FREIE UNIVERSITÄT BERLIN

MASTER THESIS

in

BIOINFORMATICS

Large Structural Variations Detection in Hi-C Maps

Author
Emel ÇOMAK

23^{rd} September, 2019

Master Thesis

Free University of Berlin, Department of Mathematics and Computer Science,
Bioinformatics Master program

Max-Planck-Institute for Molecular Genetics

Large Structural Variations Detection in Hi-C Maps

from

Emel Çomak
Matriculation number: 5214966
emec93@zedat.fu-berlin.de

from 15.04.2019 to 23.09.2019
at Max-Planck-Institute for Molecular Genetics, Ihnestraße 63, 14195 Berlin

1st Supervisor: Dr. Robert Schöpflin (robert.schoepflin@charite.de)
2nd Supervisor: Prof. Dr. Martin Vingron(martin.vingron@molgen.mpg.de)

Berlin, 21st September, 2019

Abstract

Structural variations (SVs) are a large class of alterations in the genome. SVs can alter the spatial organization of the chromatin by rearranging the parts of the genome. As a result, formerly distal parts of the genome can be proximal after an SV. SVs can be related to many diseases from cancer to obesity. Despite their importance, the identification of structural variants in the genome remains challenging. Here we issue SV calling by analyzing data from Hi-C experiments. Hi-C is a chromosome conformation capture technique combined with high-throughput sequencing that probes the chromatin architecture by counting the genomic regions that are in contact in the 3D space. These contact counts are used to generate a genome-wide contact matrix which can be visualized with a heatmap. Large SVs create high contact frequency patterns in the Hi-C map due to the spatial proximity between formerly distal genomic regions induced by the rearrangement. In this thesis, a tool has been developed to detect these SV induced patterns in Hi-C maps. As an input, the tool requires a reference and a sample Hi-C file and returns files with candidate coordinates of SVs. The analysis starts in a low-resolution Hi-C map, detects candidates and move then on to a higher resolution with the detected candidate regions. At each resolution stage, a pre-trained classification model with features influenced by the SV induced patterns is applied to the interested region. At 25 kb resolution, final SV breakpoint candidates are obtained.

Acknowledgement

I would like to express my sincere gratitude to my supervisor Dr. Robert Schöpflin for his leadership, immense knowledge, patience and support throughout my master study and thesis. I am grateful to him and Prof. Martin Vingron for being my mentors and molding me into a researcher. I would also wish to express my gratitude to Prof. Stefan Mundlos and his development disease group for providing the Hi-C data. I would like to thank Dr. Uira Melo for his suggestions and discussions. I am thankful to Vingron group members for letting me share my happiness and stress with them throughout this thesis. Finally, I would like to thank my family for being there for me whenever I had struggle standing on my own two feet.

Contents

Chapter 1

Introduction

All the information needed to build and maintain an organism contains in an organism's genome. On average, a human genome contains around 3.2 billion DNA base pairs. The information contained in a genome is carried in all cells in a human body. The genome has a spatial organization in the nucleus to fit a large amount of information. The physical description of the genome organization in the nucleus can be portrayed as in the following. The genome is partitioned into smaller subsets of DNA called chromosomes. In human, somatic cells contain 22 different types of autosomes, each present as a set of pairs (diploid), and two sex chromosomes that makes in total 46 chromosomes. Each chromosome is formed by two chromatids that are conjugated to each other from the centromere. Chromosomes occupy specific territories and together they are contained in chromatin in the nucleus. Functionally, chromatin can be separated into A and B compartments. Chromatin in the A is known to be accessible and transcriptionally active while chromatin in the B compartment is more compact and transcriptionally more silent. Parts of the A compartment are predominantly located towards the center while parts of the B compartment are usually located near the periphery of the nucleus. Besides, for chromatin, there is a further organizational principle. Chromatin organization can also be partitioned into lamina-associated domains (LADs) and topologically associating domains (TADs). LADs consist mostly of transcriptionally silent parts of the chromatin so B compartments are rich in LADs. TADs are defined as self-interacting domains. This means chromatin within a TAD physically interacts with each other more frequently than with chromatin outside the TAD [1]. The genome organization in the nucleus is schematically shown in Figure 1.1 to provide a better understanding of the basic terminology.

Even though almost all cells in an organism share the same DNA sequence, cells can be morphologically and functionally very different. The key term to explain this phenomenon is gene regulation. Some parts of DNA ($\sim$ 2%) are called the coding regions that include genes. Genes are responsible for composing proteins by being transcribed into an RNA molecule. Cells need specific proteins, thus not all genes are expressed all the time. For an organism, it is crucial, that a gene is activated at the right time and in the right part of the body. Regulatory elements play an important role in gene regula-

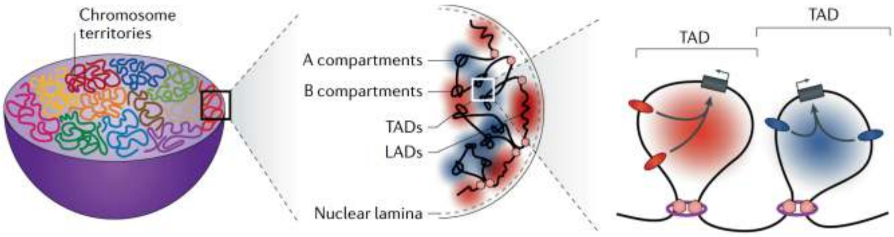


Figure 1.1: Chromatin organization schematic is shown. Chromosomes are, part of chromatin, contained in the nucleus are shown on the right. Chromatin is divided into A and B compartments that are shown in blue and red. There are also topologically associating domains (TADs) and lamina-associated domains (LADs). TADs are more to the center and usually transcriptionally active while TADs are close to the nucleus periphery. On the left, two TADs are shown. TAD boundaries and cohesin are shown by pink circles and rings at the end of the loops. Regulatory elements and genes prefer to interact within their TAD boundaries. Red regulatory elements interact in the red TAD and blue regulatory elements interact in the blue TAD. This figure is adopted [2].

tion. Two main classes of regulatory elements are promoters and enhancers of genes. On a linear genome regulatory elements can be far away from each other. However, the distal regulatory elements still need to interact with their cognate genes to activate transcription. Indeed, in the nucleus, the genome is not linear and has loops, some of which forms TADs, to enable interactions between regulatory elements and cognate genes. An alteration in the genome can disrupt the regulatory landscape, such as TADs, by removing or moving regulatory elements somewhere else or by bringing new regulatory elements into the regulatory landscape of the gene. This rearrangement in the genome can result in diseases. But, it is important to mention that alternations in a genome do not always conclude in diseases. For instance, it is known that between two individuals approximately 20 million base pairs of differences in their genomes exist [3]. Alterations of the genome are the basis for evolution, by gains or losses of genes as well as alteration of gene regulation. The general definition of alterations in a genome is termed genetic variations. The high amount of genomic variations raises the question of which variations might be disease-related. To answer this question, first, all genetic variations have to be detected and further interpreted in terms of their disease potential.

1.1 Structural Variations

Typically, genetic variations can be grouped into three main categories; single nucleotide polymorphisms (SNPs), small insertions-deletions (indels) and structural variations (SVs). SNPs are alterations of a single nucleotide (base). Small insertions-deletions (indels) are known to be insertions or deletions of a single base or a couple of bases from the genome sequence. Structural variations (SVs) are a collective term for genomic rearrangements that affect regions larger than 50 bp.

SVs can be grouped into different main types: insertions, deletions, tandem duplications, interspersed duplications, inversions and translocations as shown in Figure 1.2 [4]. Moreover, SVs can occur in nested forms when several variation events occur on top of each other and produce several breakpoints. These nested forms are common in chromothripsis where genome shatters and tries to piece itself back together [5]. Nested forms of SVs are also characteristics of cancer where the genome is unstable and cannot regulate itself properly anymore [6]. One of the challenges in SV identification comes from the fact that SVs can have more complex structures than other genomic variants.

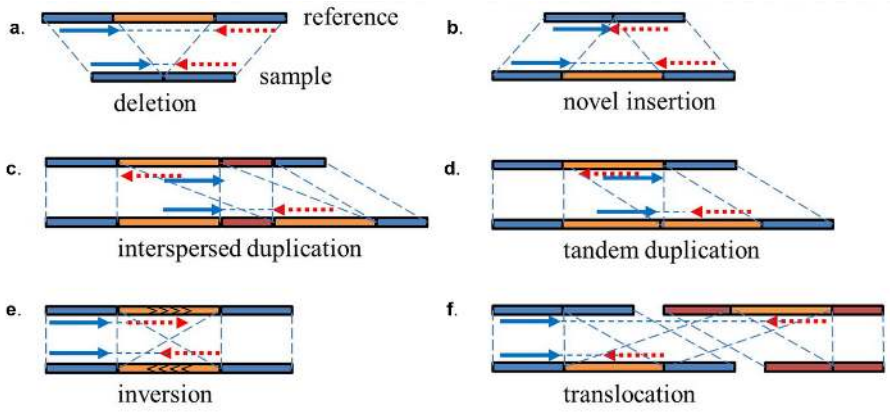


Figure 1.2: Schematic representation of structural variations (SVs). From A to F cartoon: Deletion, insertion, interspersed duplication, tandem duplication, inversion and translocation. For each cartoon upper line represents reference genome sequence, lower line represents sample genome sequence. Arrows indicate read orientations This figure is adopted. [4].

Studies have shown that SVs have a substantial impact on diseases and evolution [7–9]. For example, as illustrated in Figure 1.3, SVs can disrupt chromatin architecture which may result in a gene misregulation by changing the spatial positions between regulatory elements of genes. Another example to show the effect of SVs is that in genomes of cancer cells, which can have plenty of SVs, chromatin organization is much more different and less stable than in non-affected cells [10]. SVs influence a wide range of diseases from neurological diseases to Mendelian disorders and obesity [11][12]. According to Redon et al., more base pairs alters due to SVs than SNPs [13] which also points out the importance of SV detection for research and diagnostics.

Over the decades' several tools have been established for finding SVs. Different technologies have different throughputs and resolutions. One of the methods for SV detection is known as karyotyping which is based on staining of condensed metaphase chromosomes and microscopy. Only large SVs can be detected since it suffers from low resolution. Yet, it is a commonly used method in clinics. The second method worth to mention is fluorescence in situ hybridization (FISH). In FISH, fluorescently labeled DNA probes are designed for specific loci of interest. After the probes are hybridized to the sample genome, it can be checked if the probes hybridized at the expected locations or the loci of interest are rearranged [14]. This method is useful when the targeted SV region is known. Another SV detection method is called Array Comparative Genomic Hybridization (aCGH) [15]. It is microarray technology. A set of probes is designed to cover the whole genome. Unbalanced SVs can be detected by measuring the relative copy number changes between two samples (usually the matched normal and disease samples).

Next-generation sequencing (NGS) expedited the discoveries of new methods in SV detection. NGS provides cheap and reliable large-scale DNA sequencing. It is used extensively for de novo sequencing, for quantifying expression levels through RNA sequencing and in population genetic studies. Specifically, variant calling methods have substantially developed through NGS [16]. In NGS methods, a whole-genome, or targeted regions of the genome, is randomly digested into small fragments (or short reads) that get sequenced and are then either aligned to a reference genome or assembled. Whole-genome sequencing (WGS) with short reads are enough to detect nearly all SNPs and indels. However, short reads suffer in SV detection. One problem is that the genome has a lot of repetitive regions. In these regions mapping the short reads is too difficult. Another problem is that in case of a balanced SV detection is only possible if the reads span the breakpoint. Because of these reasons, they usually suffer from low sensitivity and high false discovery rates in SV detection [17]. NGS technology has recently evolved into 3^{rd} generation sequencing. Today, it is possible to do whole genome sequencing with long reads. As a fourth method to mention, long read whole genome sequencing can overcome repetitive regions by spanning larger regions of the genome. However, so far they have high error rates. The fifth method is newly developed Optical Mapping (OM). In OM, DNA molecules are cut at specific regions by restriction enzymes. Later, each DNA molecule at the end of the introduced brake is stained by fluorescent dye and determined with optical methods. In the end, pieced DNA is mapped. This method is capable of producing high-resolution, high-throughput genomic map data that gives information about the structure of a genome [18]. However, OM is an expensive and high level of precision required method.

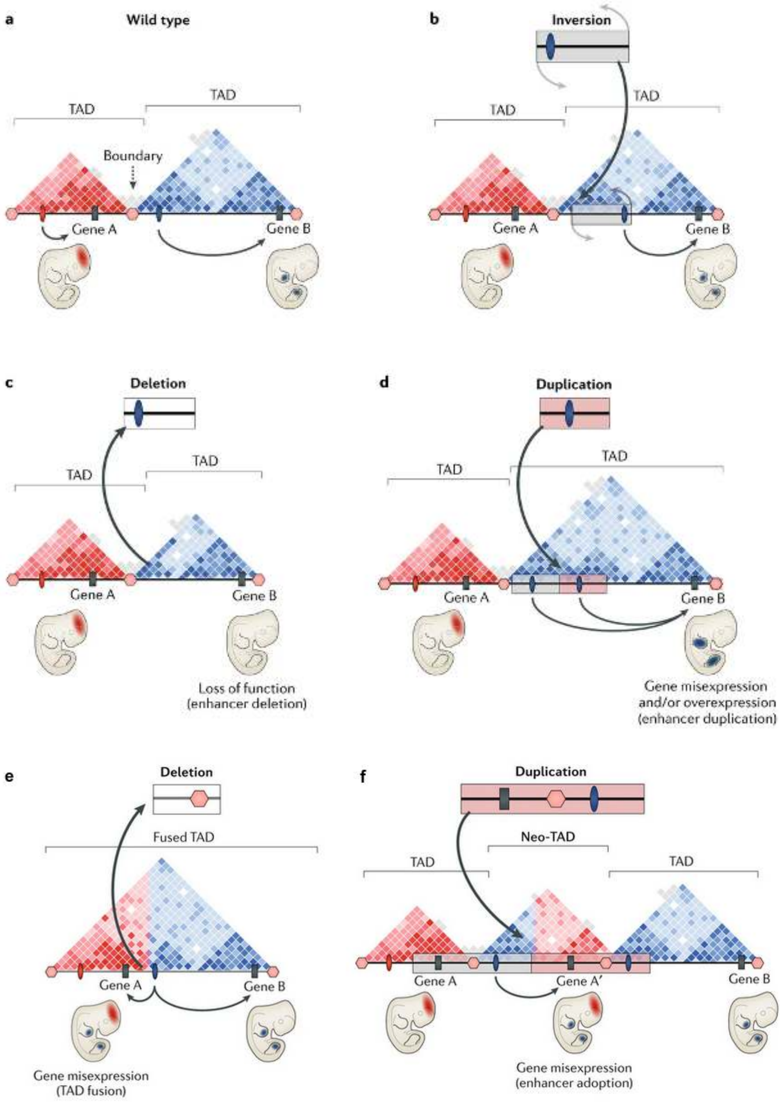


Figure 1.3: Schematics of how SVs can alter chromatin structure and cause gene misregulation of gene expression. Topologically associated domains (TADs) are shown as red and blue triangles. Different colors represent different chromatin domains. Gene A is expressed in the developing brain and gene B is expressed in the developing limbs. It is expected that the regulator of gene A in the red domain does not interact with gene B but only gene A. In panel (a) wild type is shown. In panel (b) the region that includes the regulatory element of gene B is inverted. The inversion occurs entirely within the blue TAD. Both the red and blue TADs are preserved. No functional change is indicated in development. This points out not every SV causes misregulation. In panel (c) regulatory element of gene B is deleted. Blue TAD is smaller due to deletion. Function loss is indicated due to gene B reduced or missing expression. In panel (d) the region with the regulatory element of gene B is duplicated causing overexpression of gene B. Blue TAD is larger due to duplication. Functional change is indicated. In panel (e) deletion in the TAD boundary region is shown. This deletion result in the fusion of red and blue TADs and interaction of the blue enhancer with both gene A and B. In panel (f) the part of the genome that includes gene A, TAD boundary and enhancer of gene B, is duplicated. This result in a neo-TAD formation and overexpression of the gene A. This figure is adopted [2].

1.2 3D chromosome conformation capture method (Hi-C)

In 2009 Lieberman-Aiden introduced a novel method named Hi-C. It stands for the chromosome conformation capture method combined with high-throughput sequencing [19]. It aims unbiased identification of chromatin interactions across the entire genome. Hi-C experiment starts with fixing two spatially adjacent chromatin segments that are bound to one another (cross-linked DNA) by using formaldehyde. Then DNA is fragmented with a restriction enzyme (Hind III) leaving DNA with 5′ overhang. While overhang is being filled, biotin is added as a marker. DNA fragments are ligated into one DNA sample. This sample contains a junction with a biotin marker between fragments that were spatially close to each other in the nucleus. Later on, this DNA sample is sheared and purified such that ligation products are preserved. Finally, biotin markers are pulled down and remain fragments are pair-end sequenced. In Figure 1.4, the schematic of the Hi-C experiment is shown.

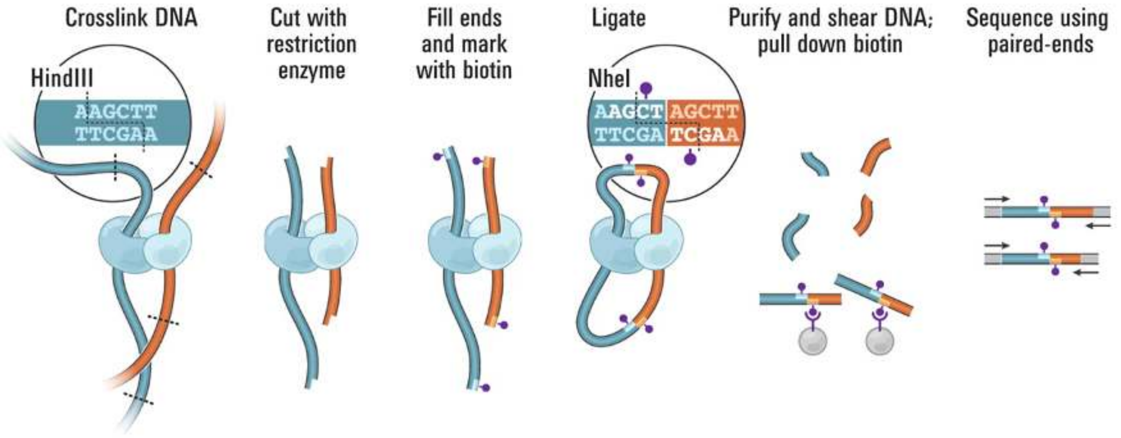

Crosslink DNA
Cut with restriction enzyme
Fill ends and mark with biotin
Ligate
Purify and shear DNA; pull down biotin
Sequence using paired-ends
HindIII
AAGCTT
TTCGAA
NheI
AAGCT AGCTT
TTCGA TCGAA

Figure 1.4: Schematic representation of Hi-C experiment. The main steps are shown. Spatially adjacent chromatin segments (blue/orange) are crosslinked by formaldehyde and cut by HindIII that results in DNA fragments with overhanging ends. A filling that includes a biotin marker is applied to overhanging ends. Later, fragments are ligated with intra-molecular favored ligation. The DNA sample is purified, sheared and biotin is pulled down. Finally, segments with previously biotin marked are pair-end sequenced. This figure is adopted [19].

Read pairs obtained from sequencing are mapped to a reference genome. Most of the reads corresponding to long-range contacts between segments of DNA that are far apart on the linear reference genome. Mapping to a reference genome is one-dimensional, but in fact, chromatin is organized in 3D space. The aim is to understand the 3D conformation of chromatin by showing long-range contacts for the whole genome. This property of Hi-C differs from other methods and makes Hi-C a promising method for detecting large structural variations.

Mapped read pairs are transformed into a hic file. This file contains the information on total contact frequencies between genomic regions. Total contact frequencies are obtained by counting the number of times distinct pairs of genomic regions that are spatially proximal. Each genomic region is obtained by partitioning the genome into equal-sized bins. Bin size, also known as resolution, is the number of sequenced nucleotides in the genomic region. For instance, 50 kb resolution (or bin size of 50 kb) stands for representation of 50,000 nucleotides contact frequencies by one bin.

A genome-wide matrix can be obtained from hic binary file at the pre-computed resolution. This matrix is a representation of the genome vs genome where each matrix cell represents a bin. The number of bins depends on the resolution chosen. To form a genome-wide matrix at 50 kb resolution, the matrix size should be 64,000 x 64,000 to represent around 3 billion nucleotides. This would be a very large matrix. Researchers either keep the resolution low (bin size large) or region of their interest in the genome small to visualize. The heatmap visualization of total contact frequencies between regions in the genome is called the Hi-C map. It is convenient to think of a Hi-C map as an image where a pixel is a bin. Pixels have color intensity whereas bins have total contact frequencies. In Figure 1.5, a genome-wide Hi-C map at 1 Mb resolution and chromosome 10 vs. chromosome 10 Hi-C map at 50 kb are shown. Figure 1.5, shows the inherent characteristics of Hi-C maps. Firstly, both the geneome-wide and the chromosome 10 vs 10 Hi-C maps, are symmetric matrices. Secondly, the intensity has not been preserved for the whole Hi-C map. It decays with respect to the linear genomic distance, so the highest signal comes from the diagonal where the genomic distance is zero. Thirdly, the signal intensity decay is not the same on every pixel that is in the same sub-diagonal of the matrix because the genome has a 3D structure and it has compartments and interaction domains in loops.

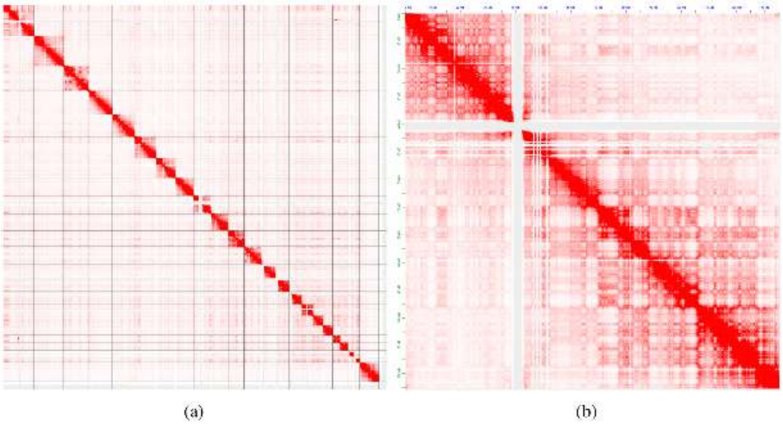


Figure 1.5: Hi-C map examples. Red color indicates a higher number of contact frequency. In panel (a), a genome-wide Hi-C map is shown. Diagonal boxes (submatrices) show chromosome's total contact frequencies with itself (cis) and non-diagonal submatrices show chromosome *i* vs chromosome *j* total contact frequencies (trans). This heatmap is obtained at 500 kb resolution. In panel (b), chromosome 10 vs 10 is shown. The diagonal has the highest contact frequencies. The centromere of chromosome 10 can be seen as a large gray cross. This region is not mappable and thus the observed contact frequency is zero.

