft-clip at 5′ end of the alignment (e.g., 20S80M) provide information about the 3′ breakpoint (read 7 in Figure 2.1(a)). Thus, in the vicinity of a deletion event, cluster of discordant, soft-clipped and split reads are observed which are helpful in accurately detecting the breakpoints of the deletion region.

Distribution of Reads at an Insertion locus

A continuous sequence of bases present in the sample but missing in the reference as shown in Figure 2.1(b) corresponds to an insertion event. It is represented by a single breakpoint on the reference. As in the case of deletion, paired-end reads in the vicinity of an insertion event can be categorized into different types based on orientation and insert size when mapped to reference genome. Paired-end reads that do not cross the insertion breakpoint, aligns to the same chromosome of the reference with the same inward orientation and insert size are called *concordant* reads (reads 1 and 8 in Figure 2.1(b)). Paired-end reads with a short insert size (< $span_{min}$) but with same inward orientation are called *discordant* reads (reads 4 and 5 in Figure 2.1(b)). When one of the read of a pair spans the insertion breakpoint (reads 2 and 7 in

Figure 2.1(b)), 5′ (3′) of the read is partially aligned, called *soft-clip* reads. These help in precisely detecting the breakpoint and the prefix (suffix) of the 'insertion' sequence. If the insertion sequence is larger than the read length, typically only one read of the pair is mapped to the reference genome (reads 3 and 6 in Figure 2.1(b)) and are called one-end anchored (OEA) reads. For identifying insertions larger than the insert size of the library, in addition to soft-clip and one-end anchor reads, orphan reads (i.e., none of the ends of a paired-end read map reference genome) are also considered (not shown in the figure). The proposed approach uses information from all these different types of reads to detect the location of insertion breakpoint and construct the insertion sequence.

Approaches for detection of INDELs

Based on the different types of read signal at an INDEL event, various tools have developed approaches for their identification. Earlier methods were based on discordant read signal as the sequencing read lengths were shorter (36-72bp). A drawback of using discordant signal is we cannot accurately detect the precise breakpoints. As the lengths of reads have increased (100-250bp), read alignment has become more sensitive. This has led to the development of methods based on split-read/softclip alignments, which can be used to accurately predict INDEL events. Further, INDEL events are in different sizes, making them even harder to detect. For example, in large deletions, one can expect discordant reads to be more prevalent, than in case of small deletions. Often methods incorporate more than one signal for better prediction of all size ranges. The different approaches used are described below.

Clustering approach: In the vicinity of a variant region we observe a group of reads with improper orientation or insert size (discordant reads), which can be used as a signal to identify the candidate variant. This approach involves two steps, classification of a paired-end read into a discordant type. And, second clustering discordant paired-end reads of the same type in a region. The type of discordant reads observed for different type of variants is summarized in Table 2.1.

Table 2.1**:** Type of discordant reads for each variant type

Type	Insert size	Orientation
Deletion	$> span_{max}$	+/-
Insertion	$< span_{min}$	+/-
Inversion	$> span_{max}$	+/+ or -/-
Tandem Duplication	Size of tandem copy	-/+

Inter-chromosomal translocation	N/A (Different chromosomes)	N/A

Clustering approach is dependent on the classification of paired-end reads into discordant reads. It cannot be used for detection of small variants, as the paired-end reads in the vicinity of the small indels can be classified as concordant reads. Further, the usage of discordant reads is limited in only giving the approximate breakpoints of a variant.

Split-mapping approach: Anomalous paired-end reads with split-reads are observed when reads span the breakpoints of a variant region. Two scenarios are observed in this case. In the first case, a part of the read aligns at the 5′ breakpoint, while the other part of the read aligns at the 3′ breakpoint. Split-reads are a direct evidence of variants as the read partially spans both the breakpoints as expected. While in the second case, alignment of the split read on the reference genome is soft-clipped, i.e., only one part of the read is aligned and the unaligned portion is called soft-clip. Softclip occurs either due to sequencing errors, errors in reference genome or due to the read spanning a variant region. At the ends of a variant a region has both start and end, we observe a group of reads with softclip alignments. Reads at the 5′ (3′) breakpoint have an upstream (downstream) softclip in case of INDELs. A group of localized softclip reads of the same type can be grouped together as a single cluster. Since, the softclip region of the read is unaligned, during clustering we further have to realign the reads to verify if they are from the same region in the sample. Using the position of softclip, the breakpoints of a variant can be accurately detected.

Assembly approach: Large and complex variants cannot be detected using short paired-end reads, as the read cannot completely span the variant region. The reads in the vicinity of the variant can be assembled to generate contigs, which are longer than the reads. The larger contigs can then be mapped back to the reference for the validation of a variant. This approach involves identification of partially aligned, one-end aligned reads in the vicinity of a variant and assembling them to build larger contigs. Typically, de-novo assemblers are used for the construction of contigs. This method is used to detect novel sequence insertions in the sample, which are not present in the reference. Assembly approaches typically align reads or shorter k-mers to find overlapping pairs in order to construct a larger contig. This makes the approach computationally more intensive compared to other approaches.

Hybrid approach: A single signal cannot comprehensively detect all types and sizes of SVs, many methods use a combination of above signals for accurate detection. Few approaches combine the information from multiple SV callers, while others use a combination of SV signals for more reliable predictions. A drawback of integrating multiple SV callers for detection would be the repeated processing of the same input file, as it involves usage of lot of computing resources. Hence, SV callers that combine multiple signals are more popular.

In Table 2.2, we have summarized the various approaches used by tools we have used in our comparison for detection of INDELs.

Table 2.2**:** Different approaches used by methods for detection of INDELs

Tool	Input	Variants detected	Method	Reference
Lumpy	BAM, filtered BAM files	SVs	Discordant, Split-read	[13]
TIDDIT	BAM	SVs	Discordant, Split-read, Depth-of-coverage	[14]
SoftSV	BAM	SVs	Discordant, Softclip	[15]
Pamir	BAM, reference	NSI	Split-read, Assembly	[16]
Popins	BAM, reference	NSI	Split-read, Assembly	[17]

2.2 HyINDEL Algorithm

In this section, we discuss the proposed approach HyINDEL for the detection of INDELs.. And involves pre-processing, clustering of soft-clip and discordant reads, identification of INDELs and post-processing.

2.2.1 Preprocessing

Eukaryotic genomes are rich in repeat sequences and low-complexity regions, especially the centromeres and telomeres. Reads originating from these regions ambiguously align to multiple regions in the reference genome. Consequently, a number of discordant and soft-clip reads with abnormally high depth of coverage are observed in these regions. To avoid predicting spurious variants in these regions, read depth profile is constructed for the sample genome in non-overlapping bins of size 1000bp. The median coverage (c_{median}) of the sample is computed and bins having read depth ($> 3 \times c_{median}$) are discarded to filter low complexity regions. This step is performed using Mosdepth [18] in our pipeline. Also, alignments with a low mapping quality (< 20) are not considered for variant detection by parsing the BAM file.

An example of region from NA12878 real dataset with abnormally high read depth is shown in Figure 2.2. The region spans from chr2:92,325,000-92,326,000 and is present in telomeric region of the chromosome.

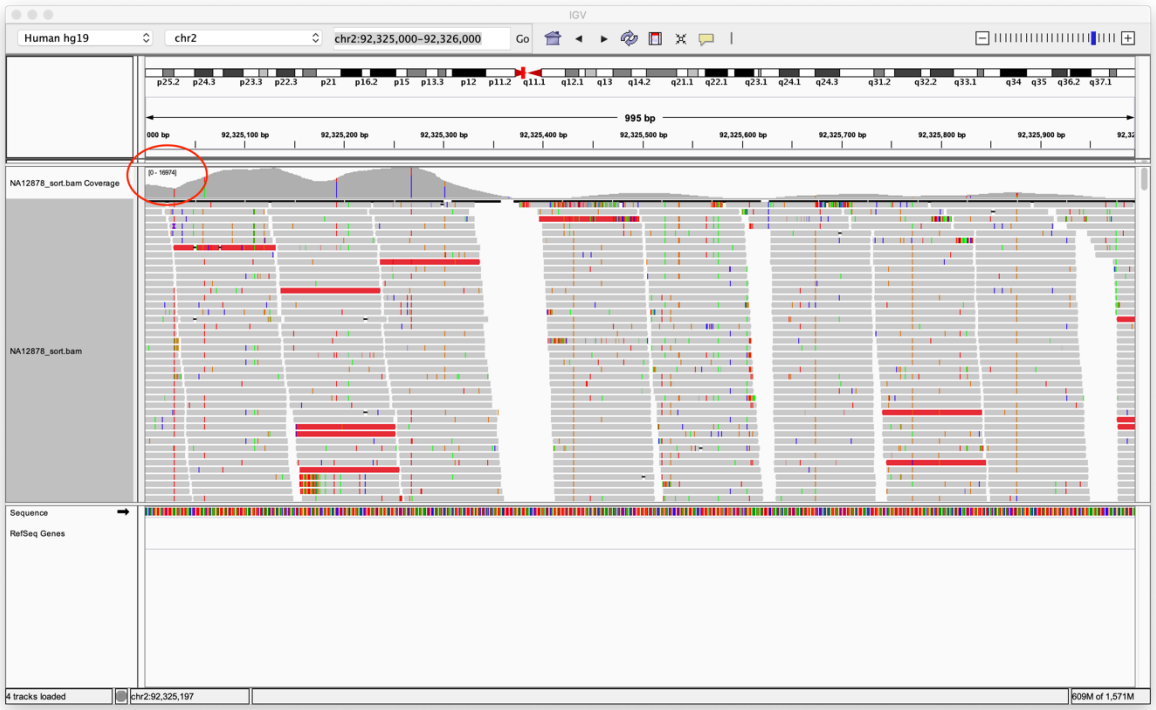

Figure 2.2: Example of region from chr2:92,325,000-92,326,000 (1000bp) with abnormally high coverage (16974) present in telomeric region of NA12878 real sample with 52x median coverage (Coverage of region is highlighted in red oval)

2.2.2 Detection of deletions

Clustering Discordant reads

As seen in Figure 2.1(a), the discordant reads in this case exhibit larger insert size, but same inward orientation of the reads. Let (f_{st}^{i}, f_{en}^{i}), (r_{st}^{i}, r_{en}^{i}) denote the start and end of the alignment of forward and reverse reads of the i^{th} discordant paired-end read on the reference. Two paired-end discordant reads are merged into a single cluster if the reciprocal overlap of the region between f_{st} and r_{en} is ≥ 0.65 (i.e., if f_{st}^{4} to r_{en}^{3} region for the reads 4 and 5 in Figure 2.1(a) is ≥ 0.65, then the two reads are merged). Other paired-end reads satisfying this criterion with any of the reads of this cluster are merged into this cluster. This results in clusters of discordant paired-end reads, which indicate approximate location of the breakpoints of probable deletion events. We expect the 5′ breakpoint of the deletion to lie within the interval

(f_{en}^{i}, f_{en}^{i} + $span_{max}$) and the 3′ breakpoint within (r_{st}^{j} - $span_{max}$, r_{st}^{j}), where *i* and *j* correspond to the reads with the farthest 5′ and 3′ coordinates in the cluster, respectively.

