TEL AVIV UNIVERSITY אוניברסיטת תל-אביב

The Raymond and Beverly Sackler Faculty of Exact Sciences
The Blavatnik School of Computer Science

Computational Problems in Genome Rearrangements: from Evolution to Cancer

THESIS SUBMITTED FOR THE DEGREE OF
"DOCTOR OF PHILOSOPHY"

by

Michal Ozery-Flato

The work on this thesis has been carried out
under the supervision of **Prof. Ron Shamir**

Submitted to the Senate of Tel-Aviv University
October 2009

Acknowledgments

The submission of this thesis brings to an end a wonderful period, of almost fifteen years, in which I was a student at Tel Aviv University. On my way to complete this thesis I have experienced many joyous moments, as well as hurdles. I would like to thank those who gave me the strength and courage to continue and press forward.

I am deeply grateful to my advisor, Prof. Ron Shamir, for his guidance, encouragement, criticism, and faith. Ron has been a role model for me, with his broad knowledge, inquisitive mind, uncompromising integrity, and enviable ability to conduct many diverse researches in parallel.

I would like to thank my current and past colleagues at Ron Shamir's lab: Chaim Linhart, Igor Ulitsky, Ofer Lavi, Adi Maron-Katz, Seagull Shavit, Guy Karlebach, Lior Mechlovich, Sharon Bruckner, Dr. Arnon Paz, Dr. Gad Kimmel, Dr. Irit Gat-Viks, Daniela Raijman, Yonit Halperin, Ofir Davidovich, Dr. Rani Elkon, Dr. Firas Swidan, Dr. Falk Hueffner, Dr. Panos Giannopoulos, Dr. Michal Ziv-Ukelson and Israel Steinfeld. Thank you for lending a sympathetic ear and giving useful advice.

Last, but not least, I would like to thank my family. Thank you to my loving parents, Dr. Shoshana and Chaim Ozery, for instilling in me the love of learning and the continuous desire for more knowledge. To Ora and Dubi Flato, my parents in law, thank you for your endless support. To my three sweet children, Yoav, Tamar and Nir, thank you for the happiness you brought into my life and for reminding me the important things in life. Finally, I thank my husband, Eyal, for always being by my side, loving, supporting, and believing in me - this thesis is dedicated to you.

Preface

This thesis is based on the following collection of four articles that were published throughout the PhD period in scientific journals and in refereed proceedings of conferences.

1. **An $O(n^{3/2}\sqrt{\log(n)})$ algorithm for sorting by reciprocal translocations**.
   Michal Ozery-Flato and Ron Shamir.
   Published in *Proceedings of the 17th Annual Symposium on Combinatorial Pattern Matching (CPM'06)* [69] and *Journal of Discrete Algorithms* [77].

2. **Sorting by reciprocal translocations via reversals theory.**
   Michal Ozery-Flato and Ron Shamir.
   Published in *Proceedings of the fourth RECOMB Satellite Workshop on Comparative Genomics (RECOMB-CG'06)* [70] and in *Journal of Computational Biology (JCB)* [73].

3. **Sorting Genomes with Centromeres by Translocations.**
   Michal Ozery-Flato and Ron Shamir.
   Published in *Proceedings of the 11th Annual International Conference on Computational Molecular Biology (RECOMB'07)* [72] and in *Journal of Computational Biology (JCB)* [75].

4. **Sorting Cancer Karyotypes by Elementary Operations.**
   Michal Ozery-Flato and Ron Shamir.
   Published in *Proceedings of the sixth RECOMB Satellite Workshop on Comparative Genomics (RECOMB-CG'08)* [74] and in *Journal of Computational Biology (JCB)* [76].

In addition, this thesis contains the following two articles. The first article was accepted for publication in a refereed proceedings of a conference. The second article

was submitted recently.

1. **On the frequency of genome rearrangement events in cancer karyotypes.**
   Michal Ozery-Flato and Ron Shamir.
   Technical report [71]. Accepted for publication in the *Proceedings of the first RECOMB Satellite Workshop on Computation Cancer Biology (RECOMB-CCB'07)*.

2. **A systematic assessment of associations among chromosomal aberrations in cancer karyotypes.**
   Michal Ozery-Flato, Chaim Linhart, Luba Trakhtenbrot, Shai Izraeli, and Ron Shamir. Submitted.

Abstract

The evolution of species is enabled by the capability of their genomes to mutate. Key events in genome evolution are large scale mutations called *genome rearrangements*, which relocate, duplicate, or delete large DNA segments. Genome rearrangements can result in dramatic phenotypic consequences and are assumed to play an important role in the evolution of species and in cancer. The study of genome rearrangements concentrates on the reconstruction of the history of genome rearrangements between two or more genomes, and on the understanding of contribution of those to the evolutionary process. In this thesis we describe our studies of genome rearrangements. We focus on the fundamental *genomic sorting problem*, which seeks for a shortest sequence of rearrangement events explaining the differences between two related genomes. We present various computational models for genome rearrangements, focusing on translocations events, and develop combinatorial algorithms for solving the genomic sorting problem under these models. In cancer, we apply our algorithms on real data, and perform statistical analyses on the reconstructed rearrangement events. We reveal new characteristics of chromosomal rearrangements in cancer, which may shed light on aberration development mechanisms during carcinogenesis.

Contents

Chapter 1

Introduction

1.1 General Background

The genetic instructions used in the development and functioning of all known living organisms are encoded in their genomes. Genomes are passed from parents to offspring during reproduction, and thus contain all the hereditary information. Genomes are stored in DNA, which in our level of abstraction is a long sequence of four letters, $\{A, C, G, T\}$, called *nucleotides*. The DNA sequence of a genome is partitioned into contiguous subsequences called *chromosomes*. A *gene*, the basic unit of heredity, is a specific sequence of nucleotides that, taken as a whole, specifies a genetic trait. At low resolution, every chromosome can be viewed as a sequence of genes, where each gene has a direction (forward or backward) along its chromosome.

Genomes can evolve in either local or global manner. *Local* alterations refer to point mutations in the DNA sequence, which delete, replace, or insert individual nucleotides. On the other hand, *global* mutations, also known as *genome rearrangements*, relocate, duplicate, or delete large fragments of the DNA. In this thesis we focus on genome rearrangements. Genome rearrangements can result in dramatic phenotypic consequences. On the organismal level, certain rearrangements are associated with mental retardation and birth defects [55]. On the cellular level, specific rearrangements were proved to contribute to cancer formation (see Section 1.4).

One of the ambitious projects of the former century was the determination of the DNA sequence in the human genome. With the advent of sequencing methods, complete genome sequences are now available for a wide range of organisms,

ranging from various bacteria to different mammals. The current major challenge is to decipher the genetic code in those genome sequences. A powerful approach to analyze genome sequences is by their comparison. By examining the differences and similarities between genomes, we can learn about the way these genomes evolved. The conserved fractions in genomes of related species, such as human and mouse, are associated with common similar functions and are likely to be inherited from their most recent common ancestor. The differences between the genomes are explained by lineage-specific events occurring after the divergence of the corresponding species.

1.2 Genome Rearrangements in Evolution

Genomes of related species are very similar. For example, over 90% of the mouse and human genomes can be partitioned into corresponding regions in which gene content and order is conserved [112] (see Fig. 1.1). The difference in the ordering of these blocks along the human and mouse genomes is attributed to rearrangement events occurring after the divergence of the two lineages. As human and mouse are believed to have diverged more than 65 millions years ago [112], the number of conserved blocks in their genomes implies that the rate of genome rearrangement events in these lineages is relatively low: few events per million years. This makes the inference of the rearrangement events between human and mouse a potentially tractable problem.

The phenomenon of genome rearrangements in evolution was discovered more than 80 years ago. In the 1930's, Sturtevant and Dobzhansky [98] demonstrated inversions between genomes of various drosophila species. In the late 1980's Jeffrey Palmer and colleagues discovered that mitochondrial genomes of related plants have essentially the same gene content but different gene ordering [78, 79, 80, 81, 45]. This discovery suggested that the evolution of these plants was driven by genome rearrangement events. Advances in molecular cytogenetics, mainly comparative chromosome painting ("Zoo-FISH" [92]), led to the generation of large-scale comparative genome maps of more than 80 mammalian species [38]. The development of bioinformatic methods for locating homolog blocks in different genome sequences [82, 30, 102], enabled the creation of finer comparative maps based on genomes sequences.

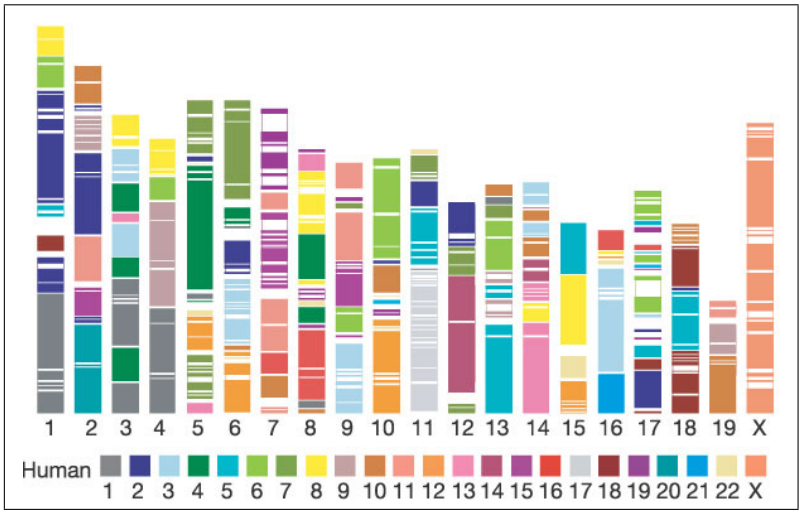


Figure 1.1: The mouse genome is comprised of regions with conserved synteny in human. Taken from [112]. Each color corresponds to a particular human chromosome.

1.3 The Genomic Sorting Problem

The computational study of genome rearrangements during species evolution was pioneered by Sankoff [90, 91, 88]. This line of research builds on the assumption that evolution is parsimonious and prefers a shortest path of events. A well studied problem is *genomic sorting*, which seeks for a shortest sequence of rearrangement events between two related genomes. The length of such shortest sequence is the *rearrangement distance* between these genomes. Genomic sorting gives rise to a spectrum of fascinating algorithmic and combinatorial problems, each defined by the representation of the genomes and the set of allowed rearrangements operations. For a review of the computational study of various genomic sorting problems see [18].

In the model we consider, a genome is a collection of chromosomes, where each chromosome is represented as a sequence of genes. A gene is identified by an (unsigned) integer. When it appears in a chromosome, a gene is associated with a sign, plus or minus, representing the direction of the gene along its chromosome. If A is a genome with N chromosomes, and the k-th chromosome in A contains n_k genes, then

$$A = \{(g_{11}, g_{12}, \cdots, g_{1n_1}), (g_{21}, g_{22}, \cdots, g_{2n_2}), \ldots, (g_{N1}, g_{N2}, \cdots, g_{Nn_N})\}$$

A *reversal* of a sequence of genes is the operation of reversing the order of the genes in the sequence and flipping their signs. For example, the reversal of $S = (g_1, g_2, \ldots, g_n)$ is $-S = (-g_n, -g_{n-1}, \ldots, -g_1)$. A reversal on an entire chromosome is called a *chromosome flip*. As chromosomes have no direction, a flip of a chromosome does not affect the chromosome it represents and is usually used to move between the two possible equivalent representations of a chromosome.

Two prominent rearrangement events are inversions and translocations, which are believed to be most common in mammals. An *inversion* is a reversal of a segment of genes in a chromosome. The following example describes an inversion on the underlined segment of genes:

$$S_1, \underline{S_2}, S_3 \longrightarrow S_1, \underline{-S_2}, S_3.$$

Inversions are commonly referred to as "reversals" in the computational research of genome rearrangements, as we shall do for the rest of this thesis.

Translocations exchange the ends of two chromosomes as described below. Consider the following two chromosomes:

$$(X_1, X_2), (Y_1, Y_2).$$

A *prefix-prefix* translocation on the two chromosomes above results in:

$$(X_1, Y_2), (Y_1, X_2).$$

Alternatively, a *prefix-suffix* translocation on these chromosomes results in:

$$(X_1, -Y_1), (-X_2, Y_2).$$

A translocation is *reciprocal* if the involved segments (i.e. X_1, X_2, Y_1, and Y_2) are all non-empty. In the following, unless specified otherwise, we consider only reciprocal translocations.

Sorting by reversals (SBR) and *sorting by translocations* (SBT) are two instances of the genomic sorting problem confined to one type of rearrangement events, either reversals (SBR), or translocations (SBT). While SBT is defined for multi-chromosomal genomes, SBR is defined for only uni-chromosomal genomes. The input genomes to SBR and SBT, say A and B, are required to satisfy the following two requirements:

1. A and B have identical gene content (i.e. no loss/gain)

2. Every gene in A (respectively, B) is unique.

While the first requirement follows from the fact that both reversals and translocations do not alter gene content, the latter requirement was made to simplify the computational analysis. In fact, when duplicate genes are allowed, SBR was proved to be NP-hard [84, 28].

Following the requirements above, a uni-chromosomal genome is represented by a *signed permutation*, which is a permutation on the integers $\{1, \ldots, n\}$, where a sign of plus or minus is assigned to each number. The following is an example of a signed permutations with eight elements:

$$(1, -3, -2, 4, -7, 8, 6, 5)$$

A special signed permutation is $(1, 2, \ldots, n)$, which we shall refer to as the *identity permutation*. Multi-chromosomal genomes are presented by *fragmented* signed permutation, where each fragment corresponds to a chromosome. Here is an example of a genome with eight genes partitioned into two chromosomes:

$$\{(1, -3, -2, 4, -7, 8), (6, 5)\}$$

A concatenation of the chromosomes in a multi-chromosomal genome thus results in a signed permutation. Given the input genomes, A and B, we can assume for simplicity and without loss of generality that genome B is the identity permutation, in case of SBR, or a fragmented identity permutation, in case of SBT. The transformation of the "permutated" genome A into the "organized" genome B is thus viewed as a sorting process.

1.3.1 Sorting by Reversals

SBR was intensively studied in the past two decades. Kececioglu and Sankoff formulated SBR and gave the first constant factor approximation algorithm for this problem [51]. The problem was further studied by Bafna and Pevzner [9] who introduced the notion of *cycle graph* (aka *breakpoint graph*) of a signed permutation and revealed important links between the cycle decomposition of this graph and the reversal distance. The cycle graph of a permutation became the foundation of subsequent analyses of SBR. The major breakthrough in the study of SBR was made

by Hannenhalli and Pevzner [41] who proved that the problem is polynomial. In [15], Berman and Hannenhalli presented a recursive algorithm for SBR that can be implemented in $O(n^2\alpha(n))$ time, where $\alpha(n)$ is the inverse of the Ackerman's function [2]. The analysis of SBR was greatly simplified by Kaplan, Shamir, and Tarjan [49] who introduced the notion of *overlap graph* of a signed permutation. Bergeron [11] further simplified the analysis by presenting a simple score-based $O(n^3)$-time algorithm using the overlap graph. An elegant algorithm was given by Tannier and Sagot [104, 103], which has a relatively simple implementation in $O(n^2)$. Using a clever data structure by Kaplan and Verbin [50], the algorithm of Tannier and Sagot was shown to have $O(n^{3/2}\sqrt{\log(n)})$ implementation [104, 103]. Very recently, Swenson et al. [101] modified the data structure of Kaplan and Verbin, and presented a new algorithm, which based on experimental results, runs in $O(n\log(n))$ on most signed permutations. The reversal distance of a signed permutation π is computed in linear time by an algorithm of Bader, Moret, and Yan [7]. Using this algorithm, the recursive algorithm in [15] can be implemented in $O(n^2)$.

1.3.2 Sorting by Translocations

SBT was introduced by Kececioglu and Ravi [52] who gave a 2-approximation algorithm for its solution. Hannenhalli extended the notion of *cycle graph* for multichromosomal genomes, and showed that SBT is polynomial [39]. Bergeron, Mixtacki and Stoye [14] pointed to an error in Hannenhalli's algorithm and presented an alternative modified $O(n^3)$ algorithm. The translocation distance can be computed in linear time, in a similar manner to the computation of the reversal distance [14]. Li et al. [56] gave a linear time algorithm for computing the translocation distance (without producing a shortest sequence). Wang et al. [111] presented an $O(n^2)$ algorithm for solving SBT. However, the algorithms in [56, 111] rely on an erroneous theorem in [39] and hence provide incorrect results in certain cases.

A genomic sorting problem that integrates both reversals and translocations was first studied by Kececioglu and Ravi [52]. In this problem, which we will refer as SBRT, translocations are allowed to be non-reciprocal, and chromosome fissions and fusions are also allowed. SBRT was proved be polynomial by Hannenhalli and Pevzner [40], by reducing it to SBR. In particular, it was shown that a translocation can be mimicked by a reversal on a concatenation of the chromosomes. The theory and algorithm for SBRT were later corrected and revised by Tesler [105], Ozery-Flato

and Shamir [68], and Jean and Nikolski [46].

1.3.3 Integrating the Centromeres

Every chromosome contains a special region called *centromere*, which is essential to the segregation of the duplicated chromosomes during cell division. An *acentric* chromosome, i.e., a chromosome that lacks a centromere, is likely to be lost during subsequent cell divisions [99]. Therefore, a rearrangement scenario that preserves a centromere in each chromosome is more biologically probable than one that does not. Previous computational studies on genome rearrangements have ignored the existence and role of centromeres, and thus may produce rearrangement scenarios involving many acentric chromosomes. Due to their highly repetitive content, current sequencing methods cannot be applied to centromeres. Therefore, we have no information about centromere sequences, nor do we have homolog mapping between centromeres in related genomes. For every centromere, we only know its location within its chromosome.

1.4 Chromosome Instability in Cancer

Carcinogenesis, the transformation of normal cells into cancer cells, can be viewed as an evolutionary process in which a normal genome accumulates mutations that eventually transform it into a cancerous one. Cancer is associated with *chromosome instability*, as most cancer cells show chromosomal abnormalities caused by genome rearrangements. Acquired chromosome abnormalities were first suggested to be factors in the origin of cancer by Boveri in 1914 [21]. It remained an attractive hypothesis until the discovery of the *Philadelphia chromosome*, an abnormal chromosome that exists in 95% of the people with chronic myelogenous leukemia (CML). The Philadelphia chromosome was discovered in 1960 by Nowell and Hungerford [67] who named it after the city in which both labs were located. In 1973, Rowley identified the mechanism by which the Philadelphia chromosome arises as a reciprocal translocation between chromosome 9 and 22 [87]. The result of this translocation is the fusion gene BCR-ABL, composed of the BCR gene from chromosome 22 and the ABL gene from chromosome 9 [31]. This gene was shown to contribute to the development of CML, thus becoming a potential target for developing a new drug

for CML. In the late 1990s the drug imatinib (aka Gleevec/Glivec) was identified as an inhibitor for BCR-ABL [34], and in 2001 it was approved for treating CML patients in the United States.

1.4.1 Chromosomal Aberrations

Chromosomal aberrations are disruptions in the normal chromosomal content, commonly classified as either numerical or structural. *Numerical* aberrations refer to an abnormal copy number of specific chromosomes. This phenomenon, called *chromosomal aneuploidy*, is caused by chromosome missegregation during cell division, leading to the loss, or gain, of particular chromosomes [113]. *Structural* aberrations refer to the existence of chromosomes with abnormal structure. In somatic cells, and cancer cells in particular, structural aberrations are commonly associated with mis-repair of double strand breaks (DSBs) in the DNA. DSBs are promoted by extrinsic (e.g., radiation, chemicals) and intrinsic (e.g., reactive oxygen, stalling of DNA replication forks) sources. They are estimated to be quite common with several DSBs per cell cycle [3]. To preserve genomic integrity, elaborate systems for DNA repair have evolved. As broken chromosome ends appear to be adhesive and tend to fuse with some other broken ends, a failure in the repair of DSBs may result in chromosomal rearrangements, including translocations, deletions, and duplications [53, 3]. Such rearrangement events can lead to carcinogenesis if, for example, a deleted chromosomal region encodes a tumor suppressor gene, or if an amplified region encodes an oncogene. Translocations can lead to the formation of new gene products, such as the BCR-ABL gene in CML, or to the dysregulation of specific genes caused by the swapping of promoter elements, such as the case of the oncogene C-MYC in certain lymphomas [29].

1.4.2 Cancer Karyotypes

The classic laboratory methods for detecting chromosomal rearrangements use painting techniques on chromosomes undergoing mitosis. In the resulting visualized genome each chromosome is partitioned into continuous genomic regions called *bands*, where each band usually spans 5-10 millions of nucleotides (see Fig. 1.2(a)). Therefore only large rearrangements are detected with these techniques. A *karyotype* is a description of the visualized genome in banding resolution. The accuracy of kary-

otypes can be enhanced by integrating the more modern techniques of FISH and SKY / M-FISH. *FISH* (Fluorescence In Situ Hybridization) [83] is a technique that uses fluorescent tags to locate the position of a specific DNA sequence along the chromosome. *SKY* (Spectral Karyotyping, [93]) and *M-FISH* (Multiplex Fluorescence In Situ Hybridization, [97]) are molecular cytogenetic techniques that permit the simultaneous visualization of all the chromosomes in different colors (see Fig. 1.2(b)). SKY / M-FISH considerably simplify the detection of material exchange between chromosomes, such as translocations, but cannot detect rearrangements internal to chromosomes, such as inversions.

Karyotyping have become an increasingly important tool in the management of cancer patients, helping to establish a correct diagnosis, select the appropriate treatment and predict outcome [63]. The largest available depository of cancer karyotypes is the Mitelman database of chromosomal aberrations in cancer [62], which records cancer karyotypes reported in the scientific literature. Currently, this database contains almost 60,000 cancer karyotypes, most of which (70%) are from hematological disorders. This bias toward hematological disorders, which consist less than 10% of cancer cases, are due to technical difficulties in getting karyotypes of solid tumors. Array-based comparative genomic hybridization (array-CGH) [96] is a modern laboratory technique that can provide information on copy number aberrations (i.e. gain / loss) at high resolution. Alas, array-CGH is incapable of detecting structural rearrangement such as translocations. Moreover, the number of currently available cases analyzed by array-CGH and other novel techniques is one or more orders of magnitudes smaller than the number of cancer karyotypes in the Mitelman database.

End Sequence Profiling (ESP) [108] is a laboratory technique that provides high resolution data on structural aberrations as follows. First, the tumor genome is split into small (100-300 kb), overlapping pieces (clones). Second, both ends ($\sim$ 500bp each) of each clone are sequenced. Third, the resulting end sequences are mapped to the human genome sequence. Each clone whose end sequences map uniquely to the human genome yields a pair (x, y) of locations in the human genome corresponding to the mapped ends. A pair of locations that are too far to fit a contiguous genomic segment in the healthy genome indicates a rearrangement. Currently, ESP data exist for only few cancer samples [108, 107, 17]. In future, with the advent of next generation sequencing techniques (see [94, 6] for reviews), more ESP data, and even whole sequence data, are expected to become available for cancer genomes.

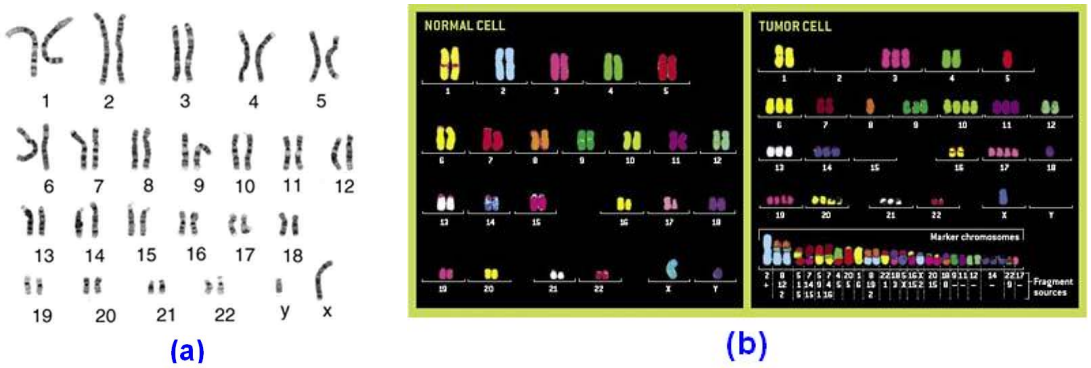


Figure 1.2: Visualization of genomes using cytogenetic techniques. (a) Classical chromosome painting (G-banding) of a normal male genome. Taken from [1]. (b) Spectral Karyotyping (SKY) of a normal male genome (left) and of an abnormal breast cancer genome (right). Taken from [35].

The karyotypes in the Mitleman database are described using the ISCN nomenclature [61], and thus can be parsed automatically. In our analyses of cancer karyotypes we used the CyDAS ISCN parser [42]. An ISCN description reports on the chromosomal aberrations observed in a sample, where a sample consists of several cells. Each aberration reported in a karyotype must be present in at least two cells in the described sample. In some cases, the cell population may be non-homogeneous, and contain cells with several distinct aberrations, resulting from the existence of different cell lineages in the evolution of the cancer. A homogeneous cell sample is described by a *simple* karyotype, while a non-homogeneous one has a *complex* karyotype, which consists of several simple karyotypes. Karyotypes may contain missing information (denoted by '?'), in case the observed aberration could not be determined. When there is no such missing information, we refer to a karyotype as *well-characterized*.

1.4.3 Genome rearrangements with duplications

The model that assumes for reversals and translocations as the only allowed rearrangements was commonly used to analyze the different gene/synteny block orderings between species (e.g. [82, 20, 65]). Is this model adequate for analyzing rearrangements in cancer genomes? The answer is probably negative, as this model does not allow for deletion or duplication events. Moreover, while in evolutionary

studies the haploid genome is considered (i.e. one representative from every pair of homologous chromosomes), in cancer studies we need to consider the diploid genome (i.e. all chromosomes), as every chromosome is free to gain its own mutations. In other words, when analyzing the evolution of a normal genome into a cancer genome, we need to consider to two copies of each chromosome. In the past decade there have been many computational studies of genome rearrangements with duplicate genes and / or duplication events. Below we briefly review some of the studies that are more pertinent to our study.

Allowing for duplicate genes and/or duplication events makes the genomic sorting problem much more difficult. For instance, the problem of sorting sequences by reversals was shown to be NP-hard [84, 28]. Thus, most current approaches for duplication analysis rely on heuristics, approximation algorithms, or restricted models of duplication. A heuristic for the sorting sequences by reversals was given in [28]. Some studies focused on the problem of finding a matching between duplicated genes in two compared genomes, based on their orderings. Sankoff [89] was the first to test this idea with the *exemplar approach* that selects a single gene, called *exemplar*, from each gene family (i.e. a set of identical genes in a genome), and discards the remaining duplicate genes. Given a pair of genomes, the exemplars are selected so as to minimize the rearrangement distance between the two reduced genomes. The problem of identifying optimal exemplars was proved to be NP-hard for the reversal distance, even when one genome contains no duplicate genes [25]. A divide-and-conquer approach to compute an exemplar-based distance between two genomes was given in [66].

Marron et al. [58] presented an approximation algorithm for computing a shortest sequence of reversals, deletions, duplications, and insertions between an arbitrary genome and the identity permutation. Although their algorithm has a large error-bound, it was suggested to compute near-minimal solutions based on experimental results. Later on, Swenson et al. [100] generalized the algorithm in [58] to work on two arbitrary genomes. The problem of *genome halving*, which seeks for a shortest sequence of non-duplicating rearrangements resulting in a perfectly doubled genome (i.e. a genome after whole-duplication event), was shown to have an exact polynomial solution under different rearrangement models [36, 4, 64]. Models considering tandem duplications were also studied in [27, 8].Finally, a model for segmental duplications in the evolution of mammalian genomes was introduced and studied by Kahn et al. [48, 47]. Under this model a duplication event copies a substring from

a fixed source string into an arbitrary location in a target string.

The integration of duplications into rearrangement models poses a major computational challenge. Therefore, many of the studies we reviewed above consider restricted models for duplications and most of them rely on various heuristics. Finally, all (duplications-aware) rearrangement models in the works cited above were designed for analyzing the genomes in the light of evolution. Following the traditional HP model, most of these models consider reversals as their main, sometimes only, reordering event. To the best of our knowledge, none of these algorithms was used to analyze cancer genomes, and cancer karyotypes in particular.

1.4.4 Associations among Chromosomal Aberrations

Cancer karyotypes exhibit a wide variety of chromosomal aberrations. For some cancers, mainly hematological disorders and sarcomas, certain abnormalities are highly specific or strongly associated with particular diagnostic entities. Typically, these abnormalities are reciprocal translocations, such as the Philadelphia translocation mentioned above. For most cancers, notably epithelial tumors, the observed aberrations appear more sporadically and hence it is more difficult to prove their significance to carcinogenesis process. Thus, for the majority of observed aberrations their importance to the formation and progress of cancer is yet to be determined.

Inspired by the four-step model for colorectal cancer evolution, suggested by Vogelstein et al. [106], many extant computational studies have focused on the inference of primary pathways in which chromosomal aberrations are accumulated in certain cancer types. Some of these methods used tree models [32, 33, 109], later extended to acyclic networks [85, 44, 43]. These evolutionary models allow the recognition of aberrations occurring at early stages of cancer. Such aberrations, often referred to as "primary", are suspected to contribute to the formation of cancer. More recently, a statistical method named GISTIC [16] was developed for identifying copy-number aberrations whose frequency and amplitude are higher than expected. As all the methods described above were designed to analyze samples from the same cancer type, they were applied to relatively small datasets, each containing a few hundred samples.

1.5 Summary of Articles Included in this Thesis

1. **An $O(n^{3/2}\sqrt{\log(n)})$ algorithm for sorting by reciprocal translocations**.
   Michal Ozery-Flato and Ron Shamir.
   Published in *Proceedings of the 17th Annual Symposium on Combinatorial Pattern Matching (CPM'06)* [69] and *Journal of Discrete Algorithms* [77].

   In this paper we proved that sorting by reciprocal translocations can be done in $O(n^{3/2}\sqrt{\log(n)})$ for a genome with n genes. Our algorithm was an adaptation of the algorithm of Tannier, Bergeron and Sagot for sorting by reversals. This improved over the $O(n^3)$ algorithm for sorting by reciprocal translocations given by Bergeron, Mixtacki and Stoye.

2. **Sorting by reciprocal translocations via reversals theory.**
   Michal Ozery-Flato and Ron Shamir.
   Published in *Proceedings of the fourth RECOMB Satellite Workshop on Comparative Genomics (RECOMB-CG'06)* [70] and in *Journal of Computational Biology (JCB)* [73].

   In this paper we focused on sorting a multichromosomal genome by translocations. We revealed new relationships between this problem and the well studied problem of sorting by reversals. Based on these relationships, we developed two new algorithms for sorting by reciprocal translocations, which mimicked known algorithms for sorting by reversals: a score-based method building on Bergeron's algorithm, and a recursive procedure similar to the Berman-Hannenhalli method. Though their proofs were more involved, our procedures for reciprocal translocations matched the complexities of the original ones for reversals only.

3. **Sorting Genomes with Centromeres by Translocations.**
   Michal Ozery-Flato and Ron Shamir.
   Published in *Proceedings of the 11th Annual International Conference on Computational Molecular Biology (RECOMB'07)* [72] and in *Journal of Computational Biology (JCB)* [75].

   In this paper, we studied for the first time centromere-aware genome rearrangements. We presented a polynomial time algorithm for computing a shortest sequence of translocations transforming one genome into the other, where all

of the intermediate chromosomes must contain centromeres. We viewed this as a first step towards analysis of more general genome rearrangement models that take centromeres into consideration.

4. **Sorting Cancer Karyotypes by Elementary Operations.**
   Michal Ozery-Flato and Ron Shamir.
   Published in *Proceedings of the sixth RECOMB Satellite Workshop on Comparative Genomics* [74] and in *Journal of Computational Biology (JCB)* [76].

   In this study, we proposed a mathematical framework for analyzing chromosomal aberrations in cancer karyotypes. We introduced the problem of sorting karyotypes by elementary operations, which seeks a shortest sequence of elementary chromosomal events transforming a normal karyotype into a given (abnormal) cancerous karyotype. Under certain assumptions, we proved a lower bound for the elementary distance, and presented a polynomial-time 3-approximation algorithm for the problem. We applied our algorithm to karyotypes from the Mitelman database, which records cancer karyotypes reported in the scientific literature. Approximately 94% of the karyotypes in the database, totaling 58,464 karyotypes, supported our assumptions, and each of them was subjected to our algorithm. Remarkably, even though the algorithm is only guaranteed to generate a 3-approximation, it produced a sequence whose length matched the lower bound (and hence optimal) in 99.9% of the tested karyotypes.

5. **On the frequency of genome rearrangement events in cancer karyotypes.**
   Michal Ozery-Flato and Ron Shamir.
   Technical report [71]. Accepted for presentation in *the first RECOMB Satellite Workshop on Computation Cancer Biology (RECOMB-CCB'07)* (peer-reviewed, but with no proceedings).

   In this study we introduced a new approach for analyzing rearrangement events in carcinogenesis. This approach built on a new effective heuristic for computing a short sequence of rearrangement events that may have led to a given karyotype. We applied this heuristic to over 40,000 karyotypes reported in the scientific literature. Our analysis implied that these karyotypes had evolved predominantly via four principal event types: chromosomes gains and losses, reciprocal translocations, and terminal deletions. We used the frequencies of

the reconstructed rearrangement events to measure similarity between karyotypes. Using clustering techniques, we demonstrated that in many cases, rearrangement event frequencies are an effective means for distinguishing between karyotypes of distinct tumor classes.

6. **A systematic assessment of associations among chromosomal aberrations in cancer karyotypes.**
Michal Ozery-Flato, Chaim Linhart, Luba Trakhtenbrot, Shai Izraeli, and Ron Shamir. Submitted.

In this paper we reported on a systematic study and a database on the characteristics of chromosomal aberrations in cancers, using the largest available repository of reported karyotypes. Our method was used to analyze chromosomal aberrations derived from over 15,000 cancer karyotypes in the Mitelman database. We compared cancer types by their manifested aberrations, computed scores for their similarity, and used these scores to draw an aberration-similarity map of cancers. This map was highly concordant with the histological classification of cancers. In addition, we revealed some novel similarities between cancers, e.g. among three embryonic tumors: Wilms' tumor, Hepatobalstoma, and Ewing's sarcoma. In another analysis we revealed a large number of significantly co-occurring aberrations, i.e., aberrations that tend to appear together, which mostly involve chromosome aneuploidy (numerical aberrations). Interestingly, the co-occurring aberrations were primarily confined to one of two aberration classes: either two chromosome gains or two chromosome losses, suggesting two separate progression paths for aneuploidy in cancer. Our results assigned solid statistical foundations to many findings reported in the literature, and also revealed novel findings that merit further research. An accompanying database, called STACK (STatistical Associations in Cancer Karyotypes), summarized all associations that were discovered and allows easy search, filtering and sifting of the results, as well as direct viewing of the relevant karyotypes in the Mitelman database.

Chapter 2

An $O(n^{3/2}\sqrt{\log(n)})$ Algorithm for Sorting by Reciprocal Translocations

An $O(n^{3/2}\sqrt{\log(n)})$ algorithm for sorting by reciprocal translocations

Michal Ozery-Flato [a] Ron Shamir [a]

[a] *The Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel*

Abstract

We prove that sorting by reciprocal translocations can be done in $O(n^{3/2}\sqrt{\log(n)})$ for an n-gene genome. Our algorithm is an adaptation of the algorithm of Tannier, Bergeron and Sagot for sorting by reversals. This improves over the $O(n^3)$ algorithm for sorting by reciprocal translocations given by Bergeron, Mixtacki and Stoye.

Key words: translocations; reversals; genome rearrangements

1 Introduction

In this paper we study the problem of sorting by reciprocal translocations (abbreviated SRT). *Reciprocal translocations* exchange non-empty ends between two chromosomes. Given two multi-chromosomal genomes A and B, the problem of SRT is to find a shortest sequence of reciprocal translocations that transforms A into B. SRT was first introduced by Kececioglu and Ravi [11] and was given a polynomial time algorithm by Hannenhalli [6]. Bergeron, Mixtacki and Stoye [4] pointed to an error in Hannenhalli's proof of the reciprocal translocation distance formula and consequently in Hannenhalli's algorithm. They presented a new $O(n^3)$ algorithm, which to the best of our knowledge, is the only extant correct algorithm for SRT [1].

Reversals (or inversions) reverse the order and the direction of transcription of the genes in a segment inside a chromosome. Given two uni-chromosomal genomes π_1 and π_2, the problem of sorting by reversals (abbreviated SBR)

[1] Li et al. [12] gave a linear time algorithm for computing the reciprocal translocation distance (without producing a shortest sequence). Wang et al. [16] presented an $O(n^2)$ algorithm for SRT. However, the algorithms in [12, 16] rely on an erroneous theorem of Hannenhali and hence provide incorrect results in certain cases.

is to find a shortest sequence of reversals that transforms π_1 into π_2. This problem has been intensively studied [8, 5, 9, 1, 2, 15]. Tannier, Bergeron and Sagot [15] presented an elegant algorithm for SBR that can be implemented in $O(n^{3/2}\sqrt{log(n)})$ using a clever data structure by Kaplan and Verbin [10]. This is currently the fastest algorithm for SBR.

In this paper we prove that SRT can be solved in $O(n^{3/2}\sqrt{\log(n)})$ for an n-gene genome. Our algorithm for SRT is similar to the algorithm by Tannier, Bergeron and Sagot [15] for SBR. The key idea is to recast translocations as reversals, and then exploit the novel theoretical improvements in SBR theory to obtain faster SRT algorithms. (It should be noted that Hanenhalli and Pevzner have already established and exploited the basic connection between translocations and reversals, in the context of sorting a genome by reversals and translocations [7]). Our approach builds on generalizing the overlap graph. Most studies of SBR to date relied explicitly or implicitly on the combinatorial structure of the overlap graph for representing the relations between two permutations. Since translocations involve multiple chromosomes, we generalize the notion of (uni-chromosomal) overlap graph to include chromosomal information, and show that the same conceptual algorithmic framework developed for SBR applies to SRT, via this generalized overlap graph. While our final algorithm is very similar to that of Tannier et al., the proofs had to be completely redone. Another contribution of this study is in showing that the general SRT problem can be reduced in linear time to a special case, and thus time complexity analysis can be done for such special cases only.

The paper is organized as follows. The necessary preliminaries are given in Section 2. In Section 3 we give a linear time reduction from SRT to a simpler restricted subproblem. In Section 4 we prove the main theorem and present the algorithm for the restricted subproblem. In Section 5 we describe an $O(n^{3/2}\sqrt{\log(n)})$ implementation of the algorithm. A preliminary version of this study was published in the proceedings of CPM 2006 [13].

2 Preliminaries

This section provides a basic background for the analysis of SRT. It follows to a large extent the nomenclature and notation of [6, 9, 4]. In the model we consider, a *genome* is a set of chromosomes. A *chromosome* is a sequence of genes. A *gene* is identified by a positive integer. All genes in the genome are distinct. When it appears in a genome, a gene is assigned a sign of plus or minus. For example, the following genome consists of 8 genes in two chromosomes:

$$A_1 = \{(1, -3, -2, 4, -7, 8), (6, 5)\}$$

The *reverse* of a sequence of genes $I = (x_1, \ldots, x_l)$ is $-I = (-x_l, \ldots, -x_1)$. A *reversal* reverses a segment of genes inside a chromosome. Two chromosomes, X and Y, are *identical* if either $X = Y$ or $X = -Y$. Therefore, *flipping* chromosome X into $-X$ does not affect the chromosome it represents. For example, the following are two equivalent representations of the same genome

$$\{\underline{(1, -3, -2, 4, -7, 8)}, (6, 5)\} \equiv \{\underline{(-8, 7, -4, 2, 3, -1)}, (6, 5)\}$$

Let $X = (X_1, X_2)$ and $Y = (Y_1, Y_2)$ be two chromosomes, where X_1, X_2, Y_1, Y_2 are sequences of genes. A *translocation* cuts X into X_1 and X_2 and Y into Y_1 and Y_2 and exchanges segments between the chromosomes. It is called *reciprocal* if X_1,X_2, Y_1 and Y_2 are all non-empty. There are two ways to perform a translocation on X and Y. A *prefix-suffix* translocation switches X_1 with Y_2 resulting in:

$$(\underline{X_1}, X_2), (Y_1, \underline{Y_2}) \Rightarrow (\underline{-Y_2}, X_2), (Y_1, \underline{-X_1})$$

A *prefix-prefix* translocation switches X_1 with Y_1 resulting in:

$$(\underline{X_1}, X_2), (\underline{Y_1}, Y_2) \Rightarrow (\underline{Y_1}, X_2), (\underline{X_1}, Y_2)$$

The following is an example of prefix-prefix and prefix-suffix translocations that cut the genome in the same place:

$$\{(\underline{1, -3, -2, 4}, -7, 8), (\underline{6}, 5)\} \Rightarrow \{(\underline{6}, -7, 8), (\underline{1, -3, -2, 4}, 5)\}$$

$$\{(\underline{1, -3, -2, 4}, -7, 8), (6, \underline{5})\} \Rightarrow \{(\underline{-5}, -7, 8), (6, \underline{-4, 2, 3, -1})\}$$

Recall that chromosome flips do not affect the genome, but rather move between different representations of the same genome. Thus we can mimic one type of translocation by a flip of one of the chromosomes followed by a translocation of the other type.

For a chromosome $X = (x_1, \ldots, x_k)$ define $\mathit{Tails}(X) = \{x_1, -x_k\}$. Note that flipping X does not change $\mathit{Tails}(X)$. For a genome A define $\mathit{Tails}(A) = \bigcup_{X \in A} \mathit{Tails}(X)$. For example:

$$\mathit{Tails}(A_1) = \mathit{Tails}(\{(1, -3, -2, 4, -7, 8), (6, 5)\}) = \{1, -8, 6, -5\}.$$

Two genomes A' and A'' are *co-tailed* if $\mathit{Tails}(A') = \mathit{Tails}(A'')$. In particular, two co-tailed genomes have the same number of chromosomes (recall that all genes in a genome are unique). Note that if A'' was obtained from A' by performing a reciprocal translocation then $\mathit{Tails}(A'') = \mathit{Tails}(A')$. Therefore, SRT is defined only for genomes that are co-tailed. For the rest of this paper the word "translocation" refers to a reciprocal translocation and we assume that the given genomes, A and B, are co-tailed.

2.1 The Cycle Graph

In this section we present the cycle graph of genomes A and B, which was first defined in [6]. Let N be the number of chromosomes in A (equivalently, B). We shall always assume that both A and B contain the genes $\{1, \ldots, n\}$. The *cycle graph* of A and B, denoted $G(A, B)$, is an undirected graph defined as follows. The set of vertices is $\bigcup_{i=1}^{n} \{i^0, i^1\}$. The vertices i^0 and i^1 are called the two *ends* of gene i (think of them as the ends of a small arrow directed from i^0 to i^1). For every pair of genes, i and j, where j immediately follows i in some chromosome of A (respectively, B) add a black (respectively, gray) (undirected) edge

$$(i, j) \equiv (out(i), in(j))$$

where

$$out(i) = \begin{cases} i^1 & \text{if } i \text{ has a positive sign in } A \text{ (respectively, } B) \\ i^0 & \text{otherwise} \end{cases}$$

and

$$in(j) = \begin{cases} j^0 & \text{if } j \text{ has a positive sign in } A \text{ (respectively, } B) \\ j^1 & \text{otherwise} \end{cases}$$

An example is given in Fig. 1(a). There are $n - N$ black edges and $n - N$ gray edges in $G(A, B)$. Since genomes A and B are co-tailed, every vertex in $G(A, B)$ has degree 2 or 0, where vertices of degree 0 (isolated vertices) belong to $Tails(A)$ (equivalently, $Tails(B)$). Therefore, $G(A, B)$ is uniquely decomposed into cycles with alternating gray and black edges.

In the following we assume, without loss of generality, that each chromosome of B is an increasing sequence of consecutive positive numbers. For example, $B_1 = \{(1, 2, 3, 4, 5), (6, 7, 8)\}$. Thus every gray edge in $G(A, B)$ is of the form $(out(i), in(i+1) \equiv (i^1, (i+1)^0) \equiv (i, i+1)$. As genomes B and A are co-tailed, once genome A is given, genome B is fixed. Thus we can define $G(A) \equiv G(A, B)$.

Let $c(A)$ denote the number of cycles in $G(A)$. Note that if $A = B$ then $c(A) = n - N$ is maximal. We denote by $A \cdot \phi$ the genome obtained after the translocation ϕ is applied to A. For any parameter ψ, let $\Delta\psi$ be the increase in ψ after applying ϕ, i.e., $\Delta\psi = \psi(A \cdot \phi) - \psi(A)$. The following lemma describes how c is affected by a translocation.

Lemma 1 ([11]) *Let ϕ be a translocation. If ϕ cuts two black edges in different cycles then the two cycles are merged into one cycle and $\Delta c = -1$. If ϕ*

acts on black edges belonging two the same cycle then either the cycle is split into two cycles and $\Delta c = 1$, or there is no change in the number of cycles (i.e. $\Delta c = 0$).

A translocation is *proper* if $\Delta c = 1$ (i.e. one cycle splits into two). A gray edge $(i, i+1)$ is *external* if i and $i+1$ belong to two different chromosomes, otherwise it is *internal*. For example, in Fig. 1(a), $(5, 6)$ is external, while $(11, 12)$ is internal. An *adjacency* is a cycle with two edges. Thus, every adjacency corresponds to a pair of genes $i, i+1$, where either $(i, i+1)$ or $(-i+1, -i)$ is contained in one of the chromosomes of A.

Observation 1 *Every external edge $(i, i+1)$ defines a (proper) translocation that creates the adjacency $(i, i+1)$.*

2.2 The Overlap Graph with Chromosomes

The overlap graph of a signed permutation was introduced in [9]. In this section we present an extension of this graph for genome A.

A *signed permutation* $\pi = (\pi_1, \ldots, \pi_n)$ is a permutation on the integers $\{1, \ldots, n\}$, where a sign of plus or minus is assigned to each number. Let A be a genome with the set of genes $\{1, \ldots, n\}$. Let π_A be an arbitrary concatenation of the chromosomes in A, in arbitrary order and orientation. Then π_A is a signed permutation of size n.

Place the vertices of $G(A)$ along a straight line according to their order in π_A. Now, every gray edge and every chromosome is associated with an interval of vertices in $G(A)$. Two intervals *overlap* if their intersection is not empty but none contains the other. The *overlap graph with chromosomes* of genome A w.r.t. π_A, denoted $OVCH(A, \pi_A)$, is defined as follows. The set of nodes is the set of chromosomes in A and gray edges in $G(A)$. Two nodes are connected if their corresponding intervals in $G(A)$ overlap. An example is given in Fig. 1(b). In order to prevent confusion, we will refer to nodes that correspond to chromosomes as "chromosomes" and reserve the word "vertex" for nodes that correspond to gray edges.

Let $OV(A, \pi_A)$ be the subgraph of $OVCH(A, \pi_A)$ induced by the set of nodes that correspond to gray edges (i.e., excluding the chromosomes' nodes). This graph is an extension of the overlap graph of a signed permutation defined in [9]. We shall use the word "component" for a connected component of $OV(A, \pi_A)$. For example, in Fig. 1(b), $OV(A_2, \pi_{A_2})$ contains six components: $\{(8, 9)\}$, $\{(1, 2), (2, 3)\}$, $\{(7, 8), (11, 12)\}$, $\{(9, 10), (10, 11)\}$, $\{(3, 4)\}$, and $\{(5, 6), (6, 7)\}$.

A vertex in $OVCH(A, \pi_A)$ is *external* if its corresponding edge in $G(A)$ is external, otherwise it is *internal*. For example, in Fig. 1(b), the vertex $(5, 6)$ is external while the vertex $(6, 7)$ is internal. Obviously a vertex is external iff it is connected to a chromosome.

A component is *external* if at least one of the vertices in it is external, otherwise it is *internal*. A component is *trivial* if it is composed of one internal vertex, which corresponds to an adjacency. For example, in Fig. 1, $\{(8, 9)\}$ is a trivial component, $\{(7, 8), (11, 12)\}$ is an internal non-trivial component, and $\{(3, 4)\}$ is an external component. Note that if $A = B$ then all the components are trivial. As we shall see later, a genome without non-trivial internal components can be sorted by a sequence of proper translocations. In case a genome does have non-trivial internal components, these components can become external after some non-proper translocations are applied.

The permutation π_A matches to every vertex v of $OV(A, \pi_A)$ an interval of genes, $I(v) \subset \pi_A$. For example, in Fig. 1(b) the vertex $(7, 8)$ is associated with the interval $(7, -11, 10, -9, -8)$. The interval associated with a component M, $I(M) \subset \pi_A$, is the minimal interval of genes for which $I(v) \subset I(M)$, for every vertex $v \in M$. For example, consider the components of $OV(A_2, \pi_{A_2})$, shown in Fig. 1(b). Then $I(\{(7, 8), (11, 12)\} = (7, -11, 10, -9, -8, 12)$ and $I(\{(5, 6), (6, 7)\}) = (-6, 7, -11, 10, -9, -8, 12, 5)$. Observe that the interval of the former component is contained within a chromosome, while the interval of the latter extends over two chromosomes.

Observation 2 *Let M be a component. Then M is internal iff $I(M)$ is contained in one chromosome.*

Observation 3 *The set of internal components is independent of the specific concatenation π_A. In other words, the set of internal components remains unchanged with all the concatenations of A.*

In [4] the term "component" is defined in a different manner. However, as we show below, the two definitions are equivalent when the components are internal. Note that the terms 'internal' and 'external' correspond to the terms 'intrachromosomal" and "interchromosomal" in [4]. To make a distinction, we refer to the term "component" defined in [4] as "BMS-component". We now define this term and prove the equivalence.

For a signed permutation π, we denote by $P(\pi)$ the signed permutation obtained from π by adding the first element 0 and the last element $n + 1$. For example, for the permutation in Fig. 1:

$$P(\pi_{A_2}) = (\mathbf{0}, 1, -2, 3, -6, 7, -11, 10, -9, -8, 12, 5, 4, \mathbf{13})$$

We refer to $P(\pi)$ as a *padded* signed permutation.

A *BMS-component* is an interval of $P(\pi)$, from i to $i+j$ or from $-(i+j)$ to $-i$, where $j > 0$, whose set of (unsigned) elements is $\{i, \ldots, i+j\}$, and that is not the union of smaller such intervals. For example, $P(\pi_{A_2})$ contains five BMS-components: $(1, -2, 3)$, $(3, \ldots, 13)$, $(7, \ldots, 12)$, $(-11, 10, -9)$, and $(-9, -8)$. The interval $(-11, 10, -9, -8)$ is not a BMS-component as it is the union of $(-11, 10, -9)$ and $(-9, -8)$.

The overlap graph of a signed permutation was originally defined for a padded permutation [9]. The connected components of this graph play a major role in the analysis of SBR. The analysis for SBR was revised in [3] and an alternative definition was given for the components of the overlap graph, namely BMS-components. It is implied in [3] that there is a bijective mapping between the set of BMS-components of $P(\pi_A)$ and the set of components in $OV(P(\pi_A))$, the overlap graph of $P(\pi_A)$. More specifically, I is a BMS-component of $P(\pi_A)$ iff $I = I(M)$ for some component M in $OV(P(\pi_A))$. A BMS-component I is *internal* if I is contained in one of the chromosomes of A.

Observation 4 *Let $I \subset \pi_A$. Then I is an internal BMS-component iff $I = I(M)$ for some internal component M.*

PROOF. Let A' be a uni-chromosomal genome whose single chromosome equals $P(\pi_A)$, i.e., $A' = \{P(\pi_A)\}$. The implied target genome is $\{(0, 1, \ldots, n+1)\}$. Following [9], $H' = OV(P(\pi_A)) \equiv OV(A', P(\pi_A))$. Thus $H = OV(A, \pi_A)$ is a subgraph of H', where the vertices in $H' \setminus H$ correspond to element pairs $(i, i+1)$ that are not adjacent in B. (In the example of Fig. 1, those will be the pairs $(0, 1)$, $(4, 5)$ and $(12, 13)$). Recall that for every BMS-component I there exists a component M in H' for which $I(M) = I$. Clearly if I is internal then all the vertices in M are internal too, and M is necessarily an internal component in H.

Observe that the vertices that are in $H' \setminus H$ cannot be adjacent to internal vertices in H, since in $G(A')$ the corresponding gray edges are adjacent to black edges bridging across chromosome ends. Therefore, if M is an internal component in H then M is also a component of H' and hence $I(M)$ is an internal BMS-component. □

2.3 The Forest of Internal Components

In this section we present the forest of internal components, originally defined in [4]. Let M' and M'' be two internal components. Then, as discussed in [4], $I(M')$ and $I(M'')$ are either disjoint, nested with different endpoints, or overlapping on one element. We define a *chain* as a sequence of internal components $(M_1, \ldots, M_t)$ in which $I(M_j)$ and $I(M_{j+1})$ overlap in exactly one

gene for $j = 1, .., t-1$. For example, in Fig. 1 let $M' = (\{(9, 10), (10, 11)\}$ and $M'' = \{(8, 9)\}$. Then (M', M'') is a chain, as $I(M')$ and $I(M'')$ overlap in one element, which is 9.

For a chain $C = (M_1, \ldots, M_t)$ define its associated interval as $I(C) = \bigcup_{j=1}^{t} I(M_j)$. A chain that cannot be extended to the left or right is called *maximal*. The *forest of internal components*, denoted $F(A)$, is defined by the following:

1. The vertices of $F(A)$ are: *(i)* the non-trivial internal components and *(ii)* maximal chains that contain at least one non-trivial component.
2. The children of a chain vertex are the non-trivial (internal)components it contains.
3. A chain vertex C is a child of the non-trivial internal component M with the smallest interval $I(M)$ satisfying $I(C) \subset I(M)$. If no such component exists then C is a root of its tree.

See Fig. 1(c) for an example. Observe that each tree in $F(A)$ is contained within one chromosome. For example, the two trees in Fig. 1(c) are contained in chromosome 1. We will refer to a component that is a leaf in $F(A)$ as simply a *leaf*. For example, there are two leaves in Fig. 1(c) corresponding to the intervals $(1, 2, 3)$ and $(-11, 10, -9)$.

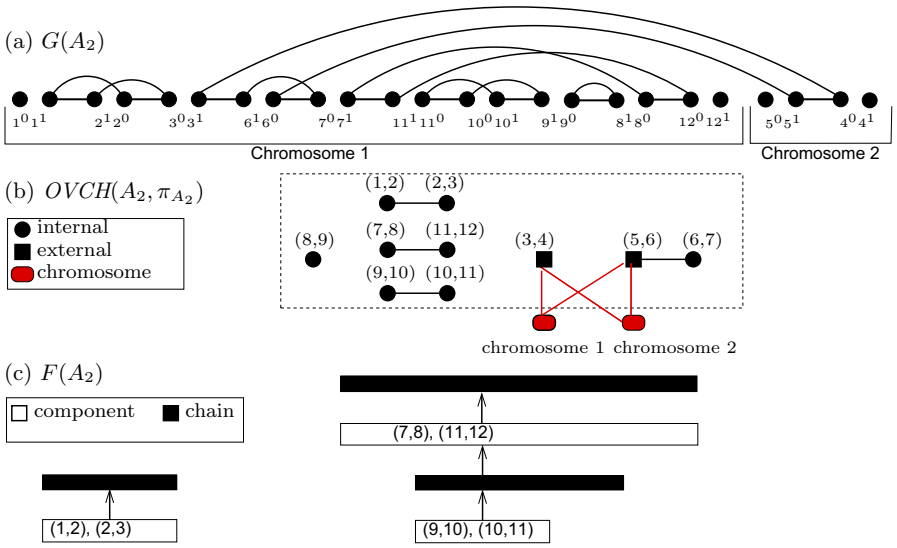


Fig. 1. Auxiliary graphs for $A_2 = \{(1, -2, 3, -6, 7, -11, 10, -9, -8, 12), (5, 4)\}$, $B_2 = \{(1, \ldots, 4), (5, \ldots, 12)\}$, $\pi_{A_2} = (1, -2, 3, -6, 7, -11, 10, -9, -8, 12, 5, 4)$. (a) The cycle graph. Black edges are horizontal; gray edges are curved (b) The overlap graph with chromosomes. The graph induced by the vertices within the dashed rectangle is $OV(A_2, \pi_{A_2})$, the same graph without the chromosome vertices. (c) The forest of internal components.

Note that if $A = B$ then all the components are trivial and hence $F(A)$ is empty. In addition, $F(A)$ is empty if no non-trivial internal component exists.

We say that a non-trivial internal component M is *eliminated* by a translocation ϕ if after ϕ is applied the vertices in M belong to external components. A translocation is called *bad* if $\Delta c = -1$ (i.e. two cycles are merged into one). The following observation describes how non-trivial internal components can be eliminated by bad translocations.

Observation 5 ([6, 4]) *A leaf M is eliminated by performing a translocation that cuts one black edge incident to a gray edge in M and one black edge in another chromosome of A. This translocation is necessarily bad. In addition, all the ancestor components of M in $F(A)$ are eliminated as well.*

An example of a translocation that eliminates two leaf components, with their ancestors, is shown in Fig. 2

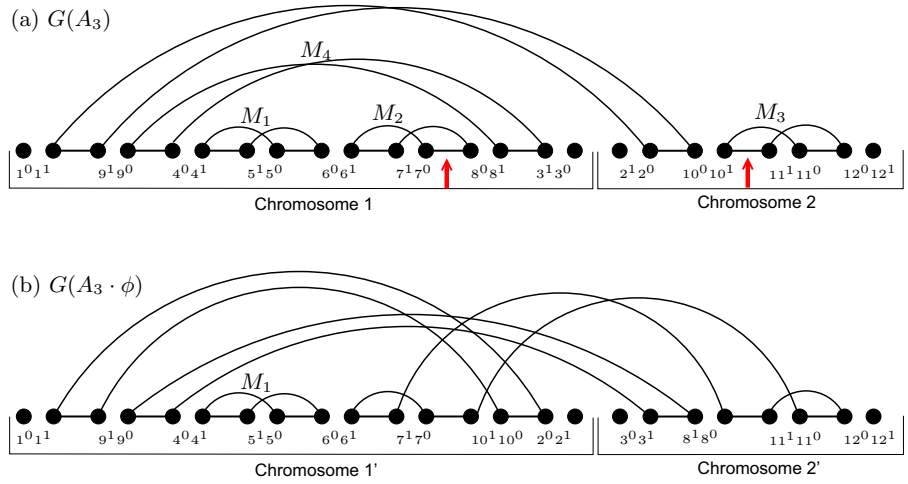


Fig. 2. An example of a bad translocation that eliminates two leaves.
(a) The cycle graph $G(A_3) \equiv G(A_3, B_3)$ where $A_3 = \{(1, -9, 4, -5, 6, -7, 8, -3), (-2, 10, -11, 12)\}$ and $B_3 = \{(1, 2), (3, 4, \ldots, 12)\}$). The four internal components are designated by $M_1, \ldots, M_4$.
(b) The cycle graph $G(A_3 \cdot \phi)$, where ϕ is a prefix-suffix translocation cutting the two black edges pointed by the vertical arrows in (a). In $A_3 \cdot \phi$ only one internal component exists, namely M_1. The other internal components, M_2, M_3, and M_4, were eliminated by ϕ.

2.4 The Translocation Distance

Let $T(A)$ and $L(A)$ denote the number of trees and leaves in $F(A)$, respectively. Obviously $T(A) \leq L(A)$. Define

$$f(A) = \begin{cases} 2 & \text{if } T(A) = 1 \text{ and } L(A) \text{ is even} \\ 1 & \text{if } L(A) \text{ is odd} \\ 0 & \text{otherwise } (T(A) \neq 1 \text{ and } L(A) \text{ is even}) \end{cases}$$

Theorem 2 ([6, 4] [2]) *The translocation distance between A and B is $d(A) = n - N - c(A) + L(A) + f(A)$*

An optimal move, i.e., a move that is part of a solution to SRT, is called *valid*.

Lemma 3 ([6, 4]) *$\Delta d = \Delta(-c + L + f) \geq -1$. A translocation ϕ is valid iff $\Delta d = -1$.*

A proper translocations is *safe* if it does not create new leaves. The analysis in [6, 4] implies that valid translocations are either: *(i)* bad, or *(ii)* proper and safe. Bad translocations are valid if $\Delta(L + f) = -2$. As was demonstrated by Bergeron et al. [4] a safe proper translocation may be invalid. However, if there are no leaves, which means that there are no non-trivial internal components, then a safe proper translocation is necessarily valid.

2.5 Analogy to SBR

For the readers familiar with the theory of SBR we now point to the analogy with the SRT theory. The minimum number of reversals needed to sort a signed permutation π (i.e., transform π into the identity permutation) depends on the number of cycles in the cycle graph $G(\pi)$, and on the "unoriented" components in $OV(\pi)$ [8, 9]. Unoriented components with minimal intervals are called "hurdles". The sorting of π requires the elimination of all hurdles by *bad reversals*, which decrease the number of cycles by one. If there are no hurdles, then π can be sorted by *proper reversals*, which increase the number of cycles by one. Thus there exists an analogy between the two distance formulas, of SBR and SRT. In particular, the parameter L, which indicates the number of leaves, is analogous to the parameter h, which indicates the number of hurdles.

[2] The formulas in [4] and [6] are equivalent: a leaf component is equivalent to a "minimal subpermutation" (minSP in short); the parameter s in [6], which denotes the number of minSPs, is equivalent to L; the term $(o + 2i)$ in [6] is equivalent to f.

The elimination of all hurdle components can be done linear time [9, 1], and is commonly performed at the beginning of the sorting algorithm. Thus SBR is linearly reduced to a simpler variant, "SBR-no hurdles". Most algorithms for SBR focus on solving this reduced form of SBR.

In the following we show that SRT can be reduced to "SRT-no leaves" in a similar manner, by eliminating all leaves in linear time. In addition, the algorithm we present in Section 4 for "SRT-no leaves" is an adaptation of an algorithm for "SBR-no hurdles". In [14] we show that two additional algorithms for "SBR-no hurdles" can be adapted to solve the "SRT-no leaves".

3 A Linear Reduction of SRT to SRTNL

A large part of the difficulty in analyzing the translocation distance (Theorem 2) is due to leaves: when there are no leaves $f(A) = L(A) = 0$ and the distance formula is much simpler. Motivated by this observation, we define SRTNL ("SRT-no leaves") as a special case of SRT when there are no leaves (i.e. $L(A) = T(A) = 0$). In this section we present a generic algorithm for solving SRT, using an algorithm for SRTNL. This algorithm, apart from two calls for solving an SRTNL instance, can be implemented in linear time.

Let $L(X)$ denote the number of leaves in chromosome X. Let $N^{\mathrm{L}}(A)$ denote the number of chromosomes of A containing at least one leaf. Equivalently, $N^{\mathrm{L}}(A)$ is the number of chromosomes for which $L(X) > 0$. The sorting of genome A into B requires the elimination of all leaves. The following lemmas describe how to eliminate leaves by valid (bad) translocations.