Ideally, the entries of the Hi-C matrix of raw contact counts would be proportional to the true contact frequency. Unfortunately, this is not the case due to biases in the

Hi-C experiment. Restriction fragment lengths, mappability and restriction site density at regions are some of the reasons that influence the contact count. There can be more reads in some regions due to experimental biases thus higher contact counts will occur even if these regions do not frequently interact. To minimize the biases in the Hi-C matrix, some normalization methods are developed. Knight-Ruiz (KR) normalization is one of the most popular methods. KR is a matrix balancing algorithm to normalize a symmetric matrix [20]. It assumes the sum of rows and columns of the matrix are equal to one. Using KR normalization can cause difficulty in SV detection. SV patterns can get lost while normalization removes experimental biases by increasing or lowering the intensity of genomic loci under the assumption that every locus has equal visibility, especially in copy number variations (CNVs). In a case of duplication, where the read counts are doubled compared to other loci, the signal would be scaled down in order to achieve the same visibility with others. More information about KR normalization is provided in the discussion section 4.3.2.

1.3 Patterns of large SVs in Hi-C maps

SV fuses parts of the genome that were previously distal on the linear genome which causes a change in spatial positions. Formerly distal parts can now interact with each other. This means more reads span that region than before. More contact counts will occur in that region than before. As a result of that, besides the parts that are close to each other on the linear genome, high intensity signals are observed between fused parts on the Hi-C map. In this way, large SVs cause deviations from the reference in the Hi-C map. These deviations are not random but rather they create specific patterns in the Hi-C map. In Figure 1.6, sample genomic rearrangements and their patterns in Hi-C maps are shown as a cartoon.

1.4 Motivation and objective

Although large SVs induce specific patterns in Hi-C, detecting them manually is a difficult task. One reason is that Hi-C maps are large. If the researcher does not know the region of interest in the genome, then the whole genome matrix has to be inspected. Another reason in the difficulty of detecting SVs manually is that the inherent patterns of the Hi-C map. The challenge is to differentiate between the patterns that are consequences of SVs and the patterns that occur due to other reasons such as compartmentalization, TADs, experimental errors or mappability issues. Except for very large SVs, manual detection of SVs is depended on researcher's skill and experience since it is relatively easy to confuse an SV with expected high-intensity counts or noisy nature of Hi-C. Analyzing the Hi-C map manually is error-prone as well as it is time-consuming. In this regard, developing an automated large SV detection tool in Hi-C maps would be helpful for research and diagnostics. With this study, an automated tool for the detection of large SVs in Hi-C maps has been introduced.

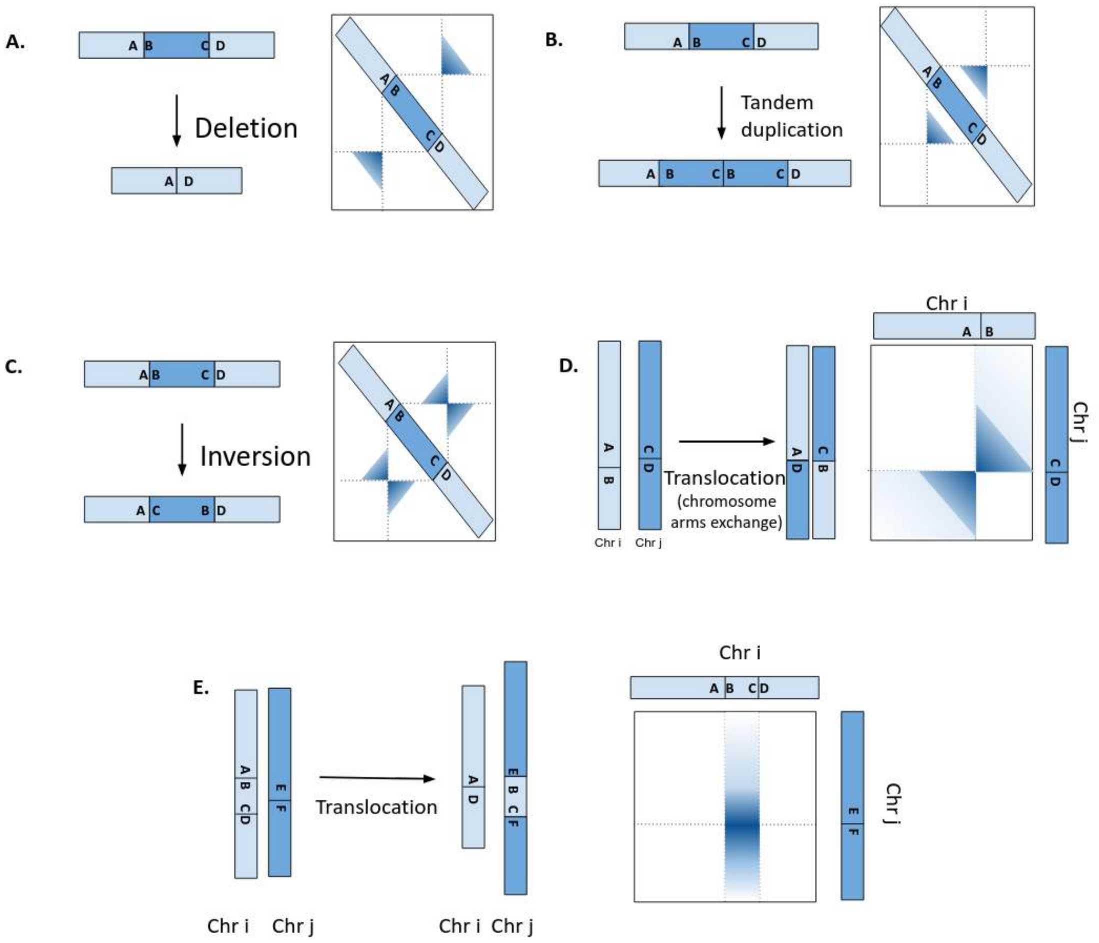


Figure 1.6: Schematic representation of SVs and SV induced patterns in Hi-C maps. In panel A, deletion of a genomic region and resulting pattern in a Hi-C map schematic is shown. In a reference genome, which has been used for mapping, the reads A and D are relatively far away from each other. If region BC is deleted in the sample genome, then the region A and D will be spatially closer. More paired-end reads will span the region A to D and more contact counts will occur. For this reason, a high signal intensity will occur at the position in the Hi-C map where region A and D meet. The highest intensity occurs directly at the breakpoints of the deletion. Contact counts will be more frequent not only in the breakpoints but also in the neighborhood. This creates a gradient, showing a "fading out" effect from a sharp corner. In panel B, a schematics of a tandem duplication is shown. In the Hi-C map, the effect of tandem duplication can be seen as a higher interaction frequency in BC region. In a reference genome previously distal B and C got closer after the tandem duplication due to the duplicated segment. For that reason, higher contact frequency occurs in the BC region. In panel C, an inversion can be seen. In reference genome B is proximate to A and C is proximal to D. Because region BC is inverted, now C is proximal to A and B is proximal to D. As a consequence, a corner of high intensity can be observed in the Hi-C map where A and C meet and the corresponding pattern in the region where B and D meet. In panel D, a reciprocal translocation is shown. The arm of chromosome *i* is exchanged with the arm of chromosome *j*. Region B of chromosome *i* is now attached to the end of region C in chromosome *j*. Region B is now proximal to region C and in return region D is proximal to region C. This results in higher intensity in Hi-C map between newly proximal regions. Since it is chromosome arm exchange (a very large piece of chromosome) the pattern observed in the Hi-C map is larger. In panel E, a piece of chromosome *i* is cut and pasted to the chromosome *j* between regions E and F. The pattern occurs in the Hi-C map can be seen as a stripe if the cut and pasted region is small. Previously, BC region was distal from chromosome *j*. After translocation, BC region is proximal to the EF region in the chromosome *j*. That proximity creates a high number of contact counts between BC and EF region. These contact counts are visualized as the higher intensity in the Hi-C map and created a stripe pattern. This pattern is common in chromothripsis and cancer where many genomic rearrangements occur.

Chapter 2

Methods

2.1 General Tool Outline

The main idea of our tool is resembling a researcher's approach in spotting an SV in the Hi-C map. A researcher would initially search for an unusually high-intensity region at low resolution in a genome-wide Hi-C map. In the case of an unusual finding, the next step would be a detailed look into the region at higher resolutions while aiming to detect SV patterns such as a sharp corner with a gradient of decreasing signal intensity. If possible, the candidate SV region would be compared to a reference sample to clarify the candidate region is not an inherent pattern but indeed it is a deviation that only occurs in the interested sample. The approach of our tool is similar to researcher's. It works on three different resolution stages. It starts searching for SVs at a lower resolution and continues step-wise at a higher resolution. Every resolution stage has similar working steps with minor differences. This section aims to give an overview of the general pipeline.

The first step of the pipeline is data preparation. The tool requires sample and reference Hi-C files as inputs. These files are the output of Juicer [21], a tool for the extraction of Hi-C files from bam files that contain read information. These two files go through the data preparation step where both files are extracted into matrices (matrix size depends on the resolution). Both sample and reference matrices are scaled according to their genome-wide total contact count to compensate for the difference in the sequencing depth. Then the reference scaled matrix is subtracted from the sample scaled matrix resulting in a difference matrix.

The second step of the pipeline is applying convolution with kernels on the difference matrix to obtain a feature set. A mathematical definition of convolution can be given as an integration that expresses the amount of overlap of one function g as another function f is shifting over the function g. In image processing and this study, function f is the input matrix and function g is the kernel matrix. The output matrix of convolution, i.e. the response, is flattened. Together with other responses from other kernels, they form the feature set. At every resolution stage, a specific feature set is used. The reason for this is, that the observed SV patterns may vary at different resolutions due to binning.

The third step of the pipeline is applying the pre-trained model for deciding if a bin represents a breakpoint of an SV or not. Each resolution stage has its training data. Training data is obtained by manual labeling on a Hi-C map of a chromothripsis sample with many SV breakpoints for every resolution stage. At 500 kb and 100 kb resolution, a logistic regression model is used for classification. At 25 kb resolution, a random forest is used as a classifier. The output of the classification model is a set of SV breakpoint candidates.

The final step of the pipeline is converting candidates into genomic coordinates in the Hi-C map and boxing. Candidates have their matrix indices. The final output of the tool is preferred to be in genomic coordinates. This is done by multiplying matrix indices with the bin size. At 500 kb and 100 kb resolution stages, a boxing algorithm is applied. Boxing can be considered as a preparation for the next resolution stage. Boxing means here aggregating the neighbor candidates by enclosing them in one rectangular region of interest. The boxing algorithm adds a margin to the candidates such that a certain neighborhood around the candidates can be investigated at the next resolution stage. In this way, the neighborhood information is not lost. This is important for pattern recognition. Furthermore, by aggregation, the number of candidates that has to be extracted in the upcoming resolution is diminished. At 25 kb resolution stage however boxing is not necessary since it will return the final output of the tool.

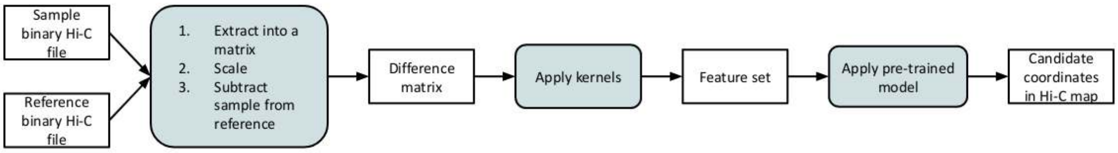


Figure 2.1: General workflow of the tool.

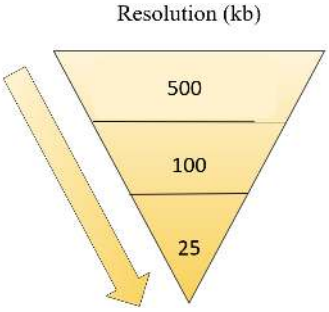


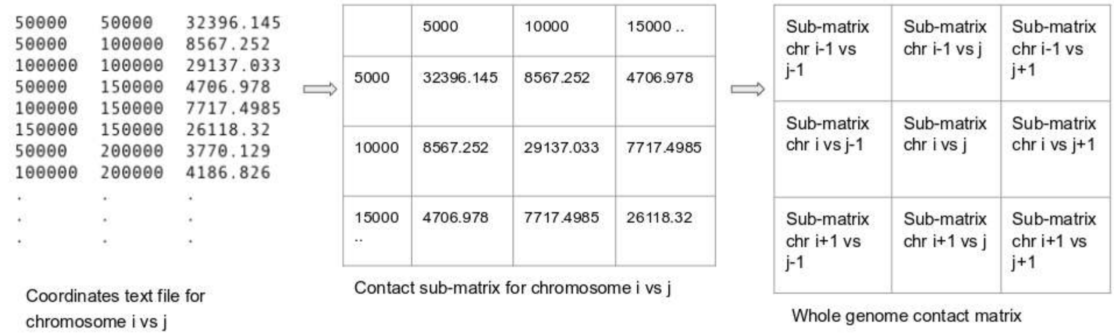
Figure 2.2: Resolution stages that are investigated by the tool. The tool starts working at low (500 kb) and then continues to a higher resolution. This step-wise design was necessary because it is infeasible to do a genome-wide scanning for SVs at 25 kb resolution. Storing, applying convolution, running models on a genome-wide matrix at high resolution would be a computationally expensive and time-consuming process.

2.2 Data Preparation

This section aims to describe the first step of the pipeline, data preparation. It is subdivided into three steps: extraction, scaling and subtracting. The input of data preparation is Hi-C files and it returns a difference matrix.

2.2.1 Data extraction

The first step of the data preparation is the extraction of the region of interest in sparse triplet format from binary Hi-C file through straw [22]. Sparse triplet format is a way to represent the sparse data with non zero values. This format has 3 columns: x,y and $value$ where x and y are for defining the genomic coordinates and $value$ represents the contact frequency. Sparse triplet format has turned into a complete matrix where the matrix indices are defined by x and y, and matrix cells are defined by the $values$. Cis matrices (chromosome i vs chromosome i) have been summed up with their transpose since only upper diagonal part of cis matrices are stored in sparse triplet format to save up space. It is alright to store the only upper diagonal part of cis matrices because cis matrices are symmetric. The matrix size is predetermined by dividing the total nucleotide number of each chromosome to the resolution stage. In the matrix, all NA counts are replaced with zero. The whole step is repeated for the reference region of interest by using the reference Hi-C file. At 500 kb resolution, all the cis matrices and trans matrices are collected into a complete genome-wide matrix. A genome-wide matrix is necessary for finding the total number of contacts for matrix scaling. Also, some of the features at 500 kb are only extractable on a genome-wide scale.



```
50000   50000   32396.145
50000   100000  8567.252
100000  100000  29137.033
50000   150000  4706.978
100000  150000  7717.4985
150000  150000  26118.32
50000   200000  3770.129
100000  200000  4186.826
.       .       .
.       .       .
.       .       .
```

Coordinates text file for chromosome i vs j

	5000	10000	15000 ..
5000	32396.145	8567.252	4706.978
10000	8567.252	29137.033	7717.4985
15000 ..	4706.978	7717.4985	26118.32

Contact sub-matrix for chromosome i vs j

Sub-matrix chr i-1 vs j-1	Sub-matrix chr i-1 vs j	Sub-matrix chr i-1 vs j+1
Sub-matrix chr i vs j-1	Sub-matrix chr i vs j	Sub-matrix chr i vs j+1
Sub-matrix chr i+1 vs j-1	Sub-matrix chr i+1 vs j	Sub-matrix chr i+1 vs j+1

Whole genome contact matrix

Figure 2.3: Schematic representation of matrix extraction steps. Left panel shows the sparse triplet format. Middle shows the matrix for chromosome i versus j created from coordinate format. Right panel is a representation of whole-genome matrix which contains every chromosome i vs j combination matrices where i, j are the chromosome names $(1, 2, ..., 22, X, Y)$.

Straw usage

Straw [21] is a pipeline for Hi-C data extraction that has been developed by Aiden lab and available on GitHub as a Python file. To use straw, the user needs to specify the

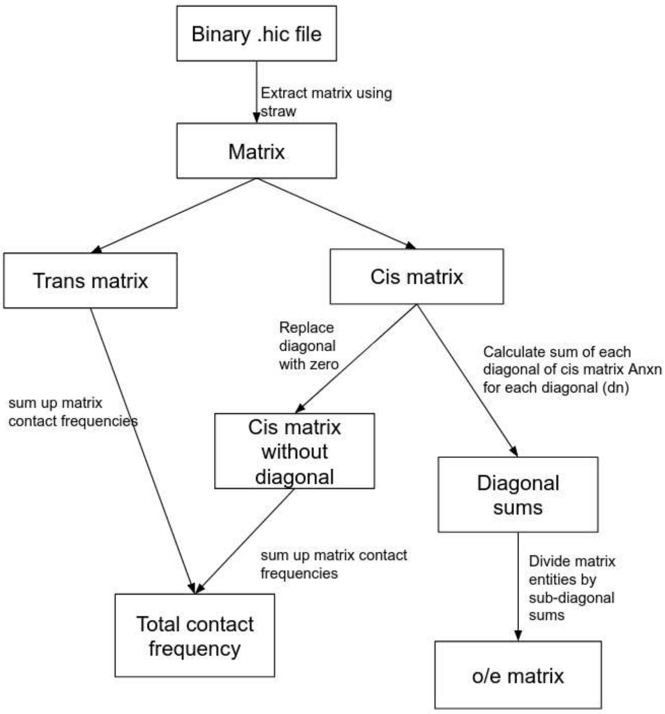
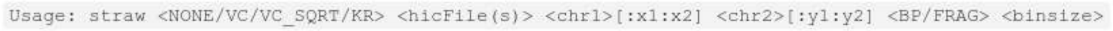

Binary .hic file
Extract matrix using straw
Matrix
Trans matrix
Cis matrix
Replace diagonal with zero
Calculate sum of each diagonal of cis matrix Anxn for each diagonal (dn)
sum up matrix contact frequencies
Cis matrix without diagonal
Diagonal sums
sum up matrix contact frequencies
Divide matrix entities by sub-diagonal sums
Total contact frequency
o/e matrix

Figure 2.4: Flowchart of preprocessing. Juicer format [22] Hi-C file for each chromosome pair is extracted into a matrix. If the chromosome pair is *i* vs *i* then it is a cis matrix, otherwise, chromosome pair *i* vs *j* is a trans matrix. Cis matrix is summed up with its transpose. The diagonal of the cis matrix replaced with zero. Sub-diagonal sums of cis matrix are calculated to find observed/expected ratio that will be used at 500 kb resolution stage.

Hi-C data normalization type (KR / NONE / VC / VC SQRT). Our tool can work with "NONE" and "KR" options. "NONE" option stands for raw Hi-C binary file without any normalization. "KR" option stands for Knight-Ruiz normalization [20]. The user also needs to specify the Hi-C binary file path, the genomic coordinates of the start and end of the interested region, base pair or fragment ("BP" / "FRAG"), and bin size. The tool works for "BP" option. The bin size is given to straw Python function according to the resolution stage; 500000, 100000 or 25000.



```
Usage: straw <NONE/VC/VC_SQRT/KR> <hicFile(s)> <chr1>[:x1:x2] <chr2>[:y1:y2] <BP/FRAG> <binsize>
```

Figure 2.5: Straw input format.

2.2.2 Matrix scaling

The second step of the data preparation is scaling the sample and reference matrices. Due to experimental reasons (sequencing depth, experimental variation, etc.), total contact counts of sample and reference Hi-C matrices are not necessarily the same. The sample region might have differ in its intensity from the reference simply because of the experiment and not because of an SV. To avoid or minimize these effects, matrices are scaled before subtraction. At low resolution total contact count of the sample genome-wide matrix is calculated by summing up all count values. Then each count is divided by the total contact count and to avoid from very small number each count is multiplied by 10^8. The same steps are applied to reference genome-wide matrix at low resolution.

2.2.3 Matrix subtraction

The final step of data preparation is a subtraction step. At the end of the scaling step, two matrices, one belongs to the sample and the other one belongs to the reference, are obtained. The reference matrix is subtracted from the sample matrix to get the difference matrix. The next steps of the tool are applied to the difference matrix. The reason for performing subtraction is to remove all inherent structures (TADs, compartments, etc.) and focus only on deviations from the reference. By setting a baseline (reference) for a sample map we aimed to minimize noise. This means our tool is sensitive to the selection of the reference map.

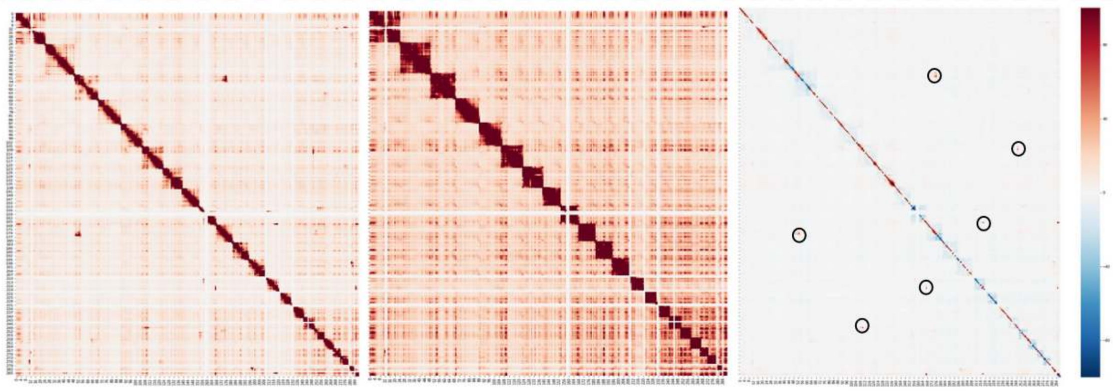


Figure 2.6: Sample, reference and difference Hi-C maps.

2.3 Deriving SV associated features

Structural variations induce specific patterns in Hi-C maps. We aimed to choose features that can recognize the patterns induced by SVs. Every resolution stage uses a different feature set because pattern visibility depends on the resolution. At low resolution, i.e. large bin sizes, a lot of counts are binned together and it is very difficult to identify the pattern of a corner unless the structural variation is extremely large. Thus, searching for high intensity bins rather than a corner pattern is cleverer. In contrast, at high resolution, the strong intensity deviation induced by an SV becomes less important and finding a corner or an edge becomes more crucial.