Clustering Soft-clip reads

Two soft-clipped reads support the 5′ (3′) breakpoint of the same deletion event if the start (end) of soft-clipped part are within 5bp and exhibit high sequence similarity over the overlap region (≥ 90%), and belong to the same cluster . The sequence similarity is assessed by carrying out semi-global alignment using the scoring scheme (1, -1, -1) for match, mismatch and gap penalty respectively. Soft-clip cluster containing reads with soft-clip at the 5′ (3′) end is known as downstream (upstream) soft-clip cluster respectively. The soft-clip clusters thus constructed are used in the detection of the breakpoint. A split-read that has an overlap of ≥ 90% with any read of the 5′ (3′) softclip cluster, is merged with the respective soft-clip cluster.

By analyzing the discordant clusters, approximate location of a deletion region is obtained. On identifying the upstream and downstream soft-clip clusters that overlap with the discordant clusters, the precise locations of the breakpoints can be obtained. Any discordant/soft-clip cluster having less than $c_{median}/10$ reads are discarded, where c_{median} is the median depth-of-coverage of the sample. Below we briefly discuss our approach in the detection of small (50, 500) and large deletions (> 500).

Detection of Large Deletions

Each discordant cluster provides approximate location of the 5′ and 3′ breakpoints of a deletion event within the interval (f_{en}^{i}, f_{en}^{i} + $span_{max}$) and (r_{st}^{j} - $span_{max}$, r_{st}^{j}), where *i* and *j* correspond to the reads with the farthest 5′ and 3′ coordinates in the cluster, respectively. If a soft-clip cluster has its coordinates lying within the interval (f_{en}^{i}, f_{en}^{i} + $span_{max}$), it defines the 5′ breakpoint, and the soft-clip cluster with coordinates lying within the interval (r_{st}^{j} - $span_{max}$, r_{st}^{j}) defines the 3′ breakpoint region. These two soft-clip clusters define the same deletion event if the alignment of reads from the two clusters exhibit high sequence similarity (as discussed below). If no soft-clip cluster pair is found to map the discordant cluster, the candidate deletion is reported as *imprecise* deletion event.

An example of a large deletion detected using HyINDEL in real dataset is shown in Figure 2.3.

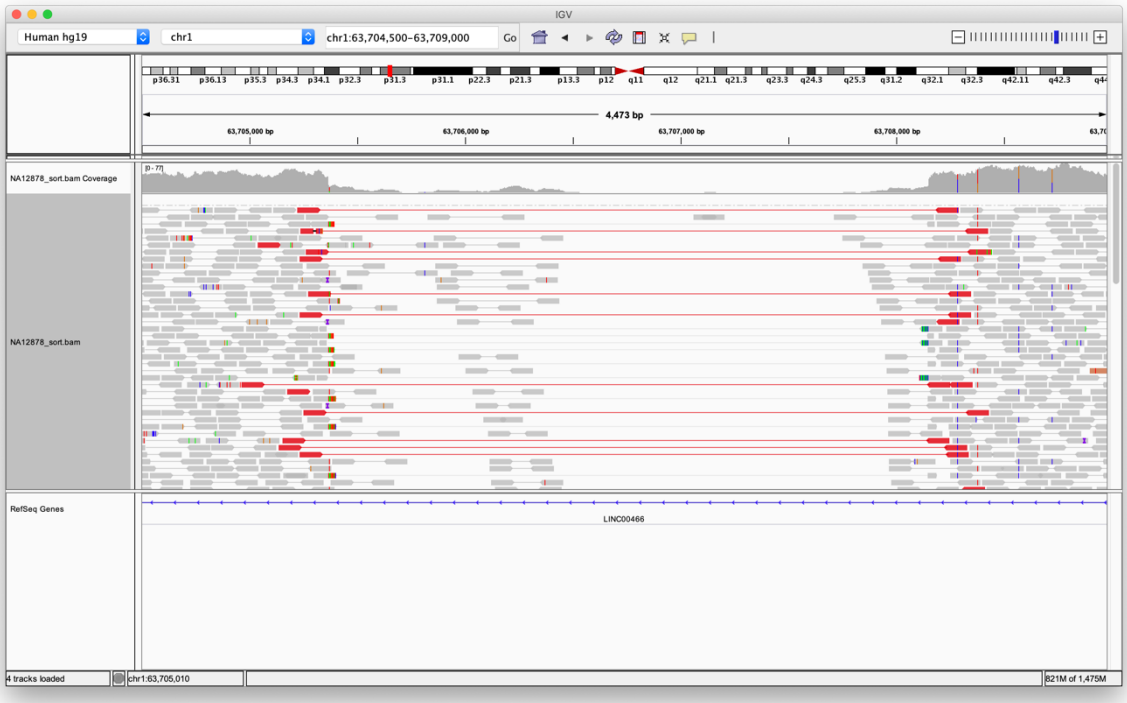

Figure 2.3: Large deletion identified using HyINDEL on NA12878, real dataset from chr1: 63,705,366-63,708,146 (2781 bp). Red lines denote discordant reads, softclip part of read (unaligned bases) are highlighted at the breakpoints

Detection of Small Deletions

Small deletions ($< L_{small} = k\delta$, $k = 3$ and δ corresponds to the standard deviation of the insert size) are missed by the above approach as in this case all the paired-end reads around the breakpoint get classified as concordant reads. Thus, only soft-clip clusters are available for the detection of small deletions. In this case for each upstream soft-clip cluster, a downstream soft-clip cluster within a distance of L_{small} and exhibiting high sequence similarity between inter-cluster reads is identified, indicating they represent the same deletion event (as discussed below). An example of a small deletion detected using HyINDEL in real dataset is shown in Figure 2.4.

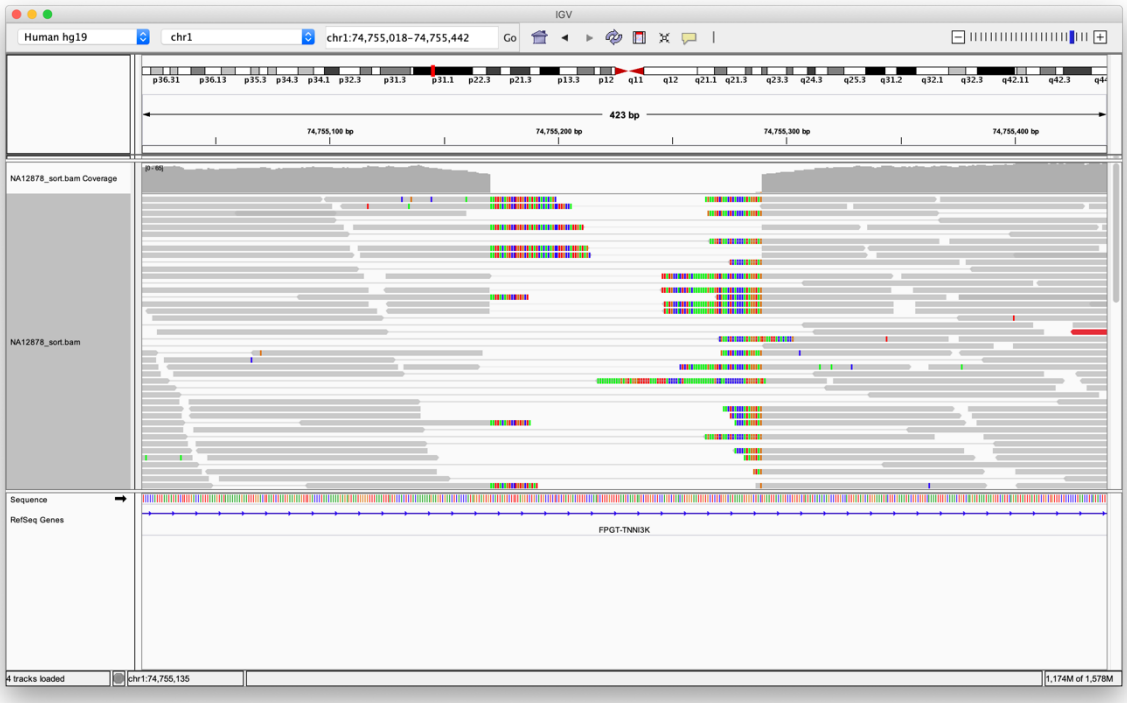

Figure 2.4: Small deletion identified using HyINDEL on NA12878, real dataset from chr1: 74,755,170- 74,755,289 (120 bp). Softclip part of read (unaligned bases) are highlighted at the breakpoints.

Breakpoint identification by alignment of soft-clipped reads

Each deletion event is represented by an upstream and a downstream soft-clip cluster as mentioned above. For correctly paired upstream and downstream soft-clip clusters (i.e., defining the same deletion event), unaligned regions of soft-clip read from the upstream cluster would be similar to the mapped regions of soft-clip reads of the downstream cluster and vice-versa, as shown in Figure 2.1(a). Hence, a semi-global alignment of the reads from the upstream soft-clip cluster and the corresponding downstream cluster is performed. The two clusters are considered to be associated to the same deletion event if the average inter-alignment score is $\geq \frac{1}{2} \times r_{len}$, where r_{len} denotes the read length. Three soft-clip reads (based on the size of the soft-clipped regions) from each cluster are considered for the alignment since they are expected to have the highest overlapping regions. The above alignment step is skipped if the two clusters contain 3 or more split-reads with their 5′ and 3′ ends mapping respectively to the 5′ and 3′ soft-clip clusters. The deletion breakpoints are identified as the median start (end) location of the soft-clip position of the upstream (downstream) cluster respectively.

