Detection of structural variants in cancer genomes using a Bayesian approach. You will find below the abstract of my PhD thesis

Daria Iakovishina

École polytechnique

Palaiseau Cedex

Laboratoire d'Informatique

Detection of structural variants in tumoral genomes with a Bayesian approach

Author:
Daria Iakovishina

Supervisors:
Mireille Régnier
Valentina Boeva

Jury:
President:
Daniel Gautheret
Rapporteurs:
Can Alkan
Gregory Kucherov
Examinateurs:
Mireille Régnier
Valentina Boeva
Michalis Vazirgiannis

Data of the defence: 30 November 2015

Contents

0.1 Introduction 6

1 State of the art 9

1.1 Cancer development 11

1.1.1 Proto-oncogenes 11

1.1.2 Tumor suppressor genes 12

1.1.3 DNA repair genes 13

1.2 Examples of structural variants that can result in cancer development 13

1.3 Sequencing technologies 17

1.3.1 Paired-end data 17

1.3.2 Mate-pair data 18

1.4 Approaches to structural variants detection 19

1.4.1 Pem based algorithms 19

1.4.2 Doc based algorithms 19

1.4.3 Split-read based approach 20

1.4.4 Combination of different approaches 20

2 Data management and alignment issues 21

2.1 Mappability of fragments 21

2.2 Alignment issues 21

2.2.1 Sequencing data 21

2.2.2 Reads alignment 22

2.2.3 Probabilistic approach to mapping position selection . 23

2.2.4 Paired reads alignment 24

2.3 Alignment algorithms 25

2.3.1 Aligner choice rationale 25

2.3.2 BWA drawbacks for structure variants detection . . . 26

3 Structural variant detection based on paired-end mapping signatures 27

3.1 Annotation of normal and abnormal read pairs 27

3.1.1 Detection of normal fragments orientation 28

3.1.2 Definition of normal insert size 28

3.1.3 Formal definition of normal and abnormal fragments . 29
3.2 Clustering of abnormal fragments 30
3.2.1 Cluster definition 30
3.2.2 Primary clustering algorithm 32
3.2.3 Splitting algorithm 35

4 Coverage issues 37
4.1 GC-content 37
4.2 Ploidy and copy number changes 39
4.3 Mappability 41
4.3.1 Repeats in human genome 41
4.3.2 Reads mapping and DOC 43
4.3.3 Mappability of a genome position 43

5 Abnormal and flanking regions 45
5.1 Definition of flanking regions and abnormal regions 45
5.1.1 Flanking regions 46
5.1.2 Abnormal region 48
5.2 Expected number of fragments in a genomic region 48
5.2.1 Expected number of fragments for a flanking region . 50
5.2.2 Estimation of the number of fragments starting on a position considering GC-content 51
5.2.3 Expected number of fragments for the abnormal region 51

6 Bayesian models 57
6.1 Model definition 57
6.2 Bayesian approach 58
6.2.1 Conditional probability of a model 58
6.2.2 A priory probability of a model 60
6.2.3 Factorization of the conditional probability 60
6.2.4 Set of models to test 61
6.2.5 Model choice 61
6.3 Evaluation of the breakpoint position 62

7 SV types and assembly workflow 63
7.1 Structural Variant types 63
7.1.1 Simple structural variations 63
7.1.2 Complex structural variants 71
7.2 SV assembly process 80

8 Overview of competitive methods for SV detection 87
8.1 Methods description 87
8.1.1 GasvPro 87
8.1.2 BreakDancer 88

8.1.3 Lumpy 88
8.1.4 Delly 89
8.2 Configuration used for considered software 90
8.3 Main features and scopes 91

9 Results on simulated and real data 93
9.1 Simulated data 93
9.1.1 Simulation of normal genome 94
9.1.2 Simulation of cancer genome 94
9.1.3 Simulation of sequencing data 95
9.1.4 Mate pair and pair-ended datasets statistics 97
9.2 Comparative performances on simulated data 97
9.2.1 Precision and recall 99
9.2.2 Detailed results 99
9.3 Comparative performances on a neuroblastoma mate-pair dataset 99
9.3.1 Experimentally validated structural variants 102
9.3.2 Predicted structural variations 102
9.3.3 SNP6-experiments 104
9.4 Discussion of the results 104
9.4.1 Sequencing technology and coverage 106
9.4.2 Influence of the presence of copy number variation on the SV prediction accuracy 106
9.4.3 Breakpoint resolution 106
9.5 Execution time comparison 107

10 Conclusion and perspectives 109
10.1 Conclusion 109
10.2 Perspectives 110

0.1 Introduction

Cancer is one of the most frequent causes of death worldwide, which makes the research on this disease extremely important. Cancer is a genetic disorder occurring as a result of progressive accumulation of mutations that lead to the malfunctioning of cells: uncontrolled growth, lack of contact inhibition and genomic instability. Cancer initiation is often caused by rearrangements of DNA sequence called structural variants (SVs). It is crucial to be able to precisely identify such SVs for early detection and accurate treatment of cancer.

With the development of next generation sequencing (NGS) methods, whole genome sequencing of paired-end reads became routine for detection of both small and large somatic mutations. Such mutations include point mutations, small indels and structural variants (SVs) in cancer genomes. Paired-end sequencing of mate-pair libraries is often employed when the aim of the study is the detection of large SVs, i.e., variants of length greater than the read length [PSO+10, SML+09, SGF+11, VB13].

Several methods for detecting structural variants with whole genome sequencing data have been developed so far. Most of them are based on paired-end mapping signatures or change in depth of coverage. Each type of large SVs (translocation, duplication, deletion, inversion, etc.) corresponds to a particular paired-end mapping signature (PEM) [ZBJL+10]. As such, deletions are characterized by an insert size (distance between mapped paired reads) larger than expected, while insertions have an insert size shorter than expected. Additionally, SVs often result in a change of copy number status around the breakpoint junction, which is reflected in changes in read depth of coverage (DOC). For instance, deleted regions have a relatively low DOC, whereas duplicated regions are characterized by a high DOC. Thus, differences in DOC and abnormal positioning of mapped reads often indicate the same genomic abnormality (e.g., a deletion or a tandem duplication).

The aim of my PhD was to create a computational method that combines both types of information, i.e., normal and abnormal reads, and demonstrate that this combination highly improves both sensitivity and specificity rates of structural variant prediction. I propose a Bayesian framework for SV detection using paired-end or mate-pair libraries, implemented as the software SV-BAY. In this framework, PEM signatures are combined with information about changes in DOC in regions flanking each candidate rearrangement. The method takes into account GC-content and mappability. The use of a Bayesian framework based on both PEM and DOC information allows to significantly decrease the level of false positive predictions while retaining high sensitivity. Additionally, SV-BAY infers 17 different types of structural variants from the detected novel genomic adjacencies. The algorithm introduced in SV-BAY makes it possible to detect SVs more accurately than any other existing method, recognizing more types of complex SVs. Moreover, unlike many other tools SV-Bay pipeline integrates a number of preprocessing and post processing steps. Preprocessing steps include filtering out reads with low mappability, determining normal fragment orientation, collecting fragment lengths statistics and estimating GC-content. Post processing steps include assembly of complex SVs and filtering out germline mutations. Rich integrated functionality makes it easy to use SV-Bay tool even without deep knowledge of sequenced data specifics. These SV-Bay advantages significantly simplify structural variants research, possibly contributing in development of efficient cancer prevention and treatment methods.

The thesis is organized in the following way. In chapter 1, the basic information about cancer development and modern sequencing technologies is provided. Different types of genes related to cancer are described, followed with the examples of structural variants that may cause cancer initiation and progression. Next generation sequencing (Ngs) technology and the difference between mate-pair and paired-end libraries are explained. Next, a brief overview of the existing approaches to structural variants detection is given, including methods based on paired-end mapping signatures, variation of Doc and split-reads.

In chapter 2, the mapping of sequenced reads to the reference genome is described. The issues of paired read mapping are covered. The choice of BWA as an aligner for our research is justified through a discussion of the advantages and drawbacks of this tool.

Chapter 3 covers the implementation of the approach for SV detection based on Pem signatures, which is a part of the SV-Bay algorithm. In this chapter the definitions of normal and abnormal fragments are provided, the algorithms to separate abnormal fragments and cluster them in order to get SV candidates are given.

Chapter 4 includes the information about factors that influence depth of coverage: GC-Content, ploidy and mappability of the region. The information about Freec and Gem tools, used in SV-Bay pipeline, is provided.

In chapter 5, special genomic areas, called flanking regions and abnormal region, are defined for each candidate SV. These areas are further used in chapter 6 to observe the Doc change around breakpoint junction. The methods to estimate the expected number of fragments in flanking and abnormal regions, considering factors described in chapter 4, are provided.

In chapter 6, the probabilistic Bayesian model, used in SV-Bay to filter out false SV candidates, is described. The probabilistic approach also allows to estimate the most probable number of gained or lost alleles and the most probable breakpoint position for the considered genomic adjacency.

In chapter 7, an exhaustive classification of structural variant types is given. For each SV the corresponding Pem signature is provided. Simple and complex SVs are defined and the algorithm used for complex SV assembly in SV-Bay is explained.

Chapter 8 provides an overview of four existing tools for SV detection:

GasvPro, BreakDancer, Lumpy and Delly. After a brief explanation of the design principles in each tool, the implementation is described; advantages and drawbacks are discussed.

In chapter 9, results on both simulated and real data are provided for SV-Bay and competitive tools considered. The process of simulation of mate-pair and paired-end sequencing data for tumor genome is described. The precision and recall rates are compared for all the tools. The quality of breakpoint resolution of each tool and the influence of dataset type (mate-pair or paired-end) and copy number variation (CNV) presence are also covered.

Finally, in Chapter 10, a general conclusion is drawn. Possible improvements and perspectives are discussed.

Chapter 1

State of the art

Cancer is the result of mutations and rearrangements in individuals DNA. Different types of structural variants are known to be related to different types and stages of cancer. For any stage, it is important to be able to accurately determine present SVs.

Some people can have a genetic predisposition to cancer by mutations in recessive tumor suppressor genes. Such mutations do not cause the disease themselves, but highly increase the probability of cancer development in case of a mutation in the second gene copy. Discovering alterations related to such mutations helps to understand that there is a probability of the appearance of the disease.

For cancer at an early stage, finding a particular genome alteration can help to detect the presence of cancer and its type. Such test allows to identify the disease even when the tumor is not yet observable and does not cause any specific symptoms. At late stages, knowing the exact structural variants involved helps to estimate the speed of cancer development and to understand whether the patient should be treated more aggressively.

One of the most important goals of structural variant detection lays in the personal medicine sphere. A lot of patients die because of a treatment not adapted to the specificity of their case. If every patient was treated with respect to the particular case, it could help to increase the survival rate and reduce the side effects of treatment.

Even after cancer has been treated, cells with mutations in responsible genes can still remain in the body. For now, it is proven that circulating tumor cells (CTC) are present in blood during the metastases development. These cells have the same structural variants as the original tumor cells. If the percentage of these cells exceeds some threshold, the probability of cancer relapse significantly increases. Such cells are usually larger than the rest, so they can be extracted from the blood test and checked for the presence of specific structural variants. This procedure can help to identify the metastases very early and decide that the treatment should be resumed.

The effect of cancer large rearrangements in cancer can be vividly demonstrated on the karyotype test results. Figure 1.1 depicts DNA karyotypes for normal (on the left) and cancer (on the right) cells. Normal DNA has two copies of each chromosome as expected, cancer DNA has several additional copies for almost each chromosome. Moreover, some chromosomes exchanged significant regions. For example, it is likely that two copies of chromosome 8 (purple) exchanged material with chromosome 10 (white).

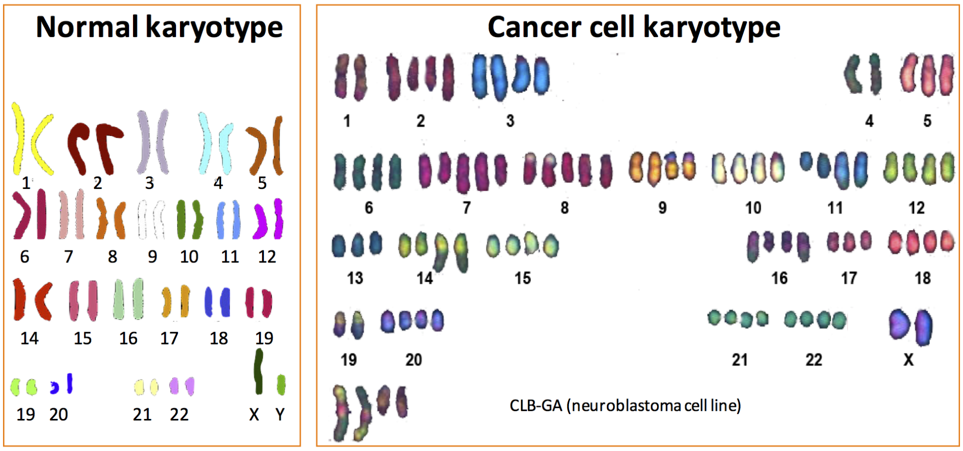


Figure 1.1: **(a) normal genome karyotype test**
(b) CLB-GA (neuroblastoma cell line) karyotype

Although karyotype test is good for visual representation, it is not an actual structural variants detection method. The reason is that it can recognise only alterations affecting regions of size comparable to the whole chromosome size. Moreover, it cannot show precise positions for structural variants, i.e. cannot identify which genes were involved in the rearrangement.

In this chapter, we provide the basic information about cancer, explain how sequencing data is produced and how it can be used for structural variant detection.

First, the origin and the main steps of cancer development are presented. Then the examples of particular structural variants which can be identified at different stages are provided.

Further we discuss genome sequencing technologies and introduce Next Generation Sequencing (NGS). Finally, a brief overview of present methods for structural variants detection is given.

1.1 Cancer development

Cancer development is a very complex process, which can significantly differ between cancer types and stages of the disease. There are approximately two hundreds of genes associated with cancer in contemporary researches. These genes are usually classified into three groups:

- Proto-oncogenes are responsible for the synthesis of proteins that stimulate cell division or prevent cell death. Malfunctioning forms of these genes are called oncogenes.
- Tumor suppressor genes are responsible for producing proteins that prevent cell division or accelerate cell death (in particular, trigger apoptosis).
- DNA repair genes help preventing mutations in all genes (including the first two groups) and thus preventing cancer development.

If the cell division becomes significantly more intensive than usual, it may contribute to the development of a tumor. The contribution to this process of mutations in the genes of each group is explained below.

1.1.1 Proto-oncogenes

The first group of genes related to cancer is proto-oncogenes. In normal cells they are responsible for production of proteins controlling the cell division. Such proteins are involved in a process called signal transduction cascade. The conversion of proto-oncogene into oncogene is called activation. It can occur with the involvement of three mechanisms ([LH00]):

- Mutation of or a translocation in the proto-oncogene. Such mutation can result in the intensified action of the encoded protein.
- Duplication (gene amplification) of a DNA segment that includes a proto-oncogene, leading to overexpression of the encoded protein.
- Rearrangement of genes in a chromosome or an inter-chromosomal translocation. Such rearrangements can move proto-oncogene to a new location under the control of different promoter, causing abnormal gene behaviour.

1.1.2 Tumor suppressor genes

The second group is tumor suppressor genes. These genes are responsible for synthesis of proteins suppressing cell growth and division. Such proteins may act in different cell areas: nucleus, cytoplasm or membrane. Mutation of these genes results in a loss of a function, which contributes to uncontrolled cell growth or division.

Tumor suppressor genes are usually recessive: the disease does not develop until both gene copies are mutated. The first mutation can be already present in the germ line cell, making all child cells inherit it. If a mutation later occurs in the second gene copy, the uncontrolled cell division starts. This leads to a higher cancer frequency among individuals inheriting the mutation in tumor suppressor gene than in the population as a whole.

It illustrates the fact that heredity can be an important cancer factor. However, even mutations in two copies of a tumor suppressor gene can occur in a somatic cell, usually caused by environmental factors.

There are several types of cancer, associated with tumor suppressor genes defects:

- Familial *adenomatous polyposis of the colon (FPC)* is caused by mutations in both copies of *APC*;
- Hereditary *breast cancer* is caused by mutations in both copies of *Brca2*;
- Hereditary *breast* and *ovarian cancer* is caused by mutations in both copies of *Brca1*.

Another typical example is *hereditary retinoblastoma*, a retina cancer that occurs in the childhood. It is caused by a mutation in the *RB1* tumor suppressor gene ([AD04]). Mutation in one copy is usually transmitted to the offspring from one of the parents. Mutation in the second copy is highly probable to occur because of a large number of retinoblasts and rapid division of cells of this type. About ninety percent of children inheriting *RB1* mutation develop retinoblastoma. Only individuals younger than eight years old have retinoblasts, so the risk of retinoblastoma exists only in the early childhood. However, *RB1* mutation is also dangerous for adults as it increases the risk of several other cancer types.

1.1.3 DNA repair genes

The third group of genes related to cancer is DNA repair genes. The proteins encoded by these genes are responsible for correcting the malformed nucleotide sequences.

The damage to DNA is very common and can be caused by various factors such as radiation, UV light, chemicals and poor environment. Errors in DNA replication can also cause DNA damage. The products of DNA repair genes fix the broken sequences and thus minimise the number of mutations in cells. When such a gene is mutated itself, it may not code for a functional corresponding protein any more. Lack of DNA repair significantly increases the frequency of cancerous DNA changes.

A well-known example of DNA repair gene is *Xeroderma pigmentosum* (XP). Malfunction of this gene causes an increased sensitivity to UV light. Individuals with such mutations have a thousand-fold increased probability for the development of all skin cancer types. Another example of a disease related to broken DNA repair is *Bloom syndrome*. It is an inherited ailment, caused by the mutation in BLM gene, required to support stable DNA structure. Individuals with this syndrome have a high frequency of DNA alterations, which leads to an increased risk of cancer and diabetes.

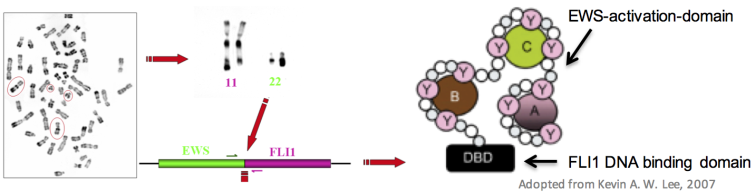
1.2 Examples of structural variants that can result in cancer development

Various examples of structural variants that are related to cancer are provided below. These examples include SVs that cause mutations in all three types of genes considered in the previous section: oncogenes, tumor suppressor and DNA repair genes.

There are several examples of inter-chromosomal translocations that cause the formation of new oncogenes. Such oncogenes are called fusion genes. Fusion genes contribute to tumor formation, producing proteins that are more (or even constantly) active. The result of such translocations is a modified form of the gene, contributing to cancer by accelerating the cell growth. Most proto-oncogene mutations are dominant: a single gene copy is enough to cause uncontrolled cell growth. The presence of an activated oncogene in germ line cells causes the child to inherit predisposition for cancer.

One of the most famous examples of this kind of genes is *EWS-FLI1* fusion gene. EWS-FLI1 is a chimeric protein formed by a tumor-specific translocation between chromosomes 11 and 22. Such translocations are found in both Ewing's sarcoma and primitive neuroectodermal tumor.

EWS-FLI1 amino-terminal domain is a much more potent transcriptional activator than the corresponding amino-terminal domain of FLI-1. Moreover, EWS-FLI1 efficiently transforms NIH 3T3 cells, while FLI-1 does not. Ews/Fli1 [Lee07], functioning as a transcription factor, leads to a phenotype dramatically different from that of cells expressing FLI-1.



Adopted from Kevin A. W. Lee, 2007

Figure 1.2: Fusion of chr11 and chr22 encoding for Ews/Fli1 causes Ewing sarcoma. Adopted from [Lee07]

Another example is the Philadelphia chromosome fusion of chromosomes 9 and 22. It gives a match of the *ABL1* gene on chromosome 9 (region q34) to a part of the breakpoint cluster region (*BCR*) gene on chromosome 22 (region q11) [RKMT03]. Such fusion encodes a new oncogenic protein called BCR/ABL. The detection of this translocation is a highly sensitive test for chronic myelogenous leukemia (CML), since 95% of patients with CML have this abnormality.

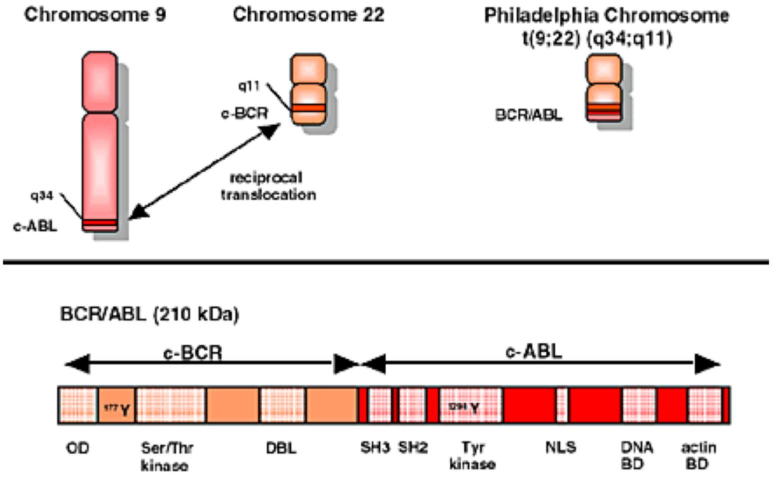


Figure 1.3: Philadelphia chromosome fusion of chr9 and chr22. (Thanks to James Griffin for the figure)

Not only translocations can damage a tumor suppressor gene or create a functional mutation in a proto-oncogene, all types of structural variants can be involved in the process. Below we consider SVs that cause gain and loss of the genetic material. Discovering such SVs helps to predict the aggressiveness of cancer development.

MYCN amplification, which occurs in approximately 22% of primary neuroblastomas, is one of the most powerful prognostic factors identified to date. It is significantly associated with advanced-stage disease, rapid tumor progression, and poor prognosis. Interestingly, it is shown in [BBPL+09] that children with MYCN-amplified, hyperdiploid, favourable-stage tumors had significantly better survival than those with diploid tumors. Figure 1.4 shows the Kaplan-Meier survival curves for 31 MYCN-amplified stage A, B, and Ds patients by ploidy ([Sch]).

Analysis of chromosomal aberrations is used to determine the prognosis of neuroblastomas (NBs) and to aid treatment decisions. It is shown in [CKN+10] that patients with with different genomic profiles have different survival rates. Authors studied MYCN gene amplification, 11q deletion and 17q gain, and genomes with numerical aberrations (i.e., whole-chromosome gains and losses).

Another example of the importance of the amplification detection is given by the *ERBB2* gene. Over expression of the receptor tyrosine kinase

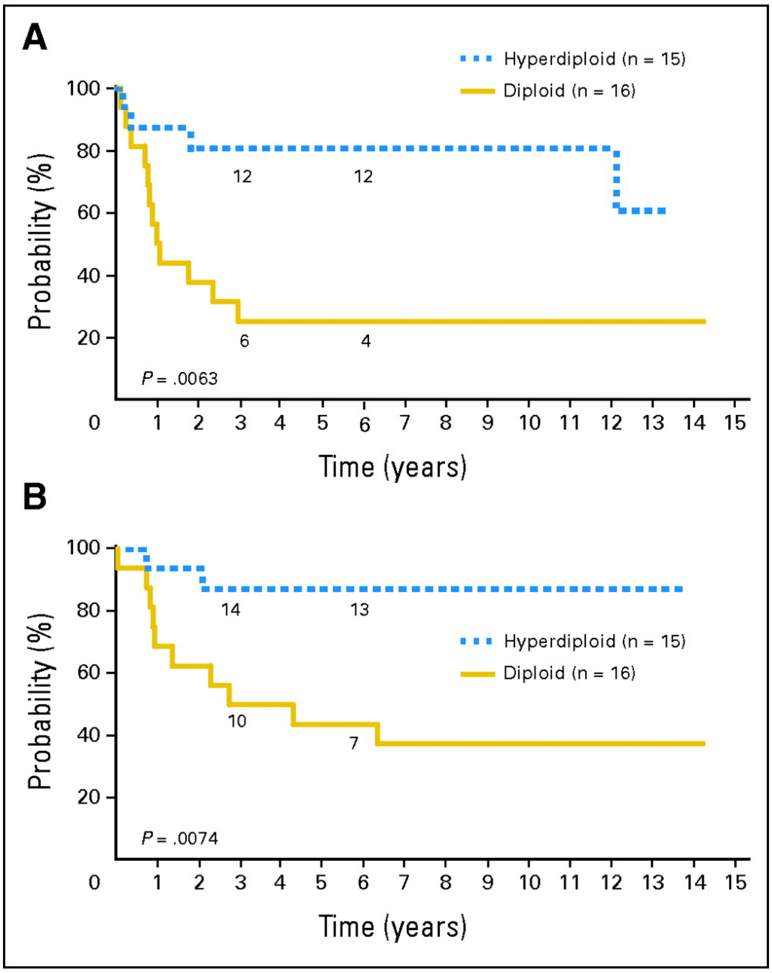


Figure 1.4: Kaplan-Meier survival curves for 31 Mycn-amplified stage A, B, and Ds patients by ploidy. *n* is the number of patients.
(A) Event-free survival, P = .0063;
(B) overall survival, P = .0074.[Sch]

ERBB2 (also known as HER2) occurs in around 15% of breast cancers and is driven by an amplification of the *ERBB2* gene.

1.3 Sequencing technologies

The ability to read a DNA sequence and produce a digital representation for it is the basis of a huge part of contemporary biological researches. The first approach answering this challenge was capillary electrophoresis (CE)-based Sanger sequencing. This technology gave the ability to extract the genomic information from any organism and thus was widely adopted by scientists around the world. However, this technology has significant limitations in speed, scalability and resolution, which make it hardly usable for various studies.

An entirely new technology overcoming these limitations, Next Generation Sequencing (NGS), was created at the beginning of the 2000s. NGS is a fundamentally different approach, which started a revolution in genomic science. This approach not only allowed to decipher whole human genome, but also reduced the cost of whole genome sequencing by three to four orders of magnitude during last 15 years.

The principle concept of NGS technology is similar to Sanger sequencing : the bases of a DNA fragment are sequentially identified from signals emitted as each fragment is resynthesized from a DNA template strand. The crucial difference is that NGS allows to read millions of fragments simultaneously. This enhancement allows the latest instruments to read large stretches of DNA in a massively parallel fashion, producing hundreds of gigabases in a single sequencing run.

NGS technologies can provide three types of data: single-end, mate-pair and paired-end short read data (Illumina, Life Technologies) and single-end long read data (PacBio). Both Illumina and SOLID paired-end and mate-pair sequencing produce pairs of reads suitable for the detection of large SVs. PacBio is the newest sequencing technology; the first commercial product, PacBio RS, was sold to a limited set of customers in 2010 and commercially released in early 2011. As it still has limited availability and high product price, we concentrate in this work on mate-pair and paired-end data and do not cover long single-end PacBio reads.

1.3.1 Paired-end data

The key steps of a sequencing project are the same for both mate-pair and paired-end technologies: preparation and amplification of template DNA, distribution of templates on a solid support, sequencing and imaging, base calling and quality control.

The first step in preparation of the sequencing library is DNA fragmentation. For this purpose, sequencing adapters are ligated to both ends of the

DNA fragments. Then PCR amplification using primers complementary to the adapters is performed.