Lemma 4 *Suppose $N^{\mathrm{L}}(A) \geq 2$. Then there exists a valid bad translocation ϕ satisfying: (i) $\Delta L = -2$, and (ii) if $L(A \cdot \phi) \geq 2$ then $N^{\mathrm{L}}(A \cdot \phi) \geq 2$.*

PROOF. Assume $N^{\mathrm{L}}(A) \geq 2$. First, we prove that any bad translocation ϕ satisfying *(i)* and *(ii)* is necessarily valid. The parity of L is the same in A and in $A \cdot \phi$ and hence $\Delta f = 0$ ($f = 1$ if L is odd, and $f = 0$ otherwise). Therefore $\Delta d = \Delta(-c + L + f) = 1 - 2 + 0 = -1$ and ϕ is valid.

We shall now prove that there exists such a bad translocation. Choose $X_1, X_2 \in A$ such that $L(X_1) + L(X_2)$ is maximal. Suppose $L(X_1) \geq L(X_2)$.

Case 1: $L(X_1) \geq 2$ and $L(X_2) \geq 2$. Let ϕ be a (bad) prefix-prefix translocation that eliminates the second leaf from the left in X_1 and X_2 (Observation 5). Then each of the new chromosomes in $A \cdot \phi$ contains at least one leaf and hence $N^{\mathrm{L}}(A \cdot \phi) \geq 2$.

Case 2: $L(X_1) \geq 2$ and $L(X_2) = 1$. Let ϕ be a (bad) prefix-prefix translocation that eliminates the second leaf from the left in X_1 and the leaf in X_2. Then at least one of the new chromosomes in $A \cdot \phi$ contains exactly one leaf. If $L(A \cdot \phi) \geq 2$ then there must be another chromosome in $A \cdot \phi$ that contains at least one leaf and hence $N^{\mathrm{L}}(A \cdot \phi) \geq 2$.

Case 3: $L(X_1) = L(X_2) = 1$. Let ϕ be a (bad) translocation that eliminates the two leaves in X_1 and X_2. Clearly in $A \cdot \phi$ every chromosome contains at most one leaf. Hence, if $L(A \cdot \phi) \geq 2$ then $N^{\mathrm{L}}(A \cdot \phi) \geq 2$. $\square$

The following lemma follows from the proof of Theorem 13 in [6], and is proven here for completion.

Lemma 5 *Suppose $N^{\mathrm{L}}(A) = 1$, $L(A) \geq 2$, and $f(A) > 0$. Let ϕ be a (prefix-prefix) translocation that eliminates the second leaf from the left in A. Then ϕ is valid. In addition, if $L(A \cdot \phi) \geq 2$ then $N^{\mathrm{L}}(A \cdot \phi) \geq 2$.*

PROOF. Clearly $\Delta(-c + L) = 1 - 1 = 0$. If $L(A \cdot \phi) = 1$ then $L(A) = 2$ and $T(A) = 1$ and thus $\Delta f = -1$ and ϕ is valid.

Suppose $L(A \cdot \phi) \geq 2$. Let X' be the chromosome containing all the leaves in A, and let X'' be the the second chromosome on which ϕ acts. Then in genome $A \cdot \phi$: $L(X'') = 1$ and $L(X') > 0$, thus $N^{\mathrm{L}}(A \cdot \phi) \geq 2$. In particular $T(A \cdot \phi) > 1$ and $L(A \cdot \phi) = L(A) - 1$, so $\Delta f = -1$ and ϕ is valid. $\square$

Suppose there are several trees that are all located in one chromosome, i.e., $N^{\mathrm{L}}(A) = 1$, but $T(A) > 1$. To be able to eliminate a pair of leaves by one (bad) translocation, we first need to perform a sequence of (valid) proper translocations that "separates" the trees (and hence the leaves) into two different chromosomes. In the following we describe how to find such a sequence. We say that a sequence of translocations *sorts* a component M, if after performing the sequence every gray edge in M becomes an adjacency.

Lemma 6 *There is a sequence of safe proper translocations that sorts all external components (internal components are unchanged).*

PROOF. For an interval of genes $I = (i_1, \ldots, i_k)$ let $IN(I) = \{i_2, \ldots, i_{k-1}\}$. Let $S = \{i | i \in IN(I), \text{where } I \text{ is an interval corresponding to a tree}\}$. For example, in Fig. 1, $S = \{2, 8, 9, 10, 11\}$. Define A' and B' as the genomes obtained from A and B respectively after the deletion of the genes in S. Note that after a gene is deleted from a genome, its two neighbors become adjacent. Thus any interval corresponding to a tree of A is replaced in A' by a pair of genes

forming an adjacency. Therefore $G(A')$ contains no leaves. Thus there is a sequence of safe proper translocations that sorts A' into B' (Theorem 2). This sequence induces a sequence of safe proper translocations on A that sorts all the external components in $G(A)$. $\square$

We call a translocation ϕ *separating* if $N^{\mathrm{L}}(A) = 1$ and $N^{\mathrm{L}}(A \cdot \phi) = 2$. The following lemma shows how to find a sequence of valid proper translocations, whose last translocation is separating.

Lemma 7 *Suppose $N^{\mathrm{L}}(A) = 1$ and $T(A) > 1$. Let $S = (\phi_1, \ldots, \phi_k)$ be a sequence of safe proper translocations that sorts all the external components in $G(A)$. Then S contains a separating translocation ϕ_l, $l \in \{1, \ldots, k\}$. Moreover, $S_l = (\phi_1, \ldots, \phi_l)$ is a sequence of valid translocations.*

PROOF. Apply the translocations in S by their order. Let $A_0 = A$ and let A_i be the genome obtained after applying $(\phi_1, \ldots, \phi_i)$ to A. Suppose that S does not contain a separating translocation. Thus, by our assumption $N^{\mathrm{L}}(A_i) = 1$ for $i = 1, \ldots, k$. Observe that a chromosome that contains two trees necessarily contains the endpoint of an external edge. Thus $T(A_k) = 1$, since in A_k there are no external edges and all the leaves belong to one chromosome. Since $T(A) > 1$, there exists $\phi_t \in S$ such that $T(A_{t-1}) > 1$ and $T(A_t) = 1$. Now, ϕ_t is a safe proper translocation and hence does not eliminate any internal component, thus A_{t-1} must contain two trees in two different chromosomes. Therefore $N^{\mathrm{L}}(A_{t-1}) > 1$, a contradiction.

Thus there exists i for which $N^{\mathrm{L}}(A_i) > 1$. Let l be the first index for which $N^{\mathrm{L}}(A_l) > 1$. Then ϕ_l is a separating translocation. As S_l contains only safe proper translocations $L(A_l) = L(A)$ and thus $f(A_l) = f(A)$. Hence $d(A_l) - d(A) = l$ and thus every translocation in S_l is valid. $\square$

Lemmas 4-7 motivate Algorithm 1 for SRT. This algorithm focuses on the efficient and optimal elimination of all leaf components. If all the leaves belong to one chromosome, then we either use Lemma 5 or Lemma 7 to separate the leaves into two chromosomes. Then we use Lemma 4 to eliminate pairs of leaves. At the end, either all leaves have been eliminated, or we are left with a single leaf, which is eliminated by one (valid) bad translocation.

Lemma 8 *Algorithm 1, excluding the two calls to a SRTNL algorithm, can be implemented in linear time.*

PROOF. The computation of all the parameters can be done in linear time,

Algorithm 1 *An algorithm for solving SRT using an algorithm for SRTNL*

1: **if** $N^{\mathrm{L}} = 1$ **and** $L \geq 2$ **then**
2: **if** $f > 0$ **then**
3: Eliminate the second leaf from the left by a prefix-prefix translocation */*Lemma 5*/*
4: **else**
5: Compute a sequence S of safe proper translocations that sorts all external components */*using an algorithm for SRTNL, Lemma 6*/*
6: Iteratively perform the translocations in S **until** $N^{\mathrm{L}} > 1$ */*Lemma 7*/*
7: **end if**
8: **end if**
9: Let Q_1 be the list of chromosomes containing exactly one leaf
10: Let Q_2 be the list of chromosomes containing at least two leaves
11: **while** $L > 0$ **do**
12: **if** $L = 1$ **then**
13: Eliminate the single leaf by a prefix-prefix translocation
14: **else**
15: **for** $i = 1, 2$ **do**
16: **if** $Q_2 \neq \emptyset$ **then**
17: $X_i \leftarrow$ an element from Q_2. Remove X_i from Q_2
18: $l_i \leftarrow$ the second leaf from the left in chromosome X_i
19: **else**
20: $X_i \leftarrow$ an element from Q_1. Remove X_i from Q_1
21: $l_i \leftarrow$ the single leaf in X_i
22: **end if**
23: **end for**
24: Eliminate l_1 and l_2 by a prefix-prefix translocation */*Lemma 4*/*
25: **for** $i = 1, 2$ **do**
26: **if** $L(X_i) \geq 2$ **then**
27: add X_i to Q_2
28: **else if** $L(X_i) = 1$ **then**
29: add X_i to Q_1
30: **end if**
31: **end for**
32: **end if**
33: **end while***/*Invariant:* $N^{\mathrm{L}} \geq 2$ **or** $L = 1$ **/*
34: Solve SRTNL on A

in a similar manner to the computation of the translocation distance [4].

Steps 5 and 6 are implemented by calling a procedure for SRTNL. However, we need to stop this procedure when a separating translocation is applied. We can locate this separating procedure in linear time by acting as follows. Suppose that $N^{\mathrm{L}} = 1$, $T > 1$ and $S = (\phi_1, \ldots, \phi_k)$ is a sequence of safe proper translocations that sorts all the external components. By Lemma 7 there exists

a separating translocation ϕ_l in S. Let I be the minimum interval of genes that contains the intervals of all the leaves. We say that a translocation ϕ *cuts* I if one of the black edges it cuts is contained in I. Note that since I is contained in a single chromosome, a translocation cuts at most one black edge in I. Clearly ϕ_l cuts I. On the other hand, the first translocation that cuts I is necessarily separating. For every translocation ϕ_i in S we can test in $O(1)$ time whether it cuts I.

We implement Steps 11-33 in linear time, as follows. For each chromosome we maintain its genes and the leaves it contains in two ordered linked lists. We use only prefix-prefix (bad) translocations that do not change the signs of the translocated genes. Thus the update of the genes and leaves lists of the chromosomes after a translocation is done in $O(1)$. $\square$

Lemma 8 immediately implies:

Theorem 9 *SRT is linearly reducible to SRTNL.*

4 An Algorithm for SRTNL

In this section we present an algorithm for SRTNL. We first describe how the overlap graph is changed after performing a chromosome flip or a proper translocation defined by an external vertex.

As was demonstrated by Hannenhalli and Pevzner [7], a reversal on π_A simulates a translocation on A:

$$(\ldots, X_1, \underline{X_2, \ldots, Y_1}, Y_2, \ldots) \Rightarrow (\ldots, X_1, \underline{-Y_1, \ldots, -X_2}, Y_2, \ldots).$$

The type of translocation depends on the relative orientation of X and Y in π_A (and not on their order): if the orientation is the same, then the translocation is prefix-suffix, otherwise it is prefix-prefix. The segment between X_2 and Y_1 may contain additional chromosomes that are flipped and thus unaffected.

4.1 Updating OVCH for chromosome flips and proper translocations

Suppose $H_1 = OVCH(A, \pi_1)$ and $H_2 = OVCH(A, \pi_2)$, where π_1 and π_2 are two different concatenations and orientations of the chromosomes in A. In this case we refer to H_1 and H_2 as *equivalent*.

Let $H = OVCH(A, \pi_A)$. Let $IN(H)$ denote the set of vertices that are in non-trivial internal components. Thus two equivalent graphs, H_1 and H_2, satisfy $IN(H_1) = IN(H_2)$ (Observation 3).

Let v be any vertex in H. Denote by $CH(v) \equiv CH(v, H)$ the set of chromosomes that are neighbors of v in H. Hence if v is external then $|CH(v)| = 2$, otherwise $CH(v) = \emptyset$ (compare Fig. 1(b)). For a chromosome X, let $\phi(X)$ denote a flip of chromosome X in π_A. Let $H \cdot \phi(X) = OVCH(A, \pi_A \cdot \phi(X))$. Hence, in particular $H \cdot \phi(X)$ and H are equivalent.

Lemma 10 ([14]) *$H \cdot \phi(X)$ is obtained from H by complementing the subgraph induced by the set $\{u : X \in CH(u)\}$ and flipping the orientation of every vertex in it.*

Let v be an external vertex in H. Denote by $\phi(v)$ the proper translocation that the corresponding gray edge defines on A (recall Observation 1). Two external vertices v_1 and v_2 in H are *equivalent* if they define the same translocation, i.e. $\phi(v_1) \equiv \phi(v_2)$.

A vertex in the overlap graph is *oriented* if its corresponding edge connects two genes with different signs in π_A, otherwise it is *unoriented*. If v is an oriented external vertex then $\phi(v)$ can be mimicked by a reversal, $\hat{\phi}(v)$, on π_A.

For an external vertex v we define $H \cdot \phi(v)$ in the following way. If v is oriented then $H \cdot \phi(v) = OVCH(A \cdot \phi(v), \pi_A \cdot \hat{\phi}(v))$. Otherwise, suppose $CH(v) = \{X, Y\}$ and that Y appears after X in π_A. Then v is an oriented external vertex in $H' = H \cdot \phi(X)$ and thus we define $H \cdot \phi(v) = H' \cdot \phi(v)$.

Denote by $N(v) \equiv N(v, H)$ the set of vertices that are neighbors of v, including v itself (but not including chromosome neighbors). Given two sets S_1 and S_2 define $S_1 \bigoplus S_2 = (S_1 \bigcup S_2) \setminus (S_1 \bigcap S_2)$. Finally, two chromosomes in $OVCH(A, \pi_A)$ are called *consecutive* if they are consecutive in π_A.

Lemma 11 ([14]) *Let v be an oriented external vertex in H and suppose the chromosomes in $CH(v)$ are consecutive. Then $H \cdot \phi(v)$ is obtained from H by the following operations. (i) Complement the subgraph induced by $N(v)$ and flip the orientation of every vertex in $N(v)$. (ii) For every vertex $u \in N(v)$ update the edges between u and $CH(u) \bigcup CH(v)$ such that $CH(u) = CH(u) \bigoplus CH(v)$. In particular, the external/internal state of a vertex $u \in N(v)$ is flipped iff u is internal or $CH(u) = CH(v)$.*

Lemmas 10 and 11 describe the change in $OVCH(A, \pi_A)$ after performing operations that can be mapped to reversals on π_A. Therefore, the described change in $OVCH(A, \pi_A)$ is similar to the change in $OV(\pi)$ after performing a reversal [9, Observation 4.1].

We now describe the main theorem and algorithm. Our algorithm is formally very similar to the algorithm for SBR presented in [15]. Instead of performing reversals on oriented edges in [15], we perform translocations on external edges. Despite of the great similarity between the algorithms our validity proof is completely new. We analyze an overlap graph with chromosomes of a multi-chromosomal genome, while [15] analyze the overlap graph of a uni-chromosomal genome. Like [15], we perform operations defined by oriented vertices (i.e. translocations). However, in our case these vertices must also be external. If an external vertex is unoriented, we can turn it into an oriented vertex by a flip of a chromosome. Hence, we consider two types of operations in our analysis.

A sequence of vertices $S = (v_1, \ldots, v_k)$ from H is *legal* if v_j is external in $H \cdot \phi(v_1) \cdots \phi(v_{j-1})$ for $j = 1, .., k$. For a legal sequence S define $\phi(S) = \phi(v_1) \cdots \phi(v_k)$. A legal sequence S is *total* if $H \cdot \phi(S)$ contains only trivial components. For an overlap graph with chromosomes H_1, let $EXT(H_1)$ denote the set of vertices that are in external components. If S is a maximal legal sequence of vertices in H then $EXT(H \cdot \phi(S)) = \emptyset$. If in addition S is not total then $IN(H \cdot \phi(S)) \neq \emptyset$.

Theorem 12 *Let $S = (v_1, \ldots, v_k)$ be a maximal legal but not total sequence of vertices in H. Let $IN = IN(H \cdot \phi(S))$. Let v_l be the first vertex in S satisfying $IN(H \cdot \phi(v_1, \ldots, v_l)) = IN$, i.e. $\phi(v_l)$ is the last unsafe translocation in $\phi(S)$. Let $S_1 = (v_1, \ldots, v_{l-1})$ and $S_2 = (v_l, \ldots, v_k)$. Then every maximal sequence of vertices $S' = (w_1, \ldots, w_m)$ in IN that satisfies (i) (S_1, S') is legal and (ii) v_l is not an adjacency in $H \cdot \phi(S_1, S')$ also satisfies: (iii) S' is not empty and (iv) (S_1, S', S_2) is a maximal legal sequence. Moreover, all the translocations in $\phi(S_2)$ are safe.*

PROOF. Let $v = v_l$, $H_0 = H \cdot \phi(S_1)$ and $IN_0 = EXT(H_0) \cap IN$. Then $IN_0 \neq \emptyset$ and none of the vertices in IN_0 is equivalent to v in H_0 (otherwise it would be an adjacency in $H \cdot \phi(S)$ and hence not in IN). Hence S' is not empty. Let $A_0 = A \cdot \phi(S_1)$ and $CH(v) = \{X, Y\}$. We choose π_0 to be a concatenation of the chromosomes in A_0 in which X and Y are the first two chromosomes. We can assume w.l.o.g. that $H = OVCH(A, \pi_0)$, hence $H_0 = OVCH(A_0, \pi_0)$. For $j = 1, .., m$ let $H_j = H_0 \cdot \phi(w_1, \ldots, w_j)$. Let $IN_j = EXT(H_j) \bigcap IN$. Then for $j = 1, \ldots, m$: *(i)* $w_j \in IN_{j-1}$ and *(ii)* w_j is not equivalent to v in H_{j-1}. Let $EXT = EXT(H_0 \cdot \phi(v))$. The following conditions hold for H_j when $j = 0$ (see Fig. 4-(a)):

(1) The subgraphs of $H_j \cdot \phi(v)$ and $H_0 \cdot \phi(v)$ that are induced by EXT are

equivalent.
(2) Every $w \in IN_j$ satisfies: $CH(w) = CH(v) = \{X, Y\}$.
(3) If v is oriented then $N(v) \bigcap IN = IN_j$.
(4) All the possible edges exist between $N(v) \bigcap EXT$ and IN_j.
(5) There are no edges between $IN \setminus IN_j$ and vertices outside IN.
(6) There are no edges between $EXT \setminus N(v)$ and vertices outside EXT.

We shall prove below that in H_m v is external and that all the above conditions are satisfied. The first condition ensures that (S_1, S', S_2) is legal. The rest of the conditions ensure that $H_m \cdot \phi(v)$ satisfies: *(i)* there are no external vertices in IN and *(ii)* there are no edges between EXT and vertices outside EXT. Hence (S_1, S', S_2) is maximal and every translocation in $\phi(v_{l+1}, \ldots, v_k)$ is safe. $\phi(v_l)$ is safe in H_m since S' is maximal. Therefore, all the translocations in $\phi(S_2)$ are safe.

Assume that v is external in H_j and that all the above conditions hold for a certain j. Since these conditions are true for every graph that is equivalent to H_j we can assume that v is oriented. We now prove, using induction on j, that these conditions are satisfied for every H_i, $i \in \{1, \ldots, m\}$ in which v is external, and that v is external in H_m.

Case 1: w_{j+1} is oriented in H_j. Let $H_{j+1} = H_j \cdot \phi(w_{j+1})$ (see Fig. 4-(b)). Then $IN_{j+1} = N(v, H_j) \bigoplus N(w_{j+1}, H_j)$. $IN_{j+1} \neq \emptyset$, otherwise v is an isolated internal vertex in H_{j+1} and hence equivalent to w_{j+1} in H_j. Hence $m \geq j + 2$.

Case 1.*a*: w_{j+2} is oriented in H_{j+1}. Let $H_{j+2} = H_{j+1} \cdot \phi(w_{j+2})$ (see Fig. 4-(c)). Clearly, v is external in H_{j+2}. Let $M = N(v, H_j) \bigcap EXT$. Then $N(w_{j+2}, H_{j+1}) \bigcap EXT = N(w_{j+1}, H_j) \bigcap EXT = M$. Hence the subgraphs of H_{j+2} and H_j that are induced by M are identical and the first condition is satisfied in H_{j+2}.

Case 1.*b*: w_{j+2} is unoriented in H_{j+1}. Let $H'_{j+1} = H_{j+1} \cdot \phi(X)$ (H'_{j+1} and H_{j+1} are equivalent) (see Fig. 4-(d)). Hence w_{j+2} is oriented in H'_{j+1}. Note that v is an internal vertex in H'_j. Let $M' = N(w_{j+1}, H'_{j+1}) \bigcap EXT$. Let $H_{j+2} = H'_{j+1} \cdot \phi(w_{j+2})$ (see Fig. 4-(e)). v is an oriented external vertex in H_{j+2} and $N(v, H_{j+2}) \bigcap EXT = M'$. Therefore, the two subgraphs of $H_{j+2} \cdot \phi(v)$ (see Fig. 4-(f)) and H'_{j+1} (see Fig. 4-(d)) that are induced by EXT are identical. The subgraphs of H_{j+1} and $H_j \cdot \phi(v)$ that are induced by EXT are also identical. Hence, the first condition is satisfied.

Looking at Figs. 4-(c) and 4-(e) it is easy to verify that the rest of the conditions are also satisfied for H_{j+2}.

Case 2: w_{j+1} is unoriented in H_j. We define the three subsets of vertices $M_1, M_2, M_3 \subset EXT$ in H_j as follows:

(1) M_1 is the set of neighbors of w_{j+1} (equivalently, v) that are either internal

or external but does not overlap chromosome X.

(2) M_2 is the set of neighbors of w_{j+1} (equivalently, v) that overlap chromosome X. Hence $M_1 \bigcup M_2 = N(v, H_j) \bigcap EXT$.

(3) M_3 is the set of vertices that overlap chromosome X but are not neighbors of w_{j+1} (equivalently, v).

For an illustration of H_j see Fig. 4-(g). Let $H'_j = H_j \cdot \phi(X)$ (see Fig. 4-(h)). In H'_j: w_{j+1} is an oriented external vertex and is not a neighbor of v. Let $H_{j+1} = H'_j \cdot \phi(w_{j+1})$ (see Fig. 4-(i)). Obviously, v remains intact in H_{j+1}. Let $H'_{j+1} = H_{j+1} \cdot \phi(X)$ (see Fig. 4-(j)). Then, the subgraphs of $H'_{j+1} \cdot \phi(v)$ (see Fig. 4-(k)) and $H_j \cdot \phi(v)$ that are induced by M_1, M_2 and M_3 are equivalent (Compare the subgraph induced by EXT in H_j in Fig. 4 (g) with the subgraph induced by EXT in $H'_{j+1} \cdot \phi(v) \cdot \phi(X)$ in Fig. 4 (l)). Hence the first condition is satisfied. Looking at Fig. 4-(i), it is easy to verify that conditions (2)-(6) hold for H_{j+1}. $\square$

The algorithm in Fig. 2 builds a sequence of gray edges in $G(A)$, (S_1, S_2), that corresponds to a total legal sequence of vertices from H. The sequence (S_1, S_2) is built by a repeated application of Theorem 12. It greedily removes external edges in $G(A)$ from an allowed subset and performs the corresponding translocations (step (2).(a)). When the allowed subset contains only internal gray edges, the algorithm repeats the last translocations in a reverse order (thereby cancelling them) until another vertex in the allowed subset becomes external (step (2).(b)). Figure 3 describes an example of a run of the algorithm. Every translocation in the algorithm is applied at most twice and so the algorithm performs at most $2n$ translocations.

5 An $\mathbf{O(n^{3/2}\sqrt{\log(n)})}$ Time Implementation of the Algorithm

The algorithm in Fig. 2 can be implemented in $O(n^2)$ time in a relatively simple manner. We provide below an $O(n^{3/2}\sqrt{\log(n)})$ algorithm. The implementation follows closely the ideas of [10] and [15].

We identify a gray edge $(i, i+1)$ by i and refer to $(i+1)$ as the *remote end* of i. The data structure we use for maintaining the genome A is as follows.

(1) A doubly linked list of $O(\sqrt{\frac{n}{\log(n)}})$ blocks. We partition π_A into continuous blocks such that the size of every block is at least $\frac{1}{2}\sqrt{n\log(n)}$ and at most $2\sqrt{n\log(n)}$.

(2) A balanced search tree for every block. The tree contains the edges in the block ordered by the positions of their remote ends. We use balanced

Algorithm 2 *An algorithm for solving SRTNL*

Let V be the set of gray edges in $G(A)$ that are in non-trivial components
$S_1 = S_2 = \emptyset$
$\Phi = \emptyset$
while $V \neq \emptyset$ **do**
  while there exists an external gray edge $v \in V$ in $G(A)$ **do**
    Remove v from V
    if v is not equivalent to the first element in S_2 **then**
      Append v to S_1
      Append $\phi(v)$ to Φ
      $A \leftarrow A \cdot \phi(v)$
    end if
  end while
  if $V = \emptyset$ **then**
    return $\phi(S_1, S_2)$
  end if
  while all the gray edges in V are internal in $G(A)$ **do**
    Let v be the last gray edge in S_1. Remove v from S_1
    Prepend v to S_2
    Let ϕ be the last translocation in Φ. Remove ϕ from Φ
    $A \leftarrow A \cdot \phi$
  end while
end while

trees that support split and concatenate operations in logarithmic time, such as red-black trees or 2-4 trees. We use $T[v]$ to denote the subtree rooted at v and containing all its descendants.

(3) An n-array of block pointers. The i^{th} entry in the array points to the block containing i.

We add the following fields to the above data structure.

(1) For each edge we keep an external-bit. If the external-bit is *on* then the edge is external, otherwise it is internal.

(2) For each block we keep the following fields: *(i)* a counter of external edges in V, *(ii)* a counter of chromosomes' left tails, and *(iii)* a reverse-flag. If the reverse-flag of a block is *on* then the order and signs of the elements in the block are reversed.

(3) For every subtree $T[v]$ of each block's search tree we keep the following fields in its root v: *(i)* counters of external and internal edges in V, *(ii)* a direction-flip-flag and *(iii)* an external-flip-flag. If the external-flip-flag of a vertex v is *on* then in $T[v]$ the external-bits of all the elements are flipped and the counters of internal and external elements from V exchange their values. If the direction-flip-flag of a vertex v is *on* then in $T[v]$ the order of the elements is reversed.

genome A	S_1	S_2	V
$(-8,-2,\underline{7,3}),(\underline{1},6,5,-4)$	$\emptyset$	$\emptyset$	$1,2,4,5,6,7$
$(\underline{-8},-2,-1),(\underline{-3},-7,6,5,-4)$	1	$\emptyset$	$2,4,5,6,7$
$(\underline{-3},-2,-1),(\underline{-8},-7,6,5,-4)$	$1,2$	$\emptyset$	$4,5,6,7$
$(-8,-2,-1),(-3,-7,6,5,-4)$	1	2	$4,5,6,7$
$(-8,-2,\underline{-1}),(\underline{-3,-7},6,5,-4)$	1	2	$4,5,6$
$(\underline{-8,-2},7,3),(\underline{1,6},5,-4)$	$\emptyset$	$1,2$	$4,5,6$
$(\underline{1},6,7,3),(\underline{-8,-2,5},-4)$	6	$1,2$	$4,5$
$(-8,-2,5,6,7,3),(1,-4)$	$6,5$	$1,2$	4
$(-8,-2,\underline{5,6,7,3}),(\underline{1},-4)$	$6,5$	$1,2$	$\emptyset$
$(\underline{-8},-2,-1),(\underline{-3},-7,-6,-5,-4)$			
$(-3,-2,-1),(-8,-7,-6,-5,-4)$			

Fig. 3. An example for a run of the algorithm on genomes $A = \{(-8,-2,7,3),(1,6,5,-4)\}$ and $B = \{(1,2,3),(4,\ldots,8)\}$. A gray edge $(i,i+1)$ (vertex of H) is represented by i. The underlined segments denote a translocation the algorithm chose. The algorithm ends when $V = \emptyset$. The top 9 lines describe the steps of the algorithm. The two bottom lines show the application of $\phi(S_2) = \phi(1,2)$ on the final genome produced by the algorithm, producing B.

We can clear the direction-flip-flag of a node by reversing the order of its children and flipping the direction-flip-flag in each of them. We can clear the external-flip-flag in a node by exchanging the values of the counters of external and internal edges in V, flipping the external-flip-flag in each of its children and flipping the external-bit of the element residing at the node. One can view this procedure as "pushing down" the flags. An direction-flip-flag and an external-flip-flag that are *on* are "pushed down" whenever $T[v]$ is searched.

We implement the algorithm using the above data structures. A search for an external edge in V is done as follows. We traverse the list of blocks until we reach a block that contains external edges from V. We then search the tree of the block for an external edge i. We locate element $i+1$ (the remote end of edge i) using the n-array and a search of its block.

Let ϕ be a translocation on A operating on the chromosomes $X = (X_1, X_2)$ and $Y = (Y_1, Y_2)$. Then ϕ is performed in $O(\sqrt{n\log(n)})$ time as follows:

(1) Split at most six blocks so that each of the four segments X_1, X_2, Y_1 and Y_2 corresponds to a union of blocks. If ϕ is a prefix-prefix translocation

exchange the blocks of X_1 and Y_1. Otherwise, reverse the order and flip the reverse-flags of the blocks of X_2 and Y_1 and then exchange the blocks of X_2 and Y_1.

(2) We now have to modify the trees of each block to reflect the order and direction changes. This is done as follows. Traverse all the blocks and for each block:
   (a) Let T be the balanced search tree of the block. If ϕ is a translocation on an edge i in V and i is contained in the block: decrease by 1 the counters of external edges in V of the block and of every node in T that contains i in its subtree.
   (b) Split T into at most seven subtrees such that each of the segments X_1, X_2, Y_1 and Y_2 has a corresponding subtree.
   (c) If the block corresponds to a segment of X_1, X_2, Y_1 and Y_2 flip the external-flip-flag at the roots of two subtrees according to Table 1.
   (d) If ϕ is a prefix-prefix translocation, exchange the subtrees of X_1 and Y_1. Otherwise, exchange the subtrees of X_2 and Y_1 and flip the direction-flip-flags of both.
   (e) Concatenate the seven subtrees into T.

(3) If necessary, concatenate small blocks and split large blocks such that the size of each block is at least $\frac{1}{2}\sqrt{n\log(n)}$ and at most $2\sqrt{n\log(n)}$.

Table 1
The subtrees for which the external-flip-flag is flipped as a function of translocation type and block type.

Block	X_1	X_2	Y_1	Y_2
prefix-prefix	X_2, Y_2	X_1, Y_1	X_2, Y_2	X_1, Y_1
prefix-suffix	X_2, Y_1	X_1, Y_2	X_1, Y_2	X_2, Y_1

Theorem 13 *SRTNL can be solved in* $O(n^{3/2}\sqrt{\log(n)})$. □

Acknowledgments

This study was supported in part by the Raymond and Beverly Sackler chair in Bioinformatics and by the Israel Science Foundation (grant no. 802/08).

References

[1] D.A. Bader, B. M.E. Moret, and M. Yan. A linear-time algorithm for computing inversion distance between signed permutations with an experimental study. *Journal of Computational Biology*, 8(5):483–491, 2001.

[2] A. Bergeron. A very elementary presentation of the Hannenhalli-Pevzner theory. *Discrete Applied Mathematics*, 146(2):134–145, 2005.

[3] A. Bergeron, J. Mixtacki, and J. Stoye. Reversal distance without hurdles and fortresses. In *Proceedings of the 15th Annual Symposium on Combinaotrial Pattern Matching (CPM)*, volume 3109 of *LNCS*, pages 388–399. Springer, 2004.

[4] A. Bergeron, J. Mixtacki, and J. Stoye. On sorting by translocations. *Journal of Computational Biology*, 13(2):567–578, 2006.

[5] P. Berman and S. Hannenhalli. Fast sorting by reversal. In *Proceedings of the 7th Annual Symposium Combinatorial Pattern Matching (CPM)*, volume 1075 of *LNCS*, pages 168–185. Springer, 1996.

[6] S. Hannenhalli. Polynomial algorithm for computing translocation distance between genomes. *Discrete Applied Mathematics*, 71:137–151, 1996.

[7] S. Hannenhalli and P. Pevzner. Transforming men into mice (polynomial algorithm for genomic distance problems). In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 581–592. IEEE Computer Society Press, 1995.

[8] S. Hannenhalli and P. Pevzner. Transforming cabbage into turnip: Polynomial algorithm for sorting signed permutations by reversals. *Journal of the ACM*, 46:1–27, 1999.

[9] H. Kaplan, R. Shamir, and R. E. Tarjan. Faster and simpler algorithm for sorting signed permutations by reversals. *SIAM Journal of Computing*, 29(3):880–892, 2000.

[10] H. Kaplan and E. Verbin. Sorting signed permutations by reversals, revisited. *Journal of Computer and System Sciences*, 70(3):321–341, 2005.

[11] J. D. Kececioglu and R. Ravi. Of mice and men: Algorithms for evolutionary distances between genomes with translocation. In *Proceedings of the 6th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 604–613. ACM Press, 1995.

[12] G. Li, X. Qi, X. Wang, and B. Zhu. A linear-time algorithm for computing translocation distance between signed genomes. In *Proceedings of the 15th Annual Symposium on Combinatorial Pattern Matching (CPM)*, volume 3109 of *LNCS*, pages 323–332. Springer, 2004.

[13] M. Ozery-Flato and R. Shamir. An $O(n^{3/2}\sqrt{\log(n)})$ algorithm for sorting by reciprocal translocations. In *Proceedings of the 17th Annual Symposium on Combinatorial Pattern Matching (CPM)*, volume 4009 of *LNCS*. Springer, 2006.

[14] M. Ozery-Flato and R. Shamir. Sorting by translocations via reversals theory. *Journal of Computational Biology*, 14(4):408–422, 2007.

[15] E. Tannier, A. Bergeron, and M. Sagot. Advances on sorting by reversals. *Discrete Applied Mathematics*, 155(6-7):881–888.

[16] L. Wang, D. Zhu, X. Liu, and S. Ma. An o(n2) algorithm for signed translocation. *Journal of Computer and System Sciences*, 70(3):284 – 299, 2005.

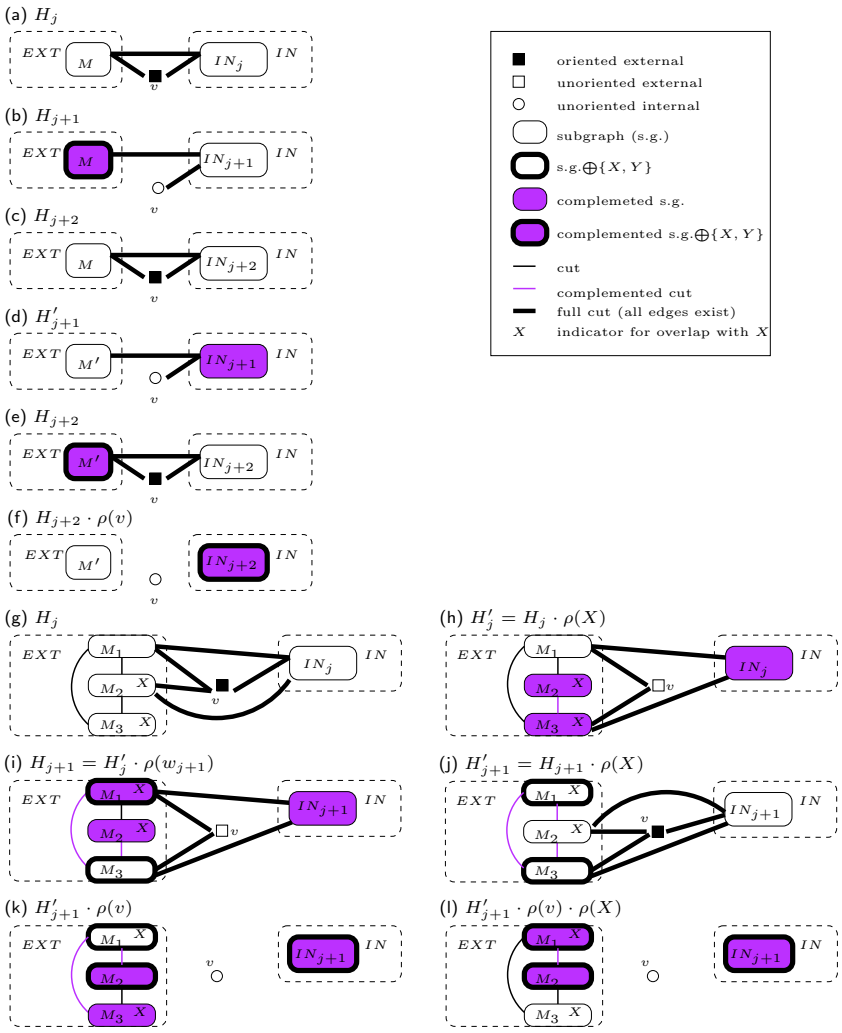


Fig. 4. Illustrations for the proof of Theorem 12.

Chapter 3

Sorting by Reciprocal Translocations via Reversals Theory

JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 14, Number 4, 2007

Pp. 408–422
DOI: 10.1089/cmb.2007.A003

Sorting by Reciprocal Translocations via Reversals Theory

MICHAL OZERY-FLATO and RON SHAMIR

ABSTRACT

The understanding of genome rearrangements is an important endeavor in comparative genomics. A major computational problem in this field is finding a shortest sequence of genome rearrangements that transforms, or sorts, one genome into another. In this paper we focus on sorting a multi-chromosomal genome by translocations. We reveal new relationships between this problem and the well studied problem of sorting by reversals. Based on these relationships, we develop two new algorithms for sorting by reciprocal translocations, which mimic known algorithms for sorting by reversals: a score-based method building on Bergeron's algorithm, and a recursive procedure similar to the Berman-Hannenhalli method. Though their proofs are more involved, our procedures for reciprocal translocations match the complexities of the original ones for reversals.

Key words: genome rearrangement, sorting by translocations, sorting by reversals.

1. INTRODUCTION

FOR OVER A DECADE NOW, much effort has been put into large-scale genome sequencing projects. Analysis of the sequences that have accumulated so far showed that genome rearrangements play an important role in the evolution of species. A major computational problem in the research of genome rearrangements is finding a most parsimonious sequence of rearrangements that transforms one genome into another. This is called the *genomic sorting problem*, and the corresponding number of rearrangements is called the *rearrangement distance* between the two genomes. Genomic sorting gives rise to a spectrum of fascinating combinatorial problems, each defined by the set of allowed rearrangement operations and by the representation of the genomes.

In this paper we focus on the problem of sorting by translocations. We reveal new similarities between sorting by translocations and the well studied problem of sorting by reversals. The study of the problem of sorting by translocations is essential for the full comprehension of any genomic sorting problem that permits translocations. Below we review the relevant previous studies and summarize our results. Formal definitions are provided on the next section.

School of Computer Science, Tel-Aviv University, Tel-Aviv, Israel.

Following the pioneering work by Nadeau and Taylor (1984), reversals and translocations are believed to be very common in the evolution of mammalian species. *Reversals* (or *inversions*) reverse the order and the direction of transcription of the genes in a segment inside a chromosome. *Translocations* exchange tails between two chromosomes. A translocation is *reciprocal* if none of the exchanged tails is empty. The genomic sorting problem where the allowed rearrangement operations are reversals (respectively, reciprocal translocations) is referred to as *sorting by reversals*, hereafter SBR (respectively, *sorting by reciprocal translocations*, hereafter SRT).

Both SBR and SRT use restricted models that allow for a single type of genome rearrangement. Clearly, a model that allows both reversals and translocations is biologically more realistic than each of these two restricted models. Still, the study of sorting by reversals only or by translocations only is of great importance to the understanding of more complex models that allow for several types of genome rearrangements. For example, the problem of sorting by reversals, translocations, fissions, and fusions is reduced to SBR in polynomial time (Hannenhalli and Pevzner, 1995; Ozery-Flato and Shamir, 2003; Tesler, 2002a). In many cases, algorithms for restricted models can be integrated into algorithms for complex models (Ozery-Flato and Shamir, 2006a; Tesler, 2002a).

SBR and SRT were both proven to be polynomial. Hannenhalli and Pevzner (1999) gave the first polynomial algorithm for SBR; since then, other, more efficient algorithms and simplifications of the analysis have been presented. Berman and Hannenhalli (1996) presented a recursive algorithm for SBR. Kaplan et al. (2000) simplified the analysis and gave an $O(n^2)$ algorithm for SBR. Using a linear time algorithm by Bader et al. (2001) for computing the reversal distance, the algorithm of Berman and Hannenhalli can be implemented in $O(n^2)$. A score-based algorithm for SBR was presented by Bergeron (2005). Tannier et al. (2007) presented an elegant algorithm for SBR that can be implemented in $O(n^{3/2}\sqrt{log(n)})$ using a clever data structure due to Kaplan and Verbin (2005).

SRT was first introduced by Kececioglu and Ravi (1995) and was given a polynomial time algorithm by Hannenhalli (1996). Bergeron et al. (2006a) pointed to an error in Hannenhalli's proof of the reciprocal translocation distance formula and consequently in Hannenhalli's algorithm. They presented a new proof and gave an $O(n^3)$ algorithm for SRT. Recently, we (Ozery-Flato and Shamir, 2006a) proved that the algorithm of Tannier et al. (2007) for SBR can be adapted to solve SRT in $O(n^{3/2}\sqrt{log(n)}))$ time.

Can the rich theory on SBR be used to solve SRT? It is well known that a translocation on a multi-chromosomal genome can be simulated by a reversal on a concatenation of the chromosomes (Hannenhalli and Pevzner, 1995). However, different translocations require different concatenations. In addition, intra-chromosomal reversals do not have matching translocations. Last but not least, the formulas of the reversal distance and the reciprocal translocation distance are different. They differ in particular in the parameters that concern difficult structures for SBR/SRT, which are sometimes referred to as "bad components."[1] Thus, from a first glance the similarity between SRT and SBT is rather superficial.

In Ozery-Flato and Shamir (2006a) we introduced a new auxiliary graph for the analysis of SRT (the "overlap graph with chromosomes" of two multi-chromosomal genomes, an extension of the "overlap graph" of two uni-chromosomal genomes) and used it to adapt the fastest extant algorithm for SBR to SRT (Ozery-Flato and Shamir, 2006a; Tannier et al., 2007). In this paper we reveal new relationships between SRT and SBR. Based on these relationships we develop two new algorithms for SRT, which mimic known algorithms for SBR: a score-based method building on Bergeron's algorithm (2005) and a recursive procedure similar to the Berman and Hannenhalli (1996) method. Though the proofs of the algorithms are more involved than those of their counterparts for SBR, our procedures for translocations match the complexities of the original ones for reversals: the score-based algorithm performs $O(n^2)$ operations on $O(n)$-long bit vectors; the recursive algorithm runs in $O(n^2)$ time.

The paper is organized as follows. Section 2 gives the necessary preliminaries. Section 3 presents the score-based algorithm and Section 4 presents the recursive algorithm. Related genomic sorting problems, as well as possible applications of our results and future research problems, are discussed in Section 5.

---

[1] *Hurdles* (Hannenhalli and Pevzner, 1999; Kaplan et al., 2000) for SBR, *leaves* (Bergeron et al., 2006a) (equivalently, *minimal sub-permutations* [Hannenhalli, 1996]), for SRT.

2. PRELIMINARIES

This section provides a basic background for the analysis of SRT. We follow to a large extent the nomenclature and notation of Hannenhalli (1996) and Kaplan et al. (2000). In the model we consider, a *genome* is a set of chromosomes. A *chromosome* is a sequence of genes. A *gene* is identified by a positive integer. All genes in the genome are distinct. When it appears in a genome, a gene is assigned a sign of plus or minus. For example, the following genome consists of 8 genes in two chromosomes:

$$A_1 = \{(1, -3, -2, 4, -7, 8), (6, 5)\}.$$

The *reverse* of a sequence of genes $I = (x_1, \ldots, x_l)$ is $-I = (-x_l, \ldots, -x_1)$. A *reversal* reverses a segment of genes inside a chromosome. Two chromosomes, X and Y, are *identical* if either $X = Y$ or $X = -Y$. Therefore, *flipping* chromosome X into $-X$ does not affect the chromosome it represents.

A *signed permutation* $\pi = (\pi_1, \ldots, \pi_n)$ is a permutation on the integers $\{1, \ldots, n\}$, where a sign of plus or minus is assigned to each number. If A is a genome with the set of genes $\{1, \ldots, n\}$ then any concatenation π_A of the chromosomes of A is a signed permutation of size n. In the following, we assume for simplicity and without loss of generality that there is a concatenation π_B of the chromosomes in the target genome B which is the identity permutation. For example,

$$B = \{(1, 2, \ldots, 5), (6, 7, 8)\}.$$

Let $X = (X_1, X_2)$ and $Y = (Y_1, Y_2)$ be two chromosomes, where X_1, X_2, Y_1, Y_2 are sequences of genes. A *translocation* cuts X into X_1 and X_2 and Y into Y_1 and Y_2 and exchanges segments between the chromosomes. It is called *reciprocal* if X_1, X_2, Y_1 and Y_2 are all non-empty. There are two ways to perform a translocation on X and Y. A *prefix-suffix* translocation switches X_1 with Y_2 resulting in:

$$(\underline{X_1}, X_2), (Y_1, \underline{Y_2}) \Rightarrow (-Y_2, X_2), (Y_1, -X_1).$$

A *prefix-prefix* translocation switches X_1 with Y_1 resulting in:

$$(\underline{X_1}, X_2), (\underline{Y_1}, Y_2) \Rightarrow (Y_1, X_2), (X_1, Y_2).$$

Note that we can mimic a prefix-prefix (respectively, prefix-suffix) translocation by a flip of one of the chromosomes followed by a prefix-suffix (respectively, prefix-prefix) translocation. As was observed by Hannenhalli and Pevzner (1995), a translocation on A can be simulated by a reversal on π_A in the following way:

$$(\ldots, X_1, \underline{X_2, \ldots, Y_1}, Y_2, \ldots) \Rightarrow (\ldots, X_1, \underline{-Y_1, \ldots, -X_2}, Y_2, \ldots).$$

The type of translocation depends on the relative orientation of X and Y in π_A (and not on their order): if the orientation is the same, then the translocation is prefix-suffix, otherwise it is prefix-prefix. The segment between X_2 and Y_1 may contain additional chromosomes that are flipped and thus unaffected.

For an interval of genes $I = (i_1, \ldots, i_k)$ define $\mathit{Tails}(I) = \{i_1, -i_k\}$. Note that $\mathit{Tails}(I) = \mathit{Tails}(-I)$. For a genome A_1 define $\mathit{Tails}(A_1) = \cup_{X \in A_1} \mathit{Tails}(X)$. For example:

$$\mathit{Tails}(\{(1, -3, -2, 4, -7, 8), (6, 5)\}) = \{1, -8, 6, -5\}.$$

Two genomes A_1 and A_2 are called *co-tailed* if $\mathit{Tails}(A_1) = \mathit{Tails}(A_2)$. In particular, two co-tailed genomes have the same number of chromosomes. Note that if A_2 was obtained from A_1 by performing a reciprocal translocation then $\mathit{Tails}(A_2) = \mathit{Tails}(A_1)$. Therefore, SRT is defined only for genomes that are co-tailed. For the rest of this paper, the word "translocation" refers to a reciprocal translocation, and we assume that the given genomes, A and B, are co-tailed.

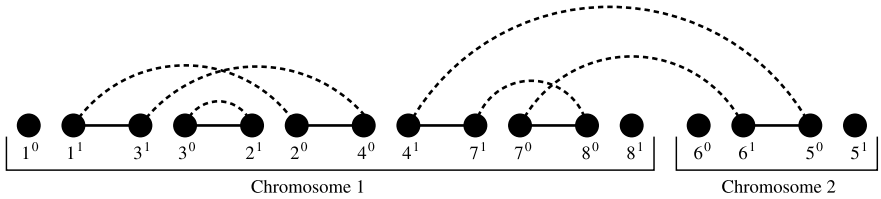


FIG. 1. The cycle graph $G(A_1, B_1)$, where $A_1 = \{(1, -3, -2, 4, -7, 8), (6, 5)\}$ and $B_1 = \{(1, \ldots, 5), (6, 7, 8)\}$. Dotted lines correspond to gray edges. The gray edge $(1, 2)$ is internal, whereas $(4, 5)$ is external. $(2, 3)$ is an adjacency.

2.1. The cycle graph

Let N be the number of chromosomes in A (equivalently, B). We shall always assume that both A and B contain genes $\{1, \ldots, n\}$. The *cycle graph* of A and B, denoted $G(A, B)$, is defined as follows. The set of vertices is $\cup_{i=1}^{n}\{i^0, i^1\}$. For every pair of adjacent genes in B, i and $i + 1$, add a gray edge $(i, i + 1) \equiv (i^1, (i + 1)^0)$. For every pair of adjacent genes in A, i and j, add a black edge $(i, j) \equiv (out(i), in(j))$, where $out(i) = i^1$ if i has a positive sign in A and otherwise $out(i) = i^0$, and $in(j) = j^0$ if j has a positive sign in A and otherwise $in(j) = j^1$. An example is given in Figure 1. There are $n - N$ black edges and $n - N$ gray edges in $G(A, B)$. A gray edge $(i, i + 1)$ is *external* if the genes i and $i + 1$ belong to different chromosomes of A, otherwise it is *internal*.

Every vertex in $G(A, B)$ has degree 2 or 0, where vertices of degree 0 (isolated vertices) belong to *Tails*(A) (equivalently, *Tails*(B)). Therefore, $G(A, B)$ is uniquely decomposable into cycles with alternating gray and black edges. An *adjacency* is a cycle with two edges.

2.2. The overlap graph with chromosomes

Place the vertices of $G(A, B)$ along a straight line according to their order in π_A. Now, every gray edge can be associated with an interval of vertices of $G(A, B)$. Two intervals *overlap* if their intersection is not empty but neither contains the other. The *overlap graph with chromosomes* of A and B w.r.t. π_A, denoted $\Omega(A, B, \pi_A)$, is defined as follows. There are two types of nodes. The first type corresponds to gray edges in $G(A, B)$. The second type corresponds to chromosomes of A. Two nodes are connected if their associated intervals overlap (Fig. 2). For the rest of this paper we will refer to overlap graphs with chromosomes as Ω-graphs.

In order to avoid confusion, we will refer to nodes that correspond to chromosomes as "chromosomes" and reserve the word "vertex" for the nodes that correspond to gray edges of $G(A, B)$. Observe that a vertex in $\Omega(A, B, \pi_A)$ is external iff there is an edge connecting it to a chromosome. Note that the internal/external state of a vertex in $\Omega(A, B, \pi_A)$ does not depend on π_A (the partition of the chromosomes is known from A). A vertex in $\Omega(A, B, \pi_A)$ is *oriented* if its corresponding edge connects two genes with different signs in π_A, otherwise it is *unoriented*.

Let $\mathrm{OV}(A, B, \pi_A)$ be the subgraph of $\Omega(A, B, \pi_A)$ induced by the set of nodes that correspond to gray edges (i.e., excluding the chromosomes' nodes). We shall use the word "component" for a connected component of $\mathrm{OV}(A, B, \pi_A)$. A component is *external* if at least one of the vertices in it is external, otherwise it is *internal*. A component is *trivial* if it is composed of one internal vertex. A trivial component

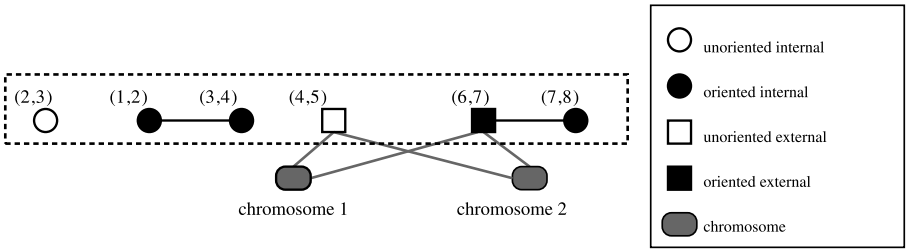


FIG. 2. The overlap graph with chromosomes $\Omega(A_1, B_1, \pi_{A_1})$, where A_1 and B_1 are the genomes from Figure 1 and $\pi_{A_1} = (1, -3, -2, 4, -7, 8, 6, 5)$. The graph induced by the vertices within the dashed rectangle is $\mathrm{OV}(A_1, B_1, \pi_{A_1})$.

corresponds to an adjacency. The *span* of a component M is the minimal interval of genes $I(M) = [i, j] \subset \pi_A$ that contains the interval of every vertex in M. If the spans of two components intersect then either they overlap by at most gene, or one span contains the other. Clearly, $I(M)$ is independent of π_A iff M is internal. Thus the set of internal components in $\Omega(A, B, \pi_A)$ is independent of π_A. Denote by $\mathcal{IN}(A, B)$ the set of non-trivial internal components in $\Omega(A, B, \pi_A)$. The following lemma follows from the definition of "sub-permutations" in Hannenhalli (1996):

Lemma 1. *Suppose I is the span of an internal component. Then the genes of I form a continuous interval I' in one of the chromosomes of B and Tails(I) = Tails(I').*

2.3. The reciprocal translocation distance

Let $c(A, B)$ denote the number of cycles in $G(A, B)$.

Theorem 1 (Bergeron et al., 2006a; Hannenhalli, 1996). *The reciprocal translocation distance between A and B is $d(A, B) = n - N - c(A, B) + F(A, B)$, where $F(A, B) \geq 0$ and $F(A, B) = 0$ iff $\mathcal{IN}(A, B) = \emptyset$.*

Let Δc denote the change in the number of cycles after performing a translocation on A. Then $\Delta c \in \{-1, 0, 1\}$ (Hannenhalli, 1996). A translocation is *proper* if $\Delta c = 1$. A translocation is *safe* if it does not create any new non-trivial internal component. A translocation ρ is *valid* if $d(A \cdot \rho, B) = d(A, B) - 1$. It follows from Theorem 1 that if $\mathcal{IN}(A, B) = \emptyset$, then every safe proper translocation is necessarily valid.

In a previous study (Ozery-Flato and Shamir, 2006a), we presented a generic algorithm for SRT that uses a sub-procedure for solving SRT when $\mathcal{IN}(A, B) = \emptyset$. The algorithm focuses on the efficient elimination of the non-trivial internal components. We showed that the work performed by this generic algorithm, not including the sub-procedure calls, can be implemented in linear time. This led to the following theorem:

Theorem 2 (Ozery-Flato and Shamir, 2006a). *SRT is linearly reducible to SRT with $\mathcal{IN}(A, B) = \emptyset$.*

By the theorem above, it suffices to solve SRT assuming that $\mathcal{IN}(A, B) = \emptyset$. Both algorithms that we describe below will make this assumption.

2.4. The effect of a translocation on the overlap graph with chromosomes

Let $\pi_{CH} \equiv \pi_{CH}(A, \pi_A)$ be the linear order of the chromosomes in A, as defined by π_A. Slightly abusing terminology, we extend the definition of the Ω-graph to include π_{CH}. In other words, an Ω-graph carries also a permutation of its chromosome nodes defined by π_A. Two chromosomes in $\Omega(A, B, \pi_A)$ are called *consecutive* if they are consecutive in π_{CH}.

Let $H = \Omega(A, B, \pi_A)$ and let v be any vertex in H. Denote by $N(v) \equiv N(v, H)$ the set of vertices that are neighbors of v in H, including v itself (but not including chromosome neighbors). Denote by $CH(v) \equiv CH(v, H)$ the set of chromosomes that are neighbors of v in H. Clearly, if v is external then $|CH(v)| = 2$, otherwise $CH(v) = \emptyset$.

Every external gray edge e defines one proper translocation that cuts the black edges incident to e. (Out of the two possibilities of prefix-prefix or prefix-suffix translocations, exactly one would be proper.) For an external vertex v denote by $\rho(v)$ the proper translocation that the corresponding gray edge defines on A. If v is an oriented external vertex then $\rho(v)$ can be mimicked by a reversal $\hat{\rho}(v)$ on π_A. For an oriented external vertex v define $H \cdot \rho(v) = \Omega(A \cdot \rho(v), B, \pi_A \cdot \hat{\rho}(v))$. The following two lemmas refine claims in Ozery-Flato and Shamir (2006a).

Lemma 2. *Let v be an oriented external vertex in H and suppose the chromosomes in $CH(v)$ are consecutive. Then $H \cdot \rho(v)$ is obtained from H by the following operations. (i) Complement the subgraph induced by $N(v)$ and flip the orientation of every vertex in $N(v)$. (ii) For every vertex $u \in N(v)$ complement the edges between u and $CH(u) \cup CH(v)$. In particular, the external/internal state of a vertex $u \in N(v)$ is flipped iff u is internal or $CH(u) = CH(v)$.*

Proof. The correctness of *(i)* follows immediately from Observation 4.1 in Kaplan et al. (2000). To prove *(ii)*, let $u \in N(v)$. Since the chromosomes in $CH(v)$ are consecutive, u is either internal or $|CH(u) \cap CH(v)| \in \{1, 2\}$. In each of these cases, $CH(u)$ is complemented w.r.t. $CH(u) \cup CH(v)$ (for illustration, see Fig. 3). Suppose $w \notin N(v)$. Let I_v and I_w be the intervals associated with v and w respectively (see Section 2.2). Then there are three possible cases:

Case 1: $I_w \subset I_v$ and w is internal. Then I_w is contained entirely in one of the exchanged segments. Thus w remains internal and hence $CH(w, H \cdot \rho(v)) = CH(w, H) = \emptyset$.

Case 2: $I_w \subset I_v$ and w is external. Then $CH(w, H) = CH(v, H)$ and the two endpoints of I_w exchange their chromosomes after $\rho(v)$ is performed. Thus $CH(w, H \cdot \rho(v)) = CH(w, H)(= CH(v, H))$.

Case 3: $I_w \cap I_v = \emptyset$ or $I_v \subset I_w$. In these two cases the endpoints of I_w are not affected by $\rho(v)$ and hence $CH(w, H \cdot \rho(v)) = CH(w, H)$. ■

We shall sometimes need to change the chromosome order or flip a chromosome. These operations can be mimicked by reversals on π_A but do not correspond to translocations, and thus are not covered by Lemma 2. For an interval of chromosomes $I \subset \pi_A$, let $\hat{\rho}(I)$ denote the flip, i.e., reversal, of I in π_A. Let $H \cdot \rho(I) = \Omega(A, B, \pi_A \cdot \hat{\rho}(I))$.

Lemma 3. *For an interval of chromosomes $I \subset \pi_A$, $H \cdot \rho(I)$ is obtained from H by the following operations. (i) Reverse the order of the chromosomes in I. (ii) Complement the subgraph induced by the set $\{v$: exactly one of the chromosomes in $CH(v)$ is contained in $I\}$, and flip the orientation of every vertex in it. In particular, if I is a single chromosome of A then $H \cdot \rho(I)$ is obtained by complementing the subgraph induced by the neighbors of I in H, and flipping the orientation of every vertex in it.*

Proof. The vertices affected by $\rho(I)$ are the ones that overlap I. A vertex v overlaps I iff exactly one of its endpoints belong to I (hence it must be external). The rest of the proof follows directly from Observation 4.1 in Kaplan et al. (2000). ■

We refer to two Ω-graphs of the same pair of genomes A and B, irrespective of the concatenation π_A, as *equivalent*. Clearly, we can transform an Ω-graph to any other equivalent graph by a sequence of flips of chromosomes intervals, as defined by Lemma 3.

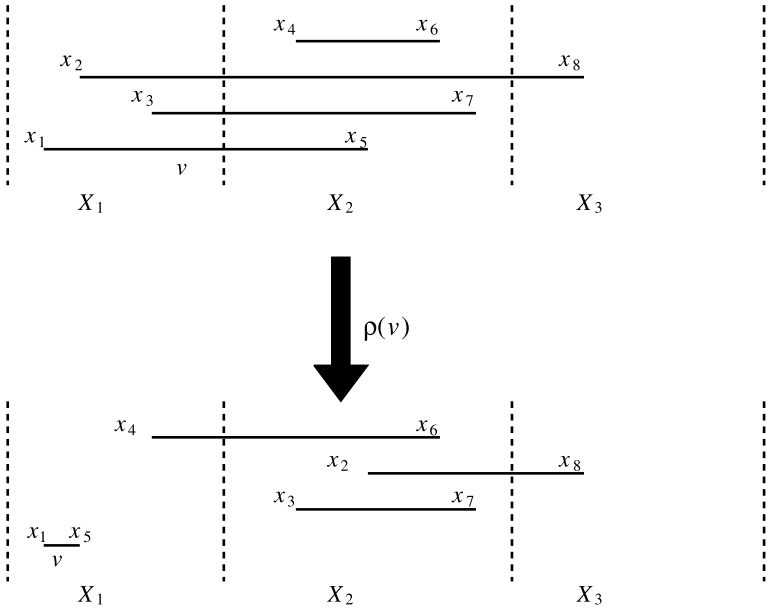


FIG. 3. The effect of performing a translocation, mimicked by a reversal, on overlapping intervals. X_1, X_2, and X_3 are chromosomes, and the dashed lines denote the borders between them in the concatenation (X_1, X_2, X_3). The letters $x_1, \ldots, x_8$ denote the endpoints of the intervals (the endpoints are vertices of the cycle graph). The interval v corresponds to an (external) edge on which a translocation is performed.

Observation 1. *Let H and H' be two equivalent graphs in which v is an oriented external vertex. Then the set of internal components is the same for $H \cdot \rho(v)$ and $H' \cdot \rho(v)$.*

Proof. We can transform $H \cdot \rho(v)$ into $H' \cdot \rho(v)$ by a sequence of flips of chromosomes intervals. By Lemma 3, a flip of an interval of chromosomes does not change the internal/external state of any vertex, and does not affect the neighborhood of any internal vertex. Thus $H \cdot \rho(v)$ and $H' \cdot \rho(v)$ must have the same set of internal components. ■

Let v be an external vertex in H, and let H' be an equivalent graph to H in which v is oriented, possibly $H = H'$ if v is already oriented in H. A key definition that will be crucial throughout the paper is the following: $\Delta\text{IN}(H, v)$ is the set of vertices that belong to external components in H (equivalently, H') but are in non-trivial internal components in $H' \cdot \rho(v)$. By Observation 1, if *(i)* v is an external vertex in H, and *(ii)* H' is equivalent to H, then $\Delta\text{IN}(H, v) = \Delta\text{IN}(H', v)$. It follows that in order to compute $\Delta\text{IN}(H, v)$, we can assume without loss of generality that v is oriented and the chromosomes in $CH(v)$ are consecutive. As we shall see, the additional work required to satisfy this assumption will not change the overall complexity of the algorithms.