2.2.3 Insertions detection

As is clear from Figure 2.1(b), each insertion event is characterized by a cluster of soft-clip reads upstream and downstream of the breakpoint (e.g., reads 2 and 7). For correctly paired upstream and downstream soft-clip clusters that define the same insertion event, we expect the start of soft-clip position from upstream cluster to be the same as the end of soft-clip position from the downstream cluster and represent the site for candidate insertion breakpoint. For each candidate insertion site, OEA reads within $\pm$ $span_{max}$ of the breakpoint are extracted. To identify the insertion sequence, *de novo* assembly of partially aligned soft-clip reads, unaligned ends of OEAs and orphan reads is performed using Minia [19], a short-read assembler based on de-Bruijn graph approach. This results in a set of contigs (in fasta format) corresponding to each insertion event. The contig is expected to span the entire insertion sequence and also partially contain region adjacent to the insertion breakpoint. When this contig is aligned to the reference, we expect a split alignment from which we can identify the insertion breakpoint and sequence as described below. Alignment to the reference is done using Minimap2 [20], a sequence alignment program for aligning long reads or assemblies to a reference genome.

Identification of insertion breakpoint and sequence

The assembled contig comprises of a prefix, insertion sequence and suffix, with both prefix and suffix ends mapping to the reference. Small insertions are directly represented in the CIGAR string of the alignment file with an insertion (e.g., 40M120I50M). The CIGAR string is parsed to obtain the position of insertion with respect to the alignment position of the read. Based on the position and length of insertion from the CIGAR string, the insertion sequence is reported.

In case of large insertions, a split-alignment of the contig is observed. In one of the split alignment we have the prefix of contig aligned to the reference, and the remaining part of contig marked with a soft-clip. While in the other alignment, we have the suffix of the contig aligned to the reference or vice-versa. This is possible as the assembled contig is expected to contain the insertion sequence and is larger than the short reads, which only partially span the insertion. The position of softclip is reported as the insertion position. The softclip sequence excluding the other split-aligned portion of contig is reported as the insertion sequence. In case the split-read is not available, the insertion event is reported as *imprecise* insertion event with a partial insertion sequence. This happens when we are unable to completely assemble the insertion

sequence into a single contig. An example of a small insertion detected using HyINDEL in real dataset is shown in Figure 2.5, and a large insertion detected is shown in Figure 2.6.

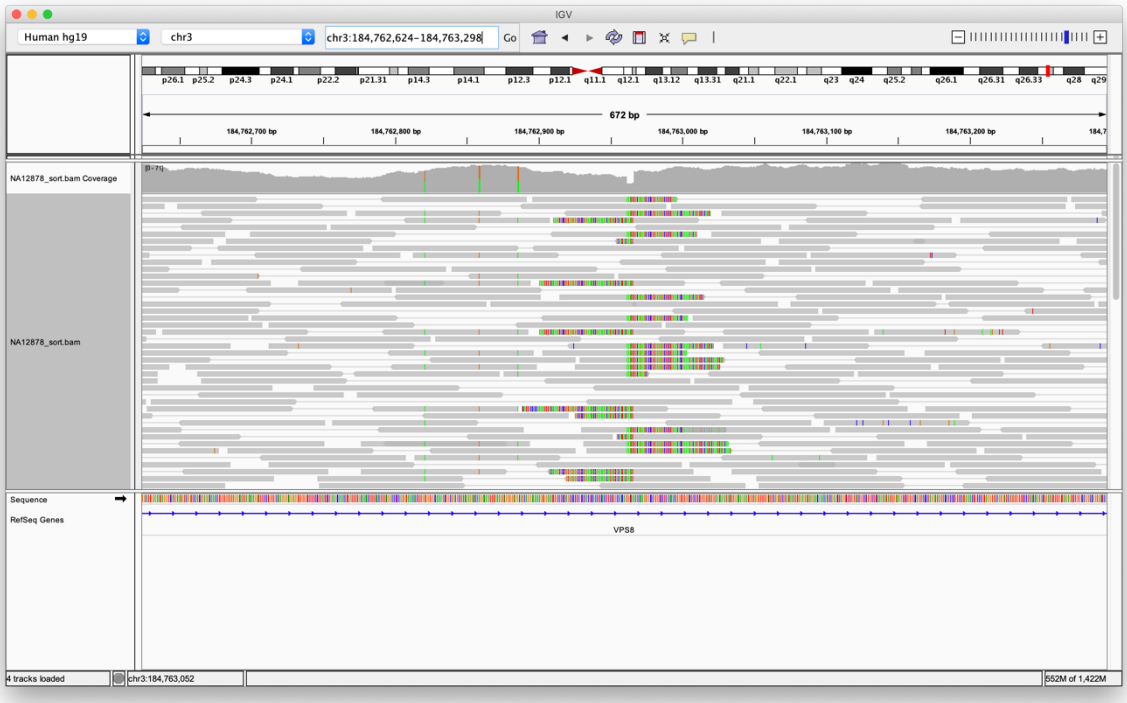

Figure 2.5: Small insertion identified using HyINDEL on NA12878, real dataset at chr1: 80,271,156 (65 bp). Softclip part of read (unaligned bases) are highlighted at the breakpoints

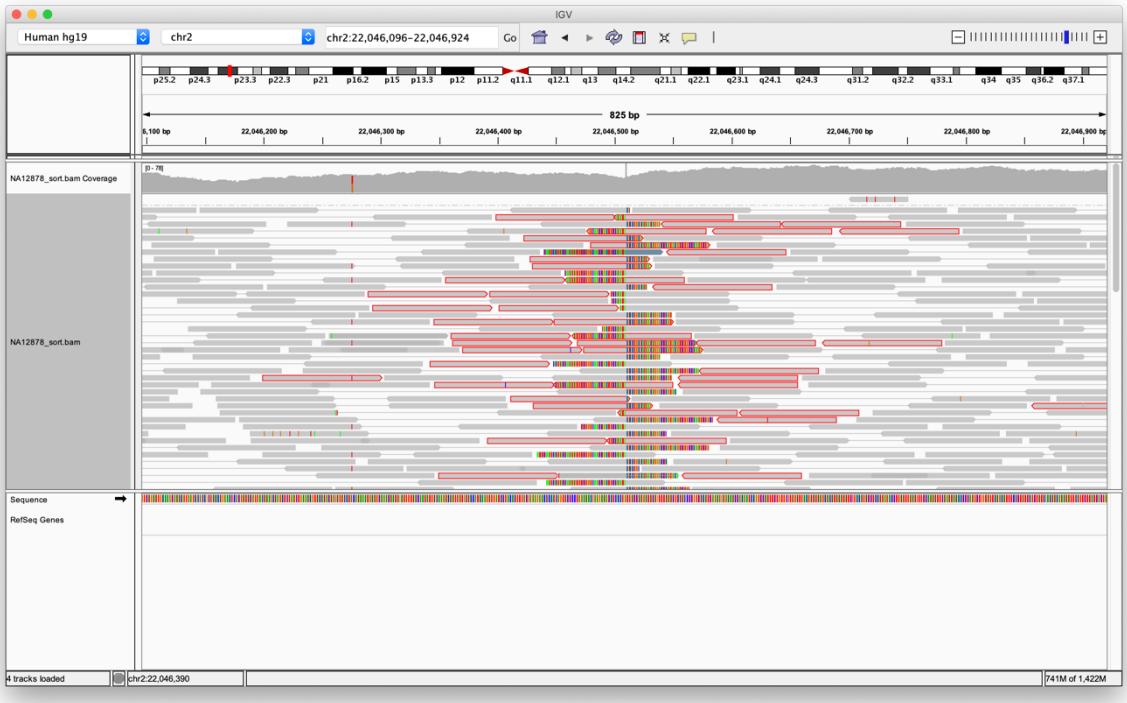

Figure 2.6: Small insertion identified using HyINDEL on NA12878, real dataset at chr2: 26937337 (2388 bp). Softclip part of read (unaligned bases) are highlighted at the breakpoints, one end anchor reads are highlighted in red

2.2.4 Postprocessing

INDEL events that have low support are filtered. For large deletions identified using both discordant and soft-clip signal, minimum support of reads required is > threshold = $c_{median}/3$, where c_{median} defines the median depth-of-coverage of the sample. While for imprecise and small deletions identified using only soft-clip reads, a threshold of $c_{median}/6$ is used. In the case of homozygous deletions, we do not expect any reads present in the deletion region, while in case of heterozygous deletions, approximately half of the reads are expected to be present. Hence, we compute the ratio of sequence coverage of the candidate deletion region to that of its upstream and downstream flanking regions of size 1000bp each. For a candidate deletion, if the ratio $cov_{event}/cov_{flank} < 0.2$ (for both flanks), it is referred to as a homozygous deletion, and if it lies in the interval $0.2 \leq cov_{event}/cov_{flank} \geq 0.9$, it is reported as a heterozygous deletion, remaining events are classified as complex variants. The variants predicted are reported in VCF output format (v4.2).

2.3 Implementation overview of HyINDEL

HyINDEL is an open source method for detection of insertions and deletions from whole genome next generation sequencing data. It is designed for handling input alignment files generated from short read Illumina sequencing platforms. The input to the tool is a coordinate sorted alignment file (.bam) along with its index (.bam.bai). An output file with name "output.vcf" is generated in the output directory specified as a command line argument. HyINDEL has 3 modules namely (i) Pre-processing, (ii) Variant calling and (iii) Post-processing. HyINDEL is freely available with open source MIT license at https://github.com/alok123t/HyINDEL. The workflow is shown in Figure 2.7.