Same adapters are placed on the flow cell (in Illumina SGS technology); then, fragments are placed on the flow cell and two complement adapters are attached to each other. The flow cell can have different shapes. For example, for the Illumina it is a flat glass plate; 454 uses beads with adapters on it, and there is place on the plate for each of the beads.

Once a fragment is attached to the corresponding adapter, polymerase creates a complement of all the sequence. Finally, the double-stranded DNA is unwinded, the original strand is washed away and the process is repeated.

The typical insert size (the distance between paired reads) for paired-end data is rather small: several hundreds of bases. The reads in a fragment in paired-end data are oriented towards each other. As explained further, paired-end data is less suitable for complex tasks including structural variants detection than mate-pair data. The main advantage of paired-end sequencing is its simple workflow making it widely spread.

1.3.2 Mate-pair data

Mate-pair libraries are created in a slightly different way. DNA is split into sequences that are longer than those for paired-end data. As for pair-ended technology, adapter sequences are ligated to both ends of the DNA fragments. Then the sequence is circularised: the two ends of the original DNA fragment are both adjacent to each other.

A special heavy biotin molecule is placed between the two adapters. When fragmentation of the circular DNA is finished, the fragment that contains original linear DNA ends is selected using biotin capture. Errors can be introduced at this stage, as it is not always possible to robustly choose the mate-pair fragments. As a result, the mate-pair data is usually contaminated with paired-end fragments with a different average insert size. Such fragments are called singletons.

The end of the sequencing process is exactly equal to the one used for paired-end, i.e. fragments are placed on the solid cell and amplified. Sequencing of both ends of the selected fragment yields reads that are separated by the length of the original fragment.

Mate-pair libraries allow larger insert sizes than paired-end, from 2 to 20 kilobases. Large inserts are especially valuable in *de novo* sequencing projects, where they can substantially improve scaffolding (ordering of assembled contigs). In contrast to paired-end reads, which are oriented towards each other, mate-pair reads are either both oriented outwards from the original fragment (Illumina protocol) or both have the same orientation (SOLiD protocol), which needs to be considered in the data analysis.

The major drawback of mate-pair sequencing is the complicated laboratory protocol. Another problem is that a substantially larger amount of

DNA (5 to 120 times) is required to prepare a mate-pair library.

1.4 Approaches to structural variants detection

In this section main SV detection approaches are briefly discussed to get a general idea about existing methods. Different in-depth details are provided in further chapters: most widely used tools implementing these approaches are presented in Chapter 8 and compared in Chapter 9.

Most of the current SV detection methods can be classified into three categories: methods based on paired end mapping (Pem) signatures, depth of coverage (Doc), and split-read mappings [MSB09]. Each approach has its own limits in terms of the types and sizes of SVs that it is able to detect.

1.4.1 Pem based algorithms

Pem-based algorithms may be based either on *read clustering* or on *fragment length distribution.*

The former category identifies discordant Pems as Pems with unexpected orientation or insert size, clusters them and applies statistical tests to validate candidate clusters [HAES09, KAM^{+}09, HHD^{+}10, ZBJL^{+}10].

The latter compare the observed insert size distribution of all read pairs in a given window versus the expected distribution. Windows with a significant proportion of read pairs having unexpected insert sizes are annotated as containing SVs (Lee et al., 2009).

In some cases the same package, e.g., BreakDancer [CWM^{+}09], provides two complementary methods for SV detection: clustering-based (BreakDancerMax) and distribution-based (BreakDancerMini) to detect large and small size SVs respectively.

1.4.2 Doc based algorithms

Doc-based methods detect regions in the genome where genomic material is gained or lost. They rely on some evaluation of the expected Doc, normalised for GC-content bias [YXM^{+}09, BZB^{+}11, VBTPKBPCJCGSIJLOD12]. A deviation from the expected Doc suggests putative gain or loss of genomic material.

Doc-based methods do not provide information about the adjacency of DNA regions involved in copy number changes. Thus, such methods are not able to indicate the type of SV (e.g., tandem duplication, fragment reinsertion, translocation) causing genomic loss or gain. Additionally, the resolution of such methods is rather low for low Doc datasets: a 30x coverage dataset allows approximately a resolution of 1Kb for rearrangement breakpoints.

1.4.3 Split-read based approach

Split-read based methods use partial read alignments for SV detection [WME+11, SHB+14, TEER14]. Although such methods may be efficient for data with high read coverage, they may fail to identify SVs with breakpoints located in repetitive elements of the genome.

Ideally, this approach should be combined with paired-end signatures; this idea was implemented in SVMerge [WKSA10], Prism [JWB12], Meerkat [YLG+13], Smufin [MGB+14] and Delly [RZS+12].

1.4.4 Combination of different approaches

Combining information about discordant Pems with changes in Doc is a promising solution for the SV detection problem. Probabilistic models integrating both the Doc signal and Pem signatures provide higher specificity together with equal or greater sensitivity than tools that simply use paired-end signatures [QZ11, ORA+12, SOP+12, ETB+13, LCQH14, HKNM11].

However, most of these methods do not take into account two important parameters that affect read count for both normal and abnormal mappings: GC-content and read mappability. Another general drawback of the majority of these methods is their lack of ability to detect all possible types of SV that can be present in cancer data including co-amplifications, tandem duplications with inversions, linking insertions, etc.

Chapter 2

Data management and alignment issues

2.1 Mappability of fragments

After the sequencing process is finished, differences between the sequenced and reference genomes can be determined. To do so, it is necessary to match sequenced reads with some positions in the reference DNA sequence.

De novo assembly, which re-builds the genome directly from the reads, is hard even for medium-size genomes. For the human genome it is impossible to do it automatically without manual control and biological correction of the results. Thus, alignment (also called mapping) of the sequenced short reads to the reference human genome is a mandatory step. During the alignment process a mapping position in the reference genome is found for each read.

The common approach to this problem is briefly discussed in section 2.2. First, we describe the structure of the data produced by sequencing machinery. Then, read mapping is discussed and the probabilistic approach to mapping position choice is introduced. Finally, the usage of paired reads information during alignment is explained. In section 2.3 the Burrows-Wheeler Aligner (BWA), used for data preparation in our work, is described. The rationale for this choice and BWA method drawbacks are given.

2.2 Alignment issues

2.2.1 Sequencing data

Most sequencing machines nowadays provide mate-pair or paired-end fragments (see chapter 1). Paired-end (mate-pair) reads usually are provided in two FASTA or FASTQ files. These files are used as an input for the alignment tools.

One file contains first reads while the second one contains second reads of each pair. Reads that correspond to the same fragment have the same names. Based on the names, the aligner can take into account connection between two reads. The length of the reads is constant for a particular dataset and depends of the data preparation or the choice of the sequencing machine.

Files in FASTA format store only the name and the nucleotide sequence for each read; possibly, some additional information can be provided, such as the name of the dataset or a particular sequence placement (mRNA, RNA, etc).

Files in FASTQ format extend this information with PHRED quality of each sequenced base. This quality score represents the probability of a correct recognition of each nucleotide. Given the base-calling error probability P, PHRED quality Q is defined as

$$Q = -10log_{10}[P]. \tag{2.1}$$

For example, if the probability of the base being called incorrectly is 0.001, Q is equal to 30.

In FASTQ files, PHRED quality for each base is encoded as an ASCII character by adding 33 (Illumina 1.8+) or 64 (Illumina 1.5+) (Standard ECMA-6: 7-bit coded character set. 6th edition. Ecma international (December 1991)) to the PHRED value. For instance, in the Illumina 1.8+ format, the character '!' represents the lowest quality. The examples of FASTA and FASTQ strings representing a single read are provided below.

Code 2.1: FASTA format : first line is *sequence id* (it can also contain some comments); second line is the raw sequence.

```
>gi|5524211|gb|AAD44166.1| cytochrome b (Elephas maximus
    maximus)
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCC
```

Code 2.2: FASTQ format: first line is *sequence id* (it can also contain some comments); second line is the raw sequence; last line provides quality for the sequence.

```
>SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCC
+[SEQ_ID]
!''*((((***+))((((++)((((().1***-+*''))**
```

2.2.2 Reads alignment

Intuitively, an alignment algorithm should search for the reference genome region exactly matching each read. Unfortunately, this approach can fail:

in several situations, the exact match cannot be found.

The most frequent reason for it is the presence of single-nucleotide polymorphisms (SNPs), that are alterations of a single base. SNPs define the genetic difference between two individuals. These polymorphisms typically form about 0.1% ([HA11]) of the whole genome. Since the sequencing data are aligned against a reference genome, which has been constructed as a consensus between several people, it is expected that many reads contain SNPs. Such reads do not match any region without mismatch.

Exact matching is also impossible in other cases: for example, sequencing errors may occur. Also, some reference genome regions are wrongly assembled and are difficult to map on.

As a consequence, the alignment algorithms look for the closest, but not necessarily exact, match. The alignment process complexity increases when the total number of allowed mismatches, k, increases. The complexity of the k-mismatch problem was shown to be $O(kn)$ [LV86, Mye86], where n is the whole genome length.

Definition 2.2.1 *A read is* uniquely mappable *if there exists exactly one position in the genome where its sequence maps with up to k mismatches.*

When a read is not uniquely mappable, it is a non trivial task to chose among the different possible mapping positions. Existence of sequencing errors and SNPs force to define a correct alignment up to some quality. A probabilistic approach to this problem is presented below.

2.2.3 Probabilistic approach to mapping position selection

Some of the fragments can have several alignment positions, possibly with different number of mismatches. For example, a read can be mapped uniquely with one mismatch at one position and with two mismatches at another. In order to choose between different positions in such situations, a Bayesian approach was proposed, published in [LRD08]. In this approach the quality of each alignment is measured by the *Phred mapping quality score*, noted Q_s, defined as a function of the probability that the mapping of fragment z at position u is correct. For any position u, let $p_s(u, z)$ denote the posterior probability that a read z originates at u in the genome x. Then Q_s is equal to the positive value

$$Q_s(u, z) = -10log_{10}[1 - p_s(u, z)] \ . \tag{2.2}$$

The higher is the probability $p_s(u, z)$, the higher is the value of Q_s, the better is the confidence for the choice. For a given short sequence read, the alignment that maximizes the score (2.2) is chosen as the best alignment.

To compute this score, it is necessary to compute the *prior probability* $p(z, v)$: the probability that z aligns at position v in genome x. A simple example is provided in [LRD08]. Let L denote the length of the reference genome and l denote the length of the read. First, a uniform prior probability is assumed. The posterior probability p_s is equal to

$$p_s(u, z) = \frac{p(z, u)}{\sum_{v=1}^{L-l+1} p(z, v)} \ . \tag{2.3}$$

This quantity is upper bounded by 1. This upper bound is reached if one prior probability $p(z, u)$ is 1, the other ones being 0. When a read can be mapped to multiple locations with similar prior probabilities, the ratio (2.3) will be different from 1. Accordingly, the mapping quality score Q_s will be lower, possibly zero. Observe that quantity $p(z, u)$ will be effectively zero for most possible alignments; therefore, only a small subset of all possible alignments (those that result in small numbers of mismatches) can be considered in evaluating the denominator.

Sequencing errors at different sites are assumed to be independent. Consequently, the probability, $p(z, u)$, is defined as the product of the probability of sequencing errors. Therefore, $-\log(p(z, u))$ is the sum of the PHRED score values of the bases that disagree with the reference sequence.

For example, if there is a single mismatch with base quality 20, we approximate the probability of sampling the read as 0.01. With two mismatches with base quality 20, the approximation becomes 0.0001.

The Q_s value can be equal for several mapping positions. This notably occurs if reads have the same number of mismatches with the same quality score. Mismatches may also have different quality scores. For example, the read can have two mismatches with score 10 at one mapping position and 1 mismatch with score 20 at another.

The alignment algorithms choose the position with the best mapping quality score Q_s for each read. The score Q_s is usually output and widely used by various genome alterations detection algorithms, such as the SAMTOOLS algorithm, to filter out reads mapped with low confidence and prevent false positive alteration discovery.

2.2.4 Paired reads alignment

To improve the mapping accuracy, alignment algorithms take into account the relation between two reads in the fragment. Fragments from the same dataset normally have approximately the same length (see chapter 1). This length depends on the data preparation process; it is called insert size.

Definition 2.2.2 insert size *for mate-pair or paired-ends fragments is the distance between the leftmost position of the left read and the rightmost position of the right read.*

In some cases, the first read of the fragment is uniquely mappable and the second read aligns to several mapping positions. If one of these positions is located within the expected distance (expected insert size) from the first read, it may be correctly mapped. A similar event occurs when both reads have several possible mapping positions and a unique couple of positions exists within the expected distance: the two positions in this couple are assumed to be correct and both reads are mapped to them.

In [BS12], the authors claim that the first case, where one read is uniquely mappable, while the other is not, occurs for 2% of fragments, only. This number is strongly related to the read length; moreover, it can be used only for a healthy genome, which did not undergo genomic rearrangements.

The second situation, where both reads not uniquely mappable, but one pair of positions can be chosen based on insert size, is much more common. For example, in our experiments during data preparation with BWA, this occurred for 20% of the reads.

2.3 Alignment algorithms

We have chosen the Bwa aligner [LD09] to map reads later used as an input to our method for SV detection. The rationale for using Bwa are given in 2.3.1 below and drawbacks are described in 2.3.2.

Bwa is a Burrows-Wheeler transform based tool that uses the Ferragina and Manzini matching algorithm to find exact matches [PF00]. Seeds are extracted from the reads and aligned to the genome with maximal exact matches. These seed alignments are extended with the affine-gap Smith-Waterman algorithm. To find inexact matches, the authors introduce a new backtracking algorithm that searches for matches between substring of the reference genome and the query within a certain defined distance.

2.3.1 Aligner choice rationale

Our choice is based on the overview of short sequence mapping tools given in [HBT13]. The authors use two types of data: synthetic (simulated by Wgsim tool) and experimental. Two main criteria are used for the comparison: *throughput* and *mapping quality*. The throughput is the number of base pairs mapped per second; this parameter reflects the tool performance. The mapping quality is addressed using three mapping characteristics: percentage of correctly mapped reads, errors (false positives) and ambiguous reads (reads mapped to more than one location with the same number of mismatches). The paper provides the dependency of the performances to the value of several parameters: read length, number of allowed mismatches, seed length for Bwt based aligners.

According to this paper, three tools outperform Bwa in mapping quality when the default options are used: Gsnap, Novoalign, Bowtie 2. But the default options for the number of possible mismatches are not the same. When all the tools are run with the same parameters, Bwa only loses under certain circumstances against Gsnap and Novoalign. As Novoalign is a commercial tool and does not have a published method explanation, we excluded it from further consideration. Gsnap, in turn, has an extremely high probability of errors. SV-Bay bases predictions on the read count number; therefore, the false-positive mapping can have a strong impact on the results. Bwa has a relatively low processing speed: almost every other tool considered in the review outperforms it in terms of throughput. Nevertheless, for our purposes, sensitivity is more important than the run time of the algorithm; therefore, we have chosen Bwa for read mapping.

2.3.2 BWA drawbacks for structure variants detection

Although Bwa shows a good mapping percentage, its mapping strategy is not adopted for the structure variants detection task. And it has an important impact on SV-Bay algorithm.

During the alignment process, Bwa and tools with similar mapping strategies take into account the paired-end or mate-pair information. If one of the reads in a fragment cannot be mapped uniquely, the mapping position can be chosen on the basis of the expected insert size (explained in 2.2.4). Insert size is also considered while calculating the mapping quality (MQ) score for the fragment. This approach has both advantages and disadvantages. On the one hand, when insert size distribution is taken into account, a paired-end fragment can be aligned with a high mapping quality even if one of the reads cannot be mapped uniquely. In this case Bwa relies upon the estimated fragment length and the mapping of the other read. If one of the multiple read mappings is located within the expected distance from the other read, Bwa assumes this fragment is mapped correctly. On the other hand, in tumor data, there are fragments coming from SVs. For such fragments, reads can be placed on different chromosomes or the insert size may be far from expected. Therefore, they can be mapped only if both reads are uniquely mappable. In this case, Bwa gives a very low or even zero mapping quality score to the fragment, because the distance between two reads significantly differs from the expected value. This leads to filtering out some possible SVs. For example, in *COLO-829* melanome data, we found out that over 16% of the fragments, in which both reads were mapped uniquely, are given a low mapping quality. The SV-Bay definition 3.1.2 takes into account these "abnormal" fragments. The SV-Bay procedure considering this effect is described in Chapter 3.

Chapter 3

Structural variant detection based on paired-end mapping signatures

Once the fragments are mapped, one can proceed to the SV detection.

As described in the previous chapter, all fragments should have approximately the same length. If the mapping is correct, we expect that after mapping fragments retain this property. But as a result of an SV in the tumor genome, fragments coming from the SV region cannot be mapped as it is expected (with the expected orientation of the reads and the expected insert size). Based on these abnormalities, SVs can be detected. Theoretically all SVs should be supported by a signature, i.e. fragments that are mapped in a special way. For example, after mapping, fragments will have insert size above average for a deletion and below average for an insertion. A full list of signatures is described in chapter 7. Below, we explain the procedure of detection of possible SVs based on abnormal fragment mapping implemented in the SV-Bay algorithm.

3.1 Annotation of normal and abnormal read pairs

All fragments are separated into two groups: normal and abnormal. This separation is based on the orientation, insert size and mapping properties of the read pairs. We assume that abnormal fragments can be related to the break-point junction, i.e., to the SV positions.

We consider as PCR duplicates and discard read pairs with identical or close (up to k bp) start and end positions; k specified by the user.

3.1.1 Detection of normal fragments orientation

As it was described before, paired-end and mate-pairs fragments consist of two reads. For a read mapping two orientations are possible: forward (F) or reverse (R). Thus, there are four kind of possible fragment mapping orientations: forward-forward (FF), reverse-reverse (RR), forward-reverse (FR) and reverse-forward (RF). For one of the most popular sequencing technologies, Illumina, normal orientation can be FR for paired-end data and RF for mate-pair data. It is expected that the reads from all fragments in the library have the same orientation unless they correspond to an SV.

The information on the expected normal orientation can be provided together with the data. When it is not the case, it must be determined. The number of fragments with the correct orientation should significantly exceed the number of fragments with other orientations. There are usually 1-2% of fragments with unexpected orientation in paired-end data. For mate-pairs, data contamination with fragments with unexpected orientation may constitute up to 10% because of a possibly high fraction of singletons.

SV-Bay is able to detect normal orientation. It simply calculates the number of fragments for each orientation and assumes that the orientation of the majority of fragments corresponds to the normal orientation of fragments in the library.

3.1.2 Definition of normal insert size

Once normal orientation of the reads in fragments is detected, SV-Bay proceeds to the evaluation of the normal insert size. To approximate normal insert size and the shape of fragment length distribution, SV-Bay takes into account only fragments with normal orientation that map on the same chromosome. All normal fragments are expected to have similar insert size. Presence of genomic SVs such as deletions and insertions in the constitutional or tumor DNA may change the insert size of the corresponding read pairs. Moreover, even for the fragments not related to the structural variants, the insert size can vary due to cross-ligation of DNA fragments during library preparation or due to mapping errors.

By default, SV-Bay considers the fragment insert size as normal if it is within the 99% of insert size distribution when we consider fragments with insert size shorter than 10 Kb. When the fragment is too long, we consider it as coming from an SV. 10 Kb is the empirical value; this value is based on the knowledge that no sequencer machine can be expected to sequence fragments whose length exceeds 10 Kb, even in the case of mate-pairs where average length could be up to 7 Kb. After the analysis of insert size distribution, we define μ (the median insert size) and σ (standard deviation of insert size Grafarend2006) .

The minimal and maximal insert sizes of a normal read pair are noted

l_{min} and l_{max}, respectively. By default, they correspond to the 99% confidence interval. The parameter 99% can be modified by the user, when the user has an *a priori* knowledge of the data. For example, some datasets can have a heavy tail in insert size distribution. Default parameters are based on the 3-sigma rule.

3.1.3 Formal definition of normal and abnormal fragments

Here we define normal and abnormal fragments to use in further analysis.

Definition 3.1.1 *A fragment (pair of reads) is considered* normal *if:*

1. *both reads are mapped to one chromosome,*
2. *the insert size is within the confidence interval*
3. *the orientation of the reads is normal*
4. *the mapping quality of both reads is greater than or equal to 20.*

When reads within a fragment have expected normal orientation and an insert size within the confidence interval, a mapping quality more than 20 is required to pass filtering: this value guarantees that the fragment is mapped correctly with 99% probability.

Definition 3.1.2 *The fragment (pair of reads) is considered* abnormal *if both reads are uniquely mapped and at least one of the following conditions is satisfied:*

1. *the insert size is out of the confidence interval*
2. *the orientation of the reads is not equal to the normal orientation*
3. *reads are mapped to different chromosomes.*

When a fragment has an unexpected orientation for the reads, or (and) has an insert size not in the confidence interval, or when reads are mapped on different chromosomes, it is used by SV-Bay if both reads in a pair are mapped uniquely.

All fragments not satisfying the above definitions are discarded from further analysis.

3.2 Clustering of abnormal fragments

Most of the abnormal fragments are in fact noise, caused by errors in library preparation and mismapping. Such fragments are not related to structural variants. To circumvent this difficulty, read coverage properties may be used: each base in the genome gives rise to several sequenced fragments. This is also true for the break-point base, which implies that each abnormal read signature should be observed for several fragments.

Therefore, once we got the array of abnormal fragments, the next step is to cluster them in order to provide SV candidates. The clustering algorithm takes into account positions, lengths and orientations of the fragments. It identifies sets of fragments that are close to each other and have the same read orientation. Below we present a two-phase clustering algorithm developed for SV-Bay. This algorithm takes into account all specific features typical for complex structural variants.

3.2.1 Cluster definition

Clustering of fragments is based on two main parameters: the difference of insert sizes and the difference of read coordinates. Given a fragment i, one denotes:

- x_i: the mapping position of the leftmost read beginning;
- y_i: the mapping position of the rightmost read end.

Additionally, the position of the middle point is denoted F_i and the total length of the fragment is denoted $Flen_i$. For any fragment i, parameters F_i and $Flen_i$ satisfy the equations:

$$\begin{aligned} F_i &= \frac{x_i + y_i}{2} \ ; \\ Flen_i &= y_i - x_i \ . \end{aligned}$$

For fragments corresponding to the same genomic adjacency, the maximum possible difference of insert sizes is denoted I_{max}. For all known SVs, this difference is not greater than $2 \cdot l_{max}$, so $I_{max} = 2 \cdot l_{max}$. To illustrate this, we consider two SV classes involving and not involving an inversion. In Figure 3.1, we provide two examples that lead to the maximum and minimum possible differences of insert sizes. Panel (A) shows an *inverted adjacency* (e.g., inversion or inverted translocation). The maximal difference in insert sizes within this SV is $2 \cdot l_{max}$. Panel (B) shows a *direct adjacency* (e.g., deletion, duplication, direct translocation). The maximal difference in insert sizes is $l_{max} - l_{min}$, where l_{min} is the minimal fragment insert size.

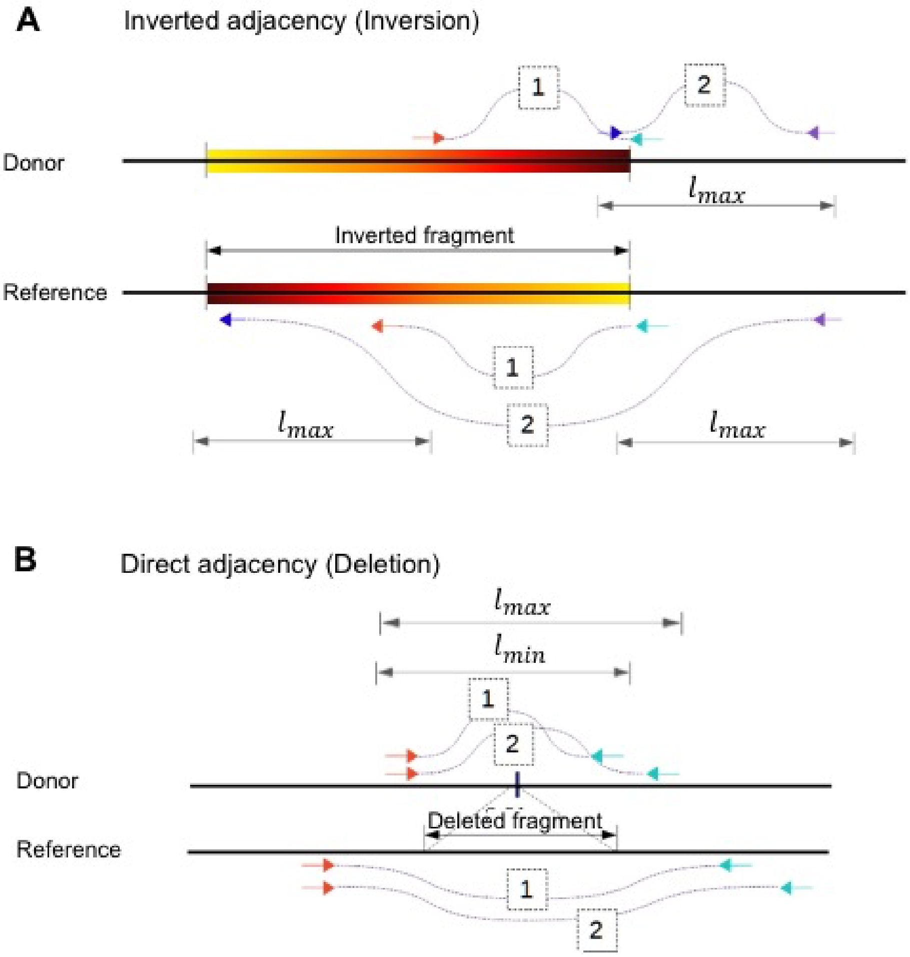


Figure 3.1: (A) Inverted adjacency (Inversion); (B) Direct adjacency (Deletion)

The maximum difference of midpoint positions for two fragments corresponding to the same genomic adjacency is denoted D_{max}. For all known SVs, this difference is not greater than l_{max}, so $D_{max} = l_{max}$. It can be achieved for a direct adjacency, shown in panel (B) of Figure 3.1. In the case of an inverted adjacency, the maximal difference of midpoint positions is $(l_{max} - l_{min})/2$, as shown in panel (A).

The SV-Bay clustering algorithm includes two phases: primary clustering and splitting of large clusters of read pairs when it is needed. On the first step a set of *super-clusters* is produced.

Definition 3.2.1 *A* super-cluster *is a set of abnormal fragments* $(x_i, y_i)_{i=1\cdots n}$ *such that :*

1. *Fragments are sorted by their midpoint position:* $F_1 \leq F_2 \leq \cdots \leq F_n$
2. *Read pairs within all fragments have the same orientation*
3. $\max_{i,j} |Flen_i - Flen_j| \leq I_{max}$
4. $\max_i |F_{i+1} - F_i| \leq D_{max}$
5. *The super-cluster is exhaustive:*

$$\forall (x_t, y_t) \notin S : \nexists i \in \{1 \cdots n\}, \text{ such that } |Flen_t - Flen_i| \leq I_{max} \text{ and } |F_t - F_i| \leq D_{max} \tag{3.1}$$

During the second step, clusters are further divided into smaller parts that bijectively correspond to a genomic adjacency. The resulting clusters are called *links.*

Definition 3.2.2 *A* link *is a set of abnormal fragments* $(x_i, y_i)_{i=1\cdots n}$*, satisfying all* super-cluster *constraints and also with* $|F_n - F_1| \leq D_{max}$*.*

3.2.2 Primary clustering algorithm

The first phase of the clustering algorithm is primary clustering. It is performed separately for the following groups of fragments:

1. Intra-chromosomal fragments of each chromosome. They are divided into four groups with read orientations FF, FR, RF and RR.
2. Inter-chromosomal fragments of each chromosome pair. They are divided into four groups with read orientations FF, FR, RF and RR.