2.3.1 Patterns and kernels

Pattern recognition is a term for finding patterns in complex data and usually goes side by side with "machine learning". A very simplified definition of machine learning can be made as in the following: a model learns the patterns with labeled data and later recognizes the patterns in other data sets to predict the labels. Labels can be binary, such as a pattern exists or not "1/0", or it can differentiate between multiple patterns. Today, pattern recognition is applied in many areas for the investigation of images, speech, DNA sequences, protein structures, and websites. In this study pattern recognition is used for the detection of SV induced pattern detection. The essential part of this work is defining these SV induced patterns for the machine. This means selecting the features such that the models can learn the patterns. Features are obtained by using kernels.

In image processing, a kernel or convolution matrix is a small matrix. It is used for blurring, sharpening, embossing, edge detection of images. This is accomplished by performing a convolution between a kernel and an image. Convolution is the integral of the product of the two functions after one is reversed and shifted as stated in Equation 2.7. In other words, sliding one function while the other function is not moving (fixed) and summing up the area under the curve where both functions meet. In this study, the sliding function is the kernel and the fixed function is the Hi-C map or parts of it. The output of the convolution is the response to the kernel and used here as a feature matrix.

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{2.1}$$

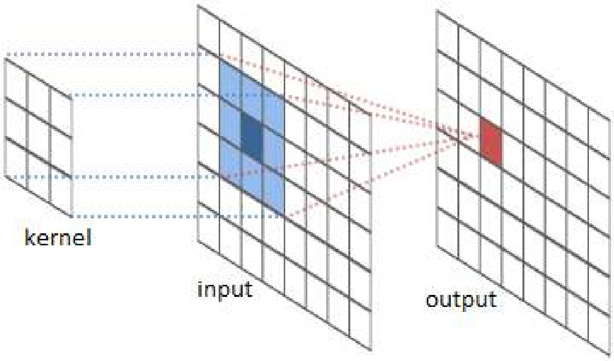


Figure 2.7: Matrix convolution representation.

2.3.2 At 500 kb resolution

At 500 kb resolution stage, we divided sub-matrices of the Hi-C map into cis and trans matrices. A cis matrix is a chromosome *i* vs *i* matrix while a trans matrix is a chromosome *i* vs *j* matrix. The reason for this separate processing is that cis matrices include the Hi-C matrix diagonals and TADs. Within these regions, high interaction frequencies are expected since TADs are self interacting chromatin domains. Because of that, these regions have higher intensity in overall compared to regions in trans matrices. In a cis matrix, the expected contact frequency of a cell increases when the distance to diagonal gets smaller. Since genome-wide Hi-C map is a symmetric matrix and thus the diagonal represents the interaction of nucleotides with themselves. In a trans matrix, the expected contact frequency is fixed to a small number for the whole matrix since the expected contact frequency is calculated by dividing the matrix-wide total contact frequency in the trans matrix to the number of bins. Usually, for trans matrices, the matrix-wide total contact frequency is very small. This is the reason for fixing the expected contact frequency to a small number. For the detection of SVs in the cis matrices, we used empirical thresholds and cells that satisfy all thresholds are called candidates. There are three criteria for cis matrix filtering. The first one is the pixel intensity. Pixel intensity stands for total contact frequency in a bin. The aim by setting this criterion is to eliminate low contact frequency regions on a difference map that are not possible to be an SV. However, filtering by only intensity is not enough because for cis matrices high intensity is expected. For that reason, we added the second and third criteria which are based on observed/expected (oe) contact frequencies. O/e allows us to include information on the distance from the diagonal. The expected contact frequencies of a pixel calculated by dividing the pixel to its sub-diagonal sum. O/e is calculated by dividing observed intensity to expected intensity separately for sample $(o/e)_s$ and reference matrix $(o/e)_r$. Then, the second criteria is defined as $(o/e)_s - (o/e)_r$ and the third criteria defined as $(o/e)_s/(o/e)_r$. The diagonal of a cis map is excluded in all thresholds since the at the diagonal intensities are very high and it creates problem in the calculation of expected values. For trans matrices, we applied a logistic regression with a feature set that depends on the intensity. The feature set contains the pixel intensity, the highest and the second-highest intensities among neighbor pixels. Neighbor pixels stand for the surrounding 8 adherent pixels.

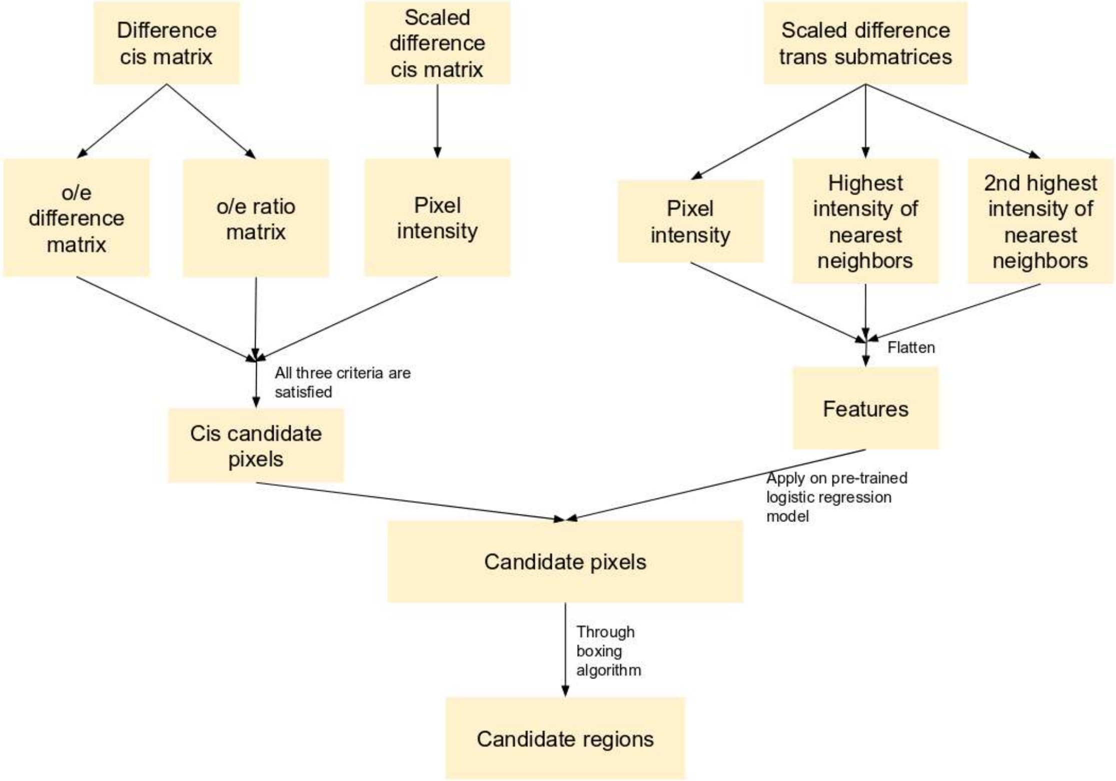

Difference cis matrix
Scaled difference cis matrix
Scaled difference trans submatrices
o/e difference matrix
o/e ratio matrix
Pixel intensity
Pixel intensity
Highest intensity of nearest neighbors
2nd highest intensity of nearest neighbors
Flatten
All three criteria are satisfied
Features
Cis candidate pixels
Apply on pre-trained logistic regression model
Candidate pixels
Through boxing algorithm
Candidate regions

Figure 2.8: Derivation of features at 500 kb resolution is shown. Difference matrices obtained previously in the data processing step by subtracting the reference matrices from interested sample matrices. Algorithm treats cis (chromosome *i* vs *i*) and trans (chromosome *i* vs *j*) matrices differently at 500 kb resolution stage for SV detection. The left side of the workflow is for cis matrices at 500 kb. The first criterion for cis matrices is the observed/expected (o/e) difference values. Expected values are calculated by summing up pixel intensities that are on the same sub-diagonal and then dividing each pixel to the sub-diagonal sum. For observed values are the pixel intensities are used. O/e values are obtained by dividing the observed values to expected values. O/e values are calculated the same way for the sample and the reference matrix. In the end, for each pixel (or cell in the matrix) we have obtained two o/e values, one comes from the interested sample matrix and the other one comes from the reference matrix. O/e difference values are obtained by subtracting the sample o/e from the reference o/e. The second criterion is obtained by getting the ratio of the o/e values of each pixel. The algorithm searched for a large quantity difference and a large proportion between the sample and the reference. The third SV candidate selection criterion is to check pixel intensities in the scaled difference cis matrices. If a pixel on cis matrix at 500 kb satisfies all three criteria, then the pixel is classified as candidate pixel and continues to boxing part of the algorithm. The right side of the workflow is for trans matrices. For trans matrices, at 500 kb resolution, features are solely on intensity. The algorithm derives 3 features for each pixel in a trans matrix that will be provided to the pre-trained classification model and the output pixels of the model continues for boxing step for the next resolution stage. The three features for the model are pixel intensity, highest intensity neighbor of the pixel, and the second highest intensity neighbor of the pixel. To obtain highest and the second highest intensity neighbors of the pixel, the surrounding 8-neighbor pixels of each pixel of a trans matrix are checked. If the pixel is in the corner or on the edge of the matrix, the padding applied with zero values.

2.3.3 At 100 kb resolution

At 100 kb resolution stage, we used a feature set that consists of pixel intensity, response to smoothing kernel and response to edge detection (Canny). Smoothing kernel is a 3x3 matrix of ones. Edge detection is a frequently used method in image recognition. It has two kernels; vertical derivation kernel and horizontal derivation kernel. Firstly vertical and horizontal kernels are convolved over the region of interest, separately. Then, the Euclidean distance matrix is calculated based on the cells from vertical derivation kernel response and horizontal derivation response $d = \sqrt{d_x^2 + d_y^2}$. Both smoothing kernel and edge detection kernels are convolved over the region of interest. The responses and pixel intensity matrix are flattened and in the end, each pixel defined with three features.

2.3.4 At 25 kb resolution

At 25 kb resolution stage, pixel intensity, response to smoothing kernel, the absolute value of response to horizontal derivation kernel, the absolute value of response to vertical derivation kernel and response to edge detection are used to obtain the feature set. Horizontal derivation kernel is applied to detect the horizontal lines in the difference matrix and for each pixel, the absolute value of the response is used due to the design of the kernel. Because the upper half of the kernel has positive values and the down half has negative values. The same logic is valid for the vertical derivation kernel. At 25 kb resolution, the Hi-C map is sparser and thus appears noisier. Deciding if a genomic rearrangement pattern is present or not based on a sparse matrix is more challenging. That is why 5 different features are used to represent each cell. Also, the information that can be gathered from neighboring pixels becomes more important. For that reason, kernel size has been increased to 5x5 for smoothing, horizontal derivation and vertical derivation.

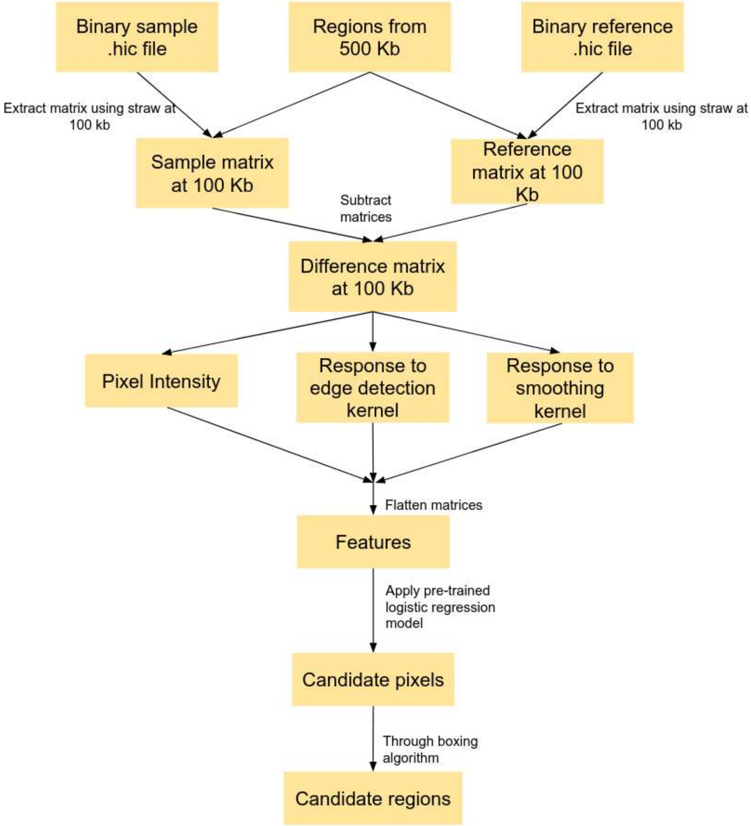

Binary sample .hic file
Regions from 500 Kb
Binary reference .hic file
Extract matrix using straw at 100 kb
Extract matrix using straw at 100 kb
Sample matrix at 100 Kb
Reference matrix at 100 Kb
Subtract matrices
Difference matrix at 100 Kb
Pixel Intensity
Response to edge detection kernel
Response to smoothing kernel
Flatten matrices
Features
Apply pre-trained logistic regression model
Candidate pixels
Through boxing algorithm
Candidate regions

Figure 2.9: Derivation of features at 100 kb resolution is shown. The boxing step at 500 kb of the algorithm returns the genomic coordinates of the candidate regions. The candidate regions at 100 kb resolution are extracted with the help of straw [22] by using these genomic coordinates provided from the previous resolution stage. Both sample and reference regions are extracted and from them, the difference matrix is calculated (The description of extraction and difference matrix calculation is given in the data processing step.) At 100 kb, three features are used: pixel intensity, response to edge detection kernel and response to smoothing kernel. The response to kernel means the output matrix of the convolution of kernel matrix on the difference matrix. At the next step the features are obtained after flattening and given to the pre-trained logistic regression model. The candidates resulted from the model are proceeded to boxing step for the 25 kb resolution stage.

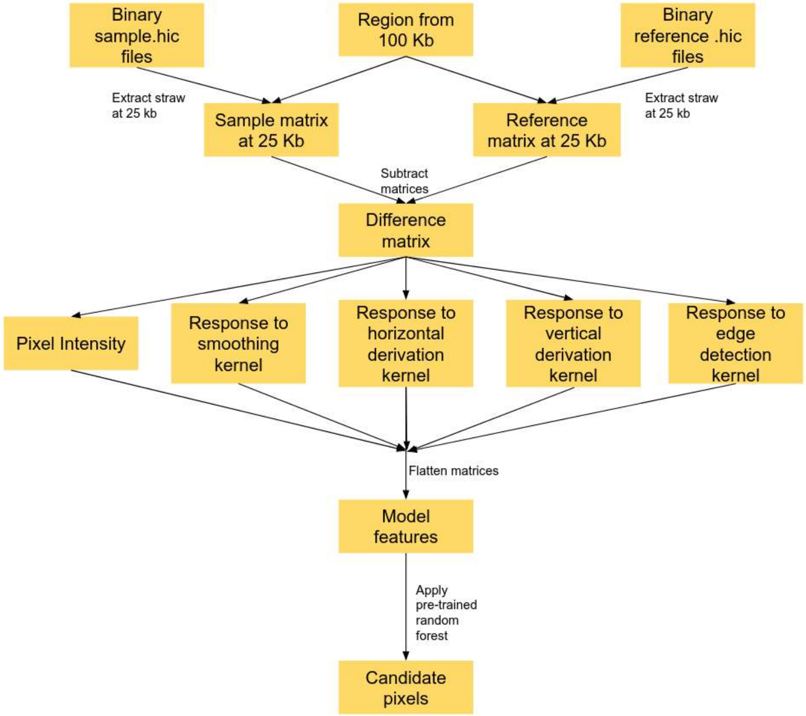
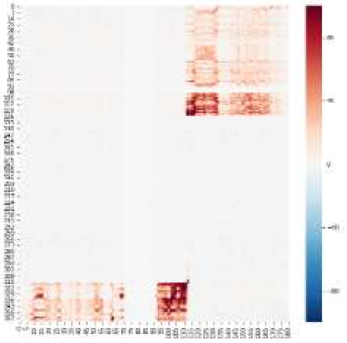

Binary sample.hic files
Region from 100 Kb
Binary reference .hic files
Extract straw at 25 kb
Sample matrix at 25 Kb
Reference matrix at 25 Kb
Extract straw at 25 kb
Subtract matrices
Difference matrix
Pixel Intensity
Response to smoothing kernel
Response to horizontal derivation kernel
Response to vertical derivation kernel
Response to edge detection kernel
Flatten matrices
Model features
Apply pre-trained random forest
Candidate pixels

Figure 2.10: Derivation of features at 25 kb resolution is shown. The genomic coordinates obtained from 100 kb resolution are used for region extraction at 25 kb by using straw [22]. The difference matrix for each region is calculated the same way with the other resolutions. The differences between 100 kb and 25 kb resolution are the usage of two more kernels and larger kernel sizes. Features are given to a pre-trained random forest model with 100 trees and the output of the model is SV candidate pixels at 25 kb. The candidate coordinates are transformed from matrix coordinates to the genomic coordinates and saved into a text file.

<table>
<tr><th colspan="2">Summary of features</th></tr>
<tr><td>Original
Chromosome 5 vs 16</td><td></td></tr>
<tr><td colspan="2">500 kb resolution</td></tr>
<tr><td colspan="2">Criteria</td></tr>
<tr><td>Cis</td><td>Trans</td></tr>
<tr><td>

- Pixel intensity
- o/e difference matrix $(o/e)_{sample} - (o/e)_{reference}$
- o/e ratio matrix $(o/e)_{sample} / (o/e)_{reference}$

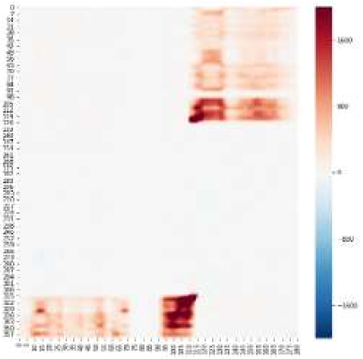
</td><td>

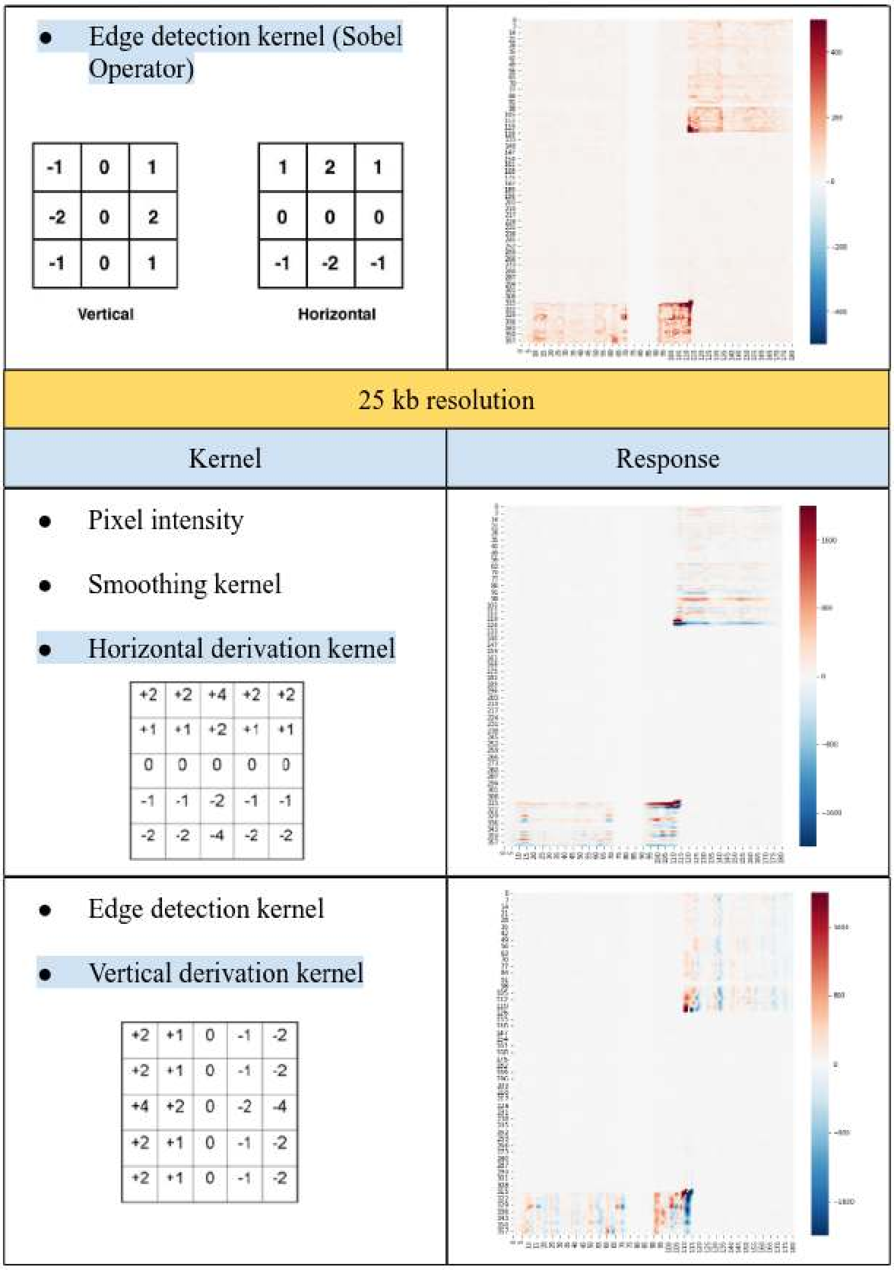
- Pixel intensity
- Highest intensity pixel of nearest neighbor
- 2nd highest intensity pixel of nearest neighbor

</td></tr>
<tr><td colspan="2">100 kb resolution</td></tr>
<tr><td>Kernel</td><td>Response</td></tr>
<tr><td>

- Pixel intensity
- Smoothing kernel

<table>
<tr><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>

</td><td>
</td></tr>
</table>



- Edge detection kernel (Sobel Operator)

-1	0	1
-2	0	2
-1	0	1

Vertical

1	2	1
0	0	0
-1	-2	-1

Horizontal

25 kb resolution

Kernel | Response

- Pixel intensity
- Smoothing kernel
- Horizontal derivation kernel

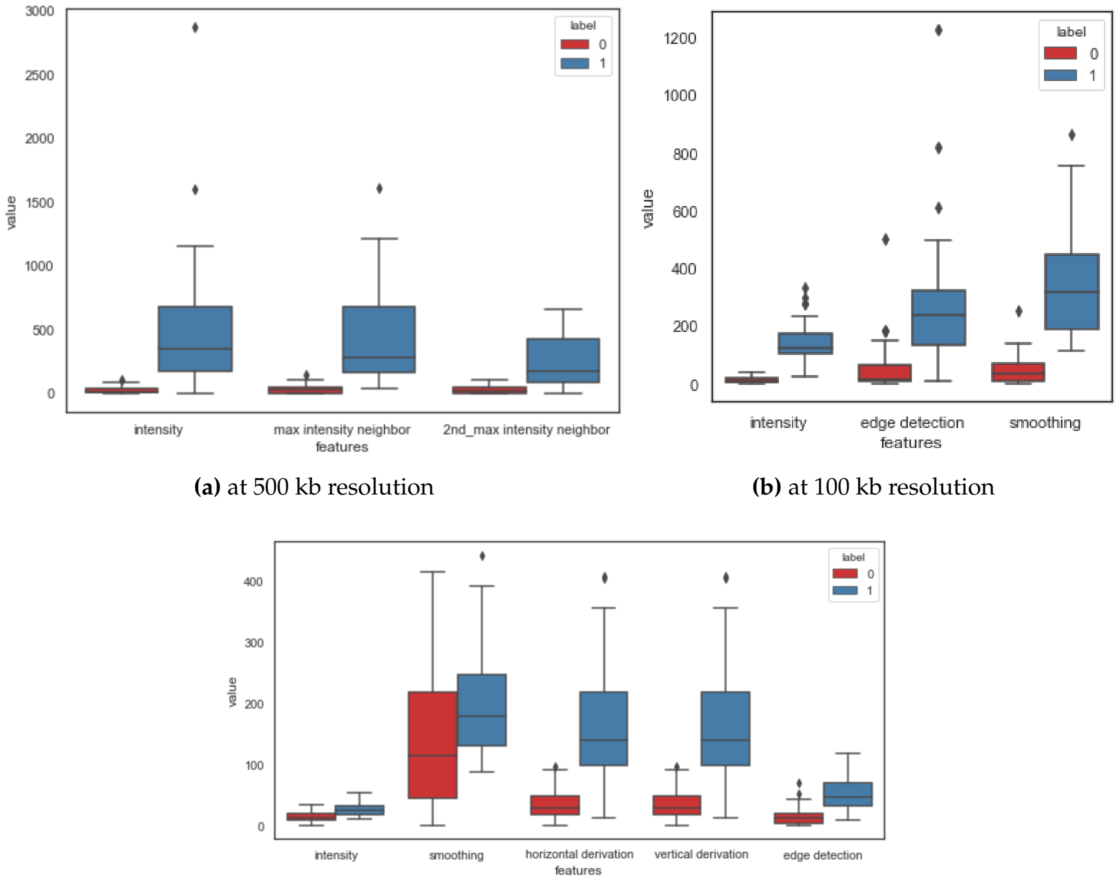
+2	+2	+4	+2	+2
+1	+1	+2	+1	+1
0	0	0	0	0
-1	-1	-2	-1	-1
-2	-2	-4	-2	-2

- Edge detection kernel
- Vertical derivation kernel

+2	+1	0	-1	-2
+2	+1	0	-1	-2
+4	+2	0	-2	-4
+2	+1	0	-1	-2
+2	+1	0	-1	-2

Figure 2.11: The summary of features used at different resolutions are shown. The responses to kernels (Hi-C maps), highlighted on the left side, are shown on the right side. Note that, at 500 kb different features are used for cis and trans maps. In the difference Hi-C maps, red color indicates high positive intensity values while blue is negative intensity values. Chromosome 5 vs 16 trans map from K562 cancer cell line is used as an example to illustrate the effect of each kernel on Hi-C map.