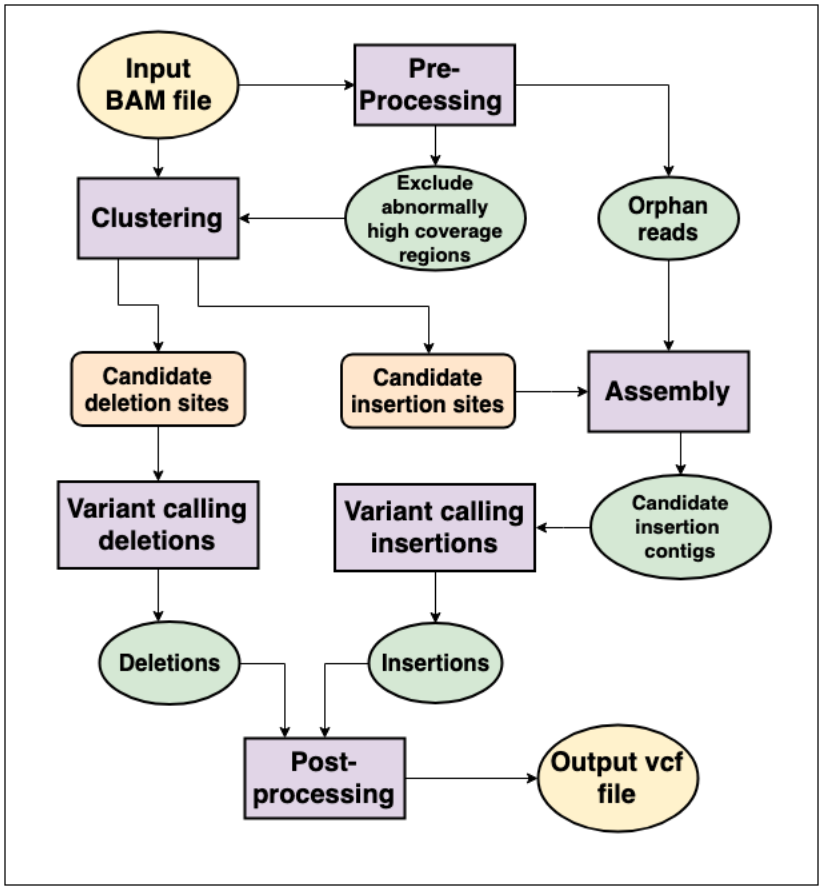

Figure 2.7: Workflow of HyINDEL for detection of INDELs (yellow: input/output files, purple: method, green: temporary files, orange: in-memory temporary files)

Implementation details

HyINDEL is a hybrid approach combining clustering, split-mapping and assembly approaches for detection of INDELs. The approach and design are based on SoftSV. HyINDEL is implemented in combination of C++ and bash. CMake, a popular tool for building and packing software is used for easier installation/setup by the end user. The input alignment files are quite often very large and in compressed format, one of the challenges is to use an external tool for reading. For this we have used Bamtools [21], an open source C++ api and toolkit designed for easier parsing of BAM data. Since, the files are very large, we have developed our method to process the input file in chunks of data. For this, we have used Transwarp [22], an open source library in C++ for task concurrency. For easier arguments parsing and proper error handling, we have used Args [23], another open source library in C++. The required version of Bamtools (v2.5.1), Transwarp and Args are present as external modules in the HyINDEL package and the setup instructions are part of the HyINDEL package. In summary, the setup of HyINDEL also handles the installation of dependencies.

Apart from the above tools, HyINDEL also requires 3 major tools for variant detection. Mosdepth is used for faster calculation of depth of coverage across the input alignment file for identifying regions with high coverage to be excluded from analysis. Minia [19] is used for *de novo* assembly for construction of contigs used in the detection of insertions. Samtools [24] is used in the post-processing step. The end-user needs to ensure that the executables to Mosdepth, Minia and Samtools are present on the UNIX $PATH variable, so they can be used by HyINDEL. The detailed steps for installation are provided in the readme file of HyINDEL package. HyINDEL is implemented to be used as a command-line utility program. The various arguments are described in Table 2.3. The default values of various parameters are summarized in Table 2.4.

Table 2.3**:** Description of command-line input arguments for HyINDEL

Options short	Options long	Description	Attributes	Mandatory
-i PATH	--inp=PATH	Path to input (input.bam) file (index input.bam.bai file also present in same directory)	Absolute path	Yes
-o PATH	--out=PATH	Path to output folder (creates folder if it doesn't exist)	Absolute path	Yes

-s VAL	--insSz=VAL	Median Insert size of library	Integer	Yes
-d VAL	--stdDev=VAL	Standard deviation of insert size	Integer	Yes
-l VAL	--readLen=VAL	Read length	Integer	Yes
-c VAL	--cov=VAL	Sequencing coverage	Integer	Yes
-t VAL	--threads=VAL	Threads	Integer	No

Table 2.4**:** Default values of various parameters in each module of HyINDEL

Step	**Description**	**Default Value**
Preprocessing	Window size considered for excluding regions	1000
	Window size considered for splitting input region, during parallel processing	10,000,000
	Overlap size between adjacent windows, during parallel processing	20,000
	Multiplier for excluding high coverage regions (k*median coverage)	3
	Threads used for Mosdepth	4
Variant calling	Minimum mapping quality of alignments	20
	Match, mismatch, gap penalty for semi-global alignment	(1, -1, -1)
	Minimum alignment threshold (k*readlength)	0.9
	Minimum support for discordant or softclip cluster	Median coverage/10
	Reciprocal overlap for two discordant reads to be in same cluster	0.65
	Maximum distance, reciprocal overlap for merging two deletion events	5, 0.95
	Maximum distance between paired-clusters for candidate insertion	10
	Minimum length of softclip used during alignment of contigs and softclip reads	20
Assembly	k-mer size used for Minia	31
Post-processing	Flank size used for coverage calculation	1000
	Minimum mapping quality of alignments in flanking regions	20

Parallelization

A whole genome dataset from Illumina platform at 50x sequencing coverage would result in about 100GB of raw reads. The compressed alignment file (BAM) generated using BWA for the input reads, results in an alignment file of approximately 100GB. Processing large files consumes lot of computational resources. Therefore, it is important to analyze the steps which require immense resources and optimizing them. Modern servers with multiple cores are often used for running large computationally intensive jobs.

The human genome is about 3 billion bases long and is made up of 23 chromosomes. One approach for faster processing, is to divide the input file by chromosome and process each chromosome using a different thread. A drawback of this approach is that, as the lengths of chromosomes are different, resulting in a sub-optimal resource utilization. Instead, we divide each chromosome into regions of 10mb with adjacent regions having an overlap of 20kb. These parameters can be changed by the user. This results in 382 regions (human assembly, GRCh37). Each region can now be processed for variant detection independently on a different thread and the output be merged finally.

Chapter 3 Results and Discussion

In this chapter, we first show the efficacy of our method on simulated data. Briefly we discuss about generation of diploid sample for simulated data containing small and large insertions and deletions. And, we compare the performance of our method HyINDEL at varying sequence coverage and for homozygous, heterozygous variants on the simulated dataset. For comparison on real data, we have used the publicly available high coverage NA12878 sample obtained from Illumina Platinum genomes [25]. Using this real dataset, we evaluate the performance of our algorithm HyINDEL on 3 benchmarks. Our INDEL predictions on both simulated and real data are also compared with state-of-the-art tools, namely, Lumpy [13], TIDDIT [14], SoftSV [15] for deletions, and Pamir [16] and Popins [17] for insertions. Results obtained are discussed in terms of metrics such as F-score, breakpoint accuracy and are compared with 5 state of the art tools for INDEL detection.

3.1. Experiments on Simulated data

Humans are diploid organisms i.e., they contain two sets of 23 chromosomes with each set obtained from their parents. At a variant locus, if both the alleles are the same then it is known as a homozygous variation while if it is present in only one allele then it is called as a heterozygous variation. Detection of heterozygous variations is difficult as compared to homozygous variants due to the presence of a smaller number of supporting reads coming only from one allele. To incorporate this behavior in our simulated sample, we generate a human diploid sample with homozygous and heterozygous variants.

3.1.1. Generation of simulated data

To evaluate the performance of HyINDEL on simulated data, a diploid sample is constructed by inserting variants into the human genome (assembly GRCh37). The location of insertions and deletions are identified randomly using SVSim [26] and respectively saved in two lists. Each list consists of 750 entries with 375 each from the two size ranges: small (50, 500) and large (500, 10000). First haploid sample is constructed by inserting 375 insertions and 375 deletions in the reference genome. The second haploid sample is constructed by inserting all the 750 insertions and 750 deletions in the reference genome. The two haploid samples are then merged to obtain a diploid sample containing homozygous (375 insertions, 375 deletions) and heterozygous (375 insertions, 375 deletions) variants. Paired-end reads of

length 100bp are generated using ART simulator [27] to simulate reads from Illumina HiSeq 2500. Mean and standard deviation of the insert size for the paired-end reads is set to 350bp and 20bp respectively. Three diploid samples corresponding to sequencing coverage 10×, 20× and 30× are generated. The reads are aligned using BWA-MEM [11] to the human genome (assembly GRCh37). The resulting BAM files are sorted by coordinate and then indexed using Samtools [24].

3.1.2. Performance of HyINDEL

The performance of our method HyINDEL is evaluated using the following metrics.

Precision (Positive predictive value, PPV): Ratio of true positive indel events to the total events detected.

$$\text{Precision} = \frac{\text{True positives events predicted}}{\text{Total events predicted}}$$

Recall (True positive rate, TPR): Ratio of true positive events to the total events present.

$$\text{Recall} = \frac{\text{True positives events predicted}}{\text{Total events present}}$$

F-score: Harmonic mean of precision and recall.

$$\text{F-score} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$$

A deletion prediction is considered to be a true positive if the reciprocal overlap of the predicted and true deletion is at least 50%. An insertion prediction is considered a true positive if the distance between the predicted and actual insertion site is within 10bp. Performance of HyINDEL is evaluated on two parameters: sequencing coverage and accuracy of length of insertions predicted.

Effect of sequence coverage

INDELs are classified based on their length into two size ranges: small (50bp-500bp) and large (>500bp). Homozygous INDELS are easier to detect as compared to heterozygous due to higher number of reads supporting the homozygous event. Also, detection of large deletions is easier as compared to that of small deletions, mainly due to the presence of both discordant and softclip signals. Precision and recall metrics are compared for small and large

deletions in Table 3.1 at varying sequence coverages. Further, recall values for homozygous and heterozygous deletions are also shown in Figure 3.1.

Recall and F-score values are higher for deletions compared to insertions and improve with increase in sequencing coverage. As expected, recall values are higher for detection of homozygous compared to heterozygous INDEL events. A high precision value of ~1 observed for all cases indicate the reliability of our predictions. The majority of indel events missed are due to fewer reads supporting the variant even at 30× coverage.