The primary clustering iteratively builds super-clusters. The fragments from the input list are sorted by their midpoint positions. Then, the list is traversed from left to right. Neighbour fragments with a midpoint distance smaller than or equal to D_{max} are added to a super-cluster S. The fragments with the difference in lengths greater than I_{max} are skipped. The iteration step ends if the midpoint distance between current and next fragment is larger than D_{max}. The fragments included in a super-cluster are removed from the input array.

This iteration step is repeated until each fragment is assigned to a super-cluster. Super-clusters with only one fragment are discarded as noise and are not further processed.

The simplified pseudo code is presented below.

```
# Initialize empty array for resulting super-clusters
res_super_clusters = []
# Assume that fragments array is sorted by midpoint position
while fragments is not empty: # Iteration steps loop
    # Initialize array of fragments in current super-cluster
    super_cluster = []
    # Initialize previous_fragment with the first fragment
    previous_fragment = fragments[0]
    # Remove it from the source array and add to
        super_cluster
    fragments.remove(previous_fragment)
    super_cluster.append(previous_fragment)
    # Initialize maximum fragment length in super_cluster
    max_length = previous_fragment.length
    # Initialize minimum fragment length in super_cluster
    min_length = previous_fragment.length
    for fragment in fragments: # Iteration step
        # Check D_max constraint
        if fragment.middle - previous_fragment.middle <=
           D_max:
            # Check I_max constraint
            if frag.length <= min_length + I_max and
               frag_length >= max_length - I_max:
                # I_max constraint passed
                # Update max_length and min_length
                if frag.length > max_length:
                    max_length = frag.length
                if frag.length < min_length:
                    min_length = frag.length
                # Update previous_fragment
                previous_fragment = frag
                # Add frag to super_cluster
                super_cluster.append(frag)
                # and remove it form source array
                fragments.remove(frag)
            else: # I_max constraint not passed
                continue # skip fragment
        else: # D_max constraint not passed
            break # Finish iteration step
    # Check resulting super_cluster length
    if (len(super_cluster) > 1:
        # If it's not noise, add it to res_super_clusters
        res_super_clusters.append(super_cluster)
```

3.2.3 Splitting algorithm

The second phase of the clustering algorithm splits the super-clusters into links with the property $F_n - F_1 \leq D_{max}$.

Let $D = F_n - F_1$ denote the maximal distance between the fragment midpoints of a super-cluster. If D is smaller than or equal to D_{max}, the super-cluster is annotated as a link. It corresponds to a possible novel genomic adjacency and does not need splitting. The remaining super-clusters (with $D > D_{max}$) cannot correspond to a single new genomic adjacency and should be divided into several links.

Splitting of each super-cluster S is performed iteratively. On each iteration, fragment midpoints of S are clustered using the k-means algorithm. The k-means implementation from SCIPY (python library for scientific computing) is used. The parameter k is set to $\frac{D}{D_{max}}$, rounded upward.

The constraint $D \leq D_{max}$ is checked for each resulting cluster. The clusters for which this constraint is satisfied are annotated as links. The fragments included in such clusters are removed from S. If there are still fragments remaining in S, values D and k are re-calculated for them and the k-means iteration is repeated. Otherwise processing of S is finished.

Similarly, to the primary clustering, resulting links are checked to contain more than one fragment. Links containing less than one fragment are discarded as noise. In simulated mate-pairs data only 23% of abnormal fragments are associated with links and the others are discarded as noise, which illustrates the importance of the clustering step.

Chapter 4

Coverage issues

In the previous chapter, we discussed the signature based approach for SV detection and its implementation in the SV-Bay algorithm. SV-Bay separates fragments into normal and abnormal fragments. Abnormal fragments are potentially related to structural variants; clustering these abnormal fragments provides SV candidates.

Another major approach for SV detection is based on the observation of the depth of coverage (DOC) change. Typically, DOC of the genomic region is defined as the average number of times each base of the region has been sequenced (read coverage); also, one can define *fragment* DOC: the average number of sequenced fragments covering each position. Structure variants often result in a change of copy number status around the breakpoint junction, which is reflected in changes in read and fragment DOC. For instance, deleted regions have a relatively low DOC, whereas duplicated regions are characterized by a higher DOC [MWS+11]. Thus, differences in DOC and abnormal positioning of mapped reads often indicate the same genomic abnormality (e.g., a deletion or a tandem duplication).

SV-Bay algorithm considers the DOC change to filter out false SV candidates with a probabilistic Bayesian approach. This approach is further described in chapter 6. To make the probabilistic model accurate, it is crucial to take into account all factors influencing DOC. These factors are discussed in this chapter.

There are three main factors that influence DOC: *GC-content*, *ploidy/-copy number changes* and *region mappability.* They are described in sections 4.1, 4.2 and 4.3 respectively.

4.1 GC-content

The first important factor influencing the depth of coverage is *GC-content.*

Definition 4.1.1 *The* GC-content *of a given region is the proportion of either guanine or cytosine nucleotides in this region.*

Example 4.1.1 *The GC-content of sequence AACCGATGAC is* $\frac{6}{10} = 0.6$.

The fragment amplification rate strongly depends on the GC-content. It is illustrated in Figure 4.1. The top panel shows the number of fragments starting in a sliding window of 5O Kb along the genome; the bottom panel one shows the GC-content calculated using the same sliding window. It is clear that the number of fragments increases in regions with a high GC-content and decreases in regions with a low GC-content. The difference of coverage between GC-rich to GC-poor regions is up to 2-3 times.

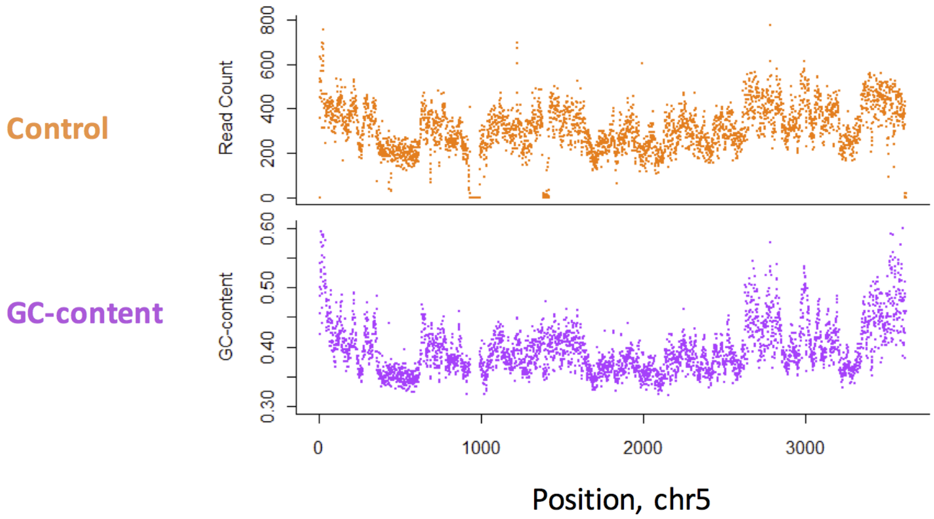


Figure 4.1

Definition 4.1.2 *The* GC-content bias *is the dependence between fragment count (DOC) and GC-content found in high-throughput sequencing assays.*

The GC-content bias is data dependent and may even vary between different experiments of the same type. This phenomenon is particularly strong with the *Illumina Genome Analyzer* technology.

A method to evaluate and correct for the GC-content bias for normal genome was recently proposed in [BS12]. In Section 5.2, we extend it to handle tumor data.

In [BS12] it is assumed that the fragments are sequenced on both ends using the standard *Illumina* procedure. Each fragment is mapped on the reference genome using (Bwa). Fragments are used in this method only if $5'$ read is uniquely mapped (flag $XT : A : U$ of Bwa). This allocation of reads, that excludes the ones mapped to multiple locations, is very sensitive to the comprehensiveness of the reference.

Authors suggest the following algorithm to count the expected number of fragments with the amendment to the GC-content. A *window length* is set. For a given GC-rate γ, let N_γ denote the number of windows in the genome with a GC-rate equal to γ. Let F_γ denote the total number of fragments that start at the beginning of such a window. According to [BS12], the expected number of amplified fragments at the starting position of such a window, denoted μ_γ, is set to

$$\mu_\gamma = \frac{F_\gamma}{N_\gamma} \ . \tag{4.1}$$

Computing the values N_γ and F_γ yields this average value μ_γ. The choice of the window length is extensively discussed in the paper. The conclusion of the authors is that the best choice for the window length is the median of the normal fragment lengths.

4.2 Ploidy and copy number changes

Other factors influencing coverage are ploidy and changes in the copy number.

Definition 4.2.1 Ploidy *is the number of sets of chromosomes in a cell.*

To evaluate the expected read count per one copy of a chromosome, SV-Bay only takes into account regions where the number of copies is equal to the ploidy (the exact value is provided by the user in the initial parameters for the tool). Copy number of each region is provided by the Control-FREEC method [BZB+11, VBTPKBPCJCGSIJLOD12] (Figure 4.2).

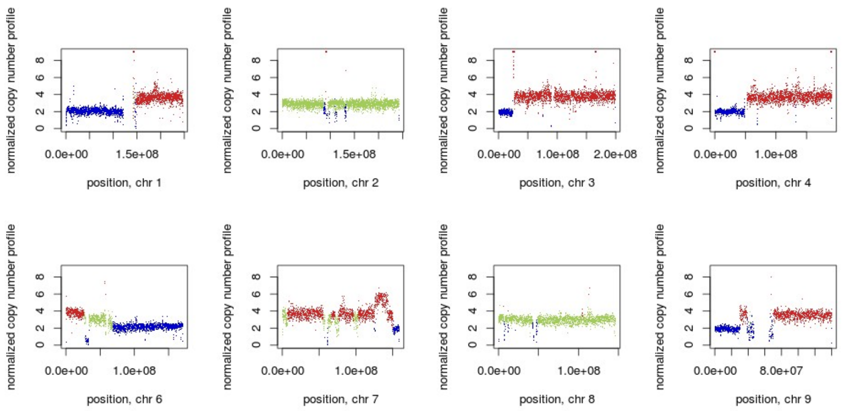


Figure 4.2: Freec output for the COLO-data, shows the variation of the copy number in the same chromosome.

The Control-Freec algorithm consists of several steps. First, it roughly calculates the copy number profile by simply counting reads in large non-overlapping windows. The second step is a normalization of the GC-content profile. This normalization is based on several assumptions:

(i) the main ploidy of the sample, P, is provided;

(ii) the observed read count in P-copy regions (i.e., regions with a copy number equal to P) can be modeled as a polynomial of GC-content;

(iii) the observed read count in a region with altered copy number is linearly proportional to the read-count in P-copy regions;

(iv) the interval of measured GC-contents in the main ploidy regions must include the interval of all measured GC-contents.

The third step is a segmentation of the normalized Copy Number Profile (CNP). Control-Freec uses a Lasso-based algorithm suggested by [HLL08]. The segmentation provided by this algorithm is robust against outliers, which makes it suitable for a segmentation of deep-sequencing CNPs. The last step involves an analysis of segmented profiles. This includes the identification of regions of genomic gain and loss and the prediction of the absolute copy numbers in these regions.

4.3 Mappability

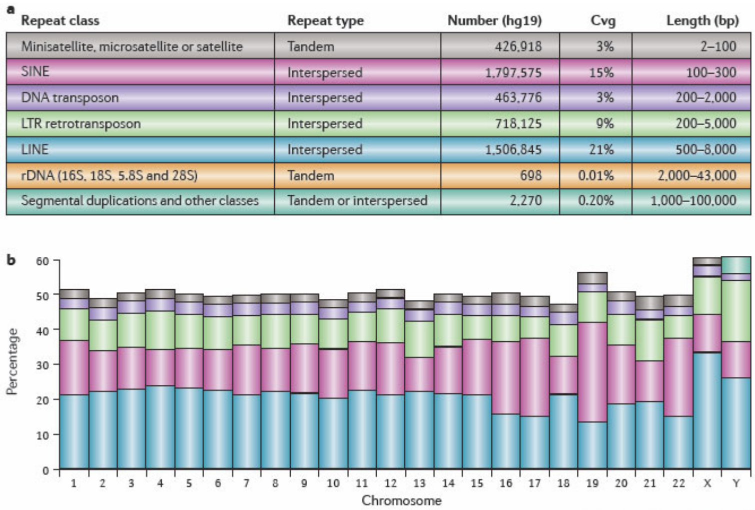
The last factor influencing the coverage is the mappability of the region. This factor is considered below. First, the repetitive structure of human genome is discussed. Then, the relationship between read mappability and coverage is explained. Finally, the mappability of a genomic position is defined and the Gem tool used in SV-Bay pipeline is discussed.

4.3.1 Repeats in human genome

The repetitive structure of the genome complicates read mapping and the assessment of mappability. A list of known types of repeats in the human genome and statistics of their distribution in different chromosomes are provided in figure 4.3.

Average lengths of the repeats are given in [GWNL00]. It is claimed in [TS12] that 50% of the genome consist of the different types of repeats. According to this work, only 18% out of these 50% repeats have a length smaller than 300 bp; all the rest have an average length greater than 500 bp.

Average size of pair-ended *Illumina* fragments varies from 250 to 400 bp. Therefore, with paired-end technology, 32% of read pairs can have ambiguous mapping positions and thus can lead to a wrong SV detection.



Repeat class	Repeat type	Number (hg19)	Cvg	Length (bp)
Minisatellite, microsatellite or satellite	Tandem	426,918	3%	2–100
SINE	Interspersed	1,797,575	15%	100–300
DNA transposon	Interspersed	463,776	3%	200–2,000
LTR retrotransposon	Interspersed	718,125	9%	200–5,000
LINE	Interspersed	1,506,845	21%	500–8,000
rDNA (16S, 18S, 5.8S and 28S)	Tandem	698	0.01%	2,000–43,000
Segmental duplications and other classes	Tandem or interspersed	2,270	0.20%	1,000–100,000

Nature Reviews | Genetics

Figure 4.3: The table in panel **a** shows various named classes of repeat in the human genome, along with their pattern of occurrence (shown as 'repeat type' in the table; this is taken from the RepeatMasker annotation). The number of repeats for each class found in the human genome, along with the percentage of the genome that is covered by the repeat class (Cvg) and the approximate upper and lower bounds on the repeat length (bp). The graph in panel **b** shows the percentage of each chromosome, based on release hg19 of the genome, covered by repetitive DNA as reported by RepeatMasker. The colours of the graph in panel **b** correspond to the colors of the repeat class in the table in panel **a**

Mate-pair data average insert size from 2,500 up to 6,000 bp. According to RepeatMasker Library, db20140131, LINE-1 repeats have an average length of 6,000 bp (17.5% of the genome), Line-2 repeats - 3,000 bp (3.7% of the genome) and LTR repeats - 640 bp (about 9% of the genome). Therefore, mate-pair data can cover almost every type of repeats in the human genome, except long LINE's repeats, rDNA repeats (0.01% of the human genome) and segmental duplications and other classes (around 0.2% of the human genome). However, we should mention that often a unique read mapping is possible even for these repetitive regions as they have undergone many point mutations during the genome evolution.

4.3.2 Reads mapping and DOC

In chapter 2, we discussed that some reads can be mapped ambiguously because of the repeats and sequencing errors. If one of the reads is mapped ambiguously, the aligner sets a low mapping quality for the fragment and this fragment is filtered out (see chapter 3).

In a repetitive region almost all fragments might be discarded, therefore the number of observed fragments on the region may be very low. This effect should be considered when the expected DOC (number of fragments) is calculated for a region. Otherwise, this loss of coverage might be, erroneously, associated with a genomic loss, for example, with a deletion.

The mappability factor is significant: over 50% of the human genome consist of regions repeated exactly or approximately in the other places. Some of the repeats occur thousands of times throughout the genome as described above.

4.3.3 Mappability of a genome position

Definition 2.2.1 measures the confidence of the mapping of an individual read. The position of the read is not considered. It is observed for some genomic regions that all, or almost all, reads from that region cannot be uniquely mapped. This occurs notably if the region is repeated in some other place of the genome. For a specific genome, the percentage of reads that are mapped uniquely mostly depends on the number of mismatches allowed during alignment and the read length. Thus, knowing the read length and the mapping algorithm parameters it is possible to compute the mappability of the whole sequence beforehand.

The approach to this problem is extensively discussed in [DEM$^+$12]. The authors give the following auxiliary definition of K-FREQUENCY:

Definition 4.3.1 *Given some read length k, the* K-FREQUENCY *$F_k(x)$ of a sequence at a given position x corresponds to the number of times the k-mer starting at position x appears in the sequence and in its reverse complement,* *considering as equivalent all the k-mers which differ by less than some predefined alignment score.*

Based on k-frequency the inverse value K-MAPPABILITY is defined:

Definition 4.3.2 *The* K-MAPPABILITY *or* K-UNIQUENESS $T_k(x)$ *is the inverse of the k-frequency:*

$$T_k(x) = \frac{1}{F_k(x)} \tag{4.2}$$

The authors introduce an algorithm to compute the mappability for the reference genome. This algorithm is implemented in the GEM tool. Given the window length k and the number of possible mismatches t, GEM tries to match each possible k-mer at some other position with up to t mismatches using a time- and memory-efficient algorithm and thus calculate the k-mappability $T_k(x)$. If t is equal to zero, the exact mappability is calculated, otherwise a good approximation is produced.

The output of the algorithm is a string of flags representing mappability equal to one for positions for which the k-mer has no other matches (mappable positions) and lower mappability for the other positions. The length of this output string is equal to the reference (chromosome or whole human genome) length.

In SV-BAY pipeline the output of GEM tool is used. The parameter k is set to the read length, and t is set to 3 (equal to the respective BWA parameter). As explained before, to estimate the number of fragments for a region for each position the SV-BAY algorithm should consider if the read is expected to be mapped uniquely on this position or not. For this reason, we use the DISCRETE MAPPABILITY value M_i instead of the mappability produced by GEM.

Definition 4.3.3 *Given the read length k, the* DISCRETE MAPPABILITY M_i *of the position i in reference genome equals 1 if k-mappability $T_k(i)$ equals 1 and 0 otherwise.*

Chapter 5

Abnormal and flanking regions

The SV-Bay probabilistic approach to SV validation is based on the estimation of the depth of coverage (Doc) in special genomic areas, related to each SV candidate. Such areas, called flanking regions and abnormal region, are introduced in section 5.1.

In section 5.2, the methods to estimate the expected number of fragments starting in a genomic region are described. The estimate is given in Section 5.2.1 for the flanking regions, in Section 5.2.3 for the abnormal regions. These estimates are based on the factors described in Chapter 4.

5.1 Definition of flanking regions and abnormal regions

Most novel genomic adjacencies have two breakpoints in the reference genome, with the exception for small insertions and mirror duplications. Without loss of generality, we further assume that there are two breakpoints per link.

As a cluster spans the breakpoints, a Doc change is expected in the surrounding genomic area. The basic idea of the SV-Bay probabilistic model is to compare the expected and observed number of two types of fragments:̃

- Normal fragments in surrounding areas not spanning breakpoint.
- Abnormal fragments spanning breakpoint.

The so-called *flanking regions* are introduced in 5.1.1. These regions are areas surrounding a possible SV, which can only contain normal fragments that do not span the breakpoint. To denote them we use an auxiliary definitions of *safety regions* : areas which can possibly contain the breakpoint and starts of fragments spanning it (considering that the exact breakpoint position is not known). An abnormal region is an area where abnormal fragments, possibly spanning a breakpoint, can start. These regions are introduced in 5.1.2.

5.1.1 Flanking regions

First, areas where the breakpoints and fragments spanning them may fall are defined. Such areas are called *safety intervals*:

Definition 5.1.1 *For any link S with reads leftmost positions* $\{(x_i, y_i)\}_{i=\{1..,n\}}$, *the* left *and* right safety intervals S_x *and* S_y *are defined as:*

$$\begin{aligned} S_x &= [min(x_i) - l_{max}, max(x_i) + l_{max}] \ ; && (5.1)\\ S_y &= [min(y_i) - l_{max}, max(y_i) + l_{max}] \ . && (5.2) \end{aligned}$$

This definition guarantees that both breakpoints are included in the safety intervals. As a consequence, any read pair starting outof the safety intervals belongs to a normal fragment, i.e., a fragment that does not contain a breakpoint junction. An evaluation of the most likely positions for breakpoints within these intervals is described later in chapter 6.

Next, we define flanking regions. These regions should not include the interval around the breakpoint itself, where we expect to observe a gap in normal Doc [SOP+12]. Moreover, they should not overlap any structural variant that could affect the number of normal read pairs within this region.

Definition 5.1.2 *Given a safety interval for a breakpoint, an upstream (respectively downstream) flanking region is the largest region located upstream (respectively downstream) of that interval, that does not intersect any safety interval of any other breakpoint.*

Closely located links may have overlapping safety intervals. In this case, the corresponding flanking regions are empty. Also, flanking regions are not allowed to span centromeric regions and long unassembled poly-N regions.

SV-Bay formally divides the regions around each link into four flanking regions. We denote these regions (A_1, A_2) and (B_1, B_2) for the left-most and right-most breakpoints, respectively.5.1

(A) *General case*: Flanking regions are defined by

- the safety intervals of the link extremities: purple links that define right boundaries of regions A_1 and B_1 and left boundaries of regions A_2 and B_2;
- safety intervals of intervening links: grey links that define left boundary of B_1 and right boundaries of A_2 and B_2;

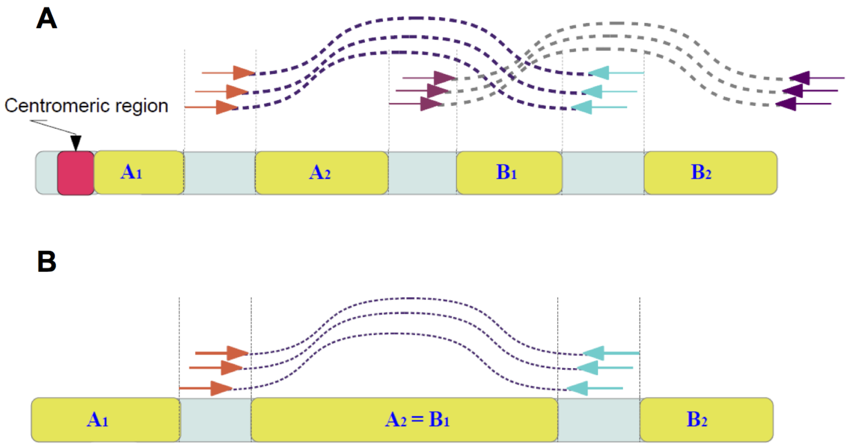


Figure 5.1: Flanking regions. (A) General case. (B) Special case when two flanking regions, A_2 and B_1 coincide.

- large regions of unassembled or unmappable genome: red region defines left boundary of region A_1. The set of large unassembled or unmappable regions is provided by the user or can be calculated with a script provided in the Sv-Bay package.

(B) *Special case*: two flanking regions, A_2 and B_1, coincide. The region between two breakpoints does not contain other candidate SVs or unmappable regions. This case occurs notably for small and medium size deletions.

5.1.2 Abnormal region

Definition 5.1.3 *Assume a breakpoint occurs at position x in the reference genome. Let $\Gamma(x)$ denote the set of genomic positions where a fragment spanning x in the tumor genome can start. This region is called the* abnormal region *for x.*

The length of an abnormal region is always equal to l_{max}: this ensures that even the longest possible fragment would start within this region. The exact coordinates depend on the orientation of fragments in the corresponding link and of the expected orientation If the orientation of the first read in the fragments is the expected orientation of the first read in a normal pair (for example, F for paired end FR reads), then the abnormal region is :

$$[x - l_{max}; x] \ ;$$

otherwise, it is

$$[x; x + l_{max}] \ .$$

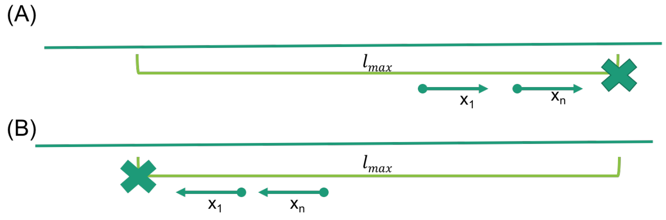


Figure 5.2: Abnormal region coordinates depending on the fragments orientation. Expected orientation for this example is equal to FR. First reads from each pair are shown ($x_{i..n}$): (A) reads have the expected orientation, (B) reads have the unexpected orientation.

This is depicted in Figure 5.2.

In practice, the exact position of the breakpoint is not known from sequencing data. A procedure in SV-Bay, described in 6.3, computes the position that is the most likely.

5.2 Expected number of fragments in a genomic region

For SV-Bay algorithm, we introduce a special definition of the depth of coverage for a genomic region:

Definition 5.2.1 *The* depth of coverage *of a genomic region is the number of fragments starting on this region.*

To evaluate the most probable number of copies involved in a possible SV, SV-Bay relies on the expected and observed number of fragments starting in flanking and abnormal regions.

In this section, estimations of the expected number of fragments starting in the flanking or abnormal regions are derived. An auxiliary method considering GC-content to approximate the number of fragments starting on a specific position is also provided.

5.2.1 Expected number of fragments for a flanking region

As flanking regions are intentionally chosen not to contain fragments overlapping the breakpoint, we estimate the expected number of normal fragments.

Let I be one of the flanking regions. Let us assume that the copy number, noted α, is constant in that region. For a diploid region, α is equal to 2. For any position i in I, let λ_i denote the number of fragments that originate at this position on one copy in the *donor* genome. Let O_I denote the number of reads aligned on the interval I in the *reference* genome.

Let us consider at first the simplified situation where the distribution of the random variable λ_i is independent of the position i and does not depend on the GC-content in the donor genome. Let λ denote the expectation of the random variable λ_i.

The number of reads that are aligned on interval I in the reference genome can be approximated as

$$O_I \sim \alpha \sum_{i \in I} M_i \cdot \lambda_i \ \ , \tag{5.3}$$

where M_i is the *mappability* at position i.

Let $E(O_I)$ be the expectation of the number of mapped reads on a large number of sequencing experiments. Equation 5.3 translates into

$$E(O_I) = \alpha \sum_{i \in I} M_i \cdot \lambda \ \ . \tag{5.4}$$

On a large interval I, $E(O_I)$ can be approximated by the observation O_I; this yields the statistical approximation for λ

$$\hat{\lambda} = \frac{1}{\alpha} \cdot \frac{O_I}{\sum_{i=1}^{L} M_i} \ \ . \tag{5.5}$$

Let us turn now to the more general case where the distribution of λ_i depends on the GC-content. Following the definitions given in 5.2.2, one denotes γ_i the GC-content at a given position i. We assume that λ_i follows a Poisson distribution with parameter $\lambda(\gamma_i)$. In particular, its expectation is $\lambda(\gamma_i)$.

Equation (5.3) now translates into

$$E(O_I) = \alpha \sum_{i \in I} M_i \cdot \lambda(\gamma_i) \ \ . \tag{5.6}$$

To account for possibly mismapped reads in homozygous deletion regions, we modify formula 5.3 for $\alpha = 0$:

$$E(O_I) = I_L \cdot N_{abnormal}/L \tag{5.7}$$

Here I_L is the length of the region I, L the genome length, and $N_{abnormal}$ the total number of abnormal read pairs, which approximates the number of incorrectly mapped read pairs in a given experiment.