2.4 Labeling training sets



(a) at 500 kb resolution

(b) at 100 kb resolution

(c) at 25 kb resolution

Figure 2.12: The relationship between feature values and labeling classes is shown. Red is for non-SV breakpoints and blue is for SV breakpoints labeled in the training set at each resolution.

Different resolution stages require different classification models and thus different training sets. The difficulty in our tool design was to train the models. Since there is no ground truth about SVs in Hi-C maps that can be used as training sets, positive and negative examples for training sets have been manually labeled at 500 kb, 100 kb, and 25 kb resolutions. This means the classification models depends on how well we were able to define the training sets. This issue is pointed out more detailed in the discussion part 4.3.1.

We decided to label all examples of positive and negative pixels in one Hi-C map. A chromothripsis case was chosen for labeling due to the high number of visible SVs. Labeling is done in a visualization tool Juicebox that was developed by the Aiden lab [23]. At 500 kb resolution, 48 pixels from trans matrices were labeled of which 20 are negative samples (non-SV) and 28 are positive samples (SV). At 100 kb resolution, 54 pixels from both cis and trans matrices are labeled of which 24 are negative samples and 30 are positive samples. At 25 kb resolution, 76 pixels are labeled of which 41 are negative samples and 35 are positive samples. Box plots are shown in Figure 2.12 indicating how well features separate the training sets at different resolutions.

2.5 Selection of the classification models

An important step in designing our tool was the selection of the classification models. Since the training sets are manually labeled, we did not have a large amount of data. So, it would not make sense to use a complex machine learning algorithms such as neural networks. Instead, we tried to keep it simple. Linear regression, logistic regression, and random forest have been tried out.

2.5.1 Linear regression

Linear regression is based on a linear function, $y = \alpha + \beta x$ where y is the response, x is feature vector, α is intercept and β is the slope, to map input variables to continuous response variables. The goal is to find the best-fit parameters, α and β, for the data points. Once fitted, a linear regression model can be used to classify the label of the new input based on fitted linear model. Usually, best-fit parameters are found with using the least square approach. A line that minimizes the sum of squared error $\widehat{\varepsilon}_i$ (the differences between actual and predicted values of the dependent variable y), each of which is given by, for any candidate parameter values α and β. The function tried to be optimized is shown in the equation 2.3. The output of linear regression is a continuous unrestricted value.

$$\min_{\alpha,\beta} \mathbf{Q}(\alpha,\beta),\ for\mathbf{Q}(\alpha,\beta) = \Sigma_{i=1}^{n}\widehat{\varepsilon}_i^2 = \Sigma_{i=1}^{n}(y_i - \alpha - \beta x_i)^2 \tag{2.2}$$

2.5.2 Logistic regression

Logistic regression is a type of linear regression. Logistic regression uses a logistic function to map the input variables to categorical response variables. Logistic regression outputs a probability between 0 and 1. The difference between simple linear regression and logistic regression is logistic regression estimates the probability of a binary outcome while linear regression predicts the outcome itself. In essence, logistic regression transform the response variable of linear regression to a response value between 0 and 1. The standard logistic function that is used for transforming linear regression response to logistic regression response[24].

Given that y is the linear regression response variable, P is the logistic regression response. y can be any value while P is between 0 and 1. y is depended on x so does P.

$$P = \frac{1}{1 + e^{-y}} \tag{2.3}$$

Since $y = \alpha + \beta x$, the logistic regression function is written as:

$$P(x) = \frac{1}{1 + e^{-(\alpha + \beta x)}} \tag{2.4}$$

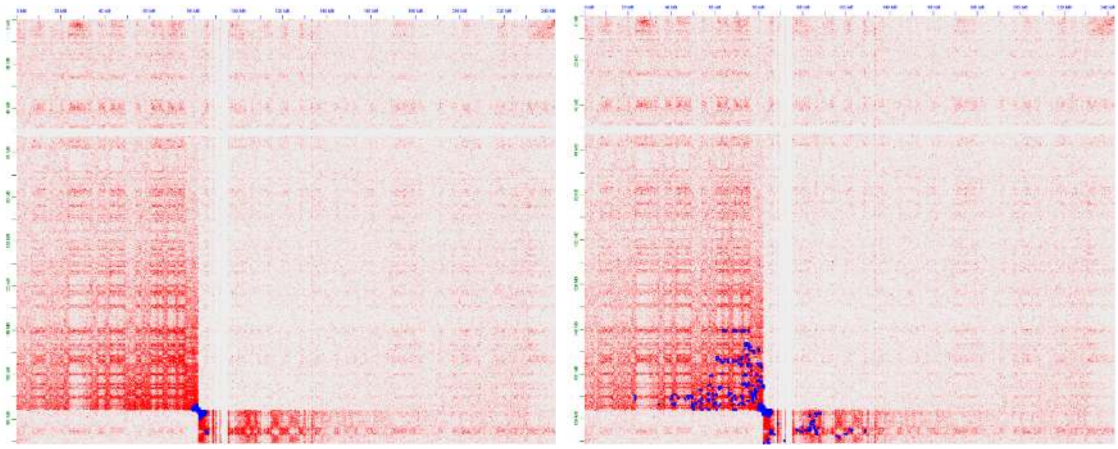
And inverse of logistic function is written as in the following:

$$\ln \frac{P(x)}{1 - P(x)} = \alpha + \beta x \tag{2.5}$$

2.5.3 Random forest

Random forest is an ensemble learning method for classification and regression that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees [25]. Decision trees try to answer sequential "yes/no" questions that send us down a certain route of the tree. Usually, a greedy algorithm is used to find the optimal choice at each node. However, a small number of trees may result in overfitting the data. That is why many decision trees are aggregated in a random forest. Each tree is build up by random sampling from the data. In this way, trees are different from each other. A random forest can handle unbalanced training data since it chooses different samples from data. In contrast to logistic regression, where input variables preferred to be not correlated, a random forest can learn the correlation between input variables (features).

All three classification methods have been tested at all resolutions. It is observed that at 500 kb and 100 kb resolution, logistic regression returns better results, in sense of precision, than other methods (shown in Figure 2.13). However, at 25 kb resolution, the random forest returns better results than logistic regression. The success of random forest at 25 kb resolution might be because of the correlation between features. A scatter plot has been provided in the supplements showing the relation between features at 25 kb resolution 5.3. Also, the number of features is higher at 25 kb stage. Logistic regression might have difficulty in optimizing higher number parameters given such a low number of training pixels. The comparison of the classification models has only done by visualizing the candidates that are returned by the models. For instance, a confusion matrix for each model was not possible for this study since there were no ground truths about SV breakpoints in Hi-C maps.



(a) Logistic regression **(b)** Random forest

Figure 2.13: SV breakpoint detection, shown in blue, for the same Hi-C map region at 500 kb resolution is shown in panel (a) and (b). Logistic regression returns fewer candidates that are more precisely located at the breakpoint of the SV. Random forest returns more candidates at more dispersed positions.

2.6 Boxing

Boxing is the final step of 500 kb and 100 kb resolution stages. In this step, the aim is to define rectangular regions of interest around candidate pixels that are close enough to each other. The first and the most important reason to do boxing is to preserve information that comes around the candidate pixel. SV breakpoints might be one pixel in the Hi-C map but SV induced patterns occur with the contribution of the neighboring pixels. Our tool based on these patterns. That is why neighbor pixels are important.

Secondly, since the tool filters candidates from low resolution to high resolution, it is better to start with a large number of candidates and filter them out in later stages. So, in other words, it is only possible to detect SVs in the genomic regions that have been boxed at the previous resolution stage. It is not possible to recover information at a higher resolution than has been lost previously at a lower resolution. Keeping this in mind, having a large number of candidates means having a large number of regions to extract at the next higher resolution stage. Extracting regions is the computational bottleneck of the algorithm. It is more time-efficient to extract a low number of medium-size regions. That is the second reason why a box has been defined that includes the candidates which are close in 2D coordinates.

Boxing step, start with placing a fixed size box around every candidate. The box size can be user-defined by changing the distance in the boxing function. The default distance parameter is 5. This means the fixed box size will be 11x11 pixels and the candidate will be in the center of the box. After defining a fixed size box around each candidate, we merged boxes if they satisfy the distance criteria. Assuming box A has a width w_A and length l_A and box B has a width w_B and length l_B, distance criteria can be described as:

1. Horizontal distance $d_x < (\frac{1}{w_A + w_B})/2$
2. Vertical distance $d_y < (\frac{1}{l_A + l_B})/2$
3. max (x coordinates of A, x coordinates of B) $< x - coordinate$ limit of the region
4. max (y coordinates of A, y coordinates of B) $< y - coordinate$ limit of the region
5. min (x coordinates of A, x coordinates of B) > 0
6. min (y coordinates of A, y coordinates of B) > 0

First two criteria checks for distance between two boxes. The last two criteria checks if the possibly merged box will be in the limits of region.

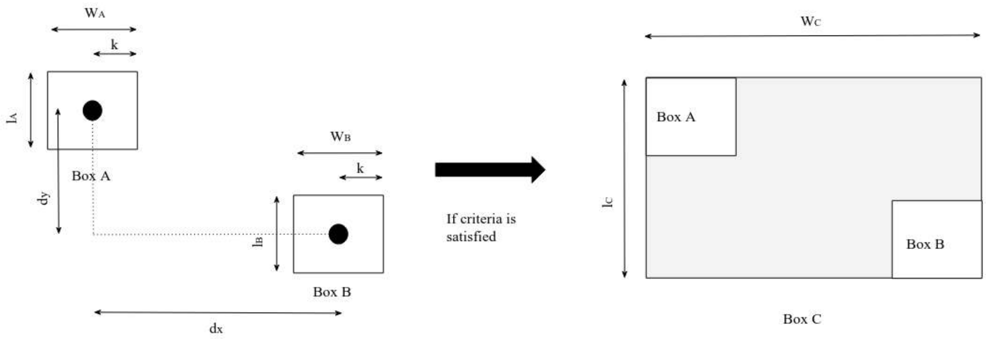


Figure 2.14: Schematics of boxing.

Chapter 3

Results

3.1 Tool usage

The tool is written in python 2.7. The tool aims to find large structural variations based on a Hi-C map in an automated way. So far, the tool is on the prototype stage. It consists of five python files, three CSV files, and one shell script. CSV files contain trained data for different resolutions. One python file contains straw data extraction function, another file contains initial preprocessing and the rest contains the SV breakpoint detection algorithm for each of three resolution stages. The tool requires two Hi-C maps; one is the map of interest (sample) and the second one is the reference map. Preferably, these two maps should come from the same cell type since reference will be used as a baseline to detect deviations in the sample. If all files are available, the user just needs to run the shell script with four input files given in the specific order. These inputs are the map of interest name, user choice reference map name, path to the sample .hic file and path to the reference .hic file. Running the shell script with these inputs will be sufficient to successfully get SV candidates for the Hi-C map of interest. The output of the tool will be six text annotation files in Juicebox format. For each resolution, it will produce one candidates and one regions annotation files. A candidate annotation file consists of two-dimensional genomic coordinates of the breakpoint candidates while a region annotation file contains the genomic coordinates of the regions that surround the breakpoint candidates. The run time of the tool depends on the given Hi-C map sequencing depth. It is recommended to run it on a cluster since parallel processing has been done for some part of the algorithm. The tool returns the outputs within 30 minutes when 32 cores are used in a dual processor EPYC computing machine.

3.2 Cis map

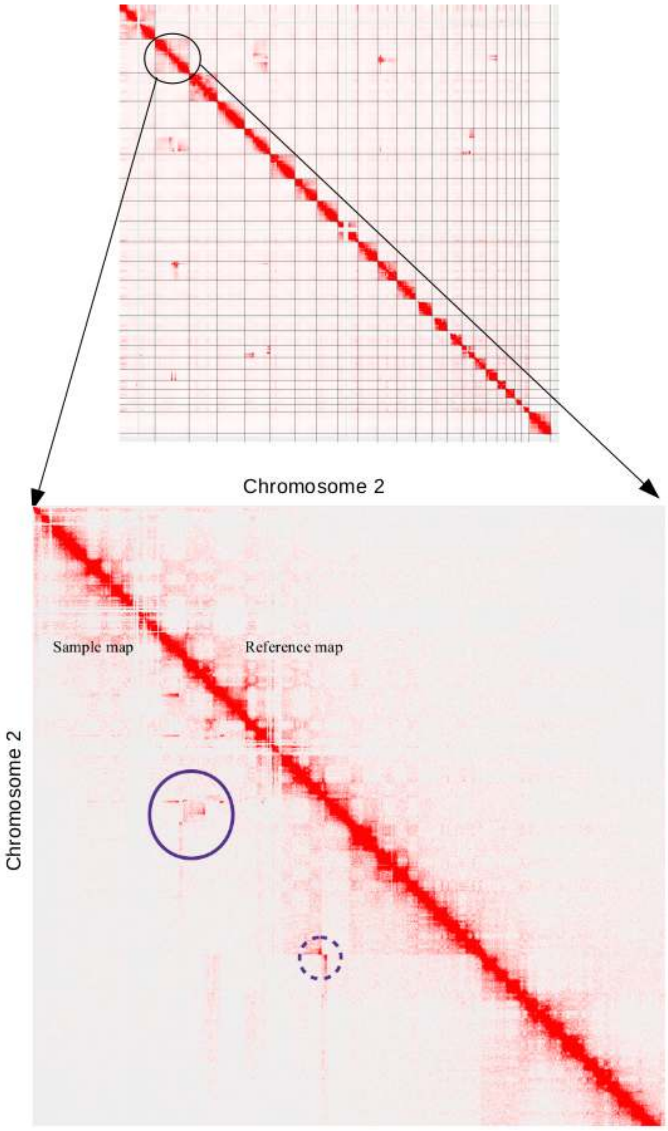
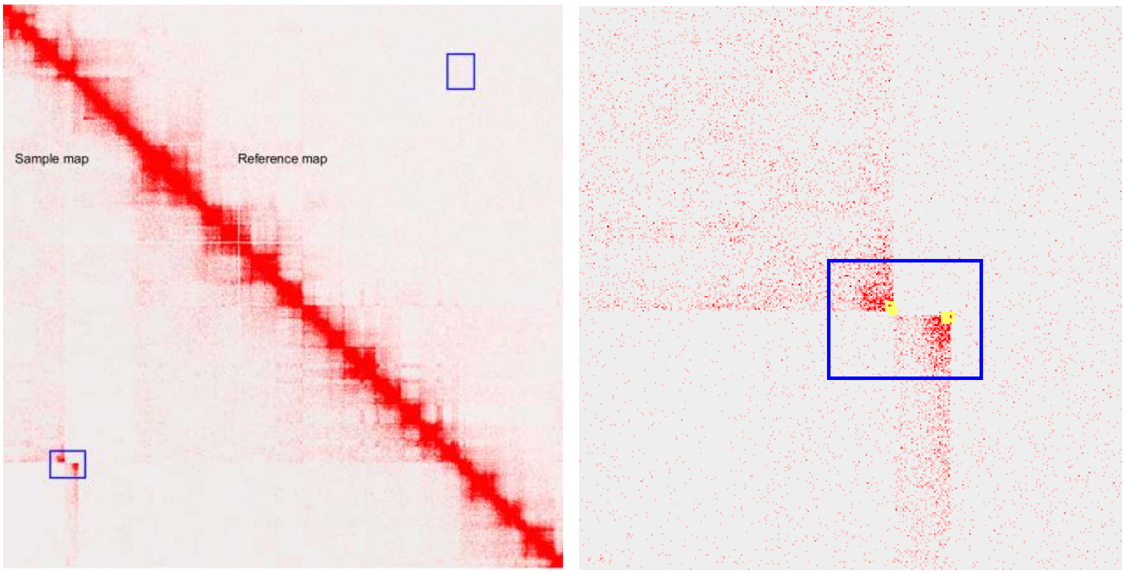


Figure 3.1: Part of the Hi-C map of chromosome 2 vs 2 of the chromothripsis case is shown. The upper triangle belongs to the reference map and the lower triangle belongs to the sample map. Red indicates a higher interaction frequency. Many unusual high-intensity patterns are visible in the lower triangle whereas in the upper triangle these patterns do not exist. The reason for that, parts of chromosome 2 that were formerly distal to each other are now proximal. This figure shows a cis map of a genome with multiple rearrangements occur. Two smaller regions, marked with dashed and solid line circles, that contain many SV breakpoints are analyzed in detail in the following Figures 3.2 and 3.3. Hi-C map is visualized with Juicebox[23].



(a) At 100 kb

(b) At 25 kb

Figure 3.2: Both panels (a) and (b) show a zoomed-in of the dashed circled region from Figure 3.1. Blue boxes are the regions detected at 100 kb resolution and yellow boxes are detected at 25 kb resolution by the algorithm. In panel (a) Hi-C heatmap upper triangle belongs to the reference and the lower triangle belongs to the sample map. The blue box shows two SVs; tandem duplication and deletion. At 100 kb resolution, the boxing algorithm gathers the SV breakpoints together and reports a bigger box because the breakpoints are close to each other. At 25 resolution both SVs are detected separately. At higher resolution, in panel (b), the heatmap shows more noise due to smaller bin size and relatively low contact counts. Also, it can be seen in panel (a) that due to the SVs, TAD formation occurs.

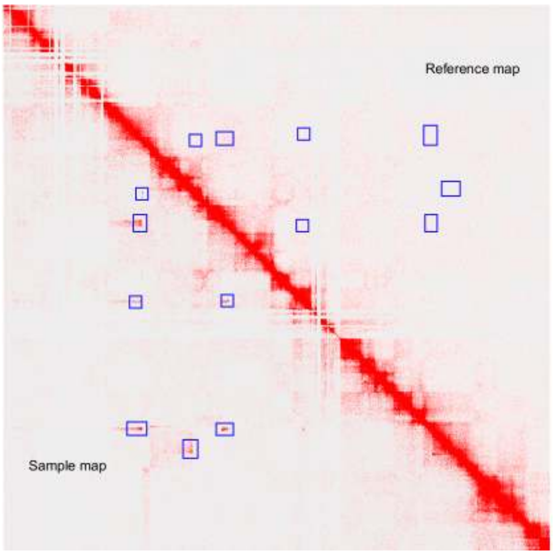


Figure 3.3: Detection results at 100 kb resolution in the region circled with a solid line in cis Hi-C map for chromothripsis. The upper triangle belongs to a reference map and the lower triangle belongs to the sample map. Blue boxes are regions detected at 100 kb resolution step with our tool. Most of the SVs shown in the figure have a stripe pattern indicating that some genome parts cut and pasted in a different area in chromosome 2. This is not surprising to see since the case in the figure from a patient with chromothripsis.