Table 3.1**:** Precision, Recall and F-score metrics for predicting deletions using HyINDEL on simulated data at varying sequence coverages

HyINDEL Deletions	**Homozygous recall**		**Heterozygous recall**		**Overall Precision**	**Overall Recall**	**Overall F-score**
	Small	**Large**	**Small**	**Large**			
10x	96.25	98.40	90.95	94.11	99.72	95.06	97.33
20x	97.32	98.40	97.87	94.65	99.59	97.20	98.38
30x	97.86	98.40	96.27	95.18	100.0	96.93	98.44

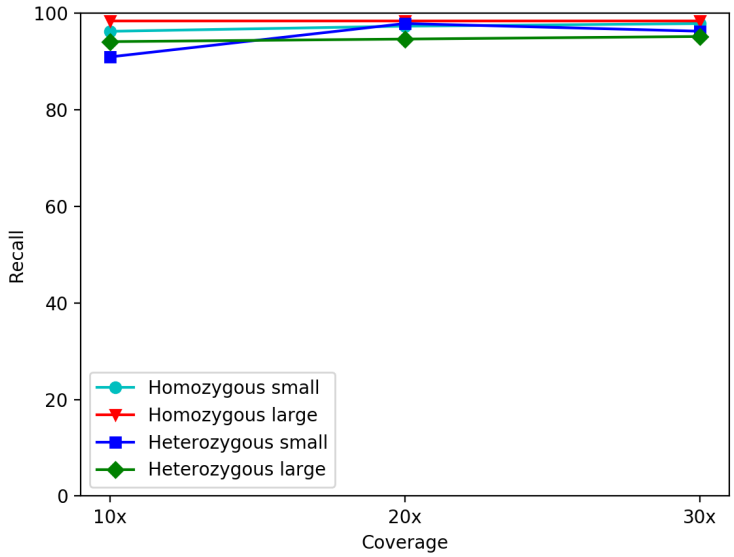

Figure 3.1: Recall values for homozygous and heterozygous deletions for varying sequence coverage on simulated data

Detection of insertions involve de-novo assembly of one end anchored reads and orphan reads for construction of contigs. Precision and recall metrics are compared for small and large insertions in Table 3.2 at varying sequence coverages. Recall values for homozygous and heterozygous insertions are shown in Figure 3.2.

Again, similar to deletions, we observe that the overall F-score increases with increase in sequence coverage due to a greater number of supporting reads. It can be observed from Table 3.2 that the recall values are significantly higher for homozygous as compared to heterozygous insertions for all sequence coverages. In case of low sequence coverage, we observe a low recall as the contig construction fails with low number of supporting reads. Also, for heterozygous insertions due to the same reason of lower number of supporting reads, a low recall is observed. While the recall is significantly higher for coverage > 10x, this indicates 20x as a good sequencing coverage for detection of insertions. A high precision value was also observed in all cases, indicating no false positives being detected.

Table 3.2**:** Precision, Recall and F-score metrics for predicting insertions using HyINDEL, on simulated data at varying sequence coverage

HyINDEL Insertions	**Homozygous recall**		**Heterozygous recall**		**Overall Precision**	**Overall Recall**	**Overall F-score**
	Small	**Large**	**Small**	**Large**			
10x	80.74	79.14	43.61	38.83	99.12	60.66	75.26
20x	85.56	87.16	78.72	79.78	99.67	82.93	90.53
30x	85.56	85.02	79.78	84.04	99.52	83.73	90.94

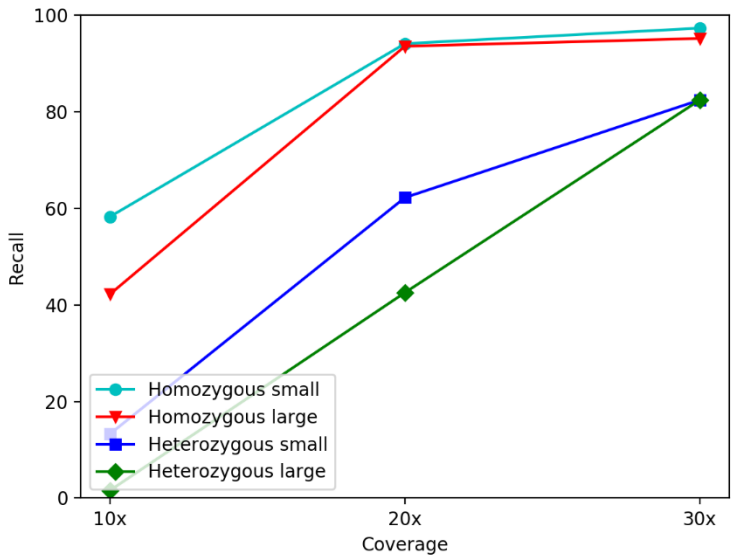

Figure 3.2: Recall values for homozygous and heterozygous insertions for varying sequence coverage on simulated data

Accuracy of Insertion sequences predicted

To evaluate performance of HyINDEL in the detection of insertions, we also compare the length of insertion sequence predicted with the actual insertion size in the simulated sample.

For all the insertions identified precisely, i.e., which could be successfully assembled using *de novo* assembly, we compare the error in insertion length prediction, defined as the absolute difference between the length of predicted insertion and length of actual insertion present at that position. The variation in error for 30x sample is shown as a boxplot, excluding outliers, in Figure 3.3. The median error in insertion length is observed to be 1bp. It can be observed that the insertion sequences are detected very accurately.

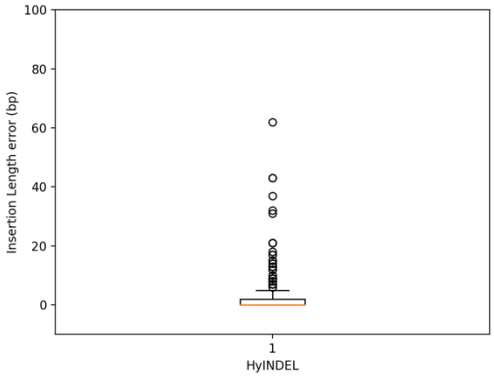

Figure 3.3: Error in insertion length for simulated sample at 30x using HyINDEL.

3.1.3. Comparison of HyINDEL with other tools

In this section we compare the INDEL predictions of our method HyINDEL with other tools *viz*., namely Lumpy (version: 0.2.14a) [13], TIDDIT (version: 2.6.0) [14], SoftSV (version: 1.4.2) [15] for deletions and Popins [17], Pamir [16] for insertion detection. The comparison is evaluated on parameters (i) Accuracy in terms of Precision, Recall, F-score metrics (ii) Breakpoint error, and (iii) Breakpoint support.

Accuracy

The Precision (P), Recall (R) and F-score (F) metrics are calculated for comparing the accuracy of each tool on the simulated data for different sequence coverage. The overall P, R, F values for deletions are summarized in Table 3.3 and for insertions in Table 3.4. F-score values for comparison of HyINDEL with other tools is shown in Figure 3.4 for deletions and Figure 3.5 for insertions.

With an increase in sequence coverage, the F-score value is observed to increase for all tools as expected. This is due to the increase in number of reads supporting an INDEL event. In case of deletions, the precision of all the tools (except SoftSV) was observed to be > 99% indicating no false positives being detected in the simulated data. A slightly lower precision

was observed in case of SoftSV due to deletions events being reported twice, once as a large deletion and again as a small deletion in the output files. It may be noted that recall and F-score values of HyINDEL are highest or comparable to other state-of-the-art tools, with precision values ~ 100%, clearly indicating the reliability of our predictions in detecting deletions.

It is observed from Table 3.4 that in case of insertions, precision of HyINDEL is > 99% even at 10× coverage, indicating the reliability of predictions. recall is observed to be low for HyINDEL and Pamir at 10× sequence coverage. The slightly lower values of precision of Pamir and Popins are observed to be due to multiple predictions of the same event. The recall and F-score values are comparable or higher compared to Pamir and Popins for ≥ 20× sequence coverage. It is observed that majority of insertions missed by all the three tools correspond to heterozygous insertions due to low read support.

Table 3.3**:** Precision, Recall and F-score metrics for predicting deletions using HyINDEL with other tools on simulated data at varying sequence coverag

Deletions	**10x**			**20x**			**30x**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
HyINDEL	99.72	**95.06**	**97.33**	99.59	**97.20**	**98.38**	**100.0**	96.93	**98.44**
Lumpy	**100.0**	89.06	94.21	**100.0**	95.33	97.61	**100.0**	96.13	98.02
TIDDIT	**100.0**	84.93	91.85	**100.0**	88.66	93.99	**100.0**	90.00	94.73
SoftSV	97.30	86.53	91.60	96.41	96.93	96.67	94.99	**98.66**	96.79

Table 3.4**:** Precision, Recall and F-score metrics for predicting insertions using HyINDEL with other tools on simulated data at varying sequence coverage

Insertions	**10x**			**20x**			**30x**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
HyINDEL	**99.12**	60.67	75.26	**99.67**	**82.93**	**90.53**	99.52	83.73	90.94
Pamir	96.88	37.33	53.89	97.64	71.86	82.79	95.73	**86.80**	**91.04**
Popins	88.12	**66.26**	**75.64**	98.98	78.00	87.24	**99.68**	83.60	90.93

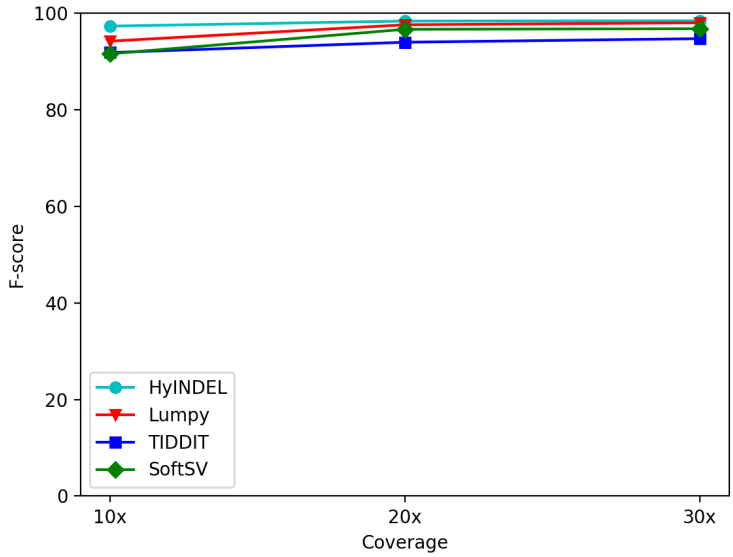

Figure 3.4: Comparison of F-scores for Deletions of our tool HyINDEL with other tools on simulated data at varying sequence coverage

