Tel-Aviv University
Raymond and Beverly Sackler
Faculty of Exact Sciences
The Blavatnik School of Computer Science

Reconstructing cancer karyotypes from paired-end reads and copy number data

Thesis submitted towards the degree of Master of Science in the School of Computer Science, Tel-Aviv University.

By

Rami Eitan

The research work for this thesis has been carried out at Tel-Aviv University under the supervision of

Prof. Ron Shamir

June 2016

Tel-Aviv University
Raymond and Beverly Sackler
Faculty of Exact Sciences
The Blavatnik School of Computer Science

Reconstructing cancer karyotypes from paired-end reads and copy number data

Thesis submitted towards the degree of Master of Science in the School of Computer Science, Tel-Aviv University.

By

Rami Eitan

The research work for this thesis has been carried out at Tel-Aviv University under the supervision of

Prof. Ron Shamir

June 2016

Acknowledgments

Firstly, I would like to deeply thank my advisor and PI Prof Ron Shamir for the tremendous support and guidance throughout this research. His patience and diligence were instrumental in teaching me how to conduct a thorough and methodical research.

I would also like to extend my thanks to all other members of the computational genomics group, whom I had the privilege to discuss, brainstorm and learn from while spending time in the lab. I would like to give special thanks to David (Didi) Amar for teaching me a great deal about statistics and data analysis methods, Ron Zeira for help with graph theory related questions and Dvir Netanely for his support and various helpful comments. I would like to thank Roy Kasher from Google NYC whose suggestions on implementation helped decrease running time significantly. I also would like to thank Nir Atias from the lab of Roded Sharan in the Edmond J. Safra Bioinformatics Center in Tel-Aviv University, as his expertise on ILP and the IBM CPLEX solver proved invaluable.

Abstract

Cancer genomes change during the disease progression in a series of rearrangements. These rearrangements include, among others, segmental deletions, insertions, translocations and inversions.

The result is a highly complex, patient-specific cancer karyotype. Using high-throughput technologies of deep sequencing and microarrays it is possible to interrogate a cancer genome and produce chromosomal copy number profiles and a list of breakpoints ("jumps") relative to the normal genome. This information is very detailed but local in nature, and does not give the overall picture of the cancer genome. One of the basic challenges in cancer genome research is to use such information to infer the cancer karyotype.

We present here an algorithmic approach, based on graph theory and integer linear programming, that receives segmental copy number and breakpoint data as input and produces a cancer karyotype that is most concordant with them. We used simulations to evaluate the utility of our approach, and applied it to real data.

By using a simulation model, we were able to estimate the correctness and robustness of the algorithm in a spectrum of scenarios. Under the conditions of our base scenario, designed according to observations in real data, the algorithm correctly inferred 69% of the karyotypes. However, when using correctness metrics that are less strict and account for incomplete and noisy data, 87% of the tested cases were correct. Furthermore, in scenarios where the data were very clean and complete, accuracy was shown to be between 90%-100%. Some examples of analysis of real data, and the reconstructed karyotypes suggested by our algorithm, are also presented.

Contents

1 Introduction

In this chapter we will review the basic background required to understand the motivation behind this thesis. We will also review the relevant work that has been done on the computational and biological problems we deal with.

1.1 Cancer rearrangements

In this section we introduce the basic concepts behind tumor genomic rearrangements and review some of the work done in the field relevant to this thesis.

The current understanding of cancer suggests that it is a disease driven by somatic mutations that accumulate in the genome, within a certain tissue, during the lifetime of an individual. These mutations vary in size and effect. They can be small, e.g., single nucleotide mutations, or large structural variations caused by rearrangements such as deletions, inversions, tandem duplications and chromosomal translocations, or duplication and losses of entire chromosomes [1]. Over time these rearrangements accumulate and result in cells that have a more different DNA sequence compared to healthy cells.

The link between chromosomal abnormalities and cancer was first suggested by Boveri in 1914 [2], But it wasn't until the discovery of the *Philadelphia Chromosome* in 1960 by Nowell and Hungerford [3] that this hypothesis was confirmed. The Philadelphia chromosome is an abnormal chromosome that exists in 95% of the cancer patients with chronic myelogenous leukemia (CML). It was discovered in 1973 to be the result of a reciprocal translocation between chromosomes 9 and 22 [4] that results in the fusion gene BCR-ABL, composed of the BCR gene from chromosome 22 and the ABL gene from chromosome 9 [4].

More recently it was discovered that a small inversion in chromosome 2 results in a new fusion gene called EML4-ALK [5]. This gene was found in a subset of lung cancer patients, and shown to give rise to tumors when injected to mice. A known ALK inhibitor was also shown to significantly reduce growth of the aberrant cells and thus a promising candidate for therapeutic target [5].

Cancer genomes are often described in the form of *karyotypes*. A *karyotype* is a high level description of the genome as a set of chromosomes and their number of copies of each (Figure 1.1). Normal karyotypes have two copies of each chromosome 1 to 22 and the sex chromosomes - in cancer karyotypes some chromosomes may contain fragments of several normal chromosomes. Generally, some types of cancer can be characterized by a unique genomic rearrangement. This information is used clinically for diagnostic and therapeutic purposes.

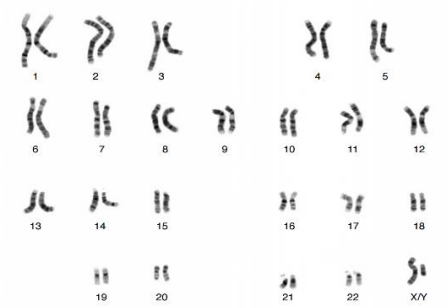

Figure 1.1: An example of a normal human male karyotype. Source: the National Human Genome Research Institute.

1.1.1 Basic types of rearrangements

Most rearrangements that happen during the progression of the disease can be categorized into the canonical forms of deletion, tandem duplication, inversion, translocation, and deletion and duplication of entire chromosomes.

A *deletion* is characterized by a missing segment of a chromosome, a *tandem duplication* happens when part of the chromosome is duplicated and thus two copies of a segment appear where normally there would only be one. An *inversion* occurs when a segment of a chromosome is reversed relative to its original orientation (Figure 1.2).

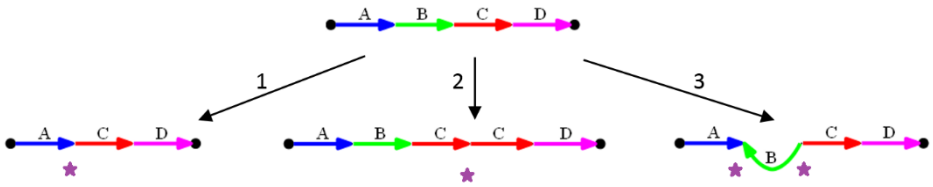

Figure 1.2: Basic types of rearrangements. (1) Deletion: segment B of the normal chromosome is deleted. (2) Tandem duplication: segment C duplicates and repeats. (3) Inversion: segment B is inversed. Stars indicate breakpoints.

A translocation happens when two different chromosomes appear to "switch" end segments. Translocations can be referred to as *balanced* or *unbalanced* and as *reciprocal* and *non-reciprocal*. A translocation is said to be balanced when the two chromosomes exchange about the same amount of genetic material, or unbalanced if the exchange is uneven. A non-reciprocal translocation occur when the transfer of chromosomal material is one way. When a tail segment of each chromosome appears in the other chromosome, the translocation is called reciprocal (Figure 1.3).

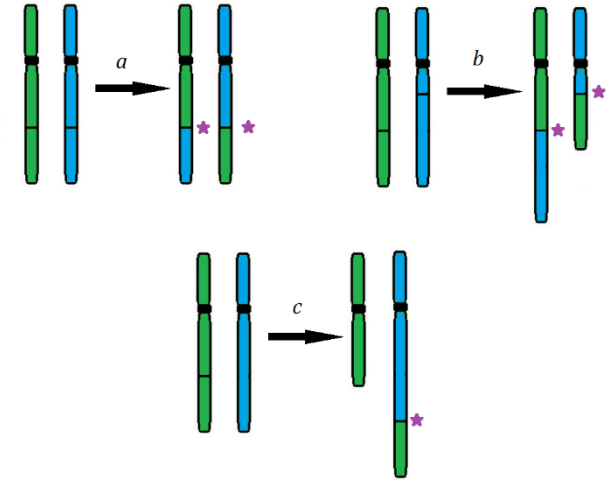

Figure 1.3: Inter-chromosomal translocations. Reciprocal balanced (a) and unbalanced (b) translocations. A non-reciprocal translocation happens in (c). Stars indicate breakpoints.

1.1.2 Breakpoints

The molecular mechanisms that cause somatic genome rearrangements are still the focus of investigation. The main paradigm is that a genome rearrangement occurs when one or more

chromosomes break and a following joining event reassembles the fragments in a different order.

When the double stranded DNA sequence is broken in more than one location, the joining event may fuse the wrong ends together, effectively rearranging the genome. A *breakpoint* is defined as a genomic location where the normal DNA sequence is interrupted and two non-adjacent sequences segments are consecutive due to a joining event. Hence two DNA segments that are distant in the normal genome will be adjacent on the tumor genome at the location of the breakpoint. A breakpoint can be considered as the most basic unit of rearrangement. The stars in Figure 1.2 and Figure 1.3 indicate breakpoints.

Examples of processes that generate single breakpoints include breakage- fusion-bridge cycles, nonhomologous end joining and homologous recombination-mediated repair [6]. If two breakpoints occur on the genome they create four different ends and three possible rejoining scenarios: If the two ends that the segment lied between are joined together leaving it out it will cause a deletion event. If the segment is reversed and fused it will create an inversion event (Figure 1.2). If the two breaks happen on two different chromosomes then a wrong joining event results in a translocation (Figure 1.3).

1.1.3 More rearrangements types

In addition to the basic rearrangement events, other types of genomic rearrangements have also been suggested to explain the development of tumors.

One such rearrangement is the Breakage-Fusion Bridge (BFB). The BFB mechanism is one of the driving mechanisms of evolution. It was first proposed by Barbara McClintock in 1938 to explain observations on chromosomes in maize [7], [8], and has since been identified in other organisms [9]–[13].

Breakage-Fusion Bridge happens when a chromosome loses one of its telomeres, along with an end segment. When the chromosome replicates, its sister chromatids fuse together and forms a bridge. The two centromeres of the fused chromosome migrate to opposing ends during anaphase and when the doubled chromosome breaks one of the daughter cells receives a doubled and truncated chromosome with missing telomere that can give rise to another BFB cycle (Figure 1.4).

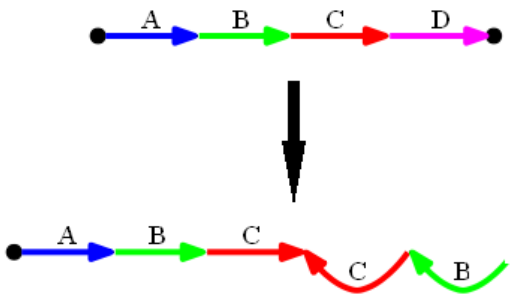

Figure 1.4: A BFB rearrangement. The chromosome breaks between C and D and loses D and its telomere. After replication the two ends at C fuse together and after anaphase the chromosome breaks between A and B, creating a new rearranged chromosome.

BFB's are known to amplify genes and chromosomal segments. A recent review identified 77 genes whose amplification is related to tumor growth [14], which suggests that BFB can also play a role in the evolution of cancer cells. Other studies implicate BFB's in the development and generation of different types of cancers [15]–[19].

Some cancer cells show evidence of rearrangements that are more complex than the basic operations. Recently discovered, *Complex Genomic Rearrangements (CGR)*, are rearrangements that involve a number of breakpoints [20]–[23]. Bignell et al (2007) observed the presence of "bizarre clusters of small genomic fragments" in samples taken from breast and lung cancer [6].

Berger et al. (2011) Characterized some of the genomic alteration in prostate cancer and discovered a type of CGR they called *Closed Chains of Brakeage and Rejoining (CCBR)* [23]. They suggested that CCBR occur when distant chromosomal locations are spatially close to one another in the nucleus, and showed that CCBR are responsible for the fusion gene TMPRSS2-ERG which is associated with carcinogenesis process of prostate cancer [24]–[28].

1.1.4 Tumor evolution model

The model of cancer as an evolutionary process was already suggested more than 40 years ago [29]. It is hypothesized that a series of rearrangements, caused by isolated events of breakage and joining of the DNA strands occur constantly, but those that have a selective advantage become more prominent. For example, rearrangements that help the cell avoid apoptosis or an immune response can become dominant and progress into what we identify as a tumor. These selective advantages are often associated with specific genes, named oncogenes, or novel fusion genes. Over-representation of such genes, caused by rearrangements that amplify their protein levels, have been associated with cancer [22], [30], [31].

This model of simple aberrations that gradually accumulate over time was however challenged in 2011 when Stephens et al. proposed a new paradigm called *chromothripsis* [32]. This paradigm suggests that in some cases a single catastrophic event occurs in which a genomic section on the chromosome is shattered into a large number of small fragments and then re-assembled, effectively creating a cluster of breakpoints that seemingly do not fit the model of simpler rearrangement operations, and has a much bigger underlying complexity. Their observations suggest that a gradual accumulations is sometimes a very unlikely interpretation of the data and that a single event is a more probable explanation. They found that chromothripsis occurs in 2%-3% of all tumors and is present in 25% of bone cancers [32]. Later studies argued higher prevalence of such events [33], [34].

Baca et al. proposed in 2014 a less catastrophic model of punctuated evolution [35]. They examined the genomes of 57 prostate tumors and systematically profiled somatic alterations in them. They characterize chains of breakpoints that are dependent events and use a statistical model to show that a sequential mechanism of events is unlikely to give rise to the observed chains [35]. They called this phenomenon *chromoplexy*.

1.1.5 Using rearrangements to infer genomic distance

Modeling the somatic evolution of cancer holds great value in potentially clustering different tumor types, and even patient specific samples, into novel groups, using methods developed for species evolution. Rearrangement of genomic sequence is known to be one of the major driving mechanisms of evolution, and is used to determine phylogenetic distance between

species. In 1995 Hannenhalli and Pevzner proposed a method to calculate the genomic distance between two species based on the minimal number of reversals required to transform the genome of one species to another [36], [37].

Another distance metric to infer evolutionary distance was used by Braga et al. in 2011. They propose a method that calculates the distance between two genomes based on *Double-Cut and Join (DCJ)* operations and *indels* (Insertions and deletions). The benefit of this metric is that it can compute the genomic distance between genomes that do not necessarily share the same content. They utilize their methods to show evidence for deletion clusters in six species of Rickettsia [38].

Feijão et al. defined yet another metric, called *Single-cut and join (SCJ)* that allows linear and polynomial time solutions to some genomic distance problems that are NP-hard under other distance metrics. They show that using the simplified SCJ distance, they can recover between 60 and 90 percent of the topology of a phylogenic tree with 200 different genomes and with as many as 3000 genes [39], [40].

Ozery-Flato and Shamir introduced the *elementary distance* between two karyotypes, defined as the least number of elementary operations – breakage, fusion, duplication and deletion – transforming one into the other. They presented the *karyotype sorting problem* as the problem of seeking the shortest elementary distance between two karyotypes and suggested a polynomial time 3-approximation algorithm for it. Applying the algorithm on more than 58,000 karyotypes taken from the Mitelman database [41], 99.9% of the resulting solutions matched the lower (optimal) bound [42].

1.1.6 Tumor heterogeneity

The current understanding of cancer as an evolutionary model adds another dimension of complexity as tumors are essentially heterogeneous in nature and cells of a tumor can have different karyotypes, resulting from different evolutionary paths [43]–[47]. However, most DNA sequencing technologies today still require DNA from numerous cells, thus resulting in measurement from a mixture of genomes. This presents a problem as the cancer cells in a tumor may not all be identical but rather a collection of cells with different karyotypes. In addition, tumor cells are part of the so-called tumor microenvironment, a heterogeneous tissue containing not only cancer cells, but also stromal and immune cells, with a normal karyotype [48].

In 2012 Mahmoody et al. formulated the problem of reconstructing k genomes derived from a single reference genome given partial information about their sequence, obtained by sequencing their mixtures. They termed the problem the *k-MCP problem* and showed that it is NP-complete for $k \geq 3$ in the general case [49]. Subsequent work by Oesper et al. [50] introduced *THetA (tumor Hetrogenity Analysis)* – an algorithm to infer the most likely collection of different genomes and their proportion in a tumor population. Later Hajirasouliha et al. [51] formulated the problem as the NP-complete *BTP (Binary Tree Partition)* problem and introduced an efficient approximation algorithm to solve it.

1.2 Next Generation Sequencing (NGS)

Until 2008, the ability to identify genomic rearrangements used to be quite limited, and restricted by low resolution of costly and time consuming sequencing. With the advent of Next Generation Sequencing (NGS) techniques and the rapid decrease in both cost and time,

it is now possible to interrogate entire exomes, genomes or transcriptomes of cancer samples, including clinical samples [52]–[54].

Optimally we would like to be able to construct the whole set of rearranged chromosomes, and get a perfect sequence of the underlying karyotype. But even though DNA sequencing technologies have improved dramatically over the past decade, and NGS now enables the sequencing of large cohorts of cancer genomes, the present established DNA sequencing technologies are still local in nature and are limited in the length of DNA sequences they produce. A typical experiment can produce millions of sequences ("reads") of length of 200 bases. De novo assembly of genomes from such data remains a difficult task [55].

1.2.1 Paired end reads data

One of the methods used for identifying rearrangements and inferring breakpoints in the genome is named *Paired End Sequencing*, and the reads it produces are called *paired end reads*. This is a sequencing based method that has been used in the past decade to detect structural variants in different genomes [56]–[59].

Paired end reads are generated by the fragmentation of genomic DNA into short (~300 base pairs) segments, followed by sequencing of both ends of the segments. The two ends of each read are then aligned back to a reference "normal" genome (In the case of cancer – a healthy cell line from the same patient). The approximate length of appropriate segments is known in advance and so we expect the two ends to be aligned to the reference genome at roughly that distance. The relative orientation of the paired reads is also known. An alignment that meets those expectations of distance and orientation is called a *concordant read*. A set of adjacent concordant reads, mapped to the same region in the normal genome, reflects a segment on the sample genome that was not altered by rearrangements. A *discordant read*, however, is a read whose ends are mapped to different locations on the reference genome or with an unexpected relative orientation (Figure 1.5).

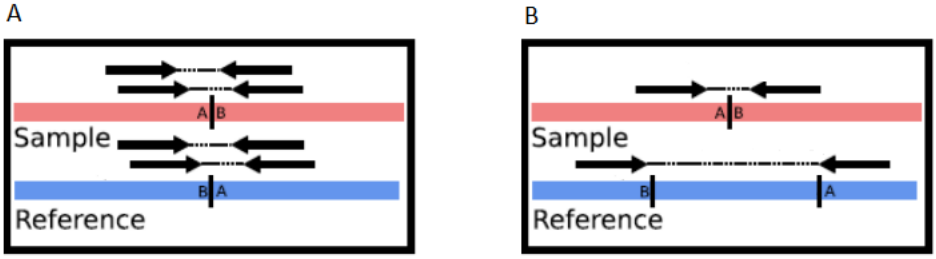

Figure 1.5: Concordant and discordant paired end reads. A. Two concordant reads where both ends of the reads are aligned with the expected distance between them and with the same relative orientation. B. A discordnat read where both ends are aligned to locations that are farther apart than expected. Source: [60].

Discordant reads suggest a breakpoint, a fusion between two nonconsecutive positions in the reference genome due to rearrangement. A read taken from that spot will have its two ends aligned to locations on the reference genome where those positions originally lie. So a list of these discordant reads can be translated into a list of novel adjacencies on the sample genome that can be seen as *bridges* between two breakpoint locations that are normally far apart but due to rearrangements are now adjacent.

The type of discordance also suggests the type of rearrangement that occurred (Figure 1.6). A deletion will cause reads to be aligned in the correct orientation but with a greater distance

than expected between them. An inversion will cause reads from around the breakpoints bordering the inversion event to have reads aligned facing the same direction instead of each other (only reads covering the end of the inverted segment will have ends reversed in relation to the reference genome). Their distance from one another might also be bigger than expected when the inverted segment is large. Similarly, tandem duplications will cause the reads to be aligned "backwards", meaning facing away from each other, and possibly at larger distance than expected from one another. Translocations will have the ends aligned to different chromosomes (Figure 1.6).