3.3 Trans map

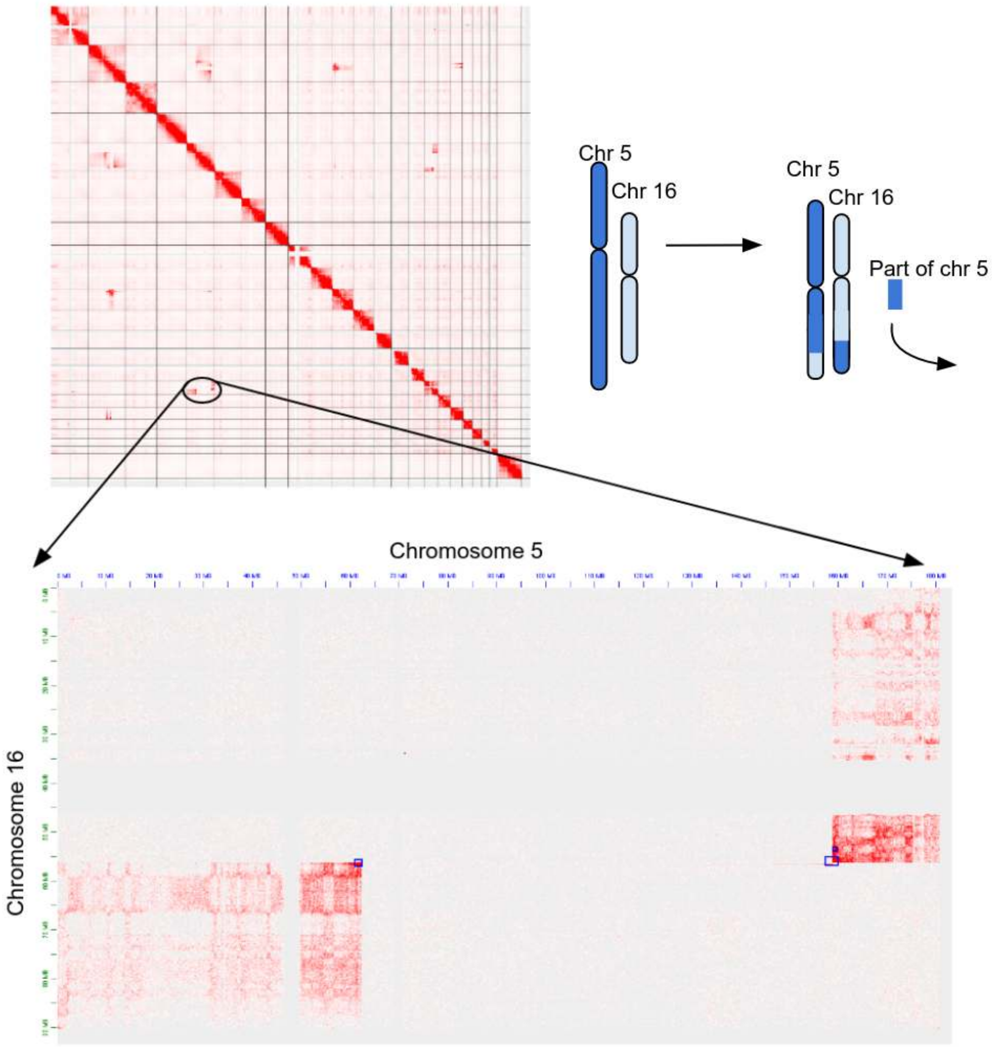


Figure 3.4: Hi-C map of chromosome 5 vs 16 in chromothripsis case is shown. Blue boxes show the region detected by the algorithm at 100 kb resolution. In this example, three translocations occur. Two big translocations are apparent even at this low resolution but the third translocation is more difficult to spot. At the margins of the heatmap, the coordinates of chromosome 5 (in blue) and chromosome 16 (in green) are indicated. The two big translocations suggesting that the bottom part of chromosome 16 is now proximal to the beginning of chromosome 5 (around 60Mb) and the bottom part of chromosome 5 is now proximal to the middle of the chromosome 16 (around 50Mb). The gap between translocations in chromosome 5 indicates that some part from the middle of chromosome 5 has been cut and gone somewhere else. More detailed versions of translocations can be seen in Figures 3.5 and 3.6.

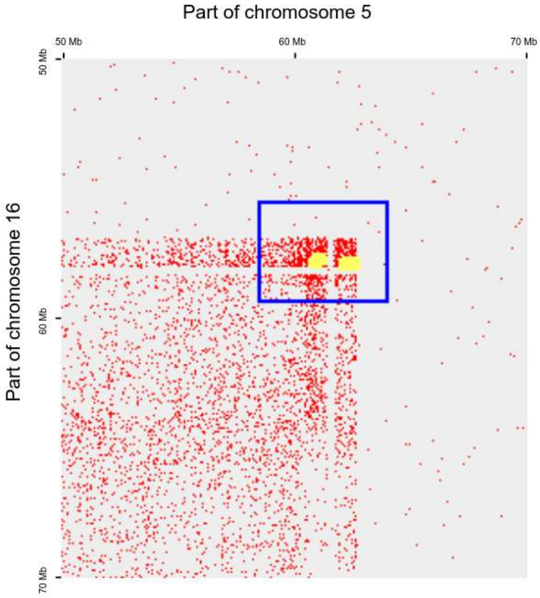


Figure 3.5: A magnified version of the left translocation in Figure 3.4 is shown here. Here, blue boxes indicate the detection results at 100 kb resolution and yellow boxes indicate the detection results at 35 kb resolution by the algorithm. The algorithm detected 2 yellow boxes which show that the tool does not provide enough precision yet. Secondly, the algorithm did not detect the exact corner that is visible in the heatmap. The probable reason for this is that these corners did not have a high intensity at 25 kb resolution.

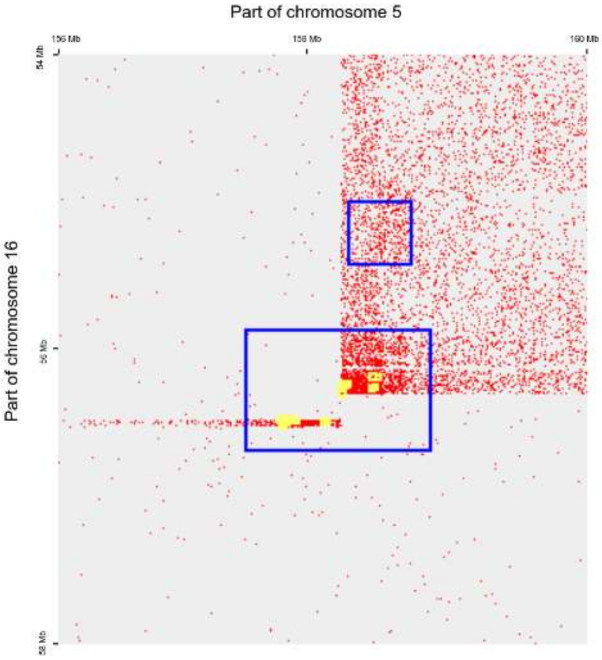


Figure 3.6: A magnified version of the right translocation in Figure 3.4 is shown. Here, blue boxes indicate 100 kb resolution detection and yellow boxes indicate 25 kb resolution detection results by the algorithm. A detailed look reveals two translocations in this area. At 100 kb resolution, the algorithm reported 2 blue boxes for this region and one of the boxes seems to be a false positive. During the later analysis step at 25 kb resolution, the algorithm did not find any further candidates within this falsely positive but found candidates in the other box. Some of the SV breakpoint candidates at 25 kb resolution are not precise enough. These candidates are within a region that is associated with an SV, but they are not localized at the putative breakpoint.

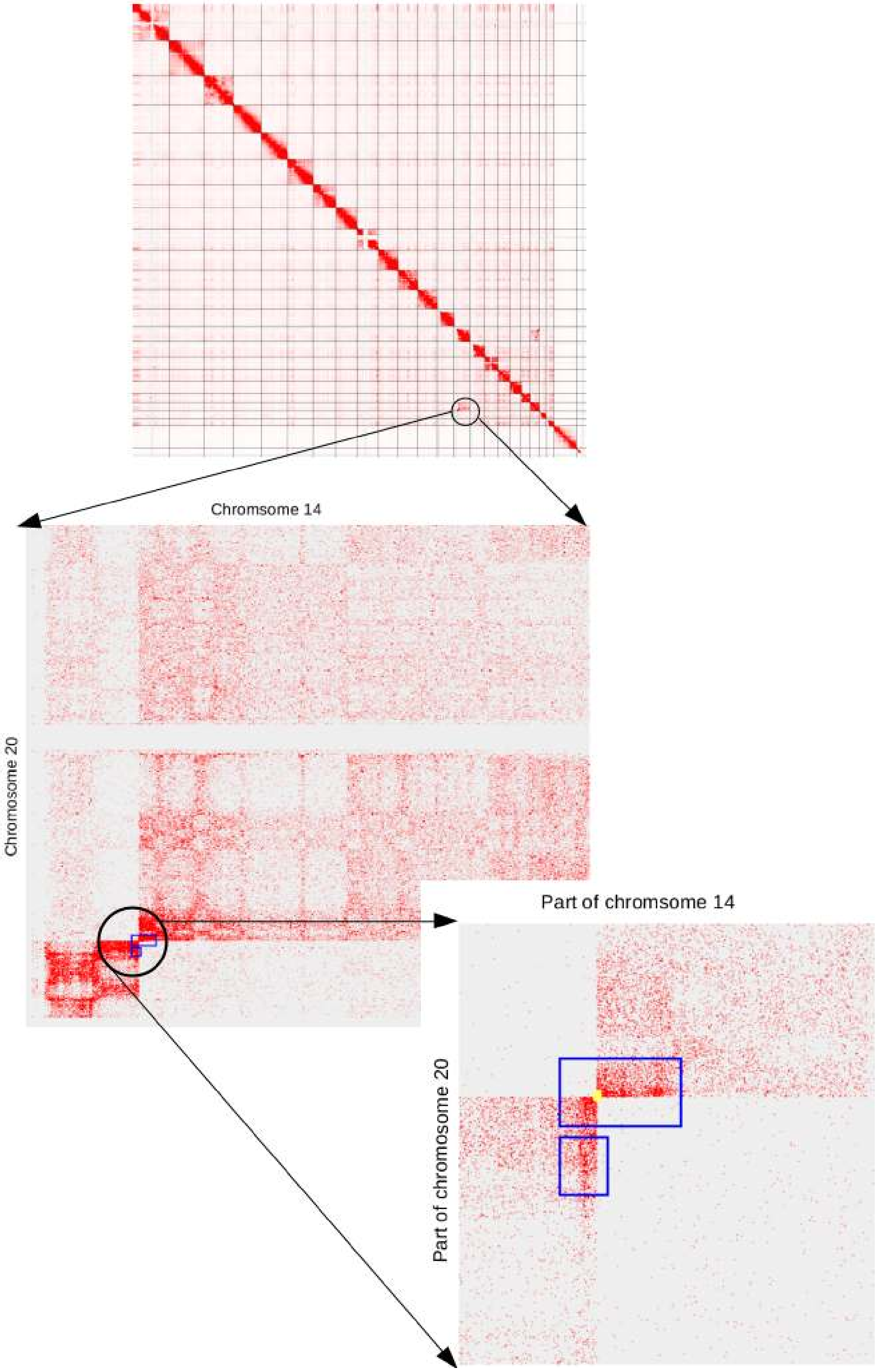
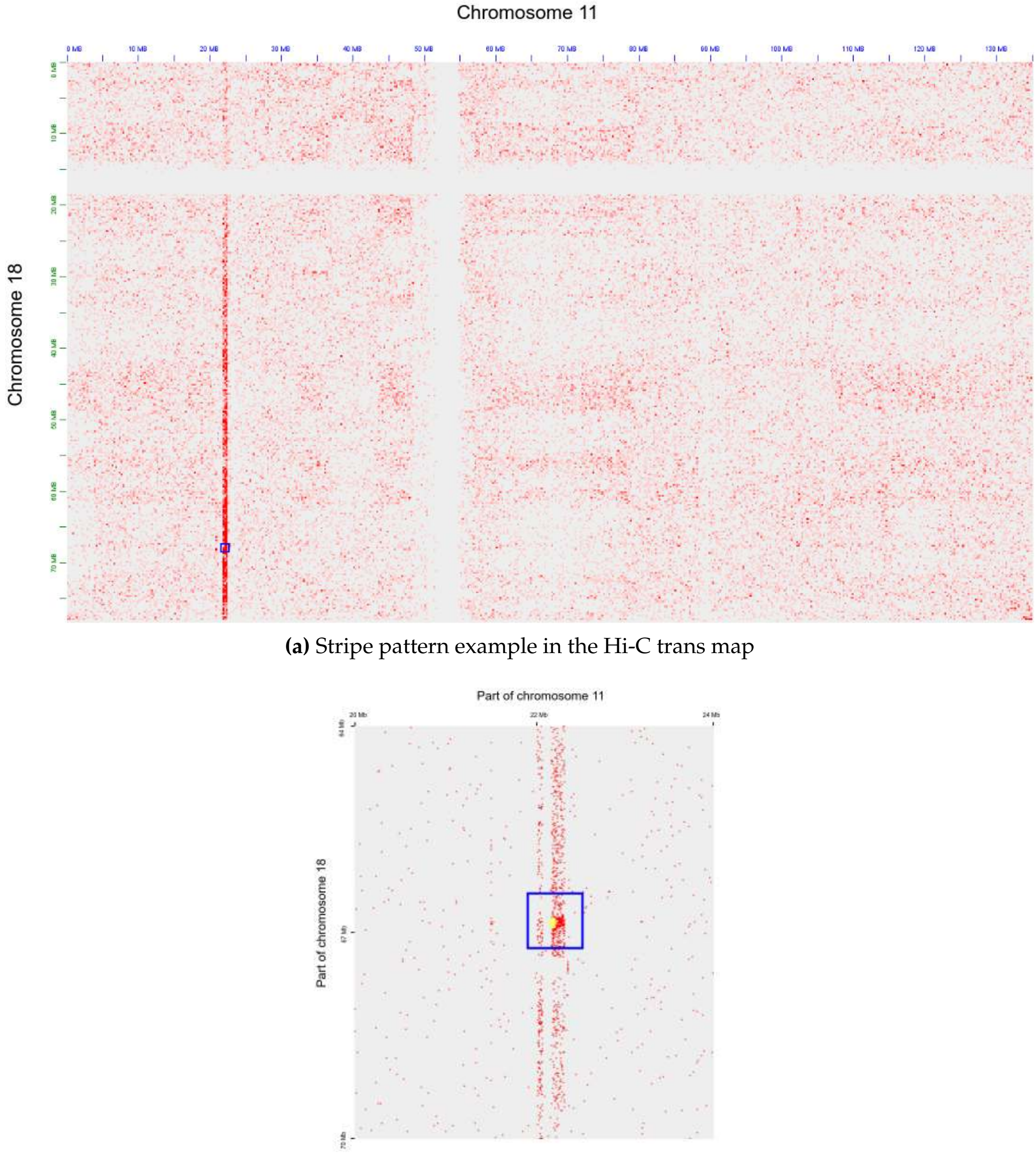


Figure 3.7: Part of a translocation is shown. Here, the end part of chromosome 20 is exchanged with the large end part of chromosome 14. This forms two fusion chromosomes: the end of chromosome 20 fused with the beginning of chromosome 14 and the large beginning of chromosome 14 fused with the large end part of chromosome 20. The color code of the boxes is same as in Figure 3.4.



(a) Stripe pattern example in the Hi-C trans map

(b) Enlargement of stripe pattern in the Hi-C map

Figure 3.8: In panel (a) the Hi-C map between chromosomes 11 and 18, an example of the stripe pattern is shown. This small part from chromosome 11 creates now an interaction pattern with whole chromosome 18. In part a, the location of the maximum gradient seems like at the end of chromosome 18. Because of this, it appears as if a small part of chromosome 11 is pasted at the very end of chromosome 18 at low resolution. But this might not be the case. A detailed analysis at high resolution, shown in panel (b), indicates that part of chromosome 11 is pasted into chromosome 18.

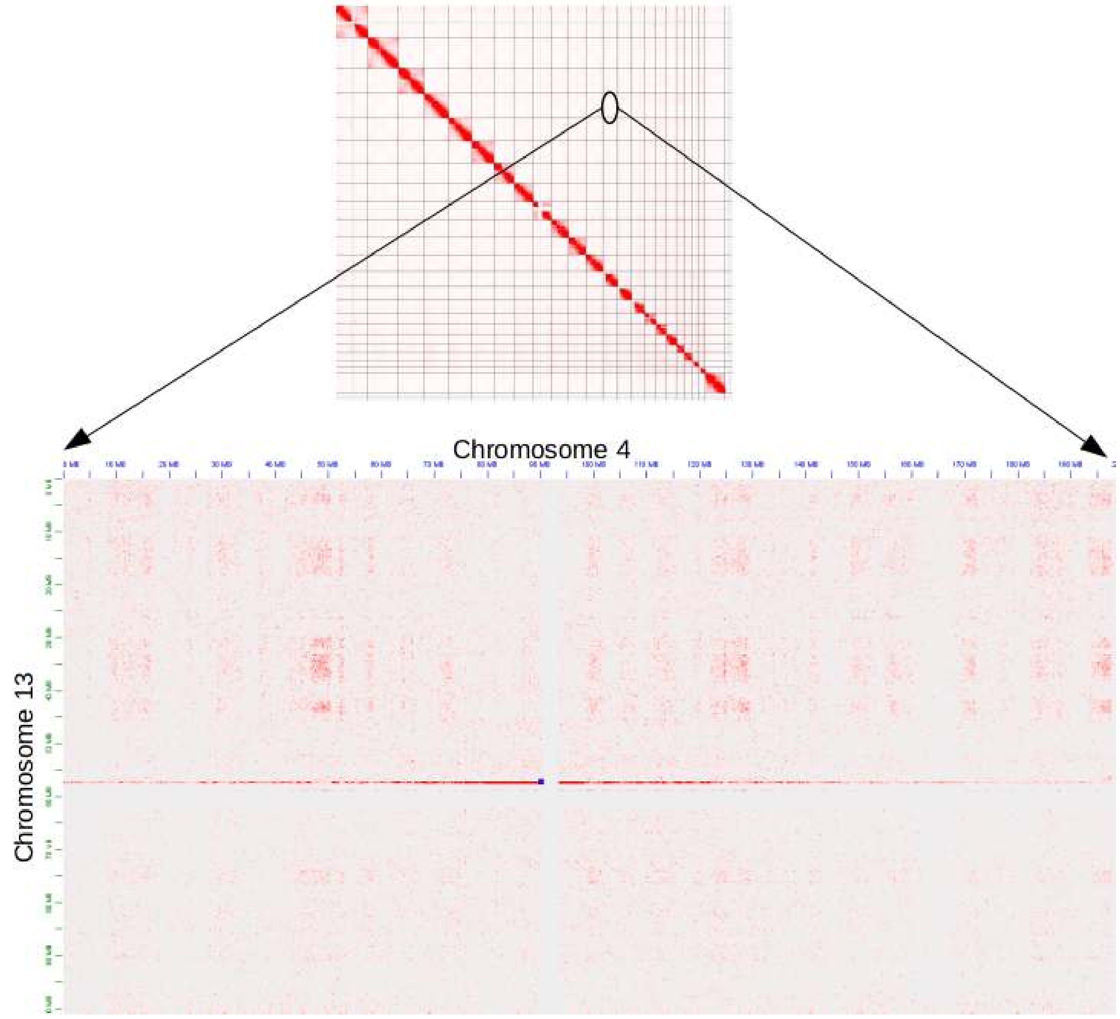


Figure 3.9: Part of the Hi-C map is shown with a stripe close to the centromere. This stripe can be a result of cut/copy and paste SV or it can be an experimental artifact. The detected region is on the edge of the centromere of a chromosome. The highly repetitive nature of the centromere causes difficulty in mapping and as a consequence, a very low number of contact frequencies are observed in the centromere region in the Hi-C maps. At the genomic part where the region is mappable and neighbor to the centromere will have a high gradient change. Even though these “neighbor to the centromere” regions do not have high signal intensity, they might be picked up by the algorithm just because their neighbors have very low signal intensity due to mappability. That is why it is difficult to judge if the call from the tool is right or wrong. This example is given to point out there are some cases where it is difficult to do a manual assessment.

3.4 Comparison of Raw vs KR normalized map

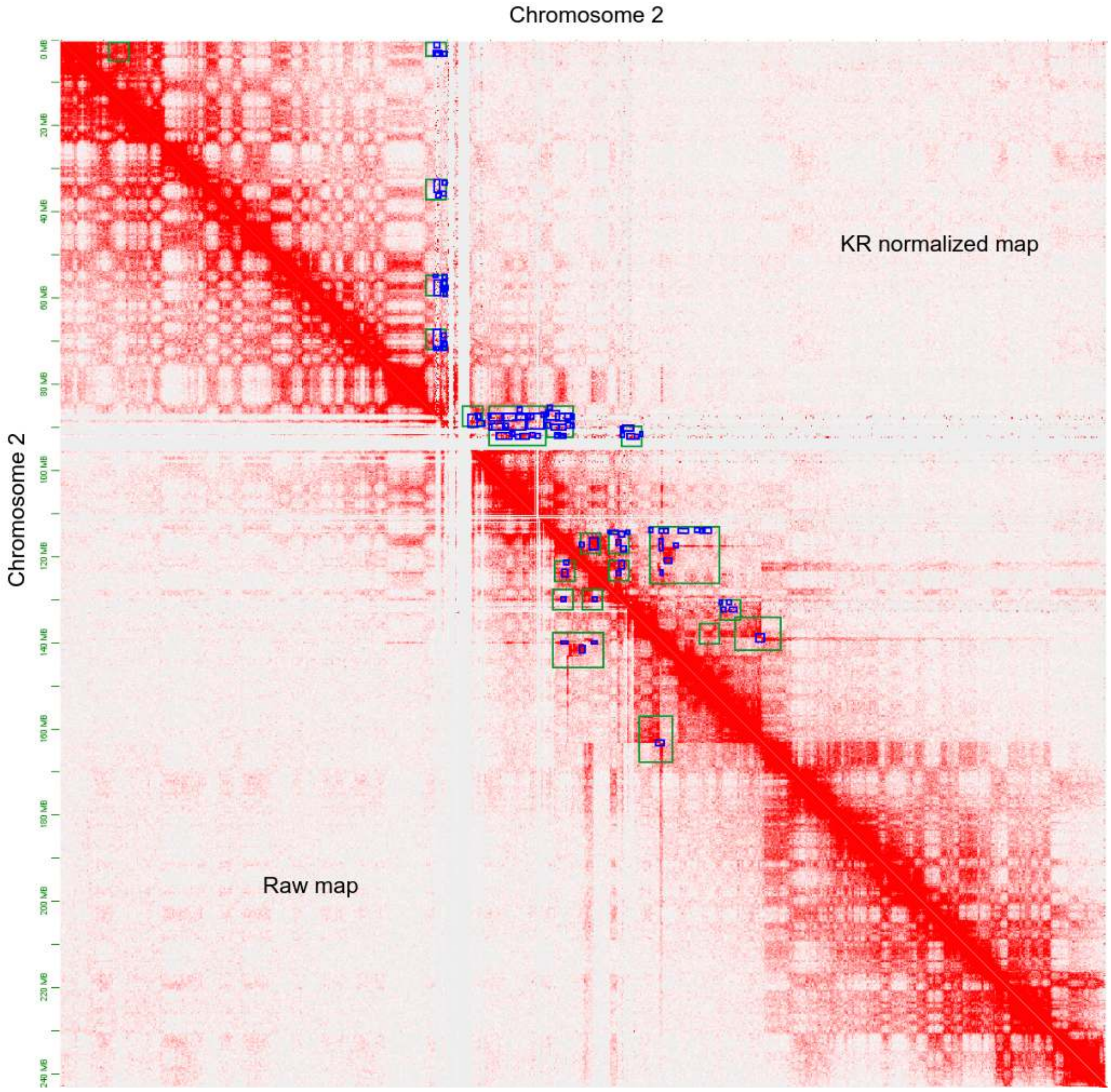
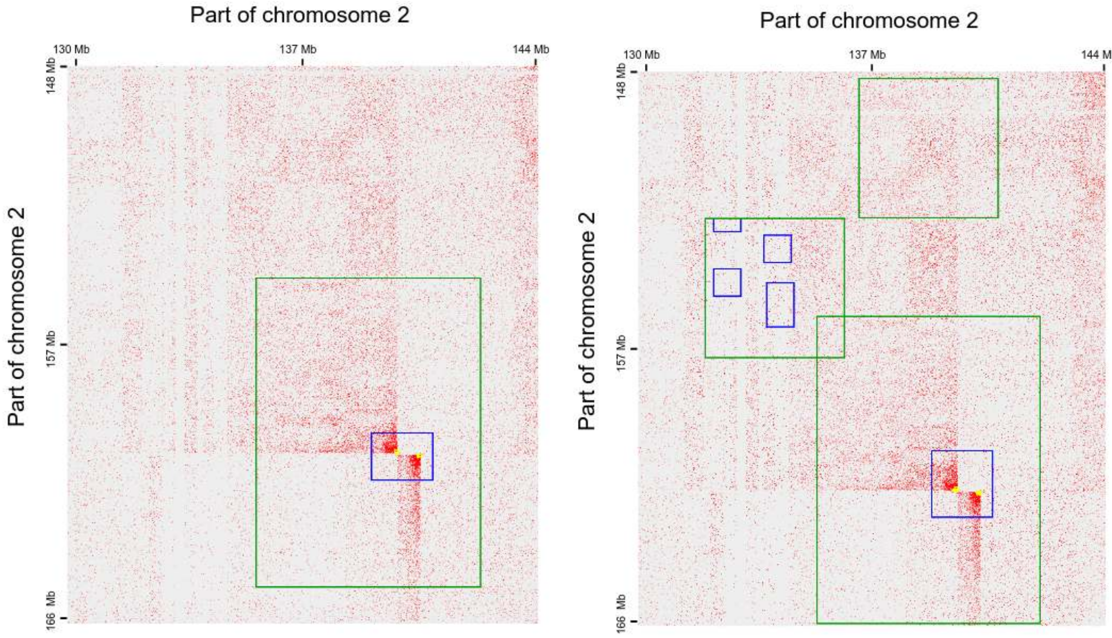


Figure 3.10: The upper triangle shows the SV detection results with KR normalized matrix and the lower triangle shows the detection results with the raw map for chromosome 2 vs 2 of chromothripsis case. It is visible that the tool detected more falsely positives, especially near the centromere, in KR normalized map.