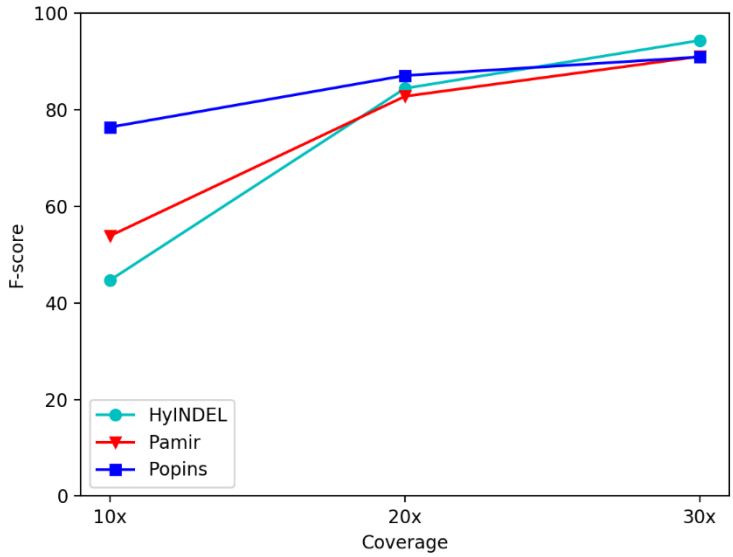

Figure 3.5: Comparison of F-scores for Insertions of our tool HyINDEL with other tools on simulated data at varying sequence coverage

Breakpoint error

To measure the accuracy in breakpoint predictions, we define the breakpoint error as the absolute difference between the predicted breakpoint positions and the actual breakpoint coordinates for each detected event. In case of deletions, the breakpoint error is calculated as the sum of breakpoint errors at the 5′ and 3′ breakpoints. The breakpoint errors for the detection

are depicted in Figure 3.6 and in Figure 3.7 for insertions for 30x sequence coverage excluding outliers. The median breakpoint error is observed to be 1 for deletions and 0 for insertions, indicating the accuracy of HyINDEL predictions because of using softclip/split reads.

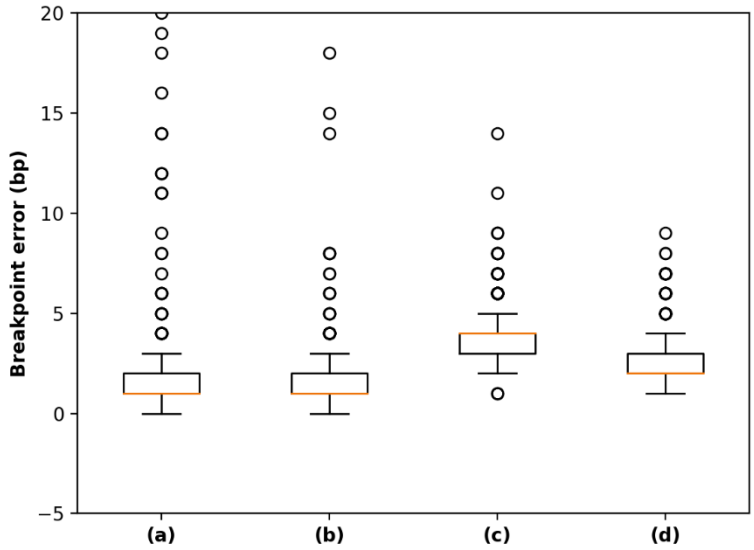

Figure 3.6: Breakpoint error in detection of deletions using (a) HyINDEL, (b) Lumpy, (c) TIDDIT and (d) SoftSV is shown on simulated data at 30x sequencing coverage

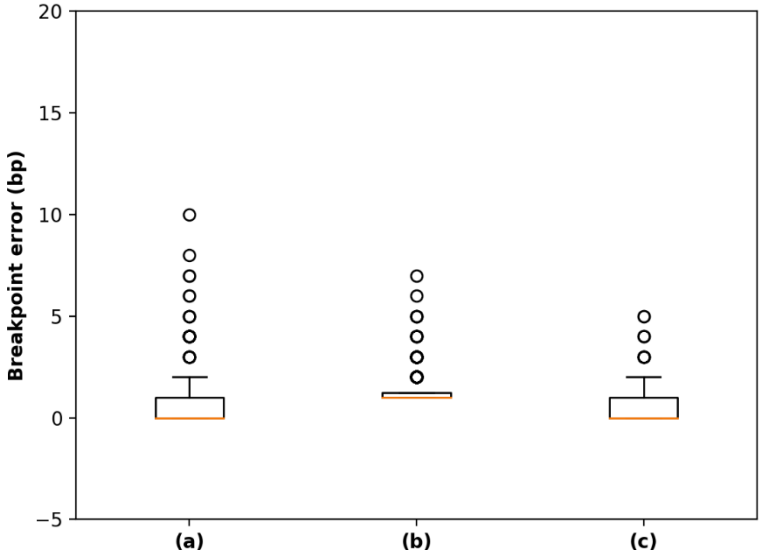

Figure 3.7: Breakpoint error for insertions using (a) HyINDEL (b) Pamir (c) Popins is shown on simulated data at 30x sequencing coverage

Breakpoint support

The softclip/split reads provide information about the exact breakpoint information, while the discordant reads provide additional supporting information for the INDEL event. Each tool reports the number and type of reads supporting the predicted event. In Figure 3.8 the distribution of number of softclip/split reads for all deletion events observed in the case of 30x sequence coverage sample is shown. It may be noted that the median number of reads supporting the breakpoint is ~ 21 for HyINDEL, much higher than the other tools, Lumpy (7), TIDDIT (7) and SoftSV (8). This is due to the usage of softclip reads by our method.

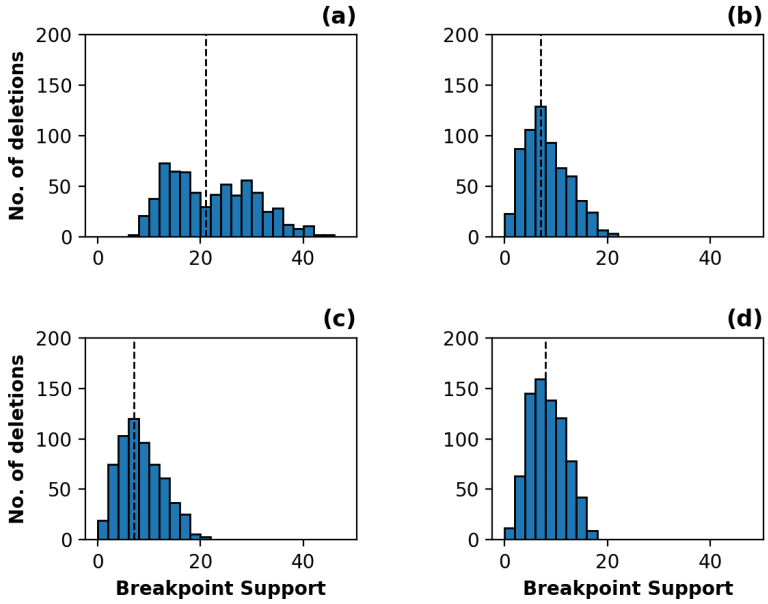

Figure 3.8: Breakpoint support for deletions (without discordant reads) for (a) HyINDEL, (b) Lumpy, (c) TIDDIT and (d) SoftSV is shown on simulated data at 30x sequencing coverage.

Time and Memory usage

We have run HyINDEL on a single node of a cluster, to analyze the performance. The configuration of node is a HP SL230 compute nodes with two Intel E5-2640 processors having 12 cores each, that is a total number of 24 cores. The maximum memory assigned to each CPU is 2048 MB, with the node having a total memory of 48GB. The time and peak memory usage for varying sequence coverage are summarized in Table 3.5. Peak memory usage is taken to be the Maximum resident set size value estimated using time command.

Table 3.5**:** Comparison of time and memory utilization for varying sequence coverage

(h: hours, m: minutes, s: seconds)

Coverage	Wall clock Time	User Time	System Time	Peak memory
10x	42m18s	5h41m	11m13s	1.07GB
20x	1h10m	9h7m	20m28s	1.06GB
30x	1h16m	6h33m	21m43s	1.07GB

3.2. Comparison on Real data

Next we analyzed the performance our method on real dataset. For this the widely studied NA12878 sample of a Caucasian female of UTAH/MORMON ethnicity from Illumina Platinum Genomes repository (ENA accession: PRJEB3381) [25]. We have used the PCR-free high coverage reads for our analysis. The reads are aligned using BWA-MEM [11] to the human reference (assembly GRCh37). The resulting BAM files are sorted by coordinate and then indexed using Samtools [24]. The sorted BAM file and its index are given as input for all tools. The reason for considering NA12878 sample is that annotations for INDELs are available from three resources, *viz.*, Genome in a bottle (GIAB) [29], Database of Genomic Variants (DGV) [30] and PacBio SV annotations [31]. The data contains paired-end reads of length 101bp sequenced using Illumina HiSeq 2000. The median insert size was estimated to be 318 and standard deviation in insert size as 78 using Picard tools [12]. The average genome coverage was estimated to be 52x using Mosdepth [18]. The reads (fastq files) were given as input to Fastqc [32] and the final BAM file as input to Bamqc [33]. Output files generated represented the files passing all quality filters (represented as green tick marks).

3.2.1. Benchmarking Dataset

For evaluation, INDELs of size ≥ 50bp are only considered. For insertions, a prediction is considered true positive if it lies within 200bp of an actual event.

Genome in a bottle (GIAB)

For the NA12878 sample, SVClassify [29] has generated a high-quality benchmark for structural variations by combining multiple forms of evidence using multiple reads from multiple sources and involving multiple sequencing technologies. Briefly, features/annotations are constructed inside and around candidate SVs and unsupervised machine learning is used to determine characteristics of various SV types using one-class model to classify candidate SVs. 39/40 calls were validated using PCR for which primer design was possible. Further, deletions

were validated using trio analysis of her father (NA12891) and mother (NA12892) and verifying if the calls were Mendelian consistent i.e., also present in parents. This was done using MetaSV, which incorporates multiple SV detection algorithms. The validation rate was 99.7% for the calls in the benchmark, indicating a very high quality. The benchmark was deposited in Genome in a bottle consortium. The set contains 2676 (1854 small, 822 large) deletions and 68 insertions. The annotation set was downloaded from https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/technical/svclassify_Manuscript/Supplementary_Information/Personalis_1000_Genomes_deduplicated_deletions.bed and https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/technical/svclassify_Manuscript/Supplementary_Information/Spiral_Genetics_insertions.bed .The precision recall and F-score metrics for predictions are computed for all the tools using GIAB annotations as reference and the results are summarized in Table 3.6 for deletions. For insertions the results are given in terms of number of true predictions in Table 3.10 and Table 3.10, based on the distance criteria of true positive being 10bp and 200bp respectively.