5.2.2 Estimation of the number of fragments starting on a position considering GC-content

Values $\lambda(\gamma)$ for all possible values γ of the GC-content can be evaluated on the basis given in [BS12] and described in Chapter 4. Following the observation by [BS12] that the best window size for the GC-content bias correction corresponds to the average fragment length, the GC-content of a given position is evaluated within a window of length μ.

The number of copies in the tumor (donor) may vary along the genome. Therefore, to evaluate $\lambda(\gamma)$, we select positions coming from copy neutral regions in the tumor genome, i.e., regions where the copy number α_i is constant and equal to the main ploidy of the tumor genome. The selection of these positions is based on the output of Control-Freec [VBTPKBPCJCGSIJLOD12, BZB$^+$11] included into the SV-Bay pipeline.

Evaluation of $\lambda(x)$ is implemented on a random subset of genomic positions with a precision of two decimal places. SV-Bay chooses a random set of K windows where K is a large number that allows to have more than zero observations for each range of GC-content, and calculates the first approximation $\lambda_0(x)$ using these windows. The default value for K is 1 million.

At the next step, the second subset of K different windows is chosen and the second approximation, $\lambda_1(x)$, is computed on these $2 \cdot K$ windows. This step is repeated until a termination criterion is satisfied: for any x in $[0:100]$, $|\lambda_0(x) - \lambda_1(x)|$ is smaller than a threshold defaulted to $10^{(-2)}$. The total number of steps depends of the values of K. In this work, the default value of $K = 10^6$ and the number of steps is typically 2.

5.2.3 Expected number of fragments for the abnormal region

For each abnormal region we need to estimate the expected number of abnormal fragments overlapping the corresponding breakpoint. To estimate the expected number of normal fragments for flanking regions it was enough to consider the starting position of the first read for each read pair. For abnormal fragments the mappability of the ending position of the mate-pair should also be considered: SV-Bay algorithm discards read pairs if at least one of the two reads is not uniquely mappable. In case where a fragment is discarded by the algorithm, it should not be counted when calculating the expected number of abnormal fragments.

Let i be some position of the abnormal region. If a read aligns to this position, the ending position of the mate-pair may vary, as the insert size is not constant. This leads to the definition of the *extended mappability.*

Definition 5.2.2 *In an abnormal region, the* extended mappability *of a position* i *is*

$$\bar{M_i} = M_i \sum_{j=i+l_{min}}^{i+l_{max}} M_{i+j} \cdot p(Flen > i+j), \tag{5.8}$$

where p(Flen¿i+j) is the probability distribution of the insert size.

Computing formula (5.8) is time-consuming, so we provide an approximation. As explained in chapter 3, the observed insert size distribution is approximated by a normal distribution with mean μ and variance σ. Fragments in the tails of the normal distribution are discarded when separating normal and abnormal fragments. For a sake of simplicity and to avoid time-consuming calculations, for the remaining fragments the normal distribution can be further approximated by the *uniform distribution* with boundaries $[\mu - c; \mu + c]$.

The boundaries are chosen to make the variance and mean values of the uniform distribution equal to μ and σ respectively. The corresponding segment $[\bar{l}_{min}, \bar{l}_{max}]$ is the region where we expect to map the rightmost mate of the leftmost read in a pair. It is slightly shrank compared to the segment $[l_{min}, l_{max}]$ defined from the normal distribution (Figure 5.3).

According to the general properties of the uniform distribution, the bounds of the segment satisfy the following:

$$\begin{aligned} \mu &= \frac{1}{2}(\bar{l}_{min} + \bar{l}_{max}) \\ \sigma &= \frac{1}{12}(\bar{l}_{max} - \bar{l}_{min})^2 \end{aligned}$$

Equivalently,

$$\begin{aligned} \bar{l}_{max} &= \mu + \sigma\sqrt{3} \ ; \\ \bar{l}_{min} &= \mu - \sigma\sqrt{3} \ . \end{aligned}$$

Assuming that the insert size is uniformly distributed with a mean μ and a variance σ, the *extended mappability* of a position i is

$$\bar{M}_i = M_i \frac{1}{2\sigma\sqrt{3}} \left(\sum_{j-i=\mu-\sigma\sqrt{3}}^{\mu+\sigma\sqrt{3}} M_j \right) \ . \tag{5.9}$$

The redefined mappability $\bar{M}_i$ allows to calculate the expected number of fragments mapped per position.

Proposition 5.2.1 *The expectation of the number of fragments mapped on a position with a given GC-content x is denoted $\bar{\lambda}(x)$ and satisfies*

$$\bar{\lambda}(x) = \frac{\sum_{i=1}^{L} \bar{M}_i \cdot \bar{O}_i \cdot 1_{\gamma_i = x}}{\sum_{i=1}^{L} \alpha_i \bar{M}_i \cdot 1_{\gamma_i = x}}, \tag{5.10}$$

where L is the genome length, α_i is the number of genomic copies for position i, $\bar{O}_i$ is the observed number of normal read pairs mapped to position i, and $1_{\gamma_i = x}$ the indicator that GC-content at position i is equal to x.

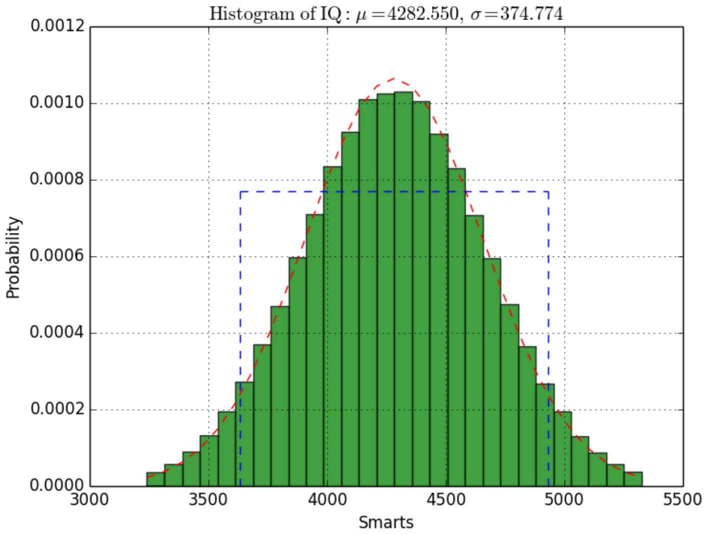


Figure 5.3: The green smarts are the insert size distribution; red dotted line is the approximation by normal distribution; blue dotted line is the approximation by a uniform distribution.

In practice, to evaluate $\lambda(x)$ we do not consider all genomic positions. Instead, we use a large enough random subset so that we can evaluate $\lambda(x)$ up to the third decimal place. For the tumor genome, we select positions coming from copy neutral regions, i.e., regions with copy number equal to the main ploidy of the tumor genome. The selection of these regions is based on the output of Control-FREEC [BZB+11, VBTPKBPCJCGSIJLOD12] included in the SV-Bay pipeline.

Proof: The total count of couples (i, j) of mappable positions, weighted with the probability of an insert size $(j - i)$, with the additional constraint that the GC-content γ_i is equal to x is

$$\sum_{i=1}^{L} \bar{M}_i \cdot 1_{\gamma_i = x} \quad .$$

The expectation of the total number of fragments in such a position is

$$\lambda(x) \sum_{i=1}^{L} \bar{M}_i \cdot \bar{O}_i \cdot 1_{\gamma_i = x}$$

The total number of fragments in the donor that are amplified and mapped to the reference genome is

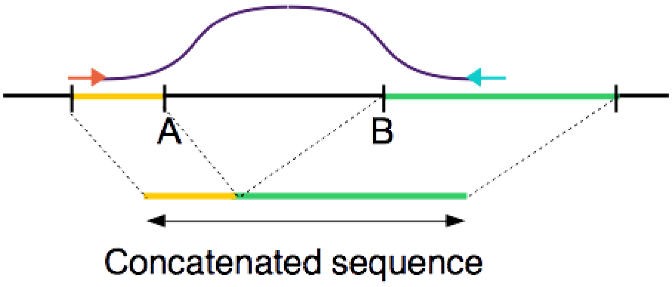


Figure 5.4: A and B show first and second breakpoint positions respectively. To evaluate the GC-content on the abnormal region, SV-Bay creates the concatenated sequence $\tilde{S}$ which approximates the sequence appearing in the tumor genome.

$$\sum_{i=1}^{L} \bar{M}_i \cdot \bar{O}_i \cdot 1_{\gamma_i = x}$$

Let $\tilde{S}$ be a sequence spanning a breakpoint. Without loss of generality, we can assume that it is aligned in the tumor genome to a sequence upstream of a point x_A and to a sequence downstream of a point x_B. On this sequence SV-Bay calculates the GC-content for the current position.

For a breakpoint junction connecting chromosomes A and B, we can now evaluate the expected number of abnormal fragments spanning the breakpoints x_A and x_B on chromosomes A and B, respectively. Without loss of generality, we assume that the junction connects a region upstream to x_A to a region downstream of x_B.

Then, the expected number of abnormal fragments spanning the breakpoints E_{x_A,x_B} can be calculated as:

$$E_{x_A,x_B,\gamma>0} = \sum_{i=x_A-l_{max}}^{x_A} \bar{M}_i(x_A,x_B) \cdot \gamma \cdot \bar{\lambda}(GC(i,x_A,x_B)) \cdot p(InsertSize \geq x_A - i),$$

Theoretically, it possible to observe several closely located abnormal read pairs that do not correspond to any SV. This may happen due to mismapping or DNA fragment fusion during the library preparation. To account for such a possibility in our Bayesian model, we need to estimate the probability to get a random cluster of abnormal read pairs. The expected number of read pairs located just by chance at a distance smaller than l_{max} is approximated as $E_{x_A,x_B,\gamma=0}$:

$$E_{x_A,x_B,\gamma=0} = l_{max} \cdot \frac{N_{abnormal}}{L} \quad , \tag{5.11}$$

where $N_{abnormal}$ is the total number of abnormal read pairs.

Chapter 6

Bayesian models

In Chapter 3, we described the separation of normal and abnormal fragments and abnormal fragments clustering. The result of the clustering process is a collection of fragments sets called links. Each link possibly corresponds to an SV.

The next step is the validation of these SV candidates. Our goal is to check whether a given link is actually related to a real SV. In which case, the number of genomic copies involved in the genomic adjacency should also be determined. A Bayesian approach used to solve this problem is described further.

6.1 Model definition

To validate a link with a Bayesian approach we introduce a *model* M for the link properties that is a set of five parameters: $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma$.

The four parameters $\alpha_1, \alpha_2, \beta_1, \beta_2$ represent the number of copies in flanking regions A_1, A_2, B_1, B_2. The last one, γ, represents the number of copies involved in the structural variant. For most SV types, e.g., duplications and deletions, the number of copies involved is the number of gained or lost copies. For other types, e.g., inversions, it is the total number of copies on the affected region.

Some flanking regions may be empty, and/or regions A_2 and B_1 may coincide. The number of parameters in the model is reduced accordingly. This occurs, for example, for a short deletion.

The parameters of the model satisfy the following *constraints*:

$$\alpha_1 = \alpha_2 \pm \gamma; \beta_2 = \beta_1 \pm \gamma. \tag{6.1}$$

The sign before γ depends on the orientation of reads in the corresponding link. If the reads have an expected orientation, then sign before γ is

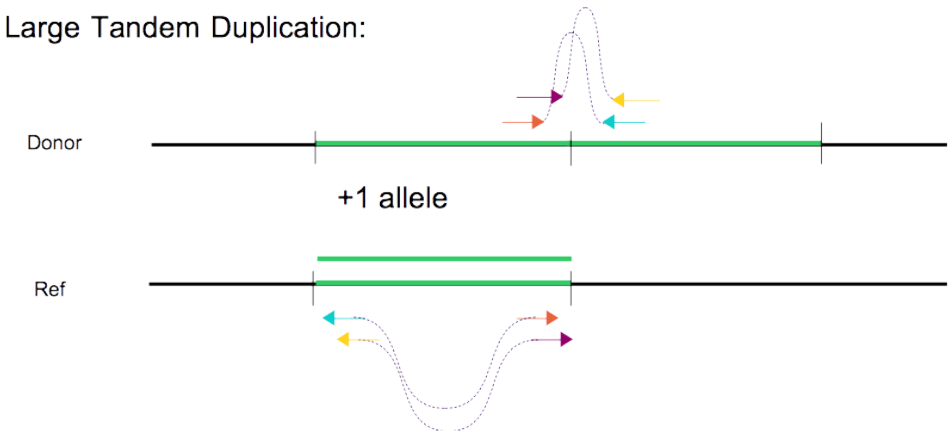


Figure 6.1: Large tandem duplication. One genomic copy is gained in-between two breakpoint positions.

minus. If the orientation of one or both reads is not expected, the respective signs change.

As an example, let us take a large tandem duplication. In this SV type, a second copy of a region is presented just after or before the original region, which causes a gain of genomic copies. After mapping to the reference genome, fragments coming from the SV junction have the inverted orientation RF (for Illumina paired-end reads). In this case, both reads have unexpected orientation, thus formulas 6.1 will be the following:

$$\alpha_1 = \alpha_2 - \gamma; \beta_2 = \beta_1 - \gamma. \tag{6.2}$$

It is explained by the gain of fragments in-between two breakpoints, illustrated in Figure 6.1.

Another example is deletion. SV deletion is characterised by a loss of a part of the genome and thus by loss in copy number. Fragments corresponding to this signature have the expected orientation of the reads and length longer than the longest normal fragment. Thus according to the rules presented above both signs before γ would be plus and the formula 6.1 should be the following:

$$\alpha_1 = \alpha_2 + \gamma; \beta_2 = \beta_1 + \gamma. \tag{6.3}$$

In this case α_1 and β_2 is greater then α_2 and β_2 which corresponds to a loss in copy number between the two breakpoints.

6.2 Bayesian approach

6.2.1 Conditional probability of a model

The aim of our Bayesian approach is to match observations, i.e., the number of fragments in the flanking regions and the abnormal region, with a model $M(\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma)$.

Observations are formalized as a 5-uple $\Delta = (n_{A_1}, n_{A_2}, n_{B_1}, n_{B_2}, n_\Gamma)$. The observed values $n_{A_1}, n_{A_2}, n_{B_1}, n_{B_2}$ are the observed number of normal fragments mapped to the flanking regions A_1, A_2, B_1, B_2, respectively. The value n_Γ is the observed number of abnormal fragments mapped in the abnormal region; this is equal to the number of abnormal fragments in the link.

According to Bayes' rule, the probability of a model M' given observed data Δ is:

$$P(M'|\Delta) = \frac{P(\Delta|M')P(M')}{\sum_M P(\Delta|M)P(M)} \tag{6.4}$$

To determine the most probable model we need to define all possible models M and to calculate, for each model, the corresponding probabilities $P(\Delta|M)$ and $P(M)$. The denominator of the formula 6.4 is constant and so can be ignored when choosing the most probable model. Further we describe how $P(M')$ and $P(\Delta|M')$ are calculated and how the set of models to be checked is chosen.

6.2.2 A priory probability of a model

We assume *a priori* probability $P(M^{\gamma>0})$ of every model M where $\gamma > 0$ to be identical. We expect $P(M^{\gamma>0})$ to be lower than $P(M^{\gamma=0})$, as we suppose that there are less links corresponding to real SVs than to read mismappings and artefacts in library preparation.

To estimate these a priori probabilities we introduce a user-defined parameter, E_{SV}, that represents the expected number of true SVs in the dataset. Then, denoting N_{links} the total number of links, the probabilities $P(M^{\gamma>0})$ and $P(M^{\gamma=0})$ are calculated as follows:

$$P(M^{\gamma>0}) \quad = \quad (min(\frac{E_{SV}}{N_{links}}, 1) \ ; P(M^{\gamma=0}) = 1 - P(M^{\gamma>0}) \ .$$

6.2.3 Factorization of the conditional probability

In the general case, when A_1, A_2, B_1, B_2 are not empty and do not overlap, the conditional probability $P(\Delta|M)$ can be factorized as follows:

$$P(\Delta|M(\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma)) = P(n_{A_1}|\alpha_1) \cdot P(n_{A_2}|\alpha_2) \cdot P(n_{B_1}|\beta_1) \cdot P(n_{B_2}|\beta_2) \cdot P(n_\Gamma|\gamma). \tag{6.5}$$

To calculate these probabilities, we assume that the number of fragments follows a Poisson distribution with mean equal to the expected number of fragments per region. This mean is calculated by formula (5.6) for normal fragments in flanking regions A_1, A_2, B_1, B_2, and by formulas (5.11) and (5.11) for abnormal fragments in the abnormal region.

$$P(n_{A_1}|\alpha_1) = Pois(\lambda_{A_1}(\alpha_1); n_{A_1}) \ ; \tag{6.6}$$
$$P(n_{A_2}|\alpha_2) = Pois(\lambda_{A_2}(\alpha_2); n_{A_2}) \ ; \tag{6.7}$$
$$P(n_{B_1}|\beta_1) = Pois(\lambda_{B_1}(\beta_1); n_{B_1}) \ ; \tag{6.8}$$
$$P(n_{B_2}|\beta_2) = Pois(\lambda_{B_2}(\beta_2); n_{A_2}) \ ; \tag{6.9}$$
$$P(n_\Gamma|\gamma) = Pois(E_{x_A^{break}, x_B^{break}, \gamma}; n_\Gamma) \quad . \tag{6.10}$$

Here $Pois(\lambda; k) = \lambda^k \cdot \frac{e^{-\lambda}}{k!}$ is the probability that the random variable has value k for a Poisson distribution with mean λ.

6.2.4 Set of models to test

Since the total number of possible models to test is infinite, in SV-Bay algorithm we limit the set of models, considering only the most plausible ones.

First, SV-Bay defines a rough approximation for the number of copies in each flanking region. For each region, the associated model parameter is initialized with the observed number of fragments divided by the expected number of fragments for one genomic copy in this region. For example, for the flanking region A_1 the initial α_1 value, denoted $\alpha_1^{[0]}$, is calculated as follows:

$$\alpha_1^{[0]} = round(\frac{n_{A_1}}{E(O_{A_1})}), \tag{6.11}$$

The expected number of fragments for one genomic copy is calculated using formula 5.6.

Then SV-Bay calculates the probability $P(n_{A_1}|\alpha_1)$ for each *positive* value α_1 in the set $[\alpha_1^{[0]}-3, \cdots, \alpha_1^{[0]}+3]$. If $\alpha_1^{[0]}$ is not the global maximum for $P(n_{A_1}|\alpha_1)$, then $\alpha_1^{[0]}$ is replaced by the new global maximum. SV-Bay repeats this operation until $\alpha_1^{[0]}$ maximizes the function $P(n_{A_1}|\alpha_1)$. After the maximum is achieved, the set $[\alpha_1^{[0]} - 3, \cdots, \alpha_1^{[0]} + 3]$ is used on the following steps. The same algorithm is applied for the other flanking regions.

The initial γ value $\gamma^{[0]}$ is calculated using the same approach. It is equal to the closet integer to the number of the abnormal fragments in a link divided by the expected number of abnormal fragments in the abnormal region expected for one allele. Like for α and β, a set $[\gamma^{[0]}3, \cdots, \gamma^{[0]} + 3]$ is built and the global maximum is adjusted for positive γ.

Once, sets of initial values for all model parameters have been calculated, SV-Bay discards the values which do not fit the constraints 6.1. To do so, for each γ from the initial set, all possible pairs (α_1, α_2) and (β_1, β_2) from the initial sets are checked. Only the pairs which fit the constraints are added to the set of models to test.

6.2.5 Model choice

After the set of models to test is created, SV-Bay calculates conditional probability of each model given the observed data using formula 6.4.

By the end of this step, for each link SV-Bay detects the model that explains the observed fragment numbers in abnormal and flanking regions with the highest likelihood. Each validated link is annotated with a value γ and a model probability. If the number of copies involved in the candidate adjacency is evaluated as 0, the candidate link is considered false positive and discarded.

6.3 Evaluation of the breakpoint position

In equations (6.6 - 6.10), it is assumed that the exact breakpoint position is known. In practice, several possible breakpoint positions are checked and the one that provides the highest likelihood is chosen.

For each candidate novel genomic adjacency S with read start positions $\{(x_i, y_i)\}_{i=1\cdots n}$, SV-Bay evaluates the most likely positions of the two breakpoints x_A^{break}, x_B^{break}. As above, without loss of generality, we assume that a junction connects a region upstream to x_A^{break} and a region downstream of x_B^{break}. Then x_A^{break} and x_B^{break} may only lay within intervals C_x^- and C_y^+:

$$C_x^- = [max(x_i), (min(x_i) + l_{max}]; \tag{6.12}$$

$$C_y^+ = [max(y_i) - l_{max}, (min(y_i)]; \tag{6.13}$$

We also assume the following dependency between positions of the two breakpoints given the positions of read pairs in the link:

$$x_A^{break} - x_B^{break} = \sum_{i=1}^{n} \frac{y_i - x_i}{n} - \mu \tag{6.14}$$

For each set of parameters (α_1, α_2, β_1, β_2, γ), we find a set of two breakpoint positions x_A^{break}, x_B^{break} that maximizes the model probability and satisfies conditions. It can be achieved by considering all pairs of positions from regions 6.12 and 6.13 matching the constraint 6.14. For each considered pair, the conditional probability 6.4 should be calculated. The two points maximizing the conditional probability should be chosen.

In practice, calculation of 6.4 for each positions pair is very time-consuming, so only a small evenly distributed subset is checked. By default, it includes ten point pairs. As shown in chapter 9, this approach gives a reasonably good breakpoint resolution accuracy.

Chapter 7

SV types and assembly workflow

7.1 Structural Variant types

Rapid development of high-throughput sequencing technologies provides an opportunity to improve identification of somatic rearrangements in cancer genome. So far, a lot of different types of structural alterations are studied. Nevertheless, no comprehensive catalogs of somatic structural variations exist. It partially arises from the fact that, due to sequencing methods constraints, a given SV detection tool can hardly address all known SV types for a given data set. We fill this gap below (see [IJLB$^+$15]). The variations considered are extracted from the literature [YLG$^+$13, CLBM$^+$13] or from empirical observations on cancer data (Curie Institute). As explained in Chapter 3, each structural variant can be associated with a signature. SV-Bay clusters fragments with common signatures. For each cluster (link), the type of the corresponding SV can be determined based on the signature. Some of the links are independent and some are related to the same SV. On this basis two classes of structural variants, *simple* and *complex*, can be distinguished. They are presented further. For each type of SV, the set of constraints to be satisfied is specified and a figure is given. For some well-known types, the examples of related diseases are also provided. By convention, the expected fragment orientation is assumed to be forward-reverse (F-R).

7.1.1 Simple structural variations

When there is only one link associated with the specific SV, such structural variant is called *simple*. In this section the following simple SVs are described: small insertion, deletion, large/small duplication, unbalanced translocation with/without inversion and mirror duplication. The signature of a simple SV includes:

1. *Reads orientation*: the orientation of the two reads of each fragment of the related link can be either forward-forward, reverse-reverse, forward-reverse or reverse-forward. These orientations are commonly abbreviated as FF, RR, FR and RF.

2. *Fragment insert size*: each fragment of the related link can be mapped to the reference genome with an insert size longer or shorter than respectively maximum or minimum normal fragment length.

3. Inter/intra chromosomal location: the SV can be either intra-chromosomal (when the two reads of each fragment of the related link are mapped on the same chromosome) or inter-chromosomal (when the two reads are mapped on different chromosomes).

Small insertion and deletion First discovered structural variants were small insertion and deletion. These common types of SVs are characterized by insert size of mapped fragments respectively smaller or larger than expected. The signatures of small insertion and deletion are depicted in the left and right panels of Figure 7.1. Small indels are usually not larger than 1-2 median insert sizes of the fragment. A larger inserted region usually origins from another DNA location, which serves as a template; in this case several links form one complex SV called linking re-insertion, which is described further.

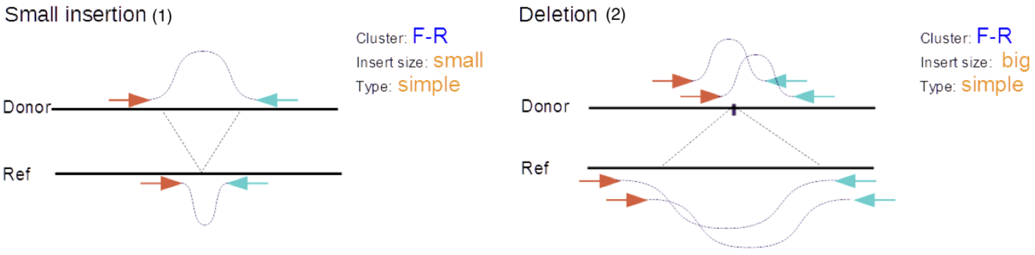


Figure 7.1: (1)Small insertion; (2) Deletion

Unbalanced translocation This abnormality, that is common for cancer diseases, is caused by a rearrangement of parts between nonhomologous chromosomes. Several forms of cancer are caused by somatic translocations; this has been described mainly in leukemia (acute myelogenous leukemia and chronic myelogenous leukemia). Translocations have also been described in solid malignancies such as Ewing's sarcoma. When a new part from a different chromosome is inserted in inverse orientation, this is an *unbalanced*

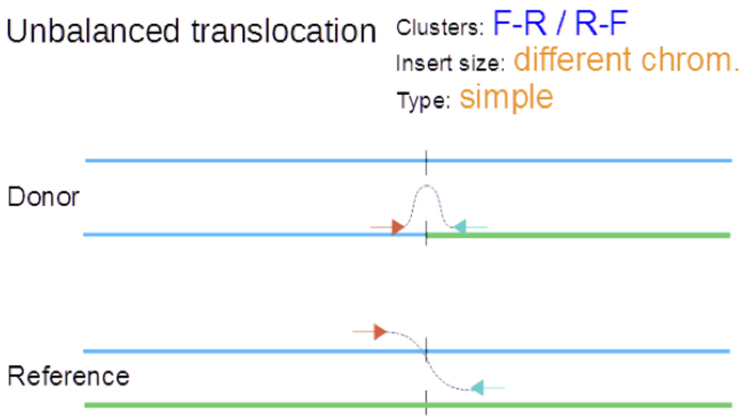


Figure 7.2: Unbalanced translocation

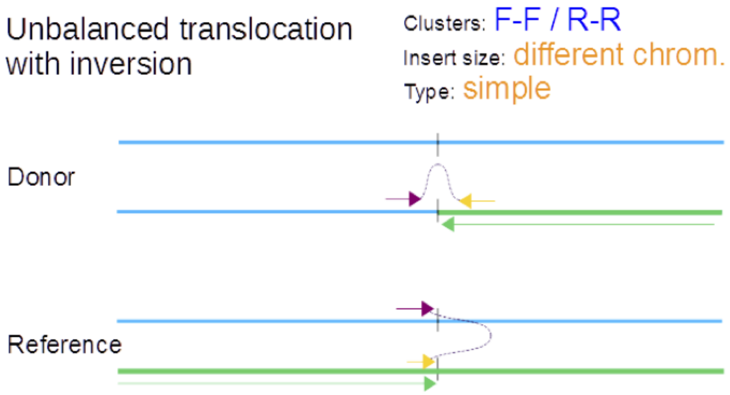


Figure 7.3: Unbalanced translocation with inversion

translocation with inversion. Otherwise, it is an *unbalanced translocation without inversion*. It is depicted in figures 7.2 and 7.3.