3. A SCORE-BASED ALGORITHM

In this section, we present a score-based algorithm for SRT when $\mathcal{IN}(A, B) = \emptyset$. This algorithm is similar to an algorithm by Bergeron (2005) for SBR. Denote by $N_{\text{IN}}(v)$ and $N_{\text{EXT}}(v)$ the neighbors of v that are respectively internal and external. It follows that $N_{\text{IN}}(v) \cup N_{\text{EXT}}(v) \cup \{v\} = N(v)$. For two chromosomes X and Y, let $V_{XY} = \{v : CH(v) = \{X, Y\}\}$.

Lemma 4. *Let X and Y be two consecutive chromosomes in $H = \Omega(A, B, \pi_A)$. Suppose $v \in V_{XY}$ is oriented. Let $w \in N(v)$. If w has no external neighbors in $H \cdot \rho(v)$ then $N_{\text{EXT}}(w) \subseteq N_{\text{EXT}}(v)$ and $N_{\text{IN}}(v) \subseteq N_{\text{IN}}(w)$.*

Proof. It follows from Lemma 2 that if $u \in (N_{\text{EXT}}(w) \setminus N_{\text{EXT}}(v)) \cup (N_{\text{IN}}(v) \setminus N_{\text{IN}}(w))$ then u is an external neighbor of w in $H \cdot \rho(v)$. ■

For each vertex v in $H = \Omega(A, B, \pi_A)$ we define the *score* of v as $|N_{\text{IN}}(v)| - |N_{\text{EXT}}(v)|$. The following lemma lays the basis for the score-based approach and is used by the implementation of the recursive algorithm as well.

Lemma 5. *Let X and Y be two consecutive chromosomes in $H = \Omega(A, B, \pi_A)$ for which $V_{XY} \neq \emptyset$. Let $O \subset V_{XY}$ be a set of oriented (external) vertices and suppose $O \neq \emptyset$. Let $v \in O$ be a vertex with a maximal score in H. Then $O \cap \Delta\text{IN}(H, v) = \emptyset$.*

Proof. Assume $u \in O \cap \Delta\text{IN}(H, v)$. Then $u \in N(v, H)$, and by Lemma 4 $N_{\text{EXT}}(u) \subseteq N_{\text{EXT}}(v)$ and $N_{\text{IN}}(v) \subseteq N_{\text{IN}}(u)$. However, since v has the maximal score in O, we get $N_{\text{EXT}}(u) = N_{\text{EXT}}(v)$ and $N_{\text{IN}}(v) = N_{\text{IN}}(u)$. Therefore, $N(u) = N(v)$, and by Lemma 2 it follows that u is an isolated internal vertex in $H \cdot \rho(v)$, a contradiction to the assumption that $u \in \Delta\text{IN}(H, v)$. ■

Theorem 3. *Let X and Y be two consecutive chromosomes in $H = \Omega(A, B, \pi_A)$. Let O be the set of all the oriented external vertices in V_{XY} and suppose $O \neq \emptyset$. Let $v \in O$ be a vertex that has the maximal score in H. Let $S = S(v)$ be the set of all the vertices w that satisfy the following conditions in H:*

1. *w is a neighbor of v,*
2. *w is an unoriented external vertex and $CH(w) = CH(v)$,*
3. *$N_{\text{EXT}}(w) \subseteq N_{\text{EXT}}(v)$,*
4. *$N_{\text{IN}}(v) \subseteq N_{\text{IN}}(w)$, and*
5. *$O \cap N(v) \subseteq N_{\text{EXT}}(w)$.*

If $S = \emptyset$ then $\rho(v)$ is safe. Otherwise, let $w \in S$ be a vertex that has a maximal score in $H \cdot \rho(X)$, where $X \in CH(v)$. Then $\rho(w)$ is safe.

Proof. Suppose $S = \emptyset$ and assume v is unsafe. Let $w \in \Delta\text{IN}(H, v)$ be a neighbor of v in H. w satisfies conditions 3 and 4 by Lemma 4, it is external and $CH(w) = CH(v)$, by Lemma 2. It follows from Lemma 5 that $O \cap \Delta\text{IN}(H, v) = \emptyset$. Hence w is unoriented in H and the last condition is satisfied (otherwise w has a neighbor from O in $H \cdot \rho(v)$, in contradiction to the choice of $w \in \Delta\text{IN}(H, v)$). It follows that $w \in S$, a contradiction.

Suppose $S \neq \emptyset$. Let $H' = H \cdot \rho(X)$, where $X \in CH(v)$. Let $w \in S$ be a vertex with maximal score in H'. We prove below that if $\Delta\text{IN}(H', w) \neq \emptyset$ then $\Delta\text{IN}(H', w) \cap S \neq \emptyset$, in contradiction to Lemma 5.

Let $O_1 = O \cap N(v)$ in H. Then in H' : *(i)* all the vertices in S are oriented (condition 2), *(ii)* O_1 contains all the unoriented external vertices with $CH = CH(v)$ that are not neighbors of v, and *(iii)* there are no edges between S and $O_1 \cup \{v\}$ (condition 5). It follows that each vertex in $O_1 \cup \{v\}$ remains external after performing a translocation on any vertex in S.

Assume that $\Delta\text{IN}(H', w) \neq \emptyset$. Let $u \in \Delta\text{IN}(H', w)$ be a neighbor of w in H'. We shall prove that $u \in S$. Clearly, u is an external vertex in H' and $CH(u) = CH(w) = CH(v)$. Since all the vertices in $O_1 \cup \{v\}$ are external and there are no edges between them and w in H', $u \notin O_1 \cup \{v\}$ and there are no edges between u and $O_1 \cup \{v\}$ in H' (Lemma 4). Since all the unoriented vertices that are not neighbors of v belong to O_1, u must be oriented. It follows that in H, u satisfies conditions 1, 2 and 5. We now prove that u satisfies conditions 3 and 4 in H as well, thus $u \in S$—a contradiction to Lemma 5.

Suppose u does not satisfy condition 4 in H. Let $x \in N_{\text{IN}}(v) \setminus N_{\text{IN}}(u)$ in H. Since w satisfies condition 4 in H, $x \in N_{\text{IN}}(w) \setminus N_{\text{IN}}(u)$ in H. Since x is internal, all its edges are the same in H and H'. Hence $x \in N_{\text{IN}}(w) \setminus N_{\text{IN}}(u)$ in H'. It follows from Lemma 4 that u has an external neighbor (x) in $H' \cdot \rho(w)$, a contradiction to $u \in \Delta\text{IN}(H', w)$. Thus u must satisfy condition 4 in H.

Suppose u does not satisfy condition 3 in H. Let $z \in N_{\text{EXT}}(u) \setminus N_{\text{EXT}}(v)$ in H.

Case 1: $X \notin CH(z)$. Since w satisfies condition 3, $z \in N_{\text{EXT}}(u) \setminus N_{\text{EXT}}(w)$ in H. Then in H': $z \in N_{\text{EXT}}(u) \setminus N_{\text{EXT}}(w)$ (Lemma 3). Then according to Lemma 4, u has an external neighbor (z) in $H' \cdot \rho(w)$, a contradiction to $u \in \Delta\text{IN}(H', w)$.

Case 2: $X \in CH(z)$. In H: since w satisfies condition 3 and $z \notin N_{\text{EXT}}(v)$ then $z \notin N_{\text{EXT}}(w)$. Thus in H': $z \notin N(u)$, $z \in N(v) \cap N(w)$ (Lemma 3). Therefore, in $H' \cdot \rho(w)$ the path u, z, v exists (Lemma 2), a contradiction to $u \in \Delta\text{IN}(H', w)$ (since v is external in $H' \cdot \rho(w)$). ■

Theorem 3 immediately implies the following polynomial time algorithm (Algorithm 1) for finding a safe proper translocation using $H = \Omega(A, B, \pi_A)$:

Algorithm 1. *Find_Safe_Translocation_Using_Scores* (H)

---

1. Let X and Y be two chromosomes for which there exists a common adjacent (external) vertex u.
2. Flip chromosomes, if necessary, to make X and Y consecutive and to make u oriented.
3. Let $v \in V_{XY}$ be an oriented (external) vertex with a maximal score.
4. Compute the set of vertices $S(v)$ defined by Theorem 3.
5. If $S(v) = \emptyset$ then return $\rho(v)$.
6. Otherwise,
   a. Flip chromosome X or Y, and recalculate the score of the vertices.
   b. Let $w \in S(v)$ be a vertex with a maximal score.
   c. Return $\rho(w)$.

The above algorithm can be implemented in $O(n^2)$ time using $O(n)$ operations on $O(n)$-long bit vectors, in a similar manner to the implementation of the algorithm of Bergeron (2005) for SBR. The implementation is presented in Figure 4 and uses the following notations. The symbols $\boldsymbol{v}$, $\boldsymbol{X}$, ***ext*** and

1. Let u be an external vertex. Suppose $CH(u) = \{X, Y\}$. *// $V_{XY} \neq \emptyset$*
2. If X and Y are not consecutive:
   (a) Let $X_1, \ldots, X_k$ be the chromosomes between X and Y (not including X and Y). Let $\boldsymbol{Z} = \boldsymbol{X_1} \bigoplus \cdots \bigoplus \boldsymbol{X_k} \bigoplus \boldsymbol{X}$
   (b) for every u satisfying $\boldsymbol{Z}[u] = 1$:
      i. $\boldsymbol{score} \leftarrow \boldsymbol{score} + \boldsymbol{u}$
      ii. $\boldsymbol{u} \leftarrow \boldsymbol{u} \bigoplus \boldsymbol{Z}$, $\boldsymbol{o}[u] = \neg \boldsymbol{o}[u]$ *// subgraph completion*
      iii. $\boldsymbol{score} \leftarrow \boldsymbol{score} - \boldsymbol{u}$

   // Flip X, if needed, so there is at least one oriented vertex in V_{XY}.
3. If $\boldsymbol{X} \bigwedge \boldsymbol{Y} \bigwedge \boldsymbol{o} = 0$ then flip chromosome X (repeat step 2.b where $\boldsymbol{Z} = \boldsymbol{X}$).
4. Choose a vertex v such that $\boldsymbol{X} \bigwedge \boldsymbol{Y} \bigwedge \boldsymbol{o}[v] = 1$ and $\boldsymbol{score}[v]$ is maximal.
5. Build the vector $\boldsymbol{S}$: $\boldsymbol{S}[w] \leftarrow 1$ if the following conditions hold:
   - $(\boldsymbol{X} \bigwedge \boldsymbol{Y} \bigwedge \boldsymbol{v} \bigwedge \neg \boldsymbol{o})[w] = 1$ *// conditions 1 and 2*
   - $(\boldsymbol{w} \bigwedge \boldsymbol{ext}) \bigvee \boldsymbol{v} = \boldsymbol{v}$ *// condition 3*
   - $(\boldsymbol{v} \bigwedge \neg \boldsymbol{ext}) \bigvee \boldsymbol{w} = \boldsymbol{w}$ *// condition 4*
   - $(\boldsymbol{v} \bigwedge \boldsymbol{X} \bigwedge \boldsymbol{Y} \bigwedge \boldsymbol{o}) \bigvee \boldsymbol{w} = \boldsymbol{w}$ *// condition 5*
6. If $\boldsymbol{S} \neq 0$:
   a. Flip X (repeat step 2.b where $\boldsymbol{Z} = \boldsymbol{X}$).
   b. Choose v such that $\boldsymbol{S}[v] = 1$ and $\boldsymbol{score}[v]$ is maximal.

   // Perform $\rho(v)$
7. $\boldsymbol{score} \leftarrow \boldsymbol{score} + \boldsymbol{v}$, $\boldsymbol{v}[v] = 1$
8. For every vertex u adjacent to v: *// i.e. $\boldsymbol{v}[u] = 1$*
   a. If $\boldsymbol{ext}[u] = 1$ then $\boldsymbol{score} \leftarrow \boldsymbol{score} + \boldsymbol{u}$. Otherwise $\boldsymbol{score} \leftarrow \boldsymbol{score} - \boldsymbol{u}$
   b. $\boldsymbol{u}[u] = 1$, $\boldsymbol{u} \leftarrow \boldsymbol{u} \bigoplus \boldsymbol{v}$, $\boldsymbol{o}[u] = \neg \boldsymbol{o}[u]$ *// subgraph completion*

      // Update vectors $\boldsymbol{ext}$, $\boldsymbol{X}$ and $\boldsymbol{Y}$.
   c. If $\boldsymbol{X}[u] + \boldsymbol{Y}[u] \neq 1$ then $\boldsymbol{ext}[u] = \neg \boldsymbol{ext}[u]$
   d. $\boldsymbol{X}[u] = \neg \boldsymbol{X}[u]$, $\boldsymbol{Y}[u] = \neg \boldsymbol{Y}[u]$

      // Update vector $\boldsymbol{score}$.
   e. If $\boldsymbol{ext}[u] = 1$ then $\boldsymbol{score} \leftarrow \boldsymbol{score} - \boldsymbol{u}$. Otherwise $\boldsymbol{score} \leftarrow \boldsymbol{score} + \boldsymbol{u}$

FIG. 4. An $O(n^2)$ implementation of Algorithm 1 using $O(n)$-long bit vectors.

$\boldsymbol{o}$ represent bit vectors of size $n - N$. The vector $\boldsymbol{v}$ corresponds to the vertex v, where $\boldsymbol{v}[u] = 1$ iff u is a neighbor of v. The vector $\boldsymbol{X}$ corresponds to chromosome X, where $\boldsymbol{X}[v] = 1$ iff $X \in CH(v)$. The chromosome vectors are ordered according to their order in π_A. The vectors ***ext*** and $\boldsymbol{o}$ correspond to the sets of external vertices and oriented vertices respectively. In other words, $\boldsymbol{ext}[u] = 1$ iff u is external, $\boldsymbol{o}[u] = 1$ iff u is oriented. The score of each vertex is stored in an integer vector ***score***. The symbols $\wedge$, $\vee$, $\oplus$ and $\neg$ respectively denote the bitwise-AND, bitwise-OR, bitwise-XOR and bitwise-NOT operators. Steps 1–6 in the algorithm in Figure 4 locate a safe proper translocation $\rho(v)$. Steps 7 and 8 perform $\rho(v)$ and update the above vectors.

Corollary 1. *The score-based algorithm solves SRT in $O(n^3)$ time.*

4. A RECURSIVE ALGORITHM

In this section, we present a recursive algorithm for SRT when $\mathcal{IN}(A, B) = \emptyset$. This algorithm is similar to the algorithm of Berman and Hannenhalli (1996) for SBR.

4.1. The algorithm

Denote the number of vertices in a graph H by $|H|$. For two chromosomes, X and Y, let O_{XY} (respectively U_{XY}) be the set of oriented (respectively unoriented) vertices in H for which $CH = \{X, Y\}$. Thus $O_{XY} \cup U_{XY} = V_{XY}$.

Theorem 4. *Let $H = \Omega(A, B, \pi_A)$. If H contains an external vertex then it contains an external vertex v for which $\Delta\mathrm{IN}(H, v) \leq \frac{|H|}{2}$.*

Proof. Let X and Y be two chromosomes for which $V_{XY} \neq \emptyset$. Assume w.l.o.g. that X and Y are consecutive and $O_{XY} \neq \emptyset$. Let $v \in O_{XY}$ be a vertex with maximal score in H. If $\Delta\mathrm{IN}(H, v) = \emptyset$ then we are done since $|\Delta\mathrm{IN}(H, v)| = 0 \leq \frac{|H|}{2}$. Suppose $\Delta\mathrm{IN}(H, v) \neq \emptyset$. By Lemma 5, $\Delta\mathrm{IN}(H, v) \cap O_{XY} = \emptyset$. Thus $\Delta\mathrm{IN}(H, v) \cap U_{XY} \neq \emptyset$. Let $H' = H \cdot \rho(X)$ and let $u \in U_{XY}$ be a vertex with maximal score in H'. Let $M_v = \Delta\mathrm{IN}(H, v)$ and $M_u = \Delta\mathrm{IN}(H', u) = \Delta\mathrm{IN}(H, u)$. We shall prove that $M_u \cap M_v = \emptyset$, and hence $\min\{|M_v|, |M_u|\} \leq \frac{|H|}{2}$. Assume $x \in M_v$ and let $x = x_0, \ldots, x_k, x_{k+1} = v$ be a shortest path from x to v in H. Then by Lemma 2, $CH(x_k) = CH(v)$ and $x_0, \ldots, x_{k-1}$ are internal. Hence the path $x_0, \ldots, x_k$ exists in H'. Moreover, $x_k \notin O_{XY}$ since the path $x_0, \ldots, x_k$ exists in $H \cdot \rho(v)$ and $M_v \cap O_{XY} = \emptyset$. Thus $x_k \in U_{XY}$. If none of the vertices in $\{x_0, \ldots, x_k\}$ is in $N(u, H')$ then the path remains intact in $H' \cdot \rho(u)$. Otherwise, let x_j be the first vertex in $x_0, \ldots, x_k$ that is in $N(u, H')$. Thus the path $x_0, \ldots, x_j$ exists in $H' \cdot \rho(v)$. If $x_j \in \{x_0, \ldots, x_{k-1}\}$ then x_j is external in $H' \cdot \rho(u)$. If $x_j = x_k$ then by Lemma 5 $M_u \cap U_{XY} = \emptyset$ and hence $x_k \notin M_u$. Thus in any case $x = x_0 \notin M_u$. ■

Theorem 5. *Let v be an external vertex in $H = \Omega(A, B, \pi_A)$. Suppose $\Delta\mathrm{IN}(H, v) \neq \emptyset$. Let $w \in \Delta\mathrm{IN}(H, v)$ be an external vertex in H. Then $\Delta\mathrm{IN}(H, w) \subset \Delta\mathrm{IN}(H, v)$.*

Proof. Assume w.l.o.g. that the chromosomes in $CH(v)$ are consecutive and v is an oriented (external) vertex in H. By Lemma 2, w is a neighbor of v in H and $CH(v) = CH(w)$ (otherwise it would remain external in $H \cdot \rho(v)$). Let x be a vertex in H such that $x \notin \Delta\mathrm{IN}(H, v)$. It suffices to prove that $x \notin \Delta\mathrm{IN}(H, w)$. Let $x = x_0, \ldots, x_k = y$ be a shortest path from x to an external vertex in $H \cdot \rho(v)$. Then in H: x_j is neighbor of v iff x_j is a neighbor of w, for $j = 1..k$ (otherwise there is a path in $H \cdot \rho(v)$ from w to the external vertex $x_k = y$).

Case 1: w is oriented in H. Then the subgraphs induced by the vertices $\{x_0, \ldots, x_k\}$ in $H \cdot \rho(w)$ and $H \cdot \rho(v)$ are the same. Hence in $H \cdot \rho(w)$: y is external and the path in $x = x_0, \ldots, x_k = y$ exists.

Case 2: w is unoriented in H. In $H \cdot \rho(v)$ the vertices in $\{x_0, \ldots, x_{k-1}\}$ are internal and $x_k (= y)$ is external. Therefore $x_j \in \{x_0, \ldots, x_{k-1}\}$ satisfies in H: *(i)* x_j is a neighbor of v iff x_j is external and $CH(x_j) = CH(w)$, and *(ii)* x_j is not a neighbor of v iff x_j is internal. Denote by H' the graph obtained from H after flipping one of the chromosomes in $CH(w)$.

Case 2.a: At least one vertex in $\{x_0, ..., x_{k-1}\}$ is a neighbor of v in H. Choose $x_j \in \{x_0, \ldots, x_{k-1}\}$ a neighbor of v in H such that $\{x_0, \ldots, x_{j-1}\}$ are not neighbors of v in H. Then in H the following conditions are satisfied: *(i)* $x_0, \ldots, x_j$ is a path, *(ii)* all the vertices in $\{x_0, \ldots, x_{j-1}\}$ are internal and *(iii)* x_j is external satisfying $CH(x_j) = CH(v)$. Therefore in H' the path $x_0, \ldots, x_j$ still exists and none of the vertices in the path is a neighbor of v (equivalently, w). Hence, the path remains intact in $H' \cdot \rho(w)$.

Case 2.b: None of the vertices in $\{x_0, \ldots, x_{k-1}\}$ is a neighbor of v in H. Then the path $x_0, \ldots, x_k$ exists in H'. v is not a neighbor of w in H' hence v remains external in $H' \cdot \rho(w)$. If x_k is a neighbor of v (and w) in H' then the path $x_0, \ldots, x_k, v$ exists in $H' \cdot \rho(w)$ and hence $x = x_0 \notin \Delta\mathrm{IN}(H, w)$. If x_k is not a neighbor of v and w in H' then x_k is necessarily external in H' (equivalently, H). In this case the path $x = x_0, \ldots, x_k = y$ remains intact in $H' \cdot \rho(w)$ and $x = x_0 \notin \Delta\mathrm{IN}(H, w)$. ■

Corollary 2. *Let v be an external vertex in H. Suppose $M = \Delta\text{IN}(H, v) \neq \emptyset$. Let H_M be the subgraph of H induced by the nodes in $M \cup CH(v)$, and let w be an external vertex in H_M. Then $\Delta\text{IN}(H, w) \subseteq \Delta\text{IN}(H_M, w)$. In particular, if $\Delta\text{IN}(H_M, w) = \emptyset$ then $\Delta\text{IN}(H, w) = \emptyset$.*

Proof. We assume w.l.o.g. that the chromosomes in $CH(w)$ are consecutive and w is oriented in H. Then $H_M \cdot \rho(w)$ is identical to the subgraph induced by $M \cup CH(v)$ in $H \cdot \rho(w)$. It follows that every component in $H \cdot \rho(w)$ contained in M is also a component of $H_M \cdot \rho(w)$. By Theorem 5 every internal component in $H \cdot \rho(w)$ is contained in M. Thus $\Delta\text{IN}(H, w) \subseteq \Delta\text{IN}(H_M, w)$. ■

The two theorems above are correct for any subgraph H' of $\Omega(A, B, \pi_A)$ that is induced by a set of vertices and their adjacent chromosomes. By recursive use of Theorem 4 and Corollary 2 we get the following algorithm for locating a safe proper translocation. Algorithm 2 receives $H = \Omega(A, B, \pi_A)$ as an input.

Algorithm 2. *Find_Safe_Translocation_Recursive* (H)

---

1. Choose v from H satisfying $\Delta\text{IN}(H, v) \leq \frac{|H|}{2}$, according to the proof of Theorem 4.
2. $M \leftarrow \Delta\text{IN}(H, v)$
3. If $M \neq \emptyset$:
   a. $H_M \leftarrow$ the subgraph of H induced by $M \cup CH(v)$
   b. $\rho(v) \leftarrow$ *Find_Safe_Translocation_Recursive*(H_M)
4. Return $\rho(v)$

4.2. A linear time implementation

We shall now prove that Algorithm *Find_Safe_Translocation_Recursive* can be implemented in linear time. We shall use an algorithm of Bader et al. (2001) for the computation of $\Delta\text{IN}(H, v)$. We shall use an algorithm by Kaplan et al. (2000) for locating an external vertex v satisfying $|\Delta\text{IN}(H, v)| \leq \frac{|H|}{2}$. A difficulty in trying to apply these algorithms is that they operate on signed permutations and not on Ω-graphs. To overcome this, the algorithm will be initially called with genomes A and B. Before every recursive call it will build two appropriate co-tailed genomes A_M and B_M and pass them as arguments to the recursive call instead of H_M.

Assume w.l.o.g. that there are no adjacencies in $G(A, B)$ (otherwise, every maximal run of adjacencies can be replaced by one element in both A and B). Thus $G(A, B)$ contains no internal components.

4.2.1. Computing $\Delta\text{IN}(H, v)$ in linear time. We apply the translocation $\rho(v)$ on A, and then compute the set of non-trivial internal components. Suppose we want to compute the set of non-trivial internal components in $\Omega(A, B, \pi_A)$. We compute the set of components in $\text{OV}(\pi_A)$ in linear time, using an algorithm by Bader et al. (2001). The output of this algorithm contains the set of components of $\text{OV}(\pi_A)$ along with the span of each. The graph $\text{OV}(\pi_A)$ contains additional vertices that are not in $\Omega(A, B, \pi_A)$. These additional vertices correspond to edges between tails of B. Since A and B are co-tailed, the neighbors of these vertices in $\text{OV}(\pi_A)$ are all external. Therefore the removal of these additional vertices does not affect the set of internal components in this graph. A component is internal iff the two endpoints of its span belong to the same chromosome of A. An internal component is non-trivial if its span contains more than two elements.

4.2.2. Finding an external vertex v satisfying $|\Delta\text{IN}(H, v)| \leq \frac{|H|}{2}$ in linear time. Let X and Y be two chromosomes that contain the endpoints of an external edge v. Build a concatenation π_A in which X and Y are consecutive. Let $H = \Omega(A, B, \pi_A)$ and let $H' = H \cdot \rho(X)$. If O_{XY} (respectively U_{XY}) does not induce a clique in H (respectively H') then we can use the following lemma:

Lemma 6. *Let $v_1, v_2 \in O_{XY}$. If $v_2 \notin N(v_1)$ then $\min\{|\Delta\text{IN}(H, v_1)|, |\Delta\text{IN}(H, v_2)|\} \leq \frac{|H|}{2}$.*

Proof. It suffices to prove that $\Delta\text{IN}(H, v_1) \cap \Delta\text{IN}(H, v_2) = \emptyset$. Assume $x \in \Delta\text{IN}(H, v_1)$ and let $x = x_0, \ldots, x_k = v_1$ be a shortest path from x to v_1 in H. Since the neighborhood of v_2 remains intact in $H \cdot \rho(v_1)$ there is no edge from v_2 to any vertex in that path. Therefore this path exists in $H \cdot \rho(v_2)$ and hence $u \notin \Delta\text{IN}(H, v_2)$. ■

Align the nodes of $G(A, B)$ according to π_A. For two nodes in $G(A, B)$, p_1 and p_2, denote $p_1 < p_2$ iff p_1 is to the left of p_2. For a vertex v in $H = \Omega(A, B, \pi_A)$, denote by *Left*(v) and *Right*(v) the left and right endpoints respectively of its gray edge. Suppose $O_{XY} = \{v_1, \ldots, v_k\}$, where *Left*$(v_j) <$ *Left*(v_{j+1}) for $j = 1..k-1$. If there exist two consecutive vertices v_j and v_{j+1} such that *Right*$(v_j) >$ *Right*(v_{j+1}), then we found two edges that do not overlap. Thus $v_{j+1} \notin N(v_j, H)$. By Lemma 6 $\min\{|\Delta\text{IN}(H, v_j)|, |\Delta\text{IN}(H, v_{j+1})|\} \leq \frac{|H|}{2}$. Otherwise, the vertices in O_{XY} form a clique in H. We can find whether U_{XY} induces a clique in H' in a similar manner by aligning the nodes of $G(A, B)$ according to $\pi_A \cdot \rho(X)$.

Suppose O_{XY} induces a clique in H and U_{XY} induces a clique in H' (one of which might be empty). In this case we use the proof of Theorem 4 in order to find a vertex v satisfying $|\Delta\text{IN}(H, v)| \leq \frac{|H|}{2}$. We calculate the score in H for every vertex in O_{XY} and the score in H' for every vertex in U_{XY} in the following way. Let $\{I_1, \ldots, I_k\}$ be a set of intervals forming a clique. Let $U = \{J_1, \ldots, J_l\}$ be another set of intervals. Let $U(j)$ denote the number of intervals in U that overlap with I_j. There is an algorithm by Kaplan et al. (2000) that computes $U(1), \ldots, U(k)$ in $O(k + l)$. We use this algorithm twice to compute $|N_{\text{EXT}}(v_j)|$ and $|N_{\text{IN}}(v_j)|$, for $j = 1..k$.

4.2.3. Performing a recursive call Suppose the external vertex v chosen in the first step of the algorithm satisfies $M = \Delta\text{IN}(H, v) \neq \emptyset$. Let $H = \Omega(A, B, \pi_A)$. Let H_M be the subgraph of H induced by $M \cup CH(v)$. We demonstrate below how to build two co-tailed genomes, A_M and B_M, in linear time, for which there exists an Ω-graph $H'_M = \Omega(A_M, B_M, \pi_{A_M})$ satisfying: *(i)* $H_M \subset H'_M$, *(ii)* $|H'_M| \leq |H_M| + 2$, and *(iii)* Every $u \in H'_M \setminus H_M$ is external and $\rho(u) = \rho(v)$.

Every internal component in $G(A \cdot \rho(v), B)$ contains in its span one of the new black edges created by $\rho(v)$. A component in M is *maximal* if its span is maximal. Since there are two new black edges in $G(A \cdot \rho(v), B)$, there are at most two maximal components in M. Note that for every $v \in M$, its two endpoints belong to the span of a maximal component. Construct genomes A_M and B_M in the following way.

Case 1: There are two maximal components in M. Let I_1 and I_2 be the spans of the two maximal components in M (after applying $\rho(v)$). I_1 and I_2 are disjoint since every maximal component belong to a different chromosome of $A \cdot \rho(v)$. By Lemma 1, there exist two intervals I'_1 and I'_2 in B, where for $i = 1, 2$ I_i and I'_i have the same set of elements and *Tails*$(I_i) =$ *Tails*(I'_i). Let $B_M = \{I'_1, I'_2\}$. Let A_M be the result of the translocation on $\{I_1, I_2\}$ that cuts the two new black edges in I_1 and I_2 and recreates the old black edges that were originally cut by $\rho(v)$ (i.e., the translocation inverse to $\rho(v)$).

Case 2: There is exactly one maximal component in M. In this case only one of the chromosomes in $A \cdot \rho(v)$ contains components from M. Let I be the span of the maximal component in M. Again, by Lemma 1 there exists an interval I' in B with the same elements as I, satisfying *Tails*$(I) =$ *Tails*(I'). Let $B_M = \{I', (i_1, i_2)\}$, where (i_1, i_2) is the new black edge in $A \cdot \rho(v)$ that is not contained in any of the components in M. Let A_M be the result of the translocation on $\{I, (i_1, i_2)\}$ that cuts the new black edge in I_1 and (i_1, i_2) and recreates the old black edges that were originally cut by $\rho(v)$ (i.e., the translocation inverse to $\rho(v)$).

Obviously in both cases A_M and B_M are co-tailed. Each of the two chromosomes in A_M (respectively, B_M) is an interval in A (respectively, B). Moreover, A_M (equivalently, B_M) contains the endpoints of each and every gray edge in M. Let $H'_M = \Omega(A_M, B_M, \pi_{A_M})$ where π_{A_M} is a concatenation of the two chromosomes in A_M in which the elements appear in the same order as in π_A. It is not hard to see that the H_M is an induced subgraph of H'_M. H'_M contains one or two additional vertices that do not belong to H_M. These additional vertices define the same translocation as v (one of which is indeed v) and correspond to isolated vertices (i.e., trivial internal components) in $H'_M \cdot \rho(v)$. Thus, the (one or two) additional vertices in H'_M are external. Since H_M does not contain adjacencies, so does H'_M.

The above described implementation implies:

Lemma 7. *Algorithm Find_Safe_Translocation_Recursive can be implemented in linear time.*

Proof. We have demonstrated how to implement the first two steps of the algorithm in linear time. Let v be the vertex chosen in step 1 of the algorithm. Suppose $M = \Delta\text{IN}(H, v) \neq \emptyset$. In this case we presented a way to construct two co-tailed genomes, A_M and B_M, whose Ω-graph is almost identical to H_M (there are one or two additional external vertices in H'_M that define the same translocation as v). Obviously this construction can be done in linear time. It is only left to prove that the number of elements in the genomes decreases by a constant factor in every call.

Let n and n_M be the number of genes in A and A_M, respectively. In every recursive call, the number of chromosomes involved is 2. Hence $|H| = n - N$ (i.e., gray edges in $G(A, B)$) and $|H'_M| = n_M - 2$. Suppose $|H_M| \leq \frac{|H|}{2}$ (step 1), then $|H_M| \leq \frac{n-N}{2} \leq \frac{n}{2} - 1$. Now $n_M = |H'_M| + 2 \leq |H_M| + 4 \leq \frac{n}{2} + 3$. Thus for $n \geq 18$, $n_M \leq \frac{2n}{3}$. We update the algorithm as follow. At the beginning, we verify that the number of genes is at least 18. In this case a recursive call (if needed) will be made with genomes with at most $\frac{2}{3}$ of the genes in A and B. Otherwise, we simply search for a proper safe translocation by computing $\Delta\text{IN}(H, v)$ for every external vertex v. ■

Corollary 3. *The recursive algorithm solves SRT in $O(n^2)$ time.*

5. DISCUSSION

The fundamental observation of Hannenhalli and Pevzner (1995) that translocations can be mimicked by reversals was made over a decade ago, but until recently the analyses of SRT and SBR had little in common. Here and in Ozery-Flato and Shamir (2006a), we tighten the connection between the two problems, by presenting a new framework for the study of SRT that builds directly on ideas and theory developed for SBR. Using this framework we show here how to transform two central algorithms for SBR, Bergeron's score-based algorithm and the Berman-Hannenhalli's recursive algorithm, into algorithms for SRT. These new algorithms for SRT maintain the time complexity of the original algorithms for SBR. These results improve our understanding of the connection between the two problems. Still, deeper investigation into the relation between SRT and SBR is needed. In particular, providing a reduction from SRT to SBR or vice versa is an interesting open problem.

Algorithms for SRT can only be applied to a pair of genomes having the same set of chromosome ends. This requirement is removed if SRT is extended to allow for non-reciprocal translocations, including fissions and fusions of chromosomes, and the latter can be viewed as translocations involving empty chromosomes (Hannenhalli and Pevzner, 1995). This more general problem of sorting by translocations can be reduced in linear time to SRT, as we intend to prove in a future work.

The problem of sorting by reversals, translocations, fissions, and fusions (SBRT) was studied (Hannenhalli and Pevzner, 1995; Ozery-Flato and Shamir, 2003; Tesler, 2002a) and proven to be polynomial. An algorithm solving SBRT is used by the applications GRIMM (Tesler, 2002b) and MGR (Bourque and Pevzner, 2002), which analyze genome rearrangements in real biological data (Bourque et al., 2004; Murphy et al., 2005; Pevzner and Tesler, 2003). The first step in the current algorithm for SBRT generates two co-tailed genomes, say A and B, with the same distance as the two input genomes (Tesler, 2002a). In the following steps, genome A is sorted into genome B using reciprocal translocations and *internal* reversals that do not alter the set of chromosome tails. In other words, SBRT is solved by a reduction to a more constrained problem that allows only for reciprocal translocations and internal reversals. We refer to this constrained problem as SBRT^{C}. SBRT^{C} is currently solved by a reduction to SBR, where each reversal simulates either a reciprocal translocation or an internal reversal. We believe that an algorithm for SBRT^{C} that explicitly treats translocations and reversals as distinct operations would be more natural and powerful than one that does not. In a future work, we intend to prove that each of the algorithms presented here and in (Ozery-Flato and Shamir, 2006a) can be extended to solve SBRT^{C}, even when reversals are given priority over translocations (i.e., a "good" reversal move have a higher priority than a "good" translocation move).

In an optimal solution to SRT, SBR, and SBRT, every move is safe, i.e., it does not create "bad components." Thus the algorithms for these problems mainly focus on finding safe moves. Finding safe moves is conceptually and algorithmically the hardest part in all these algorithms. In a ground-breaking paper, Yancopoulos et al. (2005) proposed a new formulation that bypasses the need for safe reversals/

translocations by introducing a new genome rearrangement operation called *double-cut-and-join* (DCJ). Translocations, reversals, fissions, and fusions can all be viewed as special cases of the DCJ operation. Unlike all the above operations, a DCJ operation can "loop out" a circular chromosome, which can be later reabsorbed by another operation. Thus the problem of sorting by DCJ operations (SDCJ) allows for the creation of intermediate circular chromosomes. Looping out a circular chromosome followed by its reabsorption can also simulate a *block interchange* of two blocks in the same chromosome. The problem of sorting by block interchanges was studied in Christie (1996) and Lin et al. (2005). The ability of DCJs to create and reabsorb circular chromosomes yields a powerful rearrangement model, for which no "bad components" exist. This makes the analysis, distance formula, and algorithms of SDCJ (Bergeron et al., 2006b; Yancopoulos et al., 2005) much simpler and very elegant, in comparison with SRT, SBR, and SBRT. While circular chromosomes are quite common in prokaryotes cells, they have been found sporadically in eukaryotes cells, and with some rare exceptions, they are usually not inherited (Ishikawa and Naito, 1999). Thus for the evolution of eukaryotes species, it is reasonable to assume a minimal use, if any, of circular chromosomes. In particular, when there are no bad components, any algorithm for SBRT solves SDCJ without creating circular intermediates.

In the future we intend to study SBRT with additional restrictions that will make its solutions more biologically acceptable. An example for an additional constraint is the exclusion of translocations that create acentric chromosomes (i.e., chromosomes that lack a centromere), since these chromosomes are likely to be lost during subsequent cell divisions (Sullivan et al., 2001). As a first step towards solving this problem, we recently provided a polynomial time algorithm for the constrained problem where only reciprocal translocations that do not create acentric chromosomes are allowed (Ozery-Flato and Shamir, 2007). Another interesting variant of SBRT we wish to study considers a model in which one type of operation is preferable over the other. We believe that the study of SRT and its alignment with SBR theory will assist in the study of these variants of SBRT.

ACKNOWLEDGMENTS

We thank the referees for careful and critical comments that helped to improve this paper. This study was supported in part by the Israeli Science Foundation (grant 309/02). A preliminary version of this study appeared in the proceedings of 4th RECOMB Satellite Workshop on Comparative Genomics (Ozery-Flato and Shamir, 2006b).

REFERENCES

Bader, D.A., Moret, B.M.E., and Yan, M. 2001. A linear-time algorithm for computing inversion distance between signed permutations with an experimental study. *J. Comput. Biol.* 8, 483–491.

Bergeron, A. 2005. A very elementary presentation of the Hannenhalli-Pevzner theory. *Discrete Appl. Math.* 146, 134–145.

Bergeron, A., Mixtacki, J., and Stoye, J. 2006a. On sorting by translocations. *J. Comput. Biol.* 13, 567–578.

Bergeron, A., Mixtacki, J., and Stoye, J. 2006b. A unifying view of genome rearrangements. *Lect. Notes Comput. Sci.* 4175, 163–173.

Berman, P., and Hannenhalli, S. 1996. Fast sorting by reversal. *Lect. Notes Comput. Sci.* 1075, 168–185.

Bourque, G., and Pevzner, P.A. 2002. Genome-scale evolution: reconstructing gene orders in the ancestral species. *Genome Res.* 12, 26–36.

Bourque, G., Pevzner, P.A., and Tesler, G. 2004. Reconstructing the genomic architecture of ancestral mammals: lessons from human, mouse, and rat genomes. *Genome Res.* 14, 507–516.

Christie, D.A. 1996. Sorting permutaions by block interchanges. *Inform. Process. Lett.* 60, 165–169.

Hannenhalli, S. 1996. Polynomial algorithm for computing translocation distance between genomes. *Discrete Appl. Math.* 71, 137–151.

Hannenhalli, S., and Pevzner, P. 1995. Transforming men into mice (polynomial algorithm for genomic distance problems). *36th Annual Symposium on Foundations of Computer Science (FOCS'95)*, 581–592. IEEE Computer Society Press, Los Alamitos, CA.

Hannenhalli, S., and Pevzner, P. 1999. Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. *J. ACM* 46, 1–27.

Ishikawa, F., and Naito, T. 1999. Why do we have linear chromosomes? A matter of Adam and Eve. *Mutation Res. DNA Repair* 434, 99–107.

Kaplan, H., Shamir, R., and Tarjan, R.E. 2000. Faster and simpler algorithm for sorting signed permutations by reversals. *SIAM J. Comput.* 29, 880–892.

Kaplan, H., and Verbin, E. 2005. Sorting signed permutations by reversals, revisited. *J. Comput. Syst. Sci.* 70, 321–341.

Kececioglu. J.D., and Ravi, R. 1995. Of mice and men: Algorithms for evolutionary distances between genomes with translocation. *Proc. 6th Annual Symposium on Discrete Algorithms*, 604–613. ACM Press, New York.

Lin, Y.C., Lu, C.L., Chang, H.-Y., et al. 2005. An efficient algorithm for sorting by block-interchanges and its application to the evolution of vibrio species. *J. Comput. Biol.* 12, 102–112.

Murphy, W.J., Larkin, D.M., Everts van der Wind, A., et al. 2005. Dynamics of mammalian chromosome evolution inferred from multispecies comparative maps. *Science* 309, 613–617.

Nadeau, J.H., and Taylor, B.A. 1984. Lengths of chromosomal segments conserved since divergence of man and mouse. *Proc. Natl. Acad. Sci. USA* 81, 814–818.

Ozery-Flato, M., and Shamir, R. 2003. Two notes on genome rearrangements. *J. Bioinformatics Comput. Biol.* 1, 71–94.

Ozery-Flato, M., and Shamir, R. 2006a. An $O(n^{3/2}\sqrt{\log(n)})$ algorithm for sorting by reciprocal translocations. *Lect. Notes Comput. Sci.* 4009, 258–269.

Ozery-Flato, M., and Shamir, R. 2006b. Sorting by translocations via reversals theory. *Lect. Notes Comput. Sci.* 4205, 87–98.

Ozery-Flato, M., and Shamir, R. 2007. Rearrangements in genomes with centromeres. Part I: Translocations. In *Proceedings of the 11th Annual International Conference on Computational Molecular Biology (RECOMB 2007)*, vol. 4453 of LNCS, Springer, 339–353.

Pevzner, P.A., and Tesler, G. 2003. Genome rearrangements in mammalian evolution: lessons from human and mouse genomes. *Genome Res.* 13, 37–45.

Sullivan, B.A., Blower, M.D., and Karpen, G.H. 2001. Determining centromere identity: cyclical stories and forking paths. *Nat. Rev. Genet.* 2, 584–596.

Tannier, E., Bergeron, A., and Sagot, M. 2007. Advances on sorting by reversals. *Discrete Appl. Math.* 155, 881–888.

Tesler, G. 2002a. Efficient algorithms for multichromosomal genome rearrangements. *J. Comp. Sys. Sci.* 65, 587–609.

Tesler, G. 2002b. GRIMM: genome rearrangements web server. *Bioinformatics* 18, 492–493.

Yancopoulos, S., Attie, O., and Friedberg, R. 2005. Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinformatics* 21, 3340–3346.

Address reprint requests to:
Dr. Michal Ozery-Flato
School of Computer Science
Tel-Aviv University
Tel-Aviv 69978, Israel

E-mail: ozery@post.tau.ac.il

Chapter 4

Sorting Genomes with Centromeres by Translocations

JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 15, Number 7, 2008

Pp. 793–812
DOI: 10.1089/cmb.2007.0136

Sorting Genomes with Centromeres by Translocations

MICHAL OZERY-FLATO and RON SHAMIR

ABSTRACT

A *centromere* is a special region in the chromosome that plays a vital role during cell division. Every new chromosome created by a genome rearrangement event must have a centromere in order to survive. This constraint has been ignored in the computational modeling and analysis of genome rearrangements to date. Unlike genes, the different centromeres are indistinguishable, they have no orientation, and only their location is known. A prevalent rearrangement event in the evolution of multi-chromosomal species is translocation (i.e., the exchange of tails between two chromosomes). A translocation may create a chromosome with no centromere in it. In this paper, we study for the first time centromeres-aware genome rearrangements. We present a polynomial time algorithm for computing a shortest sequence of translocations transforming one genome into the other, where all of the intermediate chromosomes must contain centromeres. We view this as a first step towards analysis of more general genome rearrangement models that take centromeres into consideration.

Key words: sorting by translocations, genome rearrangements, comparative genomics, combinatorics.

1. INTRODUCTION

GENOMES OF RELATED SPECIES tend to have similar genes that are, however, ordered differently. The distinct orderings of the genes are the result of genome rearrangements. Inferring the sequence of genome rearrangements that took place during the course of evolution is an important question in comparative genomics. The genomes of higher organisms, such as plants and animals, are partitioned into continuous units called *chromosomes*. Every chromosome contains a special region called *a centromere*, which plays a vital role during cell division. An *acentric* chromosome (i.e., one that lacks a centromere) is likely to be lost during subsequent cell divisions (Sullivan et al., 2001). Thus, a rearrangement scenario that preserves a centromere in each chromosome is more biologically realistic than one that does not. The computational studies on genome rearrangements to date have ignored the existence and role of centromeres. Hence, the rearrangement scenarios for multi-chromosomal genomes produced by current algorithms may include genomes with non-viable chromosomes. In this study, we begin to address the centromeres in the computational analysis of genome rearrangements.

School of Computer Science, Tel-Aviv University, Tel-Aviv, Israel.

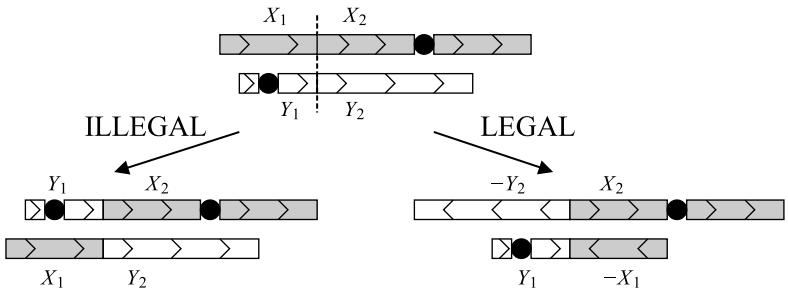


FIG. 1. An example of legal and illegal translocations for a certain cut of two chromosomes. The black circles denote the location of the centromeres; the broken line indicates the positions where the two chromosomes were cut.

Since sequencing a centromere is almost impossible due to the repeated sequences it contains, the only information we have on a centromere is its location in the genome. Therefore, in the model we define, centromeres appear as anonymous and orientation-less elements. We say that a genome is *legal* if each of its chromosomes contains a single centromere. A *legal* rearrangement operation results in a legal genome (Fig. 1). The *legal rearrangement sorting problem* is defined as follows: given two legal genomes A and B, find a shortest sequence of legal rearrangement operations that transforms A into B. The length of this sequence is the *legal distance* between A and B.

A *reciprocal translocation* is a rearrangement in which two chromosomes exchange non-empty ends. A reciprocal translocation results in an illegal genome if exactly one of the exchanged ends contains a centromere. In this paper, we focus on the problem of legal sorting by reciprocal translocations (LSRT). This problem is a refinement of the "sorting by reciprocal translocations" problem (SRT), which ignores centromeres. SRT was studied in Hannenhalli (1996), Bergeron et al. (2006), and Ozery-Flato and Shamir (2006a,b), and is solvable in polynomial time. Clearly, a solution to SRT may not be a solution to LSRT, since 50% of the possible reciprocal translocations are illegal (Fig. 1). Indeed, in many cases, more rearrangements are needed in order to legally sort a genome.

In this study we present a polynomial time algorithm for LSRT. The basic idea is to transform LSRT into SRT, by replacing pairs of centromeres in the two genomes by new unique oriented elements. Our algorithm is based on finding a mapping between the centromeres of the two given genomes such that the solution to the resulting SRT instance is minimum. We show that an optimal mapping can be found in polynomial time. To the best of our knowledge, this is the first rearrangement algorithm that considers centromeres. While a model that permits only reciprocal translocations is admittedly quite remote from the biological reality, we hope that the principles and structure revealed here will be instrumental for analyzing more realistic models in the future. One additional advantage of centromere-aware models is that they restrict drastically the allowed sequences of operations, and therefore are less likely to suffer from high multiplicity of optimal sequences.

The paper is organized as follows. Section 2 gives the necessary preliminaries. In Section 3, we model LSRT and present some elementary properties of it. Section 4 describes an exponential algorithm for LSRT, which searches for an optimal mapping between the centromeres of A and B (i.e., one that leads to a minimum SRT solution). In Section 5, we take a first step towards a polynomial time algorithm for LSRT by proving a bound that is at most two translocations away from the legal translocation distance. In Section 6, we present a theorem leading to a polynomial time algorithm for computing the legal translocation distance and solving LSRT.

A preliminary version of this study appeared in the proceedings of RECOMB 2007 (Ozery-Flato and Shamir, 2007).

2. PRELIMINARIES

This section provides the needed background for SRT. The definitions follow previous literature on translocations (Hannenhalli, 1996; Bergeron et al., 2006; Ozery-Flato and Shamir, 2006a, 2006b). In the model we consider, a *genome* is a set of chromosomes. A *chromosome* is a sequence of genes. A *gene* is

identified by a positive integer. All genes in the genome are distinct. When it appears in a genome, a gene is assigned a sign of plus or minus. The following is an example of a genome with two chromosomes and six genes: $\{(1,-5),(-4,-3,-2,6)\}$.

The *reverse* of a sequence of genes $I = (x_1,\ldots,x_l)$ is $-I = (-x_l,\ldots,-x_1)$. Two chromosomes, X and Y, are called *identical* if either $X = Y$ or $X = -Y$. Therefore, *flipping* chromosome X into $-X$ does not affect the chromosome it represents.

Let $X = (X_1, X_2)$ and $Y = (Y_1, Y_2)$ be two chromosomes, where X_1, X_2, Y_1, Y_2 are sequences of genes. A *translocation* cuts X into X_1 and X_2 and Y into Y_1 and Y_2 and exchanges segments between the chromosomes. It is called *reciprocal* if X_1,X_2, Y_1 and Y_2 are all non-empty. There are two types of translocations on X and Y. A *prefix-suffix* translocation switches X_1 with Y_2:

$$(\underline{X_1}, X_2), (Y_1, \underline{Y_2}) \Rightarrow (-Y_2, X_2), (Y_1, -X_1).$$

A *prefix-prefix* translocation switches X_1 with Y_1:

$$(\underline{X_1}, X_2), (\underline{Y_1}, Y_2) \Rightarrow (Y_1, X_2), (X_1, Y_2).$$

Note that we can mimic one type of translocation by a flip of one of the chromosomes followed by a translocation of the other type.

For a chromosome $X = (x_1,\ldots,x_k)$, define $\mathit{Tails}(X) = \{x_1, -x_k\}$. Note that flipping X does not change $\mathit{Tails}(X)$. For a genome A, define $\mathit{Tails}(A) = \bigcup_{X\in A} \mathit{Tails}(X)$. For example:

$$\mathit{Tails}(\{(1,-3,-2,4,-7,8),(6,5)\}) = \{1,-8,6,-5\}.$$

Two genomes A_1 and A_2 are *co-tailed* if $\mathit{Tails}(A_1) = \mathit{Tails}(A_2)$. In particular, two co-tailed genomes have the same number of chromosomes. Note that if A_2 was obtained from A_1 by performing a reciprocal translocation, then $\mathit{Tails}(A_2) = \mathit{Tails}(A_1)$. Therefore, SRT is solvable only for genomes that are co-tailed. For the rest of this paper, the word "translocation" refers to a reciprocal translocation, and we assume that the given genomes, A and B, are co-tailed. Denote the set of tails of A and B by *Tails*.

2.1. Cycle graph

Let n and N be the number of genes and chromosomes in A (equivalently, B), respectively. We shall always assume that both A and B consist of the genes $\{1,\ldots,n\}$. The *cycle graph* of A and B, denoted $G(A,B)$, is defined as follows. The set of vertices is $\bigcup_{i=1}^{n}\{i^0, i^1\}$. The vertices i^0 and i^1 are called the two *ends* of gene i (think of them as ends of a small arrow directed from i^0 to i^1). For every two genes, i and j, where j immediately follows i in some chromosome of A (respectively, B) add a black (respectively, gray) edge $(i,j) \equiv (\mathit{out}(i), \mathit{in}(j))$, where $\mathit{out}(i) = i^1$ if i has a positive sign in A (respectively, B) and otherwise $\mathit{out}(i) = i^0$, and $\mathit{in}(j) = j^0$ if j has a positive sign in A (respectively, B) and otherwise $\mathit{in}(j) = j^1$. An example is given in Figure 2a. There are $n - N$ black edges and $n - N$ gray edges in $G(A,B)$. A gray edge (i,j) is *external* if the genes i and j belong to different chromosomes of A, otherwise it is *internal*. A cycle is *external* if it contains an external edge, otherwise it is *internal*.

Every vertex in $G(A,B)$ has degree 2 or 0, where vertices of degree 0 (isolated vertices) belong to *Tails*. Therefore, $G(A,B)$ is uniquely decomposed into cycles with alternating gray and black edges. An *adjacency* is a cycle with two edges. A *breakpoint* is a black edge that is not part of an adjacency.

2.2. Overlap graph with chromosomes

A *signed permutation* $\pi = (\pi_1,\ldots,\pi_n)$ is a permutation on the integers $\{1,\ldots,n\}$, where a sign of plus or minus is assigned to each number. If A is a genome with the set of genes $\{1,\ldots,n\}$ then any concatenation π_A of the chromosomes of A is a signed permutation of size n.

Place the vertices of $G(A,B)$ along a straight line according to their order in π_A. Now, every gray edge and every chromosome is associated with an interval of vertices in $G(A,B)$. Two intervals *overlap* if their intersection is not empty but none contains the other. The *overlap graph with chromosomes* of A and B w.r.t. π_A, denoted $\mathit{OVCH}(A,B,\pi_A)$, is defined as follows. The set of nodes is the set of gray

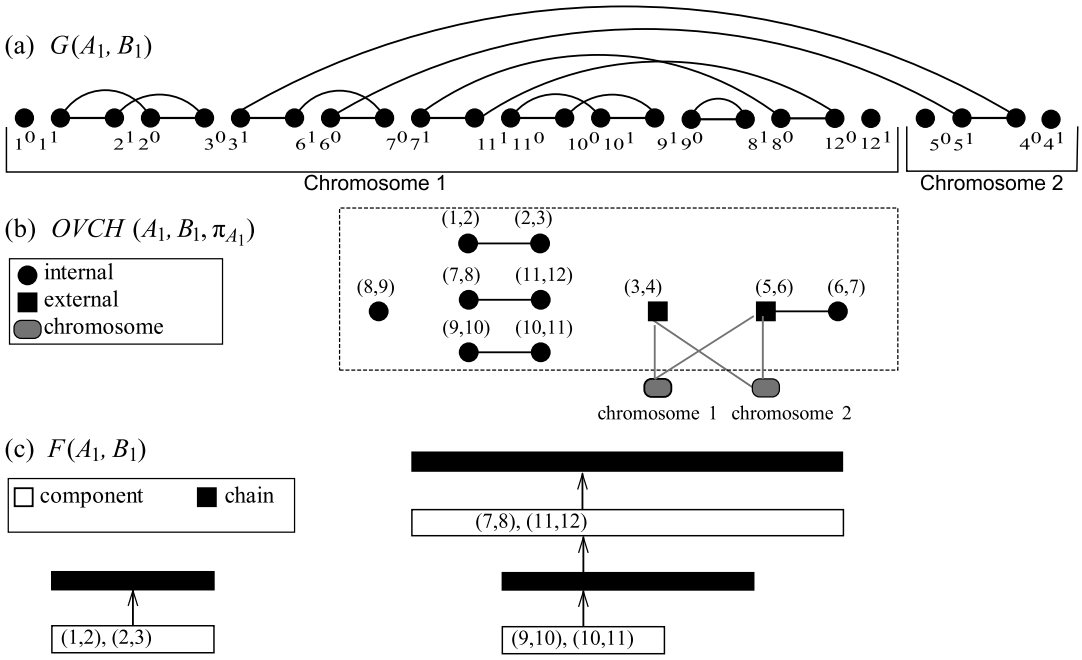


FIG. 2. Auxiliary graphs for $A_1 = \{(1, -2, 3, -6, 7, -11, 10, -9, -8, 12), (5, 4)\}$, $B_1 = \{(1, \ldots, 4), (5, \ldots, 12)\}$ ($\pi_{A_1} = (1, -2, 3, -6, 7, -11, 10, -9, -8, 12, 5, 4)$). **(a)** The cycle graph. Black edges are horizontal; gray edges are curved. **(b)** The overlap graph with chromosomes. The graph induced by the vertices within the dashed rectangle is $OV(A_1, B_1, \pi_{A_1})$. **(c)** The forest of internal components.

edges and chromosomes in $G(A, B)$. Two nodes are connected if their corresponding intervals overlap. An example is given in Figure 2b. This graph is an extension of the overlap graph of a signed permutation defined in (Kaplan et al., 2000). Let $OV(A, B, \pi_A)$ be the subgraph of $OVCH(A, B, \pi_A)$ induced by the set of nodes that correspond to gray edges (i.e., excluding the chromosomes' nodes). We shall use the word "component" for a connected component of $OV(A, B, \pi_A)$.

In order to prevent confusion, we will refer to nodes that correspond to chromosomes as "chromosomes" and reserve the word "vertex" for nodes that correspond to gray edges. A vertex is external (resp. internal) if it corresponds to an external (resp. internal) gray edge. Obviously a vertex is external iff it is connected to a chromosome. A component is *external* if it contains an external vertex, otherwise it is *internal*. A component is *trivial* if it is composed of one (internal) vertex. A trivial component corresponds to an adjacency. Note that the internal/external state of a vertex in $OVCH(A, B, \pi_A)$ does not depend on π_A. Therefore, the set of internal components in $OVCH(A, B, \pi_A)$ is independent of π_A. The *span* of a component M is the minimal interval of genes $I(M) = [i, j] \subset \pi_A$ that contains the interval of every vertex in M. Clearly, $I(M)$ is independent of π_A iff M is internal. The following lemma follows from A and B being co-tailed and (Corollary 2.2 in Kaplan et al., 2000):

Lemma 1. *Every internal component corresponds to the set of gray edges of a union of cycles in* $G(A, B)$.

The set of internal components can be computed in linear time using an algorithm in Bader et al. (2001).

2.3. Forest of internal components

$(M_1, \ldots, M_t)$ is a *chain* of components if $I(M_j)$ and $I(M_{j+1})$ overlap in exactly one gene for $j = 1, \ldots, t-1$. The *forest of internal components* (Bergeron et al., 2006), denoted $F(A, B)$, is defined as follows. The vertices of $F(A, B)$ are *(i)* the non-trivial internal components and *(ii)* every maximal chain of internal components that contains at least one non-trivial component. Let M and C be two vertices

in $F(A, B)$ where M corresponds to a component and C to a chain. $M \to C$ is an edge of $F(A, B)$ if $M \in C$. $C \to M$ is an edge of $F(A, B)$ if $I(C) \subset I(M)$ and $I(M)$ is minimal (Fig. 2c). We will refer to a component that is a leaf in $F(A, B)$ as simply a *leaf*.

2.4. Reciprocal translocation distance

The *reciprocal translocation distance* between A and B is the length of a shortest sequence of reciprocal translocations that transforms A into B. Let $c(A, B)$ denote the number of cycles in $G(A, B)$. Let $|F(A, B)|$ and $l(A, B)$ denote the number of trees and leaves in $F(A, B)$, respectively. Obviously $|F(A, B)| \leq l(A, B)$. Define

$$\delta(A, B) \equiv \delta(F(A, B)) = \begin{cases} 2 & \text{if } |F(A, B)| = 1 \text{ and } l(A, B) \text{ is even} \\ 1 & \text{if } l(A, B) \text{ is odd} \\ 0 & \text{otherwise } (|F(A, B)| \neq 1 \text{ and } l(A, B) \text{ is even}) \end{cases}$$

Theorem 1 (Bergeron et al., 2006; Hannenhalli, 1996). *The reciprocal translocation distance between A and B is $n - N - c(A, B) + l(A, B) + \delta(A, B)$.*

Let Δc denote the change in the number of cycles after performing a translocation on A. Then $\Delta c \in \{-1, 0, 1\}$ (Hannenhalli, 1996). A translocation is *proper* if $\Delta c = 1$, *improper* if $\Delta c = 0$ and *bad* if $\Delta c = -1$.

Corollary 1. *Every translocation in a shortest sequence of translocations transforming A into B is either proper or bad.*

Proof. An improper translocation cannot decrease the translocation distance since it does not affect any parameter in its formula. ■

3. INCORPORATING CENTROMERES INTO A GENOME

We extend the model described above by adding the requirement that every genome is legal (i.e., every chromosome contains exactly one centromere). We denote the location of a centromere in a chromosome by the element $\bullet$. The element $\bullet$ is unsigned and thus does not change under chromosome flips. The following is an example of a legal genome: $\{(1, 2, 3, \bullet, 4), (\bullet, 5, 6)\}$. The set of tails is defined for regular elements, thus $\mathit{Tails}(\bullet, 5, 6) = \{5, -6\}$. We assume that a cut of a chromosome does not split a centromere. Clearly, for every cut of two chromosomes one translocation is legal while the other is not (Fig. 1).

3.1. A new precondition

We present here a simple condition for the solvability of LSRT. If this condition is not satisfied then A cannot be transformed into B by legal translocations. For chromosome $X = (x_1, \ldots, x_i, \bullet, x_{i+1}, \ldots, x_k)$ define $\mathit{Elements}(X) = \{x_1, \ldots, x_i, -x_{i+1}, \ldots, -x_k\}$. Note that $\mathit{Elements}(X) = \mathit{Elements}(-X)$. For genome A we define $\mathit{Elements}(A) = \bigcup_{X \in A} \mathit{Elements}(X)$. For example:

$$\mathit{Elements}(\{(1, 2, \bullet, 3, 4), (\bullet, 5, 6)\}) = \{1, 2, -3, -4, -5, -6\}.$$

Observation 1. *Let A and B be two legal genomes. If A can be transformed into B by a sequence of legal translocations then $\mathit{Elements}(A) = \mathit{Elements}(B)$.*

We will see later that this condition is also sufficient. Thus, for the rest of this paper we assume that the input to LSRT is co-tailed genomes A and B satisfying $\mathit{Elements}(A) = \mathit{Elements}(B) = \mathit{Elements}$. The cycle graph of A and B, $G(A, B)$, ignores the $\bullet$ elements.

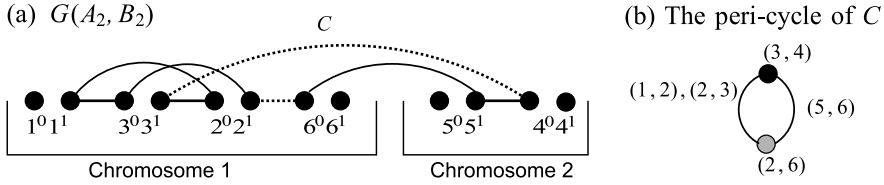


FIG. 3. Pericentric edges and peri-cycles. $A_2 = \{(1, 3, 2, \bullet, 6), (\bullet, 5, 4)\}$, $B_2 = \{(1, 2, 3, \bullet, 4), (\bullet, 5, 6)\}$. **(a)** The cycle graph $G(A_2, B_2)$. Pericentric edges are denoted by dotted lines. **(b)** The peri-cycle of the single cycle in $G(A_2, B_2)$. The labels of the edges denote the set of gray edges in the corresponding paths.

3.2. On the gap between the legal distance and the "old" distance

Let $d(A, B)$ denote the legal translocation distance between A and B. Let $d_{\text{old}}(A, B)$ denote the translocation distance between A and B when the $\bullet$ elements are ignored. Obviously $d(A, B) \geq d_{\text{old}}(A, B)$. Consider the genomes A_2 and B_2 in Figure 3. It can be easily verified that $d_{\text{old}}(A_2, B_2) = 3$ and $d(A_2, B_2) = 4$. This example is easily extendable to two genomes A_{2k} and B_{2k}, with $2k$ chromosomes each, such that $d_{\text{old}}(A_{2k}, B_{2k}) = 3k$ and $d(A_{2k}, B_{2k}) = 4k$.