Database of Genomic Variants (DGV)

Database of Genomic Variants contains a curated catalog of SVs identified in healthy control samples from various studies. We have used the latest version (Release Data: 2016-05-15) corresponding to GRCh37. It contains 108 novel sequence insertions reported from the 1000 genomes project. We use this set for benchmarking insertions predictions. Other novel sequence insertions reported are not considered for our evaluation as they are using array CGH methods. The annotation set was downloaded from http://dgv.tcag.ca/dgv/docs/GRCh37_hg19_variants_2016-05-15.txt and variants for the NA12878 sample with "novel sequence insertion" are considered. The results for deletions are summarized in Table 3.7 and insertions are discussed in terms of number of true predictions in Table 3.10 and Table 3.10, based on the distance criteria of true positive being 10bp and 200bp respectively.

PacBio SV annotations

For the NA12878 sample, SVs were identified using PacBio long reads. The annotations from PacBio were merged with variants from DGV and GIAB , the resulting set was used in a comprehensive evaluation of 69 SV callers [31]. Small deletions and insertions from DGV were not considered as they were already represented in the PacBio annotations. Deletions with a high reciprocal overlap (> 90%) with PacBio deletions were also removed.

This merged benchmark dataset contains 9241 (6062 small, 3179 large) deletions and 13669 (12538 small, 1131 large) insertions. In total there were 7322 deletions and 12686 insertions reported from PacBio in the merged benchmark. The annotations were downloaded from https://raw.githubusercontent.com/stat-lab/EvalSVcallers/master/Ref_SV/NA12878_DGV-2016_LR-assembly.vcf .The precision recall and F-score metrics for the tools computed using PacBio-merged benchmark dataset are summarized in Table 3.8 for deletions.

For insertions, there is no subclass information (Novel sequence insertions) available. It has been previously reported [16] that the predictions made using long reads were located in repeat regions, GC biased regions and only 488 of the total 12998 insertions could be identified using Pamir tool in a study using CHM1 cell line. The major reason being short Illumina reads could not be assembled in those regions.

Table 3.6: Precision, Recall and F-score metrics for predicting deletions using HyINDEL with other tools on real data (NA12878) using GIAB benchmark (n=2676)

Deletions	**Overall Precision**	**Recall**			**Overall F-score**
		Small	**Large**	**Overall**	
HyINDEL	65.86	**86.40**	90.63	**87.74**	**75.24**
Lumpy	53.82	83.76	**95.13**	87.29	66.59
TIDDIT	**76.51**	58.14	79.19	64.64	70.08
SoftSV	37.65	77.07	74.33	76.27	50.41

Table 3.7: Precision, Recall and F-score metrics for predicting deletions using HyINDEL with other tools on real data (NA12878) using DGV benchmark (n=973)

Deletions	**Overall Precision**	**Recall**			**Overall F-score**
		Small	**Large**	**Overall**	
HyINDEL	19.86	**85.81**	70.43	72.76	31.20
Lumpy	17.16	80.85	**75.72**	**76.56**	28.04
TIDDIT	**31.44**	66.67	74.03	73.07	**43.97**
SoftSV	10.79	71.63	58.05	60.12	18.29

Table 3.8**:** Precision, Recall and F-score metrics for predicting deletions using HyINDEL with other tools on real data (NA12878) using Pacbio-merged benchmark (n=9241)

Deletions	**Overall Precision**	**Recall**			**Overall F-score**
		Small	**Large**	**Overall**	
HyINDEL	79.57	**31.70**	28.75	30.7	**44.30**
Lumpy	66.22	30.07	**33.02**	**31.10**	42.32
TIDDIT	**87.74**	19.20	25.76	21.47	34.49
SoftSV	47.31	29.28	24.81	27.75	34.98

Table 3.9**:** Comparison of number of true insertions on real data (NA12878) by HyINDEL with other tools. The results are summarized for annotations on DGV (n=108), GIAB (n=68) and PacBio (n=13669) benchmarks

Insertions	**DGV**	**GIAB**	**PacBio**
HyINDEL (n=896)	52	19	42
Pamir (n=5820)	**58**	11	39
Popins (n=3204)	18	**26**	**96**

Table 3.10**:** Comparison of number of true insertion on real data (NA12878) by HyINDEL with other tools. The results are summarized for annotations on DGV (n=108), GIAB (n=68) and PacBio (n=13669) benchmarks

Insertions	**DGV**	**GIAB**	**PacBio**
HyINDEL (n=896)	60	42	611
Pamir (n=5820)	**66**	20	**1409**
Popins (n=3204)	22	**53**	861

3.2.2. Performance of HyINDEL

In total HyINDEL identified 3672 (2684 small, 988 large) deletions and 896 (220 small, 15 large, 661 imprecise) insertions on the NA12878 sample. The size of largest deletion identified was 49025bp, the mean deletion size was 1076.32 and median size 308. While the largest insertion was 4135bp, the mean insertion size was 174.73 and median 75. The distribution of the size of deletions and insertion predicted are shown in Figure 3.9 and Figure 3.10 respectively.

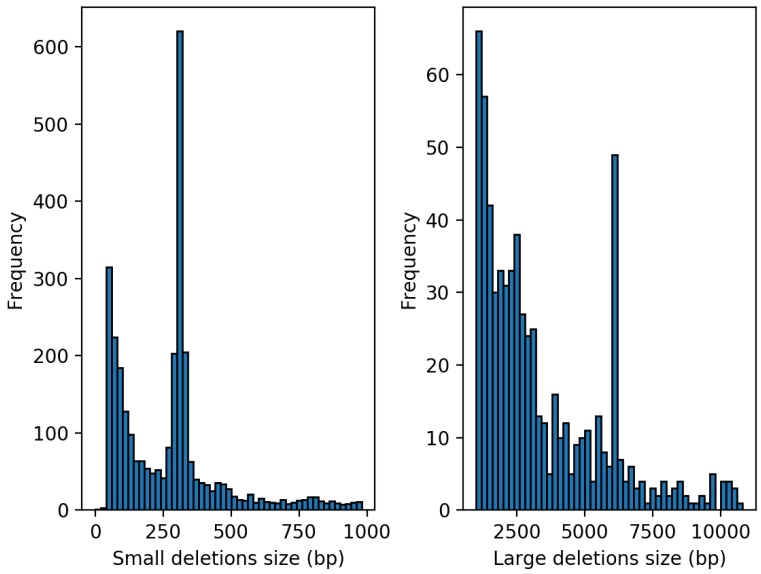

Figure 3.9: Size distribution of deletions identified on real data, NA12878 using our tool, HyINDEL

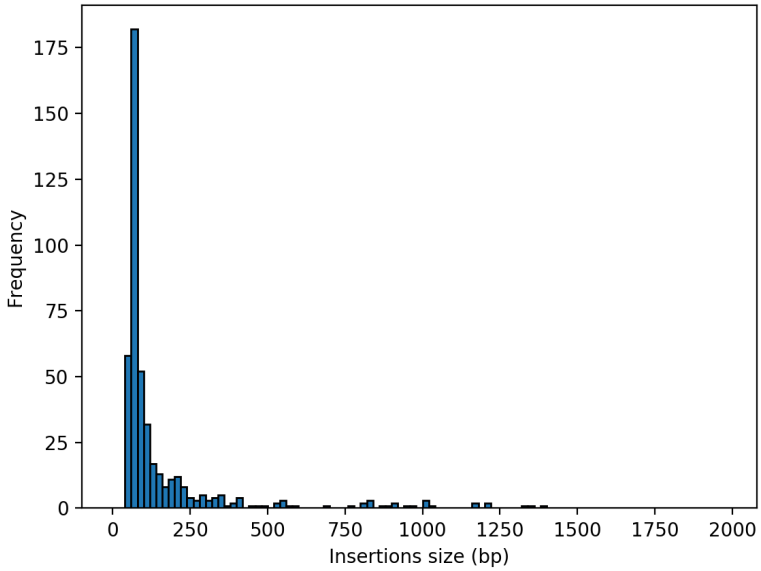

Figure 3.10: Size distribution of insertions identified on real data, NA12878 using our tool, HyINDEL

For deletions, we compare our predictions on two benchmarks GIAB and PacBio-merged. On the GIAB benchmark, we observe a high recall of 87.74%. The recall for large deletions is slightly higher than small deletions, this is expected since large deletions are easier to identify compared to smaller ones. True deletion events missed were mainly due to low

number of supporting reads. From the PacBio-merged benchmark, we observe a high precision of 79.6%, indicating a high number of true deletion calls.

In case of insertions, we compared our predictions on three benchmarks DGV, GIAB and PacBio-merged in Table 3.9 and Table 3.10. We identified 60/108 is using 200bp error novel sequence insertions reported in the DGV benchmark. The GIAB and PacBio-merged benchmark do not have subclass information for insertions. 19 and 42 insertions were identified in the GIAB and PacBio-merged benchmark respectively. Changing the true positive criterion for insertions, the maximum distance between prediction and actual event from 10bp to 200bp, increased the number of true positives identified on each of the benchmark. It may be noted that the true predictions increase to 611 on the PacBio-merged benchmark, while 60 and 42 insertions were detected on the DGV and GIAB benchmark respectively.

Time and memory usage

HyINDEL was run on a single compute node with 24 cores, 48 GB memory (same node used in simulated experiments). It took 2 hours 49 minutes for running on real data sample.

3.2.3. Comparison of HyINDEL with other tools

It may be noted from Table 3.6 that the performance of HyINDEL in detecting deletions is better than the other tools on GIAB benchmark dataset, as indicated by F-score values. Recall values of HyINDEL in detecting deletions is comparable with Lumpy and much higher than TIDDIT and SoftSV. Though TIDDIT exhibits higher precision (~ 76.5) compared to other tools, however low recall values suggest that large number of deletion events are missed by it. Precision of HyINDEL is much higher than Lumpy resulting is a higher F-score. For the DGV dataset (Table 3.7), HyINDEL exhibits higher recall for small deletions, while it is higher for Lumpy in case of detection of large deletions. TIDDIT exhibits highest precision and recall values comparable with other tools, resulting in better F-score on DGV dataset. Against the PacBio-merged benchmark (Table 3.8), we observe HyINDEL exhibiting higher recall for small deletions and the best F-score.