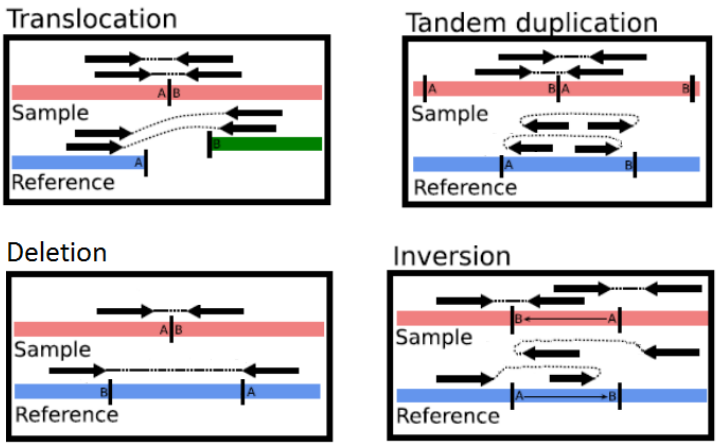

Figure 1.6: The paired ends read alignment signature of basic rearrangements. A translocation causes the ends to align to different chromosomes. An inversion and tandem duplication result in the ends aligned with a relative orientation opposite of that expected, and a duplication causes the ends to align to locations that are far apart on the same chromosome. Source: [60].

A first step in analysis of paired end reads is their mapping to the reference genome. Each read contains the two locations on the genome where its two ends aligned to and their relative orientation. Analysis of these data is done with computational approaches that try and infer the structural variations from the discordant reads and produce a set of rearrangement events [61]–[66]. While the most useful data are the set of discordant reads, other methods such as PREGO [67] take into account the concordant reads as well. BreaKmer [60] is a method that uses the misaligned reads together with the aligned concordant and discordant reads to predict rearrangements using kmers. Other algorithmic approaches have been developed to try and infer rearrangements that are less simple and have different kinds of "signatures". [21], [35], [60].

Some Methods seek to achieve higher accuracy by aggregating results from several different tools. MetaSV [68], introduced in 2015 by Mohiyuddin, Mu et al., offers an improvement of accuracy and precision in detecting different kinds of structural variants. By effectively merging the results from multiple tools, they were able to reach F1-scores (harmonic mean of sensitivity and precision) of 96.2% for deletions and 84.7% for insertions. Fang et al. developed SomaticSeq [69], a pipeline for detecting single nucleotide variants (SNVs) and small insertions and deletions (indels), using machine learning algorithms to incorporate the results from five somatic mutation callers. They applied their method on data from the ICGC-TCGA DREAM Somatic Mutation Calling Challenge [70] and report an F1 score of 90.5%.

Identifying these variations can further the understanding of one of the main driver mechanisms of cancer as well as offer better diagnostic tools and treatments. However,

inferring the underlying structure of cancer karyotypes from a list of known rearrangement events proves to be a challenge and is a subject of ongoing research in cancer genomics.

1.3 Copy number variations

Genomic rearrangements create not only novel adjacencies but may also change the copy number of different segments of the DNA sequence, i.e. the number of times a segment is present in the karyotype. A healthy normal (human) cell line will have 22 diploid chromosomes (plus a pair of sex chromosomes XX or XY) and so the copy number of the entire karyotype is 2. A rearranged genome might have variations in the copy number of different genomic segments. A gain or a loss of an entire chromosome will decrease or increase the copy number of that chromosome, respectively. A fraction of a chromosome can also be deleted or duplicated. The resulting segment or chromosome is said to have undergone a *copy number variation* (CNV).

Large CNVs can be detected using more traditional methods like Fluorescence in-situ Hybridization (FISH) [71]. Higher resolution detection of CNVs can be achieved by Array Comparative Genomic Hybridization (aCGH) [72]. aCGH is a development of the older Comparative Genomic Hybridization (CGH) method, which was the first efficient approach to scanning the entire genome for variations in DNA copy number [73]–[77]. In a typical CGH measurement, total genomic DNA is isolated from test and reference cell populations, differentially labeled and hybridized to metaphase chromosomes. In the case of aCGH, the hybridization is to a DNA microarray containing genomic probes. The relative hybridization intensity of the test and reference signals at a given location is then (ideally) proportional to the relative copy number of those sequences in the test and reference genomes. If the reference genome is normal, then increases and decreases in the intensity ratio directly indicate DNA copy-number variation in the genome of the test cells. The quantification of CGH intensity varying is achieved through the use of competitive fluorescence in situ hybridization. The test and reference samples are assigned difference colors (usually red and green) and the log ratio between the signals' strength is calculated (Figure 1.7).

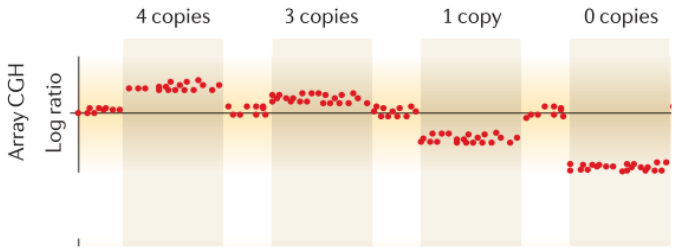

Figure 1.7: An illustration of the copy number data extracted through the use of aCGH. The x-axis lists probes in their order along the normal genome and the y-axis lists the test/reference hybridization log ratio. The log ratio in each genomic segment indicates the copy number. Source: [54].

The introduction of array CGH offers an improvement to the low resolution of conventional CGH. In array CGH, Test and control DNA are hybridized to cloned DNA fragments of size 100-200 kb that have been spotted on a glass slide (the array), and whose exact chromosomal location is known [78]. This allows for detection of aberrations in more detail, and makes it possible to map changes in copy number directly onto the genomic sequence. Other array based methods have been developed to further improve the resolution provided by aCGh, such as High-Resolution CGH (HR-CGH) [79].

With the advent of next-generation sequencing (NGS) techniques, several methods have been developed to infer CNV's using DNA sequencing [67], [80], [81] . Most of the algorithms were developed to specifically detect structural variations. The basic approach is to count the number of reads that align to each part of the DNA sequence (read depth). The assumption is that reads are distributed randomly (typically via Poisson distribution) and a divergence from the expected number of reads in a specific region suggests a duplication or deletion.

NGS based methods promise a revolution in CNV analysis and seem to eventually replace the lower resolution array methods. However they provide many computational challenges and as of today the different methods available vary widely in the results they produce on the same DNA sequence [54].

1.4 Graph models for rearrangements

Graph theory has been highly instrumental in the area of genomic rearrangements. For example, de Bruijn graphs were used for genome assembly problems [82], and breakpoint graphs were used in reconstructing rearranged genomes across species [36], [83]. More recently, similar methods were adapted for cancer genomes [42], [84].

1.4.1 The breakpoint graph

The breakpoint graph, introduced by Pevzner and Bafna in 1993 [85], remains today one of the key data structures in the study of genomic rearrangements. A breakpoint graph is used to represent the relation between two permutations of the same set of elements. By assuming w.l.o.g that one is the identity permutation I, we can view the graph as describing the other permutation.

Formally, we define the breakpoint graph $G(\pi)$ for the permutation π on the numbers $1, \dots, n$ as follows: Let $i \sim j$ if $|i - j| = 1$. Extend $\pi = \pi_1, \dots, \pi_n$ by adding $\pi_0 = 0$ and $\pi_{n+1} = n + 1$. A pair of consecutive elements $\pi_i, \pi_{i+1}, 0 \leq i \leq n$ is called an *adjacency* if $\pi_i \sim \pi_{i+1}$, and a *breakpoint* if $\pi_i \nsim \pi_{i+1}$. Define $G(\pi) = G(V, E)$ to be an edge-colored graph with $n + 2$ vertices as follows: $V = \{0,1,2, \dots, n + 1\}$. There are two types of edges: black and red: $E = E_{black} \cup E_{red}$. i and j are connected by a black edge if (i, j) is a breakpoint, and by a red edge if $i \sim j$ and i, j are not consecutive in π. An example of a breakpoint graph is given in Figure 1.8.

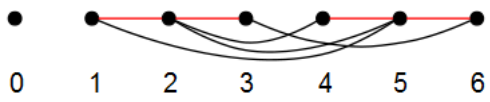

Figure 1.8: A breakpoint graph corresponding to the permutation 1,5,2,4,3.

We can consider a rearranged chromosome as a permutation of segments of the DNA sequence. By representing each such segment as a vertex we can construct a basic breakpoint graph with edges connecting segments that are adjacent in the rearranged or the reference genome.

In reality, each segment has an *orientation*. An orientation of a segment can be represented by a signed number, and a signed permutation π of order n can be transformed into an unsigned permutation π' of order $2n$. For each positive element $+i$ in π we replace i with $2i -$

$1, 2i$, and for every negative element $-i$ with $2i, 2i-1$. The signed breakpoint graph is then constructed in the same way. See Figure 1.9 for an example.

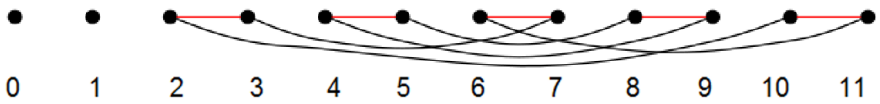

Figure 1.9: A breakpoint graph representing the signed permutation 1,-5,-2,4,3. The permutation is first transformed into the unsigned permutation 1,2,10,9,4,3,7,8,5,6.

The breakpoint graph was instrumental in applying computational methods for solving biological problems. It offers a way to calculate the genomic distances between different species using the number of reversals needed to turn one genome into another. In their paper from 1995, Hannenhalli and Pevzner gave a polynomial algorithm to compute the distance in reversals and translocations between two signed permutations, and showed that a mouse genome can be transformed into a human genome using 131 rearrangements of reversals, translocations, fusions and fissions [36]. The main limitation with this model is that it takes into account only reversals in the genome. The model finds the shortest reversal distance between two genomes, which is a simplest evolutionary path that turned the genome of one species into another. Cancer genomes however go through many types of rearrangements, such as deletions and duplications, or even catastrophic shattering and rejoining, for which a minimum reversal distance is not very telling [22], [32].

1.4.2 Allelic and somatic graphs

Greenman et al. expanded on the breakpoint graph in 2012 and introduced a construction that is essentially equivalent called the *allelic graph* and its counterpart the *somatic graph* [86]. Similar to the breakpoint graph, rearranged segments are represented in the allelic graph as nodes and two types of edges signify the two types of connections – breakpoint (i.e. a novel adjacency in the rearranged genome) and *germline* (segments that are adjacent in the reference sequence).

To deal with segment orientations, the nodes in the allelic graph have sides so an edge connecting to node v can connect to its right or left side depending on whether the connection is to the head or tail of the segment. In addition, the allelic graph holds two copies of each node representing the two alleles, and incorporates copy number for both nodes and edges.

Formally, for a karyotype divided through rearrangements into n segments the allelic graph $G(V, E)$ has $2n$ nodes for each segments' minor and major alleles: $V = \{v_{1_m}, v_{1_M}, \dots, v_{n_m}, v_{n_M}\}$. A breakpoint edge $e = (i, j)$ connects nodes i, j if they are adjacent in the rearranged karyotype. A somatic edge $e = (i, i+1)$ connects nodes that are adjacent in the non-rearranged karyotype. All nodes and edges have a copy number $f: (V \cup E) \to \mathbb{N}$. In addition each node v has two distinct sides, v^+ and v^-, corresponding to the two ends of the segment in the reference genome. Note that a somatic edge is always of the form $(i^+, (i+1)^-)$, connecting segments that are adjacent in the reference. An example of an allelic graph is given in Figure 1.10B.

The allelic graph is complemented by the *somatic graph* $G(V, E)$. For a karyotype divided into n segments G has $n-1$ nodes representing the breakpoints between the segments. An edge $e(i^{-\backslash+}, j^{-\backslash+})$ connects the sides of the breakpoints i, j if the segments laying on the

corresponding sides of each breakpoint were connected through a rearrangement (Figure 1.10 C).

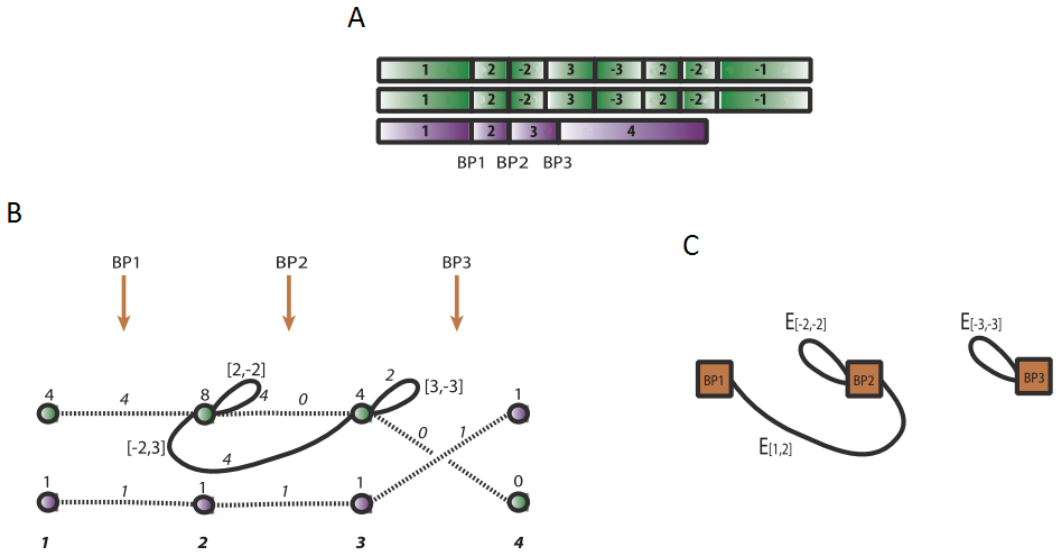

Figure 1.10: Allelic and somatic graphs. (A) A chromosome divided by three breakpoints into four segments. One copy of the chromosome (Green) underwent structural variations while the other (purple) did not. (B) The allelic graph representing the karyotype. Dashed lines are somatic edges representing adjaecencies in the unaltered chromosome, solid lines are breakpoint edges representing novel adjacencies in the rearranged chromosome (displayed in brackets). The numbers next to vertices and edges are copy numbers (C) The somatic graph representing the karyotype. The 3 breakpoints are double sided vertices in the graph with the solid lines representing edges connecting the breakpoints. Source: [86].

Using the allelic and somatic graphs, Greenman et al. suggested a model relying on finding certain connected components to infer a set of possible rearrangements and the order in which they occurred. They validated their algorithm using FISH, showing that specific chromosomal contigs are located where expected according to the inferred rearrangements. This however does not validate that the actual type and order of rearrangements inferred by the algorithm are necessarily correct.

There are some limitations to this model. Firstly, it assumes that breakpoints are unique throughout the process of the tumor evolution, and do not repeat in several unrelated rearrangement events. This assumption is problematic as many tumors seem to have breakpoints that are clustered together in very close proximity, which may suggest they are related events [20], [21]. Secondly the model assumes a pre-defined set of nine possible rearrangement events that are caused by three or less breakpoints. Some tumors have been shown to likely be caused by more complex events, sometimes catastrophic in nature, and not a series of simple rearrangements [32], [35], [87]. Lastly, inaccurate data can drastically change the structure of the allelic graph and therefore the result of the algorithm.

1.4.3 Interval adjacency graph

Another construction that expands on the breakpoint graph is the *interval adjacency graph*, proposed by Oesper et al. in 2012 [67]. The interval adjacency graph is constructed directly from copy number and novel adjacencies data, and not a reconstructed karyotype. Similar to the allelic graph proposed by Greenman [86], the discordant reads are used to infer breakpoint locations on the DNA sequence and partition it to intervals accordingly.

Each interval I_i in the partition of the genome is represented by two nodes s_i, t_i corresponding to the interval ends and connected by an *interval edge* (s_i, t_i). The tail end of each interval I_i is connected to the head of interval I_{i+1} by a *reference edge* (t_i, s_{i+1}), representing adjacency on the original reference genome. The third type of edges are *variant edges*. Like the allelic graphs' breakpoint edges or the breakpoint graphs' black edges, they connect intervals I_i, I_j if they are adjacent only in the rearranged chromosome and not in the reference, or they are adjacent in a different orientation than in the reference (Figure 1.11).

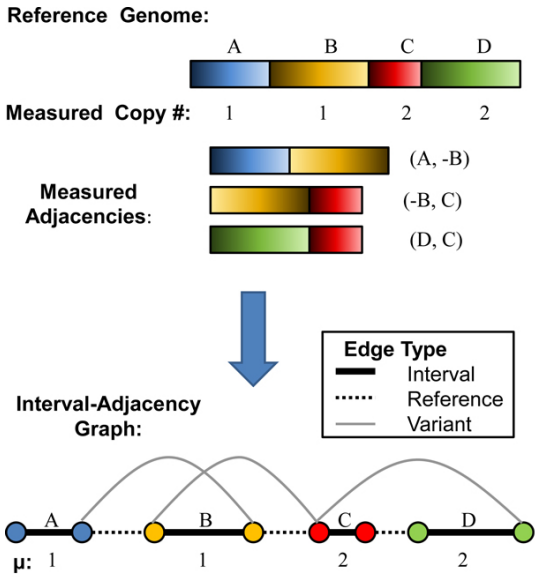

Figure 1.11: An interval adjacency graph representing the observed copy number and adjacencies measured [67].

Using the interval adjacency graph it is possible to infer rearranged sequences that agree with the data. Oesper et al showed that an Eulerian path on the graph alternating between interval edges and reference / variant edges corresponds to a rearranged sequence of the chromosome. Based on this construction they developed an algorithm called *PREGO* to determine the most likely sequence of a rearranged karyotype. Using simulations they showed their algorithm can deduce the correct multiplicity of more than 80% of the variant edges, even with large noise ratios and when the sample is heterogeneous. Furthermore, they applied PREGO to five ovarian cancer genomes and were able to identify numerous rearrangements and structural variants, some of which were consistent with known mechanisms.

The PREGO algorithm combines copy number and adjacency information from paired end sequencing to infer multiplicity of different segments in the cancer genome. However except in simple cases, the underlying sequence of the genome cannot be uniquely resolved, as many reconstructions will be consistent with the data.

1.5 Integer Linear Programming (ILP)

An integer program is a mathematical optimization problem in which the variables are restricted to be integers. An integer linear program is one in which the target function and the constraints are linear.

A standard ILP form is:

Minimize:

$$\sum_{1\leq j\leq n} c_j x_j$$

Subject to:

$$\sum_{1\leq j\leq n} a_{ij} x_j \lesseqgtr b_i \qquad \forall\, 1 \leq i \leq m$$

$$x_j \geq 0 \text{ and integer} \qquad \forall 1 \leq j \leq n$$

Where $x = (x_1, x_2, \dots x_n)$ are the variables, $c = (c_1, c_2, \dots, c_n)$ are the coefficients of the objective function, a_{ij} are elements in a matrix of size $m \times n$ and $b = (b_1, b_2, \dots, b_m)$ a vector that together consist of the m constraints. All numbers are assumed to be rational, and $\lesseqgtr$ stands for either $\leq$ or $\geq$.

1.5.1 Nonlinear models

Some models that include nonlinear features, such as absolute values, can be transformed into conventional linear programming models. Several tricks can be applied in order to achieve this, and when possible it is often advisable to do so instead of solving a nonlinear model [88], [89].

1.5.1.1 Absolute values

Let us consider the following model where the target function includes the absolute value of the variables:

Minimize:

$$\sum_{1\leq j\leq n} c_j |x_j| \qquad c_j \geq 0$$

Subject to:

$$\sum_{1\leq j\leq n} a_{ij} x_j \lesseqgtr b_i \qquad \forall\, 1 \leq i \leq m$$

x_j is free

The absolute values can be avoided by replacing each x_j with two new variables x_j^+, x_j^-, and adding constraints as follows:

$$x_j = x_j^+ - x_j^-$$
$$|x_j| = x_j^+ + x_j^-$$
$$x_j^+, x_j^- \geq 0$$

And then the model can be reformulated:

Minimize:

$$\sum_{1\leq j\leq n} c_j(x_j^+ + x_j^-) \qquad c_j \geq 0$$

Subject to:

$$\sum_{1\leq j\leq n} a_{ij}(x_j^+ - x_j^-) \leqslant b_i \qquad \forall\, 1 \leq i \leq m$$
$$x_j^+, x_j^- \geq 0 \qquad \forall\, 1 \leq j \leq n$$

The solutions for both programs is the same if for every j either x_j^+ or x_j^- is zero. This is guaranteed to happen for an optimal solution since if both are greater than zero then there exists a $\delta > 0$ such that $x_j^+ - \delta, x_j^- - \delta$ is a better solution.