3.3. Telocentric chromosomes

A chromosome is *telocentric* if its centromere is located at one of its endpoints. For example the chromosome $(\bullet, 5, 6)$ is telocentric.

Lemma 2. *Let A and B be co-tailed genomes satisfying $Elements(A) = Elements(B)$. Then A and B have the same number of telocentric chromosomes. Moreover, the set of genes adjacent to the centromeres in the telocentric chromosomes is the same.*

Proof. Let i be a gene adjacent to the centromere in a telocentric chromosome in A. Thus i is a tail of A and hence a tail of B (since A and B are co-tailed). Suppose w.l.o.g. that i is the leftmost gene in its chromosome both in A and in B and that the centromere is located to the left of i in A. In this case, since genomes A and B are co-tailed, i has the same sign in A and B. Since $Elements(A) = Elements(B)$ it follows that the centromere is located to the left of i also in B. Thus, i is adjacent to the centromere in B and its chromosome is telocentric. ■

Let η denote the number of non-telocentric chromosomes in A and B. We shall show later how mapping between centromeres in non-telocentric chromosomes in A and B can help us to solve LSRT.

3.4. Pericentric and paracentric edges

A gray (respectively, black) edge in $G(A, B)$ is said to be *pericentric* if the two genes it connects flank a centromere in genome B (respectively, A). Otherwise it is called *paracentric* (Fig. 3a). For a gene i we define:

$$cent(i^0) = \begin{cases} -1 & \text{if } i \text{ has a positive sign in } Elements, \\ 1 & \text{otherwise.} \end{cases} \qquad cent(i^1) = -cent(i^0)$$

In other words, the sign of the end closer to the centromere (in both A and B) is positive, and the sign of the remote end is negative. The legality precondition (Section 3.1) implies the following key property:

Lemma 3. *Let (u, v) be an edge in $G(A, B)$. If (u, v) is pericentric then $cent(u) = cent(v) = 1$. Otherwise $cent(u)cent(v) = -1$.*

Proof. The nodes u and v are the ends of two adjacent genes i and j, respectively, in one of the genomes. Suppose (u, v) is pericentric. Then i and j flank a centromere in one of the genomes. Thus u is

the end of i closer to j and hence closer to the centromere (i.e., $cent(u) = 1$). Using similar arguments, $cent(v) = 1$.

Suppose (u, v) is paracentric. Then there is no centromere between i and j. W.l.o.g. assume that i is closer to the centromere than j. Then u is the end of i distant from the centromere and v is the end of j closer to the centromere. Therefore, $cent(u)cent(v) = -1$. ■

3.5. Peri-cycles

Let C be a cycle in $G(A, B)$. The *peri-cycle* of C, C^P, is defined as follows. The vertices of C^P are the pericentric edges in C. A vertex in C^P is colored gray (respectively, black) if the corresponding edge in C is gray (respectively, black). A path between two consecutive pericentric edges in C is translated to an edge between the two corresponding vertices in C^P (Fig. 3). Note that if C contains no pericentric edges then its peri-cycle is a null cycle (i.e., a cycle with no vertices).

Lemma 4. *Every peri-cycle has an even length and its node colors alternate along the cycle.*

Proof. Let C be a cycle that contains a black pericentric edge (u_1, v_1). Suppose $u_1, v_1, \ldots, u_k, v_k$ is a path between two consecutive black pericentric edges in C. In other words, (u_k, v_k) is a black pericentric edge (possibly $u_1 = u_k$ and $v_1 = v_k$) and there are no other black pericentric edges in this path. Then according to Lemma 3 $cent(v_1) = cent(u_k) = 1$. There is an odd number of edges in the path between v_1 and u_k and thus there must be an odd number of pericentric edges between v_1 and u_k (Lemma 3). It follows that there must exist at least one gray pericentric edge between any two consecutive black pericentric edges. The same argument for a pair of consecutive gray pericentric edges implies that between two such edges there must be at least one black pericentric edge. ■

It follows that every vertex/edge in a peri-cycle has an *opposite* vertex/edge. Removing two opposite vertices/edges from a peri-cycle results in two paths of equal length. We define the *degree* of a cycle as the number of gray (equivalently, black) vertices in its peri-cycle. For example, the single cycle in Figure 3 is of degree 1.

4. MAPPING THE CENTROMERES

This section demonstrates how mapping between the centromeres of A and B can be used to solve LSRT. We shall first see that trying all possible mappings and then solving the resulting SRT gives an exact exponential algorithm for LSRT. Later we shall show how to get an optimal mapping in polynomial time. Let $CEN = \{n + 1, \ldots, n + \eta\}$. For a genome A, let $\dot{\mathbb{A}}$ be the set of all possible genomes obtained by the replacement of each $\bullet$ element in the non-telocentric chromosomes by a distinct element from CEN. Each $i \in CEN$ can be added with either positive or negative sign. Thus $|\dot{\mathbb{A}}| = \eta!2^\eta$. For example, if $A_1 = \{(1, 2, \bullet, 3, 4), (\bullet, 5, 6)\}$ then $\dot{\mathbb{A}}_1$ consists of the genomes $\{(1, 2, 7, 3, 4), (\bullet, 5, 6)\}$ and $\{(1, 2, -7, 3, 4), (\bullet, 5, 6)\}$. Note that every $\dot{A} \in \dot{\mathbb{A}}$ satisfies $Tails(\dot{A}) = Tails$. For each $i \in CEN$ we define $cent(i^0) = cent(i^1) = -1$. A pair $\dot{A} \in \dot{\mathbb{A}}$ and $\dot{B} \in \dot{\mathbb{B}}$ defines a mapping between the centromeres in non-telocentric chromosomes of A and B.

Observation 2. *Let $\dot{A} \in \dot{\mathbb{A}}$ and $\dot{B} \in \dot{\mathbb{B}}$. Then every edge (u, v) in $G(\dot{A}, \dot{B})$ is paracentric and satisfies $cent(u)cent(v) = -1$.*

The notion of legality is easily generalized to partially mapped genomes: a genome is *legal* if each of its chromosomes contains either a single $\bullet$ element or a single, distinct element from CEN (but not both). Since A and $\dot{A} \in \dot{\mathbb{A}}$ differ only in their centromeres, there is a trivial bijection between the set of translocations on $\dot{A}$ and the set of translocations on A. This bijection also preserves legality: a legal translocation on $\dot{A}$ is bijected to a legal translocation on A.

Lemma 5. *Let $\dot{A} \in \dot{\mathbb{A}}$ and $\dot{B} \in \dot{\mathbb{B}}$. Then every proper translocation on $\dot{A}$ is legal and $d(\dot{A}, \dot{B}) = d_{old}(\dot{A}, \dot{B})$.*

Proof. Let $k = d_{\text{old}}(\dot{A}, \dot{B})$. If $k = 0$ then $\dot{A} = \dot{B}$ and hence $d(\dot{A}, \dot{B}) = 0$. Suppose $k > 0$. Let ρ be a translocation on $\dot{A}$ satisfying $d_{\text{old}}(\dot{A} \cdot \rho, \dot{B}) = k - 1$. According to Corollary 1, ρ is either proper or bad. Suppose ρ is bad. Then there is another bad translocation ρ' that cuts the exact positions as ρ, thus satisfying $d_{\text{old}}(\dot{A} \cdot \rho', \dot{B}) = k - 1$, and either ρ or ρ' is legal. Suppose ρ is proper. We shall prove that each of the new chromosomes contains a centromere and hence ρ is legal. Let X be a new chromosome resulting from the translocation ρ and let (u, v) be the new black edge in it. Since ρ is proper, $G(\dot{A} \cdot \rho, \dot{B})$ contains a path between u and v where all the edges existed in $G(\dot{A}, \dot{B})$. This path contains an odd number of edges. Following Observation 2 for $G(\dot{A}, \dot{B})$, $cent(u)cent(v) = -1$. X is composed of two old segments, X_u and X_v, that contain u and v respectively. If $cent(u) = -1$ then X_u contains an element from *CEN*, otherwise X_v contains one. In either case X contains an element from *CEN*. ■

Theorem 2. *Let* $\dot{A} \in \dot{\mathbb{A}}$. *Then* $d(A, B) = \min\{d_{old}(\dot{A}, \dot{B}) | \dot{B} \in \dot{\mathbb{B}}\}$.

Proof. By Lemma 5, $d(\dot{A}, \dot{B}) = d_{\text{old}}(\dot{A}, \dot{B})$ for every $\dot{A} \in \dot{\mathbb{A}}$ and $\dot{B} \in \dot{\mathbb{B}}$. Obviously a legal sorting of $\dot{A}$ into any $\dot{B} \in \dot{\mathbb{B}}$ induces a legal sorting sequence of the same length, of A to B. Thus, $\min\{d_{\text{old}}(\dot{A}, \dot{B}) | \dot{B} \in \dot{\mathbb{B}}\} \geq d(A, B)$. On the other hand, every sequence of legal translocations that sorts A into B induces a legal sorting of $\dot{A}$ into some $\dot{B} \in \dot{\mathbb{B}}$, thus $\min\{d_{\text{old}}(\dot{A}, \dot{B}) | \dot{B} \in \dot{\mathbb{B}}\} \leq d(A, B)$. ■

A pair of genomes, $\dot{A} \in \dot{\mathbb{A}}$ and $\dot{B} \in \dot{\mathbb{B}}$, define an *optimal* mapping between the centromeres of A and B if $d(A, B) = d_{\text{old}}(\dot{A}, \dot{B})$. Theorem 2 and Lemma 5 imply the following algorithm for LSRT:

Algorithm 1. *Sorting by legal translocations*

1: Choose $\dot{A} \in \dot{\mathbb{A}}$ arbitrarily.
2: Compute $\dot{B} = arg\min\{d_{\text{old}}(\dot{A}, \ddot{B}) | \ddot{B} \in \dot{\mathbb{B}}\}$.
3: Solve SRT on $\dot{A}$ and $\dot{B}$—making sure that every bad translocation in the sorting sequence is legal.

It can be shown, by a minor modification of the algorithm in (Ozery-Flato and Shamir, 2006a), that solving *SRT* with the additional condition that every bad translocation is legal can be done in $O(n^{3/2}\sqrt{\log(n)})$. Step 2 can be performed by enumerating all possible mappings and computing the SRT distance for each. This implies:

Lemma 6. *LSRT can be solved in* $O(\eta! 2^{\eta} n + n^{3/2}\sqrt{\log(n)})$.

Our goal in the rest of this paper is to improve this result by speeding up Step 2 (i.e., finding efficiently an optimal mapping between the centromeres of A and B).

5. CENT-MAPPINGS

Our general strategy will be to iteratively map between two centromeres in A and B and replace them with a regular element until all centromeres in non-telocentric chromosomes are mapped. The resulting instance can be solved using SRT, but the increase in the number of elements may have also increased the solution value. The main effort henceforth will be to guarantee that the overall increase is minimal. For this, we need to study in detail the effect of each mapping step on the the cycle graph $G(A, B)$. Our analysis uses the SRT distance formula (Theorem 1). We shall ignore for now the parameter δ, and focus on the change in the simplified formula $n - c + l$ (N is not changed by mapping operations).

A mapping between two centromeres affects their corresponding black and gray pericentric edges. Let (i, i') and (j, j') be pericentric black and gray edges in $G(A, B)$ respectively. Suppose $cen \in CEN$ is added between i and i' in $\dot{A}$ and between j and j' in $\dot{B}$. In this case, (i, i') and (j, j') in $G(A, B)$ are replaced by the four (paracentric) edges (i, cen), (cen, i'), (j, cen) and (cen, j') in $G(\dot{A}, \dot{B})$. (The first two edges are black, the latter are gray.) We refer to the addition of $cen \in CEN$ between (i, i') and (j, j') as a *cent-mapping* since it maps between two centromeres. Note that for each pair of centromeres in A and B

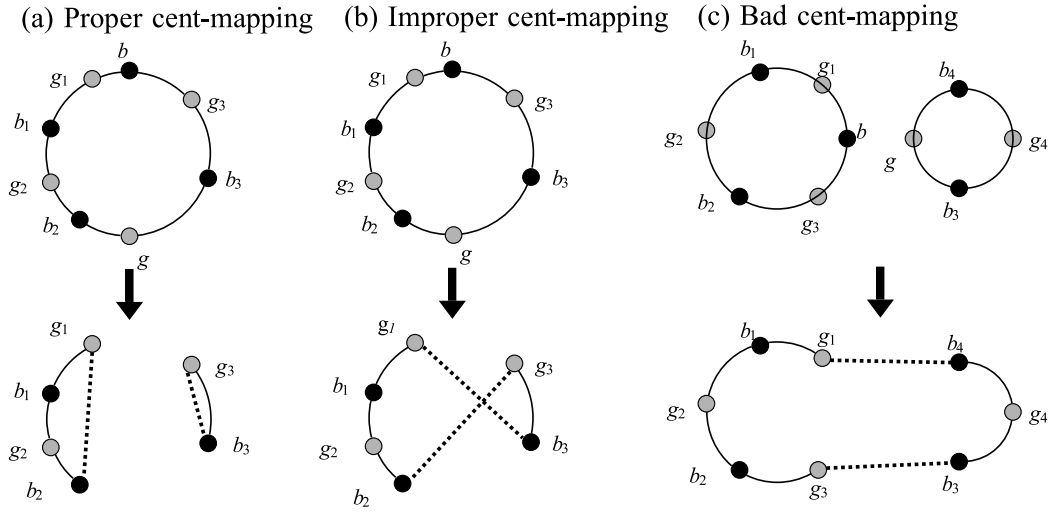


FIG. 4. The effect of a cent-mapping on peri-cycles. Each of the cycles is a peri-cycle with black and gray nodes corresponding to centromeres (pericentric edges) in A and B, respectively. In all cases, a cent-mapping on b and g in the top peri-cycles is performed, and the bottom peri-cycles are the result. Dotted lines denote new edges. **(a,b)** Two alternative cent-mappings of a pair of pericentric edges in the same cycle. **(c)** Each of the two alternatives generates a single cycle.

there are two possible cent-mappings (corresponding to the relative signs of the added elements). Given $\dot{A} \in \dot{\mathbb{A}}$, every $\dot{B} \in \dot{\mathbb{B}}$ defines η disjoint cent-mappings and vice versa. Obviously, every cent-mapping increases the number of genes by one ($\Delta n = +1$).

Lemma 7. *Every cent-mapping satisfies* $\Delta c \in \{-1, 0, 1\}$.

Proof. Let (i, i') and (j, j') be black and gray pericentric edges in $G(A, B)$, respectively. Let $cen \in CEN$ be the element between i and i' in $\dot{A}$. If (i, i') and (j, j') belong to the same cycle before the cent-mapping then $\Delta c \in \{0, 1\}$. If (i, i') and (j, j') belong to different cycles before the cent-mappings then $\Delta c = -1$. ■

In the rest of the paper, we will analyze the effect of a cent-mapping using peri-cycles. A peri-cycle can be viewed as a compact representation of a cycle focused on pericentric edges, which are the only edges affected by cent-mappings. A cent-mapping is called *proper*, *improper*, *bad* if $\Delta c = 1, 0, -1$ respectively. For illustrations of the three types of cent-mappings, see Figure 4. We say that a cent-mapping *operates* on a cycle C if C contains at least one of the mapped pericentric edges. Proper and improper cent-mappings always operate on one cycle in $G(A, B)$; a bad cent-mapping always operates on two different cycles in $G(A, B)$.

Observation 3. *Every proper cent-mapping satisfies* $\Delta l \in \{0, 1\}$. *An improper cent-mapping satisfies* $\Delta l = 0$. *A bad cent-mapping satisfies* $\Delta l \in \{0, -1, -2\}$.

It follows that a proper cent-mapping satisfies $\Delta(n - c + l) = 0$ iff $\Delta l = 0$; An improper cent-mapping satisfies $\Delta(n - c + l) = 1$; a bad cent-mapping satisfies $\Delta(n - c + l) = 0$ iff $\Delta l = -2$. A proper cent-mapping is *safe* if it satisfies $\Delta l = 0$. In the following sections we present two classes of cycles, "annoying" and "evil" for which any set of proper cent-mappings that eliminates all their pericentric edges is unsafe.

5.1. Annoying cycles

In this section we focus on cycles in leaves. The degree of every cycle in a leaf is at most 1 (otherwise it must be external). Moreover, a leaf can contain at most one cycle of degree 1 (for the same reason).

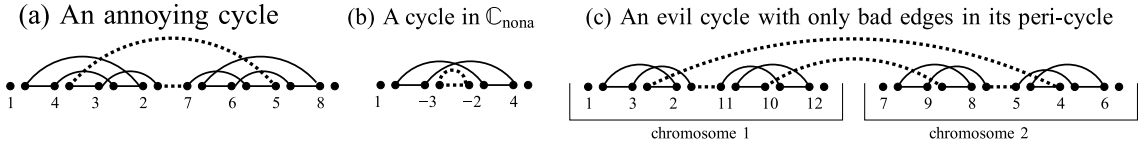


FIG. 5. Examples of cycles in $\mathbb{C}_{ann}$, $\mathbb{C}_{nona}$, and $\mathbb{C}_{evil}$. In all the figures, the target genome B is a fragmented identity permutation (i.e., every gray edge is of the form $(i, i+1)$); pericentric edges are denoted by dotted lines.

A cycle is called *annoying* if: *(i)* it is contained in a leaf, *(ii)* its degree is 1, and *(iii)* a proper cent-mapping on its two pericentric edges satisfies $\Delta l = 1$ (i.e., one leaf is split into two leaves) (Fig. 5a). Thus a proper cent-mapping on an annoying cycle satisfies $\Delta(n - c + l) = 1$. On the other hand, any bad cent-mapping on a cycle contained in the span of a leaf (annoying or not) results in the elimination of that leaf. Thus, a cent-mapping on any two cycles in (two different) leaves satisfies $\Delta(n - c + l) = 1 + 1 - 2 = 0$. Let $\mathbb{C}_{ann}$ denote the set of annoying cycles and let $ann = |\mathbb{C}_{ann}|$. Let $\mathbb{C}_{nona}$ be the set of non-annoying cycles of degree 1 that are contained in the span of a leaf (Fig. 5b). Let $nona = |\mathbb{C}_{nona}|$.

5.2. Evil cycles

In this section we focus on cycles that are not in leaves. Let C be a cycle of degree at least 1 that is not in a leaf and let C^P be its peri-cycle. Let (b, g) be an edge in C^P. Denote by $V(b, g)$ the set of gray edges in the corresponding path between b and g in C. The edge (b, g) is *bad* if after a proper cent-mapping on b and g the edges in $V(b, g)$ belong to a leaf, otherwise it is *good*. For example, in Figure 3, the edge (b, g) where $V(b, g) = \{(1, 2), (2, 3)\}$ is bad.

Lemma 8. *The "badness" of edge (b, g) in a peri-cycle is unchanged by cent-mappings not involving b and g.*

Proof. Clearly the order in which we perform cent-mappings does not affect the final cycle graph. Let M be the component containing $V(b, g)$ in the cycle graph resulting from a proper cent-mapping on (b, g). If M does not contain any pericentric edge in its span, then clearly it is not affected by later cent-mappings. Suppose M contains a pericentric edge in its span. Thus, M must be external since it contains in its span centromeres of two different chromosomes in A. If M is not split by other cent-mappings, then clearly $V(b, g)$ remains in an external component. Suppose M is split into two components by a cent-mapping on pericentric edges b' and g'. In this case, each of the two new components contains in its span one of the two new black edges replacing b'. Hence, the component that contains $V(b, g)$ is guaranteed to remain external, since it contains in its span two different centromeres in A (corresponding to b and b'). ■

Lemma 9. *Let C be a cycle satisfying: (i) $deg(C) > 0$, and (ii) C contains a new gray edge, g_{new}, that was created by a cent-mapping. Let (b, g) be an edge in the peri-cycle of C such that $V(b, g)$ contains g_{new}. Then (b, g) is good.*

Proof. The edge g_{new} is adjacent to a vertex of a previously mapped centromere, $cen_1 \in CEN$. On the other hand, after a cent-mapping on (b, g), the path $V(b, g)$ will be adjacent to a vertex of a new mapped centromere, $cen_2 \in CEN$. These two centromeres belong to different chromosomes of A. Thus $V(b, g)$ must contain an external edge after any cent-mapping of b and g and hence (b, g) is good. ■

A path in a peri-cycle is *bad* if all the edges in it are bad. For a path P, let $len(P)$ denote the number of vertices in P. A cycle C is called *evil* if its peri-cycle contains a bad path P such that $len(P) > deg(C)$. For example, the single cycle in Figure 3 is evil since it contains a bad edge, which is a bad path of length 2, and its degree is 1. An example of an evil cycle with only bad edges in its peri-cycle is presented in Figure 5. Let $\mathbb{C}_{evil}$ denote the set of all evil cycles that are not in leaves. Define $evil = |\mathbb{C}_{evil}|$.

Lemma 10. *Let C be a cycle that does not belong to a leaf. There is a set of safe proper cent-mappings of all the pericentric edges in C iff C is not evil.*

Proof. Let C^P be the peri-cycle of C and let $k = deg(C)$. Suppose C is evil. Then PC contains a bad path P with $k+1$ vertices. There are $2k$ vertices in C^P, thus any proper cent-mapping of all the pericentric edges in C must match two vertices from P. It follows that there must be a proper cent-mapping on the two ends of an edge in P. Hence, by definition this cent-mapping is unsafe.

Suppose C is not evil. If $k = 1$ then the two edges in C^P are good and the proper cent-mapping of the two pericentric edges in C is safe. Suppose $k > 1$. Let $C^P = P_1, P_2$ where P_1 is a longest bad path in C^P. Let u be the first vertex in P_1 and let v be the last vertex in P_2. Then (u, v) is a good edge in C^P. Let C_1 and C_2 be the two cycles created by the proper cent-mapping on u and v, where C_1 contains $V(u, v)$. Obviously this proper cent-mapping is safe, $deg(C_1) = 0$ and $deg(C_2) = k-1$. It suffices to prove that C_2 is not evil. Let C_2^P be the peri-cycle of C_2. Then $C_2^P = P_1'P_2'$ where $len(P_1') = len(P_1)-1$, $len(P_2') = len(P_2)-1$, and P_1' and P_2' are connected by good edges (Lemma 9). Let p be the length of the longest bad path in C_2^P. Then *(i)* $p \leq len(P_1) \leq k$ (since P_1 is a longest bad path in C), *(ii)* $p \leq \max(len(P_1'), len(P_2')) = len(P_2')$, and *(iii)* $len(P_1) + len(P_2) = 2k$. It follows that $p \leq k - 1 = deg(C_2)$. Thus by definition C_2 is not evil. ■

Corollary 2. *Every proper cent-mapping satisfies* $\Delta(l + evil) \geq 0$.

We partition $\mathbb{C}_{\text{evil}}$ into three classes:

- $\mathbb{C}^1_{\text{evil}}$: Cycles of even degree and only bad edges in their peri-cycle.
- $\mathbb{C}^2_{\text{evil}}$: Cycles of odd degree and only bad edges in their peri-cycle.
- $\mathbb{C}^3_{\text{evil}}$: Cycles with at least one good edge in their peri-cycle.

Let $evil_1 = |\mathbb{C}^1_{\text{evil}}|$, $evil_2 = |\mathbb{C}^2_{\text{evil}}|$ and $evil_3 = |\mathbb{C}^3_{\text{evil}}|$. If $C \in \mathbb{C}_{\text{evil}}$ is of degree 1 then $C \in \mathbb{C}^3_{\text{evil}}$ (since otherwise it would be in a leaf). Every new evil cycle (i.e., an evil cycle created by a cent-mapping) contains a good edge (Lemma 9) and hence belongs to $\mathbb{C}^3_{\text{evil}}$. Let $C \in \mathbb{C}^3_{\text{evil}}$ and let (b, g) be an edge opposite to a good edge in the peri-cycle of C. A proper cent-mapping on b and g satisfies $\Delta l = 1$, $\Delta evil = -1$ and hence $\Delta(n - c + l + evil) = 0$. Such a cent-mapping can be viewed as a *replacement* of an evil cycle with a leaf. On the other hand, every proper cent-mapping on a cycle in $\mathbb{C}^1_{\text{evil}} \cup \mathbb{C}^2_{\text{evil}}$ satisfies $\Delta(n - c + l + evil) = \Delta(l + evil) = 1$. Thus by applying proper cent-mappings, a cycle in $\mathbb{C}^2_{\text{evil}} \cup \mathbb{C}^1_{\text{evil}}$ can be replaced by two leaves, where each leaf belongs to a different chromosome.

Lemma 11. *Let* $C \in \mathbb{C}_{evil}$. *There exists an improper cent-mapping on* C *for which* $\Delta evil = -1$ *iff* $C \notin \mathbb{C}^1_{evil}$.

Proof. Let $C \in \mathbb{C}_{\text{evil}}$ and let C^P be its peri-cycle. Suppose that $C \notin \mathbb{C}^1_{\text{evil}}$.

Case 1: $deg(C)$ is odd. Let u and v be two opposite vertices in the peri-cycle of C. Thus u and v have opposite colors. Let C_1 be the cycle obtained from C after an improper cent-mapping between u and v. Then the peri-cycle of C_1 contains two opposite good edges (Lemma 9) and thus C_1 is not evil.

Case 2: $deg(C)$ is even. Then $C \in \mathbb{C}^3_{\text{evil}}$. Let (b, g) be an edge opposite to a good edge in the peri-cycle of C. Let C_1 be the cycle obtained from C after performing an improper cent-mapping between b and g. Then the peri-cycle of C_1 has two opposite good edges and thus C_1 is not evil.

Suppose $C \in \mathbb{C}^1_{\text{evil}}$. Then $deg(C) = k$ is even and every edge in its peri-cycle is bad. Let C_1 be the result of an improper cent-mapping on C. Then $deg(C_1) = k - 1$ and the peri-cycle of C_1 must contain a bad path with at least k vertices. Thus C_1 is evil. ■

In other words: for every cycle in $\mathbb{C}^2_{\text{evil}} \cup \mathbb{C}^3_{\text{evil}}$ there exists an improper cent-mapping satisfying $\Delta(n - c + l + evil) = 0$; Every improper cent-mapping on a cycle in $\mathbb{C}^1_{\text{evil}}$ satisfies $\Delta(n - c + l + evil) = 1$. It follows that a cent-mapping on $C \in \mathbb{C}^1_{\text{evil}} \cup \mathbb{C}_{\text{ann}}$ satisfies $\Delta(n - c + l + evil) = 0$ only if it is bad. Therefore, Corollary 2 and Lemma 11 imply:

Corollary 3. *For every cent-mapping* $\Delta(n - c + l + evil) \geq 0$.

5.3. A polynomial algorithm using at most opt + 2 translocations

In this section we present upper and lower bounds for the legal translocation distance. These bounds provide an intuition for the rather complicated formula for the legal translocation distance presented in the

next section. The proof of the upper bound implies an approximation algorithm that sorts A into B using at most $d(A, B) + 2$ legal translocations.

Lemma 12. *Let $C_1, C_2 \in \mathbb{C}_{evil} \cup \mathbb{C}_{ann}$, where $deg(C_1) \leq deg(C_2)$. If $deg(C_1) = deg(C_2)$ then every bad cent-mapping on C_1 and C_2 satisfies $\Delta(l + evil) = -2$. If $deg(C_1) < deg(C_2)$ there exists a bad cent-mapping on C_1 and C_2 satisfying $\Delta(l + evil) = -2$ iff $C_2 \in \mathbb{C}^3_{evil}$.*

Proof. If $deg(C_1) = deg(C_2)$ then any bad cent-mapping on C_1 and C_2 results in a cycle whose peri-cycle contains two opposite good edges and hence non-evil. Suppose $k_1 = deg(C_1) < deg(C_2) = k_2$ and let C_1^P and C_2^P denote the peri-cycles of C_1 and C_2 respectively.

Case 1: $C_2 \in \mathbb{C}^3_{\text{evil}}$. Let (b, g) be the opposite edge of a good edge in C_2^P. Let C_3 be a result of a (bad) cent-mapping of the b and a vertex of an opposite color in C_2^P. Let P' be a longest bad path in the peri-cycle of C_3. Then $len(P') \leq \max\{k_2, 2k_1 - 1\} \leq k_2 + k_1 - 1 = deg(C_3)$.

Case 2: $C_2 \notin \mathbb{C}^3_{\text{evil}}$. In this case all the edges in C_2^P are bad. Let C_3 be the result of a bad cent-mapping on C_1 and C_2. Then the peri-cycle of C_3 contains a bad path with $2k_2 - 1$ vertices, while $deg(C_3) = k_1 + k_2 - 1 < 2k_2 - 1$. Thus C_3 is evil. ■

The *bad cent-mappings* graph, *BCM*, is defined as follows. It is a bipartite graph whose two parts are *DEG* and *CYC*, where:

$$DEG = \{i : |\{C : C \in \mathbb{C}^1_{\text{evil}} \cup \mathbb{C}_{\text{ann}}, \deg(C) = i\}| \text{ is odd}\} \qquad CYC = \mathbb{C}^3_{\text{evil}} \cup \mathbb{C}_{\text{nona}}$$

For example, if the degrees of the cycles in $\mathbb{C}^1_{\text{evil}} \cup \mathbb{C}_{\text{ann}}$ are $\{1, 2, 2, 2, 4, 4, 6, 8\}$ then $DEG = \{1, 2, 6, 8\}$. Vertices $i \in DEG$ and $C \in CYC$ are connected by an edge if $deg(C) \geq i$ (Fig. 6). Thus an edge (i, C) represents a bad cent-mapping operating on C and $C' \in \mathbb{C}^1_{\text{evil}} \cup \mathbb{C}_{\text{ann}}$, where $\deg(C') = i$, for which $\Delta(n - c + l + evil) = 0$ and $\Delta|DEG| = -1$.

A *matching* in a graph is a collection of edges no two of which share a common vertex. The size of a matching M, denoted $|M|$, is the number of edges in it. Finding a maximum matching in *BCM* is an easy task that can be completed in linear time by a greedy algorithm that iteratively matches vertices from *CYC* in increasing order of their degrees. Define $fbad = |DEG| - |M|$, where M is a maximum matching. For a matching M let F_M be the forest of internal components after performing a bad cent-mapping on every $C \in \mathbb{C}_{\text{ann}} \cup M$. In other words, F_M is obtained from F by the deletion of every component containing a cycle from either $\mathbb{C}_{\text{ann}}$ or $\mathbb{C}_{\text{nona}} \cap M$ in its span. In the following we prove that the cent-mappings produced by Algorithm 2 lead to a sorting scenario of at most $d(A, B) + 2$ legal translocations.

Observation 4. *Every cent-mapping satisfies $\Delta\lceil fbad/3 \rceil \in \{-1, 0, 1\}$.*

Proof. Every cent-mapping involves at most three cycles (old and new). Hence $\Delta fbad \in [-3, 3]$. ■

Lemma 13. *Every cent-mapping satisfies $\Delta(n - c + l + evil + \lceil fbad/3 \rceil) \geq 0$.*

Proof. Let $\Delta \equiv \Delta(n - c + l + evil + \lceil fbad/3 \rceil)$. By Observation 4, if $\Delta(n - c + l + evil) > 0$ then $\Delta \geq 0$. Suppose $\Delta(n - c + l + evil) = 0$. We shall prove that $\Delta fbad \geq 0$:

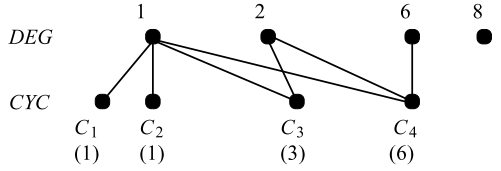


FIG. 6. An example for a bad cent-mappings (*BCM*) graph. $DEG = \{1, 2, 6, 8\}$, $CYC = \{C_1, C_2, C_3, C_4\}$. The degree of each cycle in *CYC* appears in brackets below the cycle.

Algorithm 2. Get_Mapping (a 2-additive approximation)

$M \leftarrow$ a maximum matching in *BCM*
Perform a bad cent-mapping on every $C_1, C_2 \in \mathbb{C}^1_{\text{evil}} \cup \mathbb{C}_{\text{ann}}$, where $deg(C_1) = deg(C_2)$.
/* *Now* $|\mathbb{C}^1_{evil} \cup \mathbb{C}_{ann}| = |DEG|$ */
for all $(i, C) \in M$ **do**
    Perform a bad cent-mapping on C and $C' \in \mathbb{C}^1_{\text{evil}} \cup \mathbb{C}_{\text{ann}}$, where $\deg(C') = i$, such that $\Delta(l + evil) = -2$ (Lemma 12).
end for
while $|DEG| \geq 3$ **do**
    $C_1, C_2, C_3 \leftarrow$ 3 cycles in $\mathbb{C}^1_{\text{evil}} \cup \mathbb{C}_{\text{ann}}$, where $\deg(C_1)$ is minimal.
    Perform a bad cent-mapping on C_2 and C_3 and let C_4 be the new evil cycle.
    Perform a bad cent-mapping on C_1 and C_4 such that $\Delta(l + evil) = -2$ (Lemma 12).
end while
if $|DEG| = 2$ **then**
    Perform a bad cent-mapping on $C, C' \in \mathbb{C}^1_{\text{evil}} \cup \mathbb{C}_{\text{ann}}$. /* $DEG = 2 \rightarrow DEG = 1$ */
end if
if $|DEG| = 1$ **then**
    Perform an improper cent-mapping on $C \in \mathbb{C}^1_{\text{evil}} \cup \mathbb{C}_{\text{ann}}$.
end if
/* *Now* $|\mathbb{C}^1_{evil}| = ann = 0$ */
Perform an improper cent-mapping on every $C \in \mathbb{C}_{\text{evil}}$ such that $\Delta evil = -1$ (Lemma 11).
/* *Now* $evil = 0$ */
Perform safe proper cent-mappings on every cycle of degree at least 1 (Lemma 10).
Perform a proper cent-mapping on every $C \in \mathbb{C}_{\text{nona}}$.

Case 1: $\Delta(n - c) = 0$ (i.e., proper cent-mapping). Then $\Delta(l + evil) = 0$ and thus either $\Delta l = 1$ and $\Delta evil = -1$, or $\Delta l = \Delta evil = 0$. Hence *DEG* is unchanged and $\Delta|CYC| \leq 0$. Therefore, $\Delta fbad \geq 0$.

Case 2: $\Delta(n - c) = 1$ (i.e., improper cent-mapping). Then $\Delta l = 0$ and $\Delta evil = -1$. Therefore *DEG* is unchanged, $\Delta|CYC| \leq 0$, and hence $\Delta fbad >= 0$.

Case 3: $\Delta(n - c) = 2$ (i.e., bad cent-mapping). Then $\Delta(l + evil) = -2$. Let C_1 and C_2 be the cycles on which the cent-mapping was performed. If C_1 and C_2 belong to the same class (e.g., $\mathbb{C}^1_{\text{evil}}$, $\mathbb{C}^3_{\text{evil}}$) then clearly *DEG* is unchanged and $\Delta|CYC| \leq 0$, hence $\Delta fbad \geq 0$. If C_1 and C_2 belong to different classes, then w.l.o.g. $C_1 \in \mathbb{C}^1_{\text{evil}} \cup \mathbb{C}_{\text{ann}}$ and $C_2 \in \mathbb{C}^3_{\text{evil}} \cup \mathbb{C}_{\text{nona}}$. Hence, $\Delta fbad \geq 0$. ■

Lemma 14. *Every cent-mapping performed by Algorithm 2 satisfies* $\Delta(n - c + l + evil + \lceil fbad/3 \rceil) = 0$.

Theorem 3. *Let* $d = d(A, B)$ *and let* $f = n - N - c + l + evil + \lceil fbad/3 \rceil$. *Then* $d \in [f, f + 2]$. *In particular, Algorithm 2 produces* $\dot{A} \in \dot{\mathbb{A}}$ *and* $\dot{B} \in \dot{\mathbb{B}}$ *for which* $d(\dot{A}, \dot{B}) \leq d + 2$.

Proof. Let $\dot{A} \in \dot{\mathbb{A}}$. For every $\dot{B} \in \dot{\mathbb{B}}$, $evil(\dot{A}, \dot{B}) = fbad(\dot{A}, \dot{B}) = 0$ and thus by Theorem 1, $d_{\text{old}}(\dot{A}, \dot{B}) = f(\dot{A}, \dot{B}) + \delta(\dot{A}, \dot{B})$. By Lemma 13, $f(A, B) \leq \min_{\dot{B} \in \dot{\mathbb{B}}}\{f(\dot{A}, \dot{B})\}$. By Theorem 2, $d(A, B) = \min\{f(\dot{A}, \dot{B}) + \delta(\dot{A}, \dot{B}) : \dot{B} \in \dot{\mathbb{B}}\}$. Hence $f(A, B) \leq d(A, B)$. Let $\dot{B}$ be the genome defined by the cent-mappings produced by Algorithm 2. By Lemma 14, $f(A, B) = f(\dot{A}, \dot{B})$. Therefore, $d(A, B) \leq d_{\text{old}}(\dot{A}, \dot{B}) = f(A, B) + \delta(\dot{A}, \dot{B}) \leq f(A, B) + 2$. ■

6. A POLYNOMIAL ALGORITHM FOR THE LEGAL TRANSLOCATION DISTANCE

In this section we present an exact formula for the legal translocation distance, which leads to a polynomial algorithm for the problem. The proof, and subsequently the algorithm, is focused on finding an

optimal mapping between the centromeres of genomes A and B (Step 2 in Algorithm 1). This requires an involved case analysis, which is deferred to an appendix. Let M be a maximum matching in the *BCM* graph. Denote by l_M be the number of leaves in F_M. Define $fgood(M) = |\mathbb{C}^3_{\text{evil}} \setminus M|$. Define $mbad = fbad \bmod 3$. Define $\delta' \in \{0, 1, 2\}$ as follows. $\delta' = 2$ iff all the following conditions are satisfied:

- $\mathbb{C}^2_{\text{evil}} = \mathbb{C}^3_{\text{evil}} = DEG = \emptyset$
- $|F_\emptyset| = 1$
- l and ann are even. If $ann > 0$ then $nona = 0$

If $\delta' \neq 2$ then $\delta' = 1$ iff for every maximum matching M all the following conditions are satisfied:

- $fgood(M) \in \{0, 1\}$
- l_M is even $\Rightarrow F_M = 1$
- (l_M is odd and $fgood(M) = 1$) $\Rightarrow C \in \mathbb{C}^3_{\text{evil}} \setminus M$ cannot be replaced by a leaf such that $|F_M| > 1$.
- $mbad = 1 \Rightarrow DEG = \{1\}$, $|F| = 1$, and ($l_\emptyset$ is odd $\Rightarrow evil_2 = 0$)
- $mbad = 2 \Rightarrow l_M$ is even and $fgood(M) = 0$

If $\delta' \neq 1, 2$ then $\delta' = 0$. Note that if $\delta' = 1$ and $mbad \in \{1, 2\}$ then $|F_M| = 1$.

Theorem 4. *The legal translocation distance between A and B is $d(A, B) = n - N - c(A, B) + l(A, B) + evil(A, B) + \lceil fbad(A, B)/3 \rceil + \delta'(A, B)$.*

The proof of Theorem 4, which appears in the appendix, is by a case analysis of the change in each of the parameters, $n - c$, l, *evil*, *fbad* and δ', for each cent-mapping, and hence is quite involved. It leads to a polynomial time algorithm for finding an optimal mapping between the centromeres of A and B. This algorithm, which can be viewed as an extension of Algorithm 2, has the same time complexity as Algorithm 2.

Theorem 5. *LSRT can be solved in $O(\eta n + n^{3/2}\sqrt{\log(n)})$ time.*

Proof. Finding an optimal mapping between the centromeres of A and B can be done in $O(\eta n)$ in the following manner. The set of peri-cycles can be computed in $O(n)$. For every edge in a peri-cycle we compute its "badness" in $O(n)$ by simply performing the corresponding proper cent-mapping. Computing the badness of all the edges thus takes $O(\eta n)$. Computing $\mathbb{C}^1_{\text{evil}}$, $\mathbb{C}^2_{\text{evil}}$, $\mathbb{C}^3_{\text{evil}}$, $\mathbb{C}_{\text{ann}}$, $\mathbb{C}_{\text{nona}}$, and *DEG* requires a simple traversal of all the edges in every peri-cycle. Hence, it can be done in $O(\eta)$. Overall the algorithm performs $O(\eta)$ operations where each can be implemented in $O(n)$ time. ■

7. CONCLUSION

Computational studies in genome rearrangements have overlooked centromeres to date. In this study, we presented a new model for genomes that accounts for centromeres. Using this model, we defined the problem of legal sorting by reciprocal translocations (LSRT) and proved that it can be solved in polynomial time. Unfortunately, the legal translocation distance formula appears to be quite complex and it is an interesting open problem whether it or its proof can be simplified.

A solvable LSRT instance requires the two input genomes to be co-tailed and with the same set of elements (see Section 3.1). This requirement is a rather strong and unrealistic. Allowing for reversals, non-reciprocal translocations, fissions and fusions will cancel these restrictions. Under a centromere-aware model, fissions and fusions are legal if they are centric (Perry et al., 2004; Searle, 1998). In future work, we intend to study an extension of LSRT that allows for reversals, (centric) fusions and fissions. We expect an exact algorithm for this extended problem to bring us nearer to realistic rearrangement scenarios than can be done today.

8. APPENDIX

Proof of Theorem 4

The proof follows directly from Lemmas 15 and 16 below: Lemma 15 provides a lower bound for the legal distance while Lemma 16 proves this bound is tight.

Lemma 15. *Let* $\Delta = \Delta(n - c + l + evil + \lceil fbad/3 \rceil + \delta')$. *For every cent-mapping* $\Delta \geq 0$.

Proof. In the following "before" and "after" are used to define the state before and after the current cent-mapping respectively. However, unless specified otherwise, every condition refers to the state before the cent-mapping. For example, "l_M is odd" means "l_M is odd before." Let $\mathbb{C}_{\text{good}}$ be the set of cycles that are not in $\mathbb{C}_{\text{evil}} \cup \mathbb{C}_{\text{ann}} \cup \mathbb{C}_{\text{nona}}$. Following Lemma 13, if $\Delta\delta' \geq 0$ then $\Delta \geq 0$. Thus it suffices to prove $\Delta \geq 0$ only for $\delta' \in \{1, 2\}$.

Case 1: $\delta' = 2$. Then $\Delta fbad \geq 0$, since $DEG = \emptyset$.

Case 1.1: $\Delta(n - c) = 0$. Let C be the cycle on which the cent-mapping was performed. Since $\delta' = 2$ then $C \notin \mathbb{C}^3_{\text{evil}} \cup \mathbb{C}^2_{\text{evil}}$.
- $C \in \mathbb{C}_{\text{nona}}$. Then no other parameter is affected and $\Delta\delta' = 0$.
- $C \in \mathbb{C}^1_{\text{evil}} \cup \mathbb{C}_{\text{ann}}$. Then $\Delta(l + evil) = 1$, $\Delta\lceil fbad/3 \rceil = 1$, and hence $\Delta \geq 0$.
- $C \in \mathbb{C}_{\text{good}}$. If $\Delta(l + evil) = 0$ then no other parameter is affected and $\Delta = 0$. If $\Delta(l + evil) = 2$ then clearly $\Delta \geq 0$. Suppose $\Delta(l + evil) = 1$. Note that DEG is unchanged (i.e., $DEG = \emptyset$ after). Hence $mbad = 0$ after. If $\Delta l = 1$ then after: $l_\emptyset$ is odd and $CYC = \emptyset$. If $\Delta evil = 1$ then after $l_\emptyset$ is even and $F|\emptyset| = 1$ (since F is unchanged). Thus, in either case $\Delta = 0$.

Case 1.2: $\Delta(n - c) = 1$ (i.e., an improper move). Let C be the cycle on which the cent-mapping was performed.
- $C \in \mathbb{C}^1_{\text{evil}} \cup \mathbb{C}_{\text{ann}}$. Then $\Delta(l + evil) = 0$, $\Delta\lceil fbad/3 \rceil = 1$, and hence $\Delta \geq 0$.
- $C \in \mathbb{C}_{\text{nona}}$. Then no other parameter is affected and hence $\Delta = 1$.
- $C \in \mathbb{C}_{\text{good}}$. Then $\Delta l = 0$, $\Delta evil \in \{0, 1\}$ and in either case $\Delta = 1$.

Case 1.3: $\Delta(n-c) = 2$. Let C_1 and C_2 be the two peri-cycles on which the cent-mapping was performed. If $\deg(C_1) = \deg(C_2)$ then C_1 and C_2 belong to the same class (either $\mathbb{C}^1_{\text{evil}}$ or $\mathbb{C}_{\text{ann}}$) and clearly $\Delta\delta' = 0$. Suppose $\deg(C_1) < \deg(C_2)$.
- $C_1, C_2 \in \mathbb{C}_{\text{good}}$. Then $\Delta = 2$.
- $C_1 \in \mathbb{C}_{\text{good}}$, $C_2 \in \mathbb{C}^1_{\text{evil}} \cup \mathbb{C}_{\text{ann}}$. Then $\Delta(l + evil) \in \{0, -1\}$. If $\Delta(l + evil) = 0$ then $C_2 \in \mathbb{C}^1_{\text{evil}}$ and hence $\Delta fbad = 0$. If $\Delta(l + evil) = -1$ then $\Delta fbad = 1$. Hence, in either case, $\Delta \geq 0$.
- $C_1 \in \mathbb{C}_{\text{good}}$, $C_2 \in \mathbb{C}_{\text{ann}} \cup \mathbb{C}_{\text{nona}}$. Then $\Delta l = -1$, $\Delta evil = 0$ (the new cycle is in $\mathbb{C}_{\text{good}}$). If $C_2 \in \mathbb{C}_{\text{ann}}$ then $\Delta fbad = 1$ and hence $\Delta \geq 0$. Suppose $C \in \mathbb{C}_{\text{nona}}$. Then $\Delta fbad = 0$, and after: $mbad = 0$, $l_\emptyset$ is odd, and $fgood(\emptyset) = evil_3 = 0$. Hence $\delta' = 1$ after and thus $\Delta = 0$.
- $C_1 \in \mathbb{C}^1_{\text{evil}}$, $C_2 \in \mathbb{C}^1_{\text{evil}} \cup \mathbb{C}_{\text{ann}}$ (different degrees). Then $\Delta(l + evil) = -1$, $\Delta\lceil fbad/3 \rceil = 1$ and hence $\Delta \geq 0$.
- $C_1 \in \mathbb{C}^1_{\text{evil}}$, $C_2 \in \mathbb{C}_{\text{nona}}$. Then $\Delta l = -1$, and the new resulting cycle, C_3 satisfies $C_3 \in \mathbb{C}^3_{\text{evil}}$ and $\deg(C_3) = \deg(C_1)$. Hence $\Delta evil = 0$, $\Delta fbad = 0$, and $\Delta\delta' = -1$. Hence $\Delta = 0$.

Case 2: $\delta' = 1$. If $\Delta(n - c + l + evil + \lceil fbad/3 \rceil) \geq 1$ then clearly $\Delta \geq 0$. We shall prove that if $\Delta(n - c + l + evil + \lceil fbad/3 \rceil) = 0$ then $\Delta\delta' \geq 0$ and thus $\Delta \geq 0$.

Case 2.1: $\Delta(n - c) = 0$. Then $\Delta(l + evil) \geq 0$ (Corollary 2), $\Delta(l + evil + \lceil fbad/3 \rceil) \geq 0$ (Lemma 13). If $\Delta(l + evil + \lceil fbad/3 \rceil) > 0$ then clearly $\Delta \geq 0$. Suppose $\Delta(l + evil + \lceil fbad/3 \rceil) = 0$.
- Suppose $\Delta(l + evil) = 0$. Then $\Delta\lceil fbad/3 \rceil = 0$, $C_1 \in \mathbb{C}_{\text{good}} \cup \mathbb{C}^3_{\text{evil}} \cup \mathbb{C}_{\text{nona}}$. If $C \in \mathbb{C}_{\text{good}}$ then no parameter is affected and hence $\Delta = 0$. Suppose $C \in \mathbb{C}^3_{\text{evil}} \cup \mathbb{C}_{\text{nona}}$. Then $\Delta fbad \in \{0, 1\}$ and $mbad \in \{0, 2\}$.
  - —Suppose $fbad = 0$, $\Delta l = 1$. Then $C \in \mathbb{C}^3_{\text{evil}}$ and $\Delta evil = -1$. Thus for every maximum matching M after, there exists a maximum matching M' before satisfying $fgood(M') =$

$fgood(M) + 1$ and $l_{M'} = l_M - 1$. Since $\delta' = 1$ before it follows that $mbad = 0$ and $\Delta\delta' \geq 0$.

—Suppose $fbad = 0$, $\Delta l = 0$. If $C \in \mathbb{C}_{\text{nona}}$ then every maximum matching after is a maximum matching before, with the same properties. Suppose $C \in \mathbb{C}^3_{\text{evil}}$. Then C is replaced with an evil cycle C' of a smaller degree. Hence for every maximum matching M' after there exists a maximum matching M before, where C' is replaced by C, and which has the same properties as M. Hence in both cases $\Delta\delta' \geq 0$.

—Suppose $fbad = 1$. Then $mbad = 2$ before and $mbad = 0$ after.

* Suppose $C \in \mathbb{C}^3_{\text{evil}}$. If $\Delta l = 1$ (and hence $\Delta evil = -1$) then every maximum matching M after satisfies l_M is odd and $fgood(M) = 0$. If $\Delta l = 0$ then every maximum matching M after satisfies either (l_M is even and $|F_M| = 1$) or (l_M is odd and $fgood(M) = 0$). Hence, in any case $\Delta\delta' \geq 0$.
* Suppose $C \in \mathbb{C}_{\text{nona}}$. Then every maximum matching M after satisfies l_M is odd and $fgood(M)$ is even. Hance $\delta' = 1$ after.

- Suppose $\Delta(l + evil) = 1$. Then $\Delta\lceil fbad/3\rceil = -1$.

—Suppose $\Delta fbad = -1$. Then $mbad = 1$ before and thus $evil_3 = nona = 0$ and $C \in \mathbb{C}_{\text{good}} \cup \mathbb{C}^2_{\text{evil}}$. It follows that every maximum matching M after satisfies either (l_M is even and $|F_M| = 1$) or (l_M is odd and $fgood(M) = 0$). (The later happens only if $C \in \mathbb{C}^2_{\text{evil}}$ and $\Delta l = 1$.) Hence $\Delta\delta' \geq 0$.

—Suppose $\Delta fbad = -2$. Then $mbad = 2$ before and $C \in \mathbb{C}^2_{\text{evil}} \cup \mathbb{C}^1_{\text{evil}}$. Moreover, if $C \in \mathbb{C}^1_{\text{evil}}$ then $\deg(C) \in DEG$. Then for every maximum matching M after either (l_M is even and $|F_M| = 1$) or (l_M is odd and $fgood(M') = 0$). (The latter case may happen only if $C \in \mathbb{C}^1_{\text{evil}}$.) Hence $\Delta\delta' = 0$.

—Suppose $\Delta fbad = -3$. Then $C \in \mathbb{C}^1_{\text{evil}}$ and for every maximum matching M after there exists a maximum matching M' before with the same properties. Hence $\Delta\delta' = 0$.

Case 2.2: Suppose $\Delta(n-c) = 1$. Then $\Delta l = 0$ and $\Delta(evil + \lceil fbad/3\rceil) \geq -1$. If $\Delta(evil + \lceil fbad/3\rceil) \geq 0$ then clearly $\Delta \geq 0$. Suppose $\Delta(evil + \lceil fbad/3\rceil) = -1$. Let C the cycle on which the cent-mapping was performed.

- Suppose $\Delta evil = -1$. Then $\Delta\lceil fbad/3\rceil = 0$, $C \in \mathbb{C}^3_{\text{evil}} \cup \mathbb{C}^2_{\text{evil}}$, F is unchanged. If $C \in \mathbb{C}^2_{\text{evil}}$ then clearly $\Delta\delta' \geq 0$. Suppose $C \in \mathbb{C}^3_{\text{evil}}$. Then $\Delta fbad \in \{0, 1\}$.

—Suppose $\Delta fbad = 0$. Then for every maximum matching M after there exists a maximum matching M' before such that $F_M = F_{M'}$ and $fgood(M) = fgood(M') - 1$. Hence $\Delta\delta' \geq 0$.

—Suppose $\Delta fbad = 1$. Then before $mbad = 2$. It follows that after: $mbad = 0$ and every maximum matching M satisfies $|F_M| = 1$ and l_M is even. Hence $\delta' = 1$ after.

- Suppose $\Delta evil = 0$. Then $\Delta\lceil fbad/3\rceil = -1$, $C \in \mathbb{C}^2_{\text{evil}} \cup \mathbb{C}^1_{\text{evil}} \cup \mathbb{C}_{\text{ann}}$.

—Suppose $C \in \mathbb{C}^2_{\text{evil}}$. Then before $mbad = 1$ and hence after: $mbad = 0$ and the single maximum matching satisfies l_M is even and $|F_M| = 1$. Hence $\delta' = 1$ after.

—Suppose $C \in \mathbb{C}^1_{\text{evil}}$. Then $\deg(C) \in DEG$, F is unchanged, and $mbad = 2$ before. Hence after: $mbad = 0$ and every maximum matching M satisfies l_M is even and $|F_M| = 1$. Hence $\delta' = 1$ after.

—Suppose $C \in \mathbb{C}_{\text{ann}}$. Then $mbad = 1$ before. Therefore after $DEG = \emptyset$ and $\Delta\delta' \geq 0$.

Case 2.3: $\Delta(n-c) = 2$. Let C_1 and C_2 be the cycles on which the cent-mapping was performed. In this case $\Delta|F| \leq 0$, $\Delta(l+evil) \geq -2$, $\Delta(l+evil+\lceil fbad/3\rceil) \geq -2$. If $\Delta(l+evil+\lceil fbad/3\rceil) \geq -1$ then clearly $\Delta \geq 0$. Suppose $\Delta(l + evil + \lceil fbad/3\rceil) = -2$.

- Suppose $\Delta(l + evil) = -1$. Then $\Delta\lceil fbad/3\rceil = -1$.

—Suppose $\Delta fbad = -1$. Then $mbad = 1$ before, $C_1 \in \mathbb{C}_{\text{ann}}$, $C_2 \in \mathbb{C}_{\text{good}} \cup \mathbb{C}^2_{\text{evil}}$. Hence after: $mbad = 0$, $DEG = \emptyset$, $|F_\emptyset| = 1$ ($F_\emptyset$ is unchanged). If $l_\emptyset$ is even then clearly $\Delta\delta' \geq 0$. Suppose $l_\emptyset$ is odd. Then $C \in \mathbb{C}_{\text{good}}$ and hence $fgood(\emptyset) = 0$ after. Therefore $\Delta\delta' \geq 0$.

—Suppose $\Delta fbad = -2$. Then $mbad = 2$ before and $mbad = 0$ after. Note that before F_M is fixed for every maximum matching M (i.e., $F_M = F'$). Let M be a maximum matching after. Then either $F_M = F'$ (i.e., as before), or l_M is odd and $fgood(M) = 0$.

(The latter may happen only if $nona > 0$ and $C_1 \in \mathbb{C}_{\text{ann}} \cup \mathbb{C}_{\text{nona}}$.) In both cases $\delta' = 1$ after.

—Suppose $\Delta fbad = -3$. Then $C_1, C_2 \in \mathbb{C}^1_{\text{evil}} \cup \mathbb{C}_{\text{ann}}$, $\deg(C_1), \deg(C_2) \in DEG$, and for every maximum matching M after, there exists a maximum matching M' before, such that $F_M = F_{M'}$ and $fgood(M) = fgood(M')$, hence $\delta' = 1$ after.

- Suppose $\Delta(l + evil) = -2$. Then $\Delta\lceil fbad/3 \rceil = 0$ and only the following cases are possible.

—$C_1 \in \mathbb{C}^3_{\text{evil}}$, $C_2 \in \mathbb{C}^2_{\text{evil}}$. Then $\Delta fbad \in \{0, 1\}$. If $\Delta fbad = 0$ then for every maximum matching M after there exists a maximum matching M' before such that $F_M = F_{M'}$ and $fgood(M) = fgood(M') - 1$, hence $\Delta\delta' \geq 0$. Suppose $\Delta fbad = 1$. Then $\Delta mbad = 2$ before. Hence after: $mbad = 0$, and every maximum matching satisfies l_M is even and $|F_M| = 1$, hence $\Delta\delta' = 0$.

—$C_1 \in \mathbb{C}^3_{\text{evil}}$, $C_2 \in \mathbb{C}^1_{\text{evil}} \cup \mathbb{C}_{\text{ann}}$.

* $\deg(C_2) \in DEG$. Then $\Delta fbad \in \{0, 1\}$. If $\Delta fbad = 0$ then clearly $\Delta \geq 0$. Suppose $\Delta fbad = 1$. Then $mbad = 2$ before and after: $mbad = 0$, and either (l_M is even and $|F_M| = 1$, or (l_M is odd and $fgood(M) = 0$). Hence $\Delta\delta' = 0$.

* $\deg(C_2) \notin DEG$. Then $\Delta fbad \in \{0, 1\}$ again. In both cases $C_2 \in \mathbb{C}_{\text{ann}}$, and after $mbad = 0$ and every maximum matching M after satisfies $(1, C') \in M$, where $C' \in \mathbb{C}_{\text{nona}}$, l_M is odd and $fgood(M) = 0$ (since $\delta' = 1$ before). Hence $\Delta\delta' = 0$.

—$C_1 \in \mathbb{C}^3_{\text{evil}}$, $C_2 \in \mathbb{C}_{\text{nona}}$. Then $\Delta fbad \in \{0, 1\}$.

* $\Delta fbad = 0$. Then if $1 \in DEG$ then $nona \geq 2$. Hence for every maximum matching M after there exists a maximum matching M' before such that $l_M = l_{M'} - 1$ and $fgood(M) = fgood(M') - 1$. Thus before: $mbad = 0$ and every maximum matching M' for which $fgood(M') = 1$ satisfied l_M is even. Thus $\Delta\delta' \geq 0$.

* $\Delta fbad = 1$. Then before: $mbad = 2$ and thus $nona = 1$. It follows that $1 \notin DEG$ and hence after: $mbad = 0$, and every maximum matching M satisfies l_M is odd and $fgood(M) = 0$. Thus $\delta' = 1$ after.

—$C_1, C_2 \in \mathbb{C}^2_{\text{evil}}$, or $C_1, C_2 \in \mathbb{C}^1_{\text{evil}}$, or $C_1, C_2 \in \mathbb{C}_{\text{ann}}$. Then clearly $\Delta\delta' \geq 0$.

—$C_1 \in \mathbb{C}_{\text{ann}}$, $C_2 \in \mathbb{C}_{\text{nona}}$. If $1 \in DEG$ then clearly $\Delta\delta' \geq 0$. Suppose $1 \notin DEG$. Then $\Delta fbad \in \{0, 1\}$ and for every maximum matching before $F_M = F'$ and $fgood(M) = fgood'$ are fixed (i.e., independent of M).

* $nona > 1$ before. Then $|F'| > 1$ and hence $mbad = 0$, $l(F')$ is odd and $fgood' = 0$. Thus after, every maximum matching M satisfies: $l_M = l(F') - 2$ is odd and $fgood(M) = fgood' = 0$, and thus $\delta' = 1$.

* $nona = 1$ before. Then after: $nona = 0$ and for every maximum matching M, $F_M = F''$ (i.e., independent of M) and $l(F'') = l(F') - 1$. There there are two possible cases. In the first case $fgood' = 0$ before, and then $\Delta fbad = 1$, and hence $mbad = 2$ before. In the second case $fgood = 1$, and then $\Delta fbad = 0$, $mbad = 0$ and $l(F')$ is even (since F' contains a non-annoying leaf). It follows that in both cases after: $mbad = 0$, $fgood'' = 0$ and $l(F'')$ is odd. Hence $\delta' = 1$ after.