A single breakpoint is associated with these SV types; therefore, they are supported by a single link. In both cases, reads are located on different chromosomes.

Duplication A tandem duplication of a genomic region creates an extra copy of the corresponding region. Such duplications happen in many human genetic disorders. For example, Charcot-Marie-Tooth disease type 1A might be caused by a duplication of the gene encoding peripheral myelin protein 22 (PMP22) on chromosome 17 [Lup98]. Gene duplications and increases of gene copy numbers can also be related to cancer. They can be detected in transcriptomic data or using copy number variation arrays. For example, the chromosomal region 12q13-q14 is strikingly amplified in many sarcomas. This chromosomal region encodes a binding protein called MDM2, which is known to bind to a tumor suppressor called p53. When MDM2 is amplified, it prevents p53 from regulating cell growth and contributes to tumor formation [OKM+92]. Depending on the size of duplicated region, different

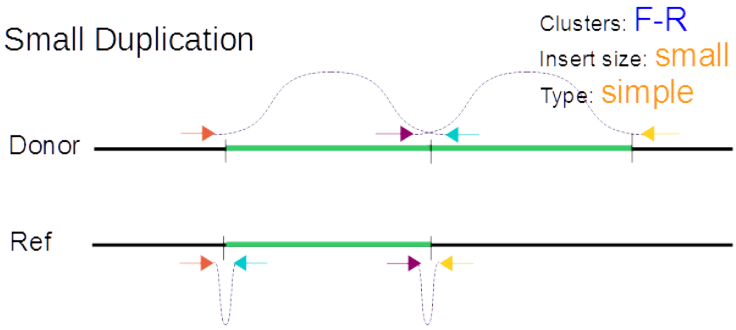


Figure 7.4: Small duplication

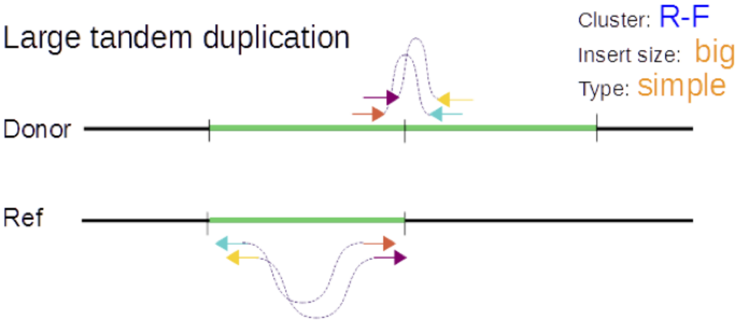


Figure 7.5: Large tandem duplication.

signatures may be associated with this structural variant. A duplication is called *small* if the duplicated region is shorter than the minimum normal fragment length. The corresponding signature is: (*i*) an insert size smaller than the median of normal fragment lengths and (*ii*) reads map with the expected orientation. It is depicted in Figure 7.4. Respectively, a duplication is called *large* if the duplicated region is larger than the maximum normal fragment length. The signature of the relevant fragments is the opposite: insert size is larger than the median normal insert size and both reads have the unexpected orientation. This is depicted in Figure 7.5.

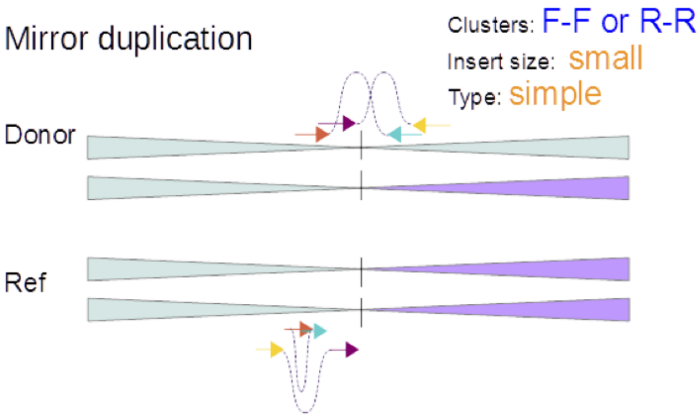


Figure 7.6: Mirror duplication.

Mirror duplication Mirror duplication is a special case of a duplication, where the original and the duplicated fragment are both connected in inversed manner. Mirror duplications can cause triplex DNA [DM11]. For this type of SV, fragment insert size is shorter than the median fragment length, and the orientations of the reads are F-F or R-R (for Illumina paired-end or mate-pair reads). This is shown in Figure 7.6.

7.1.2 Complex structural variants

When there are several links associated with a specific SV, such SV is called *complex*. The list of such SVs is provided further. We consider two links as possibly related to one complex SV only if the distance between their start and end positions does not exceed two normal fragment lengths. In addition to the signature of each link, which is similar to the signature for simple SVs, the signature of complex SVs also includes:

1. *Relative position of the links*: one link may contain another, or they may overlap.

2. *Gain or loss of genomic copies*: the number of copies gained/lost γ, estimated with the Bayesian approach. For instance, considering number of involved copies is essential to distinguish one of the complex SV types called co-amplification.

Basic inversion Inversion of a genome region is one the most common and well-recognized SV types. This type is associated with two overlapping clusters larger than the biggest normal fragment with the orientations FF and RR. There are examples of inversions related to particular types of cancer. For instance, the *CBFB-MYH11* gene fusion created by the inv(16)(p13.1;q22) inversion is associated with a favorable prognosis in AML [dSK+97]. Another example is an inversion in chromosome 2 that fuses the *ALK* gene with another gene called *EML4*; the result is the EML4-ALK fusion protein, which contributes to the development of non-small cell lung cancers [SCE+07].

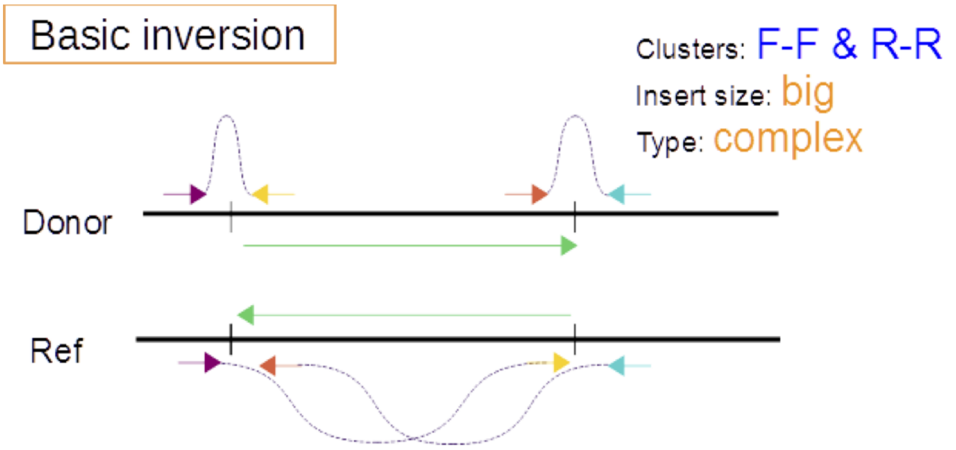


Figure 7.7: Basic inversion.

Re-insertion These SV type has been described and validated in the paper introducing the Meerkat method [YLG+13]. It corresponds to a deletion of a region and a re-insertion of this region into some other location of DNA. This insertion may occur in the initial chromosome (where this region came from) or in another chromosome. The region may be inserted in the orientation conforming with the orientation of this sequence in the reference genome or it can be inverted; in the latter case, the SV is called re-insertion with inversion. This SV type is associated with three links:

- one link corresponds to the deletion and consists of fragments having length larger than the maximum normal fragment length and expected reads orientation;

- the two other links correspond to the insertion in donor DNA. Depending on the type of SV (with or without inversion) orientation of reads in the clusters are forward-forward and reverse-reverse (with inversion) or forward-reverse and reverse forward (without inversion).

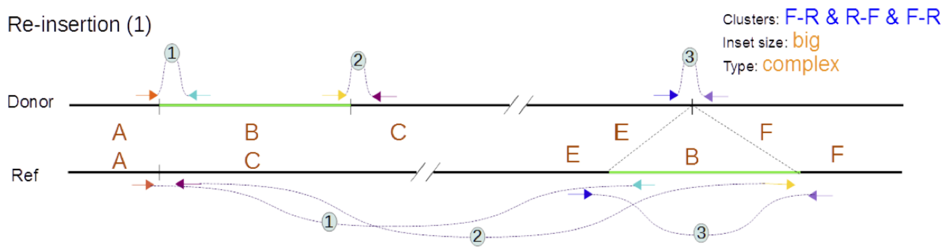


Figure 7.8: Re-insertion.

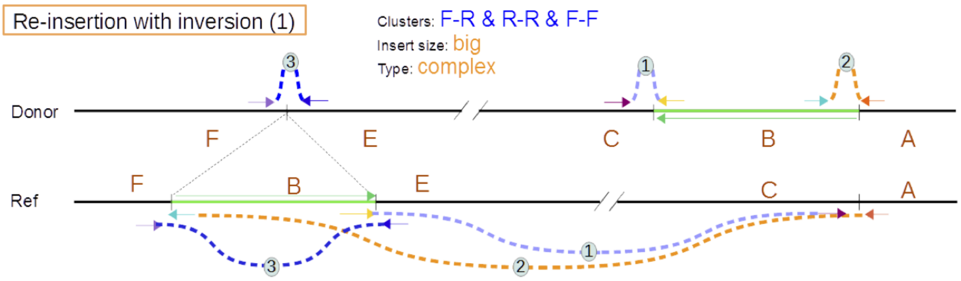


Figure 7.9: Re-insertion with inversion (1)

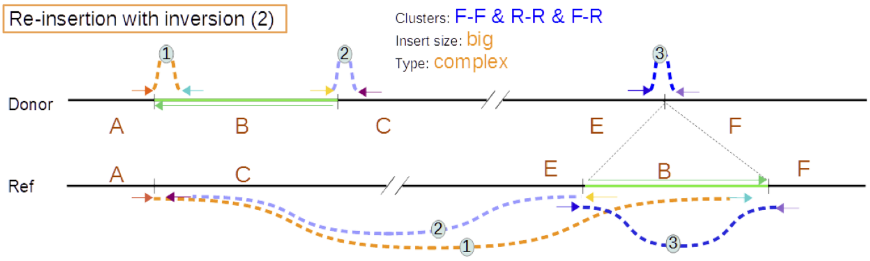


Figure 7.10: Re-insertion with inversion (2)

Large duplication with inversion This duplication has the same origins as small/large duplication described before. The difference is that the duplicated region is inserted in the opposite orientation. Detection of the duplication can vary according to the size of the duplicated region. The signature characteristics are as follows: orientations are forward-forward and reverse-reverse and one of the clusters contains another.

Linking insertion and linking insertion with inversion Linking insertion is defined on the basis of empirical observations of cancer data. This structural variant corresponds to the duplication of a chromosomal region, the duplicated region is further inserted directly or with an inversion at some other genomic location. The direct linking insertion has two corresponding clusters of abnormal fragments. The orientations of reads in these clusters are forward-reverse and reverse-forward. The clusters overlap, reads with the orientations forward and reverse are mapped closer then minimum fragment length.

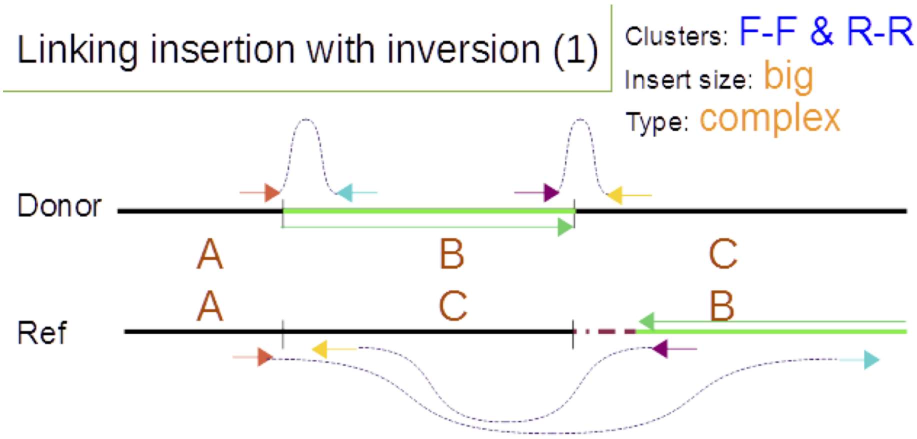


Figure 7.11: Linking insertion with inversion (1)

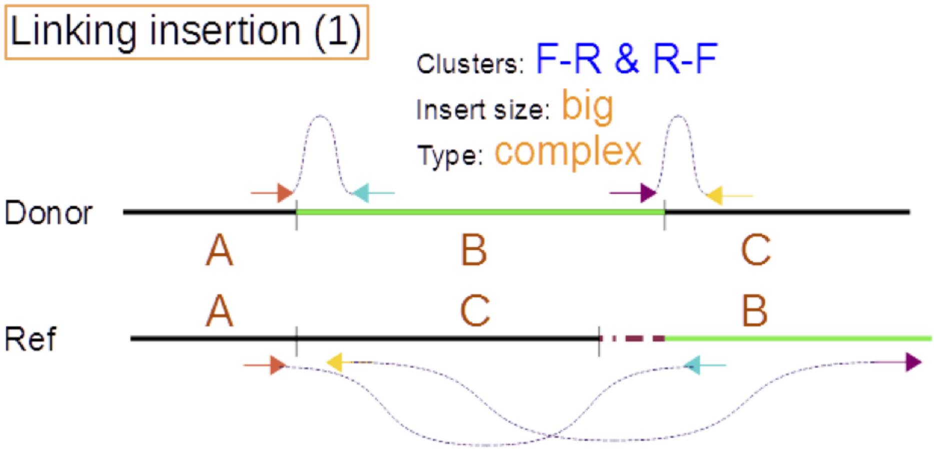


Figure 7.12: Linking insertion.

Balanced translocation and balanced translocation with inversion Balanced translocation is a structural variant resulting in the exchange of parts between two chromosomes. No genetic material is gained or lost as a result of this event. The most common case of balanced translocation is the exchange of the chromosome edges, including telomeres. However, the exchange of random chromosome regions is also possible. The exchanged region can also appear in the new chromosomes inverted, in this case the structural variant is called balanced translocation with inversion. Both balanced and unbalanced translocations can cause formation of new oncogenes and thus be involved in cancer development. A signature that corresponds to a direct balanced translocation consists of two clusters of fragments with reads mapped on different chromosomes (in case of intra-chromosomal translocation on another copy of the same chromosome). The read orientations are forward-reverse and reverse-forward. For balanced translocation with inversion the signature consists of two clusters with reads of each fragment mapped on different chromosomes. The orientation of reads is forward-forward for one cluster and reverse-reverse for another.

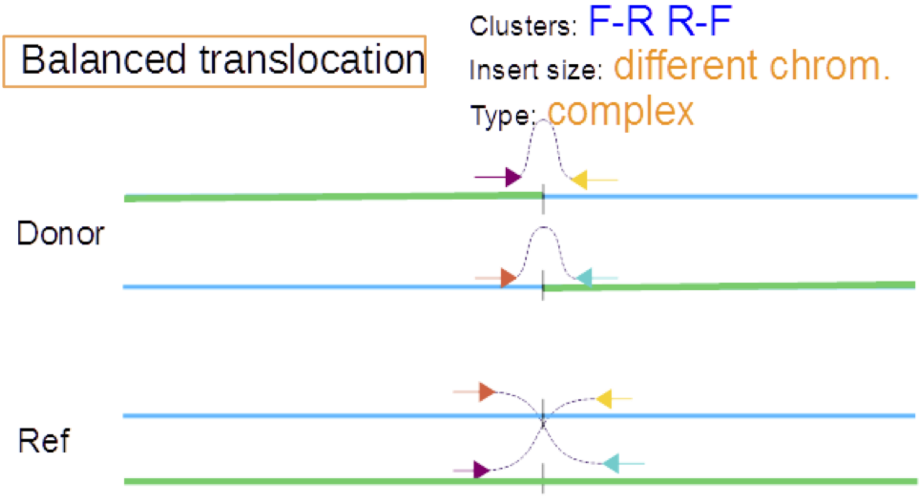


Figure 7.13: Balanced translocation.

Complex Deletion Complex deletion was introduced in [YLG$^+$13]. This structural variant type represents a deletion with an insertion or inversion at the breakpoint position. Authors showed the appearance of such SVs in CDKN2A/2 gene, which codes for two proteins: p16 and p14arf. Both act as tumor suppressors by regulating the cell cycle. p16 activates the retinoblastoma (Rb) family of proteins. p14ARF (also known as p19ARF for the mouse) activates p53 tumor suppressor. According to the International Cancer Genome Consortium, TP53 gene is the most frequently mutated human gene related to cancer (the mutation is present for more than

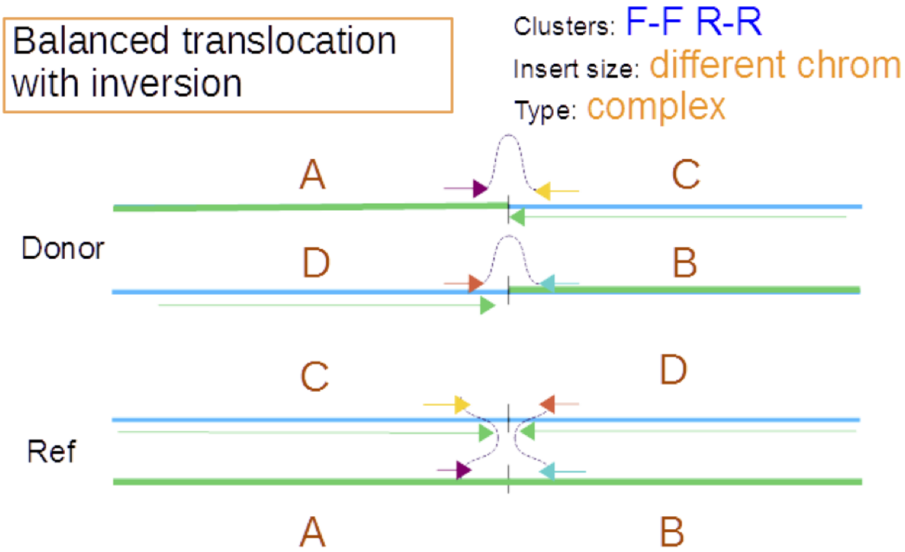

Balanced translocation with inversion
Clusters: F-F R-R
Insert size: different chrom
Type: complex
A
C
Donor
D
B
C
D
Ref
A
B

Figure 7.14: Balanced translocation with inversion.

50 % of cases) [NRR11]. Complex deletion without/with inversion is identified by the signature consisting of two clusters of fragments containing one another. Fragments in both clusters have lengths exceeding the maximum normal fragment length. In case of inversion the clusters consist of fragments with read orientations forward-reverse and forward-forward, in case without inversion the orientations are forward-reverse for both clusters.

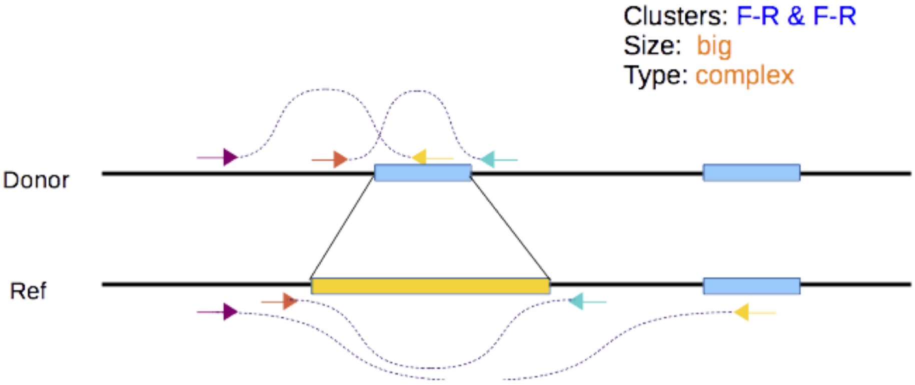


Figure 7.15: Complex deletion.

Co-amplification This SV type has been observed and validated, for example, in [CLBM+13]. Gene co-amplification is common in cancer cells, and some amplified genes may cause cancer cells to grow or become resistant to

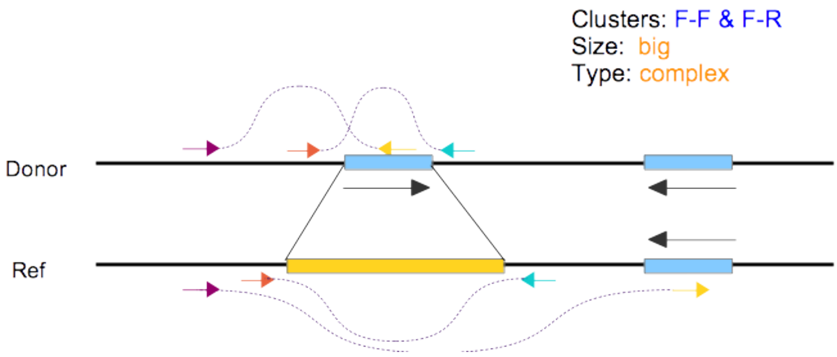


Figure 7.16: Complex deletion with inversion.

anticancer drugs. Additional examples are provided in Chapter 1. For these SVs, one or several regions are duplicated many times (up to hundred times). Thus, detection of this this SV is very specific. First, the number of DNA copies involved in each link should be approximately the same and this number should significantly differ from the normal ploidy. Second, two clusters may either overlap or be nested. This leads to many combinatorial lay-outs.

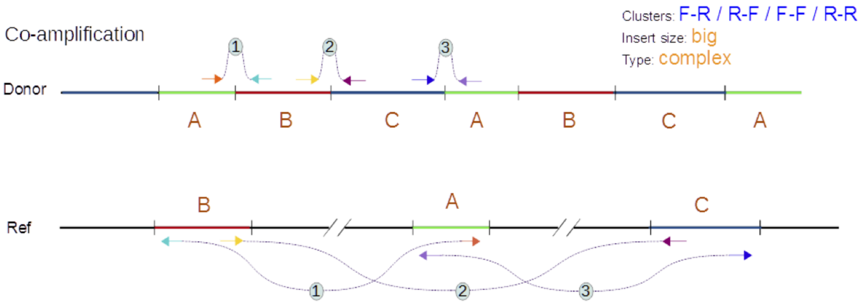


Figure 7.17: Co-amplification

Large duplication with inversion For this SV type there are two corresponding links. For one link, the leftmost and the rightmost reads of each fragment have the reverse orientation, and the insert size is larger than expected. For the second link, the leftmost and the rightmost reads of all fragments have the orientation forward-forward, and the insert size is smaller than expected. Additionally, the first link should contain the second one. This is depicted in Figure 7.18.

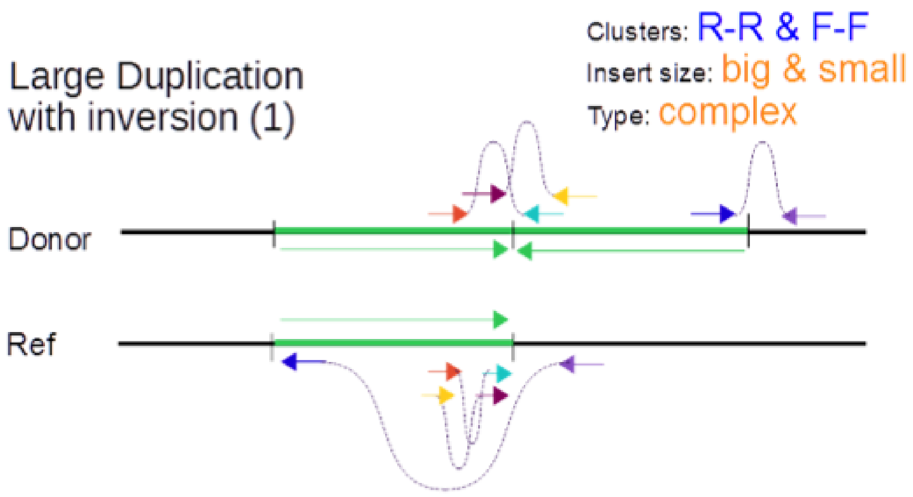


Figure 7.18: Large duplication with inversion

7.2 SV assembly process

Once the Bayesian step is finished, SV-BAY outputs the links that have at least one genomic copy $\gamma > 0$ involved in the rearrangement. These links are likely to be parts of real SVs. We designed an algorithm that assembles these links into complex SVs and determines the SV type. There are links of 4 types defined by the orientation of the fragments inside the link: FF, RR, FR and RF. For each link, information about the number of copies in the flanking regions is provided. This information is not enough to identify a complex SV, as it was described in the previous section. For example, linking-insertion with inversion has two related links with orientations RR and FF, which should be located in a special way in relation to each other and have a certain size (see Figure 7.11). All links are sorted by the left-most positions. The algorithm starts proceeding from the left to right, according to read mapping position on the reference genome. At the first step, SV-BAY distinguishes co-amplifications from other SVs. They cannot be confused with other SV, because the number of genomic copies involved in co-amplifications is significantly larger than it is for any other SV types and this makes identification simple. Thus, SV-BAY splits the set of links according to the number γ of gained or lost copies. This number should be greater than a user-defined threshold b (by default, $b = 10$). Co-amplifications are associated to an amplification level γ higher than b. Second, algorithm deals with the *inter-chromosomal* links. Unbalanced translocations and translocations with inversion are distinguished at this step. Third, the algorithm proceeds to the *intra-chromosomal* links. All the links, which were processed on previous steps are now excluded. For the each link, SV-BAY checks if it overlaps with other links (also considering nested links as overlapping)

. If there are no intersections, we associate the link with one of the following types according to the orientation of the fragments inside the link: *deletion, small insertion, large duplication, small duplication.* If there is an intersection, we keep only those links which have number of gained or lost alleles comparable with current link (usually it is plus/minus one). After that, all possible variants of complex SV are checked. For example, assume that the current link is an FF intra-chromosomal link that overlaps with an intra-chromosomal RR link. We check if both links have average size of the fragments longer then maximum normal fragment length. If it is so, then this couple of links should be associated with a linking insertion with inversion; otherwise, it could be either a large duplication with inversion or a mirror duplication according to the fragment size inside the links. All the links which were involved in complex SV are excluded from the list and not processed in the assembly of other SVs. On each step a decision tree is used for each type of link to identify the associated SV type. In total there are 9 decision trees: FF, RR, RF, FR for intra-chromosomal and inter-chromosomal links and one for co-amplifications. As an example, decision tree for FR intra-chromosomal links 7.19 is provided below (the normal fragment orientation is assumed to be FR). identification of the SV type is always unique, that is why algorithm checks all possible solutions. It is possible that a link might be associated with several complex SVs. In

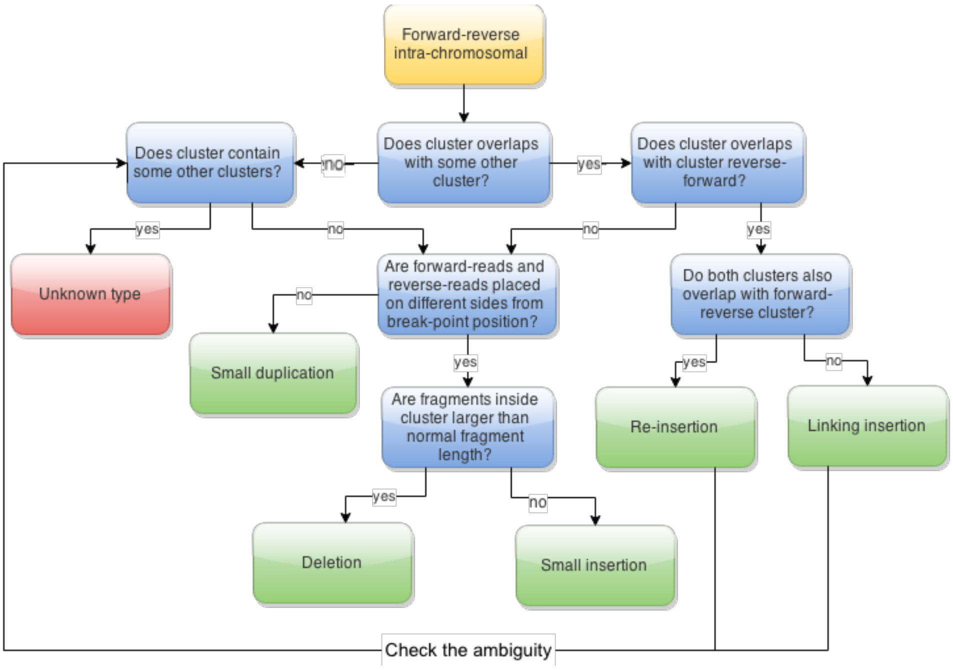

Forward-reverse intra-chromosomal
Does cluster overlaps with some other cluster?
no
yes
Does cluster contain some other clusters?
Does cluster overlaps with cluster reverse-forward?
yes
no
no
yes
Unknown type
Are forward-reads and reverse-reads placed on different sides from break-point position?
Do both clusters also overlap with forward-reverse cluster?
no
Small duplication
yes
yes
no
Are fragments inside cluster larger than normal fragment length?
Re-insertion
Linking insertion
yes
no
Deletion
Small insertion
Check the ambiguity

Figure 7.19: decision tree for assembly SVs if first observed link has orientation forward-reverse, assuming that expected orientation is RF.

this case, SV-Bay algorithm outputs all the possible complex SVs for this link, but does not make any decision. A warning is also reported. If a link could not be associated with any complex SV, it is treated as a simple event (deletion, insertion, mirror duplication, etc.). Complex events always have higher priority than simple SVs.