1.5.1.2 Min max functions

Another common object found in many models is the minmax object. Consider the following model:

Minimize:

$$\max_k \sum_{1\leq j\leq n} c_{kj}x_j$$

Subject to:

$$\sum_{1\leq j\leq n} a_{ij}x_j \leqslant b_i \qquad \forall\, 1 \leq i \leq m$$
$$x_j \geq 0 \qquad \forall 1 \leq j \leq n$$

The requirement to minimize the maximum of k options can be transformed into a conventional form by including another variable $z = \max_k \sum_{1\leq j\leq n} c_{kj}x_j$ and then solving the following model:

Minimize:

$$z$$

Subject to:

$$\sum_{1\leq j\leq n} a_{ij}x_j \leqslant b_i \qquad \forall\, 1 \leq i \leq m$$
$$\sum_{1\leq j\leq n} c_{kj}x_j \leq z \qquad \forall 1 \leq k \leq K$$
$$x_j \geq 0 \qquad \forall 1 \leq j \leq n$$

An optimal solution for this model will minimize z and make sure it is still bigger than the target function for every k, which will result in z being the minimized maximum.

1.5.2 Solving an ILP problem

Solving a linear program can be done in polynomial time, but ILP is NP-hard [90]. Several industrial solvers exist for optimization of linear programming and integer linear programming problems, such as CPLEX [91]. These solvers provide an interface to formulate LP and ILP

problems in different programming languages such as java and python, and implement different tricks to translate nonlinear models into standard linear ones. These solvers employ a variety of heuristics and exhaustive algorithms for ILP. While obtaining an optimal solution quickly is not guaranteed, in practice, many formulations used on current biological networks are solved in reasonable time [92].

2 Methods

In this section we describe the methods we used to transform a copy number and paired-end read data into a weighted graph, and the ILP formulation that we developed to find the solution most concordant with it.

Next generation sequencing methods produce paired-end reads. Recall (section 1.1.51.2.1) that discordant reads suggest structural variations on the genome. Genomic copy numbers can be obtained using NGS methods or different CGH array technologies (section 1.3). We propose a novel method that incorporates these two kind of data, which can be derived using different technologies, in order to reconstruct the karyotype that has the most agreement with both of them.

The outline of our approach is as follows (compare Figure 2.1). We use the novel adjacencies and copy number data together to construct a *bridge graph*, similar to the *adjacency graph* proposed by Oesper et al. [67], and to the more classic b*reakpoint graph* [85]. The bridge graph is then fed to an ILP solver that outputs a solution representing a valid karyotype of the rearranged genome that is most concordant with the observed data. A graphical engine is used to produce a graphical illustration of the solution.

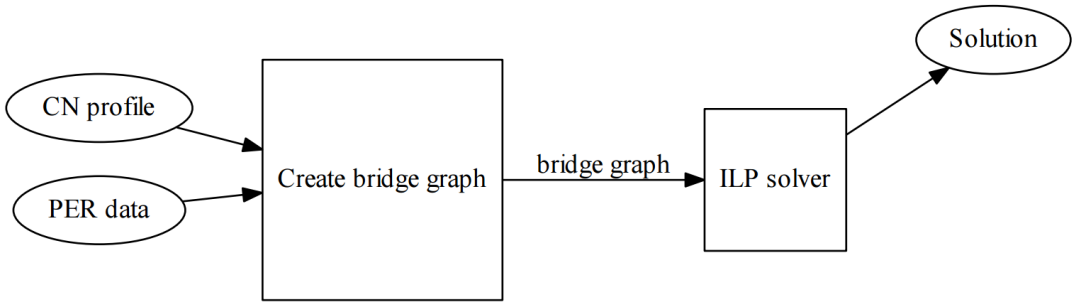

Figure 2.1 Overview of the approach. We incorporate copy number (CN) profile with a list of novel adjacencies derived from Paired-End Reads (PER) data to create a bridge graph. We then use an ILP solver to suggest a karyotype most concordant with the data.

2.1 Building the adjacency and bridge graph

In our problem setup there is a *normal (or reference) genome*, whose contents is known, and an unknown *target genome* that should be reconstructed. A *breakpoint* is a point along the reference genome involved in a structural change event in the target genome.

Let C be the set of chromosomes in the reference karyotype. The breakpoints partition each chromosome $c \in C$ into a set of k^c intervals $I_c = \{I_1^c, I_2^c \dots I_{k^c}^c\}$, such that each $I_{k^c}^c$ is an interval between consecutive breakpoints, or between a breakpoint and a chromosome end. The intervals are numbered in increasing order along c, so that c is equal to the concatenation of the intervals $I_1^c, I_2^c \dots I_{k^c}^c$. We call the start and end points of interval I the *tail* and *head* of I and denote them by t_I and h_I respectively. Hence, $I = [t_I, h_I]$, and $-I = [h_I, t_I]$ is the interval I reversed. An *extremity* is a tail or a head of an interval. The set of all intervals $\mathcal{I} =$

$\cup_{c\in C} I_j^c$ can be considered as the set of the basic building blocks of the reference and target genomes. The length of interval I_j (in bases) is denoted by l_j, and $L = \sum l_i$ is the total length of all intervals.

The target genome can be represented by a set of chromosomes, where each chromosome is a sequence of intervals and reversed intervals (Figure 2.2). A *bridge* is a pair of extremities that are not adjacent on the reference genome but are adjacent in the target genome. Bridges can be detected based on the paired-end read data of the target genome (Figure 2.2). Each bridge b_i has a certain *support level*, which is the number of paired-end reads that support it, denoted μ_i. The total support score for all bridges is denoted $\mu = \sum_{b_i} \mu_i$.

Figure 2.2: Reference and target genomes. A: reference (germline) DNA chromosome segmented into intervals separated by breakpoints. B: The rearranged chromosome represented by the series of intervals 1,4,-4,-3,2,-1. Genome B contains the bridges $\{h_1, t_4\}, \{h_4, h_4\}, \{t_3, t_2\}$ and $\{h_2, h_1\}$. Note that $\{t_4, h_3\}$ is not a bridge.

Each interval $I_i \in I$ has a *copy number* $N_i \geq 0$ indicating the number of times it appears in the target genome. The set of copy numbers of all intervals is called the *copy number profile* of the target. That profile can be derived from deep sequencing data or from array CGH data. In perfect data, N_i is exactly the number of copies of the interval in the target genome. In read data, the copy numbers are real valued estimates based on mean coverage of each interval.

We are now ready to define the graph structure. We first define the interval adjacency graph, introduced by Oesper et al. [67]. The input is (1) the reference genome represented as a sequence of intervals for each chromosome. These intervals form the set $\mathcal{I} = \{I_1, \ldots, I_n\}$; interval I_j has length l_j. (2) The copy number profile of the intervals: Interval I_j has CN N_j. (3) The set of bridges $\{a_i, b_i\}_{i=1}^m$ and the support μ_i for each bridge. Each a_i and b_i is an extremity of an interval in $\mathcal{I}$. We define a weighted graph $G(V, E, w)$ whose vertices are the interval extremities. For each interval $I_i = [t_i, h_i]$, the graph contains an *interval edge* $e_I(t_i, h_i) \in E_I$ connecting its two extremities, of weight N_i. For each two intervals I_i, I_{i+1} that are adjacent on the reference genome, a *reference edge* $e_R(h_i, t_{i+1}) \in E_R$ connects the head of the first interval to the tail of the one following it. Reference edges are unweighted. Each bridge is represented by a *bridge (or variant) edge* $e_V(a_i, b_j) \in E_V$ connecting the two extremities a_i and b_j, with weight μ_i. In total, the edge set of the graph is $E = E_I \cup E_R \cup E_V$. We denote by $S \subseteq V$ the set of vertices that represent *telomere nodes*, i.e. the nodes representing the start and end points of each reference chromosome, hence $S = \cup_{c\in C} \{t_1^c, h_{k^c}^c\}$ includes the heads of all starting intervals and the tails of all ending intervals in each chromosome's partition. See Figure 2.3 for an example.

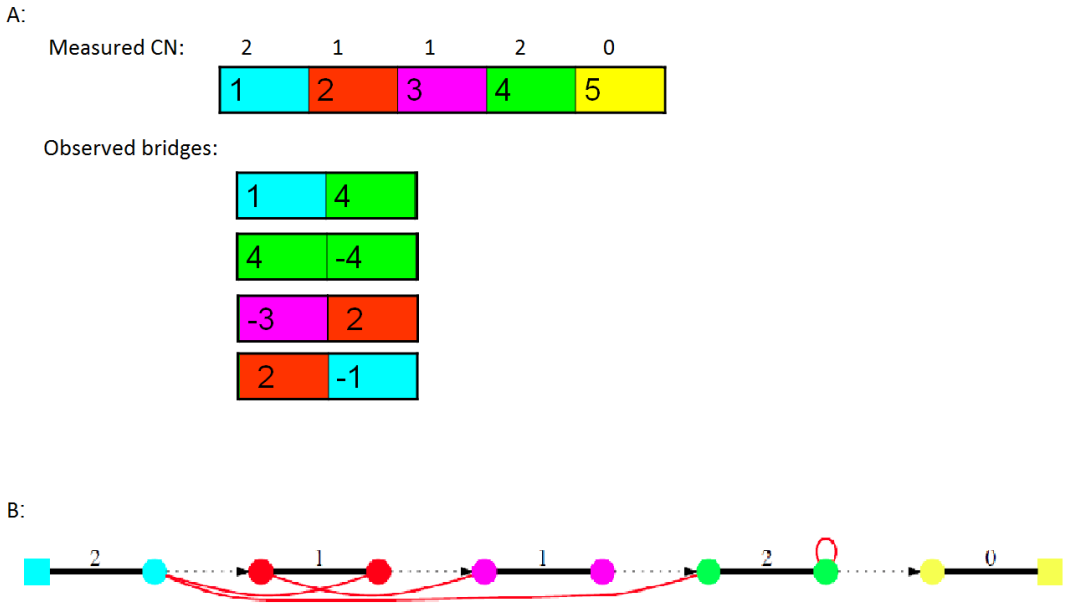

Figure 2.3: Interval adjacency representation. A: The observed bridges and measured CN of intervals. B: The interval adjacency graph representing the observed data. Bold black edges are interval edges with the relative CN, dotted edges are reference edges and red edges are variant edges. Telomere nodes are noted by a square.

2.1.1 Parallel edges

Note that the set of interval Edges E_I and reference edges E_R are disjoint, since by definition the former connect the extremities of the same segment while the latter connect extremities of adjacent segments. Variant edges however can connect two nodes that already have another edge connecting them.

When a tandem duplication occurs on segment I_i, there will be a bridge connecting the head of the segment h_i, to its tail, t_i, and so the variant edge $e_V(t_i, s_i)$ will be parallel to the interval edge $e_I(s_i, t_i)$ (Figure 2.4).

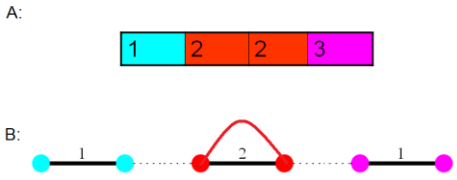

Figure 2.4: Parallel edges. A: A chromosome that underent a tandem duplication of interval 2. B: The interval adjecency graph with paralel interval and variant edges.

The sets of variant edges E_V and reference edges E_R are disjoint as variant edges represent bridges that can only exist between interval extremities that are not adjacent on the reference genome, while reference edges connect extremities of adjacent intervals. However, parallel variant and reference edges can be occur due to our manipulation on data (See section 2.1.4).

2.1.2 Same node edges

Another peculiarity in the adjacency graph that we allow are variant edges from a node to itself (Figure 2.5). These type of bridges occur usually as a result of a breakage-fusion bridge

rearrangement, and were used before in similar constructions such as the Somatic Graph proposed by Greenman et al. [86].

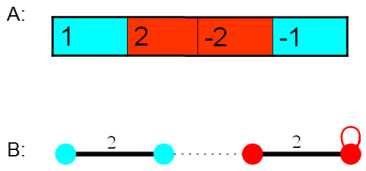

Figure 2.5: Self loops. A: A chromsome that underwent a BFB event. B: The interval adjacency graph. The variant edge representing the bridge $\{h_2, h_2\}$ connects the same node t_2 to itself.

2.1.3 From adjacency graph to bridge graph

We now provide a similar extension to the adjacency graph introduced by Oesper et al. [67]. The bridge graph has additional weights of bridge edges and not only interval edges. Recall that the weight $w(e)$ of an interval edge $e(u, v)$ is the copy number of the segment $[u, v]$. The weight $w(e)$ of the bridge $e(u, v)$ is its support score.

Additionally we transform each undirected edge $e(u, v)$ in the interval adjacency graph into two directed edges $e_{\rightarrow}: u \rightarrow v, e_{\leftarrow}: v \rightarrow u$. The original undirected edge is referred to as a *connection* to distinguish it from the directed edges and $E = E_{\rightarrow} \cup E_{\leftarrow}$ the set of edges in the graph.

We call the modified graph a *Bridge Graph* (Figure 2.6). In summary, the bridge graph is a generalization of the adjacency graph [67], with additional weights assigned to bridge edges, and antiparallel directed edges instead of undirected edges.

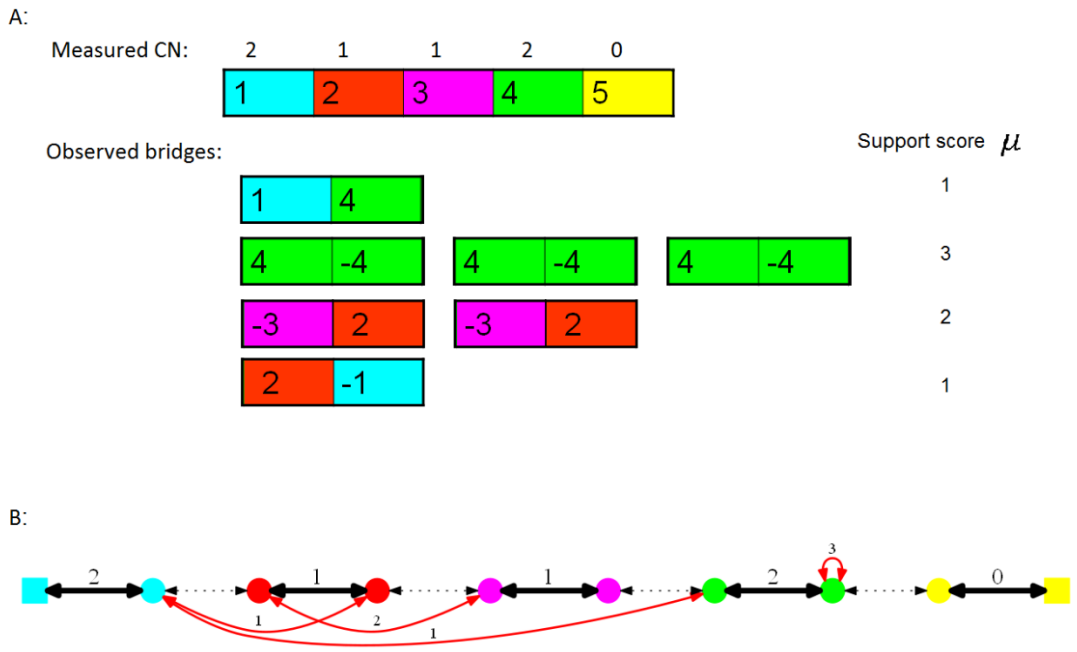

Figure 2.6: Bridge graph. A: The measured copy number and bridge data with the observed support score for each bridge. B: The corresponding bridge graph with weights for variant and reference edges. All connections are composed of two antiparallel directed edges.

2.1.4 Breakpoint filter

In analyzing real data we discovered that some breakpoints are clustered together in a very small region at close proximity to one another. Since the copy number data has a lower resolution than the paired-end data, this results an unnecessary over-fragmentation of the chromosome into a large number of small fragments that have almost the same measured copy number. To address this, we apply a filter on the list of breakpoints and group together breakpoints that are located very close to one another. The cut-off was set to be 5000 bases – all breakpoint coordinates that are within 5000 bases from each other are treated as the same breakpoint coordinate. This filtering does not throw out the actual bridge data, but ignores rearrangements that are very small and local in nature such as loss of small DNA shards that do not alter the result of the algorithm.

Ignoring the deletion of a small fragment of DNA between intervals I_i, I_{i+1} and grouping the relevant breakpoints together creates two nodes h_i, t_{i+1}, with both variant and reference edges connecting them. As a result the bridge becomes redundant and any resulting path outputted by the algorithm ignores it (Figure 2.7).

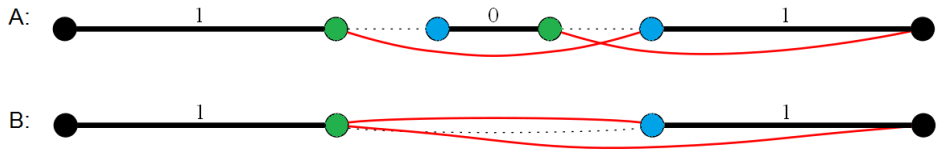

Figure 2.7: The effect of filtering small intervals. A: The interval adjacency graph including the deletion of a small segment between close breakpoints. B: The graph after applying a breakpoint filter. The blue and green nodes were joined together respectively. After the filtering the original bridge is still included but is rendered redundant and thus ignored by the algorithm.

2.2 Reconstructing the rearranged karyotype

Given the bridge graph $G(V, E, w)$, we wish to find paths in G that correspond to rearranged chromosomes. Suppose first that the input data is complete and errorless. Recall that $S \subseteq V$ is the set of vertices that represent *telomere nodes*, i.e. the nodes representing the start and end points of each chromosome. A *valid path* p is a path through G beginning and ending at $s_1, s_2 \in S$ that alternately traverses interval and non-interval edges (i.e. reference/bridge edges), and where the number of times each interval connection e_i is traversed (in either direction), denoted $f_p(e_i)$, is less than or equal to the copy number of interval i, N_i.

The requirement for an alternating path is because a traversal of an interval edge is equivalent to selection of a segment from the reference genome, while a traversal of a reference/bridge edge is equivalent to a transition between segments. Therefore, such an alternating path represents a sequence of segments from the reference genome. Note that $f_p(e_i) = f_p(e_{i\rightarrow}) + f_p(e_{i\leftarrow})$ for every connection e. A set of such paths $P = \{p_1, p_2 \dots p_n\}$ where for each interval connection e_i, $\sum_{p \in P} f_p(e_i) = N_i$ corresponds to a set of rearranged chromosomes, or a valid karyotype.

The restriction that the path alternates between interval and non-interval (reference\bridge) edges means that at each non-telomeric node $v \notin S$, every traversal on an interval edge going into v must be followed by a traversal on a reference\bridge edge going out of v, and vice-

versa. Telomeric nodes are excluded from this constraint as by definition they are the start or end of a path.

As detailed in section 2.1.3, each connection between nodes u, v is composed of two antiparallel directed edges. For each node $v \in V$ we denote $E_{I_{\leftarrow}}(v), E_{I\rightarrow}(v), E_{R\leftarrow}(v)$, $E_{R\rightarrow}(v), E_{B\leftarrow}(v), E_{B\rightarrow}(v)$ as the set of interval, reference and bridge edges that go in and out of v respectively. As above, we denote by $f_p(e)$ the number of times a connection e is traversed in path p and $f_P(e) = \sum_{p \in P} f_p(e)$ is the total number of times a connection e is traversed in P. Additionally, for a set of connections E, $f_P(E) = \sum_{e \in E} f_P(e)$ is the total number of times all connections in E are traversed in P. The constraints for a valid set of paths P, representing a rearranged karyotype, can be therefore formulated as:

$$
\begin{aligned}
&(1)\ f_P\left(E_{I_{\rightarrow}}(v)\right) = f_P\left(E_{R_{\leftarrow}}(v)\right) + f_P\left(E_{V_{\leftarrow}}(v)\right) \\
&\qquad \forall_{v \notin S} \\
&(2)\ f_P\left(E_{I_{\leftarrow}}(v)\right) = f_P\left(E_{R_{\rightarrow}}(v)\right) + f_P\left(E_{V_{\rightarrow}}(v)\right) \\
&\qquad \forall_{v \notin S} \\
&(3)\ f_P(e) \in \mathbb{N}^0 \\
&\qquad \forall_{e \in E}
\end{aligned}
$$

Equation 2.1: Constraints defining a valid set of paths P.