In the case of insertions (Table 3.10), relatively fewer predictions (870) are observed with HyINDEL compared with other tools Pamir (5820) and Popins (3204), indicating much lower false positives. On the DGV dataset, the number of true predictions by HyINDEL (65/108) is comparable with Pamir (66/108), while on the DGV dataset, HyINDEL exhibited

43/68 insertions, higher than Pamir (20/68) and comparable to Popins (53/68) on GIAB dataset, but with a significantly higher true positive rate.

Further, we also calculated the number of common insertions between each tool. Between HyINDEL and Pamir, there are 278 common insertions, while between HyINDEL and Popins there are 401. Number of common insertions between all 3 tools were 108.

Chapter 4 Conclusion

Here we proposed a hybrid approach for the detection of both insertions and deletions on a single platform from next generation sequencing data. Using soft-clip reads, HyINDEL is able to provide good support in accurately detecting the INDEL breakpoints compared to other methods, indicating the reliability of our predictions. HyINDEL is able to handle detection of both small and large indels and also identify the novel insertion sequence. Our analysis indicates that the performance of HyINDEL is comparable with other state-of-the-art tools on both simulated and real data. In future we propose to incorporate detection of INDELs in paired case-control data (e.g., tumour-normal) and multiple genomic sequencing data for the detection of population specific INDELs. The results of experiments on simulated data and real data are summarized below.

4.1 Simulated data

For simulated data, we have inserted 750 homozygous and heterozygous indels each, of varying lengths into the human genome (assembly GRCh37). Paired-end reads were generated corresponding to 3 different sequencing coverages (10x, 20x, 30x). Alignment and index files were generated for each sample. Performance of our method is evaluated in terms of accuracy (Precision, Recall and F-score), breakpoint error, breakpoint support. The effect of sequencing coverage with respect to size (small/large) and type (deletions/insertions) is also discussed. We show that F-score increase with an increase in sequencing coverage. In case of deletions, we observe that it is easier to detect large deletions as compared to smaller ones, due to the presence of both discordant and softclip signals. While, in case of insertions we observe that it is easier to detect small insertions than larger ones, as large insertions additionally need an orphan contig to be successfully assembled. The median breakpoint error was calculated to be 1 and 0 at 30x sequencing coverage for deletions and insertions respectively, indicating very accurate breakpoint predictions due to the usage of soft-clipped reads. We have also compared and benchmarked the accuracy of our predictions with state-of-the-art tools for indel detection. We have shown that our method has the highest F-score at all sequencing coverages for deletions and comparable F-score at 20x and 30x coverage for insertions. We also observe that detection of heterozygous indels is difficult as compared to homozygous indels, mainly due to the lower number of supporting reads.

4.2 Real data

We have used the widely studied NA12878 sample from the Illumina Platinum genomes repository. Paired-end reads were aligned to the human genome (assembly GRCh37) and the resulting alignment file is given as input to our tool. A total of 3672 deletions and 896 insertions were predicted. Annotations from Genome-in-a-bottle, Database of genomic variants and PacBio SV were used to benchmark indel predictions. We observe a high recall for both small and large deletions on the GIAB benchmark. While we observe a high precision for deletions on the PacBio benchmark and lower recall as a majority of annotations are from third generation sequencing technologies which are located in repeat and GC biased regions. In case of insertions, we identified 60 novel sequence insertions from the DGV benchmark and 397 on the PacBio benchmark. In comparison to other tools, our method has the highest recall and F-score for deletions on the GIAB and PacBio benchmarks. In case of insertions, the number of predictions by our method are lower as compared to other tools, we observe a significantly high true positive rate.

A major addition in our tool as compared to others is the use of Soft-clip reads for enhancing variation detection. This has resulted in significant increase in recall in case of deletions as seen on multiple real data benchmarks. In case of insertions, we perform comparably to the state-of-the-art tools on DGV and GIAB benchmarks, while observing a significant increase in precision.

It is observed that only a subset of variants is identified using second-generation methods when compared to variations identified from Pacbio-merged benchmark (containing variations from third-generation sequencing methods), as many of the variants reported are from biased regions which cannot be accurately detected.

RELATED PUBLICATIONS

HyINDEL – A Hybrid approach for Detection of Insertions and Deletions

Alok Thatikunta, Nita Parekh

bioRxiv 2021.10.08.463662; doi: https://doi.org/10.1101/2021.10.08.463662

BIBLIOGRAPHY

[1] W.-K. Sung, Algorithms for next-generation sequencing, Chapman and Hall/CRC, 2017.

[2] G. Escaramís, E. Docampo and R. Rabionet, "A decade of structural variants: description, history and methods to detect structural variation," *Briefings in functional genomics,* vol. 14, no. 5, pp. 305-314, 2015.

[3] "Wikipedia Shotgun Sequencing," [Online]. Available: https://en.wikipedia.org/wiki/Shotgun_sequencing.

[4] J. Shendure, S. Balasubramanian, G. M. Church, W. Gilbert, J. Rogers, J. A. Schloss and R. H. Waterston, "DNA sequencing at 40: past, present and future," *Nature,* vol. 550, no. 7676, pp. 345-353, 2017.

[5] "Wikipedia Maxam Gilbert sequencing," [Online]. Available: https://en.wikipedia.org/wiki/Maxam%E2%80%93Gilbert_sequencing.

[6] "Zhong lab Sequencing technologies," [Online]. Available: https://zhonglab.gitbook.io/3dgenome/chap0-preparation/0.2-sequencing-technologies.

[7] L. Liu, Y. Li, S. Li, N. Hu, Y. He, R. Pong, D. Lin, L. Lu and M. Law, "Comparison of next-generation sequencing systems," in *BioMed Research International*, 2012.

[8] J. Shang, F. Zhu, W. Vongsangnak, Y. Tang, W. Zhang and B. Shen, "Evaluation and comparison of multiple aligners for next-generation sequencing data analysis," in *BioMed research international*, 2014.

[9] "Novoalign," [Online]. Available: http://www.novocraft.com/.

[10] B. Langmead, C. Trapnell, M. Pop and S. L. Salzberg, "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome," *Genome biology,* vol. 10, no. 3, p. R25, 2009.

[11] H. Li, "Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM," arXiv preprint arXiv:1303.3997, 2013.

[12] "Picard," [Online]. Available: https://broadinstitute.github.io/picard/.

[13] R. M. Layer, C. Chiang, A. R. Quinlan and I. M. Hall, "LUMPY: a probabilistic framework for structural variant discovery," *Genome biology,* vol. 15, no. 6, p. R84, 2014.

[14] J. Eisfeldt, F. Vezzil, P. Olason, D. Nilsson and A. Lindstrand, "TIDDIT, an efficient and comprehensive structural variant caller for massive parallel sequencing data," *F1000Research,* vol. 6, 2017.

[15] C. Bartenhagen and M. Dugas, "Robust and exact structural variation detection with paired-end and soft-clipped alignments: SoftSV compared with eight algorithms," *Briefings in bioinformatics,* vol. 17, no. 1, pp. 51-62, 2015.

[16] P. Kavak, Y.-Y. Lin, I. Numanagić, H. Asghari, T. Güngör, C. Alkan and F. Hach, "Discovery and genotyping of novel sequence insertions in many sequenced individuals," *Bioinformatics,* vol. 33, no. 14, pp. i161-i169, 2017.

[17] B. Kehr, P. Melsted and B. V. Halldórsson, "PopIns: population-scale detection of novel sequence insertions," *Bioinformatics,* vol. 32, no. 7, pp. 961-967, 2015.

[18] B. S. Pedersen and A. R. Quinlan, "Mosdepth: quick coverage calculation for genomes and exomes," *Bioinformatics,* vol. 34, no. 5, pp. 867-868, 2017.

[19] R. Chikhi and G. Rizk, "Space-efficient and exact de Bruijn graph representation based on a Bloom filter," *Algorithms for Molecular Biology,* vol. 8, no. 1, p. 22, 2013.

[20] H. Li, "Minimap2: pairwise alignment for nucleotide sequences," *Bioinformatics,* vol. 34, no. 18, pp. 3094-3100, 2018.

[21] D. W. Barnett, E. K. Garrison, A. R. Quinlan, M. P. Strömberg and G. T. Marth, "BamTools: a C++ API and toolkit for analyzing and managing BAM files," *Bioinformatics,* vol. 27, no. 12, pp. 1691-1692, 2011.

[22] "Github Transwarp," [Online]. Available: https://github.com/bloomen/transwarp.

[23] "Github Args," [Online]. Available: https://github.com/Taywee/args.

[24] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis and R. Durbin, "The sequence alignment/map format and SAMtools," *Bioinformatics,* vol. 25, no. 16, pp. 2078-2079, 2009.

[25] M. A. Eberle, E. Fritzilas, P. Krusche, M. Källberg, B. L. Moore, M. A. Bekritsky and Z. I. e. al, "A reference data set of 5.4 million phased human variants validated by genetic inheritance from sequencing a three-generation 17-member pedigree," *Genome research,* vol. 27, no. 1, pp. 157-164, 2017.

[26] "Github SVSim," [Online]. Available: https://github.com/GregoryFaust/SVsim.

[27] W. Huang, L. Li, J. R. Myers and G. T. Marth, "ART: a next-generation sequencing read simulator," *Bioinformatics,* vol. 28, no. 4, pp. 593-594, 2011.

[28] "Coriell GM12878," [Online]. Available: https://www.coriell.org/0/Sections/Search/Sample_Detail.aspx?Ref=GM12878.

[29] H. Parikh, M. Mohiyuddin, H. Y. Lam, H. Iyer, D. Chen, M. Pratt and G. B. e. al, "svclassify: a method to establish benchmark structural variant calls," *BMC genomics,* vol. 17, no. 1, p. 64, 2016.

[30] M. JR, Z. R, Y. RK, F. L and S. SW, "The database of genomic variants: a curated collection of structural variation in the human genome," in *Nucleic Acids Res*, 2013.

[31] S. Kosugi, Y. Momozawa, X. Liu, C. Terao, M. Kubo and Y. Kamatani, "Comprehensive evaluation of structural variation detection algorithms for whole genome sequencing," *Genome biology,* vol. 20, no. 1, p. 117, 2019.

[32] S. Andrews, "FastQC: a quality control tool for high throughput sequence data," 2010.

[33] S. Andrews, "BamQC," 2016.