Chapter 8

Overview of competitive methods for SV detection

In this chapter we provide an overview of four published SV detection methods: GasvPro [SOP+12], BreakDancer [CWM+09], Lumpy [LCQH14] and Delly [RZS+12]. These methods are further compared to SV-Bay on simulated and experimental data. For each method, a brief description is provided, advantages and drawbacks are discussed, the basic information about the implementation is given, such as programming language, license and some performance measures.

The exact parameters used for each tool in further experiments are provided. Finally, we compare the main features of each tool. The support of different SV types is also covered.

8.1 Methods description

8.1.1 GasvPro

GasvPro uses read depth and read pair information and integrates these two signals into a probabilistic model. Unlike most SV detection algorithms, which ignore reads with several alignments, GasvPro considers all possible alignments for each read. A Markov Chain Monte Carlo (MCMC) model is used to sample the set of fragment possible mappings. This approach increases SV detection sensivity, especially in repetitive regions. A hard clustering (that uses only fragments with a high mapping quality) is also available under the name GasvPro-HQ.

In addition to the standard read pair signatures, a breakend read depth (beRD) method is designed: the read depth is analyzed to discover localized drops of coverage that occur at the breakpoints of both copy number-variant and copy number-invariant SVs. It is also used to predict zygosity of variants. GasvPro uses simultaneously the amount of discordant read pairs and the beRD signatures at each breakpoint to determine the likelihood of a potential SV and remove false positives.

The major GasvPro drawback is that it does not take into account the possible bias of read counts due to changes in the GC-content (GC-content bias). Moreover, mappability is only considered for the regions with abnormal read-pairs, while it also influences normal regions; so Doc information used is very unreliable.

GasvPro is written in Java and distributed under GNU GPLv3 open source license. The tool allows to manually set up parallel processing of different chromosomes data on several CPU cores. In our experiments GasvPro showed reasonable speed and memory usage for pair-ends dataset, processing whole genome in about 5 hours. Nevertheless, processing mate-pair data was very slow, taking several days for some chromosomes.

8.1.2 BreakDancer

BreakDancer is probably the most popular tool for SV detection. It implements two complementary algorithms: BreakDancerMax provides genome-wide detection of insertions, deletions, inversions, inter- and intra-translocations, while BreakDancerMini focuses on detecting small indels, typically in the range of 10-100 bp, that cannot be found with BreakDancerMax.

BreakDancerMax algorithm uses a standard clustering strategy. Clusters classification relies on orientation of the paired reads, mapped distance between them and insert size distribution in the dataset. Estimation of the confidence score is based on a Poisson distribution model that takes into account the number of fragments in the cluster, the coverage of the genome and the length of the region covered by the cluster. BreakDancerMini uses a sliding window test to identify the small indels: it checks the difference between the separation distances of fragments that are mapped within the window *versus* fragments in the entire genome. Abnormal regions are identified using Kolmogorov-Smirnov test. BreakDancer does not use Doc information despite its relevance for SV detection.

The tool is implemented in Perl/C++ and distributed under GNU GPLv3 open source license. BreakDancer is the fastest tool of those we have tested - the processing of the whole human genome data took around half an hour for both mate-paired (fragment coverage 8) and pair-ends datasets (fragment coverage 14).

8.1.3 Lumpy

The most recent method, Lumpy, combines three approaches: PEM, Doc and coverage by split-reads. It also uses some initial information about already validated or manually imputed mutations. Three modules are provided in Lumpy to process various SV signals: read-pair, split-read and generic. Read-pair module uses the output of a paired-end sequence alignment algorithm such as NovoAlign or Bwa, split-read module uses the output of a split-read sequence alignment algorithm (for example, Yaha, Bwa-Sw, or Bwa-Mem). The generic module allows the user to provide prior knowledge of known SVs or copy number variations discovery tool output.

Integrating different sources into a single discovery process allows sensitive SV detection. Nevertheless, Lumpy does not take into account GC-content and region mappability, so Doc information can be unreliable and lead to the wrong SV detection. Additionally, Doc calculation is not implemented in the tool: the user has to generate information about CNVs with some other tools like CVNator of Freec and provide it as input in a specific format. Basically, Lumpy only includes modules to process split-reads and perform clustering of pair-end reads. Moreover, Freec and CVNator use large windows (10-50 Kb) to estimate the Doc; so this information cannot be used for precise SV detection. Also Lumpy is able to work only with pair-end data, but not with mate-pairs.

Lumpy is an open source software written in C++ and distributed under Mit license. It can process several data samples simultaneously and proved to be rather fast, identifying SVs in a whole genome in about 12 hours on a single core using 8 gigabytes of memory according to the authors' information. In our experiments Lumpy processed the whole genome data even faster - in about 4 hours. The tool is relatively easy to use, but one has to perform several data pre-processing steps using external tools to run it.

8.1.4 Delly

Another recent method, Delly, combines short-range and long-range paired-end mapping and split-read analysis for SV discovery. Delly focuses on enabling SV calling in the presence of different paired-end sequencing libraries with distinct insert sizes. Using of different datasets allows an accurate discovery of both small and large SVs, while the split-read analysis allows for the breakpoint resolution up to a single nucleotide.

During paired-end mapping analysis Delly considers uniquely mapped read pairs which either have an abnormal orientation or have an insert size that differs from the median insert size by more than three standard deviations. Such read pairs are clustered using an specially designed graph-based algorithm. This analysis is performed separately for each library provided. During split-read analysis the clusters are interpreted as genomic intervals that contain breakpoints. Delly tries to map reads in the split-read mode to detect SV breakpoints at the single-nucleotide resolution. For this purpose, all pairs with one read mapped and the other unmapped are checked.

Pairs mapped closely to SV breakpoint are added to the set of putative split-reads. Putative split-reads are filtered with a special k-mer-based approach and finally mapped to the reference region using dynamic programming.

The major drawback of this method is that it does not use Doc information, which is important for filtering clustering results not corresponding to a real structural variant. Another problem is that Delly achieves best sensitivity and specificity only when using mate-paired and paired-ends datasets at the same time, but in real life these datasets are very rarely available simultaneously for the same genome.

Delly is implemented in R and C++ programming languages and distributed under Gnu GPLv3 license. The authors report Delly to process a sequenced human genome with median read coverage equal to 5 in 2 hours, in our experiments the dataset with nearly same coverage was processed even faster – in about 45 minutes. The processing time linearly depends on the coverage. The simple parallelization on chromosome level is available, which allows to further accelerate the processing. The memory footprint is around 4 gigabytes and does not depend on the coverage.

8.2 Configuration used for considered software

The following versions of the tools were used: SV-Bay v1.0, BreakDancer v1.3.6, GasvPro v2.0, Lumpy v0.2.9. The major configuration parameters of each tool are discussed further.

SV-Bay uses an important user-defined parameter, the *expected number of SVs*. This number gives SV-Bay a priory probability of an SV candidate being a real SV (explained in Chapter 6). This parameter is mandatory. It is assumed that the user has a prior knowledge about the patient such as the diagnosis and duration of the disease, which allows to evaluate the mutations number. The exact values used are 1200 for simulated PE data, 200 for simulated MP data and 3300 for real MP data.

BreakDancer has three principal user-defined parameters:

- n is the number of observations required to estimate the mean and the standard deviation of the insert size.

- r is the minimum number of read pairs in a cluster, starting from which the algorithm considers it as a possible SV. This parameter can be set from 1 to infinity. Value 1 will lead to a lot of false positive predictions. Additionally, it will dramatically change the runtime of the algorithm. With a large value a lot of real SVs, which are poorly covered, might be lost.

- q is the minimum mapping quality. The default value suggested for BreakDancer is 30, but we use 20 for two reasons. First, our data have a rather low coverage, which means that a high threshold can cause a major data loss. Second, minimum mapping quality equal to 20 is internally used in SV-Bay and other tools, which makes the comparison of different tools more accurate. The value 20 is common between several published methods, because it corresponds to a probability 99% for the fragment to be correctly mapped.

The same parameters are used for BreakDancer for each dataset: $n = 100000$, $q = 20$ and $r = 3$. These values are equal to the corresponding internal values in SV-Bay.

GASVPro does not have any non-trivial parameters. However, for GasvPro, we had to manually set two parameters from the SV-Bay output, because GasvPro failed to calculate these parameters correctly. These parameters are average genome coverage $\lambda = \frac{(\#normal\ fragments)}{(genomelength)}$ and median of the fragment lengths μ. These values were set respectively to 25.5 and 399 for simulated PE data, 54 and 4282 for simulated MP data and 14 and 3038 for real MP data.

Lumpy was run on paired-end data only as it does not support mate-pair datasets. The default parameters were left unchanged for this tool.

In this section the general comparison of the tested tools is provided. We discuss the features implemented and the support of different SV types. The runtime and memory consumption of each tool are also compared.

8.3 Main features and scopes

The features implemented and the different SV types that may be detected are discussed below. The major differences between SV-Bay, BreakDancer, GasvPro, Lumpy and Delly *features* are summarized in Table 8.1.

	SV-Bay	BreakDancer	GasvPro	Lumpy	Delly
Processes PE libraries	+	+	+	+	+
Processes MP libraries	+	+	+	-	+
Uses Doc information	+	-	+	±*	-
Uses split-reads	-	-	+	+	+
Uses read mappability	+	-	±**	-	-
Uses GC-content	+	-	-	-	-
Uses normal controls	+	+	-	+	+
Detects complex SVs	+	-	-	-	-

Table 8.1: * removes regions with extremely high read coverage; ** GasvPro uses read mappability information only to estimate the number of abnormal fragments spanning the breakpoint position, but not to correct Doc in flanking regions;

Different types of information about the data may be used. Most considered tools (Lumpy is the exception) are able to process both paired-end and mate-pair data.

Unlike BreakDancer and Delly, SV-Bay takes into account Doc to validate SV candidates and reduce the number of false positive results. GasvPro and Lumpy also use Doc, but SV-Bay is the only method that takes into account GC-content and mappability. As explained in Chapter 4, these two factors highly affect Doc [BZB^{+}11]. Also, similarly to all tools but GasvPro, SV-Bay can use Bam files generated from constitutive DNA in order to filter out read alignment artefacts and germline SVs.

The only type of information, which SV-Bay does not use while several other tools do, is split-read mappings. The main reason for this choice is that for datasets with rather low coverage (mate-pair datasets), the breakpoint is rarely covered by many reads. Also, split-reads do not bring additional information in the case of homology at the breakpoint junction, which is often observed in cancer samples. Results in Chapter 9 show that SV-Bay, which does not use split-reads, achieves a good breakpoint resolution using the probabilistic model. Therefore, although our method can be theoretically improved by the use of split-reads, we did not implement this possibility.

One of the most important characteristics of the SV detection software is the list of SV types the tool can detect. SV-Bay is the only tool which is able to detect complex SVs.

More precisely, existing tools are able to detect just few types of structural variants among the ones that were exhaustively listed in Chapter 7. GasvPro detects only deletions, inversions and translocations. BreakDancer and Lumpy additionally support insertions, the latter can also identify duplications. Delly is able to identify tandem duplications, but not insertions. None of these tools support complex SVs (co-amplifications, tandem duplications with an inversion of the duplicated unit, linking insertions and linking re-insertions), while SV-Bay supports all these types. This is an important advantage of SV-Bay, that is crucial for accurate interpretation of results.

This comparison illustrates that SV-Bay generally uses a more complex approach for SV detection and takes into account more sources of information.

Chapter 9

Results on simulated and real data

In this chapter, results of SV-Bay on both simulated and real datasets are given. To perform this test, simulation of the tumor genome data was performed. Then, SV-Bay is compared with other SV detection tools: GasvPro, Lumpy,BreakDancer and Delly. Mate-pair and pair-end datasets were prepared.

Section 9.1 addresses the production of simulated data. First, simulated genomes (normal or tumor) are to be created: the data simulation method is discussed. In the second step, mate-pair and pair-end datasets preparation is explained.

Then, the results for all considered tools on the simulated and real datasets are compared. For each tool we provide a precision-recall curve and extensively discuss the results.

Also we discuss the quality of breakpoint resolution of each tool, the influence of dataset type (mate-pair or paired-end) and CNV presence.

In Section 9.5, the execution time of considered tools is compared.

9.1 Simulated data

Simulation pipeline includes TGSim, a software we developed to simulate a tumor genome (`https://github.com/InstitutCurie/TGSim`), and Pirs, a read simulation software [HYS+12] (`code.google.com/p/pirs/`).

The first step of the pipeline is adding SNPs, small indels and small inversions to the reference genome using Pirs. This step produces a simulated normal genome with germline mutations. The second step is adding different SVs to the normal genome using TGSim. This step produces a simulated cancer genome. The last step is simulating mate-pair and pair-ends sequenced data for the cancer genome using Pirs. These three steps are described further.

9.1.1 Simulation of normal genome

The first step is the creation of a normal control diploid genome. To generate such a nucleotide sequence, we modify the reference human genome GRCh38/hg38 using Pirs.

Pirs modifies the reference genome by adding 3 million heterozygous SNPs (single-nucleotide polymorphisms). Pirs also introduces the list of germline mutations (small indels and inversions) in the simulated genome. Approximately 2000 small indels and 1000 small inversions with length from 100 to 2000 bp are inserted.

The resulting diploid genome is produced as output in Fasta format.

9.1.2 Simulation of cancer genome

The next step is simulation of cancer genome by adding complex structural variants to the control normal genome.

We specially designed an algorithm, TGSim, in order to modify a normal genome. This algorithm rearranges the nucleotide sequence in a special way that depends of the structural variant to be simulated. For example, for a linking insertion, a genomic region is copied and inserted in some other location and for re-insetion a region is cut from one location and inserted at another.

The main problem adding complexity to this algorithm is the fact that indices in the genome are changed after each rearrangement, even one insertion. Therefore, all coordinates for the next SVs are re-calculated.

TGSim allows to inject any simple or complex structural variant described in the SVs list in Chapter 7. Since we defined a number of novel SV types, no appropriate tool allowing to do so existed before.

During the data simulation TGSim introduces 62 novel genomic adjacencies. The number of simulated SVs reflects, in our opinion, the usual number of large SVs in a cancer genome. The list of genomic events simulated is provided below:

- 3 inversions of lengths from 6000 to 750000 bp
- 2 tandem duplications of lengths 3000 and 10000 bp
- 2 deletions of lengths 10000 and 700000 bp
- Whole genome duplication
- Random loss of 8 chromosomes
- Random duplication of 2 chromosomes
- 1 balanced translocation

- 4 unbalanced translocations
- 5 tandem duplications of lengths from 3000 to 1200000bp
- 3 deletions of lengths from 100000 to 3000000 bp
- 3 deletions of lengths from 3000 to 100000 bp
- 3 linking insertions of lengths from 4000 to 40000 bp, direct orientation
- 2 linking insertions of lengths from 10000to 30000 bp, reverse orientation
- 2 re-insertions of lengths 3000 and 80000 bp, direct orientation
- 2 re-insertions of lengths 4000 and 60000 bp, reverse orientation
- 2 inverted duplications of lengths 3000 and 40000 bp
- 3 inversions of lengths from 3000 to 200000 bp
- 5 de novo insertions of lengths from 1000 to 2000 bp
- Random duplication of 1 chromosome
- Random deletion of 1 chromosome
- 1 unbalanced translocation
- Random duplication of 1 chromosome

This scenario does not cover the chromothripsis phenomenon. However, the latter is observed in only 2-3% of tumors [FKJ12]. SV-Bay is supposed to work equally well on such samples by design; the user may only need to increase the parameter value corresponding to the expected number of SVs. Thus, we did not perform a corresponding validation.

The .Fasta file generated for the tumor genome by TGSim contains approximatively 4 copies of each chromosome (tetraploid genome) because we added a whole genome duplication frequent in cancer cells.

9.1.3 Simulation of sequencing data

Once a tumor genome is created, the last step is simulation of sequencing process that includes drawbacks known for *Illumina* sequencing. The Pirs tool was used to create the sequencing data.

When generating the sequencing data, Pirs considers the GC-content profile. The GC-content profile is built based on experimental sequenced data for the NB1142 neuroblastoma cell line from [VB13].

A procedure close to the one used in SV-Bay (see Chapter 5) is used to build a GC-content profile for these data. The dependency between GC-content and read-coverage is analyzed using a sliding window along the whole genome. In each window, the GC-content and the average base depth are computed. Then, for each GC-content interval (1% step), the mean depth value is calculated. This GC%-depth profile reflects the mean coverage depth in sequence regions with a similar GC content. The resulting profile is shown in Figure 9.1.

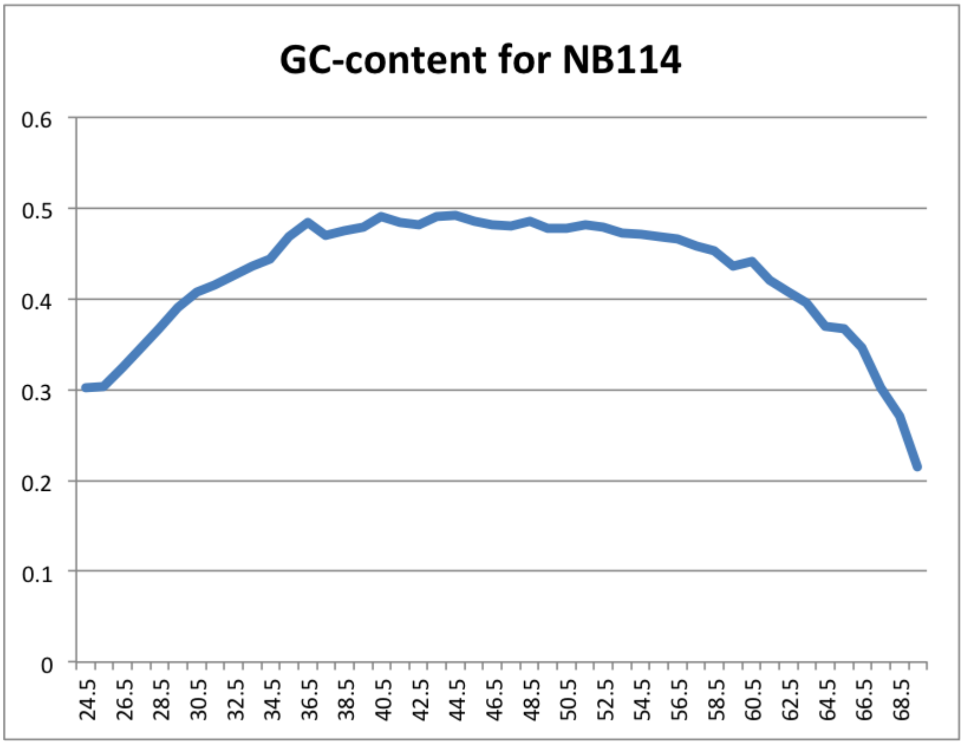


Figure 9.1: GC-content for the experimental data for the NB1142 neuroblastoma cell line that was applied to model the GC-content bias in the set of simulated data. The X-axis denotes GC-content, the Y-axis denotes fragment count.

Pirs generates reads in the Fastq format. For each read pair, the insert size is randomly drawn from the normal distribution with given values for the mean and standard deviation.

We also simulated both mate-pair (MP) and paired-end (PE) sequenced data for the diploid control genome, to be able to exclude germline mutations and mapping artefacts.

9.1.4 Mate pair and pair-ended datasets statistics

Below, we provide different statistics on the simulated MP and PE datasets. These statistics were collected by SV-Bay (no other tools tested provide such information).

The read length equal to 70 bp is used for both MP and PE data. This choice was made because the sequencing data chosen to build the GC-depth profile for Pirs has the 70bp read size. The same length was left to ensure similarity.

The simulated MP dataset contains approximately 45 millions of read pairs with an average insert size of 4282 bp. The PE data contains approximately 222 millions of read pairs with average insert size of 400 bp. By design, the MP library was contaminated by approximately 5 millions (10%) of singletons with forward-reverse orientation and average insert size of 400bp. Matched normal datasets also contain approximately 45 and 222 millions of read pairs for mate-pair and paired-end libraries respectively.

	MP	PE
Total number of fragments	45,843,392	222,042,622
Number of normal fragments	38,973,430	217,868,967
Number of abnormal fragments	409,199	4,173,655
Coverage for one allele	16.5	6

Table 9.1: Number of read pairs in mate-pair and paired-end simulated datasets.

As mentioned in Section 5.2.2, SV-Bay evaluates parameter λ in regions with expected ploidy. Such regions are identified with the Control-FREEC tool. In Figure 9.2, the calculated CN profile for simulated data is shown.

9.2 Comparative performances on simulated data

In this section we compare the results of SV-Bay and other considered tools on the simulated data.

First, the comparison scores, precision and recall, are discussed. Then we provide the detailed results for each tool.

9.2.1 Precision and recall

The tools are compared based on two scores: *precision* and *recall*. Recall is the proportion of true SVs, that were predicted by a specific tool, among all true SVs. Precision is the proportion of true SVs, that were predicted by a specific tool, among all SVs predicted by this tool.

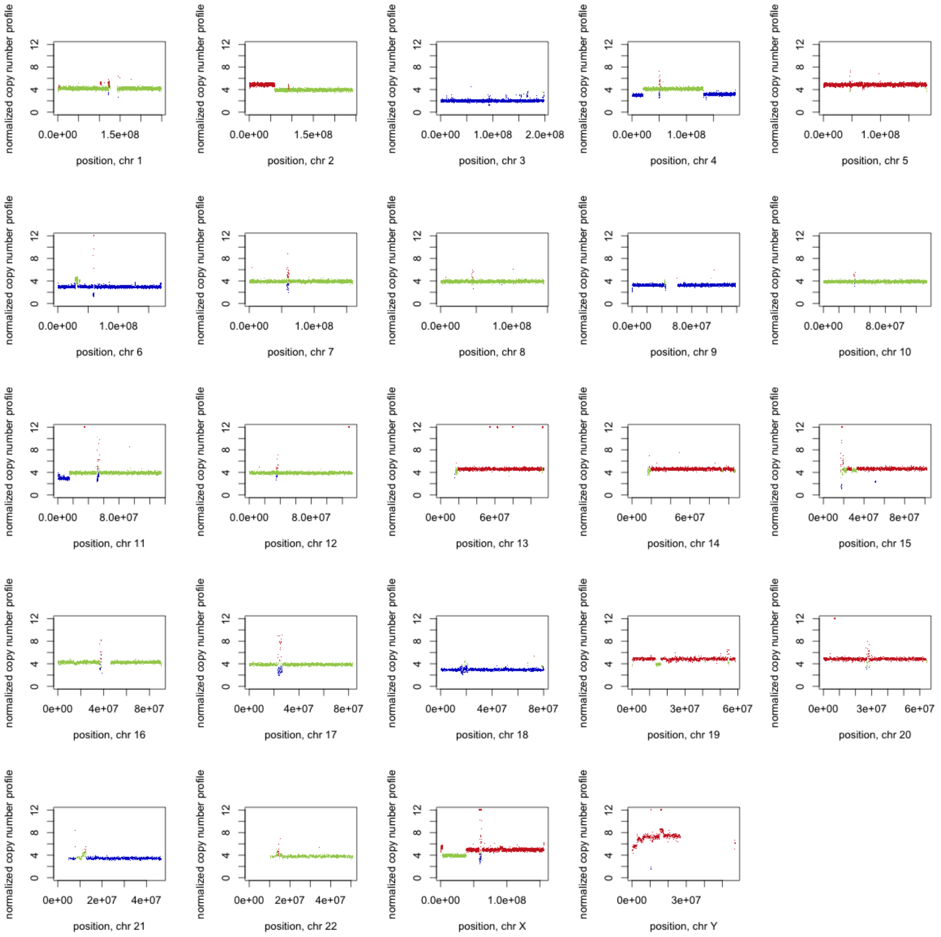


Figure 9.2: Copy number profile for mate-pair data, generated by FREEC.

All considered tools annotate the predicted SVs with a quality score. This allows to calculate recall and precision rates when setting variable thresholds for this score. Reducing the threshold, we increase the number of predicted SVs considered and thus possibly increase the recall.

Using the described approach, a precision/recall curve may be produced. It is depicted in Figure 9.3 for both MP and PE datasets for each considered tool.

As specified in the previous section, 62 genomic adjacencies were inserted into a simulated tumor genome. SV-Bay achieved maximal recall on mate-pair and paired-end datasets, detecting 59 and 25 correct novel genomic adjacencies, respectively. BreakDancer was the second best after SV-Bay in terms of recall.

However, since BreakDancer uses only information about abnormal read mappings and no split-reads or information about changes in DOC, it gives a large number of false positive predictions (Figure 9.3). Overall, on simulated data SV-Bay demonstrates better prediction accuracy than other tools, both in terms of precision and recall.

9.2.2 Detailed results

Detailed results for all tools are given in Table 9.3. The total number of predictions for each tool is shown in table 9.2.

	MP	PE
BreakDancer	1787	197
GASVPro	56	153
Lumpy	-	171
Delly	16675	479
SV-Bay	85	66

Table 9.2: Number of SVs predicted for MP and PE datasets by each tool.

9.3 Comparative performances on a neuroblastoma mate-pair dataset

To investigate the performances on experimental data, we selected a mate-pair dataset from a neuroblastoma diploid cell line CLB-GA. This dataset was recently sequenced using a mate-pair protocol together with a corresponding normal control dataset [VB13]. SVs were predicted for this data and the correlation with biologically validated SVs was studied, allowing to check the performance of SV-Bay and compare it with BreakDancer, Delly and GasvPro. Lumpy was excluded from this test as it is not able to analyse mate-pair data.