2.2.1 Distance of path from observed data

Recall that the interval and bridge edges of the bridge graph have weights, representing the measured copy number of the intervals and the support score for the bridges respectively. These values are in practice noisy. Given a bridge graph $G(V, E, w)$ and a valid set of paths P representing a rearranged karyotype, we can define the *discordance score* of P - $d_G(P)$, to measure how much P is in agreement with the data in G.

The discordance score is comprised of two parts and is described in Equation 2.2. The first part reflects how much P is in agreement with the CN profile. It is the sum over all interval edges $e_i \in E_I$, of the absolute value of the difference between $f_P(e_i)$ and the input weight $w(e_i)$, normalized by l_i. We normalize the weights of the intervals by their lengths since longer genomic intervals are expected to have more accurate CN values, and hence should be penalized more for disagreement.

The second part reflects how much P is in agreement with the bridge data. The more bridges a path is utilizing, the more it is concordant with the bridge data. To reflect this, a penalty is given for each bridge edge $e \in E_V$ that is not used in the path. The bigger the support score for a bridge is, the bigger the penalty, and so the penalties are normalized by $w(e)$. Recall that $\mu = \sum_{e \in E_V} w(e)$ is the sum of the support score for all bridges. Note that $w(e)$ and l_e values are on different scales, and the division by μ and L puts them on a similar scale.

$$
d_G(P) = \sum_{e \in E_I} \frac{l_e}{L} |f_P(e) - w(e)| + \alpha \sum_{\substack{e \in E_V \\ e \notin P}} \frac{w(e)}{\mu}
$$

Equation 2.2: General formulation of the discordancy score

The parameter α, determines how much weight the algorithm gives to paired-end reads data, i.e. how much it tries to utilize bridge edges in the solution. Using the algorithm on real tumor data, we set $\alpha = 0.5$. The effect of α on the performance of the algorithm is studied in 3.4.1.

The penalty for unused bridges is only given for bridge edges that are not in the path. We can formulate the score as a function of f_P using the min function as follows:

1. $\mathrm{d_G}(P) = \sum_{\mathrm{e}\in\mathrm{E_I}} \frac{\mathrm{l_e}}{\mathrm{L}} |\mathrm{f_P(e)} - \mathrm{w(e)}| + \alpha \sum_{\mathrm{e}\in\mathrm{E_V}} \frac{\mathrm{w(e)}}{\mu} \big(1 - \min\big(1, \mathrm{f_P(e)}\big)\big)$

Equation 2.3: The discordancy score as a function of f_P

2.2.2 The ILP formulation

Using the distance function above we can now formulate the problem at hand, of finding the rearranged karyotype that is most consistent with the observed data.

We define a rearranged karyotype to be most consistent if it corresponds to a valid set of paths with smallest discordance score. This can be formulated as an integer linear program where given a bridge graph $G(V, E, w)$ we want to minimize the discordance score of P with the constraint that P is a valid set of paths representing a rearranged karyotype.

Formally, for each connection $e_i \in E$ we define two variables $x_{i\rightarrow}, x_{i\leftarrow}$. The variables represent the number of times each edge is traversed in a path, and so $f_P(e_i) = x_{i\rightarrow} + x_{i\leftarrow}$. Each variable is noted x^I, x^B or x^R for interval, bridge or reference edges respectively. Using these variables we can describe the problem as follows. Let $G(V, E, w)$ be a bridge graph and $x = (x_1, x_2, \dots)$ corresponding to edges in E as described above.

Minimize:

$$d_G(f_P) = \sum_{e\in E_I} \frac{l_e}{L} |x^I_{e\rightarrow} + x^I_{e\leftarrow} - w(e)| + \alpha \sum_{e\in E_V} \frac{w(e)}{\mu} \big(1 - min(1, x^B_{e\rightarrow} + x^B_{\leftarrow})\big)$$

Subject to:

(1) $\forall_i x_i \in \mathbb{N}^0$

(2) $\forall_{v\notin S} \sum_{e_i\in E_{I_\rightarrow}(v)} x^I_{i\rightarrow} = \sum_{e_i\in E_{R_\leftarrow}(v)} x^R_{i\leftarrow} + \sum_{e_i\in E_{V\leftarrow}(v)} x^B_{i\leftarrow}$

(3) $\forall_{v\notin S} \sum_{e_i\in E_{I_\leftarrow}(v)} x^I_{i\leftarrow} = \sum_{e_i\in E_{R_\rightarrow}(v)} x^R_{i\rightarrow} + \sum_{e_i\in E_{V\rightarrow}(v)} x^B_{i\rightarrow}$

Where constraints (2) and (3) are the valid path constraints detailed in section 2.2 applied to the edge variables. Note that telomeric nodes in S are not constrained.

3 Simulation results

This section describes the simulations we have performed in order to test the performance of the algorithm.

3.1 Simulation setup

In order to assess the performance of our algorithm, we simulated tumor karyotypes by starting with a normal karyotype, performing on it a sequence of structural and numerical changes, and then adding noise to the results. Each correct karyotype was compared to the karyotype that was produced by the algorithm using the input data, and summary statistics were computed.

We start with a normal diploid karyotype H with a prescribed number of chromosomes (a parameter). For simplicity, each chromosome is represented by a sequence of atomic segments, which are its basic units. We perform a series of operations on the karyotype by applying deletions, inversions, tandem duplications and translocations. The types and the positions of the rearrangements are drawn uniformly at random. The span of operations that affect a single chromosomes (deletions, duplications and inversions) is limited to 30 atomic segments. This limit is set in order to avoid rapid erasure of large chromosomal segments by deletions. The total number of operations applied varies and determines the complexity of the resulting tumor karyotype T.

By comparing H and T, breakpoints are detected and each normal chromosome is partitioned into segments. Each segment has a copy number (the number of occurrences of that segment in T). Each two consecutive segments in T that are not consecutive (and/or not in the same relative orientation) in H constitute a bridge. The clean (noiseless data) can thus be summarized as an integer-valued CN profile and the set of all bridges formed.

To simulate noisy scenarios, the CN profile and the bridge information is modified as follows. Normally distributed noise x is added to the copy number of each segment independently, where $x \sim N(0, \epsilon)$. The support for each bridge (corresponding to the number of discordant reads supporting it) is drawn independently from an exponential distribution $Exp(\lambda)$. To simulate the possibility of bridges being completely missed, each bridge has probability p to completely be omitted from the final set of bridges.

In summary, the simulation program receives the following parameters, with the default value in parentheses:

- C - The number of chromosomes (default: 5).
- N - The number of structural and numerical operations applied (default: 5).
- ϵ - The standard deviation of the noise in the CN profile data (default: 0.28)
- p – The probability to completely miss a bridge (default: 0.05).

In the *base scenario*, all parameters were at their default values. These parameters correspond to a tumor sample of medium complexity and a realistic level of noise (see section 4.1). Other scenarios were explored by changing one of the parameters above from its value in the base scenario while keeping the rest at their default levels.

3.2 Correctness measures

We used five different measures for the level of correctness of a solution. Let T be the simulated (true) karyotype, let T^* be the simulated noisy karyotype, and let S be the karyotype produced by the algorithm:

1. Is S equivalent to T? We say that S is *equivalent* to T if they have the same copy number profile and both use the same bridges. Most equivalent karyotypes only differ in chromosomal orientation, and thus represent the same solution. However, in rare cases two different karyotypes can be equivalent and differ in other ways (See Figure 3.1). Our algorithm was not designed to distinguish between such solutions. This score is our main success yardstick. We call such a solution *correct.*
2. Do S and T have the same CN profile? In real data, the copy number of an interval is determined by summing over numerous reads (or probes), sometimes spanning many megabases, while bridges rely on few paired-end reads crossing a particular point, and thus are more error prone. Therefore, the CN profile is expected to be more robust. This criterion tests if S and T match in this profile. We call this criterion *Equal Copy Number* (ECN).
3. Does S have an equal or better score than T? When noise level is high, T and T^* may differ substantially, and a solution closer to T^* than to T does not indicate a failure of the algorithm but rather that the noise level is too high. Here the score was calculated according to the ILP objective. We call this criterion *Equal or Better Score* (EBS).
4. Is S equivalent to T excluding missing bridges? T^* may not include all the bridges found in T, and in that case S can never be equivalent to T. However, we consider S to be correct for all observed bridges if it has the correct CN profile for all segments that are unaffected by a missed bridge, and is using all the bridges from T that are included in T^*. This requires to take into consideration not only the bridges missing from T^*, but also all other edges whose assigned weight might be affected due to the removal of the bridge. Specifically, a missing bridge (u, v) forces the algorithm to use a different path between u and v and so the values assigned to edges in that path will differ from their values in T. Exactly which edges are considered to be affected and thus ignored depends on the specific type of operation that gave rise to that bridge. An illustration of edges affected by the removal of bridges can be seen in Figure 3.2. We call this metric *Equivalent for Observed Bridges* (EOB).
5. What fraction of the intervals have the correct copy number? This score is the percentage of intervals, weighted by length, that have the same copy number in S and T. Unlike criteria 1-4, which are binary, this criterion measures the extent of correctness of a solution, and thus is more sensitive and accounts also for partially-correct solutions. We call it the *CN score.*

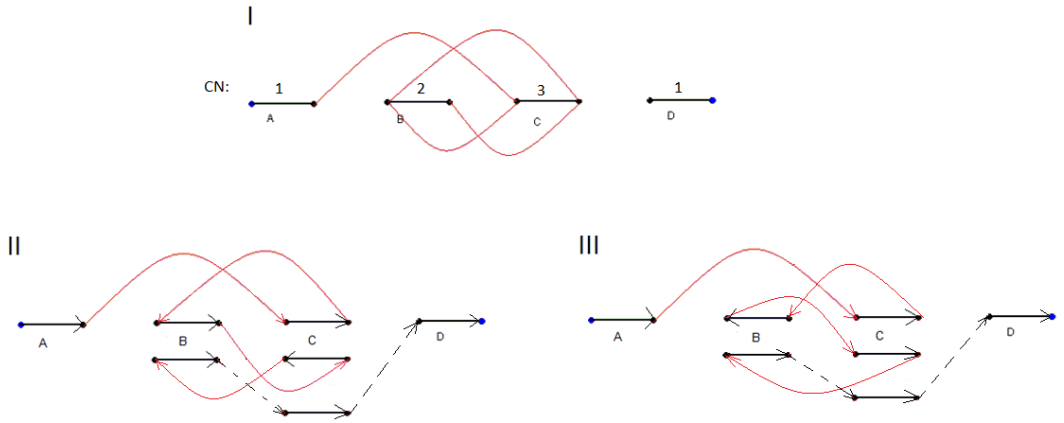

Figure 3.1: An example of graphs that represent equivalent yet not identical solutions. Graph (I) can have a path of (II) $A \to C \to \boldsymbol{B} \to -\boldsymbol{C} \to B \to C \to D$ or (III) $A \to C \to -\boldsymbol{B} \to \boldsymbol{C} \to B \to C \to D$. Both paths start and end in telomeric nodes (marked blue) and use the exact same set of bridges. In (I) the numbers above the solid edges are copy numbers. In both alternative paths, the last part of the path is connected by reference edges.

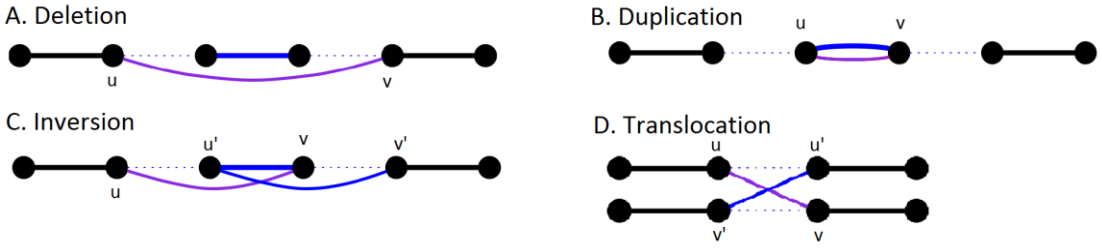

Figure 3.2: Edges affected by removing a bridge. In each of the four cases, the purple edge (u, v) is the missing bridge and the blue edges are affected by the removal. (A) In the case of deletion, the removal of the bridge will affect all edges in the path $u \to v$. (B) For duplication, the weight assigned to the duplicated segment $[u, v]$ will be affected. (C) Inversion creates two bridges. When one is omitted this will affect the score of the other bridge (u', v') and the inverted segment $[u', v]$. (D) A translocation creates two bridges. When the bridge (u, v) is omitted this will affect the other bridge (v', u') and the two corresponding reference edges.

3.3 Performance in the base scenario

10,000 karyotypes were generated for the base scenario, and the algorithm was applied with bridge support weight $\alpha = 0.1$. The performance is summarized in Figure 3.3.

To assess the distribution of each success rate criterion, the karyotypes were divided into 100 batches of 100 karyotypes each. Mean scores were captured for each batch and the variation of the mean was computed. (Figure 3.3).

The algorithm correctly identified between 55% and 73% of the karyotypes in each batch, with an average of 62%. For an additional 13% of the cases, the solution had an equal CN profile as the correct solution, a total of 75%. An average of 82% of all karyotypes resulted a solution with a score equal or better than the correct one. When disregarding missing bridges, the algorithm correctly identified an average 84% of karyotypes.

The mean CN score of all the 10,000 simulations was 0.97 with a small standard deviation of 0.009.

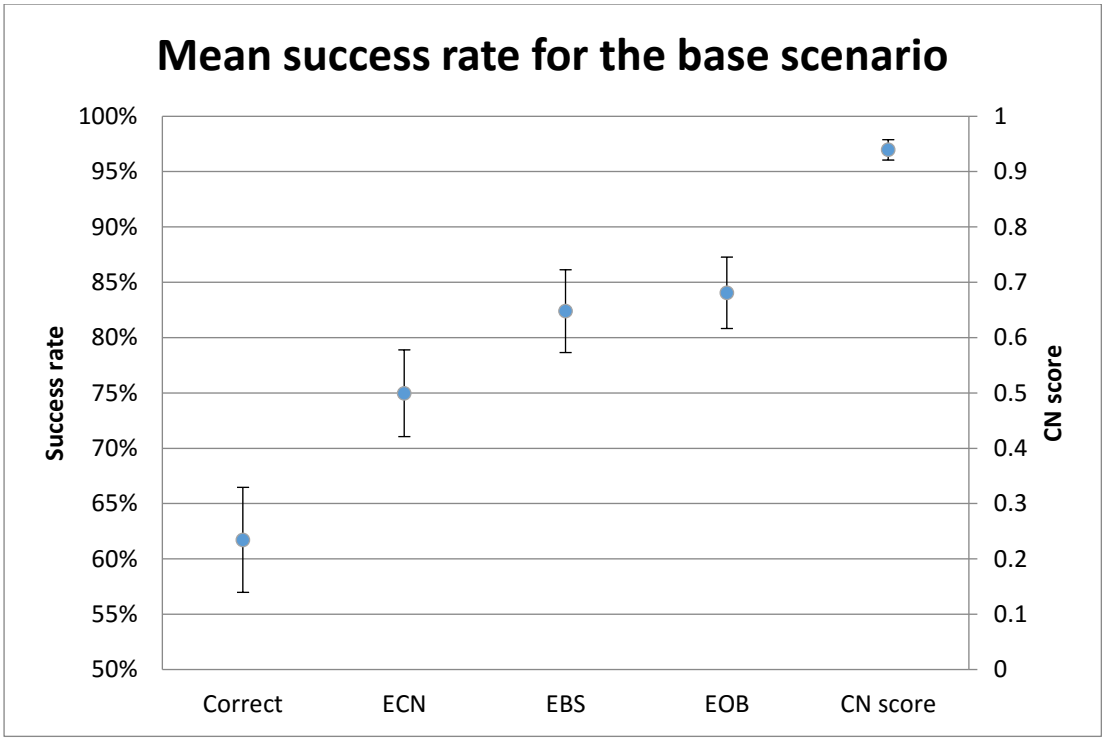

Figure 3.3: Distribution of the success rate over 100 independant simulations of the base scenario. Error bars are ± the standard deviation.

3.4 The effect of separate parameters

The effect of separate parameters was tested by simulations in which one parameter was altered, while keeping the other parameters at their value in the base scenario. 100 simulated karyotypes were generated for each value and the percentage of solutions falling into the categories of correct, ECN, EBS and EOB was evaluated.

3.4.1 The effect of bridge support weight in the objective

We first tested the effect of α on the performance for $0 \leq \alpha \leq 2$. Recall that α is the relative weight assigned the bridges data in the ILP formulation (See 2.2.1). There is a noticeable improvement when $\alpha > 0$, and little effect for the range of $0 < \alpha \leq 0.1$. For larger values of α there is a small but noticeable negative effect. (Table 3.1).

Alpha	Correct	ECN	EBS	EOB
0.00	13%	82%	85%	17%
0.01	67%	82%	85%	82%
0.02	67%	83%	86%	90%
0.03	67%	83%	86%	90%
0.04	67%	82%	85%	89%
0.05	67%	81%	84%	90%
0.10	67%	80%	83%	89%
0.25	67%	77%	81%	86%
0.50	67%	73%	77%	82%
1.00	67%	69%	73%	78%
2.00	67%	68%	72%	77%

Table 3.1: Performance of the algorithm for different values of the parameter alpha.

3.4.2 The effect of noise in copy number measurements

We tested the algorithm for different levels of CN noise ϵ under the base scenario. The results are shown in Figure 3.4. As expected, a higher level of noise makes it harder for the algorithm to find the correct solution. For $\epsilon < 0.4$ the performance of the algorithm is quite good, and for $\epsilon \geq 0.4$ the results begin to deteriorate. Naturally, at high noise levels the majority (82%) of the solutions have better score than the true one.

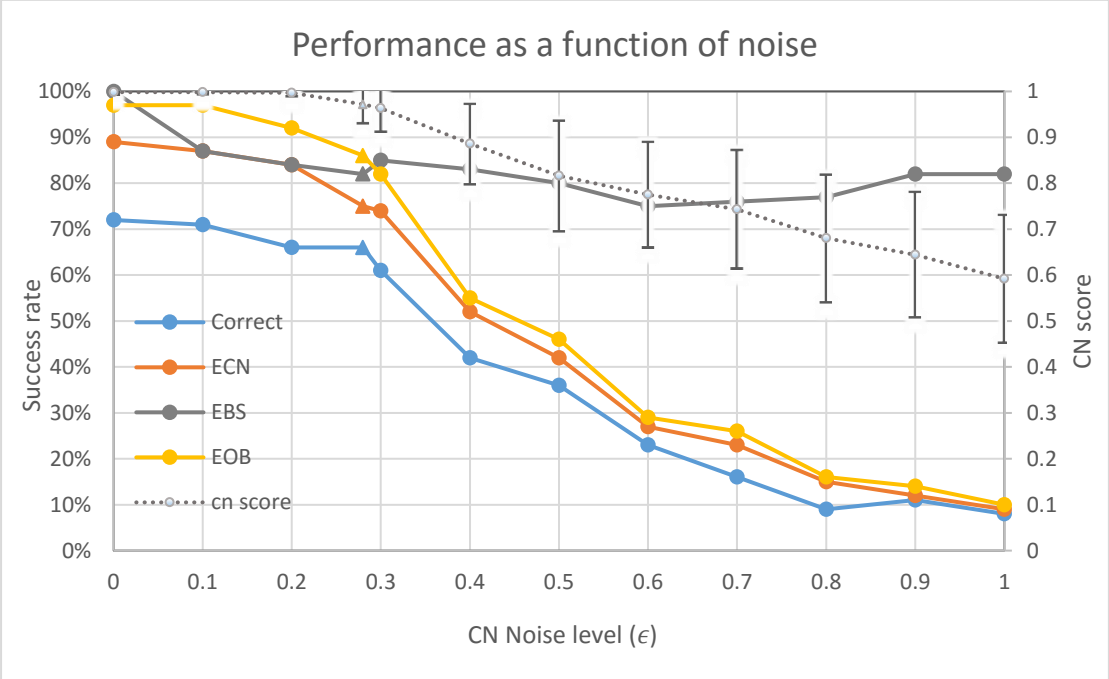

Figure 3.4: Performance of the algorithm as a function of noise level. For the CN score, the bars represent ± 0.5 std. Data points for the default value of $\epsilon = 0.28$ are marked with a triangle.