—$C_1, C_2 \in \mathbb{C}_{\text{nona}}$. If $1 \notin DEG$ or $nona > 2$ then clearly $\Delta\delta' \geq 0$. We shall prove that no other case is not possible. Suppose $1 \in DEG$ and $nona = 2$. It follows that before for every maximum matching M, $(1, C) \in M$ where $C \in \mathbb{C}_{\text{nona}}$, l_M is odd and $fgood(M) = 0$. Hence $mbad = 0$ before and $\Delta fbad = 1$, a contradiction to $\Delta\lceil fbad/3 \rceil = 0$. ■

Lemma 16. *Let $\Delta = \Delta(n - c + l + evil + \lceil fbad/3 \rceil + \delta')$. There exists a sequence of η cent-mappings where each satisfies $\Delta = 0$.*

Proof. Below we present Algorithm 3, which satisfies $\Delta = 0$ for every cent-mapping. Moreover, after the run of this algorithm the following conditions are satisfied: *(i)* $DEG = \emptyset$, *(ii)* $\delta' = 0 \Rightarrow l_\emptyset$ is even and $F_\emptyset \neq 1$, and *(iii)* $\delta' = 1 \Rightarrow l_\emptyset$ is odd. It follows that if we apply Algorithm 2 after Algorithm 3, then every cent-mapping performed by the latter algorithm satisfies $\Delta = 0$. (Note that in this case Steps 3–16 in Algorithm 2 are skipped, since $DEG = \emptyset$.) ■

Algorithm 3. Improve δ'

if $mbad = 2$ **then**
  Let M be a maximum matching, let $C_1, C_2 \in \mathbb{C}^1_{\text{evil}} \cup \mathbb{C}_{\text{ann}}$, where $\deg(C_1), \deg(C_2) \notin M$, and $\deg(C_1) \neq \deg(C_2)$. Perform a bad cent-mapping on C_1 and C_2
else if $mbad = 1$ **then**
  $i \leftarrow \max\{j : j \in DEG\}$
  if $i > 1$ **then**
    Let M be a maximum matching where i is not matched. Let $C \in \mathbb{C}^1_{\text{evil}}$ satisfying $\deg(C) = i$
    if l_M is even **then**
      Perform 2 proper cent-mapping on C such that $\Delta l = 2$ and $\Delta evil = -1$
    else
      Perform an improper cent-mapping on C followed by a proper cent-mapping satisfying $\Delta l = 1$, $\Delta evil = -1$ and $\Delta|F_M| > 1$ after
    end if
  else
    if $l_\emptyset = 0$ **then**
      Let $C \in \mathbb{C}_{\text{ann}}$, let $C_1 \neq C$ be any other cycle satisfying $\deg(C_1) > 0$.
      if $C_1 \in \mathbb{C}_{\text{good}} \cup \mathbb{C}^2_{\text{evil}}$ **then**
        Perform a bad cent-mapping on C and C_1
      else
        Then $C_1 \in \mathbb{C}^1_{\text{evil}} \cup \mathbb{C}_{\text{ann}}$. Let C_2 be a cycle of the same class as C_1, different from C and C_1, satisfying $\deg(C_2) = \deg(C_1)$. Perform a bad cent-mapping on C_1 and C_2. Let C_3 be new cycle. Perform a bad cent-mapping on C and C_3
      end if
    else if $|F| > 1$ **then**
      Depending on the parity of $l_\emptyset$: perform either a proper or an improper cent-mapping on a cycle from $\mathbb{C}_{\text{ann}}$ such that after: $l_\emptyset$ is even and $|F_\emptyset| > 1$
    else if $l_\emptyset$ is odd **then**
      if $evil_2 > 0$ **then**
        Let $C' \in \mathbb{C}^2_{\text{evil}}$. Perform a bad cent-mapping on C and C'
      else
        Perform a proper cent-mapping on C
      end if
    else
      Perform an improper cent-mapping on C
    end if
  end if
end if
call Procedure 4

Procedure 4. Handle $mbad = 0$

if $1 \in DEG$ and $nona > 0$ **then**
  Let M be a maximum matching in BCM satisfying $(1, C_1) \in M$, where $C_1 \in \mathbb{C}_{\text{nona}}$
  if $|F_M| = 1$ and $nona \geq 2$ **then**
    Let M be a maximum matching in BCM satisfying $(1, C_2) \in M$ where $C_1 \neq C_2 \in \mathbb{C}_{\text{nona}}$
  end if
else
  Let M be any maximum matching in BCM
end if

(*continued*)

Procedure 4. (*Continued*)

if l_M is odd, and $fgood(M) = 1$, and after $C \in \mathbb{C}^3_{\text{evil}} \setminus M$ is replaced by a leaf $|F_M| = 1$ **then**
  if there exists $i \in DEG$ such that $i \leq \deg(C)$ **then**
    Update M such that $(i, C) \in M$
  end if
end if
if l_M is odd and there exists $C \in \mathbb{C}^3_{\text{evil}} \setminus M$ that can be replaced by a leaf such that $|F_M| > 1$ after **then**
  Perform this replacement
else if l_M is even and $|F_M| = 1$ **then**
  if $fgood(M) \geq 2$ **then**
    Replace two unmatched cycles in $\mathbb{C}^3_{\text{evil}}$ by two leaves (each cycle is replaced by one leaf)
  else if $evil_2 > 0$ **then**
    Replace a cycle in $\mathbb{C}^2_{\text{evil}}$ by two leaves
  else if $fbad > 0$ **then**
    Let $i_1, i_2, i_3 \in DEG \setminus M$, where $i_1 < i_2 < i_3$. Let $C_1, C_2, C_3 \in \mathbb{C}^1_{\text{evil}} \cup \mathbb{C}_{\text{ann}}$, where $\deg(C_j) = j$ for $j = 1, 2, 3$. Perform a bad cent-mapping on C_1 and C_2. Replace C_3 by two leaves
  else if $|M| > 0$ **then**
    Choose $C \in \mathbb{C}^1_{\text{evil}} \cup \mathbb{C}_{\text{ann}}$, $C' \in \mathbb{C}^3_{\text{evil}} \cup \mathbb{C}_{\text{nona}}$ such that $(\deg(C), C') \in M$
    if $\deg(C) = 1$ **then**
      Perform an improper cent-mapping on C
      if $C' \in \mathbb{C}^3_{\text{evil}}$ **then**
        Replace C by a leaf
      end if
    else
      Replace C by two leaves
    end if
  else if $ann > 0$ and $nona > 0$ **then**
    Let $C_1, C_2 \in \mathbb{C}_{\text{ann}}$, $C_3 \in \mathbb{C}_{\text{nona}}$. Perform a proper cent-mapping on C_1. Perform a bad cent-mapping on C_2 and C_3
  else if $\mathbb{C}^3_{\text{evil}} >$ **then**
    Replace $C \in \mathbb{C}^3_{\text{evil}}$ by a leaf
  end if
end if
for all $(i, C) \in M$ **do**
  Perform a bad cent-mapping on C and a $C' \in \mathbb{C}^1_{\text{evil}} \cup \mathbb{C}_{\text{ann}}$, where $\deg(C') = i$, such that $\Delta(l + evil) = -2$ (Lemma 12).
end for

ACKNOWLEDGMENTS

This study was supported in part by the Israeli Science Foundation (grant 309/02).

DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

Bader, D., Moret, B.M., and Yan, M. 2001. A linear-time algorithm for computing inversion distance between signed permutations with an experimental study. *J. Comput. Biol.* 8, 483–491.

Bergeron, A., Mixtacki, J., and Stoye, J. 2006. On sorting by translocations. *J. Comput. Biol.* 13, 567–578.

Hannenhalli, S. 1996. Polynomial algorithm for computing translocation distance between genomes. *Discrete Appl. Math.* 71, 137–151.

Kaplan, H., Shamir, R., and Tarjan, R.E. 2000. Faster and simpler algorithm for sorting signed permutations by reversals. *SIAM J. Comput.* 29, 880–892.

Ozery-Flato, M., and Shamir, R. 2006a. An $O(n^{3/2}\sqrt{\log(n)})$ algorithm for sorting by reciprocal translocations. *Lect. Notes Comput. Sci.* 4009, 258–269.

Ozery-Flato, M., and Shamir, R. 2006b. Sorting by translocations via reversals theory. *Lect. Notes Comput. Sci.* 4205, 87–98.

Ozery-Flato, M., and Shamir, R. 2007. Rearrangements in genomes with centromeres—part I: translocations. *Lect. Notes Comput. Sci.* 4453, 339–353.

Perry, J., Slater, H., and Choo, K.A. 2004. Centric fission–simple and complex mechanisms. *Chromosome Res.* 12, 627–640.

Searle, J. 1998. Speciation, chromosomes, and genomes. *Genome Res.* 8, 1–3.

Sullivan, B., Blower, M., and Karpen, G. 2001. Determining centromere identity: cyclical stories and forking paths. *Nat. Rev. Genet.* 2, 584–596.

Address reprint requests to:
Michal Ozery-Flato
School of Computer Science
Tel-Aviv University
Tel-Aviv 69978, Israel

E-mail: ozery@post.tau.ac.il

Chapter 5

Sorting Cancer Karyotypes by Elementary Operations

JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 16, Number 10, 2009

Pp. 1445–1460
DOI: 10.1089/cmb.2009.0083

Research Article

Sorting Cancer Karyotypes by Elementary Operations

MICHAL OZERY-FLATO and RON SHAMIR

ABSTRACT

Since the discovery of the "Philadelphia chromosome" in chronic myelogenous leukemia in 1960, there has been ongoing intensive research of chromosomal aberrations in cancer. These aberrations, which result in abnormally structured genomes, became a hallmark of cancer. Many studies provide evidence for the connection between chromosomal alterations and aberrant genes involved in the carcinogenesis process. An important problem in the analysis of cancer genomes is inferring the history of events leading to the observed aberrations. Cancer genomes are usually described in the form of karyotypes, which present the global changes in the genomes' structure. In this study, we propose a mathematical framework for analyzing chromosomal aberrations in cancer karyotypes. We introduce the problem of sorting karyotypes by elementary operations, which seeks a shortest sequence of elementary chromosomal events transforming a normal karyotype into a given (abnormal) cancerous karyotype. Under certain assumptions, we prove a lower bound for the elementary distance, and present a polynomial-time 3-approximation algorithm for the problem. We applied our algorithm to karyotypes from the Mitelman database, which records cancer karyotypes reported in the scientific literature. Approximately 94% of the karyotypes in the database, totaling 58,464 karyotypes, supported our assumptions, and each of them was subjected to our algorithm. Remarkably, even though the algorithm is only guaranteed to generate a 3-approximation, it produced a sequence whose length matched the lower bound (and hence optimal) in 99.9% of the tested karyotypes.

Key words: combinatorics, computational molecular biology, gene expression, gene networks, genetic variation, sequence analysis.

1. INTRODUCTION

Cancer is a disease caused by genomic mutations leading to the aberrant function of genes. Those mutations ultimately give cancer cells their proliferative nature. Inferring the evolution of these mutations is an important problem in the research of cancer. Chromosomal mutations that shuffle/delete/duplicate large genomic fragments are common in cancer. Many methods for detection of chromosomal mutations use chromosome painting techniques, such as G-banding, to achieve a visualization of cancer cell genomes. The description of the observed genome organization is called a *karyotype* (Fig. 1). In a karyotype, each chromosome is partitioned into continuous genomic regions called *bands*, and the total number of bands is the *banding resolution*. Over the last decades, a large amount of data has been accumulated on cancer

The Blavatnik School of Computer Science, Tel-Aviv University, Tel-Aviv, Israel.

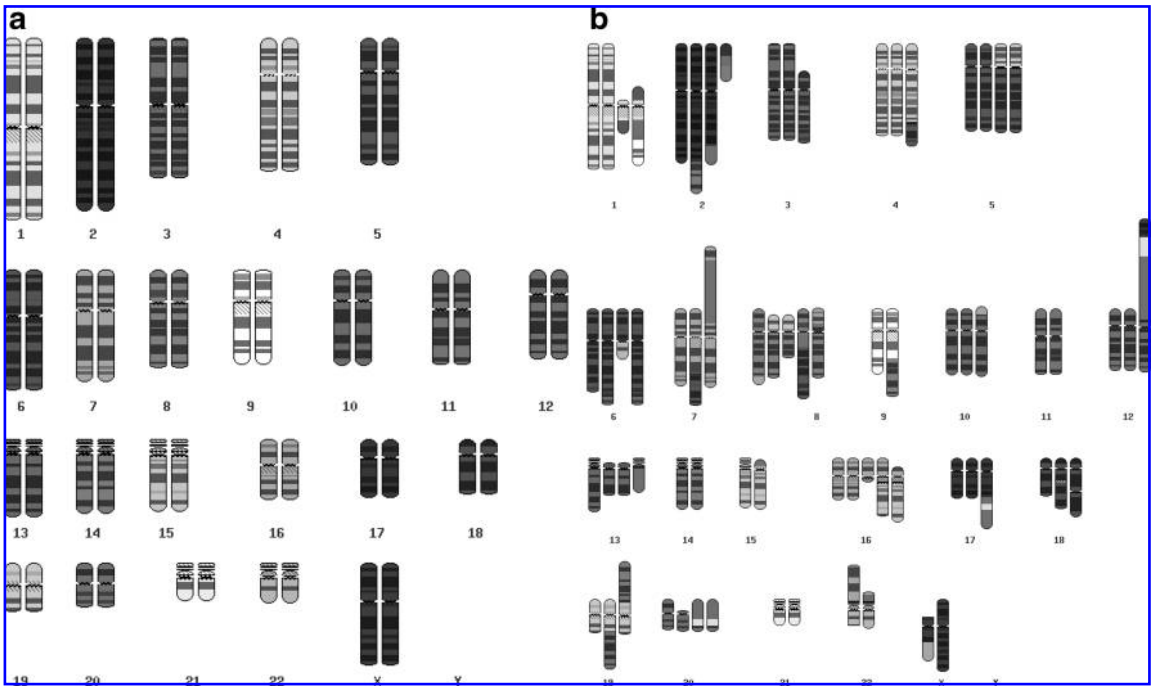


FIG. 1. A schematic view of two real karyotypes: a normal female karyotype (**a**) and the karyotype of MCF-7 breast cancer cell-line (**b**) (NCI, 2001). In the normal karyotype, all chromosomes, except X and Y, appear in two identical copies, and each chromosome has a distinct single color. In the cancer karyotype presented here, only chromosomes 11, 14, and 21 show no chromosomal aberrations.

karyotypes. One of the largest depositories of cancer karyotypes is the Mitelman database of chromosomal aberrations in cancer (Mitelman et al., 2008), which records cancer karyotypes reported in the scientific literature. These karyotypes are described using the ISCN nomenclature (Mitelman, 1995) and thus can be parsed automatically. While novel techniques can provide information at much higher resolution of the cancer karyotypes (Snijders et al., 2001; Greenman et al., 2007), the Mitelman database still contains data on a number of karyotypes a few orders of magnitudes larger.

Cancer karyotypes exhibit a wide range of chromosomal aberrations. The common classification of these aberrations categorizes them into a variety of specific types, such as translocations, and iso-chromosomes. Inferring the evolution of cancer karyotypes using this wide vocabulary of complex alteration patterns is a difficult task. Nevertheless, the entire spectrum of chromosomal alterations can essentially be spanned by four elementary operations: breakage, fusion, duplication, and deletion (Fig. 2). A *breakage*, formally

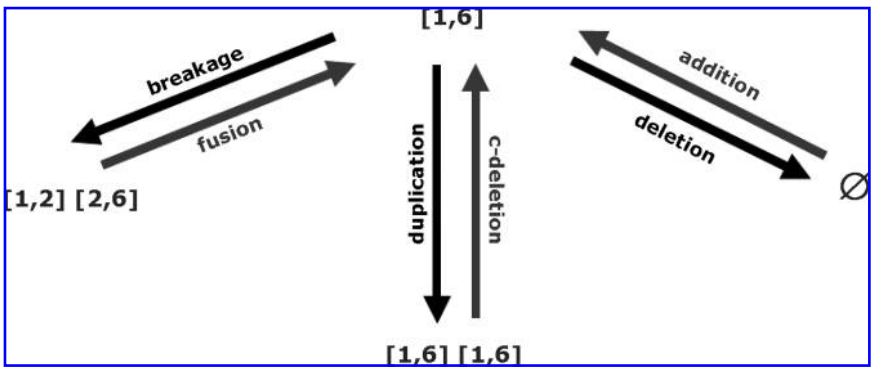


FIG. 2. Illustrations of elementary operations: breakage, fusion, duplication, and deletion. The inverse elementary operations are fusion, breakage, c-deletion, and addition, respectively.

known as a "double-strand break," cuts a chromosomal fragment into two. A *fusion* ligates two chromosomal fragments into one. Genomic breakages, which occur quite frequently in somatic cells, are normally repaired by the corresponding inverse fusion. Mis-repair of genomic breakages is believed to be a major cause of chromosomal aberrations in cancer (Ferguson and Frederick, 2001). Other prevalent chromosomal alterations in cancer genomes are *duplications* and *deletions* of chromosomal fragments. These four elementary events play a significant role in carcinogenesis: fusions and duplications can activate oncogenes, while breakages and deletions can eliminate tumor suppressor genes.

In this article, we introduce a new model for analyzing chromosomal aberrations in cancer based on the four elementary operations presented above. We study the problem of finding a shortest sequence of operations that transforms a normal karyotype into a given cancer karyotype. We call this problem *karyotype sorting by elementary operations* (KS), and the length of a shortest sequence is called the *elementary distance* between the normal and cancer karyotypes. The elementary distance indicates how far, in terms of number of operations, a cancer karyotype is from the normal one. Hence, it corresponds to the complexity of the cancer karyotype, which may give an indication of the tumor phase (Höglund et al., 2005). The reconstructed elementary operations can be used to detect common events for a set of cancer karyotypes and thus point out genomic regions suspect of containing genes associated with carcinogenesis.

Under certain assumptions, which are supported by most cancer karyotypes, the KS problem can be reduced in linear time to a simpler problem, called RKS. For the latter problem, we prove a lower bound for the elementary distance, and present a polynomial-time 3-approximation algorithm. We show that approximately 94% of the karyotypes in the Mitelman database (58,464) support our assumptions, and each of these was subjected to our algorithm. Remarkably, even though the algorithm is only guaranteed to generate a 3-approximation, it produced a sequence whose length matched the lower bound (and hence optimal) in 99.9% of the tested karyotypes. Manual inspection of the remaining cases reveals that the computed sequence for each of these cases is also optimal.

This article is organized as follows. In Section 1, we give the combinatorial formulation of the KS problem and its reduced variant RKS. In the rest of the article, we focus on the RKS problem. In Section 2, we prove a lower bound for the elementary distance for RKS. Section 3 describes our 3-approximation algorithm for RKS. Finally, in Section 4, we present the results of the application of our algorithm to the karyotypes in the Mitelman database.

2. PROBLEM FORMULATION

2.1. The KS problem

The KS problem receives two karyotypes as an input: the normal karyotype, K_{normal}, and the cancer karyotype, K_{cancer}. We represent each of the two karyotypes by a multi-set of chromosomes. Every chromosome in K_{normal} is presented as an interval of B integers, where each integer represents a *band*. For simplicity, we assume that all the chromosomes in K_{normal} share the same B, which corresponds to the banding resolution. Every two chromosomes in the normal karyotype are either identical, i.e., are represented by the same interval, or disjoint. More precisely, we represent every chromosome in K_{normal} by the interval $[(k-1)B+1, kB]$, where k is an integer that identifies the chromosome. The normal karyotype usually contains exactly two copies of each chromosomes, with the possible exception of the sex chromosomes. Every chromosome in K_{cancer} is either a fragment or a concatenation of several fragments, where a *fragment* is a maximal sub-interval, with two bands or more, of a chromosome in the normal karyotype. More formally, a fragment is a maximal interval of the karyotype of the form $[i,j] \equiv [i, i+1, \ldots, j]$, or $[j,i] \equiv [j, j-1, \ldots, i]$, where $i < j, i, j \in \{(k-1)B+1, \ldots, kB\}$, and $[(k-1)B+1, kB] \in K_{\text{normal}}$. Note that, in particular, a chromosome in K_{cancer} can be identical to a chromosome in K_{normal}. We use the symbol "::" to denote a concatenation of two fragments, e.g., $[i,j]::[i',j']$. Every chromosome, in both K_{normal} and K_{cancer}, is orientation-less, i.e., reversing the order of the fragments, and the fragments themselves, results in an equivalent chromosome. For example, $X = [i,j]::[i',j'] \equiv [j',i']::[j,i] = \bar{X}$.

We refer to the concatenation point of two intervals as an *adjacency* if the union of their intervals is equivalent to a larger interval in K_{normal}. In other words, two concatenated intervals that form an adjacency can be replaced by one equivalent interval. For example, the concatenation point in [5, 3]::[3, 1] ≡ [5, 1] is an adjacency. Typically, a breakage occurs within a band, and each of the resulting fragments contains a piece of this broken band that can still be viewed and identified by cytogenetic techniques. For example, if

[5, 1] is broken within band 3, then the resulting fragments are generally denoted the by [5, 3] and [3, 1]. For this reason, we do *not* consider the concatenation [5, 3]::[2, 1] as an adjacency. A concatenation point that is *not* an adjacency, is called a *breakpoint.*[1] Additional examples of concatenation points that are breakpoints are as follows: [1, 3]::[5, 6] and [2, 4]::[4, 3].

We assume that the cancer karyotype, K_{cancer}, has evolved from the normal karyotype, K_{normal}, by the following four *elementary operations* (Fig. 2):

I. **Fusion**: a concatenation of two chromosomes, X_1 and X_2, into one chromosome $X_1::X_2$.
II. **Breakage**: a split of a chromosome into two chromosomes. A split can occur within a fragment, or between two previously concatenated fragments, i.e., in a breakpoint. In the former case, where the break is in a fragment $[i, j]$, the fragment is split into two fragments: $[i, k]$ and $[k, j]$, where $k \in \{i+1, i+2, \ldots, j-1\}$.
III. **Duplication**: a whole chromosome is duplicated, resulting in two identical copies of the original chromosome.
IV. **Deletion**: a complete chromosome is deleted from the karyotype.

Given K_{normal} and K_{cancer}, we define the KS problem as finding a shortest sequence of elementary operations that transforms K_{normal} into K_{cancer}. The length of that sequence is called the *elementary distance* between the karyotypes, and denoted $d(K_{\text{normal}}, K_{\text{cancer}})$. An equivalent formulation of the KS problem is obtained by considering the inverse direction: find a shortest sequence of *inverse* elementary operations that transforms K_{cancer} into K_{normal}. Clearly, fusion and breakage operations are inverse to each other. The inverse to a duplication is a *constrained deletion* (*c-deletion*), where the deleted chromosome is one of two or more identical copies. In other words, a c-deletion can delete a chromosome only if there exists another identical copy of it. The inverse of a deletion is an *addition* of a chromosome. Note that in general, the added chromosome need not be a duplicate of an existing chromosome and can contain any number of fragments. For the rest of the article, we analyze KS by sorting in reverse order, i.e., starting from K_{cancer} and going back to K_{normal}. The sorting sequences will also start from K_{cancer}.

2.2. Reducing KS to RKS

In this section, we present a basic analysis of KS, which together with two additional assumptions, allows the reduction of KS to a simpler variant in which no breakpoint exists (RKS). As we shall see, our assumptions are supported by most analyzed cancer karyotypes.

We start with several definitions. A sequence of inverse elementary operations is *sorting*, if its application to K_{cancer} results in K_{normal}. We shall refer to a shortest sorting sequence as *optimal*. Since every fragment contains two or more bands, we can present any band i within it by an ordered pair of its two ends, i^0, which is the end closer to the minimal band in the fragment, and i^1, the end closer to the maximal band in the fragment. More formally, we map the fragment $[i,j], i \neq j$, to $[i^1, j^0] \equiv [i^1, (i+1)^0, (i+1)^1, \ldots, j^0]$ if $i < j$, and otherwise to $[i^0, j^1] \equiv [i^0, (i-1)^1, (i-1)^0, \ldots, j^1]$. We say that two fragment-ends, a and a', are *complementing* if $\{a, a'\} = \{i^0, i^1\}$. The notion of viewing bands as ordered pairs is conceptually similar to considering genes/synteny blocks as oriented, as is standard in the computational studies of genome rearrangements in species evolution (Bourque and Zhang, 2006). In this study, we consider bands as ordered pairs to well identify breakpoints: as mentioned previously, a breakage usually occurs within a band, say i, and the two ends of i, i^0 and i^1, are separated between the two new resulting fragments. Thus, a fusion of two fragment-ends forms an adjacency iff these ends are complementing. We identify a breakpoint, and a concatenation point in general, by the two corresponding fragment-ends that are fused together. More formally, the concatenation point in $[a, b]::[a', b']$ is identified by the (unordered) pair $\{b, a'\}$. For example, the breakpoint in $[1, 2]::[4, 3] \equiv [1^1, 2^0]::[4^0, 3^1]$ is identified by $\{2^0, 4^0\}$. Having defined breakpoint identities, we refer to a breakpoint as *unique* if no other breakpoint shares its identity, and otherwise we call it *repeated*. In particular, a breakpoint in a non-unique chromosome (i.e., a chromosome with another identical copy) is repeated. Last, we say that a chromosome X is *complex* if it contains at least one breakpoint, and *simple* otherwise. In other words, chromosome X is simple iff it consists of one fragment. Analogously, an addition is *complex* if the chromosome added is complex, and *simple* otherwise.

[1]Formally, since the broken ends of a chromosome are not considered breakpoints here, the term "fusion-point" may seem more appropriate. However, we kept the name "breakpoint" due to its prior use and for brevity.

Let $K_{\text{normal}} = \{[1,4] \times 2, [5,8] \times 2\}$, and $K_{\text{cancer}} = \{[1,4], [5,8], [1,3], [6,8], [1,4]::[5,8]\}$. An optimal sorting scenario contains 5 moves, one of which is always a complex addition. An example for an optimal sorting scenario:

$$K_{\text{cancer}} \xrightarrow{\text{addition}} \{[1,4],\ [5,8],\ [1,3], [\mathbf{3},\mathbf{4}]::[\mathbf{5},\mathbf{6}], [6,8],\ [1,4]::[5,8]\}$$
$$\xrightarrow{\text{2 fusions}} \{[1,4],\ [5,8],\ [1,4]::[5,8] \times 2\}$$
$$\xrightarrow{\text{c-deletion}} \{[1,4],\ [5,8],\ [1,4]::[5,8]\}$$
$$\xrightarrow{\text{breakage}} \{[1,4],\ [5,8],\ [1,4],\ [5,8]\} = K_{\text{normal}}$$

A sorting scenario that does not involve a complex addition contains at least 7 moves, and hence is non-optimal. An example of such sorting scenario:

$$K_{\text{cancer}} \xrightarrow{\text{breakage}} \{[1,4] \times 2,\ [5,8] \times 2,\ [1,3],\ [6,8]\}$$
$$\xrightarrow{\text{2 additions}} \{[1,4] \times 2,\ [5,8] \times 2,\ [1,3], [\mathbf{3},\mathbf{4}],\ [\mathbf{5},\mathbf{6}], [6,8]\}$$
$$\xrightarrow{\text{2 fusions}} \{[1,4] \times 3,\ [5,8] \times 3\}$$
$$\xrightarrow{\text{2 c-deletions}} \{[1,4] \times 2,\ [5,8] \times 2\} = K_{\text{normal}}$$

FIG. 3. An example K_{cancer} and K_{normal} for which any optimal sorting scenario contains a complex addition. Note that this scenario involves duplication of the breakpoint in [1,4]::[5,8], while repeated breakpoints are quite rare in the real data.

Observation 1. *Let S be an optimal sorting sequence. Suppose K_{cancer} contains a breakpoint, p, that is not involved in a c-deletion in S. Then there exists an optimal sorting sequence S′, in which the first operation is a breakage of p.*

Proof. Since K_{normal} does not contain any breakpoint, p must be eventually eliminated by S. A breakpoint can be eliminated either by a breakage or by a c-deletion. Since p is not involved in a c-deletion, p is necessarily eliminated by a breakage. Moreover, this breakage can be moved to the beginning of S since no other operation preceding it involves p. ■

Corollary 1. *Let S be an optimal sorting sequence. Suppose S contains an addition of chromosome $X = f_1::f_2:: ::f_k$, where $f_1, f_2, \ldots, f_k$ are fragments, and none of the $k-1$ breakpoints in X is involved in any subsequent c-deletion in S. Then the sequence S′, obtained from S by replacing the addition of X with the additions of $f_1, f_2, \ldots, f_k$ (a total of k additions), is an optimal sorting sequence.*

Proof. By Observation 1, the breakpoints in X can be immediately broken after its addition. Thus, replacing the addition of X, and the $k-1$ breakages following it, by k additions of $f_1, f_2, \ldots, f_k$, yields an optimal sorting sequence. ■

It appears that complex additions, as opposed to simple additions, make KS very difficult to analyze. Moreover, based on Corollary 1, complex additions can be truly beneficial only in complex scenarios in which c-deletions involve repeated breakpoints that were formerly created by complex additions (Fig. 3). Therefore, we make the following assumption:

Assumption 1. *Every addition is simple, i.e., every added chromosome consists of one fragment.*

Using the assumption above, the following observation holds:

Observation 2. *Let p be a unique breakpoint in K_{cancer}. Then there exists an optimal sorting sequence in which the first operation is a breakage of p.*

Proof. If p is not involved in a c-deletion, then by Observation 1, p can be broken immediately. Suppose there are k c-deletions involving p or other breakpoints identical to it. If p is on chromosome X that is c-deleted, then at the time of the c-deletion, another copy X' of X is present in the karyotype, with an identical breakpoint p' in it. Note that following Assumption 1, from the four inverse elementary operations, only fusion can create a new breakpoint. Thus, we can obtain an optimal sorting sequence, S', from S, by: *(i)* first breaking p, *(ii)* canceling any fusion that creates a breakpoint p' identical to p, *(iii)* replacing any c-deletion involving p, or one of its copies, with two c-deletions of the corresponding 4 unfused chromosomes, and *(iv)* not having to break the last instance of p (since it was already broken). In summary, we moved the breakage of p to the beginning of the sorting sequence and replaced k fusions and k c-deletions (i.e., 2k operations) with $2k$ c-deletions. ■

Observation 3. *In an optimal sequence, every fusion creates either an adjacency, or a repeated breakpoint.*

Proof. Let S be an optimal sorting sequence. Suppose S contains a fusion that creates a new unique breakpoint p. Then, following Observation 2, p can be immediately broken after it was formed, a contradiction to the optimality of S. ■

In this work, we choose to focus on karyotypes that do not contain repeated breakpoints. According to our analysis of the Mitelman database, 94% of the karyotypes satisfy this condition. Thus, we make the following additional assumption:

Assumption 2. *The cancer karyotype, K_{cancer}, does not contain any repeated breakpoint.*

Assumption 2 implies that we can *(i)* immediately break all the breakpoints in K_{cancer} (due to Observation 2), and *(ii)* consider fusions only if they create an adjacency (due to Observation 3). Hence, given a cancer karyotype, for each normal chromosome, its fragments can be separated from all the other fragments and used to solve a simpler variant of KS: In this variant, *(i)* $K_{\text{normal}} = \{[1, B] \times N\}$, *(ii)* there are no breakpoints in K_{cancer}, and *(iii)* neither fusions, nor additions, form breakpoints. Usually, $N = 2$, with $N = 1$ for the sex chromosomes. We refer to this reduced problem as *restricted KS* (abbreviated RKS). For the rest of the article, we shall limit our analysis to RKS only.

3. A LOWER BOUND FOR THE ELEMENTARY DISTANCE

In this section, we analyze RKS and define several combinatorial parameters that affect the elementary distance between K_{normal} and K_{cancer}. Based on these parameters, we prove a lower bound on the elementary distance. Though theoretically our lower bound is not tight, we shall demonstrate in Section 4 that in practice, for the vast majority (99.9%) of the real cancer karyotypes analyzed, the elementary distance achieves this bound.

3.1. Extending the karyotypes

For simplicity of later analysis, we extend both K_{normal} and K_{cancer} by adding to each karyotype $2N$ "tail" intervals:

$$\widehat{K}_{\text{normal}} = K_{\text{normal}} \cup \{[0, 1] \times N, [B, B + 1] \times N\}$$
$$\widehat{K}_{\text{cancer}} = K_{\text{cancer}} \cup \{[0, 1] \times N, [B, B + 1] \times N\}$$

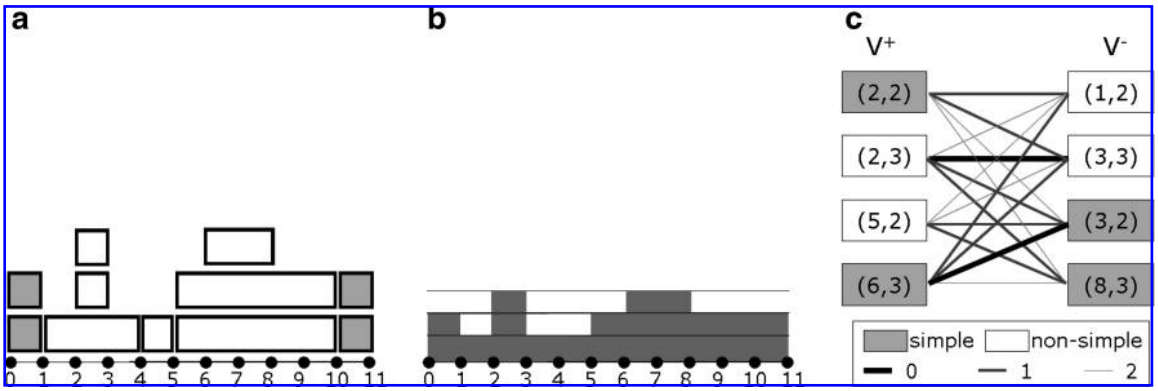


FIG. 4. An example of a cancer karyotype $\hat{K}_{\text{cancer}}$ and its combinatorial parameters. (**a**) The (extended) cancer karyotype is $\hat{K}_{\text{cancer}} = \{[0,1]\times 2, [1,4], [4,5], [5,10]\times 2, [10,11]\times 2, [2,3]\times 2, [6,8]\}$. Here $N=2$, $B=10$. The number of disjoint pairs of complementing fragment-ends, f, is 5. (**b**) The histogram $H \equiv H(\hat{K}_{\text{cancer}})$. H has walls at 1, 2, 3, 5, 6, and 8. There are four positive bricks: (2,2), (2,3), (5,2), and (6,3), and four negative bricks: (1,2), (3,3), (3,2), and (8,3). Hence $w=8$. Four of the eight bricks are simple: (2,2), (3,2), (6,3), and (8,3), thus $s=4$. (**c**) The weighted bipartite graph of BG. It is not hard to verify that $M = \{ ((2,3),(3,3)), ((6,3),(3,2)), ((2,2),(1,2)), ((5,2),(8,3)) \}$ is a minimum-weight perfect matching and hence $m=2$.

For an example, see Figure 4a. These new "tail" intervals do not take part in elementary operations: breakage and fusion are still limited to $\{2, 3, \ldots, B-1\}$, and intervals added/c-deleted are contained in $[1, B]$. Hence $d(K_{\text{normal}}, K_{\text{cancer}}) \equiv d(\widehat{K}_{\text{cancer}}, \widehat{K}_{\text{cancer}})$. Their only role is to simplify the definitions of parameters given below.

3.2. The histogram

We define the *histogram* of $\widehat{K}_{\text{cancer}}, H \equiv H(\widehat{K}_{\text{cancer}}) : \{[i-1,i] \mid i = 1, 2, \ldots, B+1\} \to \mathbb{N} \cup \{0\}$, as follows. Let $H([i-1,i])$ be the number of fragments in $\widehat{K}_{\text{cancer}}$ that contain the interval $[i-1, i]$ (Fig. 4b). From the definition of $\widehat{K}_{\text{cancer}}$, it follows that $H([0,1]) = H([B, B+1]) = N$. For simplicity, we refer to $H([i-1, i])$ as $H(i)$. The histogram H has a *wall* at $i \in \{1, \ldots, B\}$ if $H(i) \neq H(i+1)$. If $H(i+1) > H(i)$ (respectively, $< H(i)$) then the wall at i is called a *positive* wall (respectively, a *negative* wall). Intuitively, a wall is a vertical jump of H. We define w to be the total size of walls in H. More formally,

$$w = \sum_{i=1}^{B} |H(i+1) - H(i)|$$

Since $H(1) = H(B+1) = N$, the total size of positive walls is equal to the total size of negative walls, and hence w is even. Note that if $\widehat{K}_{\text{cancer}} = \widehat{K}_{\text{normal}}$ then $w=0$. The pair $(i,h) \equiv (i, [h-1, h])$, $h \in \mathbb{N}$, is a *brick* in the wall at i if $H(i)+1 \leq h \leq H(i+1)$ or $H(i+1)+1 \leq h \leq H(i)$. A brick (i, h) is *positive* (respectively, *negative*) if the wall at i is positive (respectively, negative). Note that the number of bricks in a wall is equal to its total size. Hence, w corresponds to the total number of bricks in H.

Observation 4. *For breakage and fusion, $\Delta w = 0$; For c-deletion and addition, $\Delta w = \{-2, 0, 2\}$.*

3.3. Counting complementing end pairs

Consider the case where $w=0$. Then there are no gains and no losses of bands, and the number of fragments in $\widehat{K}_{\text{cancer}}$ is greater or equal to the number of fragments in $\widehat{K}_{\text{normal}}$. Note that each of the four elementary operations can decrease the total number of fragments by at most one. Hence, when $w=0$, an optimal sorting sequence would be to fuse pairs of complementing fragment-ends, not including the tails. Let us define $f \equiv f(\widehat{K}_{\text{cancer}})$ as the maximum number of disjoint pairs of complementing fragment-ends. Note there could be many alternative choices of complementing pairs. Nevertheless, any maximal disjoint pairing is also maximum. It follows that if $w=0$, then $d(\widehat{K}_{\text{normal}}, \widehat{K}_{\text{cancer}}) = f - 2N$. Also, when $w \neq 0$, a c-deletion may need to be preceded by some fusions of complementing ends, to form two identical fragments. In general, the following holds:

Observation 5. *For breakage* $\Delta f = 1$*; For fusion,* $\Delta f = -1$*; For c-deletion,* $\Delta f \in \{0, -1, -2\}$*; For addition,* $\Delta f \in \{0, 1, 2\}$.

Lemma 1. *For breakage and addition,* $\Delta(w/2 + f) = 1$*; For fusion and c-deletion,* $\Delta(w/2 + f) = -1$.

Proof. For breakage/fusion, $\Delta w = 0$, and thus the lemma immediately follows from Observation 5. For addition: $(\Delta w = 0) \Rightarrow (\Delta f = 1)$; $(\Delta w = -2) \Rightarrow (\Delta f = 2)$; $(\Delta w = 2) \Rightarrow (\Delta f = 0)$. For c-deletion: $(\Delta w = 0) \Rightarrow (\Delta f = -1)$; $(\Delta w = -2) \Rightarrow (\Delta f = 0)$; $(\Delta w = 2) \Rightarrow (\Delta f = -2)$. ■

3.4. Simple bricks

A brick (i, h) is called *simple* if: *(i)* $(i, h-1)$ is not a brick, and *(ii)* $\widehat{K}_{\text{cancer}}$ does not contain a pair of complementing fragment-ends in i (Fig. 4b). Thus, in particular, a simple brick cannot be eliminated by a c-deletion. On the other hand, for a non-simple brick, (i, h), there are two fragments ending in the corresponding location (i.e., i). Nevertheless, it may still be impossible to eliminate (i, h) by a c-deletion if these two fragments are not identical. We define $s \equiv s(\widehat{K}_{\text{cancer}})$ as the number of simple bricks.

Observation 6. *For breakage,* $\Delta s \in \{0, -1\}$*; For fusion,* $\Delta s \in \{0, 1\}$*; For c-deletion,* $\Delta s = 0$*; For addition,* $|\Delta s| \leq 2$.

Observation 6 and Lemma 1 imply:

Lemma 2. *For every move,* $\Delta(w/2 + f + s) \geq -1$.

3.5. The weighted bipartite graph of bricks

The last parameter that we define is based upon matching pairs of bricks. Note that in the process of sorting $\widehat{K}_{\text{cancer}}$, the histogram is flattened, i.e., all bricks are eliminated, which can be done only by using c-deletion/addition operations. If a c-deletion/addition eliminates a pair of bricks, then one of these bricks is positive and the other is negative. Thus, roughly speaking, every sorting sequence defines a matching between pairs of positive and negative bricks that are eliminated together.

Given two bricks, $v = (i, h)$ and $v' = (i', h')$, we write $v < v'$ (resp. $v = v'$) if $i < i'$ (resp. $i = i'$). Let V^+ and V^- be the sets of positive and negative bricks, respectively. We say that v and v' have the same *sign*, if either $v, v' \in V^+$, or $v, v' \in V^-$. Two bricks have the same *status* if they are either both simple, or both non-simple. Let $BG = (V^+, V^-, \delta)$ be the weighted complete bipartite graph, where $\delta : V^+ \times V^- \to \{0, 1, 2\}$ is an edge-weight function defined as follows. Let $v^+ \in V^+$ and $v^- \in V^-$. Then:

$$\delta(v^+, v^-) = \begin{cases} 0 & v^+ \text{ and } v^- \text{ are both simple } \textbf{and } v^- < v^+ \\ 0 & v^+ \text{ and } v^- \text{ are both non-simple } \textbf{and } v^+ < v^- \\ 1 & v^+ \text{ and } v^- \text{ have opposite status} \\ 2 & \text{otherwise} \end{cases}$$

For an illustration of BG, see Figure 4c. Roughly speaking, $\delta(v^+, v^-)$ corresponds to the additional cost of eliminating v^+ and v^- together, either by an addition, when $v^- < v^+$, or by c-deletion, when $v^+ < v^-$. A *matching* is a set of vertex-disjoint edges from $V^+ \times V^-$. A matching is *perfect* if it covers all the vertices in BG (recall that $|V^+| = |V^-|$). Thus, a perfect matching is in particular a maximum matching. Given a matching M, we define $\delta(M)$ as the total weight of its edges. Let $m \equiv m(\hat{K}_{\text{cancer}})$ denote the minimum weight of a perfect matching in BG. The problem of finding a minimum-weight perfect matching in a bipartite graph, also known as the *assignment problem*, can be solved in $O(n^3)$ time (Kuhn, 1955; Munkres, 1957). In the Appendix, we describe a simple $O(n \log n)$ algorithm for computing m, which relies heavily on the specific weighting scheme, δ.

Below, we prove a lower bound for the elementary distance using the four parameters we have just defined: w, f, s, and m. First, we prove two technical lemmas.

Lemma 3. *Let M and M' be two perfect matchings that differ by exactly two edges (i.e., four vertices). Then $|\delta(M) - \delta(M')| \leq 2$.*

Proof. Let $M \setminus M' = \{e_1, e_2\}$ and $M' \setminus M = \{e_3, e_4\}$. Assume w.l.o.g. that $\Delta = \delta(M') - \delta(M) \geq 0$. Then $\Delta = \delta(e_3) + \delta(e_4) - \delta(e_1) - \delta(e_2) \leq 4$, since for every edge, e, $\delta(e) \in \{0, 1, 2\}$. If $\delta(e_1) + \delta(e_2) \geq 2$ then clearly $\Delta \leq 2$. Suppose $\delta(e_1) + \delta(e_2) < 2$. Now, let $e_1 = (v_1, u_1)$ and $e_2 = (v_2, u_2)$. W.l.o.g. we assume that $e_3 = (v_1, v_2)$ and $e_4 = (u_1, u_2)$.

- Case 1: $\delta(e_1) = \delta(e_2) = 0$. In this case, e_1 and e_2 connect vertices with the same status. If v_1 has a different status than v_2, then $\delta(e_3) = \delta(e_4) = 1$. Otherwise, v_1, u_1, v_2, and u_2 have the same status. In this case it is not hard to verify by considering the possible orderings of $\{v_1, u_1, v_2, u_2\}$ that $\delta(e_3) + \delta(e_4) \in \{0, 2\}$. Thus, in either case $\Delta \leq 2$.
- Case 2: $\delta(e_1) + \delta(e_2) = 1$. In this case, exactly three vertices in $\{v_1, u_1, v_2, u_2\}$ have the same status, while the remaining vertex has the opposite status. Thus, it follows that either $\delta(e_3) = 1$ or $\delta(e_4) = 1$ and thus $\Delta \leq 2$. ■

Let K' be obtained from K by an elementary operation (a move). For a function F defined on karyotypes, define $\Delta(F) = F(K') - F(K)$.

Proposition 1. *For every move, $\Delta(w/2 + f + s + m) \geq -1$.*

Proof. For a given move, let $\Delta = \Delta(w/2 + f + s + m)$. Let G_1 and G_2 be the graphs before and after we make the move, respectively, and let M_1 and M_2 be minimum-weight perfect matchings in G_1 and G_2, respectively, where $|M_2 \setminus M_1|$ is minimal. Thus $\Delta m = m_2 - m_1$, where $m_1 = \delta(M_1)$ and $m_2 = \delta(M_2)$ We shall prove $\Delta \geq -1$ by considering each move type.

- *Breakage.* We shall prove that $|\Delta| \leq 1$. Now, $\Delta(w/2 + f) = 1$ (Lemma 1), $\Delta(s) \in \{0, -1\}$ (Observation 6). If $\Delta m = 0$ then $\Delta \in \{1, 0\}$. Suppose $\Delta m \neq 0$. Then a simple brick v became non-simple due to the move and $\Delta s = -1$. It follows that every edge, e, adjacent to v satisfies $\Delta(\delta(e)) \in \{-1, 1\}$. Hence, for every perfect matching M, $\Delta(\delta(M)) \in \{-1, 1\}$. Then, in G_1: $m_1 \leq \delta(M_2) \leq m_2 + 1$, and in G_2: $m_2 \leq \delta(M_1) \leq m_1 + 1$. Hence $|\Delta| = |\Delta m| \leq 1$.
- *Fusion.* Since fusion is the inverse operation to breakage, it follows that $|\Delta| \leq 1$ for fusion as well.
- *C-deletion.* By Lemma 1 $\Delta(w/2 + f) = -1$ and by Observation 6, $\Delta(s) = 0$. We shall prove that $\Delta m \geq 0$ by analyzing the possible values of Δw.
- $\Delta w = -2$. Then two bricks, $v^+ \in V^+$ and $v^- \in V^-$, were eliminated, where $v^+ < v^-$, and both v^+ and v^- are non-simple. Let $e = (v^+, v^-)$. Clearly, $\delta(e) = 0$. Thus before we apply the move: $m_2 = \delta(M_2) = \delta(M2 \cup \{e\}) \geq \delta(M_1) = m_1$. Hence $\Delta m \geq 0$.
  - $\Delta w = 0$. In this case, a non-simple brick, v, was replaced with another non-simple brick, v' with the same sign. If $v, v' \in V^+$, then $v < v'$, otherwise, $v > v'$. Thus, for every vertex u with the same sign to v, $\delta((v, u)) \leq \delta((v', u))$. For every vertex u with the opposite sign, $\delta((v,u)) = \delta((v',u))$. Hence, $\Delta m \geq 0$.
  - $\Delta w = 2$. In this case, a pair of new non-simple bricks, $v^- \in V^-$ and $v^+ \in V^+$ was added, where $v^- < v^+$. Let $e = (v^+, v^-)$. Then clearly $\delta(e) = 2$. Recall that $|M_2 \setminus M_1|$ is minimal. We now prove that $M_2 = M_1 \cup \{e\}$ and hence $m_2 = m_1 + 2$. Suppose $e \notin M_2$. Let $u^+ \in V^+$ and $u^- \in V^-$ be the nodes matched to v^- and v^+, respectively, in M_2. Let M_1' be a minimal perfect matching in G_1 that contains $e' = (u^-, u^+)$. Then $\delta(M_1') \geq m_1$ and thus it suffices to prove that $\delta(M_2) \geq \delta(M_1')$. We will do so by proving that $\delta(v^-, u^+) + \delta(v^+, u^-) \geq \delta(e')$. If $\delta(e') = 0$ then this is certainly true. Suppose $\delta(e') > 0$.
    - —$\delta(e') = 1$. Then exactly one of u^+ and u^- is simple, hence either $\delta(v^-, u^+) = 1$ or $\delta(v^+, u^-) = 1$.
    - —$\delta(e') = 2$. Then u^+ and u^- have the same status. If they are both simple then $\delta(v^-, u^+) + \delta(v^+, u^-) = 1 + 1 = 2 = \delta(e')$. Otherwise, a simple case analysis reveals that at least one of the edges (v^+, u^-) and (u^+, v^-) has a weight 2, and thus $\delta(v^-, u^+) + \delta(v^+, u^-) \geq 2$.
- *Addition.* Then $\Delta(w/2 + f) = 1$ (Lemma 1), $\Delta s \geq -2$ (Observation 6).
  - $\Delta w = -2$. In this case, two bricks, $v^- \in V^-$ and $v^+ \in V^+$, were eliminated, where $v^- < v^+$. Let $e = (v^-, v^+)$. Then $\delta(e) = 2 + \Delta s$. Moreover, $m_2 = \delta(M_2 \cup \{e\}) - \delta(e) \geq m_1 - \delta(e)$. Thus $\Delta m + \Delta s \geq -\delta(e) + \Delta s = -2$. Hence $\Delta \geq -1$.
  - $\Delta w = 0$. In this case, one brick, v, was replaced with a new brick with the same sign, v'. Thus $\Delta s \geq -1$, and $\Delta m \geq -2$, since only the edges adjacent to v, which are now adjacent to v', are affected. If $\Delta s \geq 0$ then clearly $\Delta \geq -1$. Suppose $\Delta s = -1$. The a simple brick was replaced with a non-simple brick. Let u be a vertex with the opposite sign to v. Then $\delta((u,v)) - \delta((u,v')) \geq -1$, and thus $\Delta m \geq -1$. Therefore, $\Delta \geq -1$.
  - $\Delta w = 2$. Then two new bricks, $v^+ \in V^+$ and $v^- \in V^-$, were added, where $v^+ < v^-$. Thus $\Delta s \geq 0$. Also $\Delta(f) = 0$. It suffices to prove that $\Delta m \geq -2$ and hence $\Delta \geq -1$. Let $e = (v^+, v^-)$. If $e \in M_2$ then clearly $m_2 \geq m_1$,

and thus $\Delta m \geq 0$. Suppose $e \notin M_2$. Then there exist $e_1, e_2 \in M_2$ where $e_1 = (v^+, u^-)$, $e_2 = (v^-, u^+)$. Let $M_1' = M_2 \setminus \{e_1, e_2\} \cup \{e'\}$, where $e' = (u^+, u^-)$. Then M_1' is a perfect matching in G_1 and thus $\delta(M_1') \geq m_1$. Now, $M_2' = M_1' \cup \{e\}$ is a perfect matching in G_2, which differs from M_2 by exactly two edges. By Lemma 3, $\delta(M_2) \geq \delta(M_2') - 2$. Since $\delta(M_2') = \delta(M_1') + \delta(e') \geq m_1$, it follows that $m_2 \geq m_1 - 2$ and thus $\Delta m \geq -2$. ■

Corollary 2. $d \geq w/2 + f - 2N + s + m \geq 0$.

Proof. Since N is constant, Proposition 1 implies $\Delta(w/2 + f - 2N + s + m) \geq -1$. For $\hat{K}_{\text{cancer}} = \hat{K}_{\text{normal}}$, $w/2 + f - 2N + s + m = 0 + 2N - 2N + 0 + 0 = 0$. Thus the left inequality holds, and it suffices to prove that $t = w/2 + f - 2N + s + m \geq 0$. If $f \geq 2N$ then clearly $t \geq 0$. Suppose $f < 2N$. We shall prove that $f + s + m \geq 2N$. There are at least $2N - f$ intervals of the form [0, 1] or $[B, B+1]$, with no complementing fragment-ends at $1, B$. Each of these unmatched tails corresponds to a brick at 1 or B. Let us look at an optimal matching and focus on the edges involving these bricks. There are at least $\lceil (2N - f)/2 \rceil$ such edges. It is easy to verify that each of these edges contributes 2 to $s + m$, hence $s + \mathrm{m} \geq 2N - f$. ■

4. THE 3-APPROXIMATION ALGORITHM

Algorithm 1 is a polynomial procedure for the RKS problem. We shall prove that it is a 3-approximation, and then describe a heuristic that aims to improve it.

Lemma 4. *Algorithm 1 transforms $\hat{K}_{cancer}$ into $\hat{K}_{normal}$ using at most $3w/2 + f - 2N + s + m$ inverse elementary operations.*

Proof. Let $\Delta \equiv \Delta(w/2 + f + s + m)$. First, we prove that $\Delta = -1$ for each move except Step 13, and for Step 13 moves, $\Delta = 1$.

- Step 3: $\Delta(w/2 + f) = 1, \Delta(s + m) = -2$. Note that if there exists a negative (resp. positive) brick at 1 (resp. B), then this brick is necessarily eliminated in this step.
- Steps 7,9: $\Delta(w/2 + f) = 1$ (by Lemma 1). After Step 3, any brick at 1 (resp. B) is necessarily positive (resp. negative) and thus not simple. Thus $\Delta s = -1$. Now $\Delta m \geq -1$ (by Proposition 1). By using the maximal matching induced by M, in which v is replaced by 1 (if $v \in V^+$) or by B (if $v \in V^-$), we get $\Delta m = -1$.
- Step 13: By now, $V^+ \cup V^-$ contains only non-simple bricks, i.e., $s = 0$ and thus $\Delta s = 0$. Moreover, $m = 0$, since the matching induced by M is optimal (see previous step) and every pair (v^+, v^-) in it, where $v^+ \in V^+$ and $v^- \in V^-$, satisfies $v^+ < v^-$. Therefore, $\Delta m = 0$. $\Delta(w/2 + f) = 1$ (by Lemma 1).
- Step 18: There are no bricks at p, thus $\Delta s = \Delta m = 0$, and $\Delta = \Delta(w/2 + f) = -1$ (by Lemma 1).
- Step 20: By now, all bricks are non-simple and the negative bricks are at B. Thus $s = m = 0$ and $\Delta s = \Delta m = 0$. $\Delta(w/2 + f) = -1$ (by Lemma 1).

Algorithm 1 Elementary Sorting (RKS)

```
1: M ← a minimum-weight perfect matching in BG
2: for all (v⁻, v⁺) ∈ M where v⁻ < v⁺ do
3:   Add the interval [v⁻, v⁺].
4: end for /* Now v⁺ < v⁻ for every (v⁺, v⁻) ∈ M, where v⁺ ∈ V⁺, v⁻ ∈ V⁻ */
5: for all v ∈ V⁺ ∪ V⁻ such that v is simple, and v ≠ 1, B do
6:   if v ∈ V⁺ then
7:     Add the interval [1,v]
8:   else
9:     Add the interval [v,B]
10:  end if
11: end for /* Now v⁺ < v⁻ for every (v⁺, v⁻) ∈ M, where v⁺ ∈ V⁺, v⁻ ∈ V⁻ and
    all the bricks are non-simple. In addition, 1 ∉ V⁻ and B ∉ V⁺ */
12: for all v⁻ ∈ V⁻ such that v⁻ < B do
13:   Add the interval [v⁻, B]
```

14: **end for** /* *Now all the bricks are non-simple,* ***and*** $v^- = B, \forall v^- \in V^{-*}$ /
15: **while** $V^+ \neq \emptyset$ **do**
16: $v^+ \leftarrow \max V^+$
17: **for all** $p > v^+$, $p < B$ **do**
18: Fuse any pair of intervals complementing at p.
19: **end for**
20: C-delete an interval $[v^+, B]$
21: **end while**

Let $t = w/2 + f - 2N + s + m$. There are at most $w/2$ additions at Step 13, each of which satisfies $\Delta = 1$. For all the other operations we have shown that $\Delta = -1$. Thus the overall number of operations is less or equal to $w/2 + t + w/2 = 3w/2 + f - 2N + s + m$. ■

Theorem 1. *Algorithm 1 is a polynomial-time 3-approximation algorithm for RKS.*

Proof. By Lemma 4, the algorithm requires $\leq 3t$ moves. By Corollary 2, that number is at most $3d$. ■

Note that the same result applies to multi-chromosomal karyotypes, by summing the bounds for the RKS problem on each chromosome. Note also that the results above imply also that $d \in [w/2 + f - 2N + s + m, 3w/2 + f - 2N + s + m]$

We now present Procedure 2, a heuristic that attempts to improve the performance of Algorithm 1, by suggesting an alternative to steps 12–21. The procedure assumes that *(i)* all bricks are non-simple, and *(ii)* $v^+ < v^-$, for every $(v^+, v^-) \in M, v^- \in V^-, v^+ \in V^+$. In this case, $m = 0$, and the lower bound is reached only if no additions are made. Thus, Procedure 2 attempts to minimize the number of extra addition operations performed. For an interval I, let $L(I)$ and $R(I)$ be the left and right endpoints of I respectively.

5. EXPERIMENTAL RESULTS

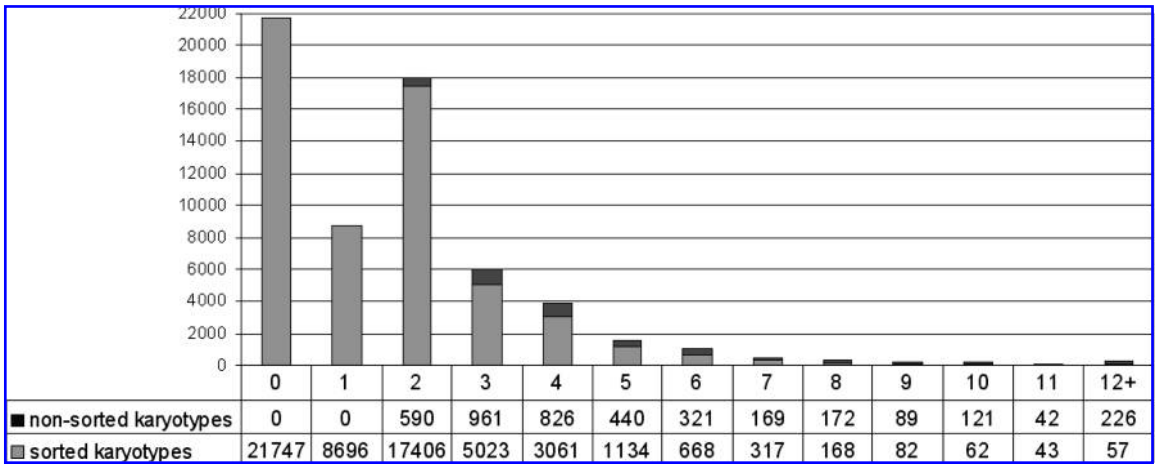
In this section, we present the results of sorting real cancer karyotypes, using Algorithm 1, combined with the improvement heuristic in Procedure 2.

Procedure 2 Heuristic for eliminating non-simple bricks

1: **while** $V^+ \neq \emptyset$ **do**
2: $v^+ \leftarrow \max V^+$
3: **for all** $p > v^+, p < B, p \notin V^-$ **do**
4: Fuse any pair of intervals complementing at p.
5: **end for**
6: **if** $\exists I_1, I_2$, where $I_1 = I_2$ **and** $L(I_1) = v^+$, **and** $R(I_1) < R(I_2) \in V^-$ **then**
7: Let I_1, I_2 be a pair of intervals with minimal length satisfying the above.
8: C-delete I_1
9: **else if** $\exists I_1, I_2$, where $L(I_1) = L(I_2) = v^+$ **and** $R(I_1) < R(I_2) \in V^-$ **then**
10: Let I_1, I_2 be a pair of intervals with minimal length satisfying the above.
11: Add the interval $[R(I_1), R(I_2)]$
12: **else**
13: Let $u^- = \min\{v^- \in V^- | v^- > v^+\}$
14: Add the interval $[u^-, B]$
15: **end if**
16: **end while**

5.1. Data preprocessing

For our analysis, we used the Mitelman database (version of November 4, 2008), which contained 57,776 cancer karyotypes, collected from 9,311 published studies. The karyotypes in the Mitelman database (henceforth, MD) are represented in the ISCN format and can be automatically parsed and analyzed using the software package CyDAS (Hiller et al., 2005). We refer to a karyotype as *valid* if it was parsed by



	0	1	2	3	4	5	6	7	8	9	10	11	12+
non-sorted karyotypes	0	0	590	961	826	440	321	169	172	89	121	42	226
sorted karyotypes	21747	8696	17406	5023	3061	1134	668	317	168	82	62	43	57

FIG. 5. The distribution of number of breakpoints (i.e., fusions of non-adjacent bands) per karyotype. ''Sorted karyotypes'' correspond to karyotypes with *no* repeated breakpoints. ''Non-sorted karyotypes'' correspond to karyotypes with repeated breakpoints. About 35% of all the karyotypes do not contain any breakpoint.

CyDAS without any error. According to our processing, 50,769 (88%) of the records gave valid karyotypes. Since some of the records contain multiple distinct karyotypes found in the same tissue, the total number of simple valid karyotypes that we deduced from MD was 62,421.

A karyotype may contain uncertainties, or missing data, both represented by a ''?'' symbol. We ignored uncertainties and deleted any chromosomal fragments that were not well defined.

5.2. Sorting the karyotypes

Out of the 62,421 karyotypes analyzed, only 3,957 karyotypes (6%) contained repeated breakpoints. Our analysis focused on the remaining 58,464 karyotypes. We note that 21,747 (35%) of these karyotypes do not contain any breakpoint at all. (In these karyotypes, there are no fusions of bands that are not adjacent in normal chromosomes, but some chromosome tails, as well as full chromosomes, may be missing or duplicated.) Following our assumptions (see Section 1.2), we broke all the breakpoints in each karyotype. To avoid over estimation of whole chromosome gains due to events of global changes in the genome ploidy, we used the ploidy of each karyotype as the normal copy-number (N) of each chromosome. (The ploidy was computed by the CyDAS parser, based on the the ISCN description of karyotype.) We first applied Algorithm 1 (without the heuristic), to the fragments of each of the chromosomes in these karyotypes. In 54,903 (94%) of the analyzed karyotypes, this algorithm achieved the lower-bound, and thus produced optimal sequences. We then applied Algorithm 1, combined with Procedure 2, and the number of karyotypes that achieved the lower bound increased to 58,434 (99.9%) of the analyzed karyotypes. Each of the remaining 30 karyotypes contained one or two chromosomes for which the computed sequence was larger by 2 than the lower-bound. Manual inspection revealed that for each of these cases the elementary distance was indeed 2 above the lower bound. Hence the computed sequences were found to be optimal in 100% of the analyzed cases.

5.3. Operations statistics

We now present statistics on the elementary operations reconstructed by our algorithm. The 58,464 analyzed karyotypes, contained 86,666 (unique) breakpoints in total. Hence the average number of fusions

TABLE 1. AVERAGE NUMBER OF ELEMENTARY OPERATIONS PER (SORTED) CANCER KARYOTYPE

Breakage	*Fusion*	*Deletion*	*Duplication*	*All*
2.4	1.5	2.6	1.1	7.6

(eq. breakpoints) per karyotype is approximately 1.5. The distribution of the number of breakpoints per karyotype, for all valid karyotypes, including the non-sorted karyotypes (i.e karyotypes with repeated breakpoints, which are not analyzed by our algorithm), is presented in Figure 5. The most frequent number of breakpoints after zero is two, which is due to the prevalence of reciprocal translocations in the analyzed cancer karyotypes. (Indeed, a direct analysis of cancer karyotypes with exactly two breakpoints shows that 75% have a single translocation.) Table 1 summarizes the average number of operations per sorted karyotype.

6. DISCUSSION

In this article, we proposed a new mathematical model for analyzing the evolution of cancer karyotypes, using four simple operations. Our model was developed following our empirical observation that chromosome gain and loss are dominant events in cancer (Ozery-Flato and Shamir, 2007). That observation relied on a purely heuristic algorithm that reconstructed for each cancer karyotype a sequence of events leading to the normal karyotype, using a wide catalog of complex rearrangement events, such as inversions, tandem-duplications, iso-chromosome creation, etc. Here we attempted to reconstruct rearrangement events in cancer karyotypes in a rigorous, yet simplified, manner.

The fact that we model and analyze bands and karyotypes may seem out of fashion in an era of CGH micro arrays and next generation sequencing. While modern techniques today allow *in principle* detection of chromosomal aberrations in cancer at an extremely high resolution, the clinical reality is that karyotyping is still commonly used for studying cancer genomes, and to date it is the only abundant data resource for cancer genomes structure. Moreover, our framework is not limited to cytogenetic banding resolution, as the ''bands'' in our model may represent any DNA blocks.

Readers familiar with the wealth of computational works on evolutionary genome rearrangements (Bourque and Zhang, 2006) may wonder why we have not used traditional operations, such as inversions and translocations, as has been previously done (Raphael et al., 2003). The reason is that while inversions and translocations are believed to dominate the evolution of species, they form less than 25% of the rearrangement events in cancer karyotypes Ozery-Flato and Shamir (2007), and 15% in karyotypes of malignant solid tumors. The extant models for genome rearrangements do not cope with duplications and losses, which are frequently observed in cancer karyotypes, and thus are not suitable for cancer genomes evolution. Extending these models to allow duplications results, even for the simplest models, in computationally hard problems (Radcliffe et al., 2005, Theorem 10). On the other hand, the elementary operations in our model can easily explain the variety of chromosomal aberrations viewed in cancer (including inversions and translocations). Moreover, each elementary operation we consider is strongly supported by a known biological mechanism (Albertson et al., 2003): breakage corresponds to a double-strand-break (DSB); fusion can be viewed as a non-homologous end-joining DSB-repair; whole chromosome duplications and deletions are caused by uneven segregation of chromosomes.