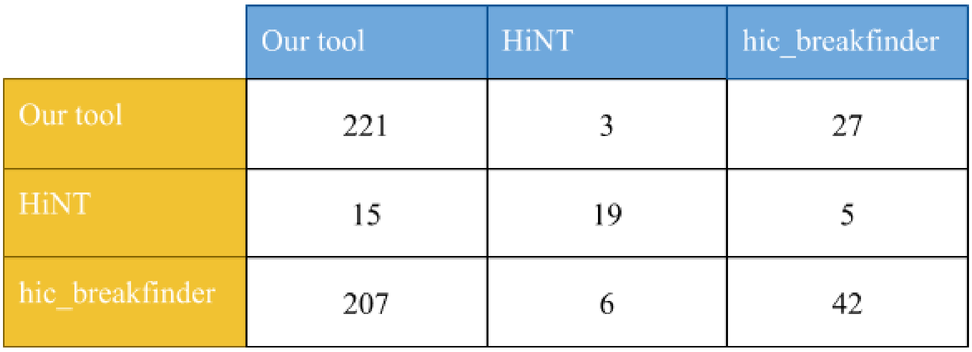
(a) Enlarged region in raw map

(b) Enlarged region in KR normalized map

Figure 3.11: Panel (a), raw, and panel (b), balanced, shows the same region. Green boxes: detection results at 500 kb resolution, blue boxes: detection results at 100 kb resolution, yellow boxes: detection results at 25 kb resolution. More boxes were detected resolutions in normalized Hi-C map at 500 and 100 kb. But at 25 kb resolution, the tool filtered out the falsely positive candidates and returned the same results with the raw map. Pixel intensity is the dominant feature in lower resolutions that is why many regions were picked up by the algorithm in the balanced matrix. However, at 25 kb resolution, other features such as edge detection play additionally important roles and many false positives are filtered out as shown in the figure, in which both panels (a) and (b) have the same number of yellow boxes.

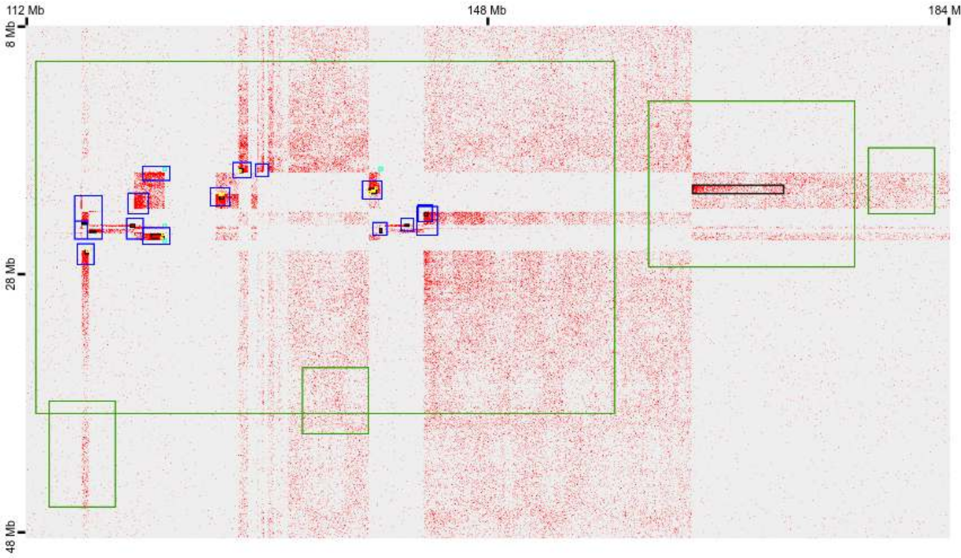
3.5 Comparison with other tools on chromothripsis case

This section is on a preliminary comparison of our tool with the existing tools that aim to detect SVs in Hi-C maps. Two tools are used for comparison; Hic_breakfinder [26] and HiNT [27]. Hic_breakfinder runs with a probabilistic model that tries to estimate the regions that have high observed intensity relative to expected intensity [26]. HiNT aims to detect SVs first at 1 Mb then at 100 kb resolution and then tries to find exact breakpoint by chimeric reads from the candidate region [27]. HiNT is focused on CNV and translocation detection. It does not work on cis maps. Hic_breakfinder is developed for cis and trans map SV detection, in this sense, similar to our tool. More detailed information about other tools is provided in the discussion section 4.2. This section aims to give an intuition about the performance of our tool compared to the others. The comparison is made on 6 maps in total. SV breakpoint calls from HiNT and hic_breakfinder are performed by my supervisor. In the following section, the chromothripsis case example and K562 cancer cell line comparisons are shown. The other four maps comparison can be found in the supplement. For the chromothripsis map, all three tools are compared. For the map of the cancer cell line, only HiNT is compared to due to time limitations.

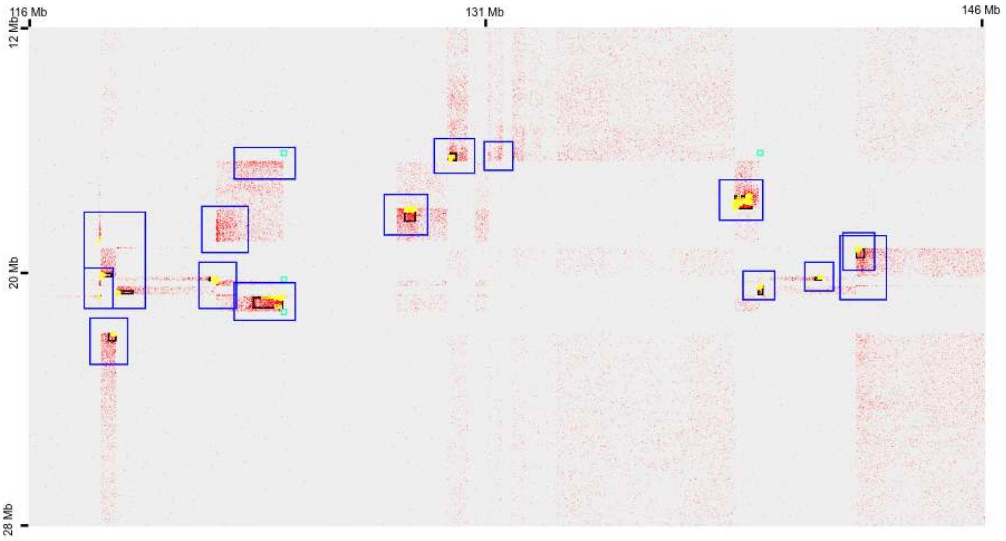


	Our tool	HiNT	hic_breakfinder
Our tool	221	3	27
HiNT	15	19	5
hic_breakfinder	207	6	42

Figure 3.12: Comparison of three tools on chromothripsis case. Matrix diagonal represents the total number of detected SVs by each tool. For instance, our tool detected in total of 221 candidate breakpoints at 25 kb resolution. This number much higher than the other tools because our tool returns more than one box for one SV. Unfortunately, our tool does not provide enough precision yet. This problem is pointed out in the discussion part 4.1.3. The rest of the matrix cells gives information about the intersection between tools. For instance, our tool matches 27 out of 42 detections from hic_breakfinder. HiNT matches 5 out of 42 detections from hic_breakfinder. Hic_breakfinder matches 207 out of 221 detections from our tool and so on.



(a) A region of chromothripsis chromosome 2 vs 11 Hi-C map



(b) Enlargement of chromothripsis chromosome 2 vs 11 Hi-C map region

Figure 3.13: Green boxes: our tool candidate regions at 500 kb resolution, dark blue boxes: our tool candidate regions at 100 kb resolution, yellow boxes: our tool candidate regions at 25 kb resolution, black boxes: Hic_breakfinder candidate regions, light blue: HiNT candidate regions.

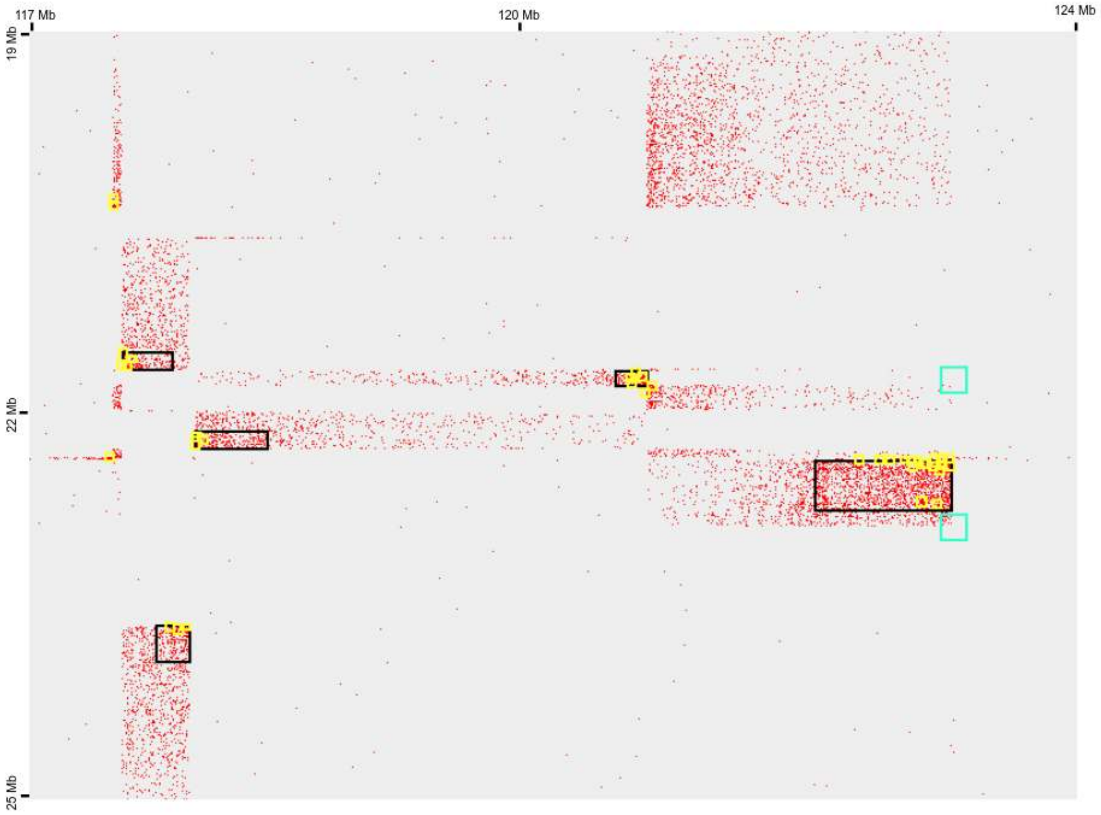


Figure 3.14: Yellow boxes: Our tool candidate regions at 25 kb resolution, Black boxes:Hic_breakfinder candidate regions, light blue: HiNT candidate regions. This figure shows that HiNT tool is a bit far from the real SV region. Hic_breakfinder and our tool detected many of the SVs. Also in this figure, the spread of annotated candidate regions around SV breakpoints of our tool at 25 kb resolution is visible.

3.6 K562 cancer cell line

Our tool has been tested on a Hi-C map of K562 with a reference map GM12878 (GEO accession number GSE63525). K562 is a known cancer cell line derived from female chronic myelogenous leukemia patient. GM12878 is a lymphoblastoid cell line obtained from female donor. Both cells have a common ancestor blood cell. Detection of SVs in cancer cases are particularly important because SVs can contribute to oncogenesis [26] SV breakpoint calls from HiNT are provided by my supervisor.

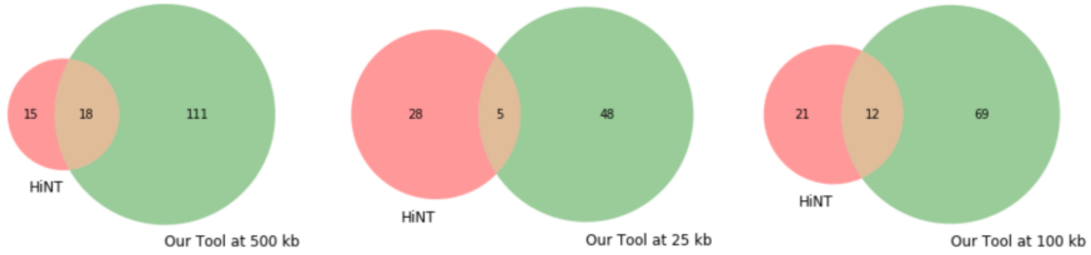


Figure 3.15: Comparison of the number of detected SV breakpoints in our tool at different resolutions and HiNT in the K562 Hi-C map.

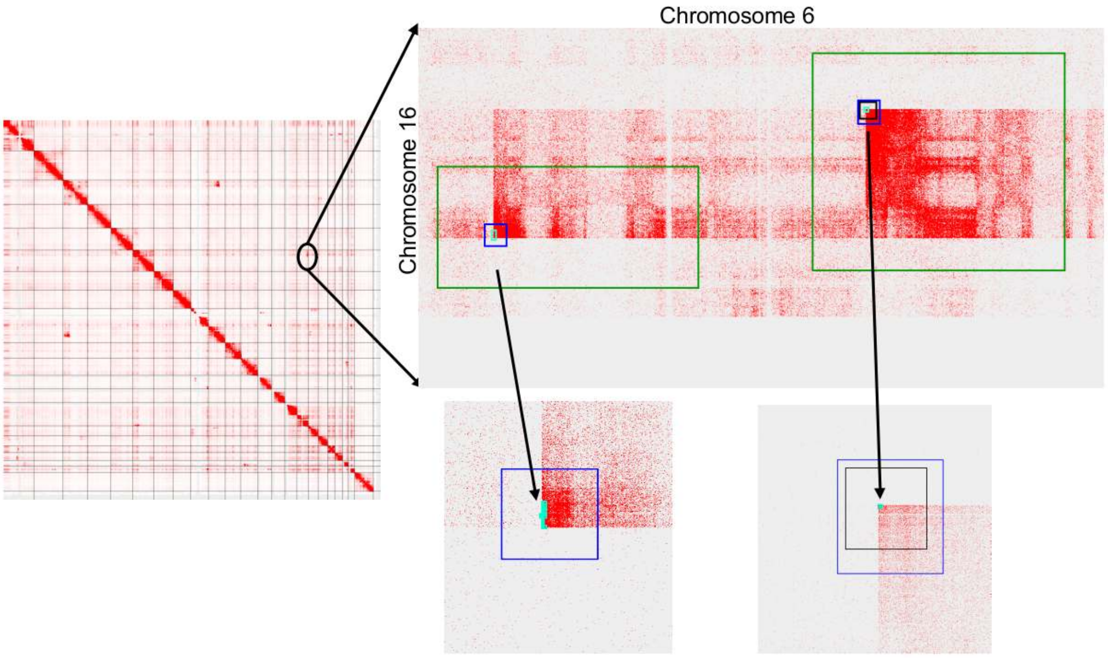


Figure 3.16: One translocation example in K562 Hi-C map is shown. The green boxes: detection at 500 kb, dark blue boxes: detection at 100 kb, light blue boxes: detection at 25 kb and black boxes: detection by HiNT tool at 1Mb. This example shows that HiNT missed one SV breakpoint to detect.

Chapter 4

Discussion

SVs can cause changes in the 3D conformation of the chromatin and in this way, they can cause deviations from the reference sample in the Hi-C map. Unfortunately, deviations between Hi-C maps can be a result of many things such as inherent structure of Hi-C data, biological variance between samples and experimental variances. On top of that, nested SVs can create complex patterns in a Hi-C map that are difficult to analyze.

The discussion part is divided into three sections. The first section is dedicated to the reasons for false positives SV candidate calls. This section contains information about compartmentalization, secondary effects of SVs, the precision of our tool, the mappability issue and CNV detection. The second section is about comparisons with other existing tools. The third section is devoted to approaches that might advance our tool. This section gives an outlook for possible future work. Approaches to improve our tool includes the selection of training set, designing features, matrix normalization, selection of reference Hi-C matrix and exact breakpoint detection.

4.1 Investigating the false positives

4.1.1 Compartmentalization

Compartments are usually observed in cis Hi-C maps and rarely occur in trans Hi-C maps. They are closely related to TADs. A part of a genome that is in a TAD tends to interact more with the parts of the genome that are in the same TAD than with the parts of the genome that are outside of the TAD. Compartments occur where the genomic parts that interact with each other more often over long distances. TADs and compartments have higher total contact frequencies than other parts of the Hi-C maps. Compartments do not produce the same pattern as SVs. Compartments neither prdouce a similarly sharp corners nor pronounced gradients. In contrast to SVs, compartments usually have more equally distributed intensities in some rectangular areas. Still, their occurrence can be confusing for our algorithm. Due to this reason a reference map has been used. It is aimed to remove the patterns in Hi-C maps originating from TADs and compartments by subtracting a reference map from a sample map. Ideally, signals on the substraction map merely come from the deviation between maps and no signal belongs to inherent chromatin structure. TADs and compartments can differ from one cell type to another, so it is important to select a reference map with the same cell type with the sample map.

Even though the same cell types are used in sample and reference maps to produce the difference map, occationally some compartmentalization patterns are still visible. The reason for that sample and reference can have different compartmentalization although they are from the same cell type. The difference in compartmentalization might come from chromatin rearrangement caused by an SV or it might be result of some other biological reason. In any case, these compartmentalization patterns, that occur in the sample but not exist in the reference, might be detected falsely positive.

4.1.2 Secondary Effect of SVs

Chromothripsis and cancer Hi-C maps have several SVs. Many of them are occurring in nested forms with several SVs consequentially affecting on the same part of the genome. This results in a disruption in the basic patterns of SVs (as in Figure 1.6) and creates new, more complex patterns. Since there is an unlimited combination of SVs possible that can one after another occur, it is difficult to derive kernels associated with complex patterns of nested SVs. For instance, in the chromothripsis case, there are cut/-copy and paste translocations that occur between chromosome 2, 5 and 11. In the Hi-C map of chromosome 2 vs chromosome 5, shown in Figure 4.1, many stripes are seen indicating many SVs. But in fact, some blank stripes occur because some parts of already structurally varied regions of genome cut and pasted into another chromosome. In other words, two translocations occur. The first translocation is between chromosome 2 and 5 and then some parts of the same region were translocated additionally to chromosome 11. This leads stripes with almost the same height and space between them. One of the stripes has a higher corner intensity. Usually, that is the primary or main SV. The following stripes are a part of the original SV, but appear as several stripes because the parts between were moved somewhere else. Differentiating the stripe patterns caused by nested SVs and orginal breakpoints of SVs is a very difficult task. A researcher who is specialized in Hi-C has to analyze the map and find the translocated regions somewhere else in the whole genome-wide Hi-C map. In the case of finding the second translocation, there will be a perfect match in the patterns. But until then, a nested SV can be easily confused with several independent SVs. Our tool is also fooled by nested SVs as well as the other tools. It detects the so-called secondary effects of nested SVs. Unfortunately, chromothripsis and cancer cases are full of nested forms of SVs. Many of the secondary effects have sufficiently high intensity to be detected, especially when cut and paste event occurs in the trans maps. This is the reason why our tool has detected some of the secondary effects of SVs.