3.4.3 The effect of the number of operations

We tested the algorithm on karyotypes that underwent $1 \leq N \leq 30$ structural and numerical operations, under the base scenario. The results are shown in Figure 3.5. As expected, more operations make the problem harder and the success rate decreases, from 88% with one operation to less than 10% with 30 operations. The gap between the four scores of the binary measurements grows throughout the changes, with the exception of ECN and EBS which remain close together.

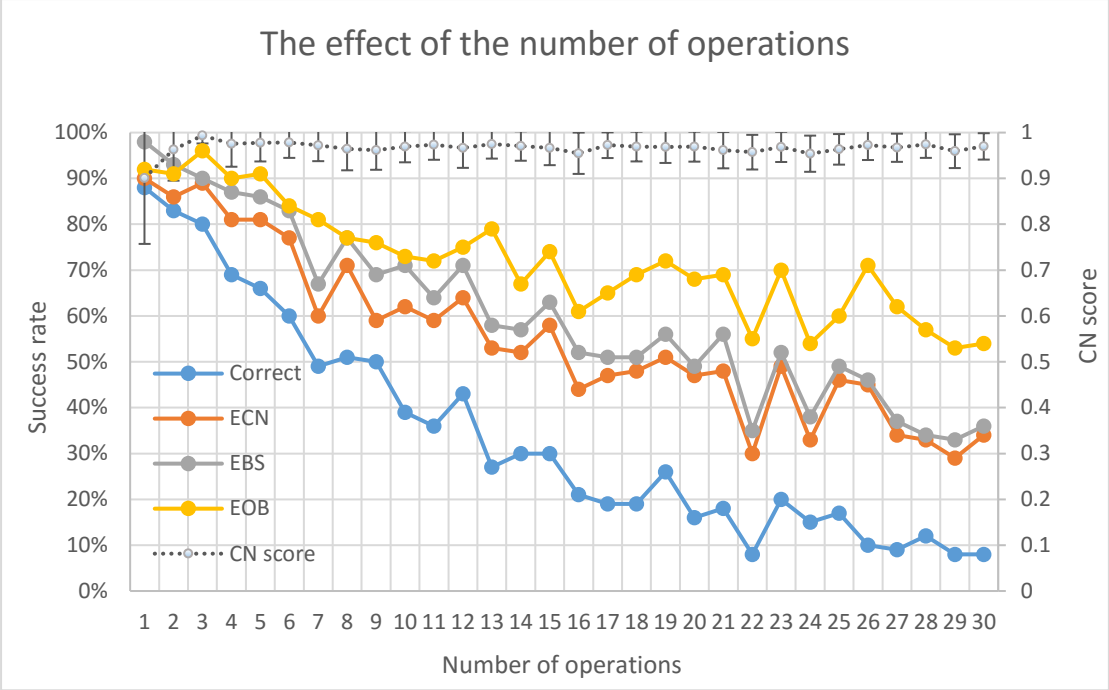

Figure 3.5: The effect of the number of operations. Success rates and copy number scores. Error bars represent ± 0.5 std.

3.4.4 The effect of the number of chromosomes

We tested the algorithm on karyotypes simulated with varying number of chromosomes $C \in \{1,4,7,10\}$. The results (Figure 3.6) were better for the case of only one chromosome, with some difference between the other cases. Note that the number of operations remains at the default value of $N = 5$, so with more chromosomes the changes due to operations are sparser.

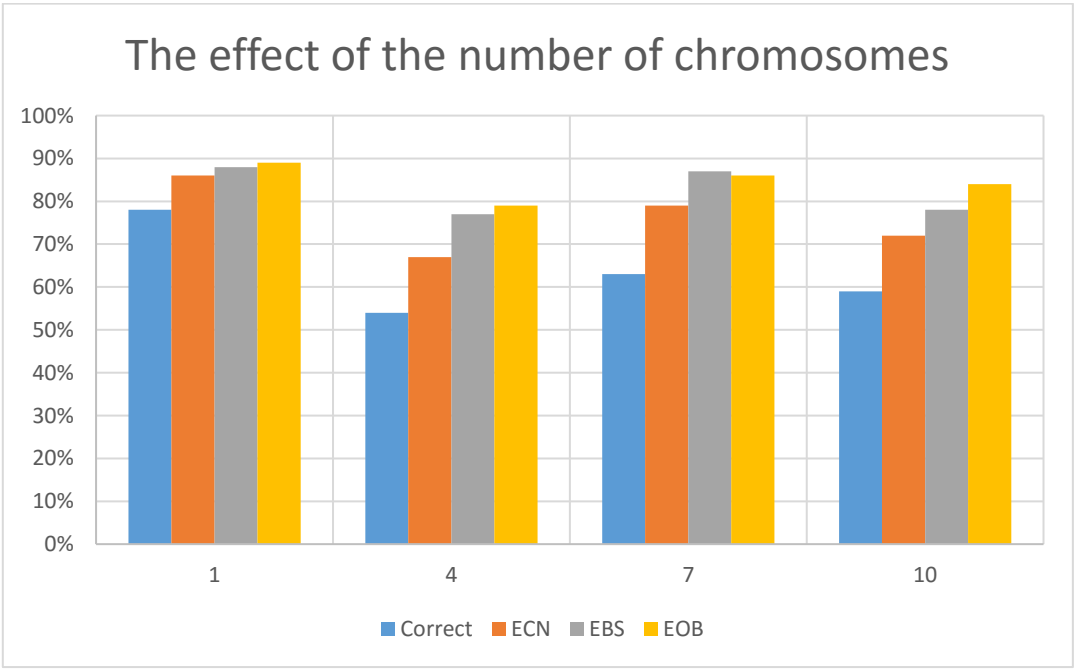

Figure 3.6: The effect of the number of chromosomes on the success rate of the algorithm.

3.4.5 The effect of chromosome ploidy

We measured the performance of the algorithm for karyotypes with diploid chromosomes and for karyotypes with a single copy of each chromosome. As expected (Figure 3.7), the results were better for karyotypes with single copy of each chromosome – but only slightly.

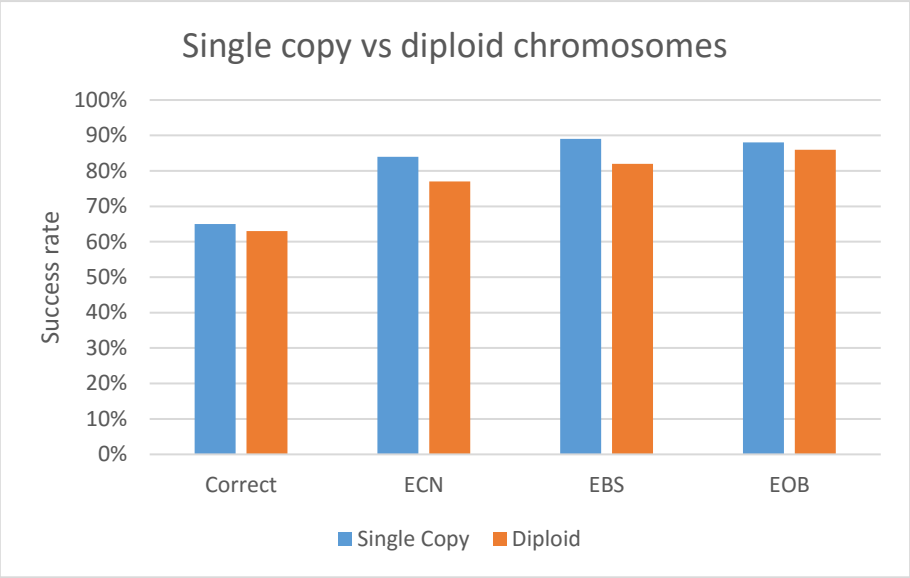

Figure 3.7: Single copy vs diploid chromsomes.

3.4.6 The effect of missing bridges

We tested the algorithm with different probabilities of completely missing a bridge $0 \leq p \leq 0.15$. As expected, the results are better when the probability of missing a bridge is smaller (Figure 3.8). Furthermore, the EOB score remains high even for bigger values of p, as expected.

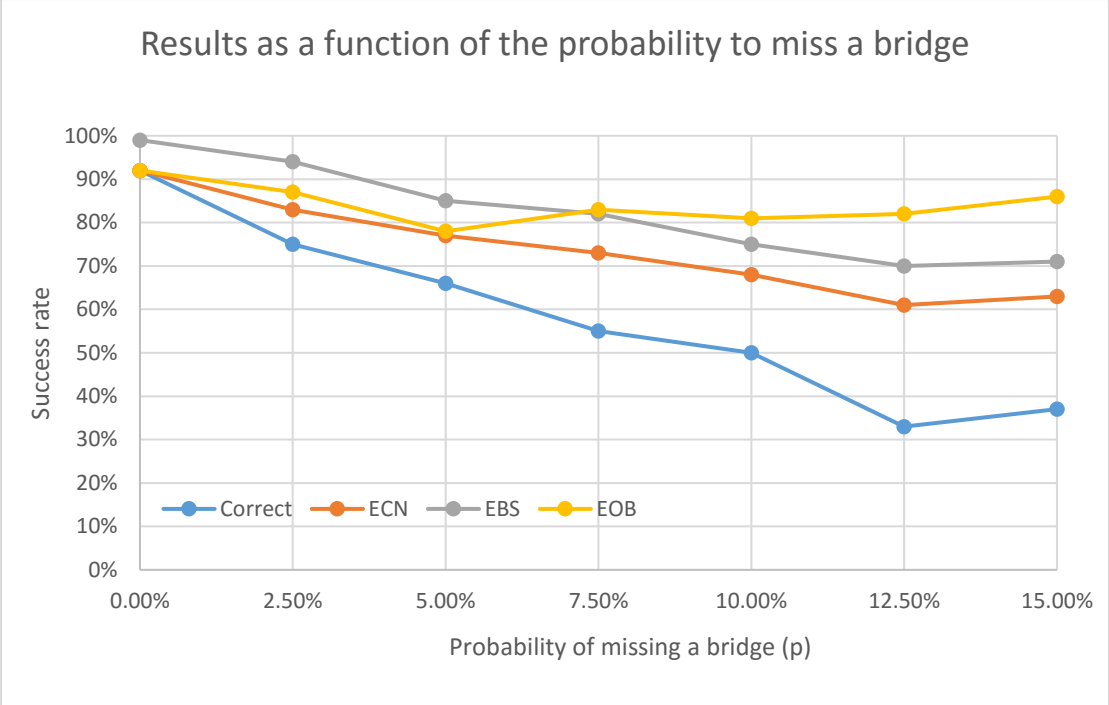

Figure 3.8: Success rate as a function of the probability to miss a bridge.

3.4.7 The effect of the different operations

To test the effect of operation frequency, we also simulated karyotypes by selecting operations with frequencies as reported in [20] and rounded to multiples of 10%. Table 3.2 shows the distributions. 250 karyotypes were generated under each distribution. As seen in Figure 3.9, there is little difference in the success rates between the uniform distribution and the uneven one.

Type	Uniform	Actual Malhotra data	Simulations
Deletion	25%	43%	40%
Duplication	25%	38%	40%
Inversion	25%	12%	10%
Translocation	25%	7%	10%

Table 3.2: operations frequencies used in the default scenario and in the alternative scenario.

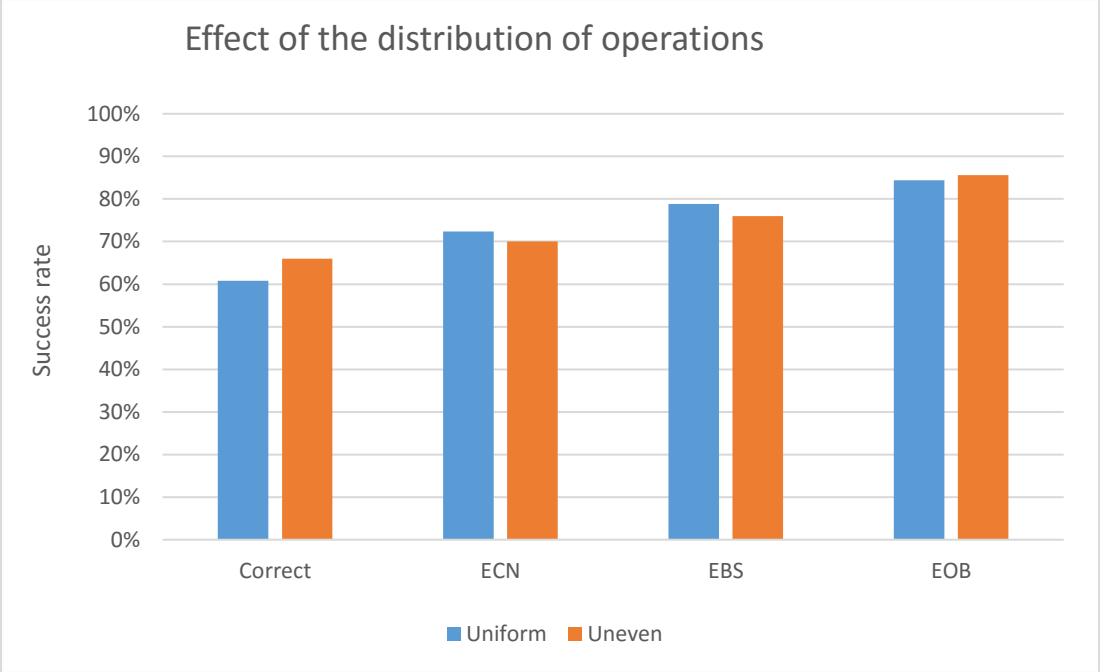

Figure 3.9: Results for different operation frequencies.

3.4.8 The effect of tumor heterogeneity

We tested the algorithm on simulated data of tumors that are heterogeneous. We first simulated a sample that contains, aside from our tumor karyotype, the normal karyotype in rates of up to 45% (Figure 3.10). This simulates the situation where the sample is a mixture of normal and tumor cells. Results were slightly better for more homogenous samples, but overall the algorithm was able to achieve a success rate of around 60% even for samples that contain only 55% tumor cells.

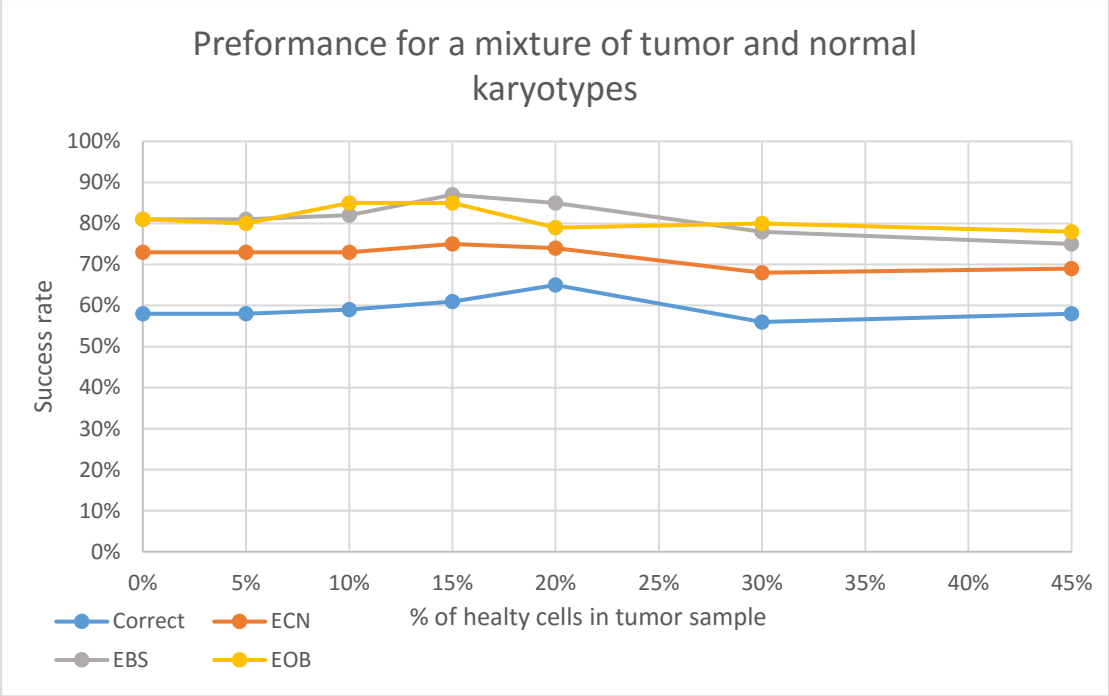

Figure 3.10: Performance for tumors with different levels of contamination from healthy cells karyotypes.

We also tested the algorithm in a scenario where the sample is a mixture of two different mutated karyotypes. The lower frequency karyotype was 0%, 5%, 10%, 15% and 20%. The dominating target karyotype underwent 5 rearrangements (as in the base scenario), while the lesser karyotype underwent an average of 3, about half of them unique and the rest are shared between the two. While this is not a strict evolutionary model, it mimics the situation where different karyotypes in the same tumor share some similarities. As expected, this proved to be a more difficult scenario and the ratio of correct predictions dropped quickly. The other score metrics exhibited a much slower decline however (Figure 3.11). Note that the evaluation is done in terms of reconstructing the dominating karyotype only.

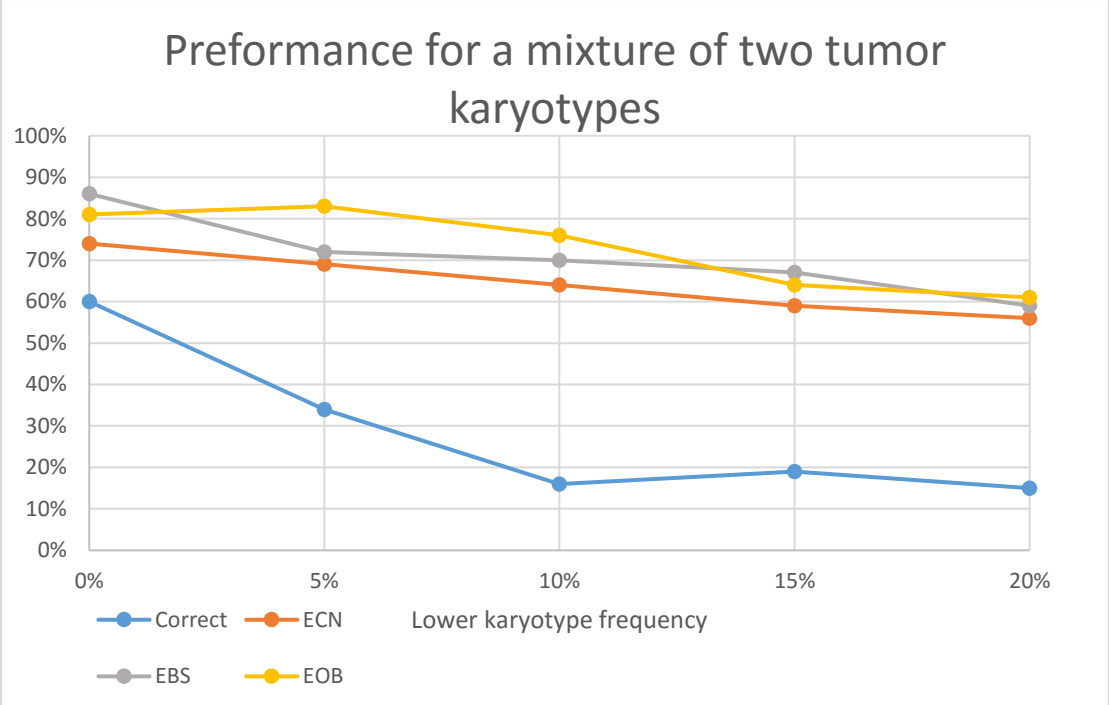

Figure 3.11: Performance of the algorithm on samples contaminated with a second, different karyotype.

4 Results on real tumor data

Our next step was to test the algorithm on data extracted from real samples. Malhotra et al. [20] examined 64 different tumor samples. In this study the reads in each sample were analyzed forming both a CN profile and a set of bridges with their support. Often the set of normal chromosomes that are involved in rearrangements and CN changes in a tumor can be partitioned into several sets of chromosomes that are independent of one another (i.e., have no segments that form a bridge between them). In terms of our graph representation, each such set is a connected component, which can be analyzed separately by the algorithm. The 64 tumor samples in [20] constituted together 570 such components, and each was analyzed separately.