Based on our new model for chromosomal aberrations, we defined a new genome sorting problem. To further simplify this problem, we made two assumptions that essentially prohibit the occurrence of repeated breakpoints in cancer karyotypes, and in their intermediates. All the cancer karyotypes we analyzed did not contain repeated breakpoints. Although we do not have direct evidence about their intermediate karyotypes, our assumption is supported by the fact that the vast majority (94%) of reported cancer karyotypes do not contain repeated breakpoints. We presented a lower bound for this simplified problem, and developed a polynomial 3-approximation algorithm. The application of this algorithm to 58,464 real cancer karyotypes yielded solutions that achieve the lower bound (and hence an optimal solution) in almost all cases (99.9%). This is probably due to the relative simplicity of reported karyotypes, especially after removing ones with repeated breakpoints (Fig. 5).

In the future, we would like to extend this work by weakening our assumptions in a way that will allow the analysis of the remaining non-analyzed karyotypes. Those karyotypes, due to their complexity, are likely to correspond to more advanced stages of cancer. Our hope is that this study will lead to further algorithmic research on chromosomal aberrations, and thus help in gaining more insight on the ways in which cancer evolves.

7. APPENDIX: FINDING A MINIMUM-WEIGHT PERFECT MATCHING

In this section, we present an $O(n \log n)$ algorithm for finding a minimum-weight perfect matching. For status T (i.e $T =$ "simple" or $T =$ "non-simple") and a set of bricks V, let $V_T \subseteq V$ denote the set of bricks in V that are of status T.

Observation 7. *Let $v_1^+, v_2^+ \in V_T^+$ and $v_1^-, v_2^- \in V_T^-$. Suppose $v_1^+ < v_2^+$ and $v_1^- < v_2^-$.*

- *If $T =$ "simple" then $\delta(v_1^-, v_2^+)) \leq \delta((v_1^-, v_1^+)) \leq \delta((v_2^-, v_1^+))$.*
- *If $T =$ "non-simple" then $\delta(v_1^+, v_2^-)) \leq \delta((v_1^+, v_1^-)) \leq \delta((v_2^+, v_1^-))$.*

Let $v_1^+, v_2^+ \in V^+$, and $v_1^-, v_2^- \in V^-$. Let $e_1 = (v_1^+, v_1^-)$, and $e_2 = (v_2^+, v_2^-)$. We say that $e_1 \leq e_2$ if $v_1^+ \leq v_2^+$ and $v_1^- \leq v_2^-$.

Lemma 5. *Suppose $e^* = \min\{e \in V_t^+ \times V_T^- \mid \delta(e) = 0\}$. Then there is a minimum-weight perfect matching that contains e^*.*

Proof. Let M' be a perfect matching that does not contain e^*, with a minimum weight. Let M be a perfect matching most similar to M' that does contain e^*. In other words M differs from M' by exactly two edges, one of which is e^*. Let $e_2 \in M \setminus M', e_2 \neq e^*$. Suppose $e^* = (v_1^+, v_1^-)$ and $e_2 = (v_2^+, v_2^-)$, where $v_1^+, v_2^+ \in V^+$ and $v_1^-, v_2^- \in V^-$. Then $M' \setminus M = \{e_3, e_4\}$, where $e_3 = (v_1^+, v_2^-)$ and $e_4 = (v_2^+, v_1^-)$. We shall prove that $\Delta m = \delta(M) - \delta(M') = \delta(e^*) + \delta(e_2) - (\delta(e_3) + \delta(e_4)) \leq 0$.

If $\delta(e_2) = 0$ then clearly $\Delta m \leq 0$. Suppose $\delta(e_2) > 0$. Since $\delta(e^*) = 0, v_1^+$ and v_1^- are of the same status, say T. Let $\bar{T}$ be the inverse status to T.

Case 1: v_2^+ and v_2^- have the same status. Then $\delta(e_2) = 2$. If the status of v_2^+ and v_2^- is $\bar{T}$ then $\delta(e_3) = \delta(e_4) = 1$ and thus $\Delta m = 0$. Suppose the status of v_2^+ and v_2^- is T. It suffices to prove that either $\delta(e_3) = 2$ or $\delta(e_4) = 2$. Suppose $\delta(e_3) = 0$. Recall that e^* ia a minimal edge in $V_T^+ \times V_T^-$ with a zero weight.

- $T =$ "simple". Then $(\delta(e_2) = 2) \Rightarrow (v_2^+ < v_2^-)$, and $\delta(e_3) = 0) \Rightarrow (v_1^+ > v_2^-)$ and thus $v_2^+ < v_1^-$ and $e_4 = (v_2^+, v_1^+) < (v_1^+, v_1^-) = e^*$. Since $e^*, e_4 \in V_T^+ \times V_T^-$ and e^* is the minimal edge in $V_T^+ \times V_T^-$ satisfying $\delta(e_1) = 0$, it follows that $\delta(e_4) = 2$.
- $T =$ "non-simple". In this case similar arguments to the case where $T =$ "simple" are used, by simply reversing the direction of each inequality.

Case 2: v_2^+ and v_2^- have a different status. In this case $\delta(e^*) + \delta(e_2) = 0 + 1 = 1$, and either $\delta(e_3) = 1$ or $\delta(e_4) = 1$. Thus $\Delta m \leq 0$. ■

Observation 7 and Lemma 5 immediately imply Algorithm 3, which finds a minimal-weight perfect matching in BG. It is not hard to verify that this algorithm can be implemented in $O(n \log n)$.

Algorithm 3 Finding a minimum-weight perfect matching in the weighted bipartite graph of bricks

```
1: M ← ∅
2: for all T = "simple","non-simple" do
3:    if T = "simple" then
4:       L1 ← increasingly ordered V_T^-
5:       L2 ← increasingly ordered V_T^+
6:    else
7:       L1 ← increasingly ordered V_T^+
8:       L2 ← increasingly ordered V_T^-
9:    end if
10:   flag ← true
11:   while flag = true and L1 ≠ ∅ do
12:      v1 ← the first brick in L1
13:      L1 ← L1 \ {v1}
14:      while v1 is unmatched and L2 ≠ ∅ do
```

```
15:          v_2 ← the first brick in L_2
16:          L_2 ← L_2 \ {v_2}
17:          if v_1 < v_2 then
18:            M ← M ∪ {(v_1, v_2)}
19:          end if
20:        end while
21:        if v_1 is unmatched then
22:          flag ← false
23:        end if
24:      end while
25:    end for
26:    while ∃ unmatched v^+ ∈ V^+, v^- ∈ V^- with different status do
27:      M ← M ∪ {(v^+, v^-)}
28:    end while
29:    while ∃ unmatched v^+ ∈ V^+, v^- ∈ V^- do
30:      M ← M ∪ {(v^+, v^-)}
31:    end while
32:    return M
```

ACKNOWLEDGMENTS

This study was supported in part by the Israeli Science Foundation (grants 385/06 and 802/08).

DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

Albertson, D., Collins, C., McCormick, F., et al. 2003. Chromosome aberrations in solid tumors. *Nat. Genet.* 34, 369–376.

Bourque, G., and Zhang, L. 2006. Models and methods in comparative genomics. *Adv. Compu.* 68, 60–105.

Ferguson, D., and Frederick, W. 2001. DNA double-strand break repair and chromosomal translocation: lessons from animal models. *Oncogene* 20, 5572–5579.

Greenman, C., Stephens, P., Smith, R., et al. 2007. Patterns of somatic mutation in human cancer genomes. *Nature* 446, 153.

Hiller, B., Bradtke, J., Balz, H., et al. 2005. CyDAS: a cytogenetic data analysis system. *BioInformatics* 21, 1282–1283. Available at: www.cydas.org.

Höglund, M., Frigyesi, A., Säll, T., et al. 2005. Statistical behavior of complex cancer karyotypes. *Genes Chromosomes Cancer* 42, 327–341.

Kuhn, H. 1955. The hungarian method for the assignment problem. *Naval Res. Logist. Q.* 2, 83–97.

Mitelman, F., ed. 1995. *ISCN (1995): An International System for Human Cytogenetic Nomenclature*. S. Karger, Basel.

Mitelman, F., and Johansson, B., eds. 2008. Mitelman database of chromosome aberrations in cancer. Available at: http://cgap.nci.nih.gov/Chromosomes/Mitelman.

Munkres, J. 1957. Algorithms for the assignment and transportation problems. *J. Soc. of Indust. Appl. Math.* 5, 32–38.

NCI. 2001. NCI and NCBI's SKY/M-FISH and CGH database. Avialable at: www.ncbi.nlm.nih.gov/sky/skyweb.cgi/.

Ozery-Flato, M., and Shamir, R. 2007. On the frequency of genome rearrangement events in cancer karyotypes. Tech Report, Tel Aviv University.

Radcliffe, A.J., Scott, A.D., and Wilmer, E.L. 2005. Reversals and transpositions over finite alphabets. *SIAM J. Discret. Math.* 19, 224–244.

Raphael, B., Volik, S., Collins, C., et al. 2003. Reconstructing tumor genome architectures. *Bioinformatics* 27, 162–171.

Snijders, A.M., Nowak, N., Segraves, R., et al. 2001. Assembly of microarrays for genome-wide measurement of DNA copy number. *Nat. Genet.* 29, 263–264.

Address correspondence to:
Michal Ozery-Flato
School of Computer Science
Tel-Aviv University
Tel-Aviv 69978, Israel

E-mail: ozery@post.tau.ac.il

Chapter 6

On the Frequency of Genome Rearrangement Events in Cancer Karyotypes

On the frequency of genome rearrangement events in cancer karyotypes

Michal Ozery-Flato and Ron Shamir

School of Computer Science, Tel-Aviv University, Tel Aviv 69978, Israel
{ozery,rshamir}@post.tau.ac.il

Abstract. Chromosomal instability is a hallmark of cancer. The results of this instability can be observed in the karyotypes of many cancerous genomes, which often contain a variety of aberrations. In this study we introduce a new approach for analyzing rearrangement events in carcinogenesis. This approach builds on a new effective heuristic for computing a short sequence of rearrangement events that may have led to a given karyotype. We applied this heuristic on over 40,000 karyotypes reported in the scientific literature. Our analysis implies that these karyotypes have evolved predominantly via four principal event types: chromosomes gains and losses, reciprocal translocations, and terminal deletions. We used the frequencies of the reconstructed rearrangement events to measure similarity between karyotypes. Using clustering techniques, we demonstrate that in many cases, rearrangement event frequencies are a meaningful criterion for distinguishing between karyotypes of distinct tumor classes. Further investigations of this kind can provide insight on the scenarios by which particular cancer types have evolved.

1 Introduction

It is well known that many cancerous genomes exhibit abnormal karyotypes. The abnormalities found in these karyotypes include *numerical aberrations*, i.e. changes in chromosome copy number, and *structural aberrations*, i.e. rearrangements within the genome (see Fig. 1). Some of the malignancies, mostly hematological ones, are associated with specific patterns of aberrations. A classical example of such association is between the "Philadelphia chromosome" abberation (a specific translocation between chromosomes 22 and 9) and chronic myelogenous leukemia [17, 19]. This translocation leads to the formation of the oncogene BCR-ABL [5].

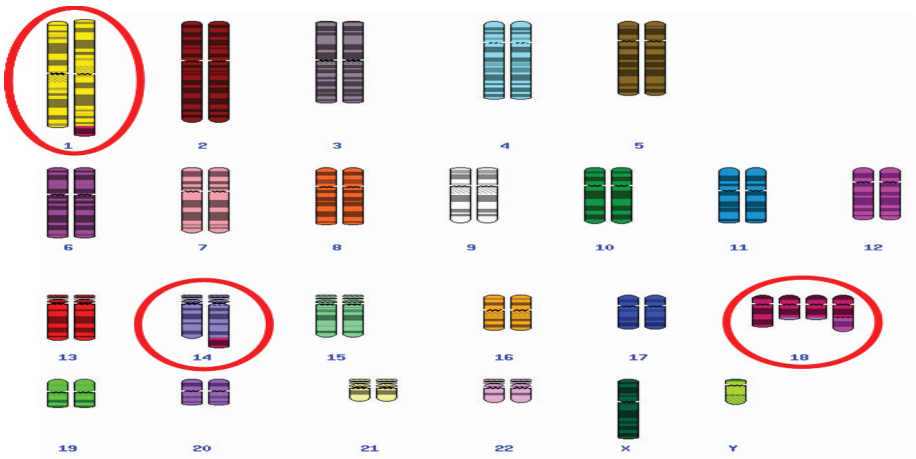


Fig. 1. A schematic view of an aberrant karyotype (produced by the SKYGRAM converter tool [1]). Chromosomes 1,14, and 18 show structural aberrations, and chromosome 18 shows a numerical aberration. (An ISCN description of this karyotype is 47,XY,der(1)t(1,18)(p36;q21),t(14,18)(q32;q21),+der(18)t(12;18)(p11;q21),+der(18)t(14;18).)

Over the last few decades, intensive research on chromosomal abberations in cancer has led to the accumulation of large amount of data on cancerous karyotypes. The largest available public depository of

such data is the Mitelman database [15], which contains over 50,000 karyotypes collected from over 8,000 publications. In this study we analyze this database. Our goal is to understand the main abberation types and their frequency in different cancers. Our hope is that such studies will provide insights and better understanding of the evolution of karyotypes in specific cancer types.

Traditionally, karyotypes have been constructed using chromosome staining methods, mostly G-banding. SKY [22] and M-FISH [25] are relatively new molecular cytogenetic techniques that permit the simultaneous visualization of all the chromosomes in different colors, considerably improving the detection of material exchange between chromosomes. The Mitelman database contains primarily karyotypes based on G-banding. The resolution and the detectable level of details in such karyotypes is lower than what can be observed with SKY and M-FISH or with novel high throughput methods (e.g. array-based CGH [24] and ESP [26]). Nevertheless, we chose to focus on the Mitelman database since it is the largest collection of cancerous karyotypes.

Karyotypes are usually described using the ISCN nomenclature [14]. In this system, every aberrant chromosome is described using specific rearrangement and numerical events, e.g., translocations, inversions, deletions, and duplications. Although ISCN attempts to describe the correct set of events leading to the observed karyotypes, it has almost no ability to do so when there are overlapping rearrangements, e.g. a chromosome involved in two translocations, each at a different position. Moreover, while the inference of the events is an easy task for many modestly rearranged karyotypes of hematological disorders, it can be a computationally hard task when the karyotypes are complex, as often happens in solid tumors.

There are many computational studies analyzing large data sets of cancerous genomes. Most of these analyses consider a cancerous genome as a collection of chromosomal abberations easily computed from the data. For example, in a series of studies, reviewed in [12], Högland et al. analyzed cytogenetic data from individual tumor types, by inspecting various parameters, including the number of gains or losses of genomic fragments, the number of aberrations, and the frequency at which bands are involved in breaks. In another study [21], Sankoff et al. compared the distributions of cancer-related breakpoints, derived from the Mitelman database, and evolutionary breakpoints, derived from a human-mouse comparative map. Another important branch of computational studies searches for statistical dependencies between chromosomal aberrations, usually in the form of tree or directed acyclic graph, such as [6, 7, 12, 11].

Chromosomal aberrations observed in cancer are by and large somatic and thus non-inheritable. When a rearrangement occurs in a genome of a germ-line cell, it can be inherited by offsprings. Indeed, the comparison of genomes of related species reveals that genome rearrangements play a significant role during the evolution of species. In a pioneering paper [20], Sankoff raised the problem of computing a shortest sequence of rearrangement operations between two given genomes, when genomes are represented by linear orders of oriented genes. Over the last fifteen years, this problem was intensively studied for many types of rearrangement events and their combinations, including inversions, translocations, block exchanges, deletions and insertions (see [4] for a review). All these studies ignored the *ploidy* in the genomes, i.e., the number of copies of each chromosome. Since numerical aberrations are prevalent in cancer, every model of cancer rearrangements must contain both numerical and structural events. This makes the reconstruction task more complicated and prevents direct use of results from the rich algorithmic literature on germ-line rearrangements.

The main purpose of this study was to estimate the prevalence of specific types of genome rearrangement events in cancer karyotypes. For this purpose we developed a new efficient heuristic for reconstructing a sequence of events that best explain the transformation from the normal karyotype into a given cancer karyotype. We applied this algorithm to over 40,000 karyotypes published in scientific literature, and collected statistics on event frequency across cancer types. The algorithm is deliberately simplistic, mimicking the process of detecting obvious events and "undoing" them, going back from the given karyotype towards the normal. As such, it does not guarantee finding the shortest solution or finding any solution. However, we reasoned that most reported karyotypes are of limited complexity and thus may be amenable to such approach. Reassuringly, over 98% of the karyotypes were solved by this method. Our study provides for the

first time a broad picture of event frequency in hematological and solid cancers. Our analysis shows that chromosome gains and losses, reciprocal translocations, and terminal deletions, dominate the evolution of cancer karyotypes. By using the event frequencies in each karyotype as its profile, we show that many different cancer types have clearly distinguishable profiles, which can be meaningful for further understanding of the cancers.

This paper is organized as follows. In Section 2 we provide a short background on chromosome aberrations in cancer. In Section 3 we present some basic statistics regarding the complexity of cancer karyotypes. In Section 4 we describe our heuristic for reconstructing genome rearrangement events for a given karyotype. The analysis of the reconstructed events is reported in Section 5. For lack of space, some details are deferred to an appendix.

2 Background

2.1 Mechanisms for chromosomal aberrations

Many molecular mechanisms are involved in the formation of chromosomal aberrations. The following mechanisms are reviewed in [2, 9, 16, 18].

A *double strand break* (DSB) is one of the frequent lesions in DNA. The repair of DSBs in eukaryotic cells is carried out by two main pathways: *non-homologous end joining* (NHEJ) and *homologous recombination* (HR). NHEJ repairs DSBs by directly re-ligating DNA ends, which may create a deletion if sequences surrounding the lesion were lost. Another potential risk of NHEJ is the ligation of two non-matching broken ends, leading to genome rearrangements. HR repairs breaks through interaction of a free DNA end with an intact homologous sequence, which is used as a template to copy missing information prior to religation. Because of the ability to fill in gaps by copying information from a sister chromatid or homologous chromosome, HR runs the risk of generating rearrangements through interaction of similar sequences on non-homologous chromosomes or regions. In particular, HR may extend to the end of a chromosome, resulting in a duplication of the whole "tail" of that chromosome.

Another possible lesion to the DNA is the loss of a telomere. The telomeres protect the ends of chromosomes from fusion with other ends. Thus a chromosome end that lacks a functioning telomere tends to be adhesive and may initialize a *breakage-fusion-bridge* process [13]. Stabilization of the genome occurs only through the net gain of a telomere, either through duplications of protected chromosome ends, or by direct telomere addition. Indeed, telomerase activity has been detected in the majority of malignant epithelial tumors [8].

A direct cleavage through a centromere generates two *telocentric* (i.e. single-arm) chromosomes, each containing a portion of the kinetochore (the functional component of an active centromere). Non-disjunction of sister chromatids of a telocentric chromosome results in the formation of an *isochromosome* or *isoderivative*, i.e. a chromosome with two identical, mirror-image arms.

As elaborated above, DSBs, telomeres dysfunction and centric fissions may lead to structural aberrations. Numerical aberrations may occur when genes involved in chromosome segregation or cytokinesis are deregulated. In particular, failure in cytokinesis (e.g. endomitosis) and multipolar mitoses may alter the ploidy of the genome.

2.2 The Mitelman database

The "Mitelman database of chromosome aberrations in cancer" [15] (henceforth abbreviated MD) contains the description of cancer karyotypes manually culled from the literature over the last twenty years. For our analysis we used the version of March 27, 2007, which contained 53,573 cancerous karyotypes, collected from 8748 published studies. The karyotypes in the database are represented in the ISCN format and can be automatically parsed and analyzed by the software package CyDAS [10]. We shall use here a simplified

version of ISCN for representing karyotypes (see Appendix A). We refer to a karyotype as *valid* if it can be parsed by CyDAS without any errors. According to our processing, 47,045 (87.8%) of the records were valid karyotypes.

2.3 Complex karyotypes

When the cytogeneticist analyzes a sample, several cells are checked. Each abberation described in a cancerous karyotype must be present in at least two cells in the described sample. In some cases the cell population may be non-homogeneous, and contain cells with several distinct karyotypes, resulting from evolution of the cell population during the development of the cancer. A homogeneous cell sample is described by a *simple karyotype*, and a non-homogeneous one has a *complex karyotype*, which consists of several karyotype species. In this study we derive simple karyotypes from complex karyotypes and analyze each of them independently.

About 17% of all valid karyotypes in MD are complex. The total number of simple (valid) karyotypes that we deduced from MD is 57941 (33% of which originate from complex karyotypes). For the rest of this paper we assume that every analyzed karyotype is simple.

3 Basic statistics on karyotype complexity

In this section we present some simple statistics based on the MD regarding the complexities of cancerous karyotypes. Human malignancies can be divided into two main categories: hematological disorders and solid tumors. Our first step was to distinguish between hematological malignancies and solid tumors. The type of neoplasia can be identified by its *morphology*, i.e. the cancer classification based on neoplasm histology, and its *topography*, i.e. the tumor site (applicable only for solid tumors). Based on the morphology and topography descriptors of each karyotype, we partitioned the karyotypes in the database into three categories:

- HEMA: hematological neoplasms, e.g.: leukemia, myeloma, lymphoma.
- BENIGN: solid benign tumors, e.g.: meningioma, leiomyoma, lipoma.
- SOLID: solid malignant tumors, e.g.:adenocarcinoma, Wilms tumor, malignant melanoma.

The HEMA category covers 71.2% of the valid simple karyotypes derived from the MD, while SOLID and BENIGN cover only 22.9% and 5.9% respectively. In the following, we compare the distributions of simple variables defined on karyotypes between these categories. We define a chromosome as *abnormal* if it does not match any chromosome in the standard normal karyotype. As expected, the distribution of the number of abnormal chromosomes per karyotype had the longest tail for solid tumors, while benign and hematological karyotypes seldom have more than five abnormal chromosomes (Fig. 5-a). The number of fragments (maximal contiguous interval in the normal) per an abnormal chromosome (Fig. 5-b) had a similar distribution across categories, with less than 1% of the abnormal chromosomes having four or more fragments. We defined *karyotype ploidy level* as $\lfloor \frac{n+11}{23} \rfloor$, where n is the total number of chromosomes. As expected, solid tumors tended to have higher ploidy, reflecting their higher complexity (Fig. 5-c). Multicentric chromosomes (i.e. chromosomes with more than one centromere) are considered non-stable, as each of the centromeres in these chromosomes may be passed to opposite poles in the mitotic anaphase. Interestingly, all three categories had some 2-4% of karyotypes with multicentric chromosomes (Fig. 5-d). Overall, the difference between the categories are quite subtle. Karyotypes of solid tumors, in particular malignant solid tumors, tend to have more complex abnormal chromosomes and ploidy changes, in comparison to hematological malignancies.

Do the statistics above - as well as those we shall report later - reflect the distributions of properties in cancer karyotypes "in the real world"? The answer is probably no. For example, although up to 80% of all human malignancies are solid, most of the karyotypes in MD belong to hematological malignancies.

One major reason for this bias is the difficulty in cytogenetically analyzing solid tumors. Solid tumor genomes often demonstrate poor visual quality during metaphase. Moreover, the karyotypes of solid tumors are often much more complex and thus more difficult to interpret. In addition, the database contains *reported* karyotypes from the literature, and there is a bias in this reporting. For example, the hematological karyotypes in MD are probably of higher complexity than those simple cases seen regularly in the clinic, which are not deemed publish-worthy as they are too simple or fully understood. While this means that the statistics we are collecting should be interpreted with caution, we believe they can still be useful in understanding how to model cancer evolution on the karyotype level and how different classes and subclasses differ.

4 A sorting algorithm

In this section we describe an algorithm, which we call SKS (Simple Karyotype Sorter), for reconstructing the sequence of rearrangement events (structural and numerical) that have led from the normal karyotype to a given cancer karyotype. We call this process *sorting* the karyotype. The SKS algorithm aims to mimic the intuitive way a cytogeneticist would perform this task, i.e., starting with the cancer karyotype and going backwards towards the normal karyotype one event at a time, taking the simplest and most evident step whenever possible. The SKS algorithm is a heuristic and does not guarantee finding an optimal or even finding any solution sequence when one exists. In Section 5 we shall report on the performance of this heuristic on the MD karyotypes.

4.1 An abstract data structure of a karyotype

A chromosome is *indefinite* if its description includes unknown items. For example, ?→? and 1pter→1p? are indefinite chromosomes. Note that a definite chromosome may contain uncertain items, e.g. 1pter→1p?12. Similarly, a karyotype is *definite* if it contains only definite chromosomes. In what follows we analyze only definite karyotypes, and ignore any uncertainties, e.g. 1p?12 will be considered as 1p12. As can be expected, the percentage of indefinite karyotypes in malignant solid tumors (39.6%) is higher than in hematological neoplasms (28%), and is the lowest for benign tumors (24.2%). Hence, the overall number of karyotypes we analyze here is 40,298.

We represent a karyotype K by the following abstract data structure:

- *Abnormal_Chrs*(K): A set of distinct, orientation-less, abnormal chromosomes. For each abnormal chromosome in *Abnormal_Chrs*(K) we maintain its multiplicity and list of fragments.
- *multiplicity*: a mapping assigning to each normal chromosome id (i.e. 1, ... ,22, X, Y) its multiplicity in K.

4.2 Orphan fragments

Denote by *Frags*(K) the multiset of fragments found in *Abnormal_Chrs*(K). A fragment in *Frags*(K) is *orphan* if there is no other fragment in *Frags*(K) from the same normal chromosome. For example, suppose *Abnormal_Chrs*(K) = {9pter → 9q32::1p36 → 1pter, 14qter → 14p21::9q32 → 9qter, 14p21 → 14qter} then *Frags*(K) = { 9pter → 9q32, 9q32 → 9qter, 14qter → 14p21 × 2, 1p36 → 1pter} and K contains exactly one orphan fragment: 1p36→1pter.

The easiest way to explain an occurrence of an orphan fragment is by a translocation event followed by a loss of one of the two resulting abnormal chromosomes. For an acentric orphan fragment there is an alternative, less conservative explanation: The orphan fragment resulted from a duplication during a process of HR DSB-repair (recall Section 2.1). In Section 5.2 we describe some statistics regarding acentric orphan fragments that suggest the latter explanation is more likely for many cases.

4.3 Algorithm SKS

The SKS algorithm computes a sequence of events $S = \rho_1, \ldots, \rho_t$ that transforms a normal karyotype into a given (cancerous) karyotype K. Starting from K and applying the corresponding inverse operations $S^{-1} = \rho_t^{-1}, \ldots, \rho_1^{-1}$ generates a normal karyotype. The SKS algorithm works in two phases. First, all the abnormal chromosomes are sorted. Then, simple numerical operations "correct" the multiplicities of the normal chromosomes.

We need a few definitions first. A fragment is *centric* if it contains a centromere, and *acentric* otherwise. Let f and g be two fragments from the same normal chromosome. The concatenation $f::g$ is an *adjacency* if f and g have exactly one shared band - which is their fused ends. For example, 1pter→1p11::1p11→1q22 is an adjacency. In this case, f and g are said to be *complementing*. Fragments $f, g \in \mathit{Frags}(K)$ are *uniquely* complementing if no other fragment $h \in \mathit{Frags}(K)$ is complementing to f or g. The types of rearrangement events that we consider will be introduced in the description of algorithm.

Initialization. We first detect simple changes in the karyotype ploidy as follows. Let μ and g be the the median and greatest common divisor of all distinct chromosome multiplicities (both normal and abnormal) respectively. Clearly, $\mu \geq g$. Suppose $g > 1$. In this case we divide all chromosome multiplicities by $d = g$. A single exception is when $\mu = g$ and g is even - in this case we divide by $d = g/2$ (instead of by g). If the chromosome multiplicities were changed (i.e. $d > 1$) - we set $S = \{\rho\}$, where ρ is a corresponding PLOIDY CHANGE event.

Phase I: Sorting the abnormal chromosomes. The abnormal chromosomes are sorted by repeatedly detecting and undoing one of the following events. The phase ends successfully if there are no more abnormal chromosomes, and ends with failure if there are still abnormal chromosomes but no additional event is detected.

- **CHR GAIN**: A *chromosome gain* is a duplication of a complete chromosome. To detect such event, seek an abnormal chromosome, *chr*, whose multiplicity, m, is greater than 1. Perform the inverse operation, i.e., the removal of one copy of *chr*, decreasing its multiplicity to $m - 1$.
- **ISOCHROMOSOME CREATION**: Detect any iso-chromosome or iso-derivative (see Sec. 2). Perform the inverse operation, by removing one of the identical arms.
- **TRANSLOCATION** and **FISSION**: A *translocation* is the exchange of tails between two chromosomes; a *fission* is the split of one chromosome into two contiguous segments. Let f and g be two uniquely complementing fragments found on different chromosomes. Then there are two possible cases. In the first case, the complementing ends of both f and g correspond to chromosome ends. In this case, a FISSION event is detected and the inverse operation is a simple fusion of f and g in their complementing ends (i.e. chromosome fusion). The latter case is when at least one of the complementing ends of f and g is fused to another fragment. In this case, a TRANSLOCATION event is detected and the inverse translocation that fuses the complementing ends of f and g is applied to K.
- **INVERSION**: An *inversion* is the reversal of a DNA segment within a chromosome. This event is detected for a pair of uniquely complementing fragments, f and g, on the same chromosome, that have different orientation. The inverse operation is an inversion that fuses the complementing ends of f and g. For example, suppose the chromosome containing f and g is of the form $f::h_1::{-g}::h_2$, where $-g$ is the inverse of g and $f :: g$ is an adjacency. In this case, the detected INVERSION event inverts the segment $h_1::{-g}$.
- **TANDEM DUP**: A *tandem duplication* creates two identical consecutive fragments on the same chromosome creating $h \equiv f_1 :: f_2 :: f_2 :: f_3$. For example, 1pter→1q44::1q31→1qter is a tandem duplication since 1pter→1q44 ≡ 1pter→1q31::1q31→1q44 and 1q31→1qter ≡ 1q31→1q44::1q44→1qter. When identifying such a repetition, simply remove it, forming $h \equiv f_1 :: f_2 :: f_3$.

- **INTERNAL DELETION**: An internal deletion of a fragment within a chromosome is discovered as follows. Detect a non-adjacency pair of concatenated fragments, f::g, for which there exists a fragment h such that *(i)* f::h and h::g are adjacencies, and *(ii)* h does not contain in its span any fragment in *Frags*(K). Replace f::g by fragment $f' \equiv f$::h::g.
- **TAIL DELETION**: A deletion of a chromosome tail (acentric end fragment) is detected by identifying an abnormal chromosome end lacking a pter or a qter, and whose complementing fragment, f, is *(i)* acentric and *(ii)* does not contain in its span any fragment in *Frags*(K). To undo the operation, concatenate f to the chromosome's end such that a new adjacency is formed.
- **ACENTRIC ORPHAN TAIL**: Detect an acentric orphan fragment f that is found on one end of an abnormal chromosome. Eliminate this aberration by a removal of f.
- **CENTRIC ORPHAN FUSION**: Detect a multicentric chromosome *chr* containing a centric orphan f. To undo the operation, perform a fission of *chr* near f such that each of the resulting two chromosomes contains a centromere.

Phase II: Gain/loss events and ploidy changes. If this phase is reached the current karyotype K satisfies *Abnormal_Chrs*$(K) = \emptyset$. Define $\mu(K)$ as the median multiplicity of all chromosomes in K (for gain/loss computations we consider the sex chromosomes as homologs). For any chromosome *chr* whose multiplicity differs from $\mu(K)$, adjust its ploidy to $\mu(K)$ by CHR LOSS or CHR GAIN events. Then, when the ploidy of all chromosomes is $\mu(K)$, adjust the ploidy globally to 2 by prepending a corresponding PLOIDY CHANGE event to S.

5 Experimental results

We ran algorithm SKS on each of the 40,298 definite simple karyotypes derived from MD. We say that a karyotype is *sortable* if SKS transforms it successfully to the normal karyotype. Table 1 shows that the vast majority (>98%) of the karyotypes are sortable. Hence, our rather naive heuristic, which makes only straightforward moves, performs very well on the MD karyotypes.

Table 1. Sortability of MD karyotypes. Numbers are percent out of the karyotypes in each category.

	HEMA	**BENIGN**	**SOLID**	**ALL**
Sortable - numerical aberration only	21.8%	41.1%	43.8%	27.4%
Sortable - with structural aberrations	76.7%	56.7%	54.3%	71.0%
Not sortable	1.5%	2.2%	1.9%	1.7%

5.1 Event rates

Figure 2-a presents the average number of each type of event per karyotype in our reconstruction. The most prevalent reconstructed events in all categories are chromosome gains and losses, tail deletions and translocations. In contrast, most other events are relatively rare, occurring in a tenth of the karyotypes or even less. For example, the translocation rate is 0.54 per karyotype, while inversion rate is only 0.06[1]. Note that while the events of chromosome gain and loss and tail deletion are dominant in the arrangement of malignant solid tumor karyotypes, translocations are relatively more frequent in hematological karyotypes.

Translocations are called *reciprocal* of both of the exchanged fragments are non-empty. Our analysis shows that most (>96%) reconstructed translocations are reciprocal (Fig. 2-b). Additional support to this observation is obtained by analyzing the breakpoint graphs of karyotypes (Appendix B). Interestingly, non-reciprocal translocations are more than twice as common in solid tumors than in hematological karyotypes.

[1] The surprisingly low inversion rate should be taken with caution: clearly, only relatively long inversions covering several bands are detectable in G-banded karyotypes in MD.

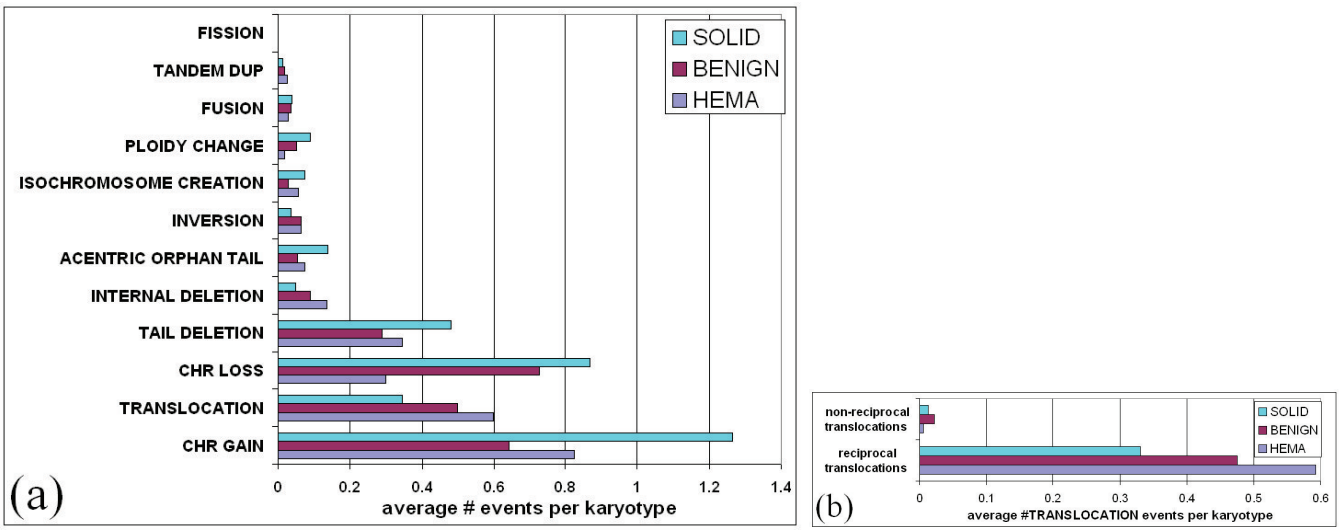


Fig. 2. Frequencies of each rearrangement event. Numbers are based on applying the sorting algorithm to all valid simple karyotypes in the database. (a) The average number of events per karyotype. (b) Average number of reciprocal and non-reciprocal translocations.

5.2 The origin of ACENTRIC ORPHAN TAILs

For a fragment $f \in \mathit{Frags}(K)$, let $\mathit{chr}(f)$ be the normal chromosome of f. Figure 3 presents the distributions $\mathit{multiplicity}(\mathit{chr}(f))$, for centric orphan fragments and for acentric orphan tail fragments. For comparison, we include the distribution of $\mathit{chr}(i)$, $i \in \{1, \ldots, 22\}$, after all abnormal chromosomes have been sorted (i.e. at the completion of Phase I of SKS algorithm). As can be expected, the ploidy of normal autosomal chromosomes is mostly 2. The ploidy of the normal chromosome of centric orphan fragments is usually 1. Thus the most reasonable explanation is that centric fragments evolved from normal chromosomes by translocations or tail deletions. Surprisingly, the ploidy of the normal chromosomes of acentric tail orphans is mostly 2. Since most (98%) of these acentric orphan fragments have one complete end (i.e. pter or qter), this suggests that many of these acentric orphan fragments are the result of a tail duplication event, caused by the HR DSB repair mechanism (see Section 2.1). The alternative scenario is a translocation event, and an additional event of chromosome gain. The latter explanation is more complex and hence less likely.

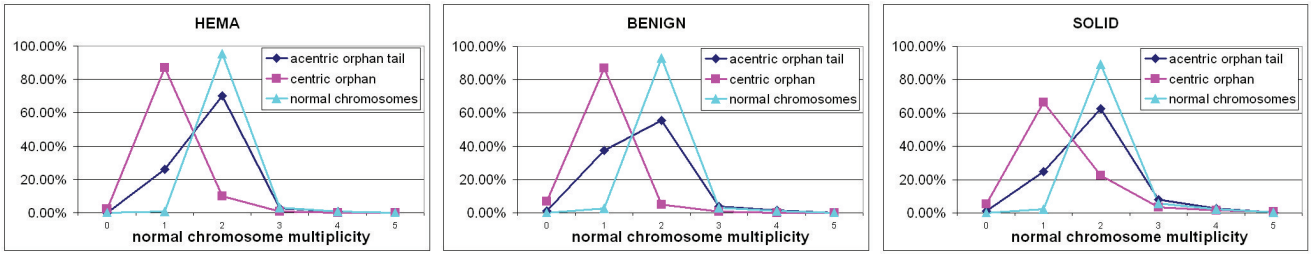


Fig. 3. Orphans and their parent chromosomes. The plots show the distributions of the multiplicity of normal chromosomes corresponding to acentric orphan tail fragments, and to centric orphan fragments. For comparison, each plot also includes the multiplicity of normal (autosomal) chromosomes, after all abnormal chromosomes have been sorted. The distributions are computed separately for categories HEMA, BENIGN and SOLID.

5.3 Rearrangement events as characteristics of cancer classes

Are the events that constitute the history of karyotypes, as reconstructed by the SKS algorithm, meaningful to understanding and distinguishing the different cancer types? To answer this question, we defined several similarity measures between distinct karyotypes, using the event rates reconstructed by the algorithm, and used them to compare cancer classes. Our analysis focused on karyotypes from 14 cancer classes, containing 60–885 karyotypes each (See Tables 2 and 3 for the class descriptions and detailed results). In our tests below we called a test *significant* if it attained p-value $< .0001$, after Bonferroni correction for multiple testing.

Clustering cancer classes by their event profiles. For a karyotype K we define its *event profile*, $\bar{v}(K)$, as a vector whose entries are the frequencies of each event in K (event order is as in Fig. 2a, bottom to top). For example, $\bar{v}(K) = (2, 0, 2, 1, 0, 1, 0, 0, 0, 0, 0, 0)$ for the karyotype K in Fig. 6. Given a set of karyotypes we define the *average event profile* as the coordinate-wise average of the event profiles of the karyotypes. Using Pearson correlation as a similarity measure, we applied an average linkage hierarchical clustering algorithm [23] on the average profiles of the 14 classes. As can be seen in Fig. 7, related cancers tend to cluster close to each other, implying they have similar average event profiles.

Partitioning karyotypes by event profiles. Let C_1 and C_2 be two distinct cancer classes, and let $\Omega = C_1 \cup C_2$. Can the karyotypes in Ω be distinguished, as to which belongs to C_1 and which belongs to C_2, by their event profiles? We partitioned Ω into two clusters, D_1 and D_2 ($\Omega = D_1 \cup D_2$), by applying k-means clustering [23], with $k = 2$, on the event profiles in Ω, and using Pearson correlation as the similarity measure. We measured the p-value of the correspondence between the new partition, $\{D_1, D_2\}$, and the original one, $\{C_1, C_2\}$, using the hypergeometric distribution (see Appendix C for details). We performed this test for all $\binom{14}{2} = 91$ pairs of classes. 26 (28.6%) of the tested pairs were significant.

Partitioning karyotypes by total event frequency. We define *NEvents* as the total number of reconstructed events for the karyotype (i.e., the sum of the entries in $\bar{v}(K)$). Given $\Omega = C_1 \cup C_2$ as before and an integer t, let $D_1^{(t)} = \{K \in \Omega : \text{NEvents}(K) \leq t\}$ and $D_2^{(t)} = \{K : \text{NEvents}(K) > t\}$. We computed the p-value of the match between $\{D_1^{(t)}, D_2^{(t)}\}$ and the original partition, for $t = 0, \ldots, 9$. 45 of the 91 pairs (49.5%) had a significant NEvents-based partition. We repeated the same test with the *NAPT* score [12], which is the number of aberrations in the karyotype's ISCN description[2]. NEvents and NAPT are different indicators of a karyotype's complexity. Interestingly, although NAPT is much less exact than NEvents, 53.8% of the tested pairs had a significant NAPT-based partition. A possible explanation is that the relatively large differences between the classes are captured better by a cruder measure. On the other hand, there is meaningful additional information in individual events. For example, 76.9% of the significant partitions based on event profiles had p-values lower than the corresponding partitions based on NEvents and NAPT, and 6 (14.3%) of the non-significant NAPT-based partitions had corresponding significant partitions based on event profiles.

Partitioning karyotypes using a single type of event. For each type of event, e, let *SEvent(e)* be the number of reconstructed events from type e. For example, SEvent(CHR GAIN) is the number of CHR GAINs (i.e. the first entry in the event profile). Our last test was to partition Ω using SEvent(e), for each type of event e, in the same fashion as above. Due to the relatively low values, we checked only five thresholds ($t = 0 \ldots 4$) for each type of event. Surprisingly, 81.3% of the tested pairs had a significant SEvent-based partition. The lowest p-values were achieved for partitions based on TRANSLOCATIONs (35.6%), CHR LOSSes (27.4%), and CHR GAINs (16.9%).

[2] The NAPT score is calculated by simply counting the number of comma-separated tokens in the ISCN description, disregarding the first two tokens that correspond to the total number of chromosomes and the sex chromosomes description. For example, the NAPT score for the karyotype in Fig. 1 is 5.

6 Conclusion

In this paper we presented novel methods for analyzing and comparing aberrant karyotypes observed in hematological malignancies and in solid tumors cells. We presented a simple yet effective heuristic (the SKS algorithm) for sorting aberrant karyotypes. On over 40,000 karyotypes of the Mitleman database, the algorithm attained a very high success rate (98%) in sorting the karyotypes. We believe that this shows that on such karyotypes of moderate complexity, the set of rearrangement events reconstructed by our algorithm (though not necessarily their order) is a close approximation of the actual gross chromosomal rearrangements that occurred in their evolution. Our analysis implies that the evolution of aberrant karyotypes in somatic cells is dominated by four events: chromosome gains and losses, reciprocal translocations and terminal deletions. The prevalence of chromosome gains and losses is expected, since these events are more easily detected than other more local events, e.g. inversions. Nevertheless, these results emphasize that duplication and deletion events must play a key role in any computational modeling of genome rearrangements in cancer.

By using clustering techniques, we demonstrated that karyotypes belonging to the same cancer class have characteristic event rates, since they often have more similar event frequencies than karyotypes belonging to different classes. Moreover, this suggests that carcinogenesis involves different pathways of gaining chromosomal aberrations for different cancer classes, and further analysis may shed light on the events characterizing different pathways.

One of the goals of this study was to lay the factual foundations for proposing a mathematical model of somatic genome rearrangements that will allow an accurate, non-heuristic systematic analysis of aberrant karyotypes. The simplest model that can generate the spectrum of the aberrations observed in cancerous karyotypes includes four types of events: chromosome gain and loss, breakage, and fusion. For example, a reciprocal translocation can be mimicked by two breaks followed by two fusions. While this simplistic model favors non-reciprocal translocations over reciprocal ones, our study observed the opposite preference in the MD karyotypes. Thus, a more realistic model should consider reciprocal translocations as atomic operations, to reflect the increased probability of their occurrence. Another operation that is worth considering is the duplication of a segment in an existing chromosome (see Section 5.2). Our hope is that a computational investigation of many reconstructed rearrangement sequences will help in pointing out the dominant scenarios through which chromosomal aberrations evolve in specific types of cancer.

Acknowledgments

We are grateful to Igor Ulitsky for his tremendous help in analyzing the event rate profiles, and to Gideon Rechavi, Luba Trakhtenbrot, and Chaim Linhart for helpful discussions and insightful comments. We thank Felix Mitelman and John Wiley & Sons, Inc. for granting us permission to analyze the data in the Mitelman database of chromosome aberrations in cancer.

References

1. NCI and NCBI's SKY/M-FISH and CGH Database, 2001. `http://www.ncbi.nlm.nih.gov/sky/skyweb.cgi`.
2. D.G. Albertson, C. Collins, F. McCormick, and J. W. Gray. Chromosome aberrations in solid tumors. *Nature Genetics*, 34:369–376, 2003.
3. V. Bafna and P. A. Pevzner. Genome rearragements and sorting by reversals. *SIAM Journal on Computing*, 25(2):272–289, 1996.
4. G. Bourque and L.Zhang. Models and methods in comparative genomics. *Advances in Computers*, 68:60–105, 2006.
5. A. de Klein et al. A cellular oncogene is translocated to the philadelphia chromosome in chronic myelocytic leukaemia. *Nature*, 300:765–767, 1982.
6. R. Desper, F. Jiang, O. Kallioniemi, H. Moch, C. Papadimitrou, and A. Schäffer. Inferring tree models for oncogenesis from comparative genome hybridization data. *Journal of Computational Biology*, 6:37–51, 1999.

7. R. Desper, F. Jiang, O. Kallioniemi, H. Moch, C. Papadimitrou, and A. Schäffer. Distance-based reconstruction of tree models for oncogenesis. *Journal of Computational Biology*, 7:789–803, 2000.
8. G. Krupp et al. Telomerase, immortality and cancer. *Biotechnology Annual Review*, 6:103–140, 2000.
9. D.O. Ferguson and W.A. Frederick. DNA double strand break repair and chromosomal translocation: Lessons from animal models. *Oncogene*, 20(40):5572–5579, 2001.
10. B. Hiller, J. Bradtke, H. Balz, and H. Rieder. CyDAS: a cytogenetic data analysis system. *BioInformatics*, 21(7):1282–1283, 2005. `http://www.cydas.org`.
11. M. Hjelm, M. Höglund, and J. Lagergren. New probabilistic network models and algorithms for oncogenesis. *Journal of Computational Biology*, 13(4):853 –865, 2006.
12. M. Höglund, A. Frigyesi, T. Säll, D. Gisselsson, and F. Mitelman. Statistical behavior of complex cancer karyotypes. *Genes, Chromosomes and Cancer*, 42(4):327–341, 2005.
13. B. McClintock. The stability of broken ends of chromosomes in zea mays. *Genetics*, 26(2):234–282, 1941.
14. F. Mitelman, editor. *ISCN (1995): An International System for Human Cytogenetic Nomenclature*. S. Karger, Basel, 1995.
15. F. Mitelman, B. Johansson, and F. Mertens (Eds.). Mitelman Database of Chromosome Aberrations in Cancer, 2007. `http://cgap.nci.nih.gov/Chromosomes/Mitelman`.
16. J.P. Murnane and Laure Sabatier. Chromosome rearrangements resulting from telomere dysfunction and their role in cancer. *BioEssays*, 26:1164–1174, 2004.
17. P.C. Nowell and D.A. Hungerford. A minute chromosome in human chronic granulocytic leukemia. *Science*, 132:1497, 1960.
18. J. Perry, H.R. Slater, and K.H.A Choo. Centric fission simple and complex mechanisms. *Chromosome Research*, 12(6):627–640, 2004.
19. J.D. Rowley. A new consistent chromosomal abnormality in chronic myelogenous leukaemia identified by quinacrine fluorescence and giemsa staining. *Nature*, 243:290–293, 1973.
20. D. Sankoff. Edit distance for genome comparison based on non-local operations. *Lecture Notes in Computer Science*, 644:121–135, 1992.
21. D. Sankoff, M. Deneault, P. Turbis, and C. Allen. Chromosomal distributions of breakpoints in cancer, infertility, and evolution. *Theoretical Population Biology*, 61(4):497–501, 2002.
22. E. Schröck, S. du Manoir, T. Veldman, B. Schoell B, J. Wienberg, M.A. Ferguson-Smith, Y. Ning Y, D.H. Ledbetter, I. Bar-Am, D. Soenksen D, Y. Garini, and T. Ried. Multicolor spectral karyotyping of human chromosomes. *Science*, (5274):494–497, 1996.
23. R. Shamir, A. Maron-Katz, A. Tanay, C. Linhart, I. Steinfeld, R. Sharan, Y. Shiloh, and R. Elkon. Expander: an integrative suite for microarray data analysis. *BMC Bioinformatics*, 6(232), 2005.
24. A. M. Snijders and N. Nowak et al. Assembly of microarrays for genome-wide measurement of DNA copy number. *Nature Genetics*, 29:263–264, 2001.
25. M.R. Speicher, S.G. Ballard, and D.C. Ward. Karyotyping human chromosomes by combinatorial multi-fluor FISH. *Nature Genetics*, 12(4):368–375, 1996.
26. S. Volik and S. Zhao et al. End-sequence profiling: Sequence-based analysis of aberrant genomes. *Proceedings of the National Academy of Science USA*, 100:7696–7701, 2003.

Appendices

A Formal representation of karyotypes

A chromosome is divided by its centromere into two *arms*: a short arm, denoted p, and a long arm, denoted q. Every chromosome arm is partitioned into bands. The bands in each arm are numbered, starting from the centromere, whose assigned to the number 10. The symbol *ter* indicates the (normal) end of a chromosome arm. A position in the chromosome is identified by three fields: *(i)* chromosome, *(ii)* arm, and *(iii)* band designation (either a number or ter). For example, 1p11 corresponds to band 11 in the long arm of chromosome 1; 2p10 and 2q10 both refer to the centromere of chromosome 2; 3pter is the (normal) end of the short arm of chromosome 3.

We refer to a chromosome as *abnormal* if its structure is abnormal. Abnormal chromosomes are defined by their band composition. In the following, we describe abnormal chromosomes in a similar (but not identical) manner to the *detailed system* of ISCN [14]. The term *fragment* refers to a continuous interval of a normal chromosome, identified by the positions of its two ends. When a fragment appears in a chromosome

it has an orientation, denoted by an arrow symbol → between its two ends. For example, 2p12→2qter is a fragment of chromosome 2 that starts in band 2p12 and ends in band 2qter. Two fragments are *identical* if the corresponding chromosome intervals are identical (disregarding orientation). A double colon (::) indicates a concatenation of two fragments. For example, a concatenation of 1p36→1pter to the end of 9pter→9q32 is denoted as 9pter→9q32::1p36→1pter. An abnormal chromosome is presented as a list a concatenation of fragments[3].

The description of a karyotype may contain question marks (?) to indicate uncertainties or unknown items. A question mark may be placed either before an uncertain item, or it may replace an unknown chromosome, arm, or band designation. For example, 1p?12 indicates a questionable identification of band number; 5p? represents an unknown band designation.

B Using cycles and paths for analyzing translocation types

For a cancerous karyotype K we define its *breakpoint graph*, $G(K)$, similarly to [3], as follows. The vertices of $G(K)$ are the ends of the fragments in *Frags*(K). The edges in $G(K)$ are colored either black or gray. Black edges correspond to fused ends in K. Grey edges correspond to complementing ends. For an example, see Fig. 6-c-1.

Let S be a sequence of events reconstructed for K by SKS. Each of the inverse operations for INVERSION, TRANSLOCATION, and FISSION events, forms one or two new adjacencies by fusing complementing ends. Let $G(K,S)$ be the subgraph of $G(K)$ induced by *(i)* the set of black edges, and *(ii)* the grey edges corresponding to pairs of fused complementing ends during the reconstruction of INVERSION, TRANSLOCATION, and FISSION events in S. See Fig. 6-c-2 for an example. It follows that $G(K,S)$ is composed of simple cycles and paths. The *length* of a cycle or path in $G(K,S)$ is the number of grey edges in it. Note that while a path of size l corresponds to l reconstructed events, a cycle of the same length corresponds only to $l-1$ events. We define the *caliber* of a path or cycle to be the number of corresponding events. A path or a cycle with caliber greater than 1 imply a *breakpoint reuse*, i.e. a break of a formerly created fusion. Figure 4 depicts the average numbers of cycles and paths in a karyotype, for each caliber. It is quite clear that cycles are much more prevalent than paths, even in solid tumors, which indicates that reciprocal translocations are indeed more favored than non-reciprocal ones. Moreover, both structures, cycles and paths, usually have a small caliber.

C Measuring the significance of a partition

In this section we describe the standard hypergeometric score that was used for evaluating the match of two partitions. Let $\{C_1, C_2\}$ and $\{D_1, D_2\}$ be two partitions of Ω. Let $n = |\Omega|$, $n_1 = |C_1|$, $m = |D_1|$, and $k = |C_1 \cap D_1|$. Hence $k \leq \min\{n_1, m\}$. The significance of the correspondence between $\{D_1, D_2\}$ and $\{C_1, C_2\}$ can be evaluated by the probability of having $|C' \cap D_1| \geq k$ where $C' \subset \Omega$ is randomly chosen and $|C'| = n_1$. This probability is given by:

$$p(n, m, n_1, k) = \sum_{i=k}^{\min\{n_1,m\}} \frac{\binom{m}{i}\binom{n-m}{n_1-i}}{\binom{n}{n_1}}$$

The smaller $p(n, m, n_1, k)$, the more significant the correspondence between D_1 and C_1. To compare D_1 with C_2, we compute $p(n, m, n-n_1, m-k)$. The final p-value for the partition $\{D_1, D_2\}$ is thus

$$p\text{-value}(\{D_1, D_2\}, \{C_1, C_2\}) = 2\min\{p(n, m, n_1, k), p(n, m, n-n_1, m-k)\}.$$

(The multiplier 2 is due to Bonferroni correction for multiple testing.)

[3] The exception for this are homogenously staining regions (HSRs), which are regions that contain multiple copies of small DNA fragments. Thus a stained HSR is uniform in appearance (no bands) and its content cannot be identified by cytogenetic methods.

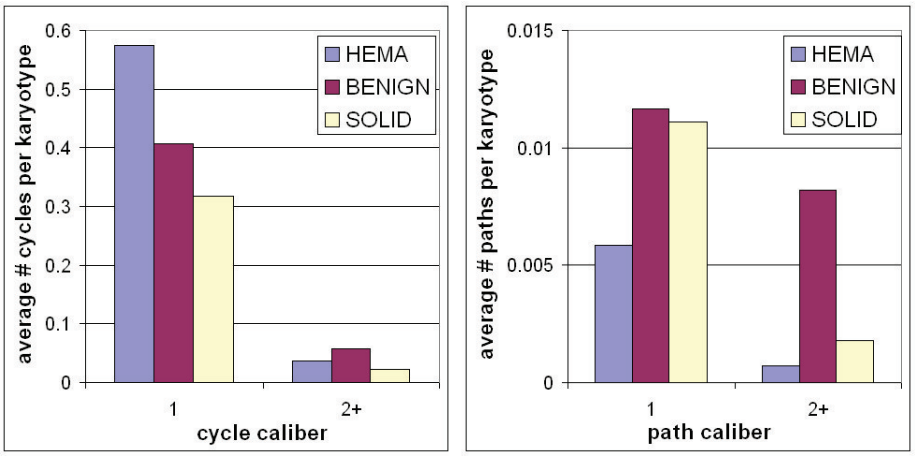


Fig. 4. The distributions of the average numbers of cycles and paths in a karyotype.

Table 2. Cancer classes.

class ID	class name	#karyotypes
27	HEMA-Acute monoblastic leukemia without differentiation (FAB type M5a)	332
28	HEMA-Refractory anemia with excess of blasts	885
31	HEMA-Refractory anemia	875
34	HEMA-Refractory anemia with ringed sideroblasts	230
36	HEMA-Acute myeloblastic leukemia with minimal differentiation (FAB type M0)	286
43	SOLID-Adenocarcinoma-Breast	590
52	HEMA-Acute monoblastic leukemia with differentiation (FAB type M5b)	196
58	HEMA-Refractory anemia with excess of blasts in transformation	424
70	SOLID-Adenocarcinoma-Kidney	859
111	BENIGN-Benign epithelial tumor special type-Breast	97
112	SOLID-Adenocarcinoma-Large intestine	208
118	BENIGN-Adenoma-Large intestine	149
143	SOLID-Adenocarcinoma-Ovary	119
577	BENIGN-Benign epithelial tumor NOS-Breast	60

Table 3. Partition p-values for pairs of cancer classes in Table 2. The p-values presented are after the Bonferroni correction for multiple testing.

Class 1	Class 2	event profile	NEvents	NAPT	SEvent
27	28	1.00E+00	3.11E-03	7.11E-03	3.32E-69
27	34	1.13E-04	7.99E-03	5.15E-03	1.85E-46
27	43	4.50E-13	4.52E-03	2.05E-05	8.04E-37
27	58	1.18E-06	1.00E+00	1.00E+00	3.49E-30
27	111	2.82E-01	3.38E-01	5.89E-02	1.01E-10
27	118	1.48E-31	5.12E-05	8.54E-05	4.73E-43
27	577	8.92E-23	1.02E-14	5.15E-18	1.02E-24
28	34	1.00E+00	1.00E+00	1.00E+00	1.00E+00
28	43	1.00E+00	3.33E-06	1.07E-02	3.41E-16
28	58	1.00E+00	4.17E-01	7.66E-01	1.97E-03
28	111	1.00E+00	1.00E+00	3.35E-02	6.97E-03
28	118	1.36E-01	3.33E-07	9.73E-08	1.58E-23
28	577	1.00E+00	4.47E-18	8.04E-25	5.11E-19
31	36	2.57E-02	1.49E-04	2.84E-06	8.75E-21
31	52	1.00E+00	2.48E-01	1.00E+00	1.72E-50
31	70	1.56E-15	7.16E-74	1.05E-92	1.96E-92
31	112	1.06E-08	2.49E-22	5.80E-22	8.68E-22
31	143	1.00E-13	6.67E-22	7.74E-27	1.92E-20
34	36	2.59E-01	6.59E-01	1.00E+00	7.52E-08
34	52	1.00E+00	3.78E-01	1.00E+00	2.48E-26
34	70	1.90E-01	1.21E-25	2.68E-25	1.76E-29
34	112	8.69E-04	8.19E-07	8.94E-07	1.31E-08
34	143	1.93E-03	1.13E-09	2.63E-10	6.71E-09
36	43	1.00E+00	3.29E-02	5.67E-01	5.60E-01
36	58	1.00E+00	1.00E+00	1.00E+00	2.54E-02
36	111	3.60E-01	1.00E+00	3.15E-01	1.00E+00
36	118	6.02E-09	1.24E-04	1.69E-04	2.70E-10
36	577	1.00E+00	4.91E-14	6.17E-17	4.32E-17
43	58	1.17E-01	7.66E-01	1.26E-01	3.42E-10
43	111	1.00E+00	1.00E+00	1.00E+00	1.00E+00
43	118	1.00E+00	2.17E-02	1.32E-04	4.09E-14
43	577	3.10E-10	2.27E-10	5.85E-16	1.36E-15
52	70	2.39E-63	9.88E-31	1.96E-31	1.46E-53
52	112	7.15E-40	8.01E-10	1.61E-08	1.10E-30
52	143	1.01E-19	1.02E-13	7.03E-13	7.22E-15
58	70	1.00E+00	1.82E-28	2.61E-30	1.15E-31
58	112	1.00E+00	6.64E-05	1.45E-04	6.24E-08
58	143	1.00E+00	3.72E-07	2.73E-08	5.12E-09
70	111	1.00E+00	9.84E-10	1.01E-10	5.51E-14
70	118	6.20E-11	4.41E-09	2.25E-05	2.97E-21
70	577	2.46E-02	5.29E-02	3.37E-06	2.78E-08
111	118	1.00E-06	2.51E-01	2.38E-02	1.09E-12
111	577	1.00E+00	7.50E-08	1.02E-10	9.46E-09
112	143	1.00E+00	1.59E-01	8.58E-01	1.00E+00
118	143	3.01E-01	2.80E-01	1.00E+00	8.93E-02
143	577	1.85E-04	1.79E-02	5.83E-04	7.26E-04

Class 1	Class 2	event profile	NEvents	NAPT	SEvent
27	31	1.00E+00	2.63E-09	1.78E-10	1.45E-98
27	36	2.69E-09	1.00E+00	1.00E+00	4.50E-17
27	52	1.00E+00	7.68E-02	1.10E-02	1.00E+00
27	70	1.00E+00	1.18E-25	2.95E-27	3.21E-83
27	112	3.42E-17	1.12E-07	4.47E-07	3.93E-51
27	143	6.47E-31	6.05E-10	1.42E-09	7.13E-26
28	31	1.00E+00	1.47E-02	1.78E-06	1.60E-10
28	36	1.00E+00	1.00E+00	1.00E+00	2.68E-07
28	52	1.00E+00	4.78E-02	5.89E-02	6.32E-32
28	70	1.00E+00	1.36E-55	1.33E-55	2.83E-45
28	112	1.00E+00	2.06E-12	2.36E-11	8.86E-11
28	143	9.20E-01	1.97E-13	3.61E-13	4.90E-11
31	34	1.00E+00	7.67E-01	1.67E-01	1.90E-01
31	43	9.10E-03	1.55E-14	1.17E-12	2.06E-38
31	58	4.16E-01	4.92E-07	5.52E-08	6.22E-15
31	111	1.00E+00	1.23E-03	6.85E-07	2.26E-09
31	118	1.88E-22	5.32E-14	3.94E-17	1.31E-33
31	577	1.00E+00	1.52E-26	7.82E-34	6.39E-30
34	43	1.00E+00	9.69E-03	3.03E-01	6.82E-09
34	58	1.00E+00	1.00E+00	1.00E+00	1.17E-04
34	111	1.00E+00	1.00E+00	9.73E-02	2.77E-03
34	118	2.80E-15	3.86E-05	5.25E-05	1.23E-18
34	577	1.00E+00	2.19E-14	3.24E-17	8.09E-13
36	52	2.22E-06	9.93E-02	6.79E-03	6.37E-06
36	70	1.00E+00	8.34E-23	3.47E-24	3.47E-39
36	112	4.51E-01	8.74E-07	2.73E-06	2.37E-11
36	143	3.13E-05	2.72E-09	5.01E-09	1.26E-08
43	52	1.61E-03	5.65E-06	6.87E-03	6.59E-15
43	70	1.00E+00	1.13E-33	1.36E-39	4.04E-66
43	112	1.41E-03	1.05E-02	3.59E-04	5.87E-12
43	143	1.00E+00	1.08E-04	1.69E-07	7.25E-06
52	58	1.30E-13	9.73E-04	3.28E-02	2.68E-12
52	111	1.00E+00	3.56E-02	1.08E-02	1.40E-04
52	118	4.17E-33	4.07E-08	2.86E-07	5.38E-27
52	577	5.81E-20	2.43E-17	3.39E-18	2.11E-23
58	111	1.00E+00	1.00E+00	6.47E-01	2.08E-02
58	118	2.52E-03	9.20E-03	4.91E-03	3.66E-15
58	577	1.00E+00	6.84E-12	2.43E-15	4.37E-15
70	112	5.15E-01	5.92E-12	3.21E-07	3.82E-16
70	143	2.90E-01	3.03E-01	2.42E-01	1.62E-13
111	112	5.61E-01	4.46E-01	6.85E-01	8.46E-05
111	143	1.00E+00	4.56E-03	2.72E-02	1.33E-04
112	118	2.77E-03	1.00E+00	1.00E+00	6.93E-05
112	577	5.07E-05	4.23E-07	2.92E-07	4.15E-05
118	577	4.25E-12	8.22E-04	7.80E-05	1.43E-10

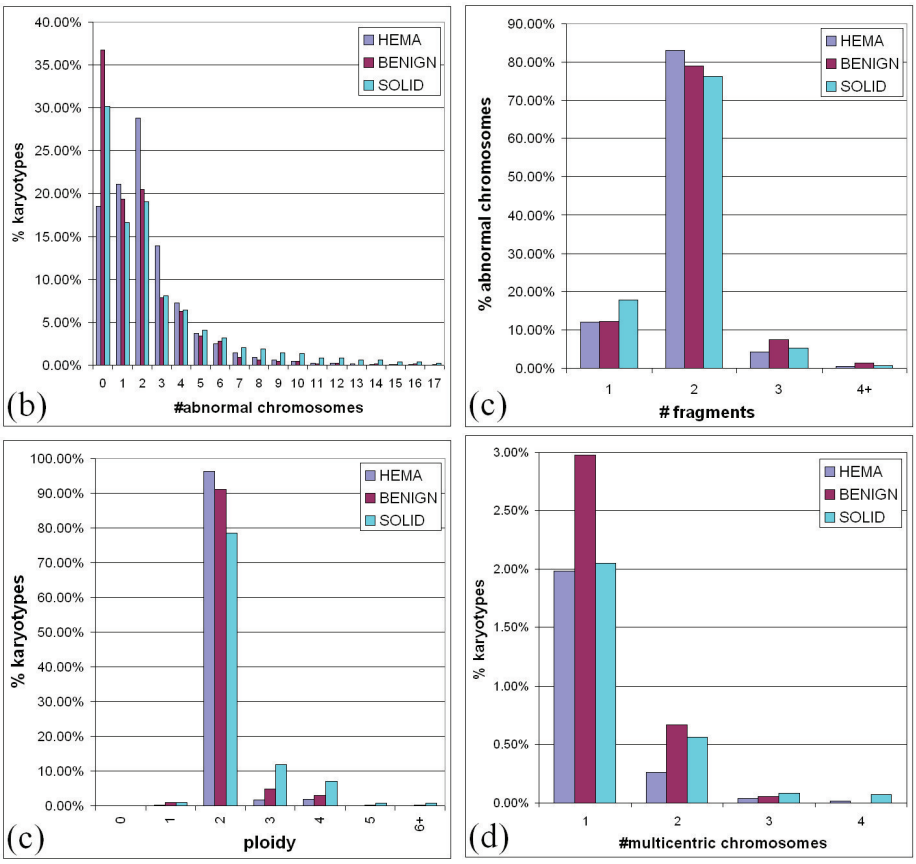


Fig. 5. Basic statistics on karyotype complexity in the Mitelman database. (a) The distribution of the number of abnormal chromosomes per karyotype. (b) The number of fragments per abnormal chromosome. (c) The distribution of karyotype ploidy. (d) The distribution of number of multicentric chromosomes per karyotype. More than 97% of all the karyotypes have no multicentric chromosomes.