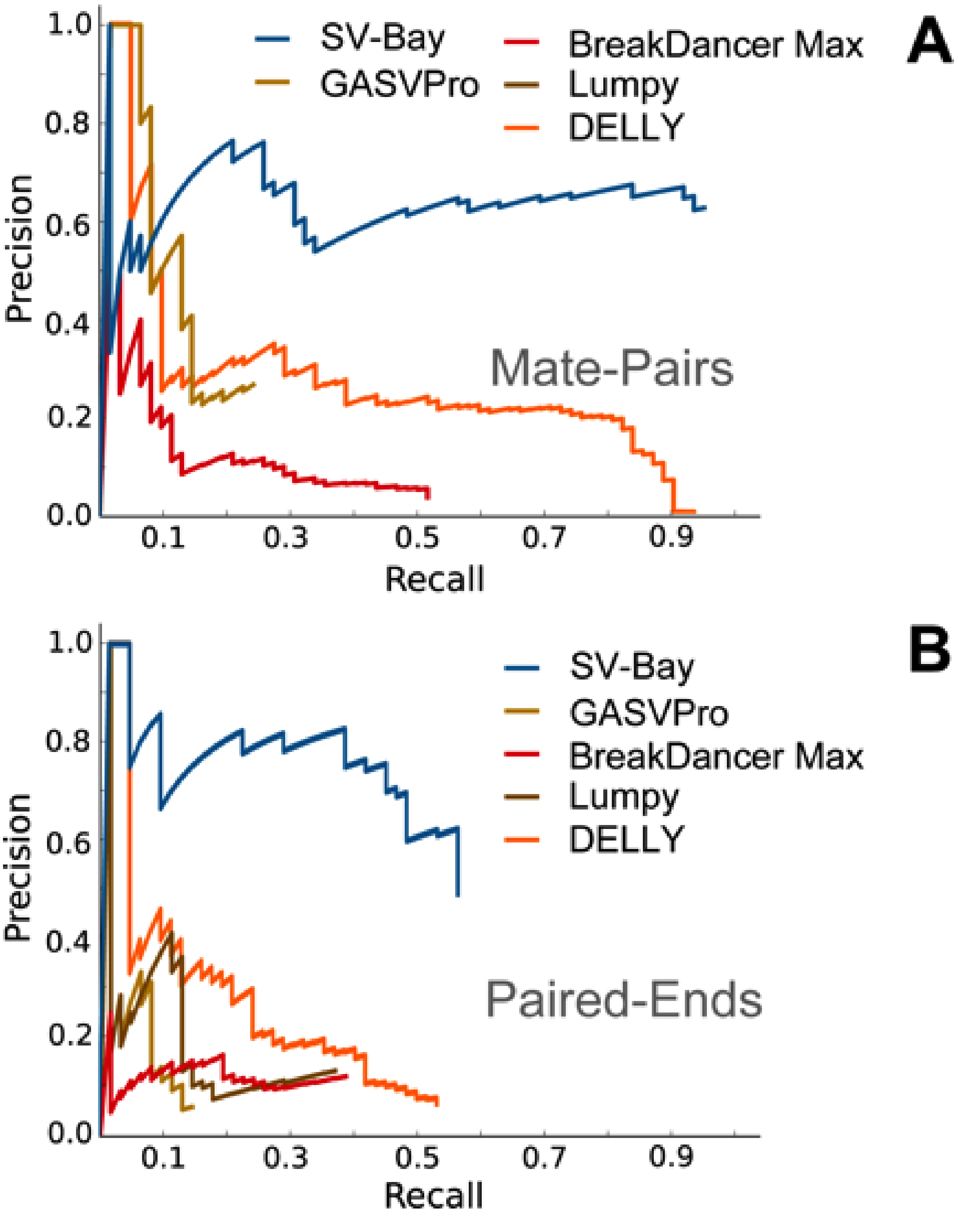
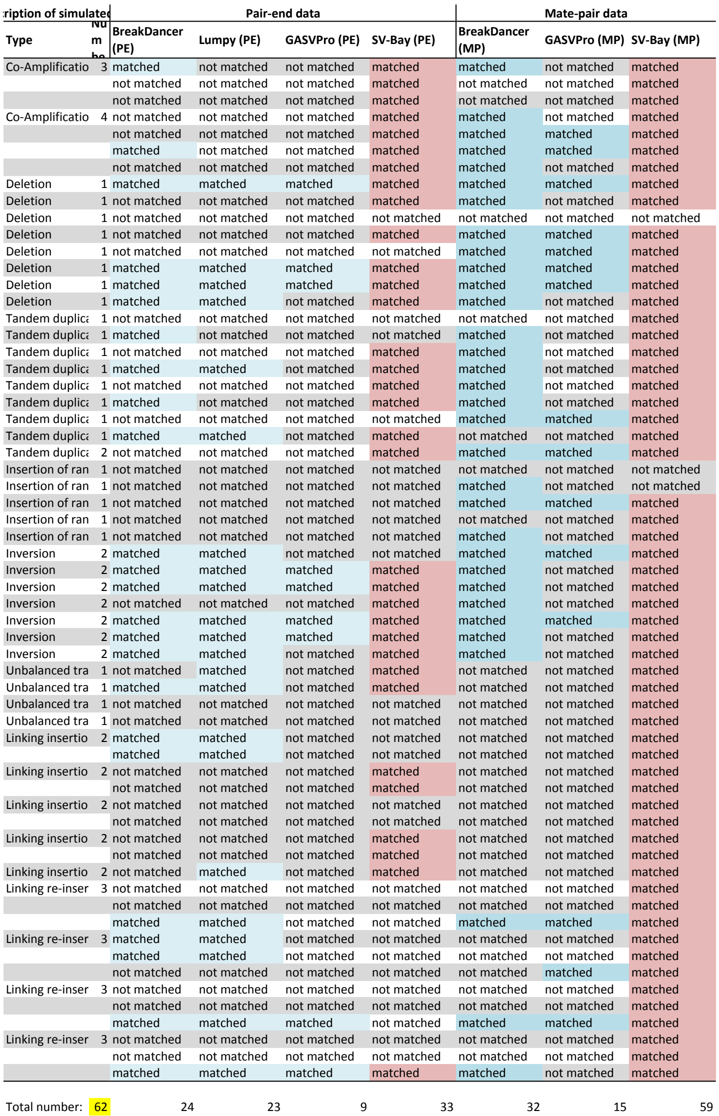


Figure 9.3: Prediction accuracy on simulated data for BreakDancer, GasvPro, Lumpy, and SV-Bay. (A) Precision/recall curves for simulated mate-pair library; (B) Precision/recall curves for simulated paired-end library. The results for Delly are shown only for PE data. For all tools, we kept only SVs with average insert size larger than 100bp and 500bp for PE and MP data respectively.



:ription of simulated		Pair-end data				Mate-pair data		
Type	Nu m be	BreakDancer (PE)	Lumpy (PE)	GASVPro (PE)	SV-Bay (PE)	BreakDancer (MP)	GASVPro (MP)	SV-Bay (MP)
Co-Amplificatio	3	matched	not matched	not matched	matched	matched	not matched	matched
		not matched	not matched	not matched	matched	not matched	not matched	matched
		not matched	not matched	not matched	matched	not matched	not matched	matched
Co-Amplificatio	4	not matched	not matched	not matched	matched	matched	not matched	matched
		not matched	not matched	not matched	matched	matched	matched	matched
		matched	not matched	not matched	matched	matched	matched	matched
		not matched	not matched	not matched	matched	matched	not matched	matched
Deletion	1	matched	matched	matched	matched	matched	matched	matched
Deletion	1	not matched	not matched	not matched	matched	matched	not matched	matched
Deletion	1	not matched	not matched	not matched	not matched	not matched	not matched	not matched
Deletion	1	not matched	not matched	not matched	matched	matched	matched	matched
Deletion	1	not matched	not matched	not matched	not matched	matched	matched	matched
Deletion	1	matched	matched	matched	matched	matched	matched	matched
Deletion	1	matched	matched	matched	matched	matched	matched	matched
Deletion	1	matched	matched	not matched	matched	matched	not matched	matched
Tandem duplica	1	not matched	not matched	not matched	not matched	not matched	not matched	matched
Tandem duplica	1	matched	not matched	not matched	not matched	matched	not matched	matched
Tandem duplica	1	not matched	not matched	not matched	matched	matched	not matched	matched
Tandem duplica	1	matched	matched	not matched	matched	matched	not matched	matched
Tandem duplica	1	not matched	not matched	not matched	matched	matched	not matched	matched
Tandem duplica	1	matched	not matched	not matched	matched	matched	not matched	matched
Tandem duplica	1	not matched	not matched	not matched	not matched	matched	matched	matched
Tandem duplica	1	matched	matched	not matched	matched	not matched	not matched	matched
Tandem duplica	2	not matched	not matched	not matched	matched	matched	matched	matched
Insertion of ran	1	not matched	not matched	not matched	not matched	not matched	not matched	not matched
Insertion of ran	1	not matched	not matched	not matched	not matched	matched	not matched	not matched
Insertion of ran	1	not matched	not matched	not matched	not matched	matched	matched	matched
Insertion of ran	1	not matched	not matched	not matched	not matched	not matched	not matched	matched
Insertion of ran	1	not matched	not matched	not matched	not matched	matched	not matched	matched
Inversion	2	matched	matched	not matched	not matched	matched	matched	matched
Inversion	2	matched	matched	matched	matched	matched	not matched	matched
Inversion	2	matched	matched	matched	matched	matched	not matched	matched
Inversion	2	not matched	not matched	not matched	matched	matched	not matched	matched
Inversion	2	matched	matched	matched	matched	matched	matched	matched
Inversion	2	matched	matched	matched	matched	matched	not matched	matched
Inversion	2	matched	matched	not matched	matched	matched	not matched	matched
Unbalanced tra	1	not matched	matched	not matched	matched	not matched	not matched	matched
Unbalanced tra	1	matched	matched	not matched	matched	not matched	not matched	matched
Unbalanced tra	1	not matched	not matched	not matched	not matched	not matched	not matched	matched
Unbalanced tra	1	not matched	not matched	not matched	not matched	not matched	not matched	matched
Linking insertio	2	matched	matched	not matched	not matched	not matched	not matched	matched
		matched	matched	not matched	not matched	not matched	not matched	matched
Linking insertio	2	not matched	not matched	not matched	matched	not matched	not matched	matched
		not matched	not matched	not matched	matched	not matched	not matched	matched
Linking insertio	2	not matched	not matched	not matched	not matched	not matched	not matched	matched
		not matched	not matched	not matched	not matched	not matched	not matched	matched
Linking insertio	2	not matched	not matched	not matched	matched	not matched	not matched	matched
		not matched	not matched	not matched	matched	not matched	not matched	matched
Linking insertio	2	not matched	matched	not matched	matched	not matched	not matched	matched
Linking re-inser	3	not matched	not matched	not matched	not matched	not matched	not matched	matched
		not matched	not matched	not matched	not matched	not matched	not matched	matched
		matched	matched	not matched	not matched	matched	matched	matched
Linking re-inser	3	matched	matched	not matched	not matched	not matched	not matched	matched
		matched	matched	not matched	not matched	not matched	not matched	matched
		not matched	not matched	not matched	not matched	not matched	matched	matched
Linking re-inser	3	not matched	not matched	not matched	not matched	not matched	not matched	matched
		not matched	not matched	not matched	not matched	not matched	not matched	matched
		matched	matched	matched	not matched	matched	matched	matched
Linking re-inser	3	not matched	not matched	not matched	not matched	not matched	not matched	matched
		not matched	not matched	not matched	not matched	not matched	not matched	matched
		matched	matched	matched	matched	matched	not matched	matched
Total number:	62	24	23	9	33	32	15	59

Table 9.3: Structural variants in simulated datasets and prediction sensitivity of SV-Bay, BreakDancer, GasvPro and Lumpy

9.3.1 Experimentally validated structural variants

For the considered cancer cell line, we had a set of 11 SVs validated by PCR and Sanger sequencing. Most of these (table 9.4) SVs correspond to two breakpoints in the SNP array copy number profile. The following SVs correspond to only one breakpoint:

(i) SV between the ALK gene (chromosome 2p, 29Mb) and a repetitive peri-telomeric sequence; the exact position of the breakpoint could not be defined;

(ii) SV between chromosomes 12q and 20q, as it corresponds to a more complex SV on chromosome 12q;

(iii) the inverted duplication at chromosome 5q.

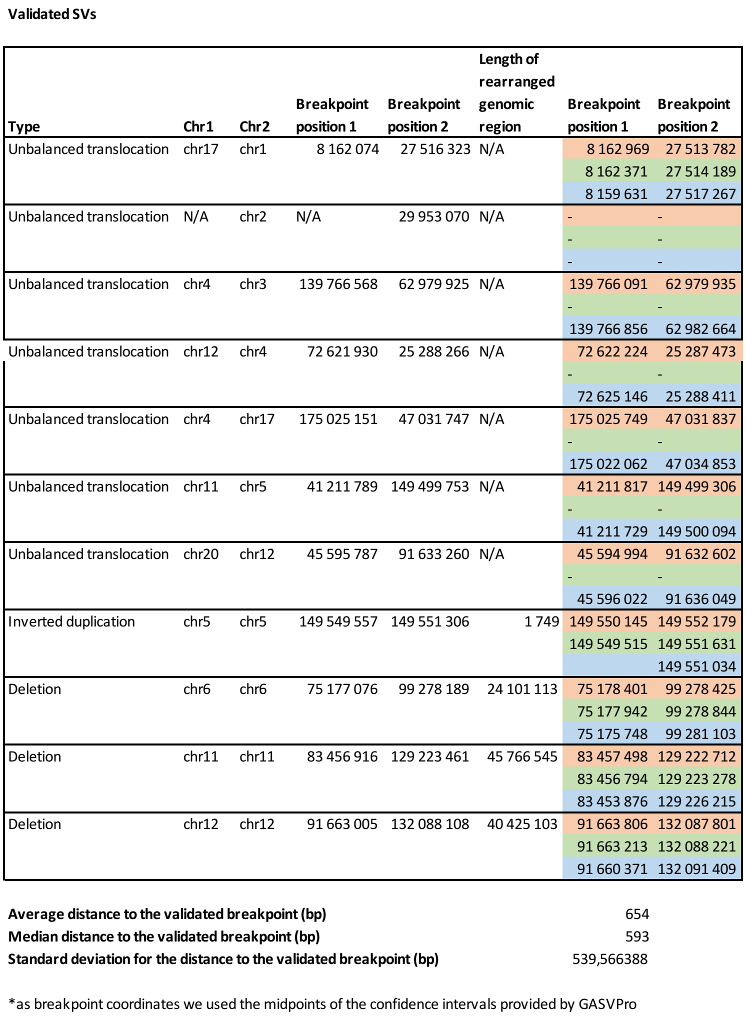
9.3.2 Predicted structural variations

Among the 11 experimentally validated SVs, 10 were successfully detected by SV-Bay, Delly and BreakDancer (see Table 9.4). These three methods missed only one translocation between the *ALK* gene and a repetitive region in a telomere: for this translocation the input data contains only one read pair uniquely mapped to the corresponding peri-telomeric repetitive region. In the future, we plan to improve our approach by taking into account non-uniquely mapped reads. This is expected to improve the sensitivity of predictions. The GasvPro method was able to identify only five out of 11 validated SVs.

The total number of SVs predicted by Delly, GasvPro and BreakDancer was significantly higher than the number of SVs predicted by SV-Bay (62822, 1648 and 5543 vs 733). The total number of predictions in the SV-Bay output was 8 times less than in the output of BreakDancer and 85 times less than in the output of Delly. Thus, although the three tools have equally good recall for the experimental data, SV-Bay has a much better precision. This is explained by the use of a Bayesian probabilistic approach in SV-Bay in addition to clustering of the abnormal fragments performed by all the methods.

9.3.3 SNP6-experiments

For the dataset used in our comparison Curie Institute recently performed experiments to characterize genotype and copy number alterations independently from WGS data. A copy number profile was calculated using Affymetrix SNP 6.0 array for the CLB-GA neuroblastoma cell line. The Gap [PMSL+09] genotyping software was used to detect breakpoints in this profile.



Validated SVs

Type	Chr1	Chr2	Breakpoint position 1	Breakpoint position 2	Length of rearranged genomic region	Breakpoint position 1	Breakpoint position 2
Unbalanced translocation	chr17	chr1	8 162 074	27 516 323	N/A	8 162 969	27 513 782
						8 162 371	27 514 189
						8 159 631	27 517 267
Unbalanced translocation	N/A	chr2	N/A	29 953 070	N/A	-	-
						-	-
						-	-
Unbalanced translocation	chr4	chr3	139 766 568	62 979 925	N/A	139 766 091	62 979 935
						-	-
						139 766 856	62 982 664
Unbalanced translocation	chr12	chr4	72 621 930	25 288 266	N/A	72 622 224	25 287 473
						-	-
						72 625 146	25 288 411
Unbalanced translocation	chr4	chr17	175 025 151	47 031 747	N/A	175 025 749	47 031 837
						-	-
						175 022 062	47 034 853
Unbalanced translocation	chr11	chr5	41 211 789	149 499 753	N/A	41 211 817	149 499 306
						-	-
						41 211 729	149 500 094
Unbalanced translocation	chr20	chr12	45 595 787	91 633 260	N/A	45 594 994	91 632 602
						-	-
						45 596 022	91 636 049
Inverted duplication	chr5	chr5	149 549 557	149 551 306	1 749	149 550 145	149 552 179
						149 549 515	149 551 631
							149 551 034
Deletion	chr6	chr6	75 177 076	99 278 189	24 101 113	75 178 401	99 278 425
						75 177 942	99 278 844
						75 175 748	99 281 103
Deletion	chr11	chr11	83 456 916	129 223 461	45 766 545	83 457 498	129 222 712
						83 456 794	129 223 278
						83 453 876	129 226 215
Deletion	chr12	chr12	91 663 005	132 088 108	40 425 103	91 663 806	132 087 801
						91 663 213	132 088 221
						91 660 371	132 091 409

Average distance to the validated breakpoint (bp)	654
Median distance to the validated breakpoint (bp)	593
Standard deviation for the distance to the validated breakpoint (bp)	539,566388

*as breakpoint coordinates we used the midpoints of the confidence intervals provided by GASVPro

Figure 9.4: SV experimentally validated in the CLB-GA neuroblastoma cell line. Orange color identifies SV-Bay results, green - GASV Pro and blue is BreakDancer.

GAP identified 27 breakpoints in the neuroblastoma cell line genome. Results for considered SV detection tools are presented in Figure 9.5. The copy number profile is shown in black, short vertical bars indicate centromeres; absolute copy numbers identified by GAP are shown in blue. Change points in the copy number profile are shown with long vertical bars (explained with SVs predicted by SV-BAY: red, unexplained: grey). Green question marks indicate copy number changes not explained by each tested tool. Purple question marks correspond to the cases where detected SVs are likely to correspond to false positive predictions.

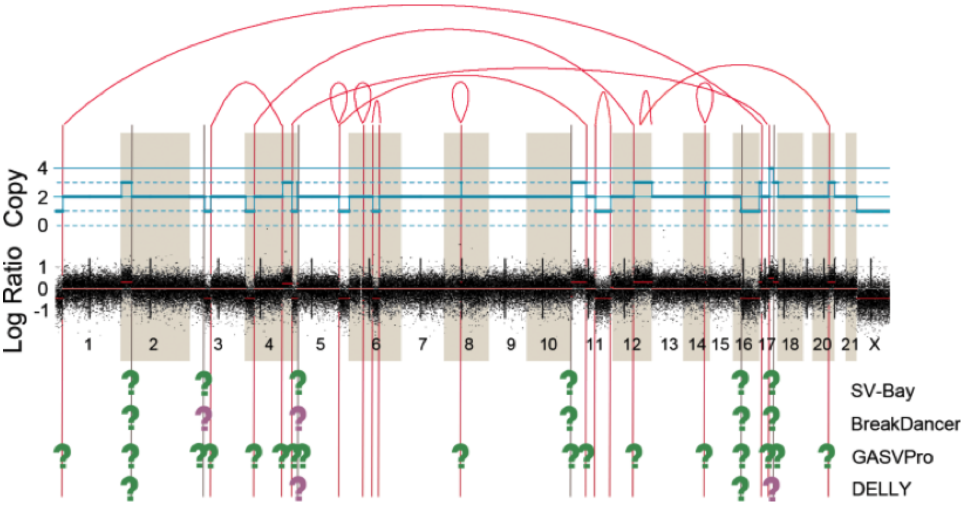


Figure 9.5: Prediction sensitivity on experimental data (neuroblastoma cell line CLB-GA mate-pair dataset).

21 out of the 27 breakpoints predicted by GAP are explained by SVs predicted by SV-BAY. The same 21 breakpoints are also explained by SVs discovered by BREAKDANCER and DELLY. The breakpoints on chromosomes 3 and 10 is predicted only by DELLY. However, the corresponding SV is marked ”LowQual”, which makes this prediction in fact unusable (considering that the total output of DELLY includes around 62 thousand SVs).

In addition, DELLY predicted two SVs that can potentially explain the presence of breakpoints on chromosomes 5 and 17. These SVs were not detected by other tools, but they do not seem to be accurate:

- they are confirmed only by 2 read pairs;
- they have a ”LowQual” tag in the output;
- their type is unbalanced translocation where the second ends are located in chromosomes 17q25 (79Mb) and 18q12 (27Mb). Both regions do not show any copy number change point according to the SNP array analysis.

GasvPro was able to identify SVs corresponding to only 8 breakpoints in the SNP array copy number profile.

9.4 Discussion of the results

The comparison of SV-Bay with other SV calling methods on both simulated and real data demonstrates clear advantages of SV-Bay: it is able to predict at least equal number of true SVs, showing a lower false positive rate.

In this section we discuss several important aspects of the results. We explain how the type of data (mate-pair or paired-end) and copy number variation influence the SV prediction accuracy. Quality of breakpoint resolution of each tool is also covered.

9.4.1 Sequencing technology and coverage

During the data simulation we assumed that the average number of abnormal fragments required to confirm each novel genomic adjacency (physical coverage) is higher for mate-pair data than for paired-end. We estimated the physical coverage of each allele of the simulated tetraploid cancer genome to be approximately 17 and 6 for mate-pair and paired-end data respectively.

Despite the fact that the simulated mate-pair library contains 5 times less reads than the paired-end one, we observed (see Table 9.3) that all tested methods could identify more correct SVs in the mate-pair dataset. This observation supports the common choice of mate-pairs for annotation of structural variants in tumor genomes, even though creating a mate-pair library requires a more elaborate protocol.

9.4.2 Influence of the presence of copy number variation on the SV prediction accuracy

The copy number variation around possible SV position is the main factor, on which the Bayesian approach implemented in SV-Bay is based. The accurate analysis of CNV considering mappability and GC-contents allows to filter out false SVs and thus to reduce the false positive rate. It gives SV-Bay a comparative advantage over the other algorithms.

This approach also gives another effect: the accuracy of SV-Bay may differ for SVs with and without CNV. The sensitivity for SVs with a variation of the copy number, such as indels and linking-reinsertions, should be better than for SVs without CNV, such as inversions or re-insertions.

To check this assumption, the list of SVs predicted by SV-Bay was split into two groups, according to the possible presence of CNV. Two separate precision/recall curves were created. This was done for both simulated paired-end and mate-pair libraries.

For simulated mate-pairs library, 99% of SVs were predicted by SV-Bay; therefore, no variation can be observed between CNV-dependent and CNV-independent SVs. For paired-end library, the prediction rate dramatically varies: it is 40% for CNV-independent validated SVs and 70% for validated CNV-dependent SVs.

9.4.3 Breakpoint resolution

SV-Bay shows a fair accuracy of identification of the exact breakpoint position. Figure 9.4 shows that SV-Bay significantly outperforms BreakDancer, even without using split-reads. The average distance between validated and predicted breakpoints is 654bp vs 1906bp for SV-Bay and BreakDancer. However, GasvPro and Delly provide a better breakpoint resolution by taking into account split-reads. The average distance between validated and predicted breakpoints for these methods is 314bp and 373bp respectively.

SV-Bay does not use split-reads (explained in Chapter 8), but uses a probabilistic approach to choose the most probable breakpoint position. This approach gives reasonably good results. Using split-reads is one of the possibilities to improve SV-Bay in the future.

9.5 Execution time comparison

In this section the runtime of considered tools is compared. The comparison is presented in Table 9.4.

Data type	SV-Bay	BreakDancer	GasvPro	Lumpy	Delly
Mate-pair dataset	1h 55m	32m	298h 47m	N/A	45m
Paired-end dataset	3h 58m	19m	4h 39m	4h02	1h44

Table 9.4: Time of processing the whole genome data for each considered tool.

BreakDancer and Delly are the fastest tools among the five for both paired-end and mate-pair simulated datasets. SV-Bay demonstrates the third best execution time: less than two and four hours for mate-pair and paired-end datasets respectively.

GasvPro needed more than 12 days to analyse the mate-pair dataset. The reason for this may be the long insert size of mate-pair data (more than 4 Kb in our case). As a consequence, the range of all possible breakpoint positions for each SV was extremely large and required a significant amount of time to be analysed. SV-Bay also attempts to predict the most likely breakpoint position for each SV. However, if the interval in which the breakpoint can be possibly located is large, SV-Bay limits the analysis to only several positions equally spaced within the interval. Although this approach worsens the breakpoint detection accuracy, it significantly speeds up the execution time for mate-pair libraries.

Chapter 10

Conclusion and perspectives

10.1 Conclusion

In my thesis work, I introduce SV-Bay, a new computational method and software to detect structural variants from whole genome sequencing mate-pair or paired-end data. The proposed method does not only use information about abnormal read mappings, but also assesses changes in the copy number profile and tries to associate these changes with candidate SVs. The likelihood of each novel genomic adjacency is evaluated using a probabilistic Bayesian model.

SV-Bay Bayesian model takes into account depth of coverage by normal reads and abnormalities in read pair mappings. To estimate the model likelihood, SV-Bay considers GC-content, ploidy and read mappability of the genome, thus making important corrections to the expected fragment count. This ensures sensitivity and selectivity, as many artefact clusters of mismapped read pairs are discarded. Indeed, in comparison with other methods, SV-Bay demonstrates a noticeably better SV detection accuracy.

SV-Bay was validated on mate-pair and paired-end simulated datasets along with an experimental mate-pair dataset for the CLB-GA neuroblastoma cell line. A comparison of SV-Bay with BreakDancer, Lumpy, Delly and GasvPro demonstrated its superior performance on simulated and experimental datasets. SV-Bay has a better prediction accuracy in terms of sensitivity and false positive detection rate. Moreover, for the experimental neuroblastoma dataset, SV-Bay predictions explained 78% of breakpoints in the copy number profile, calculated using an Affymetrix SNP6.0 array, providing significantly less candidate SVs than other tools.

For the detection of somatic variants, SV-Bay makes use of a matched normal sample when it is available. SV-Bay also allows to annotate discovered genomic adjacencies according to their type and, where possible, assembles detected genomic adjacencies into complex SVs such as balanced translocations, co-amplifications, linking insertions, tandem duplications with inversion, etc. Notably, SV-Bay is the only tool which is able to assemble the co-amplification events, which are important for many tests related to cancer identification (for example, Mycn amplification).