4.1 Estimation of noise in the real data.

We wanted to assess the noise level in the actual data affecting the reported copy number values. Since CN in noiseless data should be integer, we estimate the noise d_i for the reported copy number c_i as $c_i - [c_i]$, where $[x]$ is the nearest integer value to x. Note that d_i can be either negative or positive and cause the reported CN to be higher or lower than the true value respectively. Obviously $d_i < 0.5$, and therefore d_i is a lower bound for the real noise level. For lack of better yardstick, we use this value in lieu of the actual noise.

The CN data include 22,321 copy number segments. As expected, the mean noise level across the data was 0, showing that the noise is unbiased towards negative nor positive values. The standard deviation was 0.28, a value that we used as our default scenario (see 3.1). A scatter plot of the standard deviation of the noise level vs. the number of bridges in each component can be seen in Figure 4.1.

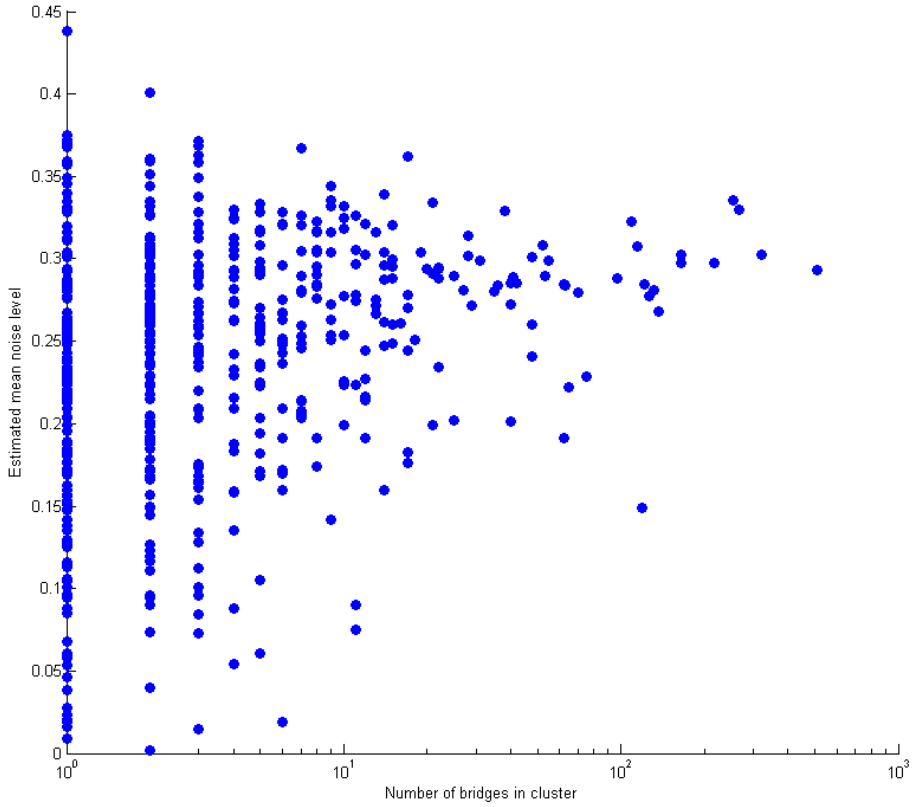

Figure 4.1: Estimated noise level in real cancer samples. The plot shows for each of the 670 components in the tumor samples in [20]*, the number of bridges and an estimate of the noise level calculated as standard deviation of the distances of the CN in the sample from the closest integer value.*

In addition to copy numbers, the data include bridges and for each bridge an integer value, called *support,* representing the number of paired end reads (PERs) supporting that bridge. The expected average support can be derived from the read depth and the insert size. We assume that in order for a bridge to be supported by a PER, the breakpoint causing it has to fall within the gap of the PER's insert. In other words, each read of the PER has to be mapped in full to one of the two sides of the breakpoint. Let ins be the total insert length and end be the length of each end, so that the read gap is $gap = ins - 2 * end$. The depth of coverage is the average number of times a base is sequenced, i.e. covered by one of the ends (as the gap is not sequenced). Equivalently, it is the average time it is covered by an end. Hence, the expected support score for a given breakpoint is $E_{supp} = \frac{d}{\frac{(2*end)}{ins}} * \frac{gap+1}{ins} = d * \frac{gap+1}{2*end}$. In the data examined the mean size of each read is $ins = 242$, with mean end length of $end = 95$. The average coverage is $d = 40$, and so the expected support for a given bridge is

$$E_{supp} = 40 * \frac{242 - 2 * 95 + 1}{2 * 95} = 10.7$$

The observed mean support score across all the data was 10.8. Figure 4.2 shows the distribution of the support scores across the data. A total of 6170 bridges were reported. Ignoring a few bridges with unusually high support, 6131 bridges (99%) with support score lower than 100 had mean score of 8.63 and standard deviation of 8.44. The support scores

across the real data closely resemble an exponential distribution with $\lambda = 0.1866$ (Figure 4.3), the distribution used in our simulation model.

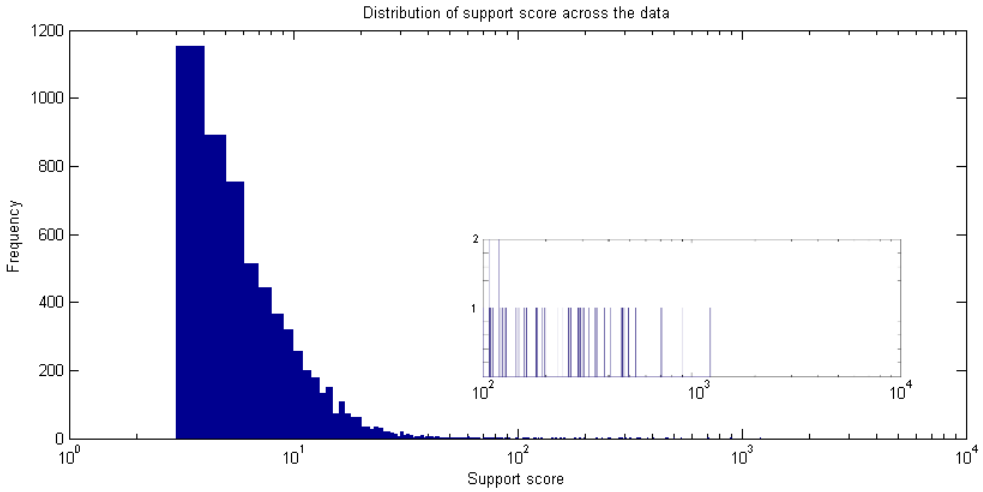

Figure 4.2: Histogram of bridge support scores across the data. Bridges with support score≤2 are not included in the data. The inlaid plot shows the distribution of the support scores for values >100.

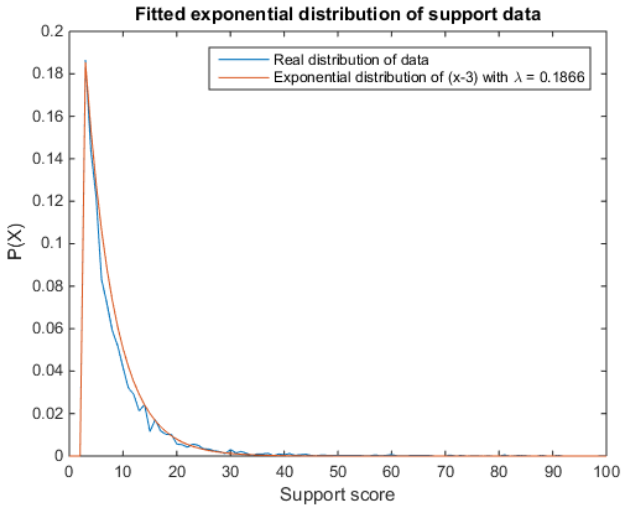

Figure 4.3: The distribution of the support score across the data plotted against an exponential distribution with $\lambda = 0.1866$. In both distributions values below 3 are ignored.

4.2 Results on selected samples of real tumor data

Table 4.1 shows information about three components that we analyzed in detail. Each has undergone 7-8 rearrangements, involving 1-4 chromosomes. For each component, the ILP algorithm outputs a directed weighted graph with a weight function that minimizes the distance and that can be broken into a set of paths $P = \{p_1, \dots p_n\}$, starting and ending at a telomere nodes, and alternating interval and non-interval edges. Another script translates the solution of the ILP solver to a dot language representation [93] that can then be visualized using a graph visualization tool such as GraphViz [94].

Sample	Chromosomes comprising the component	Number of bridges	Number of CNV's
LUAD_6	1	8	40
GBM_10	4, X	7	40
LUSC_5	6, 12, 15, 16	8	28

Table 4.1: Components from the Malhotra data [20] *the algorithm was tested on.*

4.2.1 Sample GBM 10

Figure 4.4A shows the graph corresponding to the component of chromosomes 4 and X in tumor sample GBM 10 (Glioblastoma multiforme). The chromosomes were divided into segments according to the breakpoints inferred from the paired ends reads data and were named a-l. Segment sizes are not shown to scale in the figure. The number above each segment (interval edge) is the observed copy number of the corresponding genomic region, while the number next to a red edge (bridge) is the number of observed supporting reads for that bridge.

The resulting karyotype suggested by our algorithm for this example is shown in Figure 4.4B. This graph can be broken into four different paths, representing both copies of the rearranged chromosomes 4 and X (Figure 4.4C).

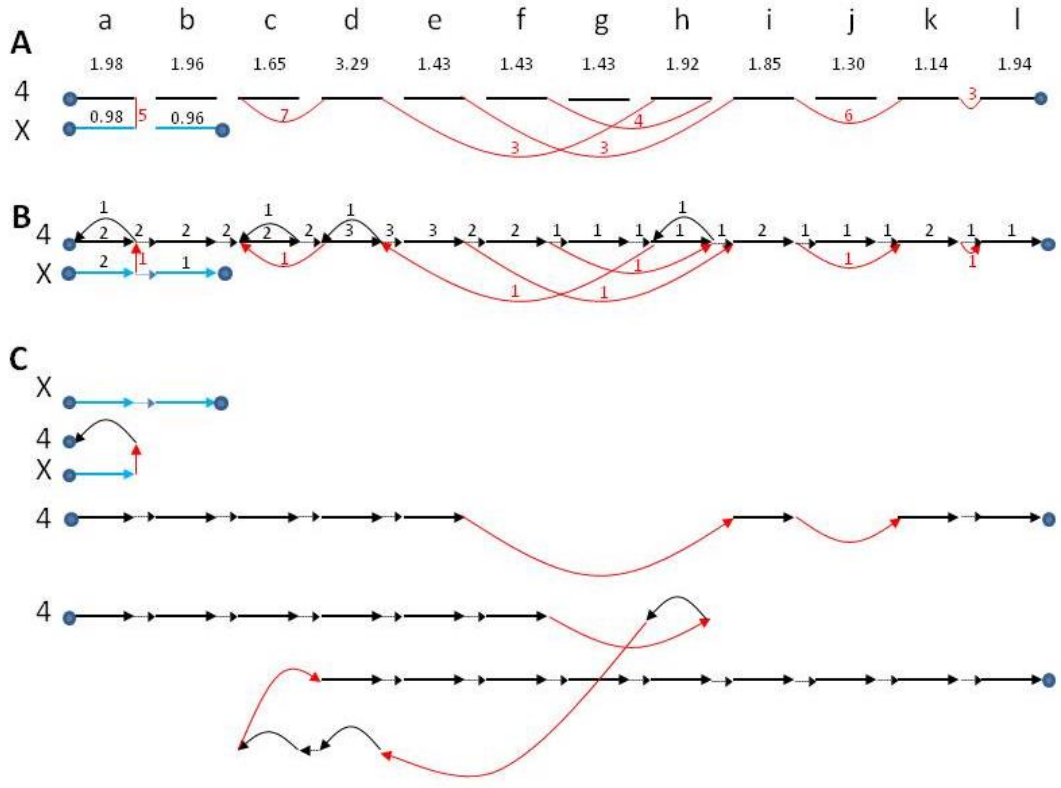

Figure 4.4: Results on sample GBM 10. In these figures we mark interval, reference and bridge edges by black, dotted and red arcs respectively. In all subfigures the same intervals (here: a through l for Chr. 4 and a,b for Chr. X) are aligned. The numbers in the second line are observed coverage values. (A) Bridge graph for chromosomes X and 4. The bridge between segments k and l is a result of our breakpoint filterring (see section 2.1.4). (B) Solution suggested by our algorithm. For this sample the average distance of the resulting karyotype from the data, weighted by segment length, is 0.28. Note that segments a,c,d, and h have edges in both directions suggesting the solution includes traversal of these segments in both directions. (C) The different paths comprising the solution, representing the rearranged karyotype of chromosomes 4 and X.

4.2.2 Sample LUAD 6

Figure 4.5 shows the bridge graph constructed from the data of chromosome 1 in the tumor sample LUAD_6 (Lung Adenocarcinoma). This figure and the next one were automatically drawn by the graphical software GraphViz [94] using the output of the algorithm.

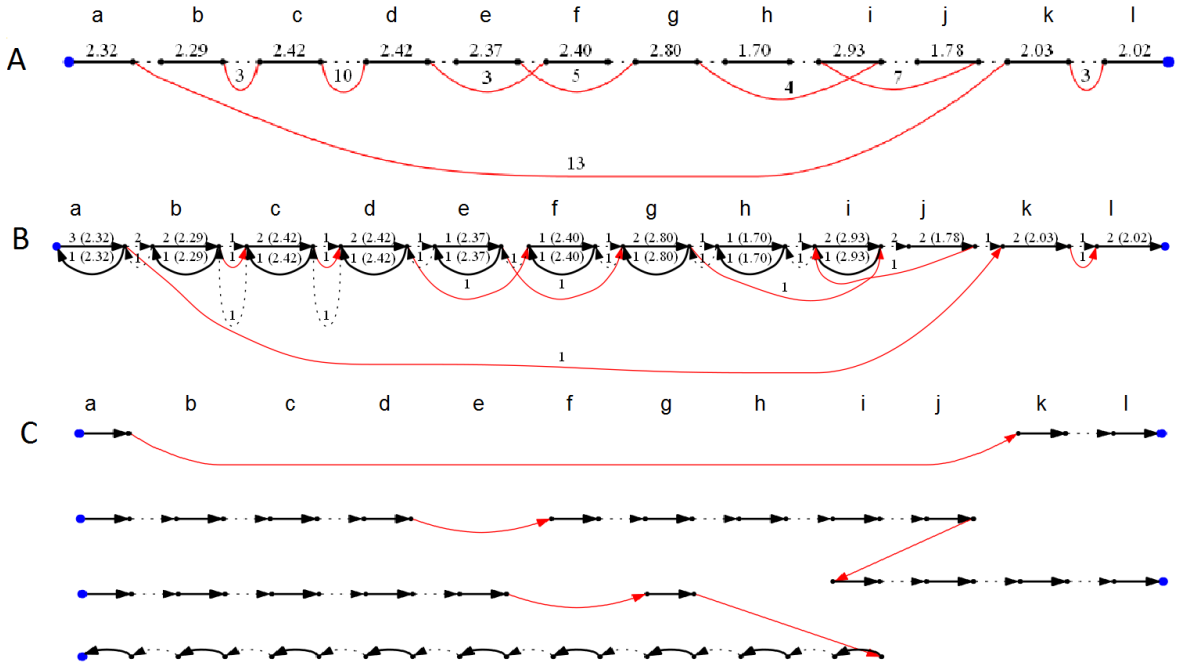

Figure 4.5: Results of sample LUAD 6. (A) Bridge graph for chromosome 1. (B) Solution suggested by our algorithm. For this sample the average distance of the resulting karyotype from the data, weighted by segment length, is 0.24. (C) The different paths comprising the solution, representing the rearranged karyotype of chromosome 1.

4.2.3 Sample LUSC 5

Figure 4.6 shows the graph corresponding to the component comprising of chromosomes 6, 12, 15 and 16 of tumor sample LUSC 5 (Lung squamous cell carcinoma).

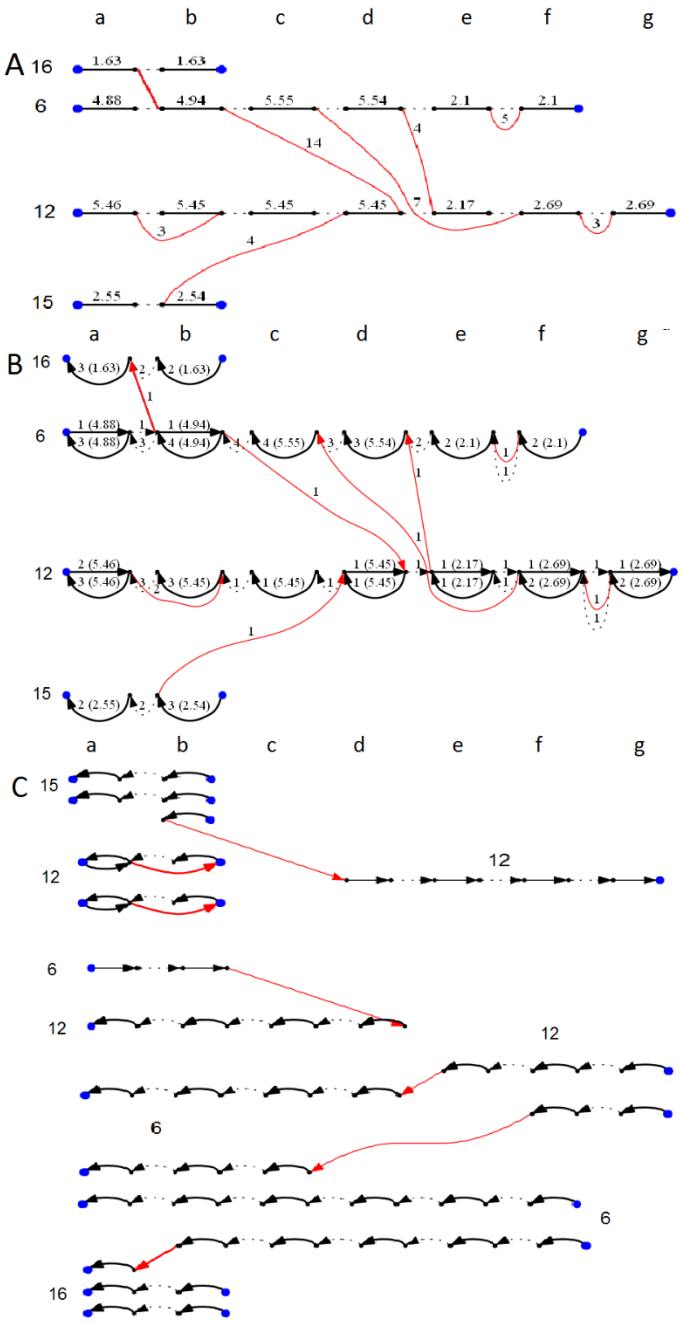

Figure 4.6: Results of sample LUSC 5. (A) Bridge graph for chromosomes 6, 12, 15, 16. (B) Solution suggested by our algorithm. For this sample the average distance of the resulting karyotype from the data, weighted by segment length, is 0.24.(C) The different paths comprising the solution, representing the rearranged karyotype of chromosomes 6, 12, 15, 16.

5 Discussion

In this work, the problem of inferring a tumor karyotypes from short paired end read data was investigated. A novel algorithm based on graph theory and ILP was introduced to solve the problem, and simulations were performed in order to evaluate the utility of such an approach. Some examples of analysis of real data were also presented.

5.1 Overall success rate

To accurately estimate the correctness and robustness of the algorithm, validation against a data set of verified karyotypes is needed. However, a comprehensive set of sequenced tumor samples with copy number profiles and paired-end reads data, matched with entire reconstructed karyotypes, is not currently available. Data sets that currently exist either do not include a fully reconstructed karyotype, or include karyotypes of a very low resolution, such as the Mitelman database [41]. We therefore used a simulation model to test and measure the success of our algorithm in a spectrum of scenarios, as well as to point out potential pitfalls.

The analysis of simulated data suggests that most meaningful factors affecting the accuracy of solutions produced by our method are the noise and completeness levels of the data. We tested the algorithm in a scenario, designed according to observations in real data. Under these conditions, the algorithm correctly inferred 69% of the karyotypes. However, the success rate increased to 79% when considering solutions that are correct relative to the noisy input, and when accounting for unreported bridges, 87% of the tested cases were correct (Figure 3.3).