(a) An abstract data structure for a karyotype K:

$$Abnormal\_Chrs = \left\{ \begin{array}{c} 18\text{pter} \rightarrow 18\text{q21}::12\text{p11} \rightarrow 12\text{pter}, \\ 1\text{qter} \rightarrow 1\text{p36}::18\text{q21} \rightarrow 18\text{qter}, \\ 14\text{pter} \rightarrow 14\text{q32}::18\text{q21} \rightarrow 18\text{qter}, \\ 18\text{pter} \rightarrow 18\text{q21}::14\text{q32} \rightarrow 14\text{qter} \times 2 \end{array} \right\}$$

$multiplicity[1] = multiplicity[14] = multiplicity[18] = 1$, $multiplicity[i] = 2$ for $i \notin \{1, 14, 18\}$

(b) A sequence of reconstructed events S:

1. ACENTRIC ORPHAN TAIL:	12p11→12pter,
2. CHR GAIN:	18pter→18q21::14q32→14qter,
3. TRANSLOCATION(reciprocal):	14pter→14q32, 14q32→14qter,
4. TRANSLOCATION(non-reciprocal):	18pter→18q21, 18q21→18qter,
5. TAIL DELETION:	1p36→1pter
6. CHR GAIN:	18

(c) The breakpoint graph $G(K)$ (1) and its induced subgraph $G(K, S)$

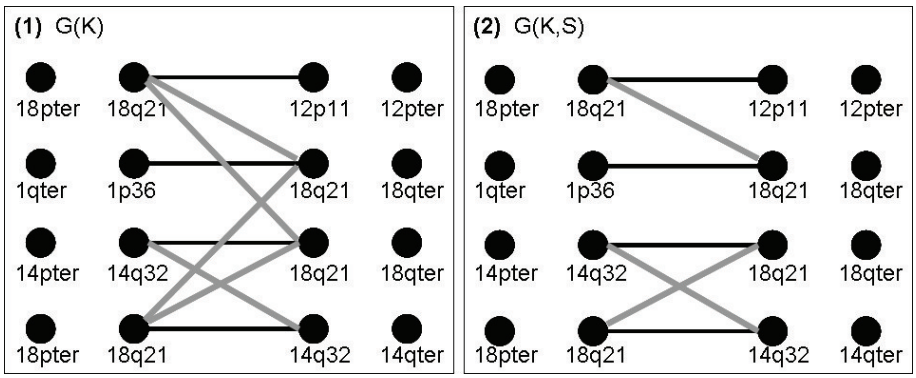


Fig. 6. An analysis of the karyotype in Fig. 1.

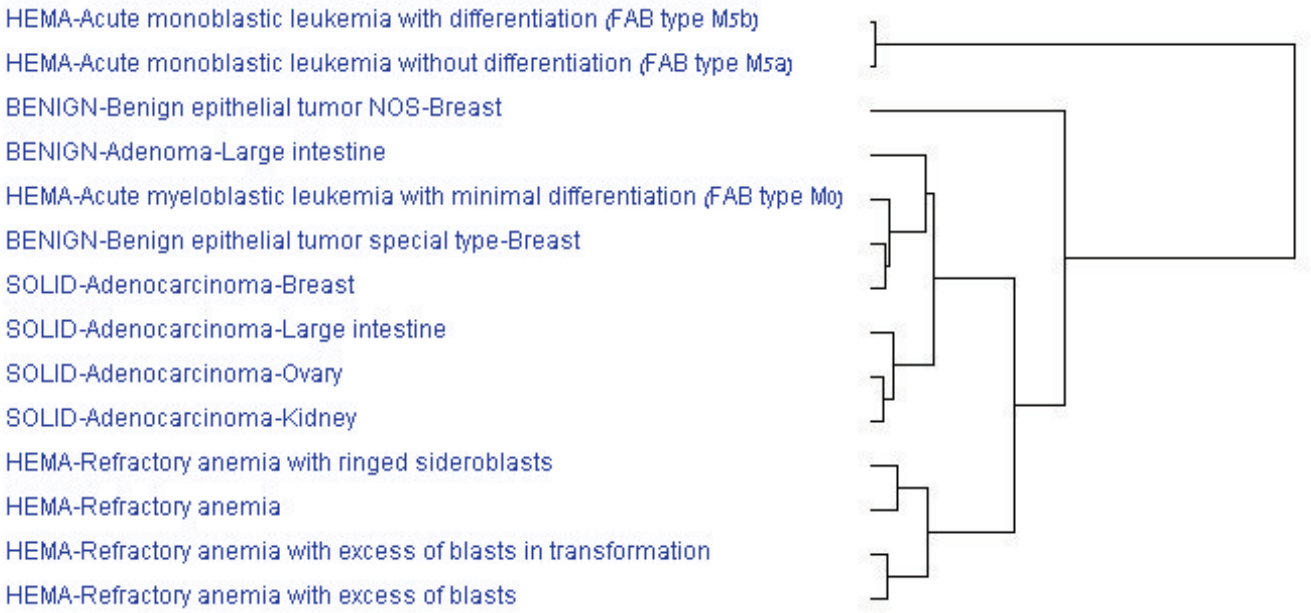


Fig. 7. An hierarchical clustering of different cancer classes based on their average event profiles, using Pearson correlation as similarity function. Each cancer is identified by its category, morphology, and topography (if it is a solid tumor).

Chapter 7

A Systematic Assessment of Associations among Chromosomal Aberrations in Cancer Karyotypes

A systematic assessment of associations among chromosomal aberrations in cancer karyotypes

Michal Ozery-Flato[a,b], Chaim Linhart[a], Luba Trakhtenbrot[c,d], Shai Izraeli[c,e,f], and Ron Shamir[a]

[a] The Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv, 69978, Israel; [b] Machine learning and data mining group, IBM Haifa Research Lab, Israel;[c] Chaim Sheba Cancer Research Center, [d] Institute of Hematology, and [e] Department of Pediatric Hemato-Oncology, Sheba Medical Center, Tel Hashomer, Israel; [f] Sackler School of Medicine, Tel Aviv University, Tel Aviv 69978, Israel.

Address for correspondence:
Ron Shamir, Ph.D.
Blavatnik School of Computer Science
Tel Aviv University
Tel Aviv, 69978, Israel
Tel: 972-3-5383
Fax: 972-3-5384
Email: rshamir@post.tau.ac.il

ABSTRACT

Chromosomal aberrations are a hallmark of cancer. Certain ones are known to be strongly connected with specific cancers, while many others appear to be nonspecific and arbitrary. We report on a systematic study of the characteristics of chromosomal aberrations in cancers, using the largest repository of reported karyotypes, the Mitelman database. We compared cancer types by their manifested aberrations and drew an aberration-similarity map of them. In addition to being highly concordant with the histological classification of cancers, the map also revealed novel similarities, such as between three embryonic tumors– Wilms' tumor, Ewing's sarcoma, and Hepatoblastoma. In another analysis we discovered that chromosome gains tended to co-occur with other chromosome gains, and losses with losses. This discovery was confirmed on an independent comparative genomic hybridization dataset of cancer samples. It suggests that aneuploid cancer cells may use extra chromosome gain / loss events to restore a balance in their altered proteins ratios.

Our results assign solid statistical foundations to many findings reported in the literature, and reveal novel observations that merit further research. An accompanying website summarizes all the discovered associations and allows easy search, filtering and sifting through the results, as well as direct viewing of the relevant karyotypes in the Mitelman database.

INTRODUCTION

Most cancer genomes undergo large scale alterations that dramatically alter their content and structure (1). This phenomenon of genomic instability is responsible for the wide repertoire of chromosomal aberrations observed in cancer genomes. While the role of most aberrations in the carcinogenesis process remains to be determined, the common perception (2) is that some of these aberrations are functionally important to the initiation and growth of cancer (drivers), while others merely represent random somatic changes that carry no selective advantage to the cancer cell (passengers). The identification of strong associations among aberrations, i.e. associations that are observed significantly more than expected by chance, may help in the detection of driver aberrations or point to mechanisms that promote the selection of certain aberrations. As data on chromosomal aberrations in cancer accumulate, the detection of such strong associations can become more accurate and powerful.

Following the four-step model for colorectal cancer evolution suggested by Vogelstein et al.(3, 4), several computational methods were developed for reconstructing common evolutionary paths of chromosomal aberrations in specific cancers. Some of these methods used tree models (5-7), later extended to acyclic networks (8-10). These evolutionary models enable recognition of aberrations that occur at early stages of cancer; often referred to as "primary", they are suspected of being cancer drivers. More recently, a statistical method named GISTIC (11) was developed for identifying copy-number aberrations whose frequency and amplitude are higher than expected. As all the methods described above were designed to analyze samples from the same cancer type, they were applied to relatively small datasets, each containing a few hundred samples.

The Mitelman database[*] (12) is the largest depository of chromosomal aberrations in cancer. Although the aberrations are described using karyotypes of low resolution, these methods are widely used, notably in hospital labs where the database is the leading source of information for clinicians who diagnose and treat cancer. The large number of samples in the database makes it ideal for statistical analyses, which are capable of overcoming random errors. In this study we present the results of large-scale analysis of chromosomal aberrations from over 15,000 karyotypes of the Mitelman database. By exploiting the huge number of karyotypes, reconstructing the aberrations in them, and developing appropriate statistical tests, we were able

[*] http://cgap.nci.nih.gov/Chromosomes/Mitelman.

to recognize significant cross-cancer associations among aberrations and to identify correlations among tumor types.

Most observed alterations include chromosome gains / losses and translocations. As translocations directly affect a small number of genes, the role of many translocations in cancer causation has become much clearer over the years (13). Chromosome gains and losses, on the other hand, are broad alterations affecting numerous genes whose significance to the carcinogenesis process is much less understood. In this study we demonstrate strong associations involving chromosome gain and loss aberrations, suggesting selection preferences for aneuploid cells.

The results of our analysis, mainly the computed associations, are publicly available via our website for further investigation.

RESULTS

Figure 1 summarizes our karyotype analysis. Starting from 59,579 karyotypes in the Mitelman database (November 2009 version), we used only 34,107 karyotypes that were annotated as unselected in order to avoid over- or under-estimation of aberration frequencies due to biases in sample selection (14). We then filtered out any partially characterized or possibly redundant karyotypes, as well as karyotypes that were not near diploid. Tumor classes were defined according to tissue morphology and organ. Karyotypes belonging to classes with small representation (<50 karyotypes) in the remaining dataset were omitted from analysis, resulting in a total of 62 classes and 15,495 karyotypes (Table 1).

Each class was assigned to one of four sets: lymphoid disorders, non-lymphoid hematological disorders, benign solid tumors, and malignant solid tumors (Table 1). Due to its higher rate of successful karyotypic analyses, the group of hematological disorders dominated our dataset, with 11,324 (73%) karyotypes, of which 6,913 (45%) belong to non-lymphoid hematological disorders. We computed for each karyotype a set of most likely aberrations involved in its formation using 11 types of chromosomal rearrangement, deletion, and duplication events (Methods, supporting information (SI) Table S1). Of those events, chromosome gain / loss and translocation were most frequent (Fig. S1). An aberration was identified by its causing event and the chromosomal locations it involved. For example, the translocation involving bands 9q34 and 22q11 was identified by t(9;22)(q34;q11), following the ISCN terminology (15)

Cancer similarity by observed aberrations

The karyotypes in our dataset contained 5,179 distinct aberrations, including all possible chromosome gains and losses. We computed the significance of the correlation of each aberration-class pair using the hypergeometric test. Out of 9,208 distinct observed aberration-class pairs, 1705 were found to be significantly correlated at false discovery rate (FDR) of 5% (website). These correlations encompassed all 62 tumor classes in our dataset, involving 1,360 distinct aberrations, where more than half of these correlations (907, 53%) involved translocations. Many of these strong correlations, notably the ones involving translocations, have been well documented in the literature: for example, t(9;22) in chronic myelogenous leukemia (16) and t(11;22) in Ewing sarcoma (17). This supports the use of our dataset as a valid sample of karyotypes from the considered classes, as well as the soundness of our results.

Which tumor classes have highly similar aberrations? Using the set of significant (FDR 5%) aberration-class correlations, we assessed the statistical significance of the overlap in aberrations for every pair of tumor classes. Of all 1891 possible class pairs, 56 pairs were found to significantly share common aberrations at an FDR of 5% (Fig. S2a). Considering benign and malignant solid tumors as one category, all but three (53, 95%) of these pairs belong to the same category, with two of the three exceptions linking between lymphoid disorders and (malignant) solid tumors. We repeated the analysis, expanding the set of correlative aberrations by considering also weaker correlations with (uncorrected) P-value <0.05. The results show a remarkably similar partition, with 86 significant class pairs (FDR 5%), forming three distinct clusters, with only six links between the sets of lymphoid disorders and solid tumors (Fig S1b). The fact that the categories were very well separated serves as confirmation of the data and of our methodology.

For more in-depth study of similarity among classes, we defined a similarity measure between classes based on the significance of their common aberrations (Methods) and used it to hierarchically cluster the classes (Fig. 2). As before, classes of the three sets – non-lymphoid-hematological disorders, lymphoid disorders and solid tumors – clustered separately. A deeper look into each cluster (Fig. 2) revealed that many closely clustered classes were histologically related. For example: diffuse large B-cell lymphoma, follicular lymphoma, and mature B-cell neoplasm (B-cell lymphomas); adenoma and adenocarcinoma in the large intestine; and AML M5 and AML M5a. The correlated aberrations shared by two similar classes can be viewed through our website. One of the interesting results was the close proximity of three embryonic cancers: Wilms' tumor (kidney), Ewing sarcoma (skeleton) and Hepatoblastoma (liver).

Significant co-occurrence of aberrations

Many of the specific associations we found between chromosomal aberrations and tumor classes are known, and serve here primarily as confirmation of the validity of our approach. We now address a question that can be answered only by more complex analysis of a large database: which aberration pairs tend to co-occur significantly more than expected by chance? Such associations may reveal either cooperation between different oncogenic events or common mechanisms creating chromosomal aberrations. To answer this question we tested the significance of co-occurrence for 7,202 aberration pairs in our dataset that satisfied the following two conditions: each aberration appeared in at least 10 karyotypes, and the pair appeared together in at least one karyotype. We first filtered pairs with hypergeometric P-value >0.001, leaving 623 pairs whose significance was further evaluated by a permutation test. Our analysis yielded 218 significantly co-occurring aberration pairs ($P<0.05$, after Bonferroni correction), of which 154 (71%) were chromosome gain pairs, and 47 (22%) were chromosome loss pairs. The induced network split clearly into two disjoint parts: one dominated by chromosome gains and one by chromosome losses (Fig. 3a). We carried out the same analysis separately for lymphoid disorders, non-lymphoid hematological disorders, solid tumors, and carcinomas (Fig. S3-S6). Each of these groups showed the same clear strong co-occurrence of specific gain-gain and loss-loss pairs, with almost no cases of significant co-occurrence for any mixed gain-loss pairs. We also detected the trisomy of 1q (18), which appeared in all tumor categories in the associations involving gain of chromosome 1 (Fig. 3a, Fig. S3-S6).

Comparative genomic hybridization (CGH) is a laboratory method to measure gains and losses in the copy number of chromosomal regions in tumor cells. To verify our findings, we analyzed an independent dataset of 1084 samples obtained by CGH, downloaded from the NCI and NCBI's SKY/M-FISH and CGH database (March 16, 2009 version). This database contains CGH records contributed by molecular cytogeneticists for open investigation. Each sample was assigned a corresponding set of whole chromosome gain/loss aberrations, yielding 648 (60%) samples with non-empty aberration sets. Using a permutation test similar to the one used for karyotypes data (Methods), we computed a P-value for the co-occurrences of specific aberration pairs in the CGH dataset. Out of 856 distinct co-occurring aberrations pairs, 47 were significantly co-occurring at FDR of 5%. The picture obtained by these pairs (Fig. 3b) is strikingly similar to the one produced by the karyotype data. This reaffirms our observation that the progression of aneuploidy in cancer is driven by either multiple chromosomal gains or multiple chromosomal losses.

The website

All the associations described above can be viewed via the website http://acgt.cs.tau.ac.il/stack/, which contains summary tables for the different types of associations: aberration-class, class-class, and aberration-aberrations. Table rows can be filtered textually and numerically, allowing investigations of associations for a specific group of cancer types, a set of aberrations of interest, or both. For example, the user can view all aberrations whose correlation with a certain tumor class is below some specified P-value. Alternatively, all aberrations significantly co-occurring with a specified aberration can be examined, with their P-values. For aberration-class and aberration-aberration associations, researchers can examine the karyotypes that led to these associations, where each karyotype is linked to its corresponding record in the Mitelman database website.

To demonstrate the utility of the website, we focused on hyperdiploid multiple myeloma (H-MM), a subtype of multiple myeloma (MM) with better prognosis, characterized by having 48-74 chromosomes (19-21). There were 385 MM karyotypes in the database, and 110 (29%) of which were hyperdiploid. H-MM is associated with recurrent gains of chromosomes 3, 5, 7, 9, 11, 15 and 19 (19). Indeed, the website's class-aberration table, filtered for MM associations, confirmed this observation: +3, +5, +9, +11, +15, and +19 were the aberrations most associated with MM, and the 142 karyotypes involved in these associations spanned all H-MM karyotypes (hyper-geometric $P < 1E-76$). Chng et al. (22) suggested a FISH-based trisomy index for identifying H-MM, employing probes for chromosomes 9, 11 and 15, and designating a tested MM cell as H-MM if it contains two or more trisomies in these chromosomes. They reported specificity of 0.98 and sensitivity of 0.69 for that index. The corresponding F-Score (a measure combining sensitivity and specificity, see Methods) was 0.8. We analyzed the 385 MM karyotypes in the same fashion as (22); the criterion of any two trisomies in 9, 15, 19 was best with specificity 0.996 and sensitivity 0.88 [F-Score 0.93]. In fact, the same combination has the highest F-Score on the data of (22) as well (0.83). Thus, the criterion of two or more trisomies of chromosomes 9, 15, 19 should be considered for identifying H-MM.

DISCUSSION

In this study we computationally analyzed a large number of cancer karyotypes from the Mitelman database, the largest available compendium of cancer karyotypes. Based on statistical analysis of more than 15,000 karyotypes, our results provide strong additional evidence for the non-randomness of many chromosomal aberrations in cancer. Our approach is validated by the

demonstration of known relationships, including associations between specific aberrations and specific tumor types, and similarities among certain tumors (e.g. adenoma and adenocarcinoma of the large intestines). More importantly, the analysis led to new discoveries, most notably that chromosomal aneuploidy tends to consist of either a pattern of chromosomal gains or a pattern of chromosomal losses. This novel discovery was verified by similar analysis of a separate molecular database.

To avoid ambiguities and reduce potential biases in the results, we excluded from our dataset karyotypes that were not random samples (i.e., reported because of a specific/unusual karyotypic feature), and those with missing information. Inclusion of partially-characterized karyotypes (omitting non-characterized fragments) increased the number of karyotypes to 22,425 (45% increase). The results on that set closely matched those reported here (Fig. S7, S8), indicating the robustness of both the results and our statistical methods.

Chromosome gains/losses and translocations were the most abundant aberrations in our dataset. While many translocations were shown to contribute to carcinogenesis, the role of chromosomal aneuploidy in cancer has been debated for almost a century. We report for the first time a striking dichotomy of aneuploidy across numerous tumor classes, discovered in an analysis of two independent datasets: significantly co-occurring aberration pairs are almost exclusively either both chromosome gains or both chromosome losses. A similar tendency was observed by Höglund et al. (9) for several specific solid cancers. The karyotypic evolution models of (9) contained two converging paths, one dominated by gains of chromosomal fragments and the other by losses.

The observed chromosome gain/loss dichotomy suggests a partial explanation for the following conundrum: A single chromosome gain/loss in the germline is usually hazardous, both at the cellular and the organism levels, while the abundance of chromosome gains/losses in cancer cells implies that aneuploidy is beneficial, or at least not harmful, to their vitality (23-26). As most chromosomes contain dosage-sensitive genes, the strong gain-gain and loss-loss correlations may imply a mechanism for balancing the ratios of proteins that function in complexes. Such balancing may be required to protect the cancer cell from the detrimental effects of partially assembled protein complexes or free subunits by molecular chaperones caused by prior chromosome gain / loss events. This novel hypothesis is testable by large-scale quantitative proteomics. An alternative explanation for these observations is that chromosomal gains and losses are caused by different mechanisms of genomic instability.

One limitation of the use of the Mitelman database is its inherent bias towards hematological cancers. However, the number of solid karyotypes in the database is still substantial, and allowed us to obtain results on class similarity among solid cancers (Fig. 2). Moreover, the results on aberration co-occurrence tendency were similar using the full data (Fig. 3) and the solid karyotypes only (Fig. S5).

The methodologies developed in this study can be used on other large datasets describing genetic events. As high resolution genetic information on tumors accumulates, similar analysis can be applied to it – using for instance Next-Generation Sequencing. Moreover, our website can be useful both for additional global investigations like those reported here and for in-depth analysis of individual associations.

MATERIALS AND METHODS

Karyotypes selection and analysis. We evaluated all 34,107 karyotypes marked as unselected (i.e. chosen in a non-biased manner) in the Mitelman database on November 17, 2009. Karyotypes were parsed using the CyDAS ISCN parser (27), and any karyotype detected as invalid during the parsing was excluded, leaving 29,911 (88%) valid karyotypes. We refer to a karyotype as *well-defined* if it is complete and does not contain any of the following: 1) double minutes, 2) marker chromosomes, 3) ring chromosomes, 4) chromosomes with homogeneous staining regions (HSRs), 5) chromosomes with additional material of unknown origin, 6) approximated breakpoints, e.g. del(1)(q21~q24), or 7) alternative interpretations of an aberration (designated by "or" symbol). Question marks (?) indicating questionable identification of a chromosome or chromosome structure (e.g. del(1)(q?23)) were ignored. We refer to a karyotype as *multiclonal* if it is composed of several distinct karyotypes (separated by a dash "/" representing different subclones in the sample). Given a multiclonal karyotype, we avoided dependency between its karyotypes by choosing only the first well-defined karyotype it contained. In case of multiple karyotypes from the same patient ("case" in the Mitelman database), only one karyotype was taken into account. To avoid potential biases in chromosome gain/loss aberrations, we excluded any karyotype that was not near-diploid (i.e., we omitted karyotypes whose total chromosome number was less than 35 or more than 57). Altogether, 18,813 karyotypes were selected for analysis.

Aberrations reconstruction. We previously identified 11 frequent chromosomal events in tumor karyotypes (chromosome gain/loss, translocation, deletion, duplication and more, see Table S1), and developed an algorithm for reconstructing a most plausible set of events leading

to a given karyotype (28). We applied the algorithm to all relevant karyotypes from the Mitelman database, obtaining unambiguous reconstruction in 99% (18,600) of the karyotypes. We recorded each such karyotype's set of aberrations, where an aberration is defined by an event and the chromosomal locations involved. For example, +1 is the aberration resulting from a chromosome gain event on chromosome 1, and t(9;22)(q34;q11) is a translocation involving bands q34 and q11 on chromosomes 9 and 22, respectively.

Karyotypes classification. We classified karyotypes by their tissue morphology and topography as specified in the Mitelman database. To permit robust statistical analysis, we omitted all karyotypes whose class had less than 50 karyotypes. Our final dataset contained 15,445 karyotypes.

CGH data. We used the NCBI's SKY/M-FISH and CGH database[†] (version March 16, 2009), consisting of 1084 records. Every record has a list of chromosomal segments with abnormal copy number, each classified as a gain or a loss; and the header of the record contains information on the cancer tissue. As most tumor classes in this dataset were relatively small, we ignored the histological classification. For each record we derived chromosome gain / loss aberrations in the following manner: every gained (lost) chromosomal fragment that spanned the centromere was considered a whole chromosome gain (loss).

Computing P-values for aberration-class correlations. For an aberration *Ab* and a class *C*, we calculated the significance of the enrichment of karyotypes with *Ab* in *C* using the hypergeometric test.

Computing P-values for classes sharing common aberrations. We developed the following method for evaluating the significance of shared aberrations between tumor classes. We constructed a binary matrix M_t whose rows and columns correspond to aberrations and classes, respectively. We set $M_t[Ab,C]=1$ if the correlation between aberration Ab and class C had a hypergeometric P-value $\leq t$ (in that case we say that Ab is t-correlative to C), and otherwise $M_t[Ab,C]=0$. For t=0.05, the maximal t used in our analysis, the matrix M_t was already quite sparse, less than 2% 1's. For two classes, C and C', we computed a P-value for their number of shared events as follows. Let $n_{t,C,C'}$ be the number of t-correlative aberrations that C and C' shared. More formally, $n_{t,C,C'} = \Sigma_{Ab} M_t[Ab,C] \cdot M_t[Ab,C']$. For every pair of classes, C and C', that shared at least one t-correlative aberration, we estimated the probability of having at least

[†] http://www.ncbi.nlm.nih.gov/sky/skyweb.cgi.

$n_{t,C,\ C'}$ t-correlative aberrations by chance when the marginal distributions of the rows (aberrations) and columns (classes) of M_t are fixed. We did this by randomly sampling $N=10^7$ permutations of M_t that preserve row and column sums. Therefore, the minimal P-value we could achieve was lower bounded by $1/N = 10^{-7}$.

Hierarchical clustering of classes. We performed average-linkage hierarchical clustering of the classes using the Expander software package (29). The similarity measure between classes was defined as follows. We first built a symmetric matrix, S, satisfying $S[C_1,C_2] = -\log(p)$, where p is the P-value described above for the significance of the number of t-correlative aberrations that C_1 and C_2 share. For each class C, we set $S[C,C]=\log(N)$, where $N=10^7$ as above. The similarity between classes was now defined as the Pearson correlation between their rows of S.

Computing P-values for co-occurring aberration pairs. Let Ω denote the entire dataset of karyotypes. For two aberrations, *Ab* and *Ab'*, let n(Ab, Ab') be the number of karyotypes in Ω that contain both aberrations. We estimated the significance of n(Ab, Ab') for all pairs of distinct aberrations using a permutation test as follows. We constructed a binary matrix, M', whose rows correspond to aberrations that occur in at least 10 karyotypes, and columns to the karyotypes in Ω. Aberrations that did not co-occur with any other aberration in M were excluded. For an aberration Ab and karyotype K, we set M'[Ab,K]=1 if K contained Ab, and M'[Ab,K]=0 otherwise. We randomly sampled permutations of M' that preserved row and column sums. Moreover, to account for the different distributions of aberrations within each tumor class, the sampled permutations were also required to preserve (sub-)row sum for each class. We enhanced the performance of this test by filtering aberration pairs whose hypergeometric test P-value was above 0.001, and removing from M' any aberration that did not appear in the remaining pairs.
We performed a similar test for the CGH dataset, but since it was smaller in size we used all aberrations (i.e. irrespective of the number of samples in which they were found), and without the step of filtering pairs by the hypergeometric test.

Trisomy index test. Sensitivity (respectively, specificity) was calculated as the percentage of H-MM (respectively, non-H-MM) karyotypes that are correctly identified as such by the trisomy index test (TTI). The positive predictive value (PPV) was calculated as the percentage of H-MM karyotypes among all karyotypes identified as H-MM by TTI. The F-score was calculated as the harmonic mean of sensitivity and PPV: $F = 2\times PPV\times sensitivity/(PPV+sensitivity)$.

URLs. More details on our results can be found on our website (http://acgt.cs.tau.ac.il/stack). Supporting information is found on http://acgt.cs.tau.ac.il/stack/suppI.

Acknowledgements. We thank Gideon Rechavi, Avi Orr-Urtreger, and Uta Francke for helpful discussions. We are grateful to Lior Mechlovich for programming an early version of the analysis code and to Igor Ulitsky for help with the hierarchical clustering code. RS was supported in part by the Raymond and Beverly Sackler Chair in Bioinformatics and by the Israel Science Foundation (Grant 802/08). SI was supported by the Israel Science Foundation (Morasha program).

Author contribution. R.S. and M.O-F. designed research. M.O-F performed research and built the website. C.L. and M.O-F. developed the statistical scores. M.O-F., R.S., S.I. and L.T. analyzed and interpreted the data. M.O-F., R.S. and S.I. wrote the paper.

References

1. Bayani J, *et al.* (2007) Genomic mechanisms and measurement of structural and numerical instability in cancer cells. *Semin Cancer Biol* 17(1):5-18.
2. Haber DA, Settleman J (2007) Cancer: drivers and passengers. *Nature* 446(7132):145-146.
3. Vogelstein B, *et al.* (1988) Genetic alterations during colorectal-tumor development. *N Engl J Med* 319(9):525-532.
4. Fearon ER, Vogelstein B (1990) A genetic model for colorectal tumorigenesis. *Cell* 61(5):759-767.
5. Desper R, *et al.* (1999) Inferring tree models for oncogenesis from comparative genome hybridization data. *J Comput Biol* 6(1):37-51.
6. Desper R, *et al.* (2000) Distance-based reconstruction of tree models for oncogenesis. *J Comput Biol* 7(6):789-803.
7. von Heydebreck A, Gunawan B, Fuzesi L (2004) Maximum likelihood estimation of oncogenetic tree models. *Biostatistics* 5(4):545-556.
8. Radmacher MD, *et al.* (2001) Graph models of oncogenesis with an application to melanoma. *J Theor Biol* 212(4):535-548.
9. Hoglund M, Frigyesi A, Sall T, Gisselsson D, Mitelman F (2005) Statistical behavior of complex cancer karyotypes. *Genes Chromosomes Cancer* 42(4):327-341.
10. Hjelm M, Hoglund M, Lagergren J (2006) New probabilistic network models and algorithms for oncogenesis. *J Comput Biol* 13(4):853-865.
11. Beroukhim R, *et al.* (2007) Assessing the significance of chromosomal aberrations in cancer: methodology and application to glioma. *Proc Natl Acad Sci U S A* 104(50):20007-20012.
12. Mitelman F, Johansson B, Mertens F (2009) Mitelman Database of Chromosome Aberrations in Cancer.
13. Mitelman F, Johansson B, Mertens F (2007) The impact of translocations and gene fusions on cancer causation. *Nat Rev Cancer* 7(4):233-245.

14. Mitelman F, Mertens F, Johansson B (2005) Prevalence estimates of recurrent balanced cytogenetic aberrations and gene fusions in unselected patients with neoplastic disorders. *Genes Chromosomes Cancer* 43(4):350-366.
15. Shaffer L, Tommerup N (2005) *ISCN 2005: an international system for human cytogenetic nomenclature (2005): recommendations of the International Standing Committee on Human Cytogenetic Nomenclature* (S Karger Pub).
16. Nowell P, Hungerford D (1960) A minute chromosome in chronic granulocytic leukemia. *Science* 132:1497.
17. Turc-Carel C*, et al.* (1988) Chromosomes in Ewing's sarcoma. I. An evaluation of 85 cases of remarkable consistency of t(11;22)(q24;q12). *Cancer Genet Cytogenet* 32(2):229-238.
18. Ghose T*, et al.* (1990) Role of 1q Trisomy in Tumorigenicity, Growth, and Metastasis of Human Leukemic B-Cell Clones in Nude Mice. *Cancer Res* 50(12):3737-3742.
19. Smadja NV*, et al.* (1998) Chromosomal analysis in multiple myeloma: cytogenetic evidence of two different diseases. *Leukemia* 12(6):960-969.
20. Smadja NV*, et al.* (2001) Hypodiploidy is a major prognostic factor in multiple myeloma. *Blood* 98(7):2229-2238.
21. Fonseca R*, et al.* (2004) Genetics and cytogenetics of multiple myeloma: a workshop report. *Cancer Res* 64(4):1546-1558.
22. Chng WJ*, et al.* (2005) A validated FISH trisomy index demonstrates the hyperdiploid and nonhyperdiploid dichotomy in MGUS. *Blood* 106(6):2156-2161.
23. Ganmore I, Smooha G, Izraeli S (2009) Constitutional aneuploidy and cancer predisposition. *Hum Mol Genet* 18(R1):R84-93.
24. Williams BR*, et al.* (2008) Aneuploidy affects proliferation and spontaneous immortalization in mammalian cells. *Science* 322(5902):703-709.
25. Weaver BA, Silk AD, Montagna C, Verdier-Pinard P, Cleveland DW (2007) Aneuploidy acts both oncogenically and as a tumor suppressor. *Cancer Cell* 11(1):25-36.
26. Roper RJ, Reeves RH (2006) Understanding the basis for Down syndrome phenotypes. *PLoS Genet* 2(3):e50.
27. Hiller B, Bradtke J, Balz H, Rieder H (2005) CyDAS: a cytogenetic data analysis system. *Bioinformatics* 21(7):1282-1283.
28. Ozery-Flato M, Shamir R (2007) On the frequency of genome rearrangement events in cancer karyotypes. (Tel Aviv University).
29. Shamir R*, et al.* (2005) EXPANDER--an integrative program suite for microarray data analysis. *BMC Bioinformatics* 6:232.
30. Shannon P*, et al.* (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. (Cold Spring Harbor Laboratory Press), pp 2498-2504.

Figure 1: Overview of karyotypes analysis and the STACK website. A large fraction of the karyotypes in the Mitelman database was removed to avoid potential bias in the analysis. These included partially characterized karyotypes, multiple karyotypes from the same individual, and karyotypes that were not randomly selected in the original report. Tumor type and location were used to classify karyotypes into tumor classes, and classes with small representation (< 50 karyotypes) were removed from the dataset. An algorithm was used to reconstruct the set of aberrations leading to each remaining karyotype. Three types of statistical correlations were computed: aberration co-occurrence, association between class and aberration, and class similarity (based on their common aberrations). All computed correlations, with their P-values, are available for further investigation via our website and are directly linked to the full description of the relevant karyotypes in the Mitelman database. Repeating the analysis without filtering ambiguities (yielding 22,425 karyotypes) led to essentially the same conclusions.

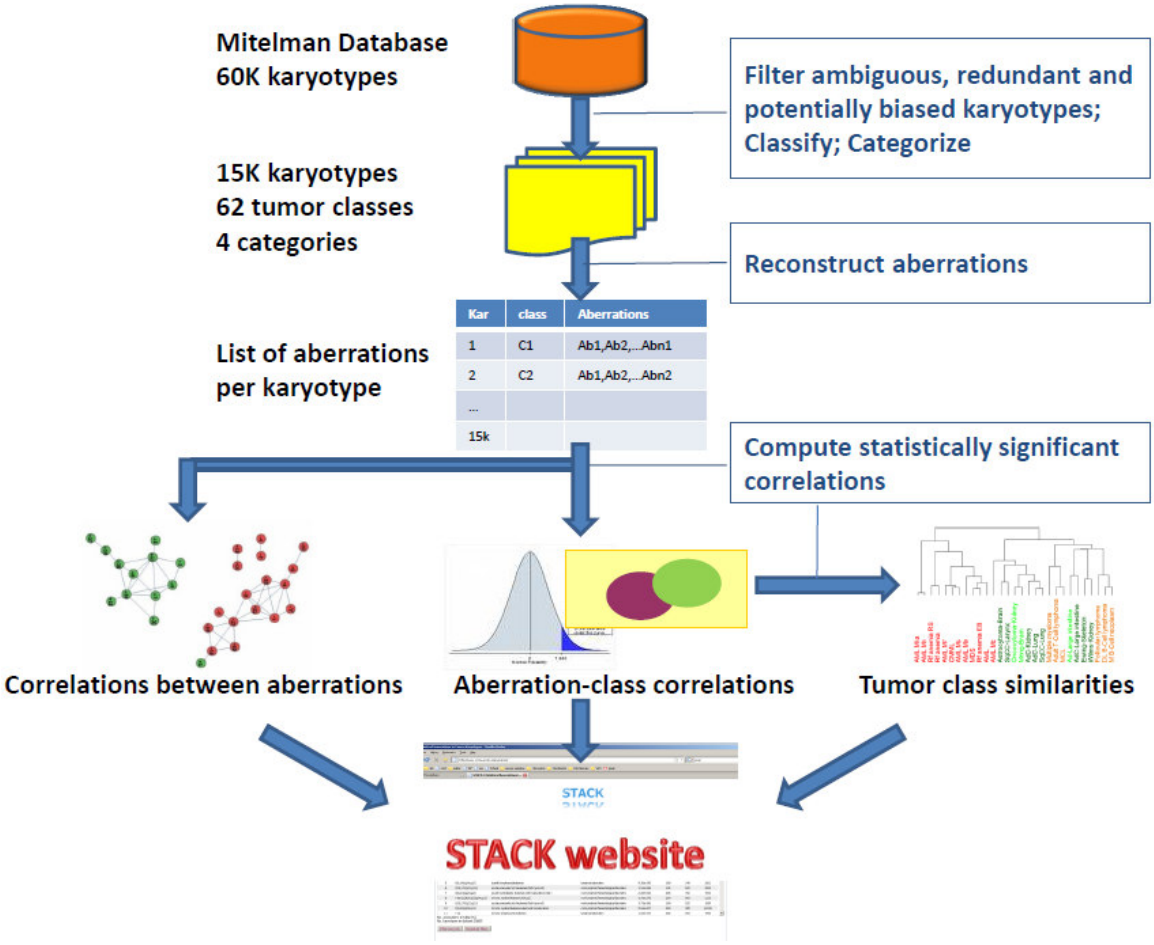


Figure 2: Hierarchical clustering of classes based on class similarity in sharing common aberrations. The square at the intersection of each two diagonals shows the similarity of their classes, as measured by the aberrations associated with them (Methods). (An aberration was associated with a tumor class if their correlation had (uncorrected) P-value < 0.05.) Names of cancer classes are colored as follows: orange: lymphoid disorders; red: non-lymphoid hematological disorders; light green: benign solid tumors; dark green: malignant solid tumors.

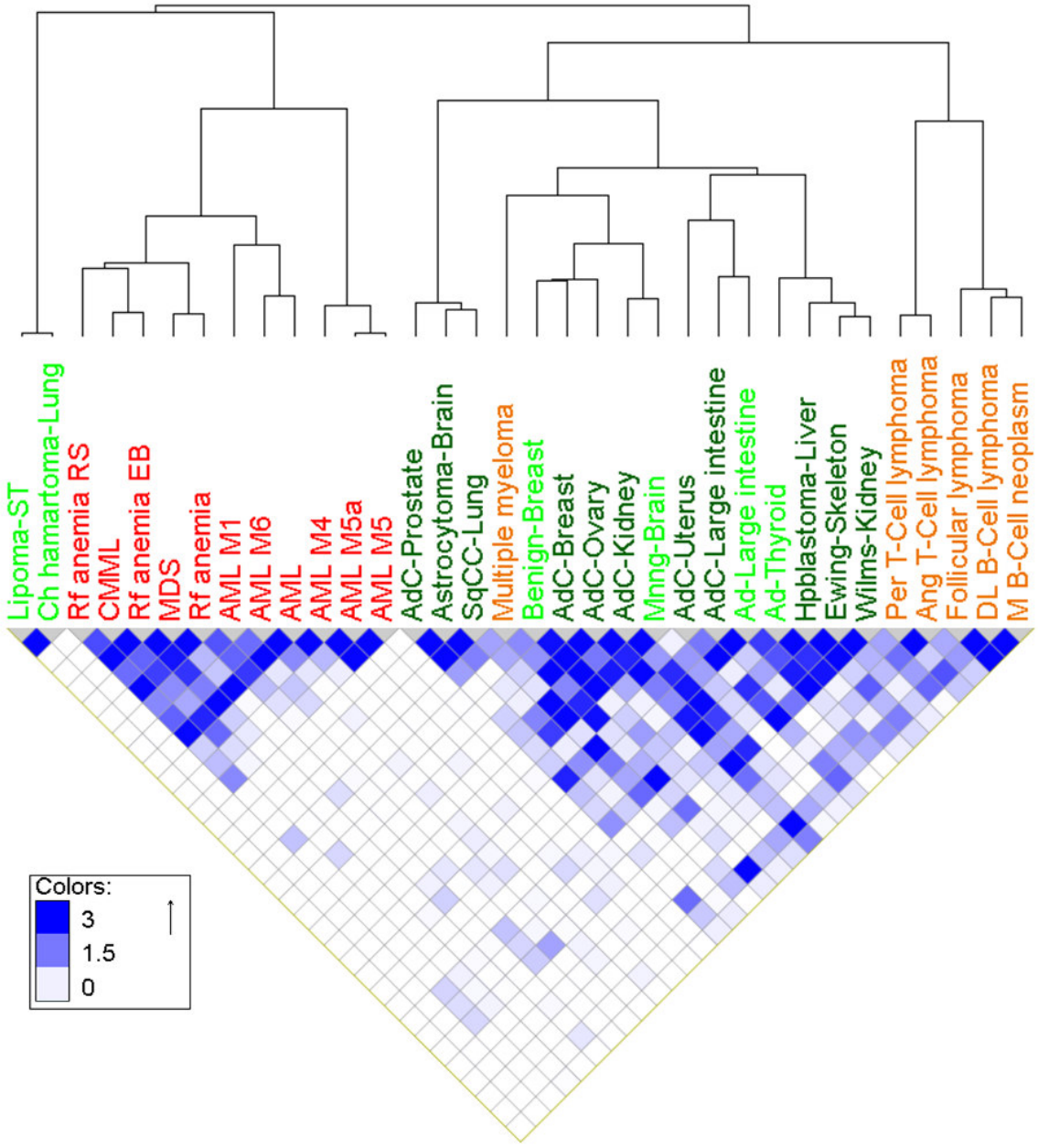


Figure 3: Highly co-occurring aberration pairs. Highly co-occurring aberrations in the entire karyotype dataset are connected by lines. Aberrations that are involved only in expected links (e.g. a link between a translocation and a gain /loss of one of its derivative chromosomes; a link between two (two-break) translocations originating from one three-break (15) rearrangement) are not shown. For explanation on aberration names, see Table S1. (a) Highly co-occurring pairs in the Mitelman Database karyotypes (links are significant at P<0.05, after Bonferroni correction). (b) Highly co-occurring pairs in the CGH dataset (links are significant at FDR 5%). The only gain-loss link is (+1, -16), which has the second worst (i.e. highest) P-value among the 47 pairs that passed the FDR 5% criterion. The figure was drawn using Cytoscape (30).

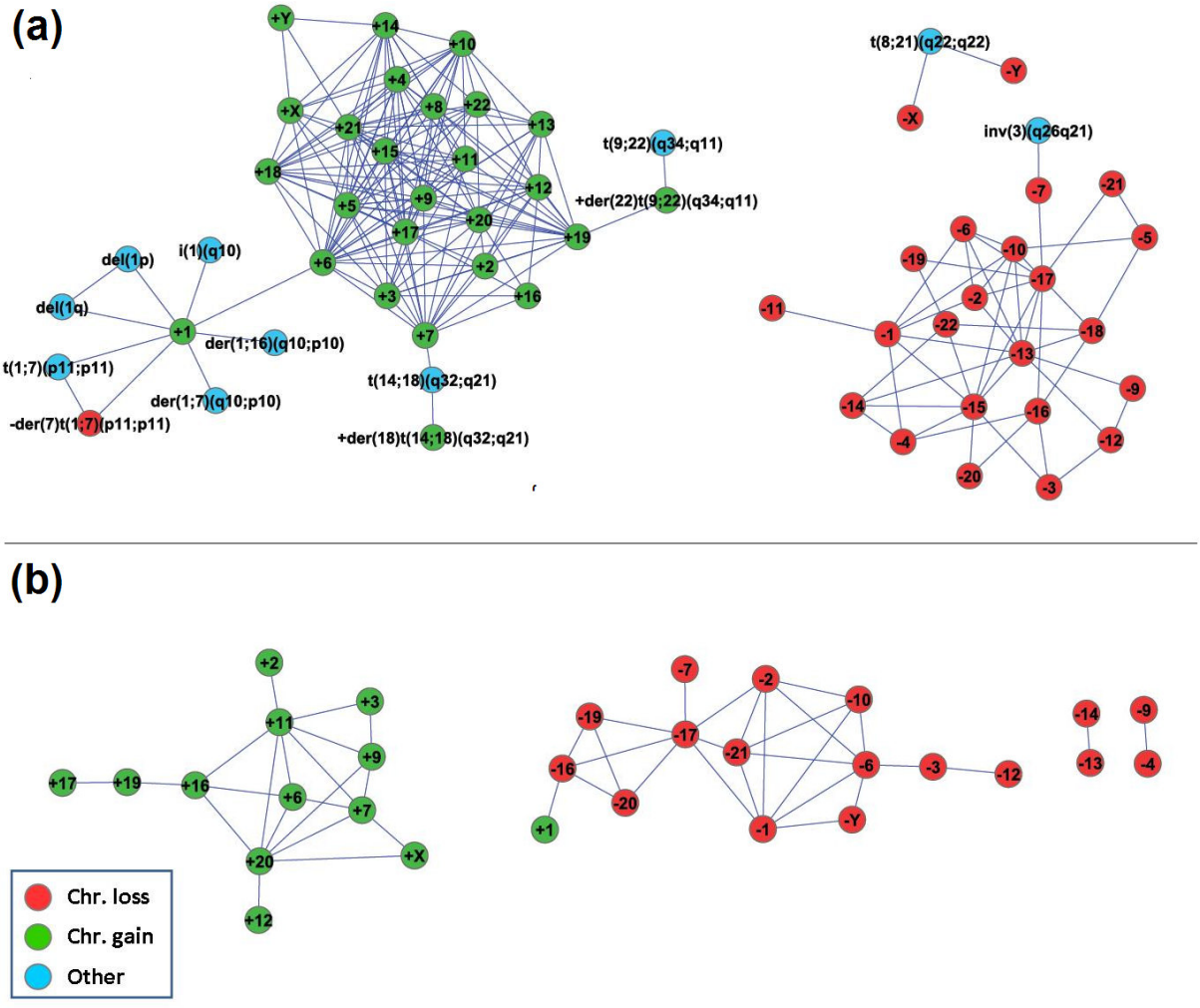


Table 1: Tumor classes and categories in the dataset. The table contains tumor classes used in our study, arranged by categories. The Details column contains class description as given in the Mitelman database.

Class	Details	No. of classes
benign solid tumors		**1567**
Ad-Large intestine	Adenoma-Large intestine	100
Ad-Salivary gland	Adenoma-Salivary gland	191
Ad-Thyroid	Adenoma-Thyroid	66
Benign-Breast	Benign epithelial tumor special type-Breast	69
Ch hamartoma-Lung	Chondroid hamartoma-Lung	99
Leiomyoma-Uterus	Leiomyoma-Uterus corpus	214
Lipoma-ST	Lipoma-Soft tissue	269
Mnng-Brain	Meningioma-Brain	508
Oncocytoma-Kidney		51
non-lymphoid hematological disorders		**6913**
AML	Acute myeloid leukemia NOS	1026
AML M0	Acute myeloblastic leukemia with minimal differentiation (FAB type M0)	144
AML M1	Acute myeloblastic leukemia without maturation (FAB type M1)	315
AML M2	Acute myeloblastic leukemia with maturation (FAB type M2)	776
AML M3	Acute promyelocytic leukemia (FAB type M3)	525
AML M4	Acute myelomonocytic leukemia (FAB type M4)	621
AML M5	Acute monoblastic leukemia (FAB type M5)	266
AML M5a	Acute monoblastic leukemia without differentiation (FAB type M5a)	52
AML M6	Acute erythroleukemia (FAB type M6)	133
AML M7	Acute megakaryoblastic leukemia (FAB type M7)	168
BBL	Bilineage or biphenotypic leukemia	137
CMD	Chronic myeloproliferative disorder NOS	69
CML at	Chronic myeloid leukemia aberrant translocation	409
CML t(9;22)	Chronic myeloid leukemia t(9;22)	808
CMML	Chronic myelomonocytic leukemia	147
Id myelofibrosis	Idiopathic myelofibrosis	115
JML	Juvenile myelomonocytic leukemia	50
MDS	Myelodysplastic syndrome NOS	187
Polycythemia Vera	Polycythemia vera	166
Rf anemia	Refractory anemia	374
Rf anemia EB	Refractory anemia with excess of blasts (FAB)	344
Rf anemia RS	Refractory anemia with ringed sideroblasts	81
lymphoid disorders		**4411**
ALL	Acute lymphoblastic leukemia/lymphoblastic lymphoma	1817
Adult T-Cell lymphoma	Adult T-cell lymphoma/leukemia (HTLV-1+)	64
Ang T-Cell lymphoma	Angioimmunoblastic T-cell lymphoma	71

Burkitt lymphoma	Burkitt lymphoma/leukemia	248
CLL	Chronic lymphocytic leukemia	884
DL B-Cell lymphoma	Diffuse large B-cell lymphoma	197
Follicular lymphoma		274
HCL	Hairy cell leukemia	57
M B-Cell neoplasm	Mature B-cell neoplasm NOS	166
MCL	Mantle cell lymphoma	78
Multiple myeloma		385
Per T-Cell lymphoma	Peripheral T-cell lymphoma unspecified	62
SMZ B-Cell lymphoma	Splenic marginal zone B-cell lymphoma	108
malignant solid tumors		**2554**
AdC-Breast	Adenocarcinoma-Breast	323
AdC-Kidney	Adenocarcinoma-Kidney	610
AdC-Large intestine	Adenocarcinoma-Large intestine	125
AdC-Ovary	Adenocarcinoma-Ovary	56
AdC-Prostate	Adenocarcinoma-Prostate	124
AdC-Thyroid	Adenocarcinoma-Thyroid	84
AdC-Uterus	Adenocarcinoma-Uterus corpus	62
Astrocytoma-Brain	Astrocytoma grade III-IV-Brain	234
BCC-Skin	Basal cell carcinoma-Skin	87
Ewing-Skeleton	Ewing tumor/peripheral primitive neuroectodermal tumor-Skeleton	181
Giant cell-Skeleton	Giant cell tumor of the bome-Skeleton	60
Hpblastoma-Liver	Hepatoblastoma-Liver	65
Liposarcoma M-ST	Liposarcoma myxoid/round cell-Soft tissue	59
Melanoma-Eye	Malignant melanoma-Eye	72
SqCC-Larynx	Squamous cell carcinoma-Larynx	58
SqCC-Lung	Squamous cell carcinoma-Lung	64
Synovial sarcoma-ST	Synovial sarcoma-Soft tissue	58
Wilms-Kidney	Wilms tumor-Kidney	232

Chapter 8

Discussion

In this thesis we described our study on genome rearrangements occurring in the evolution of species and in cancer cells. Considering different models for evolution and cancer, we focused on finding a shortest sequence of rearrangement events explaining large-scale differences between two genomes (Chapters 2-5). We built on extant mathematical theory and generalized it (Chapter 2-4). We presented a new set of simpler and more efficient algorithms for a previously analyzed model (Chapters 2,3). We extended this model by adding new biological constraints and presented an accurate polynomial time solution for the corresponding problem (Chapter 4). We proposed an original model suited for cancer karyotypes and provided a 3-approximation polynomial time algorithm for computing a shortest sequence of rearrangements transforming a normal genome into a given cancer genome, under certain assumptions supported by most real data (Chapter 5). The last part of this thesis was dedicated to a statistical analysis of rearrangements, reconstructed by an effective heuristic, in a large public database of cancer karyotypes (Chapter 6,7). In this chapter we briefly review the results introduced in this thesis and discuss their importance and relevance to other works. In addition, we raise open problems that stem from the analysis and from the results in this thesis.

8.1 Sorting by Translocations

From a bird's-eye view, genomes of related species are built from essentially the same set of large (synteny) blocks of DNA. The different ordering of these blocks

in the genomes inspired the computational problem of inferring a shortest sequence of rearrangement events between related genomes. Reversals (aka inversions) and translocations are common miotic rearrangements in mammals. While translocations mix the content of two chromosomes, the effect of inversions is localized to a single chromosome. Sorting uni-chromosomal genomes by reversals (SBR) became one of the most analyzed problems in the computational study of genome rearrangements and hence there is a rich theory on it [41, 15, 49, 7, 12, 103]. The problem of sorting by translocations (SBT) was analyzed in the context of SBR by the same authors [39, 14], and was shown to share a similar combinatorial formulation with SBR. Nevertheless, the extant algorithms for solving SBT had little in common with the algorithms for solving SBR. In Chapters 2 and 3 we described a new combinatorial framework for analyzing SBT, which built on extant framework for analyzing SBR. This new framework allowed us to exploit the wealth of theory on SBR and provide analogous results for SBT. In particular, we managed to adapt three most efficient algorithms for solving SBR to solve SBT, while preserving the original time complexities. One of these new algorithms, which runs in sub-quadratic rime, is currently the fastest algorithm for solving SBT. Testing whether the latest improvement in the time complexity of SBR, achieved by Swenson et al. [101], can be applied to SBT remains as a task for future work.

By developing a combinatorial representation of SBT akin to the extant one for SBR, we revealed novel similarities between the two problems. Moreover, this implied that the problem of sorting by reversals and translocations (SBRT) can be analyzed in a similar manner, without having to reduce it to SBR as the current algorithm does [40, 105, 68]. Despite the common properties we revealed for SBR and SBT, we did not prove an equivalence between the problems, nor did we prove that one is reducible to the other. Proving whether there exists such stronger relation between the two problems remains an open problem.

Reversals and translocations are two special cases of the *double-cut-and-join* (DCJ) operations introduced by Yancopoulos et al. [114]. The DCJ operation is equivalent to the *2-break* operation studied by Alekseyev and Pevzner [5]. The distance formula and the algorithms for sorting by DCJs (SDCJ) were shown to be much simpler, in comparison with SBR, SBT, and SBRT [114, 13, 5]. The major reason for the relative simplicity of SDCJ is its powerful ability to create intermediate circular chromosomes, which are later reabsorbed. This ability facilitates an elegant bypass to the difficulty of avoiding the creation of "bad components", which is

the source of complication for SBR, SBT, and SBRT. We note that the version of SDCJ in which circular chromosome creation is not allowed is equivalent to SBRT. An intriguing question if whether there exists an alternative generalization of the overlap graph in which all DCJ operations are modeled on the same manner.

8.2 Sorting by Translocations with Centromeres

In this thesis we made the first attempt to take into account centromeres in rearrangement scenarios (Chapter 4). As no mapping ("ortholog assignment") is given between centromeres of related genomes, we treated all centromeres as equivalent anonymous elements whose location is the only information given for them. As a chromosome must have a centromere in order to survive the subsequent cell divisions, we regarded translocations creating acentric chromosomes as illegal and forbade their use. We studied the problem of sorting by legal translocations (SBLT) and provided an accurate polynomial-time solution for it using a reduction to SBT that mapped the centromeres in the two genomes.

Using our definition for legality, exactly half of all possible translocations are illegal. In contrast, every reversal is legal, as reversals do not alter the number of centromeres in a chromosome. Allowing for legal translocations only, as we did, imposed an additional constraint on the signs of the genes in the input genomes (Observation 1, Chapter 4). We note that disallowing reversals and considering (legal) translocations only, severely limits the practicality of our algorithm in analyzing real data. Extending SBLT to allow for reversals eliminates this "artificial" constraint and results in a new interesting open problem, which is also biologically more reasonable. Another research direction is to extend SDCJ, which is much simpler than SBRT, to account for centromeres and legal sorting.

SBR, SBT, and SBRT, are all based on simplistic models for genome rearrangements. Apart from the constraint we considered for centromere-aware operations, there are many other biologically motivated constraints and requirements that can be integrated into these problems. These include different weights/probabilities for different rearrangement events, depending on the rearrangement type, location (e.g. considering breakpoint "hotspots"), and overall effect (e.g. length of inverted segment - for reversals [10]). Amajor difficulty in using SBR / SBT / SBRT algorithms for analyzing real genomes is the non-uniqueness of their solutions. Several ways

have been proposed to tackle this problem, such as enumerating all optimal solutions [95, 24], finding a compact representation of the solution space [23, 22], sampling the solution space [60], and reconstruction of partial "reliable" solutions [115, 116]. We believe that the addition of biologically plausible constraints on SBR / SBT / SBRT, such as our exclusion of illegal translocations, will help to reduce the number of optimal scenarios, and thus may bring us closer to the true rearrangement scenarios that took place.

Although scenarios involving acentric chromosomes are less favorable, they are not absolutely impossible. In a major discovery in 1993, it was shown that a newly formed chromosome that lacks a centromere can be rescued by the emergence of a new centromere in a seemingly random location [110]. Since this initial discovery, over ninety cases of neocentromere formation in humans have been described in the literature, among which are five cases of centromere repositioning (i.e. neocentromere formation accompanies by an inactivation of an existing centromere) [59]. This discovery supports a new model for rearrangements that considers two extra operations: forming neocentromere and inactivation of an existing centromere. Nevertheless, as translocations are far more common than neocentromere formation, and the mechanisms underlying neocentromeres are not well understood, the use of neocentromeres in sorting scenarios should be done in moderation. An interesting question is thus to find a shortest sequence of translocations requiring k centromere formations / repositioning events, where k is a parameter. Whether there exists a fixed-parameter-tractable algorithm to this parametric problem is an open problem.

8.3 Sorting Cancer Karyotypes

Cancer karyotypes display a wide variety of chromosomal aberrations caused by rearrangement events. In Chapter 5 we made a first attempt to rigorously reconstruct a sequence of plausible rearrangement events that led to a given cancer karyotype. We presented an original model of rearrangements in cancer genomes using four biologically-motivated elementary operations. We used this model to define the problem of karyotypes sorting (KS), which seeks for a shortest sequence of these elementary operations that transforms a normal karyotype into a given abnormal (cancer) karyotype. Under the simplifying assumption that no breakpoint is duplicated, which is supported by the vast majority (94%) of cancer karyotypes in the

Mitelman database, we reduced KS to a simpler variant RKS, in which no breakpoint exists. We proved lower and upper bounds for the length of a solution to RKS, which yielded a 3-approximation polynomial-time algorithm. We applied this algorithm on 58,464 karyotypes with no recurrent breakpoints. For 99.9% of those karyotypes our algorithm produced a solution that achieved the lower bound and hence was optimal. Manual inspection of the remaining cases revealed that the solutions produced by algorithm were optimal (i.e shortest) for all the remaining (30) cases as well.

The complexity of KS problem, and its reduced form, RKS, remained an open theoretical problem for future research. Another requested future extension of this work is to weaken the assumption that prohibits breakpoint duplication in a way that allows the analysis of the remaining 6% of the karyotypes, which are likely to correspond to more advanced stages of cancer. Our hope is that this study will lead to further algorithmic research on the evolution of chromosomal aberrations in cancer.

The model we proposed for the evolution of cancer karyotypes allowed for duplication and deletion events, which were shown to be most common in cancer karyotypes (Chapter 6) and hence must not be neglected. Rearrangement models that do not allow duplications and deletions, such as the ones used by SBR/SBT/SBRT, are inadequate for modeling the evolution of chromosomal aberrations in cancer. Moreover, karyotypes exhibit complex structural aberrations that are difficult to explain by mere reversals and translocations. Conversely, the consideration of breakage and fusion as two independent events, added much more power and flexibility in the generation of complex aberrations, albeit at the cost of using less conventional events. We note that most of the statistical studies of rearrangement events in cancer that we are aware of, analyze elementary events: duplications/deletions of segments (commonly CGH data), and breakages (commonly referred as "breakpoints"). We also note that a similar model, which considers breakage and fusion as independent events but with no deletions/duplications, was previously used by Levy et al. [54] to analyze chromosomal aberrations caused by ionizing radiation in M-FISH data.