SV-Bay allows the user to skip several data post-processing steps. One example is filtering out links with low number of fragments that do not correspond to copy number changes. Such links are discarded by SV-Bay Bayesian approach. In other tools they contaminate the output making it harder to interpret the results. Another example is filtering out events present simultaneously in the tumor data and the matched normal control (artifacts and germline SVs). Using clustering results for control sample, SV-Bay discards structural variants knowingly not related to cancer development. In several other tools such filtering is not available.

Another important part of my thesis work is the preparation of a novel exhaustive catalogue of SV types. This catalogue is based on the previous publications and experience in cancer data research, accumulated in Institute Curie. It is the most comprehensive SV classification existing to date. I provide an illustrated list of seventeen structural variant types, including seven SV types ignored by the existing SV calling algorithms. This list is used in SV-Bay tool to automatically annotate and assemble predicted SVs. Previously existing tools did not provide such functionality, making it necessary to manually analyze the output and find corresponding complex SVs. Considering big number of SVs in cancer genome, such manual analysis can introduce errors and often is not possible at all. Thus, automatic SV assembly algorithm is an important advantage of SV-Bay.

The advantages of the proposed method allow SV-Bay to play an important role in simplifying the cancer rearrangement detection and understanding the mechanisms of cancer development.

10.2 Perspectives

Although the current version of SV-Bay algorithm, presented in this thesis, outperforms other considered tools in terms of SV detection recall and precision, several directions for further method improvement can be proposed. Possible improvements are related to both biological and algorithmic aspects. First the possible development directions from the biological point of view are discussed. 10.1

Like other methods, SV-Bay is tolerant to a certain degree to contamination of the tumor sample by normal cells. In figure 10.1 the average tumor tissue is shown. Normal and tumor cells are shown in brown and blue colors respectively. Contamination by normal cells can be detected by various pattern matching algorithms.

In the future, it is possible to extend the SV-Bay model to handle different normal cell contamination levels. The most straightforward way to do so is to request the contamination rate from the user and use this coefficient in the probabilistic model.

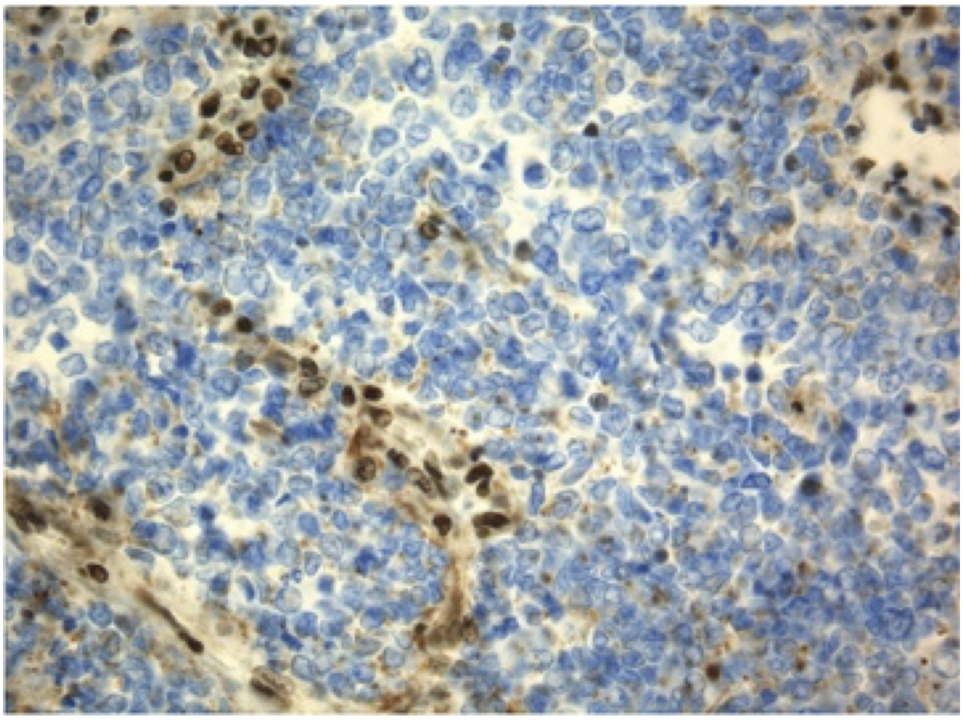


Figure 10.1: Contamination tumor tissue by normal cells. In blue colour are shown cancer cells and in brown - normal cells.

Another direction that can be considered is the detection of sub-clonal events. In tumor tissue different SVs can be presented in different cells. In is illustrated in figure 10.2.

Sub-clonal mutations are challenging to detect. The DOC method is not directly applicable, as the proportion of the expected and observed fragments number should be calculated with respect to the number of cells containing the considered structural variant. By now no methods exist to detect cells with sub-clonal events. A probabilistic approach can be proposed, which increases probability score of a structural variant based on the number of confirmations, such as a signature, split-reads presence, association with a complex SV and annotation with given diagnosis.

There are also several ways of SB-Bay improvement from the algorithmic point of view.

SV-Bay does not use split-reads to improve the resolution of predicted breakpoints. There are two main reasons for this. First, the read coverage on breakpoints must be sufficiently high. This is achieved only for paired-end libraries, whereas SV-Bay was designed to also be applicable to mate-pair data. Second, structural variants in cancer often occur in low

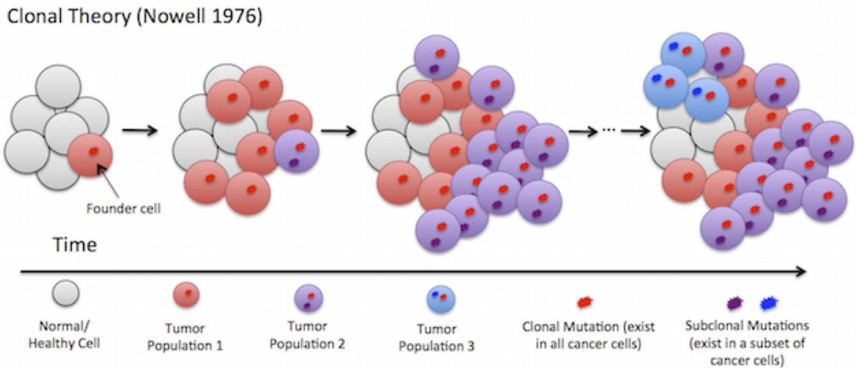

Clonal Theory (Nowell 1976)
Founder cell
Time
Normal/ Healthy Cell
Tumor Population 1
Tumor Population 2
Tumor Population 3
Clonal Mutation (exist in all cancer cells)
Subclonal Mutations (exist in a subset of cancer cells)

Figure 10.2: Sub-clonal event. Sub-clonal SVs are presented in violet and blue coloured cells. Figure is created by Layla Oesper

mappability repetitive regions or regions that have partial homology. These incidents reduce the capacity of read mappers to align correctly reads coming from SV junctions. However, in some cases, considering split-reads can improve the power of the probabilistic model, especially for paired-end data. If the breakpoint is covered by a split-read, this approach also allows for a breakpoint resolution at the single nucleotide level. Thus, adding split-reads support is a perspective direction of the method development.

The current version of SV-Bay analyzes only uniquely mapped fragments. But, according to the known annotated SVs in real data, rearrangements often occur in repetitive regions, such as telomeres or centromeres. To be able to detect such SVs it is necessary to take into account fragments that were mapped with multiple matches. This approach increases the algorithm complexity and leads to performance degradation, but allows to increase the sensitivity of the method.

Finally, the current version of SV-Bay analyzes only one tumor/normal pair at once. One of the interesting possible extensions to the method is to add the ability to analyze several tumor datasets extracted from the same patient in order to increase the sensitivity of SV detection.

Bibliography

[AD04] Schefler AC. Abramson DH. Update on retinoblastoma. *Retina*, 2004.

[BBPL^{+}09] Rochelle Bagatell, Maja Beck-Popovic, Wendy B. London, Yang Zhang, Andrew D J. Pearson, Katherine K. Matthay, Tom Monclair, Peter F. Ambros, Susan L. Cohn, and International Neuroblastoma Risk Group . Significance of mycn amplification in international neuroblastoma staging system stage 1 and 2 neuroblastoma: a report from the international neuroblastoma risk group database. *J Clin Oncol*, 27(3):365–370, Jan 2009.

[BS12] Yuval Benjamini and Terence P. Speed. Summarizing and correcting the GC content bias in high-throughput sequencing. *Nucleic Acids Research*, 40(10), 2012.

[BZB^{+}11] Valentina Boeva, Andrei Zinovyev, Kevin Bleakley, Jean-Philippe Vert, Isabelle Janoueix-Lerosey, Olivier Delattre, and Emmanuel Barillot. Control-free calling of copy number alterations in deep-sequencing data using gc-content normalization. *Bioinformatics*, 27(2):268–269, Jan 2011.

[CKN^{+}10] Helena Carén, Hanna Kryh, Maria Nethander, Rose-Marie Sjöberg, Catarina Träger, Staffan Nilsson, Jonas Abrahamsson, Per Kogner, and Tommy Martinsson. High-risk neuroblastoma tumors with 11q-deletion display a poor prognostic, chromosome instability phenotype with later onset.

Proceedings of the National Academy of Sciences, 107(9):4323–4328, 2010.

[CLBM$^+$13] Alex Cazes, Caroline Louis-Brennetot, Pierre Mazot, Florent Dingli, Brangre Lombard, Valentina Boeva, Romain Daveau, Julie Cappo, Valrie Combaret, Gudrun Schleiermacher, Stphanie Jouannet, Sandrine Ferrand, Galle Pierron, Emmanuel Barillot, Damarys Loew, Marc Vigny, Olivier Delattre, and Isabelle Janoueix-Lerosey. Characterization of rearrangements involving the alk gene reveals a novel truncated form associated with tumor aggressiveness in neuroblastoma. *Cancer Res*, 73(1):195–204, Jan 2013.

[CWM$^+$09] Ken Chen, John W. Wallis, Michael D. McLellan, David E. Larson, Joelle M. Kalicki, Craig S. Pohl, Sean D. McGrath, Michael C. Wendl, Qunyuan Zhang, Devin P. Locke, Xiaoqi Shi, Robert S. Fulton, Timothy J. Ley, Richard K. Wilson, Li Ding, and Elaine R. Mardis. Breakdancer: an algorithm for high-resolution mapping of genomic structural variation. *Nat Methods*, 6(9):677–681, Sep 2009.

[DEM$^+$12] Thomas Derrien, Jordi Estell, Santiago Marco Sola, David G. Knowles, Emanuele Raineri, Roderic Guig, and Paolo Ribeca. Fast computation and applications of genome mappability. *PLoS One*, 7(1):e30377, 2012.

[DM11] Bell Stephane. DePamphilis Melvin. *Genome duplication.* New York: Garland Science, 2011.

[dSK$^+$97] M. de Lima, S. S. Strom, M. Keating, H. Kantarjian, S. Pierce, S. O'Brien, E. Freireich, and E. Estey. Implications of potential cure in acute myelogenous leukemia: development of subsequent cancer and return to work. *Blood*, 90(12):4719–4724, Dec 1997.

[ETB$^+$13] Gergia Escarams, Cristian Tornador, Laia Bassaganyas, Raquel Rabionet, Jose M C.

Tubio, Alexander Martnez-Fundichely, Mario Cceres, Marta Gut, Stephan Ossowski, and Xavier Estivill. Pesv-fisher: identification of somatic and non-somatic structural variants using next generation sequencing data. *PLoS One*, 8(5):e63377, 2013.

[FKJ12] Josep V. Forment, Abderrahmane Kaidi, and Stephen P. Jackson. Chromothripsis and cancer: causes and consequences of chromosome shattering. *Nat Rev Cancer*, 12(10):663–670, Oct 2012.

[GWNL00] Z. Gu, H. Wang, A. Nekrutenko, and W. H. Li. Densities, length proportions, and other distributional features of repetitive sequences in the human genome estimated from 430 megabases of genomic sequence. *Gene*, 259(1-2):81–88, Dec 2000.

[HA11] Donald F. Conrad Jonathan E.M. Keebler Mark A. DePristo Sarah J. Lindsay Yujun Zhang Ferran Cassals Youssef Idaghdour Chris L. Hartl Carlos Torroja Kiran V. Garimella Martine Zilversmit Reed Cartwright Guy Rouleau Mark Daly Eric A. Stone Matthew E. Hurles and Philip Awadalla. Variation in genome-wide mutation rates within and between human families. *Ann Intern Med*, June 2011.

[HAES09] Fereydoun Hormozdiari, Can Alkan, Evan E. Eichler, and S Cenk Sahinalp. Combinatorial algorithms for structural variation detection in high-throughput sequenced genomes. *Genome Res*, 19(7):1270–1278, Jul 2009.

[HBT13] Ayat Hatem, Doruk Bozda, Amanda E. Toland, and mit V. atalyrek. Benchmarking short sequence mapping tools. *BMC Bioinformatics*, 14:184, 2013.

[HHD$^+$10] Fereydoun Hormozdiari, Iman Hajirasouliha, Phuong Dao, Faraz Hach, Deniz Yorukoglu, Can Alkan, Evan E. Eichler, and S Cenk

Sahinalp. Next-generation variationhunter: combinatorial algorithms for transposon insertion discovery. *Bioinformatics*, 26(12):i350–i357, Jun 2010.

[HKNM11] Robert E. Handsaker, Joshua M. Korn, James Nemesh, and Steven A. McCarroll. Discovery and genotyping of genome structural polymorphism by sequencing on a population scale. *Nat Genet*, 43(3):269–276, Mar 2011.

[HLL08] Z. Harchaoui and C. Lvy-Leduc. Catching change-points with lasso. *Adv. Neural Inform. Process. Syst.*, 20:617624., 2008.

[HYS+12] Xuesong Hu, Jianying Yuan, Yujian Shi, Jianliang Lu, Binghang Liu, Zhenyu Li, Yanxiang Chen, Desheng Mu, Hao Zhang, Nan Li, Zhen Yue, Fan Bai, Heng Li, and Wei Fan. pirs: Profile-based illumina pair-end reads simulator. *Bioinformatics*, 28(11):1533–1535, Jun 2012.

[IJLB+15] D Iakovishina, Isabelle Janoueix-Lerosey, Emmanuel Barillot, Mireille Regnier, and Valentina Boeva. SV-Bay: structural variant detection in cancer genomes using a Bayesian approach with correction for GC-content and read mappability. to appear., 2015.

[JWB12] Yue Jiang, Yadong Wang, and Michael Brudno. Prism: pair-read informed split-read mapping for base-pair level detection of insertion, deletion and structural variants. *Bioinformatics*, 28(20):2576–2583, Oct 2012.

[KAM+09] Jan O. Korbel, Alexej Abyzov, Xinmeng Jasmine Mu, Nicholas Carriero, Philip Cayting, Zhengdong Zhang, Michael Snyder, and Mark B. Gerstein. Pemer: a computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data. *Genome Biol*, 10(2):R23, 2009.

[LCQH14] Ryan M. Layer, Colby Chiang, Aaron R. Quinlan, and Ira M. Hall. Lumpy: a probabilistic framework for structural variant discovery. *Genome Biol*, 15(6):R84, 2014.

[LD09] Heng Li and Richard Durbin. Fast and accurate short read alignment with burrows-wheeler transform. *Bioinformatics*, 25(14):1754–1760, Jul 2009.

[Lee07] Kevin A. W. Lee. Ewings family oncoproteins: drunk, disorderly and in search of partners. *Cell Research*, 2007.

[LH00] Zipursky SL et al. Lodish H, Berk A. *Molecular Cell Biology.* New York: W. H. Freeman, 2000.

[LRD08] Heng Li, Jue Ruan, and Richard Durbin. Mapping short dna sequencing reads and calling variants using mapping quality scores. *Genome Res*, 18(11):1851–1858, Nov 2008.

[Lup98] James R Lupski. Charcot-marie-tooth polyneuropathy: Duplication, gene dosage, and genetic heterogeneity. *Pediatric Research*, 45:159–165, 1998.

[LV86] Gad M Landau and Uzi Vishkin. Efficient string matching with k mismatches. *Theoretical Computer Science*, 43:239–249, 1986.

[MGB$^+$14] Valent Moncunill, Santi Gonzalez, Slvia Be, Lise O. Andrieux, Itziar Salaverria, Cristina Royo, Laura Martinez, Montserrat Puiggrs, Maia Segura-Wang, Adrian M. Sttz, Alba Navarro, Romina Royo, Josep L. Gelp, Ivo G. Gut, Carlos Lpez-Otn, Modesto Orozco, Jan O. Korbel, Elias Campo, Xose S. Puente, and David Torrents. Comprehensive characterization of complex structural variations in cancer by directly comparing genome sequence reads. *Nat Biotechnol*, 32(11):1106–1112, Nov 2014.

[MSB09] Paul Medvedev, Monica Stanciu, and Michael Brudno. Computational methods for discovering structural variation with next-generation sequencing. *Nat Methods*, 6(11 Suppl):S13–S20, Nov 2009.

[MWS+11] Ryan E. Mills, Klaudia Walter, Chip Stewart, Robert E. Handsaker, Ken Chen, Can Alkan, Alexej Abyzov, Seungtai Chris Yoon, Kai Ye, R Keira Cheetham, Asif Chinwalla, Donald F. Conrad, Yutao Fu, Fabian Grubert, Iman Hajirasouliha, Fereydoun Hormozdiari, Lilia M. Iakoucheva, Zamin Iqbal, Shuli Kang, Jeffrey M. Kidd, Miriam K. Konkel, Joshua Korn, Ekta Khurana, Deniz Kural, Hugo Y K. Lam, Jing Leng, Ruiqiang Li, Yingrui Li, Chang-Yun Lin, Ruibang Luo, Xinmeng Jasmine Mu, James Nemesh, Heather E. Peckham, Tobias Rausch, Aylwyn Scally, Xinghua Shi, Michael P. Stromberg, Adrian M. Sttz, Alexander Eckehart Urban, Jerilyn A. Walker, Jiantao Wu, Yujun Zhang, Zhengdong D. Zhang, Mark A. Batzer, Li Ding, Gabor T. Marth, Gil McVean, Jonathan Sebat, Michael Snyder, Jun Wang, Kenny Ye, Evan E. Eichler, Mark B. Gerstein, Matthew E. Hurles, Charles Lee, Steven A. McCarroll, Jan O. Korbel, and 1000 Genomes Project . Mapping copy number variation by population-scale genome sequencing. *Nature*, 470(7332):59–65, Feb 2011.

[Mye86] Eugene W Myers. Ano (nd) difference algorithm and its variations. *Algorithmica*, 1(1-4):251–266, 1986.

[NRR11] Moshe Oren Noa Rivlin, Ran Brosh and Varda Rotter. Mutations in the p53 tumor suppressor gene. *Genes Cancer*, June 2011.

[OKM+92] J. D. Oliner, K. W. Kinzler, P. S. Meltzer, D. L. George, and B. Vogelstein. Amplification of a gene encoding a p53-associated protein in human sarcomas. *Nature*, 358(6381):80–83, Jul 1992.

[ORA$^+$12] Layla Oesper, Anna Ritz, Sarah J. Aerni, Ryan Drebin, and Benjamin J. Raphael. Reconstructing cancer genomes from paired-end sequencing data. *BMC Bioinformatics*, 13 Suppl 6:S10, 2012.

[PF00] Giovanni Manzini Paolo Ferragina. Opportunistic data structures with applications. In *Proceedings. 41st Annual Symposium on Foundations of Computer Science*, 2000.

[PMSL$^+$09] Tatiana Popova, Elodie Mani, Dominique Stoppa-Lyonnet, Guillem Rigaill, Emmanuel Barillot, and Marc Henri Stern. Genome alteration print (gap): a tool to visualize and mine complex cancer genomic profiles obtained by snp arrays. *Genome Biol*, 10(11):R128, 2009.

[PSO$^+$10] Erin D. Pleasance, Philip J. Stephens, Sarah O'Meara, David J. McBride, Alison Meynert, David Jones, Meng-Lay Lin, David Beare, King Wai Lau, Chris Greenman, Ignacio Varela, Serena Nik-Zainal, Helen R. Davies, Gonzalo R. Ordoez, Laura J. Mudie, Calli Latimer, Sarah Edkins, Lucy Stebbings, Lina Chen, Mingming Jia, Catherine Leroy, John Marshall, Andrew Menzies, Adam Butler, Jon W. Teague, Jonathon Mangion, Yongming A. Sun, Stephen F. McLaughlin, Heather E. Peckham, Eric F. Tsung, Gina L. Costa, Clarence C. Lee, John D. Minna, Adi Gazdar, Ewan Birney, Michael D. Rhodes, Kevin J. McKernan, Michael R. Stratton, P Andrew Futreal, and Peter J. Campbell. A small-cell lung cancer genome with complex signatures of tobacco exposure. *Nature*, 463(7278):184–190, Jan 2010.

[QZ11] Ji Qi and Fangqing Zhao. ingap-sv: a novel scheme to identify and visualize structural variation from paired end mapping data. *Nucleic Acids Res*, 39(Web Server issue):W567–W575, Jul 2011.

[RKMT03] MD; Brian J. Druker MD; Razelle Kurzrock, MD; Hagop M. Kantarjian and MD Moshe Talpaz. Philadelphia chromosomepositive leukemias: From basic mechanisms to molecular therapeutics. *Ann Intern Med*, May 2003.

[RZS$^+$12] Tobias Rausch, Thomas Zichner, Andreas Schlattl, Adrian M. Sttz, Vladimir Benes, and Jan O. Korbel. Delly: structural variant discovery by integrated paired-end and split-read analysis. *Bioinformatics*, 28(18):i333–i339, Sep 2012.

[SCE$^+$07] Manabu Soda, Young Lim Choi, Munehiro Enomoto, Shuji Takada, Yoshihiro Yamashita, Shunpei Ishikawa, Shin ichiro Fujiwara, Hideki Watanabe, Kentaro Kurashina, Hisashi Hatanaka, Masashi Bando, Shoji Ohno, Yuichi Ishikawa, Hiroyuki Aburatani, Toshiro Niki, Yasunori Sohara, Yukihiko Sugiyama, and Hiroyuki Mano. Identification of the transforming eml4-alk fusion gene in non-small-cell lung cancer. *Nature*, 448:561–566, 2007.

[Sch]

[SGF$^+$11] Philip J. Stephens, Chris D. Greenman, Beiyuan Fu, Fengtang Yang, Graham R. Bignell, Laura J. Mudie, Erin D. Pleasance, King Wai Lau, David Beare, Lucy A. Stebbings, Stuart McLaren, Meng-Lay Lin, David J. McBride, Ignacio Varela, Serena Nik-Zainal, Catherine Leroy, Mingming Jia, Andrew Menzies, Adam P. Butler, Jon W. Teague, Michael A. Quail, John Burton, Harold Swerdlow, Nigel P. Carter, Laura A. Morsberger, Christine Iacobuzio-Donahue, George A. Follows, Anthony R. Green, Adrienne M. Flanagan, Michael R. Stratton, P Andrew Futreal, and Peter J. Campbell. Massive genomic rearrangement acquired in a single catastrophic event during cancer development. *Cell*, 144(1):27–40, Jan 2011.

[SHB$^+$14] Jan Schrder, Arthur Hsu, Samantha E. Boyle, Geoff Macintyre, Marek Cmero, Richard W. Tothill, Ricky W. Johnstone, Mark Shackleton, and Anthony T. Papenfuss. Socrates: identification of genomic rearrangements in tumour genomes by realigning soft clipped reads. *Bioinformatics*, Jan 2014.

[SML$^+$09] Philip J. Stephens, David J. McBride, Meng-Lay Lin, Ignacio Varela, Erin D. Pleasance, Jared T. Simpson, Lucy A. Stebbings, Catherine Leroy, Sarah Edkins, Laura J. Mudie, Chris D. Greenman, Mingming Jia, Calli Latimer, Jon W. Teague, King Wai Lau, John Burton, Michael A. Quail, Harold Swerdlow, Carol Churcher, Rachael Natrajan, Anieta M. Sieuwerts, John W M. Martens, Daniel P. Silver, Anita Langerd, Hege E G. Russnes, John A. Foekens, Jorge S. Reis-Filho, Laura van 't Veer, Andrea L. Richardson, Anne-Lise Brresen-Dale, Peter J. Campbell, P Andrew Futreal, and Michael R. Stratton. Complex landscapes of somatic rearrangement in human breast cancer genomes. *Nature*, 462(7276):1005–1010, Dec 2009.

[SOP$^+$12] Suzanne S. Sindi, Selim Onal, Luke C. Peng, Hsin-Ta Wu, and Benjamin J. Raphael. An integrative probabilistic model for identification of structural variation in sequencing data. *Genome Biol*, 13(3):R22, 2012.

[TEER14] Kathrin Trappe, Anne-Katrin Emde, Hans-Christian Ehrlich, and Knut Reinert. Gustaf: Detecting and correctly classifying svs in the ngs twilight zone. *Bioinformatics*, 30(24):3484–3490, Dec 2014.

[TS12] Todd J. Treangen and Steven L. Salzberg. Repetitive dna and next-generation sequencing: computational challenges and solutions. *Nat Rev Genet*, 13(1):36–46, Jan 2012.

[VB13] Stphanie Jouannet Romain Daveau Valrie Combaret Valentina Boeva. Breakpoint features of genomic rearrangements in neuroblastoma with unbalanced translocations and chromothripsis. *PLoS One*, Aug 2013.

[VBTPKBPCJCGSIJLOD12] Emmanuel Barillot Valentina Boeva Tatiana Popova Kevin Bleakley Pierre Chiche Julie Cappo Gudrun Schleiermacher Isabelle Janoueix-Lerosey Olivier Delattre. Control-freec: a tool for assessing copy number and allelic content using next-generation sequencing data. *Bioinformatics*, Feb 2012.

[WKSA10] Kim Wong, Thomas M. Keane, James Stalker, and David J. Adams. Enhanced structural variant and breakpoint detection using svmerge by integration of multiple detection methods and local assembly. *Genome Biol*, 11(12):R128, 2010.

[WME$^+$11] Jianmin Wang, Charles G. Mullighan, John Easton, Stefan Roberts, Sue L. Heatley, Jing Ma, Michael C. Rusch, Ken Chen, Christopher C. Harris, Li Ding, Linda Holmfeldt, Debbie Payne-Turner, Xian Fan, Lei Wei, David Zhao, John C. Obenauer, Clayton Naeve, Elaine R. Mardis, Richard K. Wilson, James R. Downing, and Jinghui Zhang. Crest maps somatic structural variation in cancer genomes with base-pair resolution. *Nat Methods*, 8(8):652–654, Aug 2011.

[YLG$^+$13] Lixing Yang, Lovelace J. Luquette, Nils Gehlenborg, Ruibin Xi, Psalm S. Haseley, Chih-Heng Hsieh, Chengsheng Zhang, Xiaojia Ren, Alexei Protopopov, Lynda Chin, Raju Kucherlapati, Charles Lee, and Peter J. Park. Diverse mechanisms of somatic structural variations in human cancer genomes. *Cell*, 153(4):919–929, May 2013.

[YXM$^+$09] Seungtai Yoon, Zhenyu Xuan, Vladimir Makarov, Kenny Ye, and Jonathan Sebat. Sensitive and accurate detection of copy number variants using read depth of coverage. *Genome Res*, 19(9):1586–1592, Sep 2009.

[ZBJL$^+$10] Bruno Zeitouni, Valentina Boeva, Isabelle Janoueix-Lerosey, Sophie Loeillet, Patricia Legoix-né, Alain Nicolas, Olivier Delattre, and Emmanuel Barillot. Svdetect: a tool to identify genomic structural variations from paired-end and mate-pair sequencing data. *Bioinformatics*, 26(15):1895–1896, 2010.