Furthermore, in scenarios where there is almost no noise, or when no bridges are unreported, the results are much better: accuracy was 90% and 100%, respectively (Figure 3.4, Figure 3.8). This strongly suggests that our method is limited mostly by the completeness and accuracy of the measured data. It suggests that more accurate sequencing technologies are needed in order to increase the chance to solve the karyotype reconstruction problem correctly.

We have also shown our method to be robust when implemented on data taken from tumor cells contaminated by healthy tissue (Figure 3.10). A sample that includes reads taken from a mixture of different tumor cells poses a bigger challenge, and the resulting karyotype is incorrect more often than it is correct (Figure 3.11).

5.2 Limitations of the simulation model

Using simulations allows us to gain better understanding of the capabilities and limitations of our algorithm, but it requires us to make assumptions about the mechanisms driving genomic rearrangements in tumor cells and about the statistical properties of the read data. Both types of assumptions limit the generality of conclusions we can draw.

Firstly, our model defines a limited set of possible rearrangements (deletion, duplication, inversion and chromosomal translocation) and assumes that they occur with equal probabilities. Furthermore, the simulation of rearrangement events (except translocations) limits the genomic range they can span (see 3.1) and assumes that events are equally likely to occur in any position on the genome. While these assumptions are very far from the real process of mutating cancer cells, they do provide a mechanism that can generate any rearranged karyotype. Our method proved robust when changing the frequency of each type of rearrangement from equal to that observed in the data obtained from Malhotra et al.

(Figure 3.9), but other possible rearrangement mechanisms and their possible effect on the performance of the algorithm were not explored.

A second problem arises when attempting to create very complicated karyotypes using a large number of rearrangements. While all possible karyotypes can be generated using our model, very complex ones are unlikely. Note that once a deletion operation has been performed, the deleted segment cannot reappear and will therefore be absent from the final karyotype. When performing a large number of rearrangements on a chromosome, deletions will occur and sometime remove segments that were rearranged by a previous operation, essentially limiting the complexity of the resulting final karyotype. We tested our method on karyotypes that have undergone a maximum of 30 operations (Figure 3.5), but a modified simulation model needs to be used in order to generate more complex karyotypes. Currently our results reflect more faithfully the ability of the algorithm on relatively simple karyotypes, which constitutes the majority in real data.

A third type of limitation is due to the noise model assumptions. While we tried to borrow values of noise as estimated from the real data (section 4.1), there are other parameters that affect the noise and thus the quality of the analysis, including incorrectly mapped reads due to sequencing errors, non-uniquely mappable reads, insert length variance, breakpoints that fall within a read (and not in the gap), non-uniform read coverage, etc. These are all left to future work.

5.3 Future directions

One of the limitations of our algorithm is its inability to "predict" bridges that were not observed in the data. The algorithm will look for a path on the graph corresponding to a karyotype that best fits the observed CN profile, yet it will overlook potential paths that can be constructed by bridging two unconnected interval edges – essentially predicting a bridge. This implies that data produced using sensitive methods, even with higher rates of false positives, might be preferable over data with false negatives.

One important aspect of the technology in detecting bridges is the size of the insert gap, i.e. the part of the PER insert that is not read. A bridge will usually be detected only when the two reads of a PER are on the two different sides of it (see section 4.1). Therefore, the larger the gap in the insert - the higher the bridge coverage. This implies that sequencing techniques with longer inserts can dramatically change the performance of the algorithm. Several such techniques are forthcoming, and some methods for detecting structural variations were already developed for them [68] [69]. Note however that very short rearrangements that span less base pairs than the length of the read may be missed altogether.

Our algorithm currently focuses on finding a best path using the observed bridges only. Finding a best path including all possible unreported bridges requires considering an exponential number of possible paths. One approach can be for the algorithm to consider a limited number of possible unreported bridges. This is left for future work.

A possible extension to our method can be the addition of weights to the reference edges. Recall that reference edges represent a connection between two segments that is expected according to the reference genome. Unlike interval edges or variant edges, reference edges are weightless in our model. One metric that can be used to establish a confidence score for a reference edge is the number of PERs whose ends fall on the two segments bordering the reference connection.

6 References

[1] B. Vogelstein, N. Papadopoulos, V. E. Velculescu, S. Zhou, L. a Diaz, and K. W. Kinzler, "Cancer Genome Lanscapes," *Science (80-.).*, vol. 339, no. 6127, pp. 1546–1558, 2013.

[2] T. Boveri, *Zur Frage der Entstehung maligner Tumoren*. 1914.

[3] P. C. Nowell and D. A. Hungerford, "A minute chromosome in human chronic granulocytic leukemia," *Science (80-.).*, vol. 132, pp. 1488–1501, 1960.

[4] J. D. ROWLEY, "A New Consistent Chromosomal Abnormality in Chronic Myelogenous Leukaemia identified by Quinacrine Fluorescence and Giemsa Staining," *Nature*, vol. 243, no. 5405, pp. 290–293, Jun. 1973.

[5] M. Soda, Y. L. Choi, M. Enomoto, S. Takada, Y. Yamashita, S. Ishikawa, S. Fujiwara, H. Watanabe, K. Kurashina, H. Hatanaka, M. Bando, S. Ohno, Y. Ishikawa, H. Aburatani, T. Niki, Y. Sohara, Y. Sugiyama, and H. Mano, "Identification of the transforming EML4-ALK fusion gene in non-small-cell lung cancer.," *Nature*, vol. 448, no. 7153, pp. 561–6, Aug. 2007.

[6] G. R. Bignell, T. Santarius, J. C. M. Pole, A. P. Butler, J. Perry, E. Pleasance, C. Greenman, A. Menzies, S. Taylor, S. Edkins, P. Campbell, M. Quail, B. Plumb, L. Matthews, K. McLay, P. a W. Edwards, J. Rogers, R. Wooster, P. A. Futreal, and M. R. Stratton, "Architectures of somatic genomic rearrangement in human cancer amplicons at sequence-level resolution," *Genome Res.*, vol. 17, no. 9, pp. 1296–1303, 2007.

[7] B. McClintock, "The Production of Homozygous Deficient Tissues with Mutant Characteristics by Means of the Aberrant Mitotic Behavior of Ring-Shaped Chromosomes.," *Genetics*, vol. 23, no. 4, pp. 315–76, Jul. 1938.

[8] B. McClintock, "The Stability of Broken Ends of Chromosomes in Zea Mays.," *Genetics*, vol. 26, no. 2, pp. 234–82, Mar. 1941.

[9] J. A. Hackett, D. M. Feldser, and C. W. Greider, "Telomere dysfunction increases mutation rate and genomic instability.," *Cell*, vol. 106, no. 3, pp. 275–86, Aug. 2001.

[10] F. Toledo, G. Buttin, and M. Debatisse, "The origin of chromosome rearrangements at early stages of AMPD2 gene amplification in Chinese hamster cells.," *Curr. Biol.*, vol. 3, no. 5, pp. 255–64, May 1993.

[11] G. H. Rank, W. Xiao, A. Kolenovsky, and G. Arndt, "FLP recombinase induction of the breakage-fusion-bridge cycle and gene conversion in Saccharomyces cerevisiae," *Curr. Genet.*, vol. 13, no. 4, pp. 273–281, Apr. 1988.

[12] X. Bi, S.-C. D. Wei, and Y. S. Rong, "Telomere protection without a telomerase; the role of ATM and Mre11 in Drosophila telomere maintenance.," *Curr. Biol.*, vol. 14, no. 15, pp. 1348–53, Aug. 2004.

[13] D. Croll, M. Zala, and B. A. McDonald, "Breakage-fusion-bridge Cycles and Large Insertions Contribute to the Rapid Evolution of Accessory Chromosomes in a Fungal Pathogen," *PLoS Genet.*, vol. 9, no. 6, p. e1003567, Jun. 2013.

[14] T. Santarius, J. Shipley, D. Brewer, M. R. Stratton, and C. S. Cooper, "A census of amplified and overexpressed human cancer genes.," *Nat. Rev. Cancer*, vol. 10, no. 1, pp. 59–64, Jan. 2010.

[15] K. Kitada and T. Yamasaki, "The complicated copy number alterations in chromosome 7 of a lung cancer cell line is explained by a model based on repeated breakage-

fusion-bridge cycles.," *Cancer Genet. Cytogenet.*, vol. 185, no. 1, pp. 11–9, Aug. 2008.

[16] S. Selvarajah, M. Yoshimoto, P. C. Park, G. Maire, J. Paderova, J. Bayani, G. Lim, K. Al-Romaih, J. A. Squire, and M. Zielenska, "The breakage-fusion-bridge (BFB) cycle as a mechanism for generating genetic heterogeneity in osteosarcoma.," *Chromosoma*, vol. 115, no. 6, pp. 459–67, Dec. 2006.

[17] S. M. Bailey and J. P. Murnane, "Telomeres, chromosome instability and cancer.," *Nucleic Acids Res.*, vol. 34, no. 8, pp. 2408–17, Jan. 2006.

[18] D. Gisselsson, L. Pettersson, M. Höglund, M. Heidenblad, L. Gorunova, J. Wiegant, F. Mertens, P. Dal Cin, F. Mitelman, and N. Mandahl, "Chromosomal breakage-fusion-bridge events cause genetic intratumor heterogeneity.," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 97, no. 10, pp. 5357–62, May 2000.

[19] S. E. Artandi and R. A. DePinho, "Telomeres and telomerase in cancer.," *Carcinogenesis*, vol. 31, no. 1, pp. 9–18, Jan. 2010.

[20] A. Malhotra, M. Lindberg, G. G. Faust, M. L. Leibowitz, R. A. Clark, M. Ryan, A. R. Quinlan, and I. M. Hall, "Breakpoint profiling of 64 cancer genomes reveals numerous complex rearrangements spawned by homology-independent mechanisms," *Genome Res.*, vol. 23, no. 5, pp. 762–776, 2013.

[21] a. McPherson, C. Wu, a. W. Wyatt, S. Shah, C. Collins, and S. C. Sahinalp, "nFuse: Discovery of complex genomic rearrangements in cancer using high-throughput sequencing," *Genome Res.*, vol. 22, no. 11, pp. 2250–2261, Nov. 2012.

[22] P. J. Campbell, P. J. Stephens, E. D. Pleasance, S. O'Meara, H. Li, T. Santarius, L. a Stebbings, C. Leroy, S. Edkins, C. Hardy, J. W. Teague, A. Menzies, I. Goodhead, D. J. Turner, C. M. Clee, M. a Quail, A. Cox, C. Brown, R. Durbin, M. E. Hurles, P. a W. Edwards, G. R. Bignell, M. R. Stratton, and P. A. Futreal, "Identification of somatically acquired rearrangements in cancer using genome-wide massively parallel paired-end sequencing.," *Nat. Genet.*, vol. 40, no. 6, pp. 722–729, 2008.

[23] M. F. Berger, M. S. Lawrence, F. Demichelis, Y. Drier, K. Cibulskis, A. Y. Sivachenko, A. Sboner, R. Esgueva, D. Pflueger, C. Sougnez, R. Onofrio, S. L. Carter, K. Park, L. Habegger, L. Ambrogio, T. Fennell, M. Parkin, G. Saksena, D. Voet, A. H. Ramos, T. J. Pugh, J. Wilkinson, S. Fisher, W. Winckler, S. Mahan, K. Ardlie, J. Baldwin, J. W. Simons, N. Kitabayashi, T. Y. MacDonald, P. W. Kantoff, L. Chin, S. B. Gabriel, M. B. Gerstein, T. R. Golub, M. Meyerson, A. Tewari, E. S. Lander, G. Getz, M. A. Rubin, and L. A. Garraway, "The genomic complexity of primary human prostate cancer.," *Nature*, vol. 470, no. 7333, pp. 214–20, Feb. 2011.

[24] S. A. Tomlins, B. Laxman, S. Varambally, X. Cao, J. Yu, B. E. Helgeson, Q. Cao, J. R. Prensner, M. A. Rubin, R. B. Shah, R. Mehra, and A. M. Chinnaiyan, "Role of the TMPRSS2-ERG gene fusion in prostate cancer.," *Neoplasia*, vol. 10, no. 2, pp. 177–88, Feb. 2008.

[25] N. Cerveira, F. R. Ribeiro, A. Peixoto, V. Costa, R. Henrique, C. Jerónimo, and M. R. Teixeira, "TMPRSS2-ERG gene fusion causing ERG overexpression precedes chromosome copy number changes in prostate carcinomas and paired HGPIN lesions.," *Neoplasia*, vol. 8, no. 10, pp. 826–32, Oct. 2006.

[26] S. Perner, J.-M. Mosquera, F. Demichelis, M. D. Hofer, P. L. Paris, J. Simko, C. Collins, T. A. Bismar, A. M. Chinnaiyan, A. M. De Marzo, and M. A. Rubin, "TMPRSS2-ERG fusion prostate cancer: an early molecular event associated with invasion.," *Am. J. Surg. Pathol.*, vol. 31, no. 6, pp. 882–8, Jun. 2007.

[27] M. J. Soller, M. Isaksson, P. Elfving, W. Soller, R. Lundgren, and I. Panagopoulos,

"Confirmation of the high frequency of the TMPRSS2/ERG fusion gene in prostate cancer.," *Genes. Chromosomes Cancer*, vol. 45, no. 7, pp. 717–9, Jul. 2006.

[28] F. Demichelis, K. Fall, S. Perner, O. Andrén, F. Schmidt, S. R. Setlur, Y. Hoshida, J.-M. Mosquera, Y. Pawitan, C. Lee, H.-O. Adami, L. A. Mucci, P. W. Kantoff, S.-O. Andersson, A. M. Chinnaiyan, J.-E. Johansson, and M. A. Rubin, "TMPRSS2:ERG gene fusion associated with lethal prostate cancer in a watchful waiting cohort.," *Oncogene*, vol. 26, no. 31, pp. 4596–9, Jul. 2007.

[29] P. C. Nowell, "The Clonal Evolution of Tumor Cell Populations," *Science*. pp. 23–28, 1976.

[30] M. Schwab, "Oncogene amplification in solid tumors.," *Semin. Cancer Biol.*, vol. 9, no. 4, pp. 319–25, Aug. 1999.

[31] L. Savelyeva and M. Schwab, "Amplification of oncogenes revisited: from expression profiling to clinical application.," *Cancer Lett.*, vol. 167, no. 2, pp. 115–23, Jun. 2001.

[32] P. J. Stephens, C. D. Greenman, B. Fu, F. Yang, G. R. Bignell, L. J. Mudie, E. D. Pleasance, K. W. Lau, D. Beare, L. a. Stebbings, S. McLaren, M. L. Lin, D. J. McBride, I. Varela, S. Nik-Zainal, C. Leroy, M. Jia, A. Menzies, A. P. Butler, J. W. Teague, M. a. Quail, J. Burton, H. Swerdlow, N. P. Carter, L. a. Morsberger, C. Iacobuzio-Donahue, G. a. Follows, A. R. Green, A. M. Flanagan, M. R. Stratton, P. A. Futreal, and P. J. Campbell, "Massive genomic rearrangement acquired in a single catastrophic event during cancer development," *Cell*, vol. 144, no. 1, pp. 27–40, 2011.

[33] J. V Forment, A. Kaidi, and S. P. Jackson, "Chromothripsis and cancer: causes and consequences of chromosome shattering.," *Nat. Rev. Cancer*, vol. 12, no. 10, pp. 663–70, Oct. 2012.

[34] H. Cai, N. Kumar, H. C. Bagheri, C. von Mering, M. D. Robinson, and M. Baudis, "Chromothripsis-like patterns are recurring but heterogeneously distributed features in a survey of 22,347 cancer genome screens.," *BMC Genomics*, vol. 15, no. 1, p. 82, 2014.

[35] S. C. Baca, D. Prandi, M. S. Lawrence, J. M. Mosquera, A. Romanel, Y. Drier, K. Park, N. Kitabayashi, T. Y. MacDonald, M. Ghandi, E. Van Allen, G. V. Kryukov, A. Sboner, J. P. Theurillat, T. D. Soong, E. Nickerson, D. Auclair, A. Tewari, H. Beltran, R. C. Onofrio, G. Boysen, C. Guiducci, C. E. Barbieri, K. Cibulskis, A. Sivachenko, S. L. Carter, G. Saksena, D. Voet, A. H. Ramos, W. Winckler, M. Cipicchio, K. Ardlie, P. W. Kantoff, M. F. Berger, S. B. Gabriel, T. R. Golub, M. Meyerson, E. S. Lander, O. Elemento, G. Getz, F. Demichelis, M. a. Rubin, and L. a. Garraway, "Punctuated evolution of prostate cancer genomes," *Cell*, vol. 153, no. 3, pp. 666–677, 2013.

[36] S. Hannenhalli and P. a. Pevzner, "Transforming men into mice (polynomial algorithm for genomicdistance problem)," *Proc. IEEE 36th Annu. Found. Comput. Sci.*, 1995.

[37] P. P. Sridhar Hannenhalli, "Transforming Cabbage into Turnip (polynomial algorithm for sorting signed permutations by reversals)," *JACM*, vol. 46, no. 1, pp. 1–27, 1999.

[38] M. D. V Braga, E. Willing, and J. Stoye, "Double cut and join with insertions and deletions.," *J. Comput. Biol.*, vol. 18, no. 9, pp. 1167–1184, 2011.

[39] P. Feijão and J. Meidanis, "SCJ: A breakpoint-like distance that simplifies several rearrangement problems," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 8, no. 5, pp. 1318–1329, 2011.

[40] P. Biller, P. Feijão, and J. Meidanis, "Rearrangement-based phylogeny using the single-cut-or-join operation," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 10, no. 1, pp. 122–134, 2013.

[41] Mitelman F, B. Johansson, and Mertens F (Eds.), “Mitelman Database of Chromosome Aberrations and Gene Fusions in Cancer,” 2016. [Online]. Available: http://cgap.nci.nih.gov/Chromosomes/Mitelman.

[42] M. Ozery-Flato and R. Shamir, “Sorting cancer karyotypes by elementary operations,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5267 LNBI, no. 10, pp. 211–225, 2008.

[43] V. Almendro, Y.-K. Cheng, A. Randles, S. Itzkovitz, A. Marusyk, E. Ametller, X. Gonzalez-Farre, M. Muñoz, H. G. Russnes, A. Helland, I. H. Rye, A.-L. Borresen-Dale, R. Maruyama, A. van Oudenaarden, M. Dowsett, R. L. Jones, J. Reis-Filho, P. Gascon, M. Gönen, F. Michor, and K. Polyak, “Inference of tumor evolution during chemotherapy by computational modeling and in situ analysis of genetic and phenotypic cellular diversity.,” *Cell Rep.*, vol. 6, no. 3, pp. 514–27, Feb. 2014.

[44] E. C. de Bruin, T. B. Taylor, and C. Swanton, “Intra-tumor heterogeneity: lessons from microbial evolution and clinical implications.,” *Genome Med.*, vol. 5, no. 11, p. 101, Jan. 2013.

[45] C. A. Klein, “Selection and adaptation during metastatic cancer progression.,” *Nature*, vol. 501, no. 7467, pp. 365–72, Sep. 2013.

[46] P. L. Bedard, A. R. Hansen, M. J. Ratain, and L. L. Siu, “Tumour heterogeneity in the clinic.,” *Nature*, vol. 501, no. 7467, pp. 355–64, Sep. 2013.

[47] L. Ding, B. J. Raphael, F. Chen, and M. C. Wendl, “Advances for Studying Clonal Evolution in Cancer,” *Cancer Lett.*, vol. 340, no. 2, pp. 212–219, 2013.

[48] A. Albini and M. B. Sporn, “The tumour microenvironment as a target for chemoprevention.,” *Nat. Rev. Cancer*, vol. 7, no. 2, pp. 139–47, Mar. 2007.

[49] A. Mahmoody, C. L. Kahn, and B. J. Raphael, “Reconstructing genome mixtures from partial adjacencies.,” *BMC Bioinformatics*, vol. 13 Suppl 1, no. Suppl 19, p. S9, 2012.

[50] L. Oesper, A. Mahmoody, and B. J. Raphael, “Inferring intra-tumor heterogeneity from high-throughput DNA sequencing data,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7821 LNBI, no. 7, pp. 171–172, 2013.