A solution for karyotype sorting, i.e. a shortest sequence of events that led to a given abnormal karyotype, is usually non-unique. In particular, if two homolog broken ends exist - then there may be two alternative fusion events for the solution. Moreover, different solutions may differ only by the order of their events. Therefore,

in many cases it is preferable to consider the reconstructed events as a set rather than as a sequence. As we already argued above, imposing further preferences / limitations on the reconstructed sequence of events is likely to decrease the number of possible solutions and is an important direction for future work. A potential preference is to favor solutions that induce translocations (i.e., two consecutive breakages immediately followed by two fusions between the corresponding four broken ends), and other complex rearrangements that are frequent in real data (Chapter 6).

Finally, the association of cancer karyotypes with a plausible set of rearrangements can be viewed as the first step in their analysis. Later steps may include various statistical analyses, such as identifying rearrangements that are likely to be of importance to the carcinogenesis process (e.g. [16]), reconstructing common evolutionary pathways (e.g. [32, 33, 109, 85, 44, 43]), or discovering interesting properties and associations among rearrangements (e.g. Chapters 6 and 7). We note that most of the statistical studies of rearrangement events in cancer, at the least the ones that we are aware of, analyze elementary events: duplications/deletions of segments (commonly CGH data), and fusions.

8.4 Analyzing Rearrangements in Cancer Karyotypes

In chapter 6 we analyzed and compared rearrangement frequencies in different cancers. The analyzed rearrangements were reconstructed by a heuristic algorithm that given a cancer karyotype iteratively detects a most probable event and undoes it. We ignored the order of the reconstructed events, as many of the events commute. The algorithm fails if it cannot reconstruct a unique set of events. The algorithm was shown to succeed on more than 98% of the data, totalling 40,298 well-characterized karyotypes derived from the Mitelman database [62]. We note that the high effectiveness of the algorithm may be due to the relative simplicity of the karyotypes in the data. For example, the average number of reconstructed events per karyotype is less than 3 (see Fig. 2.(a) for average event rates).

The classification into cancer classes was based on the histological data provided for each karyotype. We showed that the vast majority (98%) of cancer karyotypes can be explained using 12 types of rearrangement events, among which the most common were: chromosome gains and losses, translocations, and terminal deletions. One goal of this study was to set a basis for modeling rearrangements in cancer.

Our results showed that unlike the modeling of rearrangement in species evolution, a realistic model for cancer cannot ignore the dominance of duplications and deletions in cancer genomes.

We used the reconstructed rearrangement frequencies to compare distinct cancers. More specifically, the question we asked was: Are there significant differences in the frequencies of rearrangement events between distinct cancers? To answer this question we designed several methods to compare event frequencies in different cancers. We applied these methods to cancer classes with a sufficient number of samples (i.e. more than 60 karyotypes). The results showed that for most compared cancer pairs, the observed distributions of rearrangement frequencies were significantly distinguishable.

To the best of our knowledge, this study presented the first large-scale analysis of the frequencies of rearrangement events in different cancers. Previous comparable studies were either applied to very small datasets (such as the NCI-60 [86]) or focused on the behavior of a single parameter of karyotypic complexity, such as the total number of aberrations [37, 26]. We note that the distinct distributions of event rates observed for different cancers may result from different recurrent aberrations, such as the Philadelphia translocation in CML. Our results imply that the mechanisms underlying chromosome instability vary for cancers of different histological origins.

Our next step was to analyze rearrangement events with their specific chromosomal locations (Chapter 7). We used the term *aberration* to refer to the result of a rearrangement event on a specific chromosomal location(s), and used an ISCN-like notation to identify it. For example, the aberration "t(9;22)(q34;q11)" referred to the result of a translocation event on the chromosomal locations 9q34 and 22q11. We employed our heuristic for rearrangement reconstruction on a set of over 15,000 karyotypes from the Mitelman database, and assigned each karyotype with its set of reconstructed aberrations. The remaining karyotypes in the Mitelman database were excluded from our analysis to avoid potential biases in the reconstructed aberrations. This study was comprised of two complementing parts. In the first, we computed a P-value for the correlation of each aberration-class pair. Reassuringly, the lowest P-values matched well-known strong correlations. These P-values were then used to compare distinct tumor classes by their aberrations. Our results proved that class similarity based on manifested chromosomal aberrations is remarkably concordant with histological similarity. In addition, we revealed a novel significant similarity

among three childhood cancers, Wilms tumor, Ewing sarcoma, and hepetoblastoma. Very recently, Liu et al. presented an evolutionary tree of cancers based on copy number alterations derived from CGH data [57]. As the cancers in the tree of Liu et al. are different from the ones we analyzed, it is almost impossible to compare between the results. Nevertheless, despite using different data and methods, the tree constructed by Liu et al. was also highly concordant with histological classification, supporting our conclusion.

In the second part of our study, we detected aberration pairs that showed significant co-occurrence rates, regardless of the cancer class they were found in. Interestingly, there was a clear dichotomy in the significantly co-occurring aberrations: almost all strong couples involved either two chromosome gain or two chromosome loss aberrations, but not both. In other words, while there were many strong chromosome gain couples and chromosome loss couples, any co-occurrence of chromosome gain and chromosome loss aberrations appeared to be random. We repeated this test in an independent CGH dataset. Strikingly, we found that the CGH dataset showed the same chromosome gain/loss co-occurrence dichotomy. A similar result was obtained in the study of karyotypic evolution models of several specific solid cancers by Höglund et al. [44]. The models developed in [44] contain two converging karyotype evolution paths, one dominated by gains of chromosomal fragments and the other by losses. Since the analysis methods in [44] were completely different from ours, this result lends further support to our observation of whole chromosome gains and losses dichotomy in aneuploid karyotypes. The strong gain-gain and loss-loss correlations we found suggests that these links are required for balancing the ratios of proteins that function in complexes. As chromosome gain and loss events may result in partially assembled protein complexes or free subunits, which put significant stress on the cell [113], such balancing can be crucial for the survival of the aneuploid cell.

8.5 Concluding Remarks

Our research of genome rearrangements initially focused on genomic sorting under different models. Genomic sorting has been the source of many intriguing problems that caught the attention of many computer scientists and mathematicians over the past two decades. Despite its over-simplification of biology, genomic sorting

turned out to be computationally very complicated, and often NP-hard, for most considered models. Looking at the history of SBR, the most studied genomic sorting problem, we can conclude that the research of genomic sorting has been fruitful, both computationally and biologically. The mathematical theory underlying SBR has been greatly extended and simplified since the problem was introduced by Kececioglu and Sankoff, leading to faster and simpler algorithms for solving it. Computational knowledge on simplistic genomic sorting problems can be used for devising clever heuristics for computing parsimonious rearrangement scenarios involving more than two species, as was done in [19].

In this thesis we extended and simplified the theory of an existing problem, namely SBT, by developing a combinatorial framework akin to the framework of SBR. Later on, we presented two new models for genome rearrangements. The first built on the model of SBT, while the second used a novel set of rearrangements suited for cancer. For the first model we succeeded in providing an accurate polynomial time solution, but the computational analysis was very complicated. For the second model, we managed to provide a 3-approximation polynomial time solution, under certain assumptions, while the overall complexity of the problem remained unknown. We hope that further investigations of these models will simplify and improve our results.

Finally, we developed an effective heuristic to sort cancer karyotypes using 12 common rearrangement events and used the reconstructed rearrangements to carry out statistical analyses. We conducted large-scale robust statistical investigations of the rearrangements reconstructed from thousands of karyotypes, searching for differences / relationships between distinct cancers and identifying significant co-occurring aberrations. Our results revealed new characteristics of chromosomal rearrangements in cancer, which may shed light on aberration development mechanisms in cancer. We believe that the wealth of cancer karyotypes merits additional investigations of these data which will hopefully provide more insights on the role and importance of chromosomal aberrations in cancer.

Acronyms

Array-CGH - Array-based Comparative Genomic Hybridization

CML - Chronic Myelogenous Leukemia

DCJ - Double-Cut-and-Join

DSB - Double Strand Break

ESP - End Sequence Profiling

FISH - Fluorescence In Situ Hybridization

ISCN - International System for human Cytogenetic Nomenclature

KS - Karyotype Sorting

M-FISH - Multiplex Fluorescence In Situ Hybridization

RKS - Reduced Karyotype Sorting

SBLT - Sorting By Legal Translocations

SBR - Sorting By Reversals

SBRT - Sorting By Reversals and Translocations

SBT - Sorting By Translocations

SDCJ - Sorting By Double-Cut-and-Join operations

SKY - Spectral Karyotyping

Bibliography

[1] `http://www.biotechnologyonline.gov.au/popups/img_karyotype.html`.

[2] W. Ackermann. Zum hilbertshen aufbau der reelen zahlen. *Math. Ann.*, 99:118–133, 1928.

[3] D.G. Albertson, C. Collins, F. McCormick, and J. W. Gray. Chromosome aberrations in solid tumors. *Nature Genetics*, 34:369–376, 2003.

[4] M.A. Alekseyev and P.A. Pevzner. Whole genome duplications and contracted breakpoint graphs. *SIAM Journal on Computing*, 36(6):1748–1763, 2007.

[5] M.A. Alekseyev and P.A. Pevzner. Multi-break rearrangements and chromosomal evolution. *Theoretical Computer Science*, 395(2):193–202, 2008.

[6] W.J. Ansorge. Next-generation DNA sequencing techniques. *New biotechnology*, 25(4):195–203, 2009.

[7] D.A. Bader, B. M.E. Moret, and M. Yan. A linear-time algorithm for computing inversion distance between signed permutations with an experimental study. *Journal of Computational Biology*, 8(5):483–491, 2001.

[8] M. Bader. Sorting by reversals, block interchanges, tandem duplications, and deletions. *BMC bioinformatics*, 10(Suppl 1):S9, 2009.

[9] V. Bafna and P. A. Pevzner. Genome rearragements and sorting by reversals. *SIAM Journal on Computing*, 25(2):272–289, 1996. A preliminary version appeared in Proc. 34th IEEE Symp. of the Foundations of Computer Science, p. 148–157, 1994.

[10] M.A. Bender, D. Ge, S. He, H. Hu, R.Y. Pinter, S. Skiena, and F. Swidan. Improved bounds on sorting by length-weighted reversals. *Journal of Computer and System Sciences*, 74(5):744–774, 2008.

[11] A. Bergeron. A very elementary presentation of the Hannenhalli-Pevzner theory. *Discrete Applied Mathematics*, 146(2):134–145, 2005.

[12] A. Bergeron, J. Mixtacki, and J. Stoye. Reversal distance without hurdles and fortresses. *Lecture Notes in Computer Science*, 3109:388–399, 2004.

[13] A. Bergeron, J. Mixtacki, and J. Stoye. A unifying view of genome rearrangements. *Lecture Notes in Computer Science*, 4175:163–173, 2006.

[14] A. Bergeron, J. Mixtacki, and J. Stoye. On sorting by translocations. *Journal of Computational Biology*, 13(2):567–578, 2006.

[15] P. Berman and S. Hannenhalli. Fast sorting by reversal. In *Proceedings of the 7th Annual Symposium Combinatorial Pattern Matching (CPM)*, volume 1075 of *LNCS*, pages 168–185. Springer, 1996.

[16] R. Beroukhim, G. Getz, L. Nghiemphu, J. Barretina, T. Hsueh, D. Linhart, I. Vivanco, J.C. Lee, J.H. Huang, S. Alexander, et al. Assessing the significance of chromosomal aberrations in cancer: methodology and application to glioma. *Proceedings of the National Academy of Sciences*, 104(50):20007, 2007.

[17] G.R. Bignell, T. Santarius, J. Pole, A.P. Butler, J. Perry, E. Pleasance, C. Greenman, A. Menzies, S. Taylor, S. Edkins, et al. Architectures of somatic genomic rearrangement in human cancer amplicons at sequence-level resolution. *Genome Research*, 17(9):1296, 2007.

[18] G. Bourque and L.Zhang. Models and methods in comparative genomics. *Advances in Computers*, 68:60–105, 2006.

[19] G. Bourque and P.A. Pevzner. Genome-scale evolution: Reconstructing gene orders in the ancestral species. *Genome Research*, 12(1):26–36, 2002.

[20] G. Bourque, P.A. Pevzner, and G. Tesler. Reconstructing the genomic architecture of ancestral mammals: Lessons from human, mouse, and rat genomes. *Genome Research*, 14(4):507–516, 2004.

[21] T. Boveri. *Zur frage der entstehung maligner tumoren*. Gustav Fischer, 1914.

[22] M.D.V. Braga. baobabLUNA: the solution space of sorting by reversals. *Bioinformatics*, 25(14):1833, 2009.

[23] M.D.V. Braga, M.F. Sagot, C. Scornavacca, and E. Tannier. Exploring the solution space of sorting by reversals, with experiments and an application to evolution. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pages 348–356, 2008.

[24] M.D.V. Braga and J. Stoye. Counting All DCJ Sorting Scenarios. In *Comparative Genomics: International Workshop, RECOMB-CG 2009, Budapest, Hungary, September 27-29, 2009, Proceedings*, page 36. Springer, 2009.

[25] D. Bryant. The complexity of calculating exemplar distances. *Comparative Genomics: Empirical and Analytical Approaches to Gene Order Dynamics, Map Alignment, and the Evolution of Gene Families*, pages 207–212, 2000.

[26] M.A.A. Castro, T.T.G. Onsten, R.M.C. de Almeida, and J.C.F. Moreira. Profiling cytogenetic diversity with entropy-based karyotypic analysis. *Journal of theoretical biology*, 234(4):487–495, 2005.

[27] K. Chaudhuri, K. Chen, R. Mihaescu, and S. Rao. On the tandem duplication-random loss model of genome rearrangement. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm (SODA)*, pages 564–570. ACM, 2006.

[28] X. Chen, J. Zheng, Z. Fu, P. Nan, Y. Zhong, S. Lonardi, and T. Jiang. Assignment of orthologous genes via genome rearrangement. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pages 302–315, 2005.

[29] R. Dalla-Favera, S. Martinotti, R.C. Gallo, J. Erikson, and C.M. Croce. Translocation and rearrangements of the c-myc oncogene locus in human undifferentiated B-cell lymphomas. *Science*, 219(4587):963–967, 1983.

[30] A.C.E. Darling, B. Mau, F.R. Blattner, and N.T. Perna. Mauve: multiple alignment of conserved genomic sequence with rearrangements. *Genome research*, 14(7):1394–1403, 2004.

[31] A. de Klein, A.G. van Kessel, G. Grosveld, C.R. Bartram, A. Hagemeijer, D. Bootsma, N.K. Spurr, N. Heisterkamp, J. Groffen, and J.R. Stephenson. A cellular oncogene is translocated to the Philadelphia chromosome in chronic myelocytic leukaemia. *Nature*, 300:765–767, 1982.

[32] R. Desper, F. Jiang, O. Kallioniemi, H. Moch, C. Papadimitrou, and A. Schäffer. Inferring tree models for oncogenesis from comparative genome hybridization data. *Journal of Computational Biology*, 6:37–51, 1999.

[33] R. Desper, F. Jiang, O. Kallioniemi, H. Moch, C. Papadimitrou, and A. Schäffer. Distance-based reconstruction of tree models for oncogenesis. *Journal of Computational Biology*, 7:789803, 2000.

[34] B.J. Druker, M. Talpaz D., Resta, B. Peng, E. Buchdunger, J. Ford, and C.L. Sawyers. Clinical efficacy and safety of an ABL specific tyrosine kinase inhibitor as targeted therapy for chronic myelogenous leukemia. *Blood*, 94(suppl 1):368a, 1999.

[35] P. Duesberg. `http://berkeley.edu/news/media/releases/2007/06/26_drugresistance.shtml`.

[36] N. El-Mabrouk and D. Sankoff. The reconstruction of doubled genomes. *SIAM Journal on Computing*, 32(3):754–792, 2003.

[37] A. Frigyesi, D. Gisselsson, F. Mitelman, and M. Hoglund. Power Law Distribution of Chromosome Aberrations in Cancer. *Cancer Research*, 63(21):7094–7097, 2003.

[38] L. Froenicke, M.G. Caldés, A. Graphodatsky, S. Muller, L.A. Lyons, T.J. Robinson, M. Volleth, F. Yang, and J. Wienberg. Are molecular cytogenetics and bioinformatics suggesting diverging models of ancestral mammalian genomes?, 2006.

[39] S. Hannenhalli. Polynomial algorithm for computing translocation distance between genomes. *Discrete Applied Mathematics*, 71:137–151, 1996.

[40] S. Hannenhalli and P. Pevzner. Transforming men into mice (polynomial algorithm for genomic distance problems). In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 581–592. IEEE Computer Society Press, 1995.

[41] S. Hannenhalli and P. Pevzner. Transforming cabbage into turnip: Polynomial algorithm for sorting signed permutations by reversals. *Journal of the ACM*, 46:1–27, 1999.

[42] B. Hiller, J. Bradtke, H. Balz, and H. Rieder. CyDAS: a cytogenetic data analysis system. *BioInformatics*, 21(7):1282.–1283, 2005. `http://www.cydas.org`.

[43] M. Hjelm, M. Höglund, and J. Lagergren. New probabilistic network models and algorithms for oncogenesis. *Journal of Computational Biology*, 13(4):853–865, 2006.

[44] M. Höglund, A. Frigyesi, T. Säll, D. Gisselsson, and F. Mitelman. Statistical behavior of complex cancer karyotypes. *Genes, Chromosomes and Cancer*, 42(4):327–341, 2005.

[45] S. B. Hoot and J. D. Palmer. Structural rearrangements, including parallel inversions, within the chloroplast genome of Anemone and related genera. *J. Molecular Evooution*, 38:274–281, 1994.

[46] G. Jean and M. Nikolski. Genome rearrangements: a correct algorithm for optimal capping. *Information Processing Letters*, 104(1):14–20, 2007.

[47] C.L. Kahn, S. Mozes, and B.J. Raphael. Efficient algorithms for analyzing segmental duplications with deletions and inversions in genomes. *Algorithms for Molecular Biology*, 5:11, 2010.

[48] C.L. Kahn and B.J. Raphael. Analysis of segmental duplications via duplication distance. *Bioinformatics*, 24(16):i133, 2008.

[49] H. Kaplan, R. Shamir, and R. E. Tarjan. Faster and simpler algorithm for sorting signed permutations by reversals. *SIAM Journal of Computing*, 29(3):880–892, 2000.

[50] H. Kaplan and E. Verbin. Sorting signed permutations by reversals, revisited. *Journal of Computer and System Sciences*, 70(3):321–341, 2005.

[51] J. Kececioglu and D. Sankoff. Exact and approximation algorithms for sorting by reversals, with application to genome rearrangement. *Algorithmica*, 13(1/2):180–210, January 1995. A preliminary version appeared in *Proc. CPM93*, Springer, Berlin, 1993, pages 87–105.

[52] J. D. Kececioglu and R. Ravi. Of mice and men: Algorithms for evolutionary distances between genomes with translocation. In *Proceedings of the 6th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 604–613. ACM Press, 1995.

[53] K.K. Khanna and S.P. Jackson. DNA double-strand breaks: signaling, repair and the cancer connection. *Nature genetics*, 27(3):247–254, 2001.

[54] D. Levy, M. Vázquez, B.D. Loucas, M.N. Cornforth, R.K. Sachs, and J. Arsuaga. Comparing DNA damage-processing pathways by computer analysis of chromosome painting data. *Journal of Computational Biology*, 11(4):626–641, 2004.

[55] J.R. Lupski L.G. Shaffer. Molecular mechanisms for constitutional chromosomal rearrangements in humans. *Annu. Rev. Genet*, 34:297–329.

[56] G. Li, X. Qi, X. Wang, and B. Zhu. A linear-time algorithm for computing translocation distance between signed genomes. In *Proceedings of the 15th Annual Symposium on Combinatorial Pattern Matching (CPM)*, volume 3109 of *LNCS*, pages 323–332. Springer, 2004.

[57] J. Liu, N. Bandyopadhyay, S. Ranka, M. Baudis, and T. Kahveci. Inferring Progression Models for CGH data. *Bioinformatics*, 2009.

[58] M. Marron, K.M. Swenson, and B.M.E. Moret. Genomic distances under deletions and insertions. *Theoretical Computer Science*, 325(3):347–360, 2004.

[59] O.J. Marshall, A.C. Chueh, L.H. Wong, and K.H.A. Choo. Neocentromeres: new insights into centromere structure, disease development, and karyotype evolution. *The American Journal of Human Genetics*, 82(2):261–282, 2008.

[60] I. Miklos and A.E. Darling. Efficient sampling of parsimonious inversion histories with application to genome rearrangement in Yersinia. *Genome Biology and Evolution*, 2009(0):153, 2009.

[61] F. Mitelman, editor. *ISCN (1995): An International System for Human Cytogenetic Nomenclature*. S. Karger, Basel, 1995.

[62] F. Mitelman, B. Johansson, and F. Mertens (Eds.). Mitelman database of chromosome aberrations in cancer, 2009. `http://cgap.nci.nih.gov/Chromosomes/Mitelman`.

[63] F. Mitelman, B. Johansson, and F. Mertens. The impact of translocations and gene fusions on cancer causation. *Nature Reviews Cancer*, 7(4):233–245, 2007.

[64] J. Mixtacki. Genome halving under DCJ revisited. *Lecture Notes In Computer Science*, 5092:276–286, 2008.

[65] W.J. Murphy, D.M. Larkin, A. Everts van der Wind, G. Bourque, G. Tesler, L. Auvil L, J.E. Beever, B.P. Chowdhary, F. Galibert, L. Gatzke, C. Hitte, S.N. Meyers, D. Milan, E.A. Ostrander, G. Pape, H.G. Parker, T. Raudsepp, M.B. Rogatcheva, L.B. Schook, L.C. Skow, M. Welge, J.E. Womack, S.J. O'brien, P.A. Pevzner, and H.A. Lewin. Dynamics of mammalian chromosome evolution inferred from multispecies comparative maps. *Science*, 309:613–617, 2005.

[66] C.T. Nguyen, YC Tay, and L. Zhang. Divide-and-conquer approach for the exemplar breakpoint distance. *Bioinformatics*, 21(10):2171, 2005.

[67] P.C. Nowell and D.A. Hungerford. A minute chromosome in human chronic granulocytic leukemia. *Science*, 132:1497, 1960.

[68] M. Ozery-Flato and R. Shamir. Two notes on genome rearragnements. *Journal of Bioinformatics and Computational Biology*, 1(1):71–94, 2003.

[69] M. Ozery-Flato and R. Shamir. An $O(n^{3/2}\sqrt{\log(n)})$ algorithm for sorting by reciprocal translocations. In *Proceedings of the 17th Annual Symposium on Combinatorial Pattern Matching (CPM)*, volume 4009 of *LNCS*. Springer, 2006.

[70] M. Ozery-Flato and R. Shamir. Sorting by translocations via reversals theory. In *Proceedings of the fourth RECOMB Satellite Workshop on Comparative Genomics*, volume 4205 of *LNCS*, pages 87–98. Springer, 2006.

[71] M. Ozery-Flato and R. Shamir. On the frequency of genome rearrangement events in cancer karyotypes. Technical report, Tel Aviv University, 2007.

[72] M. Ozery-Flato and R. Shamir. Rearrangements in genomes with centromeres - part I: translocations. In *Proceedings of the 11th Annual International Conference on Computational Molecular Biology (RECOMB'07)*, volume 4453 of *LNCS*. Springer, 2007.

[73] M. Ozery-Flato and R. Shamir. Sorting by translocations via reversals theory. *Journal of Computational Biology*, 14(4):408–422, 2007.

[74] M. Ozery-Flato and R. Shamir. Sorting cancer karyotypes by elementary operations. In *Proceedings of the sixth RECOMB Satellite Workshop on Comparative Genomics*, volume 5267 of *LNCS*. Springer, 2008.

[75] M. Ozery-Flato and R. Shamir. Sorting genomes with centromeres by translocations. *Journal of Computational Biology*, 15(7):1–20, 2008.

[76] M. Ozery-Flato and R. Shamir. Sorting cancer karyotypes by elementary operations. *Journal of Computational Biology*, 16(10):1445–1460, 2009.

[77] M. Ozery-Flato and R. Shamir. An $O(n^{3/2}\sqrt{\log(n)})$ algorithm for sorting by reciprocal translocations. *Journal of Discrete Algorithms*, 2010. To appear.

[78] J. D. Palmer and L. A. Herbon. Tricircular mitochondrial genomes of Brassica and Raphanus: reversal of repeat configurations by inversion. *Nucleic Acids Research*, 14:9755–9764, 1986.

[79] J. D. Palmer and L. A. Herbon. Unicircular structure of the Brassica hirta mitochondrial genome. *Current Genetics*, 11:565–570, 1987.

[80] J. D. Palmer and L. A. Herbon. Plant mitochondrial DNA evolves rapidly in structure, but slowly in sequence. *J. Molecular Evolution*, 28:87–97, 1988.

[81] J. D. Palmer, B. Osorio, and W.R. Thompson. Evolutionalry significance of inversions in legume chorloplast DNAs. *Current Genetics*, 14:65–74, 1988.

[82] P.A. Pevzner and G. Tesler. Genome rearrangements in mammalian evolution: Lessons from human and mouse genomes. *Genome Research*, 13(1):37–45, 2003.

[83] D. Pinkel, T. Straume, and JW Gray. Cytogenetic analysis using quantitative, high-sensitivity, fluorescence hybridization. *Proceedings of the National Academy of Sciences*, 83(9):2934–2938, 1986.

[84] A. J. Radcliffe, A. D. Scott, and E. L. Wilmer. Reversals and transpositions over finite alphabets. *SIAM J. Discret. Math.*, 19(1):224–244, 2005.

[85] M.D. Radmacher, R. Simon, R. Desper, R. Taetle, A.A. SCHÄFFER, and M.A. Nelson. Graph models of oncogenesis with an application to melanoma. *Journal of theoretical biology*, 212(4):535–548, 2001.

[86] A.V. Roschke, G. Tonon, K.S. Gehlhaus, N. McTyre, K.J. Bussey, S. Lababidi, D.A. Scudiero, J.N. Weinstein, and I.R. Kirsch. Karyotypic complexity of the NCI-60 drug-screening panel, 2003.

[87] J.D. Rowley. A new consistent chromosomal abnormality in chronic myelogenous leukaemia identified by quinacrine fluorescence and giemsa staining. *Nature*, 243:290293, 1973.

[88] D. Sankoff. Edit distance for genome comparison based on non-local operations. In *Proceedings of the third Annual Symposium on Combinatorial Pattern Matching (CPM)*, volume 644 of *LNCS*, pages 121–135, 1992.

[89] D. Sankoff. Genome rearrangement with gene families. *Bioinformatics*, 15(11):909, 1999.

[90] D. Sankoff, R. Cedergren, and Y. Abel. Genomic divergence through gene rearrangement. *Methods in Enzymology*, 183:428–438, 1990.

[91] D. Sankoff, G. Leduc, N. Antoine, B. Paquin, B.F. Lang, and R. Cedergren. Gene order comparisons for phylogenetic inference: Evolution of the mitochondrial genome. *Proceedings of the National Academy of Sciences*, 89(14):6575–6579, 1992.

[92] H. Scherthan, T. Cremer, U. Arnason, H.U. Weier, A. Lima-de Faria, and L. Frönicke. Comparative chromosome painting discloses homologous segments in distantly related mammals. *Nature genetics*, 6(4):342–347, 1994.

[93] E. Schröck, S. Du Manoir, T. Veldman, B. Schoell, J. Wienberg, MA Ferguson-Smith, Y. Ning, DH Ledbetter, I. Bar-Am, D. Soenksen, et al. Multicolor spectral karyotyping of human chromosomes. *Science*, 273(5274):494, 1996.

[94] J. Shendure and H. Ji. Next-generation DNA sequencing. *nature biotechnology*, 26(10):1135–1145, 2008.

[95] A. Siepel. An algorithm to enumerate sorting reversals for signed permutations. *Journal of Computational Biology*, 10(3-4):575–597, 2003.

[96] A. M. Snijders and N. Nowak et al. Assembly of microarrays for genome-wide measurement of DNA copy number. *Nature Genetics*, 29:263–264, 2001.

[97] M.R. Speicher, S.G. Ballard, and D.C. Ward. Karyotyping human chromosomes by combinatorial multi-fluor FISH. *Nature Genetics*, 12(4):368–375, 1996.

[98] A. H. Sturtevant and Th. Dobzhansky. Inversions in the third chromosome of wild races of drosophila pseudoobscura, and their use in the study of the history of the species. *Proceedings of the National Academy of Science USA*, 22:448–450, 1936.

[99] B.A. Sullivan, M.D. Blower, and G.H. Karpen. Determining centromere identity: Cyclical stories and forking paths. *Nature Reviews Genetics*, 2(8):584–596, 2001.

[100] K.M. Swenson, M. Marron, J.V. Earnest-DeYoung, and B.M.E. Moret. Approximating the true evolutionary distance between two genomes. *Journal of Experimental Algorithmics (JEA)*, 12:3–5, 2008.

[101] K.M. Swenson, V. Rajan, Y. Lin, and B.M. Moret. Sorting Signed Permutations by Inversions in $O(n \log n)$ Time. In *Proceedings of the 13th Annual International Conference on Research in Computational Molecular Biology*, pages 386–399. Springer-Verlag, 2009.

[102] F. Swidan, E.P.C. Rocha, M. Shmoish, and R.Y. Pinter. An integrative method for accurate comparative genome mapping. *PLoS Computational Biology*, 2:e75, 2006.

[103] E. Tannier, A. Bergeron, and M. Sagot. Advances on sorting by reversals. *Discrete Applied Mathematics*, 155(6-7):881–888, 2007.

[104] E. Tannier and M. Sagot. Sorting by reversals in subquadratic time. In *Proc. 15th Annual Symposium on Combinaotrial Pattern Matching (CPM '04)*, pages 1–13. Springer, 2004.

[105] G. Tesler. Efficient algorithms for multichromosomal genome rearrangements. *J. Comp. Sys. Sci.*, 65(3):587–609, 2002.

[106] B. Vogelstein, E.R. Fearon, S.R. Hamilton, S.E. Kern, A.C. Preisinger, M. Leppert, Y. Nakamura Y, R. White, A.M. Smits, and J.L.Bos. Genetic alterations during colorectal tumor development. *N. Engl. J. Med.*, 319:525–532, 1988.

[107] S. Volik and B.J. Raphael et al. Decoding the fine-scale structure of a breast cancer genome and transcriptome: Implications for a tumor genome project. *Genome Research*, 16(3):394–404, 2006.

[108] S. Volik and S. Zhao et al. End-sequence profiling: Sequence-based analysis of aberrant genomes. *Proceedings of the National Academy of Science USA*, 100:7696–7701, 2003.

[109] A. von Heydebreck, B. Gunawan, and L. Füzesi. Maximum likelihood estimation of oncogenetic tree models. *Biostatistics*, 5:545556, 2004.

[110] LE Voullaire, HR Slater, V. Petrovic, and KH Choo. A functional marker centromere with no detectable alpha-satellite, satellite III, or CENP-B protein: activation of a latent centromere? *American journal of human genetics*, 52(6):1153, 1993.

[111] L. Wang, D. Zhu, X. Liu, and S. Ma. An o(n2) algorithm for signed translocation. *Journal of Computer and System Sciences*, 70(3):284 – 299, 2005.

[112] R.H. Waterston, K. Lindblad-Toh, E. Birney, J. Rogers, J.F. Abril, R. Agarwala P. Agarwal, R. Ainscough, M. Alexandersson, and P. An et al. Initial sequencing and comparative analysis of the mouse genome. *Nature*, 420(6915):520–562, 2002.

[113] B.R. Williams and A. Amon. Aneuploidy: Cancer's Fatal Flaw? *Cancer Research*, 69(13):5289, 2009.

[114] S. Yancopoulos, O. Attie, and R. Friedberg. Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinformatics*, 21(16):3340–3346, 2005.

[115] H. Zhao and G. Bourque. Recovering true rearrangement events on phylogenetic trees. *Lecture Notes in Computer Science*, 4751:149, 2007.

[116] H. Zhao and G. Bourque. Recovering genome rearrangements in the mammalian phylogeny. *Genome Research*, 19(5):934, 2009.

מאורע שכפול כרומוזום עם מאורע של מחיקת כרומוזום הסתבר כאקראי לחלוטין. דיכוטומיה זו בין מאורעות שכפול ומחיקה של כרומוזום רומזת על קיומו של מנגנון, או תהליך העדפה, המאפשר לתא הסרטני יכולת לאזן מחדש את היחס המספרי בין הגנים, לאחר שזה הופר עקב מאורע קודם של שכפול / מחיקת כרומוזום.

5. On the frequency of genome rearrangement events in cancer karyotypes.

Michal Ozery-Flato and Ron Shamir.
Technical report [71]. Accepted for presentation in the first RECOMB Satellite Workshop on Computation Cancer Biology (RECOMB-CCB'07) (peer-reviewed, no proceedings).

במאמר זה הצגנו גישה חדשנית לניתוח מאורעות רה-ארגון גנומים המתרחשים במהלך התפתחות מחלת הסרטן. גישה זו התבססה על יוריסטיקה יעילה לחישוב סדרת מאורעות רה-ארגון קצרה ביותר המסבירה את היווצרותו של קריוטיפ סרטני נתון. יוריסטיקה זו עשתה שימוש ב-12 פעולות רה-ארגון מורכבות נפוצות והוחלה על למעלה מ- 40,000 קריוטיפים סרטניים מבסיס הנתונים של מיטלמן. מניתוח התוצאות עולה כי הפרעות כרומוזומליות בקריוטיפים נוצרות בעיקר עקב פעולות של שכפול ומחיקה של כרומוזומים שלמים, שחלוף, ומחיקות קצה של כרומוזומים. התדירויות של הפעולות ששוחזרו שמשו כדי להשוות בין סוגי סרטן שונים. באמצעות שיטות קיבוץ (clustering) שונות, הראינו כי ההתפלגויות של תדירויות פעולות רה-ארגון גנומי נבדלות באופן משמעותי עבור מרבית סוגי הסרטן.

6. A systematic assessment of associations among chromosomal aberrations in cancer karyotypes.

Michal Ozery-Flato, Chaim Linhart, Luba Trakhtenbrot, Shai Izraeli, and Ron Shamir.
Submitted.

בעבודה זו, המהווה המשך ישיר של העבודה הקודמת, ביצענו ניתוח שיטתי של מאורעות רה-ארגון גנומים בסוגי סרטן שונים. מחקר זה הקיף למעלה מ- 15,000 קריוטיפים המשתייכים ל-64 סוגי סרטן שונים. בדומה לעבודה הקודמת, הקריוטיפים נלקחו מבסיס הנתונים של מיטלמן ושחזור המאורעות בוצע באמצעות אותה היוריסטיקה. להבדיל מהעבודה הקודמת, אשר התמקדה רק בניתוח סוגי המאורעות (קרי 12 הפעולות), בעבודה זו נלקחו בחשבון המיקומים הכרומוזומליים המעורבים לכל מאורע משוחזר. לכל סוג סרטן חישבנו קבוצת מאורעות אופייניים, ובה השתמשנו לצורך מדידת הדמיון בין סוגי סרטן בעלי מאורעות משותפים. בהסתמך על מידת הדמיון שחישבנו בין זוגות של סוגי סרטן שונים, בצענו תהליך קיבוץ מדרגי (hierarchical clustering) אשר תוצאותיו תאמו במידה רבה את החלוקה הקיימת לפי סוג הרקמה הסרטנית, ובאופן כללי הפריד בין שלוש מחלקות סרטן: סרטני לימפה, סרטני דם שאינם לימפואידים, וסרטנים סולידים. בנוסף, התוצאות חשפו נקודת דמיון חדשה בין שלושה סוגי סרטן בילדים שמקורם ברקמה עוברית. בניתוח אחר של המאורעות המשוחזרים התגלתה תופעה מובהקת מאוד סטטיסטית: מאורעות שכפול כרומוזום נוטים להופיע עם מאורעות אחרים של שכפול כרומוזום, ומאורעות מחיקת כרומוזום נוטים להופיע עם מאורעות אחרים של מחיקת כרומוזום, ועם זאת כל הופעה של

3. Sorting Genomes with Centromeres by Translocations.

Michal Ozery-Flato and Ron Shamir.
Published in *Proceedings of the 11th Annual International Conference on Computational Molecular Biology (RECOMB'07)* [72] and in *Journal of Computational Biology (JCB)* [75].

במאמר זה המשכנו ללמוד את בעיית המיון באמצעות פעולות שחלוף, אולם לראשונה תחת מודל הכולל גם צנטרומרים. מכיוון שקיום צנטרומר יחיד הוא חיוני להמשך קיומו של כרומוזום, פעולות רה-ארגון היוצרות כרומוזומים חסרי צנטרומר הן בעלות סבירות נמוכה יותר מאלו השומרות על קיום הצנטרומרים בכל כרומוזום חדש. במאמר זה הצגנו לראשונה אלגוריתם פולינומיאלי לחישוב סדרה קצרה ביותר של פעולות שחלוף בין שני גנומים נתונים, כאשר כל כרומוזום שנוצר במהלך סדרה זו מכיל צנטרומר יחיד. עבודה זו מהווה צעד ראשון בשקלולם של הצנטרומרים בתרחישי רה-ארגון גנומיים ושחזור של סדרות מיון סבירות יותר מבחינה ביולוגית.

4. Sorting Cancer Karyotypes by Elementary Operations.

Michal Ozery-Flato and Ron Shamir.
Published in *Proceedings of the sixth RECOMB Satellite Workshop on Comparative Genomics* [74] and in *Journal of Computational Biology (JCB)* [76].

בעבודה זו הצגנו וניתחנו את בעיית המיון של קריוטיפים סרטניים. בעיה זו התמקדה במציאת רצף פעולות רה-ארגון גנומי קצר ביותר אשר מוביל להיווצרותו של קריוטיפ סרטני נתון. חקרנו בעיה זו תחת מודל מתימטי מקורי להתפתחות הפרעות כרומוזומליות בקריוטיפים סרטניים אשר מניח קיומן של ארבע פעולות רה-ארגון בסיסיות על כרומוזומים: שכפול, מחיקה, שבירה ואיחוד. תחת הנחות מסויימות, הוכחנו חסם תחתון וחסם עליון לאורך הפתרון והצגנו אלגוריתם קירוב-3 פולינומיאלי לבעיה. אלגוריתם זה הופעל על קריוטיפים מבסיס הנתונים של מיטלמן אשר אוסף קריוטיפים סרטניים מהספרות המדעית. כ- 94% מהקריוטיפים בבסיס נתונים זה, 58,464 בסך הכל, תמכו בהנחותינו, וכל אחד מאלו מוין באמצעות האלגוריתם שהצגנו. במפתיע, למרות שהוכחת האלגוריתם מבטיחה רק רצף פעולות הארוך לכל היותר פי שלוש מהרצף הקצר ביותר, הרצפים שיוצרו בפועל השיגו את החסם התחתון (ולפיכך הינם קצרים ביותר) עבור 99.9% מהקריוטיפים שמוינו.

תקציר המאמרים הכלולים בתזה

עבודה זו מבוססת על שישה מאמרים. חמשת המאמרים הראשונים פורסמו בכתבי עת מדעיים או הוצגו בכנסים מדעיים. המאמר האחרון הוגש לאחרונה לכתב עת מדעי. להלן פירוט תקציר המאמרים:

1. An $O(n^{3/2}\sqrt{\log(n)})$ algorithm for sorting by reciprocal translocations.

Michal Ozery-Flato and Ron Shamir.
Published in *Proceedings of the 17th Annual Symposium on Combinatorial Pattern Matching (CPM'06)* [69] and Journal of Discrete Algorithms [77].

במאמר זה הוכחנו שבעיית המיון של גנום רב-כרומוזומים בעל n גנים באמצעות פעולות שחלוף ניתנת לפתרון בזמן $O(n^{3/2}\sqrt{\log(n)})$. האלגוריתם שפתחנו מבוסס על האלגוריתם של תנייר, ברג'רון וסגוט [104] עבור מיון גנום חד-כרומוזומלי ע"י פעולות היפוך. בכך שיפרנו את סיבוכיות הזמן הידועה לעומת האלגוריתם של ברג'רון, מיצטקי וסטוי [14], אשר רץ בזמן $O(n^3)$.

2. Sorting by reciprocal translocations via reversals theory.

Michal Ozery-Flato and Ron Shamir.
Published *in Proceedings of the fourth RECOMB Satellite Workshop on Comparative Genomics* [70] and in *Journal of Computational Biology* (JCB) [73].

מאמר זה הינו המשך ישיר למאמר הקודם. כאן חשפנו נקודות דמיון נוספות בין בעיית המיון ע"י פעולות היפוך (מע"ה) לבעיית המיון ע"י פעולות שחלוף (מע"ש). בפרט, בנינו שני אלגוריתמים חדשים לפתרון בעיית מע"ש, אשר מחקים אלגוריתמים ידועים לפתרון בעיית מע"ה. האלגוריתם הראשון הינו מבוסס ציונים, בדומה לאלגוריתם של ברג'רון [11], בעוד האלגוריתם השני מבוסס על פרוצדורה רקורסיבית, בדומה לאלגוריתם של ברמן והננהלי [15]. שני האלגוריתמים שהצגנו, וכן הוכחות הנכונות שלהם, מורכבים יותר בהשוואה לאלגוריתמים המקוריים, אך בד בבד, שומרים על אותם זמני ריצה אסימפטוטיים.

לקצוות שבורים אחרים ולכן כשלים בתיקון שד"גים יכולים להוביל לארועי רה-ארגון גנומי, ובינהם פעולות שחלוף, מחיקה ושכפול [53, 3]. ארועי רה-ארגון גנומי עלולים לתרום להיווצרות מחלת הסרטן, אם למשל מקטע שנמחק הכיל גן מעכב סרטן (tumor suppressor), או מקטע ששוכפל מכיל גן התורם לתהליך הסרטני (ongogene). פעולות המשנות את סדור הגנום, כמו פעולת שחלוף, יכולות להוביל ליצירתם של גנים חדשים (כמו הגן BCR-ABL) או לשנות את אופן הביטוי של גנים מסויימים ע"י החלפת אזורי הבקרה שלהם באלו של גנים אחרים (כמו הגן C-MYC בסרטני לימפה מסויימים [29]).

קריוטיפים סרטניים מראים לרוב מגוון רב של הפרעות כרומוזומליות. ישנן הפרעות כרומוזומליות מסויימות, ע"פ רוב כתוצאה מפעולת שחלוף, שאופייניות מאוד לסוגי סרטן מסוימים (בד"כ סרטני דם), כדוגמת כרומוזום פילדלפיה ב- CML. אולם עבור מרבית ההפרעות הכרומוזומליות, במרבית סוגי הסרטן ובעיקר כאלה של רקמה מוצקה (solid), תדירות ההופעה היא יותר נמוכה ולכן מידת חשיבותן לתהליך הסרטני פחות ברורה. ניתוח סטטיסטי של דגימות סרטן רבות יכול לאתר קשרים משמעותיים המערבים הפרעות כרומוזומליות מסויימות ובכך לרמוז על משמעותן ותפקידן בתהליך הסרטני.

הפרעות כרומוזומליות בסרטן

תא סרטני הוא תוצאה של תהליך התפתחותי במהלכו גנום של תא נורמלי צובר שינויים המקנים לו יכולת בלתי נשלטת להתרבות. מחלת הסרטן קשורה באופן הדוק לחוסר יציבות כרומוזומלית המתבטאת, בין היתר, בארועי רה-ארגון גנומיים המובילים להפרעות כרומוזומליות במרבית תאי הסרטן. ההשערה שהפרעות כרומוזומליות תורמות להתפתחות מחלת הסרטן הועלתה כבר בשנת 1914 ע"י בוברי [21], אולם רק כחמישים שנה לאחר מכן הגיעה פריצת הדרך שאפשרה את הוכחתה של השערה זו, והיא גילויו של "כרומוזום פילדלפיה". כרומוזום פילדלפיה, שנתגלה ע"י נוואל והנגרפורד ב-1960 [67], הוא כרומוזום בעל מבנה לא תקין שנמצא בדגימות תאים סרטניים של כ-95% מהחולים בסרטן דם מסוג chronic mylogenous leukemia (CML). כרומוזום זה הוא תוצאה של פעולת שחלוף הדדית בין כרומוזומים 9 ו-22 [87] שמובילה ליצירתו של גן היתוך (fusion gene) בשם BCR-ABL [31]. גן זה הוכח כבעל תרומה לתהליך הסרטני ולפיכך הפך ליעד חשוב לפיתוח תרופה למחלת הסרטן. בסוף שנות ה-90 פותחה תרופת גליבק (Imatinib/Gleevec Glivec/) המונעת את פעולתו של הגן BCR-ABL [34] ואשר משמשת כיום לטיפול בחולי סרטן מסוג CML.

מאז גילויו של כרומוזום פילדלפיה בשנות השישים של המאה הקודמת, נערכו מחקרים רבים אודות הפרעות כרומוזומליות בגנומים סרטניים. זיהוי הפרעות כרומוזומליות מתבצע לרוב באמצעות צביעת הכרומוזומים בטכניקות שונות ובחינת התוצאה המתקבלת תחת מיקרוסקופ. תיאור מבנה הגנום, עם כל ההפרעות הכרומוזומליות שנצפו בו, נקרא קריוטיפ, והוא נתון לרוב ברזולוציה נמוכה בה כל נקודת ציון כרומוזומלית ממופה למקטע רציף בעל 5-10 מליוני בסיסים. שיטות מתקדמות יותר המאפשרות זיהוי עדין יותר של הפרעות כרומוזומליות, כגון array-CGH [96], מספקות מידע רק לגבי שינויים כמותיים (מחיקה / שכפול) ולא לגבי שינויי סדר של מקטעים, כמו אלו הנגרמים עקב פעולות שחלוף והיפוך. השימוש בקריוטיפים הוא נפוץ למדי, במיוחד בבתי חולים, שם הוא מהווה נדבך חשוב בתהליך האבחון, התאמת הטיפול, וחיזוי התוצאה בחולי סרטן [63]. כתוצאה מכך, קריוטיפים מהווים כיום את עיקר המידע הקיים אודות הפרעות כרומוזומליות בסרטן. מצבור הקריוטיפים הסרטניים הגדול ביותר הוא בסיס הנתונים של מיטלמן [62], עם כמעט 60,000 קריוטיפים סרטניים אשר נאספו מהספרות המדעית.

הפרעות כרומוזומליות מסווגות כמספריות (numerical) או מבניות (structural). הפרעות מספריות מתבטאות בחוסר או עודף של כרומוזומים שלמים מסויימים, תופעה הנקראת הנקראת אנאופלואידיה (aneuploidy) ואשר נצפית במרבית תאי הסרטן [113]. אנאופלואידיה נגרמת עקב כישלון בחלוקת הכרומוזומים המשוכפלים בזמן חלוקת התא: תא בת אחד מקבל שני עותקים של אותו כרומוזום (עודף) בעוד תא הבת השני לא מקבל אף עותק (חוסר). הפרעות מבניות מתייחסות לכרומוזומים בעלי מבנה לא תקין ומיוחסות לתיקון שגוי של שבר דו-גדילי (שד"ג) [double strand break (DSB)] בגנום, אשר ברמת ההפשטה ניתן להצגה כשבירה של כרומוזום לשניים. שד"גים מהווים חבלה מסוכנת ביותר לשלמות הגנום ולפיכך התפתחו מנגנונים לתיקונם. קצה שבור של כרומוזום נוטה להידבק

ביותר שמסדרת, או ממיינת, את גנום המקור "המעורבל" לידי גנום המטרה ה"מסודר" (או "ממוין").

שתי פעולות רה-ארגון גנומי הנפוצות ביותר אצל יונקים הן היפוך (reversal) ושחלוף (translocation). הפעולה הראשונה מתייחסת להיפוך של מקטע רציף של גנים בתוך כרומוזום. הפעולה השנייה מתייחסת לשבירה של שני כרומוזום לארבעה מקטעים והחלפתם בין שני הכרומוזומים כך שהאיחוד מתבצע רק בין קצוות שבורים. בהינתן שבירה מסויימת של שני כרומוזומים, ישנן שתי אפשרויות לשחלוף: החלפת שני מקטעי רישא (רישא-רישא), והחלפה מקטע רישא של כרומוזום אחד במקטע סיפא של כרומוזום שני (רישא-סיפא). פעולת שחלוף הדדית (reciprocal) היא פעולת שחלוף המערבת ארבעה מקטעים לא ריקים.

בעיית מיון ע"י פעולות היפוך - מע"ה (sorting by reversals - SBR) ובעיית מיון ע"י פעולות שחלוף –מע"ש (sorting by translocations - SBT) הן שתי נגזרות שונות של בעיית המיון הגנומי המתירה פעולת רה-ארגון יחידה: היפוך או שחלוף בהתאמה. מכיוון שפעולת היפוך פועלת על כרומוזום בודד ניתן להניח שכל אחד מהגנומים לבעית מע"ה מכיל כרומוזום יחיד (גנום חד-כרומוזומלי). כנגד זה, כל אחד מהגנומים לבעיית מע"ש חייב להכיל מספר כרומוזומים (גנום רב-כרומוזומים). בעיית מע"ה נחקרה רבות [51, 9, 41, 15, 49, 11, 50, 103, 104, 101, 7] וניתנת לפתרון בזמן תת-ריבועי [103, 104, 101]. גם בעיית מע"ש נלמדה בעבר וקיים עבורה פתרון בזמן פולינומי [39, 14].

צנטרומרים

כל כרומוזום מכיל אזור מסוים הנקרא צנטרומר אשר לו תפקיד בעל חשיבות עליונה בחלוקת הכרומוזומים המשוכפלים במהלך חלוקת התא. כרומוזום חסר צנטרומר יעלם קרוב לוודאי במהלך חלוקות התא הבאות. לפיכך, תרחיש פעולות רה-ארגון המשמר צנטרומר בכל כרומוזום הוא סביר יותר מבחינה ביולוגית מאשר תרחיש המערב גנומי ביניים בעלי כרומוזומים חסרי צנטרומר. עד כה, כל העבודות החישוביות שעסקו בפעולות רה-ארגון גנומי התעלמו מקיומם וחשיבותם של צנטרומרים, ומכאן שהאלגוריתמים הקיימים למיון גנומים עלולים לייצר תרחישים המערבים כרומוזומים חסרי צנטרומר. הסיבה המרכזית להעלמת עיין זו מעצם קיומם של הצנטרומרים טמונה בכך שאין מידע לגבי התאמה בין צנטרומרים בגנומים שונים, כפי שיש עבור גנים. בעוד שההתאמה בין גנים מתבססת על דמיון בין רצפי הבסיסים שלהם, רצף הבסיסים של הצנטרומרים אינו ידוע וזאת מכיוון שהם מורכבים מחזרות רבות אשר שיטות הריצוף הקיימות לא מסוגלות לפענח. המידע הנתון לנו לגבי הצנטרומרים מסתכם לפיכך רק במיקומם היחסי לשאר הגנים בכל כרומוזום.

(הומולוגים) בין רצפי גנומים שונים, כגון אלו המתוארות ב [82, 30, 102], הוביל למיפוי עדין יותר של מקטעים שמורים, המאפשר שחזור מדויק יותר של מאורעות רה-ארגון.

בעיית המיון הגנומי

המחקר החישובי של פעולות רה-ארגון גנומי, אשר הוצג לראשונה ע"י סנקוף ועמיתים [90, 91, 88], מתבסס על ההנחה שהסבר טוב לאבולוציה של גנומים צריך להיות חסכוני ועל כן מחפש נתיבי התפתחות קצרים ביותר. אחת הבעיות הבסיסיות בתחום זה היא בעיית המיון הגנומי (genomic sorting problem), אשר מחפשת סדרת פעולות רה-ארגון קצרה ביותר בין שני גנומים נתונים. אורכה של סדרה זו מכונה בשם מרחק הרה-ארגון בין שני הגנומים. בעיית המיון הגנומי היא מקור למספר רב של בעיות קומבינטוריות מעניינות הנבדלות באופן הצגת הגנומים ופעולות הרה-ארגון הגנומיות המותרות. סקירה של בעיות מיון גנומיות שונות והמחקר החישובי שבוצע עבורן מופיעה ב- [18].

במודל שאנו מניחים, גנום מיוצג ע"י אוסף הכרומוזומים שבו, וכל כרומוזום מיוצג ע"י סדרת גנים[1]. כל גן מיוצג ע"י מספר שלם המזהה אותו, וכיוונו בכרומוזום מיוצג ע"י סימן: חיובי (קדימה) או שלילי (אחורה). היפוך של סדרת גנים משמעו היפוך סדר הגנים בסדרה יחד עם סימניהם (חיובי לשלילי, ושלילי לחיובי). היפוך של כרומוזום, כפי שנאמר לעיל, אינו משנה את הכרומוזום המיוצג אלא מסתכם במעבר לייצוג שקול אחר של אותו הכרומוזום. בנוסף, אנו מניחים את הנחות הבאות על שני הגנומים הנתונים:

1. שני הגנומים חולקים את אותה קבוצת הגנים. יהי n מספר הגנים בקבוצה זו. נייצג קבוצה זאת ע"י {1,2,...,n}.
2. כל גן הוא יחודי בגנום. במילים אחרות, כל אחד משני הגנומים מורכב מ-n גנים שונים.

שתי הנחות אלו גוררות שכל גנום ניתן לייצוג כתמורה מקוטעת של המספרים {1,...,n}, כאשר לכל מספר יש גם סימן, -/+, המייצג את כיוונו, וכל מקטע של התמורה מייצג כרומוזום. להלן דוגמא לגנום בעל שני כרומוזומים ושמונה גנים (סימני "+" הושמטו):

{(1, -3, -2, 4, -7, 8), (6, 5)}

לצורך פשטות ההצגה וללא הגבלת הכלליות ניתן להניח שגנום אחד, אותו נכנה גנום המטרה, מורכב מסדרות עולות של מספרים חיוביים עוקבים, כמו למשל

{(1, 2, 3, 4, 5, 6), (7,8)}

והגנום השני, אשר יכונה גנום המקור, הוא תמורה מקוטעת כלשהי של מספרים חיובים ושליליים[2]. באופן זה ניתן להציג את בעיית המיון כמציאת סדרה פעולות רה-ארגון קצרה

---

[1] לצורך פשטות ההצגה אנו משתמשים במינוח "גן", אף על פי שבמודל שלנו גן מייצג רצף בסיסים מסוים בכרומוזום אשר אינו בהכרח גן אמיתי.

[2] הנחה זו אפשרית מכיוון שהמידע הנחוץ לבעיית המיון הוא המיפוי בין הגנים והיחס בין כיווניהם של גנים זהים. לפיכך ניתן לקבוע באופן שרירותי את מספרי הגנים וכיווניהם בגנום אחר (גנום המטרה), ולשנות בהתאם את מספר הגנים וכיווניהם בגנום השני (גנום המקור).

תקציר

רקע כללי

הגנום הינו כלל החומר התורשתי של אורגניזם מסויים ומכיל את אוסף ההוראות הגנטיות הדרושות להתפתחותו ולתפקודו של אותו אורגניזם. כל גנום מאוחסן בדנ"א (DNA) באמצעות קידוד של ארבע אותיות {A,C,G,T}. הנקראות בסיסים. רצף הבסיסים של גנום מתחלק ליחידות רציפות הנקראות כרומוזומים. גן הינו סדרה רציפה של בסיסים בכרומוזום המייצגת תכונה גנטית מסויימת. באופן גס ניתן להציג כל כרומוזום כסדרה של גנים, כאשר לכל גן יש כיוון (קדימה או אחורה) לאורך הכרומוזום בו הוא נמצא. הכרומוזומים עצמם הינם חסרי כיוון, ועל כן כל כרומוזום ניתן להצגה בשני אופנים שקולים, כאשר האחד מתקבל מהשני ע"י היפוך סדרת הגנים וכיווניהם.

גנומים מתפתחים הן באמצעות שינויים מקומיים, אשר מחליפים / מוחקים / מוסיפים מספר בסיסים בודדים, או כללים, אשר מתבטאים במחיקה, שכפול או הזזה של מקטעים גדולים של דנ"א. פעולות אלו, המשנות את מבנה הגנום, נקראות פעולות רה-ארגון גנומי (genome rearrangements) והן העומדות במרכז המחקר המתואר בתיזה זו. לפעולות רה-ארגון גנומי יש השפעה מכרעת, הן ברמת האורגניזם, שם הן עלולות לגרום לתוצאות הרסניות כמו הפלות ופיגור שכלי [55], והן ברמת התא, שם הן מקושרות להתפתחות מחלת הסרטן כפי שיפורט בהמשך.

בכדי ללמוד על האופן בו גנומים התפתחו עלינו להשוות ביניהם. באופן כללי, גנומים של מינים קרובים הם בעלי תוכן דומה מאוד. לדוגמא, למעלה מ-90% מהגנומים של האדם והעכבר ניתנים לחלוקה למקטעים בהם הגנים והסדר בו הם מופיעים זהה בין המינים [112]. הסדור השונה של מקטעים שמורים אלו בשני הגנומים הוא תוצאה של פעולות רה-ארגון גנומי שהתרחשו בשושלות האדם והעכבר לאחר התפצלותן, לפני כ-65 מיליון שנים [112]. מספר המקטעים השמורים בין האדם לעכבר (מאות ספורות) מעיד כי תדירות פעולות הרה-ארגון הגנומיות בשושלות אלו היא יחסית נמוכה: פעולות ספורות בכל מליון שנים. מכיוון שגנומים של מינים קרובים נבדלים במספר לא רב יחסית של פעולות רה-ארגון, ישנה אפשרות עקרונית לשחזר פעולות אלו.

קיומן של פעולות רה-ארגון גנומי במהלך האבולוציה של מינים ידועה כבר למעלה מ-70 שנה. בשנות השלושים של המאה הקודמת, סטורטוונט ודובז'נסקי [98] הדגימו מאורעות היפוך בין גנומים של זנים שונים של זבובי פירות (drosophila). כחמישים שנה לאחר מכן, ג'פרי פלמר וביולוגים נוספים גילו כי גנומים של זני צמחים קרובים מסויימים חלקו את אותם גנים אך נבדלו בסידורם [78, 79, 80, 81, 45]. גילוי זה חשף כי האבולוציה של זנים אלו הונעה על ידי פעולות רה-ארגון גנומי. פיתוחן של שיטות בציטוגנטיקה מולקולרית, ובמיוחד צביעת כרומוזומים השוואתית (comparative chromosome painting , "Zoo-FISH") [92], הובילו למיפויים של מקטעים דנ"א גדולים השמורים בגנומים של למעלה מ-80 מיני יונקים [38]. פיתוחן של שיטות ביואינפורמטיות לזיהוי מקטעים דומים

תמצית

פעולות רה-ארגון גנומי הן מוטציות רחבות היקף בגנום המשפיעות על מבנהו ותוכנו באמצעות הזזתם, שכפולם או מחיקתם של מקטעי דנ"א גדולים. רה-ארגון של הגנום משפיע באופן ישיר על פעילותו והינו בעל תפקיד חשוב במהלך האבולוציה של מינים שונים ובהתפתחותה של מחלת הסרטן. חקר רה-ארגון הגנום עוסק בעיקר בשחזורם של מאורעות רה-ארגון שהתרחשו ובהבנת תרומתם של אלו לתהליך ההתפתחותי של הגנום. אחת הבעיות החישוביות הבסיסיות בחקר רה-ארגון הגנום היא בעיית המיון הגנומי שמטרתה לשחזר רצף פעולות רה-ארגון קצר ביותר בין שני גנומים נתונים. המחקר המתואר בתיזה זו עוסק בפיתוח מודלים מתימטיים לתיאור פעולות רה-ארגון, בניית אלגוריתמים קומבינטוריים לבעיית המיון הגנומי תחת מודלים אלו, וניתוח סטטיסטי של פעולות רה-ארגון משוחזרות.

בראשית עבודת המחקר התמקדנו בשני מודלים מתימטיים קיימים עבור פעולות רה-ארגון במהלך האבולוציה: מיון ע"י היפוכים (מע"ה) ומיון ע"י שחלופים (מע"ש), וחשפנו נקודות דמיון חדשות בין התאוריות המקבילות. בפרט הראינו כי אלגוריתמים קיימים מסויימים המשחזרים רצף פעולות רה-ארגון קצר ביותר עבור בעיית מע"ה, ניתנים להתאמה עבור בעיית מע"ש, וכך שיפרנו את סיבוכיות הזמן הידועה עבור בעיית מע"ש. לאחר מכן, הרחבנו את המודל של מע"ש לכלול גם את הצנטרומרים, שהם אזורים בגנום בעלי תפקיד חיוני בשמירה על שלמותו, ובנינו אלגוריתם פולינומיאלי מדוייק לשחזור רצף קצר ביותר של ארועי רה-ארגון תחת מודל זה.

בהמשך עבודת המחקר התמקדנו ברה-ארגון הגנום בתאים סרטניים. מחקר זה כלל ניתוח של עשרות אלפי קריוטיפים, המתארים גנומים של תאים סרטניים ברזולוציה נמוכה. הגדרנו את בעיית המיון של קריוטיפים סרטניים העוסקת בחיפוש סדרת פעולות רה-ארגון קצרה ביותר שמובילה מקריוטיפ תקין לקריוטיפ סרטני נתון, ובנינו אלגוריתמים לפתרונה תחת מודלים חדשים שפיתחנו. מודל ראשון כלל ארבע פעולות רה-ארגון פשוטות ועבורו בנינו אלגוריתם קירוב-3 תחת תנאים מסויימים אשר התקיימו במרבית הקריוטיפים הסרטניים שנבחנו. אלגוריתם זה ייצר פתרונות מיטביים (קרי קצרים ביותר) על כל 58,464 הקריוטיפים הסרטניים עליהם הופעל. המודל השני כלל 12 פעולות רה-ארגון מורכבות נפוצות ועבורו בנינו אלגוריתם יוריסטי שהצליח בשחזור רצף מאורעות סביר ביותר עבור 98% מהקריוטיפים עליהם הופעל. בהמשך, ערכנו ניתוח סטטיסטי למאורעות רה-ארגון ששוחזרו ע"י אלגוריתם זה. ניתוח זה חשף ממצאים מקוריים אודות ההבדלים ונקודות הדמיון בין סוגי סרטן שונים על סמך ארועי רה-ארגון המופיעים בהם, וכן אודות ארועי רה-ארגון הנוטים להופיע יחדיו. תוצאות אילו הצביעו על תופעות משמעותיות בסרטן המערבות פעולות רה-ארגון גנומי ורומזות על מכניזמים המעורבים בתהליך התפתחותן.

אוניברסיטת תל-אביב TEL AVIV UNIVERSITY

הפקולטה למדעים מדויקים ע"ש ריימונד ובברלי סאקלר

בית הספר למדעי המחשב ע"ש בלבטניק

בעיות חישוביות ברה-ארגון גנומי:

מאבולוציה לסרטן

חיבור לשם קבלת תואר "דוקטור לפילוסופיה"

מאת **מיכל עוזרי-פלטו**

בהנחייתו של פרופ' **רון שמיר**

הוגש לסנאט של אוניברסיטת ת"א

אוקטובר 2009