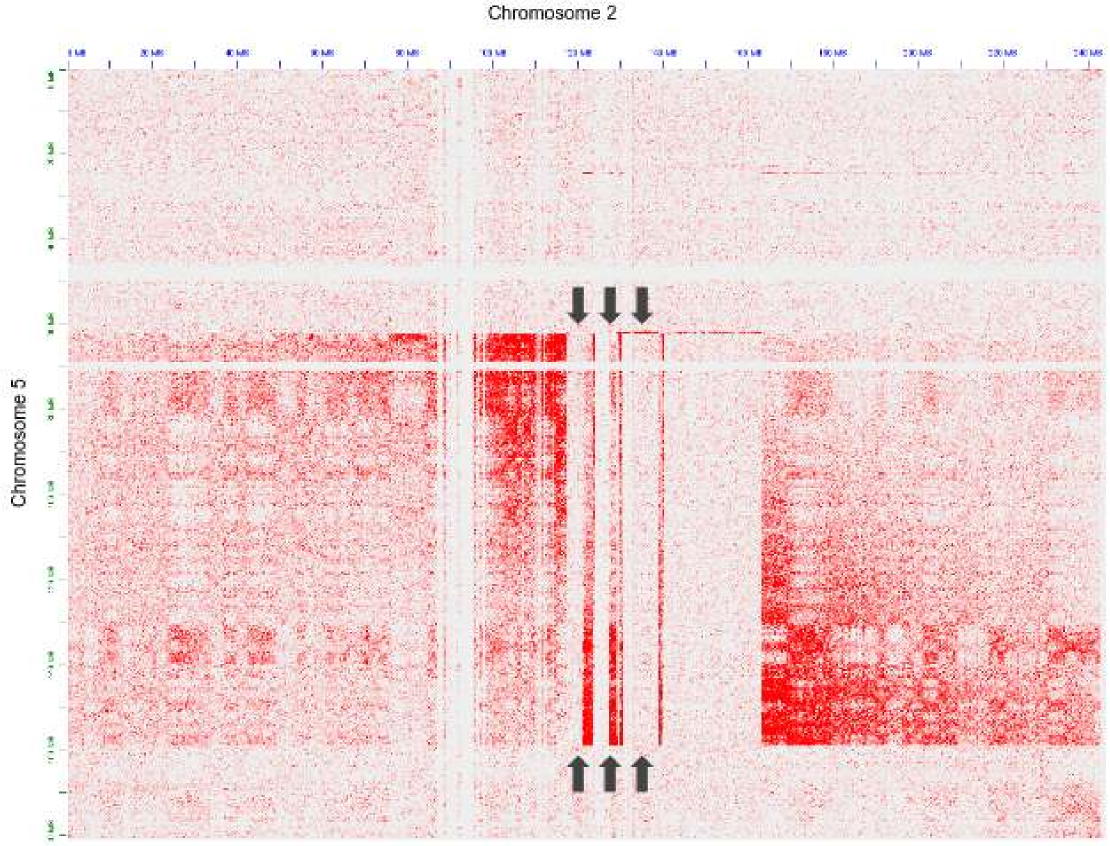


Figure 4.1: Chromsome 2 vs 5 Hi-C map. Example of an stripe patterns that occur with the overlaping translocations is shown.

4.1.3 Precision

The tool returns at every resolution stage one file with candidate Hi-C pixels and one file with regions surrounding one or several candidates, thus in total 6 output files for one sample Hi-C binary file. Each output file contains the prediction of the classification models that have been constructed. Logistic regression and random forest methods have been tried out to decide which method classifies data better. We observed that logistic classification usually returns better results at a lower number of features while random forest returns better results at a higher number of features. For 500 kb and 100 kb resolution, we decided to use logistic regression since at lower resolutions not many features were necessary to describe the SV induced pattern. For 25 kb, we decided to switch the method for classification and use random forest classification since a higher number of features was used. However, our model with the random forest method returns accurate results with not enough precision. Meaning that the output candidates are accurate in labeling parts of SVs, but the locations of them are not necessarily at the breakpoint. For most of the cases, not only a 25 kb pixel that includes the breakpoint is found but also neighboring pixels are detected. In other words, the candidates that our tool found sometimes seems to spread across the SV region rather than localizing at the breakpoint. To get rid of this problem, the model has to be better defined. Either training set has to be chosen more carefully, or the feature set has to be improved to enable filtering out any neighbor pixel but keeping the exact breakpoint pixel. Because of precision problem, the total number of detected SV breakpoints might be higher than it should have been.

4.1.4 Mappability

Mappability is a known problem for genome studies. Some regions of the genome are highly repetitive. For instance, the centromere and telomere of chromosomes are known high-repetitive regions of the genome. In these problematic regions, it is not possible to map any reads that support interaction with any other part of the genome and that is why in Hi-C maps they form a line of zero intensities. Unmappable regions might slightly differ for the reference and interested sample which can cause fluctuations in the signal the difference map near the regions that have mappability issue. An intensity gradient change can occur if some pixels have zero intensity and some pixels have some intensity value due to fluctuations in the problematic region. Since these parts also have the shape of the line, they can be falsely detected by the tool some times.

4.1.5 CNV Detection

For copy number variation (CNV) detection usually, read depth-based methods are used. The idea is simple. If one part of the genome has half more amount mapped reads than the rest of the genome than there is a good chance that this part in one of two chromosomes is duplicated so that 1.5 amount of reads are mapped to this region. The read depth information comes from the 1D coverage of the genome. It is not necessary to know the 3D organization of chromatin structure for this analysis. However, CNVs disrupt the 3D organization and their disruption is visible in Hi-C maps. Even though the design of our tool does not give separate attention to CNV detection, our tool manages to detect large CNVs since they disrupt 3D organization as well as 1D coverage.

4.2 Comparison with other SV detection in Hi-C map tools

So far, three other SV detection tools based on Hi-C maps exist hic_breakfinder [26], HiNT[27] and HiCtrans[28]. Hic_breakfinder follows an iterative approach to locate local clusters that deviate from the expected interaction frequencies in a contact matrix. Hic_breakfinder uses negative binomial distribution with different parameters for trans and cis Hi-C matrices. On Hi-C maps, that we compared our tool with, in general, hic_breakfinder results are compatible with our tool. On the other hand, SV detection results of HiNT were often off the SV breakpoint target. HiNT is a CNV and translocation detection tool and so far it is still at the level of a preprint. HiNT uses a rank product score that is based on Gini index to check the inequal distribution of the signal between surrounding pixels at 1 Mb resolution. The second parameter in their rank score is based on the signal intensity. HiCtrans uses change-point statistics that founded on 1D coverage profile to check for abrupt change points that occur in SV breakpoints. Both change-point statistics from HiCtrans and Gini inequality from HiNT approaches share the same idea to our tool's edge detection, horizontal derivation kernel, and vertical derivation kernel since these kernels are based on finding the high gradient change. HiNT works initially at 1Mb and then at 100kb resolutions, which is similar approach to our algorithm working steps. Despite these two similarities with HiNT and our tool, the SV detection results are not similar. HiNT returned rarely true positive SV regions at 100 kb resolution whereas our tool detected most of the SVs at 25kb that are missed by HiNT suggesting that either HiNT itself or our execution or parametrization of HiNT still has problems. HiCtrans has not been used in the comparison. Some authors from Hic_breakfinder paper are also authors in HiCtrans and hic_breakfinder is more recently published than HiCtrans. The SV calls from hic_breakfinder and HiNT in the comparison parts are provided by my supervisor.

4.3 Outlook

4.3.1 Improving the detection algorithm

The classification models are key in our tool. A better model can be obtained by adjusting two essentials: training sets and features. In the following section, possible improvements for the tool will be discussed.

Selection of training sets

The training set is hand-picked. Both negative and positive samples were annotated manually in one Hi-C map (chromothripsis case). Each resolution stage has its training set. In the beginning, we decided to have one training set at 25 kb resolution. For 100 kb resolution, 16 pixels at 25 kb would be aggregated (binned) and for 500 kb resolution, further aggregation of 16 pixels at 100 kb would be applied. However, binning can cause a shift in the intensity values. By chance, the breakpoint pixel, which is on the border of high and low intensity regions, could be binned with low intensity pixels and this causes a shift in the labeling between resolutions. That is why we decided to label a specific training set for each resolution. The training set is chosen carefully to represent each scenario of SV and non-SV regions. We also tried to compose a balanced training set by annotating almost equally large numbers of positive and negative labels. No matter how careful we are, there are some pixels we could not decide on labeling. Mostly secondary effects of SVs made labeling a challenging task. It was difficult to decide if they were SV on their own, or a result of secondary effects. Moreover, some corner pixels at low resolution did not show the highest intensity in their neighborhood due to binning effects. The training set can be improved by a more detailed analysis of the Hi-C map chosen or by considering more Hi-C maps for the training.

Design of kernels

As previously mentioned in the methods part, the design of the features is the most crucial part of our tool. Features aim to highlight the SV induced patterns and by this way facilitate the separation of the breakpoint-associated pixels from the rest of the map. Better design of the kernels will result in better classification by the model. Unfortunately, this task is challenging. Representing the occurring patterns into a matrix (kernel) format is challenging. Features also have to be compatible with every SV pattern in every Hi-C map or they should be comprehensive. The noisy nature of Hi-C data is also not helpful for SV induced pattern recognition. That is why at high resolution (25 kb) a larger kernel size is used such that more information from neighboring pixels can be integrated as well. Applying convolution with larger kernels has similarities with just detecting at a lower resolution for instance at 50 kb because the resolution is also a binned, summed, version of higher resolution stage. However, so far, each classification step is performed only based on the features obtained at the respective resolution and does not integrate features from other resolutions.

4.3.2 Matrix normalization

Hi-C matrix normalization methods aim to minimize the biases caused by the experiment and sequencing. These biases include fragment length, GC content, and mappability. One common normalization method is Knight and Ruiz matrix-balancing approach that aims to sum of matrix rows and columns adds up to 1. With normalization, signal intensities in some genome parts are increased while some parts are decreased such that each locus in the Hi-C map achieves an “equal visibility” [29].

However, in our case, SVs disrupt the chromatin organization and cause unexpected patterns. This means the tool looks for unbalances caused by deviations between the sample and reference Hi-C maps. SV induced pattern might be lost in KR normalized matrix, especially for CNVs since normalization will reduce the signal in these regions which should have a higher signal than the rest. One other problem with KR normalization is the introduction of artificially high values in low coverage regions which can fool the detection algorithm. On the other hand, in the case of raw maps, one has to deal with biases due to the experiment. The design of features and first trials of the tool has done on raw Hi-C maps. Later on, we tried only one case of KR normalization which is shown in the results part. Regions with the mappability issue received higher signal intensity due to matrix balancing and this makes SV detection more difficult, especially at lower resolutions. Since at lower resolutions, pixel intensity is more important than other features. At 25 kb resolution the importance of the features shifts from pixel intensity to edge detection. That is why our tool managed to filter out non-SV candidates at 25 kb and return a similar candidate list with the raw map. When the run time of KR normalized map and the raw map is compared, KR normalized map needs more time. Firstly, extracting KR normalized matrix is slower and secondly, there are more regions to analyze at 25 kb resolution since 500 kb and 100 kb resolution could not filter out many regions. The second part can be faster if the tool can be adjusted according to KR normalization needs. Features at 500 kb and 100 kb have to be more advanced such that models can filter out more regions. Another approach to get better results with normalization might be achieved by working with a ratio of matrices rather than differences since all features are based on a difference matrix and the magnitude of differences are influenced by the biases introduced by matrix balancing. These adjustments are on our list for future research.

4.3.3 Selection of a reference Hi-C binary file

Every genome has variations compared to a standard reference genome. Additionally, from one cell type to another the chromatin organization slightly differs. Since different cell types have different responsibilities in the species, it is not surprising to observe different compartmentalization and TADs in different cell types. Because of this reason, it is important to choose reference Hi-C file the same cell type as the sample Hi-C file. Reference Hi-C file has the responsibility of being the baseline for sample Hi-C binary file. In case of usage of the not compatible reference file, the tool is prone to detect false positives. However, it is not always easy to find a reference Hi-C file and even if it can be found there might be batch effects. Instead, a generalized reference Hi-C map for each cell type that is obtained by different experiments could be more standardized control. Moreover, the tool would be simpler to use since the input argument would be only the sample map and cell type. For these reasons, forming a standardized reference Hi-C map for each cell type will be the next step for the advancement of the tool.

4.3.4 Additional challenge in SV detection in X chromosome

Male mammalians have one X and one Y sex chromosome while females have two X chromosomes. If the reference and sample maps are from different sex, the read coverage for the X chromosome will be different and therefore the signal intensity in the chromosome X subtraction matrix will be different. If the sample is female and reference is male, then the subtraction matrix of chromosome X will have signals from the inherent chromatin structure which can be confused as an SV by the tool. If the sample is male and reference is female, then the subtraction matrix of chromosome X will have negative values and in existence of an SV in the sample, SV will not have the high signal intensity as expected because of subtraction of a reference matrix with a larger number of contacts. This issue might be solved with, either finding the same sex samples or by separating the matrix scaling stage for chromosome X.

Another challenge in SV detection X chromosomes is the X inactivated chromosome (XIC) in female mammalians. In the early stages of the female embryo, every cell decides randomly which X chromosome will stay active and which will be silenced. It is shown that the 3D chromatin structure of these active and inactive X chromosomes are different [30]. The Hi-C map represents diploid chromosomes as one. This means active and inactive X chromosomes overlap. If the reference and sample maps have different X chromosomes active, then the difference matrix will include deviations not only from probable SVs but also from X chromosome inactivation.

4.3.5 Exact breakpoint detection

As the resolution gets higher, fewer reads fall into the same bin and as a result, the Hi-C matrix gets sparser. Thus, it is difficult to observe specific patterns in high resolution Hi-C maps. This is one of the biggest challenges in detecting the breakpoint bases of SVs in Hi-C data. Obtaining visible SV induced patterns at higher resolution is only possible with more depth sequencing. However, at the moment this is not expected due to the budget limitations in the clinics. Our tool was designed for the detection of SV breakpoints until 25 kb resolution but not beyond this resolution. As a follow-up step after using our tool, one can go back to the reads that span the detected regions at 25 kb resolution and check for any split-reads that span the putative breakpoint.

4.4 Conclusion

Structural variation detection is an ongoing challenge despite their frequent occurrence in various diseases. Hi-C can be a useful method for SV detection by providing information on 3D chromatin organization. It provides a genome-wide contact matrix that can be visualized as a heatmap, called as a Hi-C map. SVs induce patterns in Hi-C maps by creating a high number of chromatin interactions between two formerly distal parts of the genome. This thesis aimed to develop an algorithm that recognizes the SV induced patterns in Hi-C maps in an automatized way. It works on three different resolution stages. It starts searching for SVs at the lower resolution and continues stepwise at a higher resolution. At each resolution, a scaled difference matrix is obtained by subtracting sample Hi-C matrix from the reference Hi-C matrix. Depending on the resolution stage, different sets of features are given to a pre-trained model to detect SVs on this difference matrix. At the end of each step, the candidates are aggregated if they are close to each other and genomic coordinates of these aggregated regions are passed on to the next step until the final SV candidates are obtained at 25kb resolution. The SV detection tool has been tested on 6 maps including one cancer and one chromothripsis samples. The results are briefly compared with other existing tools. It appears that most of SV calls are accurately reffering to SVs but the precision with respect to the breakpoints needs to be improved. The comparison with other tools indicates that all tools missed some breakpoints and in all there were some false positive SV calls.

Bibliography

[1] M. J. Rowley and V. G. Corces, "Organizational principles of 3d genome architecture," *Nature Reviews Genetics*, p. 1, 2018.

[2] M. Spielmann, D. G. Lupiáñez, and S. Mundlos, "Structural variation in the 3d genome," *Nature Reviews Genetics*, vol. 19, no. 7, p. 453, 2018.

[3] . G. P. Consortium *et al.*, "A global reference for human genetic variation," *Nature*, vol. 526, no. 7571, p. 68, 2015.

[4] P. Guan and W.-K. Sung, "Structural variation detection using next-generation sequencing data: a comparative technical review," *Methods*, vol. 102, pp. 36–49, 2016.

[5] V. Moncunill, S. Gonzalez, S. Beà, L. O. Andrieux, I. Salaverria, C. Royo, L. Martinez, M. Puiggròs, M. Segura-Wang, A. M. Stütz *et al.*, "Comprehensive characterization of complex structural variations in cancer by directly comparing genome sequence reads," *Nature biotechnology*, vol. 32, no. 11, p. 1106, 2014.

[6] D. A. Quigley, H. X. Dang, S. G. Zhao, P. Lloyd, R. Aggarwal, J. J. Alumkal, A. Foye, V. Kothari, M. D. Perry, A. M. Bailey *et al.*, "Genomic hallmarks and structural variation in metastatic prostate cancer," *Cell*, vol. 174, no. 3, pp. 758–769, 2018.

[7] C. Sismani, C. Koufaris, and K. Voskarides, "Copy number variation in human health, disease and evolution," in *Genomic Elements in Health, Disease and Evolution*. Springer, 2015, pp. 129–154.

[8] J.-B. André and T. Day, "The effect of disease life history on the evolutionary emergence of novel pathogens," *Proceedings of the Royal Society B: Biological Sciences*, vol. 272, no. 1575, pp. 1949–1956, 2005.

[9] P. Stankiewicz and J. R. Lupski, "Structural variation in the human genome and its role in disease," *Annual review of medicine*, vol. 61, pp. 437–455, 2010.

[10] M. R. Corces and V. G. Corces, "The three-dimensional cancer genome," *Current opinion in genetics & development*, vol. 36, pp. 1–7, 2016.

[11] J. Sebat, B. Lakshmi, D. Malhotra, J. Troge, C. Lese-Martin, T. Walsh, B. Yamrom, S. Yoon, A. Krasnitz, J. Kendall *et al.*, "Strong association of de novo copy number mutations with autism," *Science*, vol. 316, no. 5823, pp. 445–449, 2007.

[12] J. Weischenfeldt, O. Symmons, F. Spitz, and J. O. Korbel, "Phenotypic impact of genomic structural variation: insights from and for human disease," *Nature Reviews Genetics*, vol. 14, no. 2, p. 125, 2013.

[13] R. Redon, S. Ishikawa, K. R. Fitch, L. Feuk, G. H. Perry, T. D. Andrews, H. Fiegler, M. H. Shapero, A. R. Carson, W. Chen *et al.*, "Global variation in copy number in the human genome," *nature*, vol. 444, no. 7118, p. 444, 2006.

[14] B. J. Trask, "Human genetics and disease: human cytogenetics: 46 chromosomes, 46 years and counting," *Nature Reviews Genetics*, vol. 3, no. 10, p. 769, 2002.

[15] A. L. Tarca, R. Romero, and S. Draghici, "Analysis of microarray experiments of gene expression profiling," *American journal of obstetrics and gynecology*, vol. 195, no. 2, pp. 373–388, 2006.

[16] R. Nielsen, J. S. Paul, A. Albrechtsen, and Y. S. Song, "Genotype and snp calling from next-generation sequencing data," *Nature Reviews Genetics*, vol. 12, no. 6, p. 443, 2011.

[17] F. J. Sedlazeck, H. Lee, C. A. Darby, and M. C. Schatz, "Piercing the dark matter: bioinformatics of long-range sequencing and mapping," *Nature Reviews Genetics*, vol. 19, no. 6, p. 329, 2018.

[18] K. Mukherjee, D. Washimkar, M. D. Muggli, L. Salmela, and C. Boucher, "Error correcting optical mapping data," *GigaScience*, vol. 7, no. 6, p. giy061, 2018.

[19] S. S. Rao, M. H. Huntley, N. C. Durand, E. K. Stamenova, I. D. Bochkov, J. T. Robinson, A. L. Sanborn, I. Machol, A. D. Omer, E. S. Lander *et al.*, "A 3d map of the human genome at kilobase resolution reveals principles of chromatin looping," *Cell*, vol. 159, no. 7, pp. 1665–1680, 2014.

[20] P. A. Knight and D. Ruiz, "A fast algorithm for matrix balancing," *IMA Journal of Numerical Analysis*, vol. 33, no. 3, pp. 1029–1047, 2013.

[21] N. C. Durand, M. S. Shamim, I. Machol, S. S. Rao, M. H. Huntley, E. S. Lander, and E. L. Aiden, "Juicer provides a one-click system for analyzing loop-resolution hi-c experiments," *Cell systems*, vol. 3, no. 1, pp. 95–98, 2016.

[22] E. Lieberman-Aiden, N. L. Van Berkum, L. Williams, M. Imakaev, T. Ragoczy, A. Telling, I. Amit, B. R. Lajoie, P. J. Sabo, M. O. Dorschner *et al.*, "Comprehensive mapping of long-range interactions reveals folding principles of the human genome," *science*, vol. 326, no. 5950, pp. 289–293, 2009.

[23] N. C. Durand, J. T. Robinson, M. S. Shamim, I. Machol, J. P. Mesirov, E. S. Lander, and E. L. Aiden, "Juicebox provides a visualization system for hi-c contact maps with unlimited zoom," *Cell systems*, vol. 3, no. 1, pp. 99–101, 2016.

[24] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, and M. Klein, *Logistic regression*. Springer, 2002.

[25] R. Couronné, P. Probst, and A.-L. Boulesteix, "Random forest versus logistic regression: a large-scale benchmark experiment," *BMC bioinformatics*, vol. 19, no. 1, p. 270, 2018.

[26] J. R. Dixon, J. Xu, V. Dileep, Y. Zhan, F. Song, V. T. Le, G. G. Yardımcı, A. Chakraborty, D. V. Bann, Y. Wang *et al.*, "Integrative detection and analysis of structural variation in cancer genomes," *Nature genetics*, vol. 50, no. 10, p. 1388, 2018.

[27] S. Wang, S. Lee, C. Chu, D. Jain, G. Nelson, J. M. Walsh, B. H. Alver, and P. J. Park, "Hint: a computational method for detecting copy number variations and translocations from hi-c data," *bioRxiv*, p. 657080, 2019.

[28] A. Chakraborty and F. Ay, "Identification of copy number variations and translocations in cancer cells from hi-c data," *Bioinformatics*, vol. 34, no. 2, pp. 338–345, 2017.

[29] E. Yaffe and A. Tanay, "Probabilistic modeling of hi-c contact maps eliminates systematic biases to characterize global chromosomal architecture," *Nature genetics*, vol. 43, no. 11, p. 1059, 2011.

[30] L. Giorgetti, B. R. Lajoie, A. C. Carter, M. Attia, Y. Zhan, J. Xu, C. J. Chen, N. Kaplan, H. Y. Chang, E. Heard *et al.*, "Structural organization of the inactive x chromosome in the mouse," *Nature*, vol. 535, no. 7613, p. 575, 2016.