[51] I. Hajirasouliha, A. Mahmoody, and B. J. Raphael, “A combinatorial approach for analyzing intra-tumor heterogeneity from high-throughput sequencing data,” *Bioinformatics*, vol. 30, no. 12, pp. 78–86, 2014.

[52] M. Greaves and C. C. Maley, “Clonal evolution in cancer.,” *Nature*, vol. 481, no. 7381, pp. 306–13, Jan. 2012.

[53] Y.-F. Guan, G.-R. Li, R.-J. Wang, Y.-T. Yi, L. Yang, D. Jiang, X.-P. Zhang, and Y. Peng, “Application of next-generation sequencing in clinical oncology to advance personalized treatment of cancer.,” *Chin. J. Cancer*, vol. 31, no. 10, pp. 463–70, Oct. 2012.

[54] C. Alkan, B. P. Coe, and E. E. Eichler, “Genome structural variation discovery and genotyping.,” *Nat. Rev. Genet.*, vol. 12, no. 5, pp. 363–376, 2011.

[55] M. Schatz, “Assembly of large genomes using cloud computing,” *Illumina Seq. Panel, Toronto, Canada*, pp. 1165–1173, 2010.

[56] E. Tuzun, A. J. Sharp, J. A. Bailey, R. Kaul, V. A. Morrison, L. M. Pertz, E. Haugen, H. Hayden, D. Albertson, D. Pinkel, M. V Olson, and E. E. Eichler, “Fine-scale structural variation of the human genome.,” *Nat. Genet.*, vol. 37, no. 7, pp. 727–32, Jul. 2005.

[57] J. O. Korbel, A. E. Urban, J. P. Affourtit, B. Godwin, F. Grubert, J. F. Simons, P. M. Kim,

D. Palejev, N. J. Carriero, L. Du, B. E. Taillon, Z. Chen, A. Tanzer, A. C. E. Saunders, J. Chi, F. Yang, N. P. Carter, M. E. Hurles, S. M. Weissman, T. T. Harkins, M. B. Gerstein, M. Egholm, and M. Snyder, "Paired-end mapping reveals extensive structural variation in the human genome.," *Science*, vol. 318, no. 5849, pp. 420–6, Oct. 2007.

[58] J. M. Kidd, G. M. Cooper, W. F. Donahue, H. S. Hayden, N. Sampas, T. Graves, N. Hansen, B. Teague, C. Alkan, F. Antonacci, E. Haugen, T. Zerr, N. A. Yamada, P. Tsang, T. L. Newman, E. Tüzün, Z. Cheng, H. M. Ebling, N. Tusneem, R. David, W. Gillett, K. A. Phelps, M. Weaver, D. Saranga, A. Brand, W. Tao, E. Gustafson, K. McKernan, L. Chen, M. Malig, J. D. Smith, J. M. Korn, S. A. McCarroll, D. A. Altshuler, D. A. Peiffer, M. Dorschner, J. Stamatoyannopoulos, D. Schwartz, D. A. Nickerson, J. C. Mullikin, R. K. Wilson, L. Bruhn, M. V. Olson, R. Kaul, D. R. Smith, and E. E. Eichler, "Mapping and sequencing of structural variation from eight human genomes," *Nature*, vol. 453, no. 7191, pp. 56–64, May 2008.

[59] D. R. Bentley, S. Balasubramanian, H. P. Swerdlow, G. P. Smith, J. Milton, C. G. Brown, K. P. Hall, D. J. Evers, C. L. Barnes, H. R. Bignell, J. M. Boutell, J. Bryant, R. J. Carter, R. Keira Cheetham, A. J. Cox, D. J. Ellis, M. R. Flatbush, N. A. Gormley, S. J. Humphray, L. J. Irving, M. S. Karbelashvili, S. M. Kirk, H. Li, X. Liu, K. S. Maisinger, L. J. Murray, B. Obradovic, T. Ost, M. L. Parkinson, M. R. Pratt, I. M. J. Rasolonjatovo, M. T. Reed, R. Rigatti, C. Rodighiero, M. T. Ross, A. Sabot, S. V Sankar, A. Scally, G. P. Schroth, M. E. Smith, V. P. Smith, A. Spiridou, P. E. Torrance, S. S. Tzonev, E. H. Vermaas, K. Walter, X. Wu, L. Zhang, M. D. Alam, C. Anastasi, I. C. Aniebo, D. M. D. Bailey, I. R. Bancarz, S. Banerjee, S. G. Barbour, P. A. Baybayan, V. A. Benoit, K. F. Benson, C. Bevis, P. J. Black, A. Boodhun, J. S. Brennan, J. A. Bridgham, R. C. Brown, A. A. Brown, D. H. Buermann, A. A. Bundu, J. C. Burrows, N. P. Carter, N. Castillo, M. Chiara E Catenazzi, S. Chang, R. Neil Cooley, N. R. Crake, O. O. Dada, K. D. Diakoumakos, B. Dominguez-Fernandez, D. J. Earnshaw, U. C. Egbujor, D. W. Elmore, S. S. Etchin, M. R. Ewan, M. Fedurco, L. J. Fraser, K. V Fuentes Fajardo, W. Scott Furey, D. George, K. J. Gietzen, C. P. Goddard, G. S. Golda, P. A. Granieri, D. E. Green, D. L. Gustafson, N. F. Hansen, K. Harnish, C. D. Haudenschild, N. I. Heyer, M. M. Hims, J. T. Ho, A. M. Horgan, K. Hoschler, S. Hurwitz, D. V Ivanov, M. Q. Johnson, T. James, T. A. Huw Jones, G.-D. Kang, T. H. Kerelska, A. D. Kersey, I. Khrebtukova, A. P. Kindwall, Z. Kingsbury, P. I. Kokko-Gonzales, A. Kumar, M. A. Laurent, C. T. Lawley, S. E. Lee, X. Lee, A. K. Liao, J. A. Loch, M. Lok, S. Luo, R. M. Mammen, J. W. Martin, P. G. McCauley, P. McNitt, P. Mehta, K. W. Moon, J. W. Mullens, T. Newington, Z. Ning, B. Ling Ng, S. M. Novo, M. J. O'Neill, M. A. Osborne, A. Osnowski, O. Ostadan, L. L. Paraschos, L. Pickering, A. C. Pike, A. C. Pike, D. Chris Pinkard, D. P. Pliskin, J. Podhasky, V. J. Quijano, C. Raczy, V. H. Rae, S. R. Rawlings, A. Chiva Rodriguez, P. M. Roe, J. Rogers, M. C. Rogert Bacigalupo, N. Romanov, A. Romieu, R. K. Roth, N. J. Rourke, S. T. Ruediger, E. Rusman, R. M. Sanches-Kuiper, M. R. Schenker, J. M. Seoane, R. J. Shaw, M. K. Shiver, S. W. Short, N. L. Sizto, J. P. Sluis, M. A. Smith, J. Ernest Sohna Sohna, E. J. Spence, K. Stevens, N. Sutton, L. Szajkowski, C. L. Tregidgo, G. Turcatti, S. Vandevondele, Y. Verhovsky, S. M. Virk, S. Wakelin, G. C. Walcott, J. Wang, G. J. Worsley, J. Yan, L. Yau, M. Zuerlein, J. Rogers, J. C. Mullikin, M. E. Hurles, N. J. McCooke, J. S. West, F. L. Oaks, P. L. Lundberg, D. Klenerman, R. Durbin, and A. J. Smith, "Accurate whole human genome sequencing using reversible terminator chemistry.," *Nature*, vol. 456, no. 7218, pp. 53–9, Nov. 2008.

[60] R. P. Abo, M. Ducar, E. P. Garcia, a. R. Thorner, V. Rojas-Rudilla, L. Lin, L. M. Sholl, W. C. Hahn, M. Meyerson, N. I. Lindeman, P. Van Hummelen, and L. E. MacConaill, "BreaKmer: detection of structural variation in targeted massively parallel sequencing data using kmers," *Nucleic Acids Res.*, vol. 43, no. 19, pp. 1–13, 2014.

[61] A. R. Quinlan, R. a. Clark, S. Sokolova, M. L. Leibowitz, Y. Zhang, M. E. Hurles, J. C.

Mell, and I. M. Hall, “Genome-wide mapping and assembly of structural variant breakpoints in the mouse genome,” *Genome Res.*, vol. 20, no. 5, pp. 623–635, 2010.

[62] K. Chen, J. W. Wallis, M. D. Mclellan, D. E. Larson, J. M. Kalicki, C. S. Pohl, S. D. Mcgrath, M. C. Wendl, Q. Zhang, D. P. Locke, X. Shi, R. S. Fulton, T. J. Ley, R. K. Wilson, L. Ding, and R. Elaine, “BreakDancer: An algorithm for high resolution mapping of genomic structural variation,” vol. 6, no. 9, pp. 677–681, 2013.

[63] J. O. Korbel, A. Abyzov, X. J. Mu, N. Carriero, P. Cayting, Z. Zhang, M. Snyder, and M. B. Gerstein, “PEMer: a computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data.,” *Genome Biol.*, vol. 10, no. 2, p. R23, Jan. 2009.

[64] F. Hormozdiari, C. Alkan, E. E. Eichler, and S. C. Sahinalp, “Combinatorial algorithms for structural variation detection in high-throughput sequenced genomes.,” *Genome Res.*, vol. 19, no. 7, pp. 1270–8, Jul. 2009.

[65] F. Hormozdiari, I. Hajirasouliha, A. McPherson, E. E. Eichler, and S. C. Sahinalp, “Simultaneous structural variation discovery among multiple paired-end sequenced genomes.,” *Genome Res.*, vol. 21, no. 12, pp. 2203–12, Dec. 2011.

[66] F. Hormozdiari, I. Hajirasouliha, P. Dao, F. Hach, D. Yorukoglu, C. Alkan, E. E. Eichler, and S. C. Sahinalp, “Next-generation VariationHunter: combinatorial algorithms for transposon insertion discovery.,” *Bioinformatics*, vol. 26, no. 12, pp. i350–7, Jun. 2010.

[67] L. Oesper, A. Ritz, S. J. Aerni, R. Drebin, and B. J. Raphael, “Reconstructing cancer genomes from paired-end sequencing data.,” *BMC Bioinformatics*, vol. 13 Suppl 6, no. Suppl 6, p. S10, Jan. 2012.

[68] M. Mohiyuddin, J. C. Mu, J. Li, N. Bani Asadi, M. B. Gerstein, A. Abyzov, W. H. Wong, and H. Y. K. Lam, “MetaSV: an accurate and integrative structural-variant caller for next generation sequencing.,” *Bioinformatics*, vol. 31, no. 16, pp. 2741–4, Aug. 2015.

[69] L. T. Fang, P. T. Afshar, A. Chhibber, M. Mohiyuddin, Y. Fan, J. C. Mu, G. Gibeling, S. Barr, N. B. Asadi, M. B. Gerstein, D. C. Koboldt, W. Wang, W. H. Wong, and H. Y. K. Lam, “An ensemble approach to accurately detect somatic mutations using SomaticSeq.,” *Genome Biol.*, vol. 16, no. 1, p. 197, Jan. 2015.

[70] A. D. Ewing, K. E. Houlahan, Y. Hu, K. Ellrott, C. Caloian, T. N. Yamaguchi, J. C. Bare, C. P’ng, D. Waggott, V. Y. Sabelnykova, M. R. Kellen, T. C. Norman, D. Haussler, S. H. Friend, G. Stolovitzky, A. A. Margolin, J. M. Stuart, and P. C. Boutros, “Combining tumor genome simulation with crowdsourcing to benchmark somatic single-nucleotide-variant detection.,” *Nat. Methods*, vol. 12, no. 7, pp. 623–30, Jul. 2015.

[71] A. Kallioniemi, T. Visakorpi, R. Karhu, D. Pinkel, and O. Kallioniemi, “Gene Copy Number Analysis by Fluorescence in Situ Hybridization and Comparative Genomic Hybridization,” *Methods*, vol. 9, no. 1, pp. 113–21, Feb. 1996.

[72] D. Pinkel and D. G. Albertson, “Array comparative genomic hybridization and its applications in cancer.,” *Nat. Genet.*, vol. 37 Suppl, pp. S11–7, Jun. 2005.

[73] A. Kallioniemi, O. P. Kallioniemi, D. Sudar, D. Rutovitz, J. W. Gray, F. Waldman, and D. Pinkel, “Comparative genomic hybridization for molecular cytogenetic analysis of solid tumors.,” *Science*, vol. 258, no. 5083, pp. 818–21, Oct. 1992.

[74] S. du Manoir, M. R. Speicher, S. Joos, E. Schröck, S. Popp, H. Döhner, G. Kovacs, M. Robert-Nicoud, P. Lichter, and T. Cremer, “Detection of complete and partial chromosome gains and losses by comparative genomic in situ hybridization.,” *Hum. Genet.*, vol. 90, no. 6, pp. 590–610, Feb. 1993.

[75] M. Kinsella and V. Bafna, “Combinatorics of the Breakage-Fusion-Bridge Mechanism,” *J. Comput. Biol.*, vol. 19, no. 6, pp. 662–678, 2012.

[76] A. J. Iafrate, L. Feuk, M. N. Rivera, M. L. Listewnik, P. K. Donahoe, Y. Qi, S. W. Scherer, and C. Lee, “Detection of large-scale variation in the human genome.,” *Nat. Genet.*, vol. 36, no. 9, pp. 949–51, Sep. 2004.

[77] M. M. Weiss, M. a Hermsen, G. a Meijer, N. C. van Grieken, J. P. Baak, E. J. Kuipers, and P. J. van Diest, “Comparative genomic hybridisation.,” *Mol. Pathol.*, vol. 52, no. 5, pp. 243–251, 1999.

[78] A. Oostlander, G. Meijer, and B. Ylstra, “Microarray-based comparative genomic hybridization and its applications in human genetics.,” *Clin. Genet.*, vol. 66, no. 6, pp. 488–95, Dec. 2004.

[79] A. E. Urban, J. O. Korbel, R. Selzer, T. Richmond, A. Hacker, G. V Popescu, J. F. Cubells, R. Green, B. S. Emanuel, M. B. Gerstein, S. M. Weissman, and M. Snyder, “High-resolution mapping of DNA copy alterations in human chromosome 22 using high-density tiling oligonucleotide arrays.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 103, no. 12, pp. 4534–4539, 2006.

[80] S. Yoon, Z. Xuan, V. Makarov, K. Ye, and J. Sebat, “Sensitive and accurate detection of copy number variants using read depth of coverage.,” *Genome Res.*, vol. 19, no. 9, pp. 1586–92, Sep. 2009.

[81] A. Abyzov, A. E. Urban, M. Snyder, and M. Gerstein, “CNVnator: an approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing.,” *Genome Res.*, vol. 21, no. 6, pp. 974–84, Jun. 2011.

[82] P. a Pevzner, H. Tang, and M. S. Waterman, “An Eulerian path approach to DNA fragment assembly.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 98, no. 17, pp. 9748–9753, 2001.

[83] P. Pevzner, *Computational Molecular Biology: An Algorithmic Approach*. MIT Press, 2000.

[84] B. J. Raphael, S. Volik, C. Collins, and P. a. Pevzner, “Reconstructing tumor genome architectures,” *Bioinformatics*, vol. 19, no. SUPPL. 2, 2003.

[85] V. Bafna and P. A. Pevzner, “Genome Rearrangements and Sorting by Reversals,” *SIAM J. Comput.*, vol. 25, no. 2, pp. 272–289, Jul. 1994.

[86] C. D. Greenman, E. D. Pleasance, S. Newman, F. Yang, B. Fu, S. Nik-Zainal, D. Jones, K. W. Lau, N. Carter, P. a. W. Edwards, P. A. Futreal, M. R. Stratton, and P. J. Campbell, “Estimation of rearrangement phylogeny for cancer genomes,” *Genome Res.*, vol. 22, no. 2, pp. 346–361, Feb. 2012.

[87] M. Shen, “Chromoplexy: A New Category of Complex Rearrangements in the Cancer Genome,” *Cancer Cell*, vol. 23, no. 5, pp. 567–569, 2013.

[88] P. Williams H., *Model Building in Mathematical Programming,* 4th ed. Wiley, 1999.

[89] J. Bisschop, “Linear Programming Tricks,” in *AIMMS - Optimization Modeling*, AIMMS 3 Ed., Paragon Decision Technology, 2006, pp. 63–75.

[90] R. M. Karp, “Reducibility among combinatorial problems.” p. University of California at Berkeley, 1972.

[91] IBM, “IBM ILOG CPLEX V12.1.” IBM, 2009.

[92] N. Atias and R. Sharan, “Comparative analysis of protein networks,” *Commun. ACM*, vol. 55, no. 5, p. 88, May 2012.

[93] G. Emden, K. Eleftherios, and N. Stephen, “Drawing graphs with dot.” 2006.

[94] E. John, G. Emden, K. Eleftherios, N. Stephen, and W. Gordon, “Graphviz and Dynagraph – Static and Dynamic Graph Drawing Tools,” in *Graph Drawing Software*, M. Jünger and P. Mutzel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 127–148.

תקציר

הגנום של תא סרטני משתנה במהלך התפתחות המחלה ועובר סדרה של שינויים מבניים. שינויים אלו כוללים בין השאר מחיקות, תוספות, התקות (טרנסלוקציות) והיפוכים. התוצאה היא קריוטיפ סרטני מורכב במיוחד וספציפי לחולה. טכנולוגיות ריצוף מהדור החדש ומערכי DNA מאפשרים לחלץ מתוך גנום סרטני פרופיל של מספר עותקים ורשימה של נקודות שבירה ("קפיצות") ברצף הגנטי ביחס לרצף הייחוס הנורמלי. המידע הזה הוא מאוד מפורט אך מקומי מטבעו, ואינו מספק את התמונה הרחבה על מבנה הגנום הסרטני. אחד האתגרים הבסיסיים במחקר של גנום סרטני הוא להשתמש במידע כזה כדי לשחזר את הקריוטיפ הסרטני המלא.

אנחנו מציגים כאן גישה אלגוריתמית, שמתבססת על על תורת הגרפים ועל תכנות לינארי בשלמים, אשר מקבלת כקלט פרופיל מספרי עותקים של מקטעים בגנום ומידע על נקודות שבירה, ומפיקה את הקריוטיפ סרטני בעל התאימות המירבית למידע. השתמשנו בסימולציות כדי להעריך את מידת היישומיות של הגישה שלנו, וכמו כן הפעלנו אותה על מידע אמיתי שנלקח מתוך מאגר הTCGA.

על ידי שימוש במודל סימולציות הצלחנו לתת הערכה למידת הנכונות והחסינות של האלגוריתם שלנו במגוון רחב של תרחישים. תחת התנאים של התרחיש הבסיסי, אשר תוכנן לפי תצפיות שנלקחו מדגימות אמיתיות, האלגוריתם שחזר במלואם 69% מהקריוטיפים. אולם, כשמודדים את ההצלחה לפי מדדים פחות נוקשים המתחשבים במידע מורעש ולא מלא, 87% מהמקרים שנבחנו הראו תוצאה נכונה. יתרה מכך, בתרחישים בהם המידע המסופק לאלגוריתם הוא מאוד נקי ומלא, רמת הדיוק הגיעה ל-100%-90%. ניתוחים של מספר דגימות אמיתיות, כמו גם הפתרון המוצע על ידי האלגוריתם שלנו, מוצגים גם כן.

אונברסיטת תל-אביב
הפקולטה למדעים מדוייקים ע"ש ריימונד ובברלי סאקלר
ביה"ס למדעי המחשב ע"ש בלבטניק

שחזור קריוטיפים סרטניים מתוך נתוני ריצוף עמוק

חיבור זה הוגש כעבודת גמר לקראת התואר "מוסמך אונברסיטה" במדעי המחשב.

על ידי

רמי איתן

עבודת המחקר נעשתה בפקולטה למדעים מדוייקים, באונברסיטת תל-אביב בהנחיית

פרופ' רון שמיר

סיון התשע"ו

אונברסיטת תל-אביב
הפקולטה למדעים מדוייקים ע"ש ריימונד ובברלי סאקלר
ביה"ס למדעי המחשב ע"ש בלבטניק

שחזור קריוטיפים סרטניים מתוך נתוני ריצוף עמוק

חיבור זה הוגש כעבודת גמר לקראת התואר "מוסמך אונברסיטה" במדעי המחשב.

על ידי

רמי איתן

עבודת המחקר נעשתה בפקולטה למדעים מדוייקים, באונברסיטת תל-אביב בהנחיית

פרופ' רון שמיר

סיון התשע"ו