Chapter 5

Supplements

5.1 Relationship between features

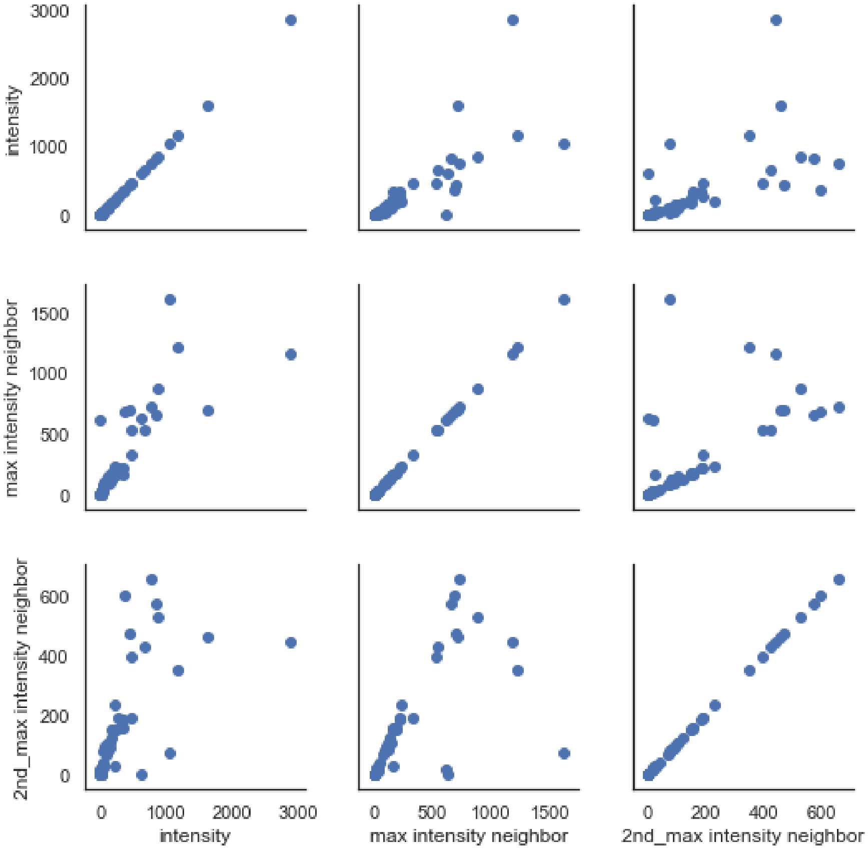


Figure 5.1: Relation between different feature pairs at 500 kb resolution is shown.

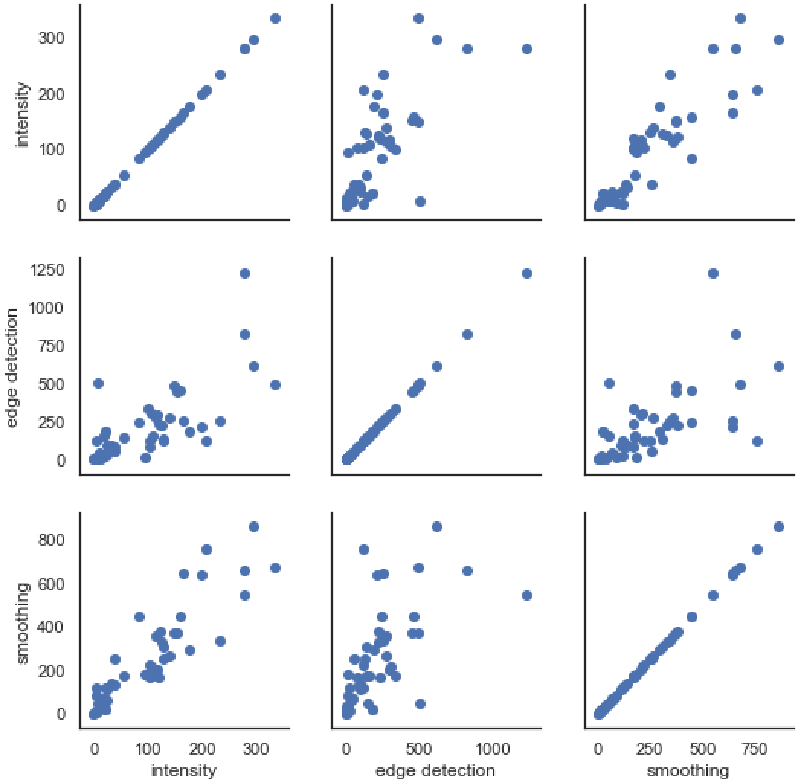


Figure 5.2: Relation between different feature pairs at 100 kb resolution is shown.

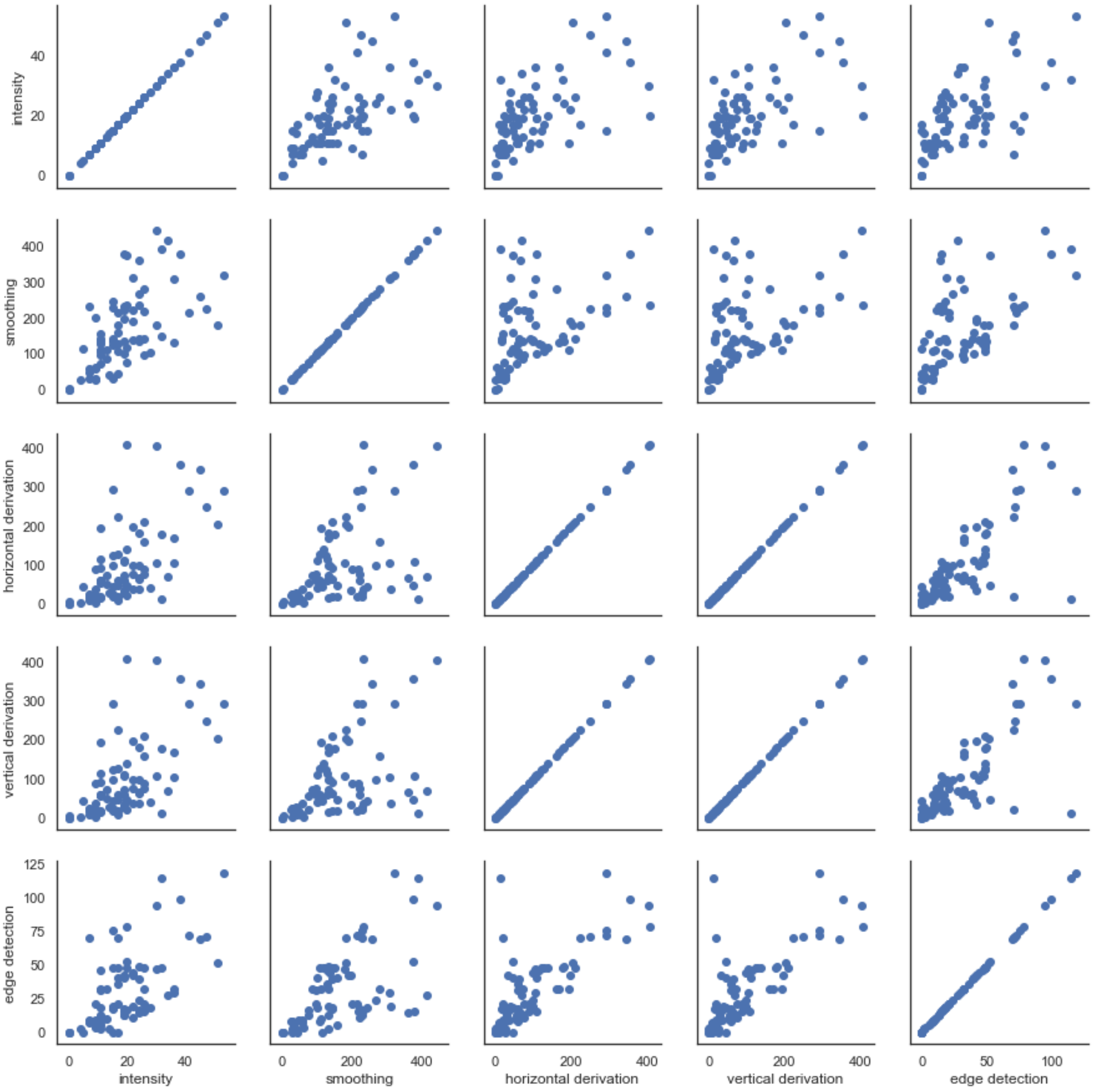


Figure 5.3: Relation between different feature pairs at 25 kb resolution is shown.

5.2 Further comparison with existing SV detection tools

5.2.1 Comparison of tools: Map 1

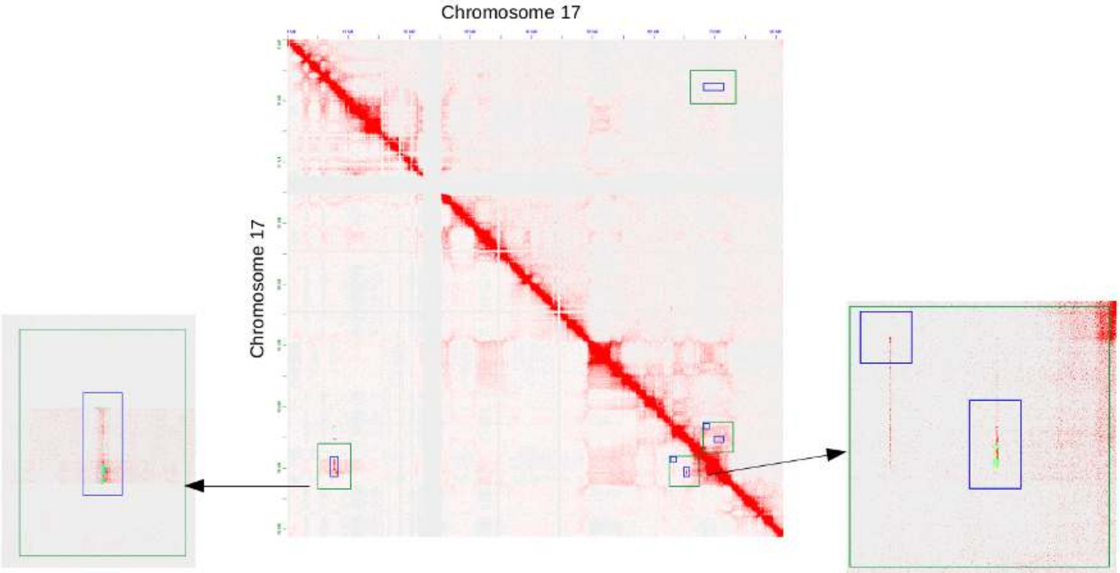
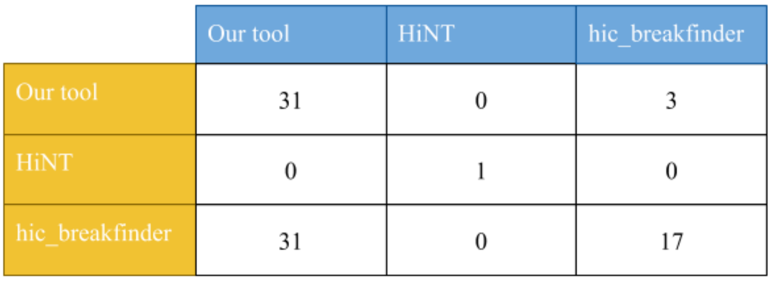


Figure 5.4: Chromosome 17 vs 17 Hi-C map is shown. Green boxes: SV detection at 500 kb, dark blue boxes: at 100 kb, light blue boxes: detection at 25 kb, yellow boxes: detection with hic_breakfinder. HiNT only detects translocations in trans maps and this example map is a cis map that is why there is no result from HiNT.



	Our tool	HiNT	hic_breakfinder
Our tool	31	0	3
HiNT	0	1	0
hic_breakfinder	31	0	17

Figure 5.5: Number of total detected SVs in all three tools are given in the diagonal of the table. The non-dioganal parts of the table, each cell shows how much of the detected SVs by cell column is also represented by cell row. For our tool the total number of SVs detected at 25 kb resolution is shown.

5.2.2 Comparison of tools: Map 2

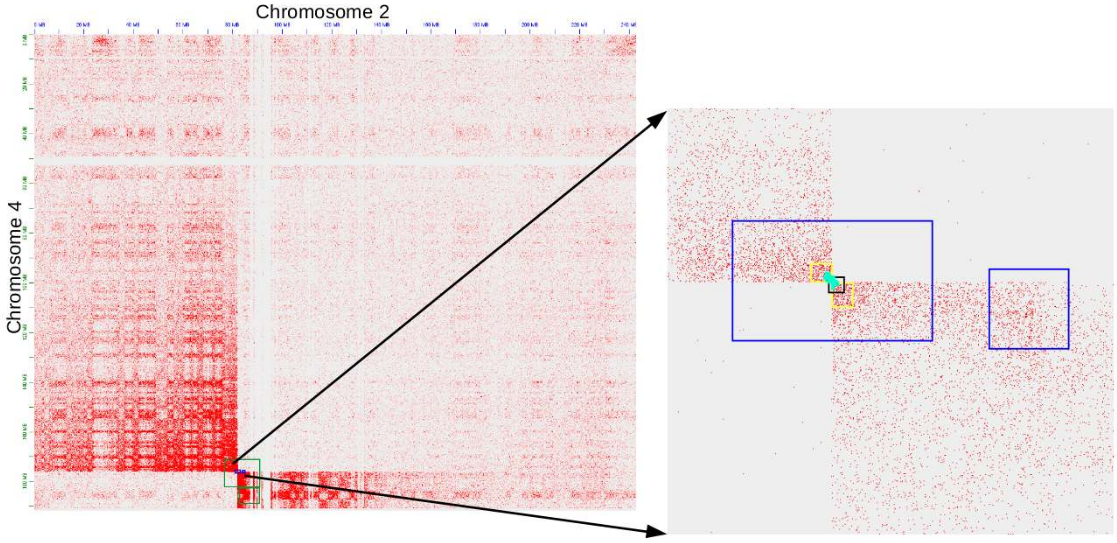
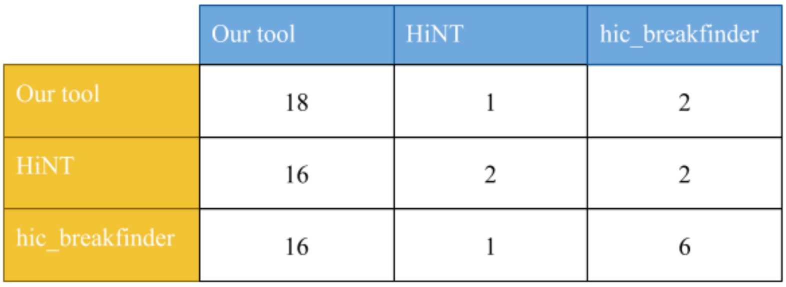


Figure 5.6: Chromosome 2 vs 4 region of a Hi-C map is shown. Green boxes: SV detection at 500 kb, dark blue boxes: at 100 kb, light blue boxes: detection at 25 kb, yellow boxes: detection with hic_breakfinder, black boxes: detection with HiNT. A translocation is shown in the Hi-C map. The smallest box that includes the breakpoints is provided by our tool.



	Our tool	HiNT	hic_breakfinder
Our tool	18	1	2
HiNT	16	2	2
hic_breakfinder	16	1	6

Figure 5.7: Number of total detected SVs in all three tools are given in the diagonal of the table. The non-dioganal parts of the table, each cell shows how much of the detected SVs by cell column is also represented by cell row. For our tool the total number of SVs detected at 25 kb resolution is shown.

5.2.3 Comparison of tools: Map 3

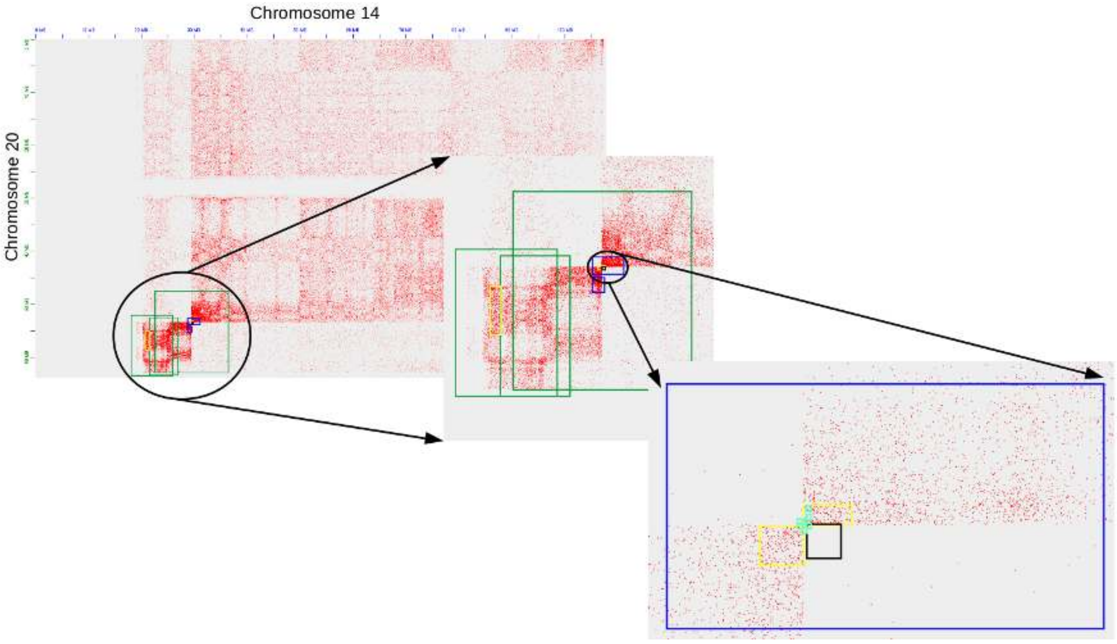
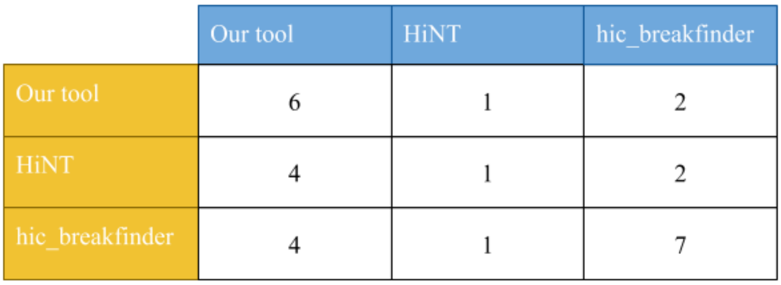


Figure 5.8: Chromosome 14 vs 20 region of a Hi-C map is shown. Green boxes: SV detection at 500 kb, dark blue boxes: at 100 kb, light blue boxes: detection at 25 kb, yellow boxes: detection with hic_breakfinder, black boxes: detection with HiNT. HiNT is bit off the breakpoint target.



	Our tool	HiNT	hic_breakfinder
Our tool	6	1	2
HiNT	4	1	2
hic_breakfinder	4	1	7

Figure 5.9: Number of total detected SVs in all three tools are given in the diagonal of the table. For the non-dioganal parts of the table, each cell shows how much of the detected SVs by cell column is also represented by cell row. For our tool the total number of SVs detected at 25 kb resolution is shown.

5.2.4 Comparison of tools: Map 4

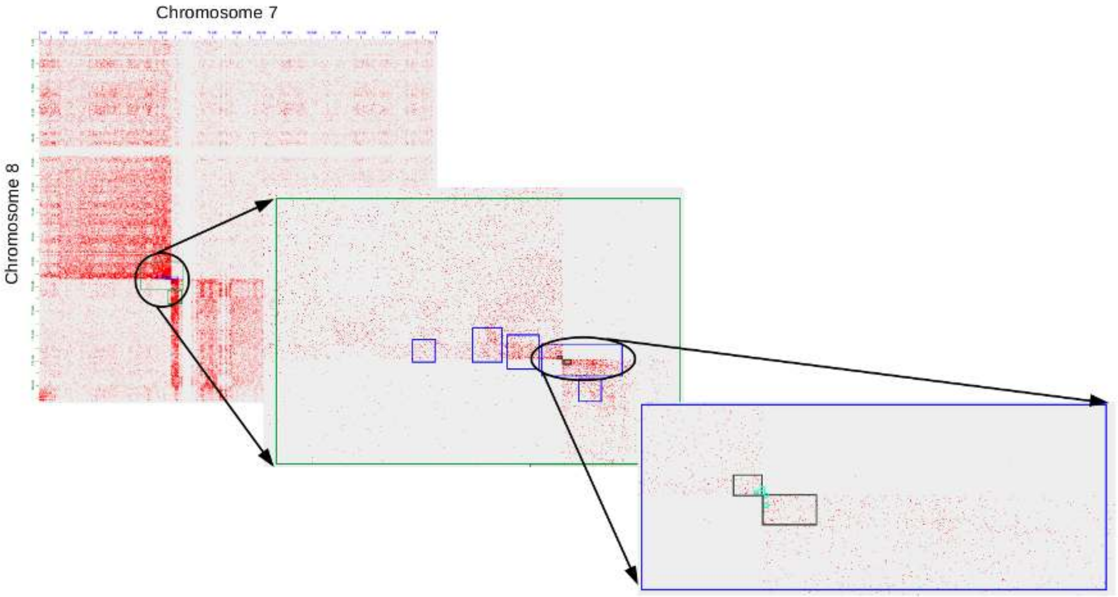
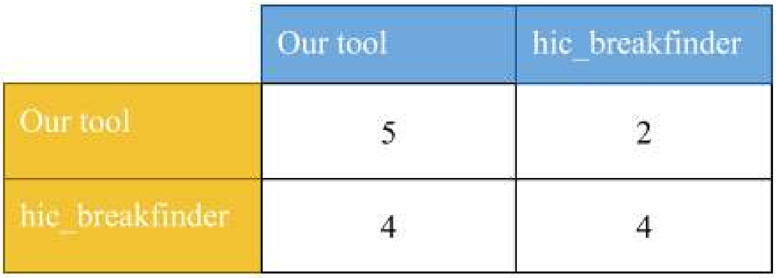


Figure 5.10: Chromosome 7 vs 8 region of a Hi-C map is shown. Green boxes: SV detection at 500 kb, dark blue boxes: at 100 kb, light blue boxes: detection at 25 kb, black boxes: detection with hic_breakfinder.



	Our tool	hic_breakfinder
Our tool	5	2
hic_breakfinder	4	4

Figure 5.11: Number of total detected SVs in two tools are given in the diagonal of the table. For our tool the total number of SVs detected at 25 kb resolution is shown. For the non-dioganal parts of the table, each cell shows how much of the detected SVs by cell column is also represented by cell row. For this table, 2 out of 4 SVs called by breakfinder are also represented by our tool. 4 out of 5 SVs called by our tool are also represented by hic_breakfinder.

Declaration of Authorship

I hereby declare that the thesis submitted is my own unaided work. All direct or indirect sources used are acknowledged as references.

I am aware that the thesis in digital form can be examined for the use of unauthorized aid and in order to determine whether the thesis as a whole or parts incorporated in it may be deemed as plagiarism. For the comparison of my work with existing sources I agree that it shall be entered in a database where it shall also remain after examination, to enable comparison with future theses submitted. Further rights of reproduction and usage, however, are not granted here.

This paper was not previously presented to another examination board and has not been published.

Date, Place

Signature