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.

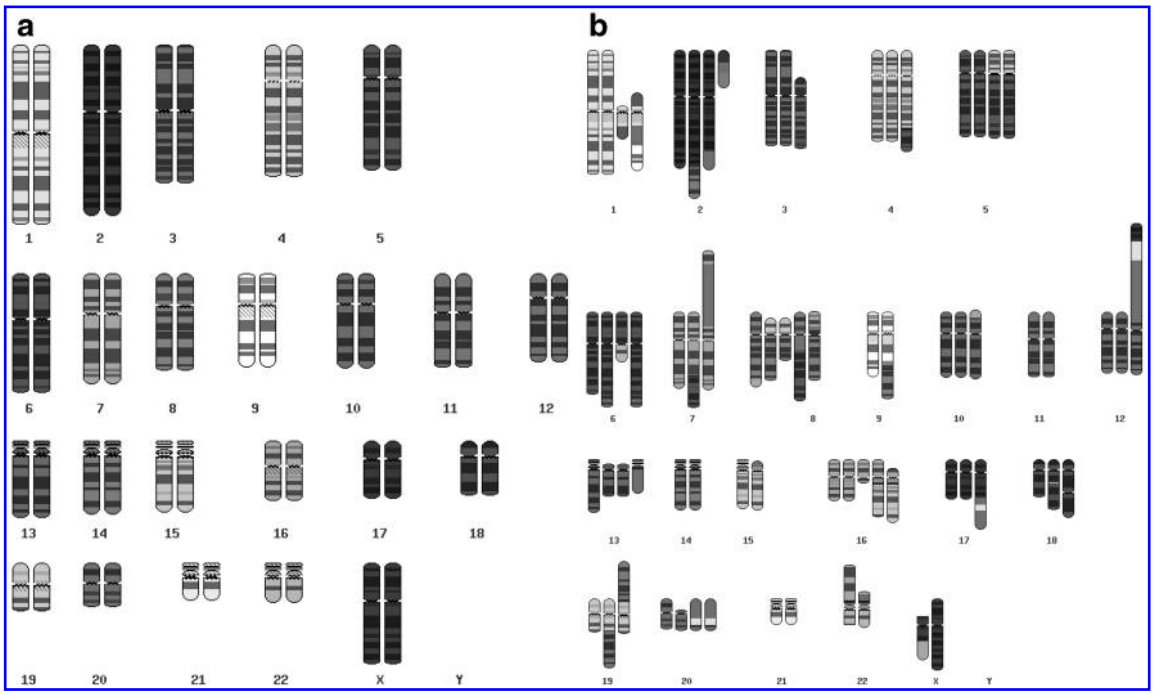


FIG. 1. A schematic view of two real karyotypes: a normal female karyotype (**a**) and the karyotype of MCF-7 breast cancer cell-line (**b**) (NCI, 2001). In the normal karyotype, all chromosomes, except X and Y, appear in two identical copies, and each chromosome has a distinct single color. In the 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

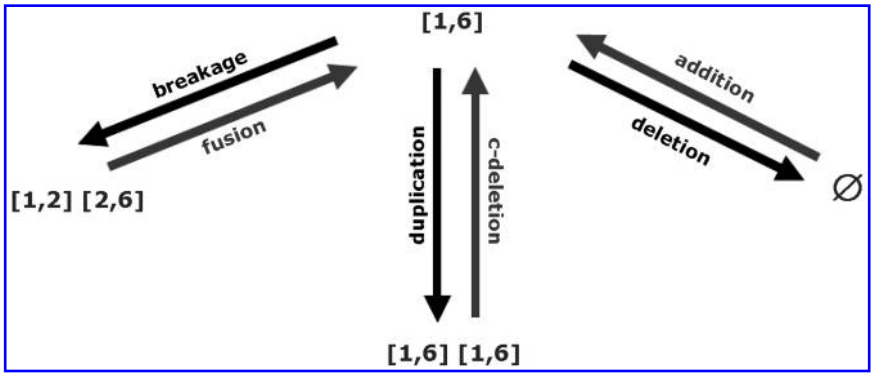


FIG. 2. Illustrations of elementary operations: breakage, fusion, duplication, and deletion. The inverse elementary operations are fusion, breakage, c-deletion, and addition, respectively.

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:

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

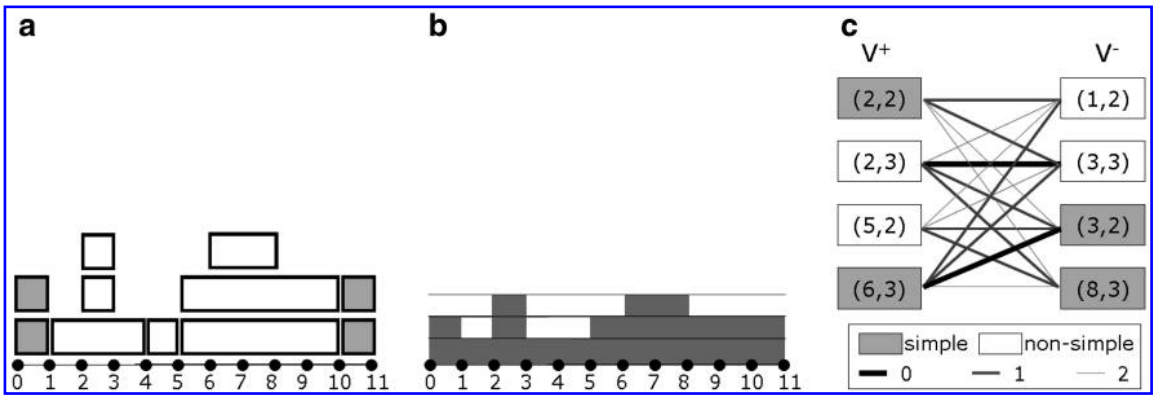


FIG. 4. An example of a cancer karyotype $\hat{K}_{\text{cancer}}$ and its combinatorial parameters. (**a**) The (extended) cancer karyotype is $\hat{K}_{\text{cancer}} = \{[0,1]\times 2, [1,4], [4,5], [5,10]\times 2, [10,11]\times 2, [2,3]\times 2, [6,8]\}$. Here $N=2$, $B=10$. The number of disjoint 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

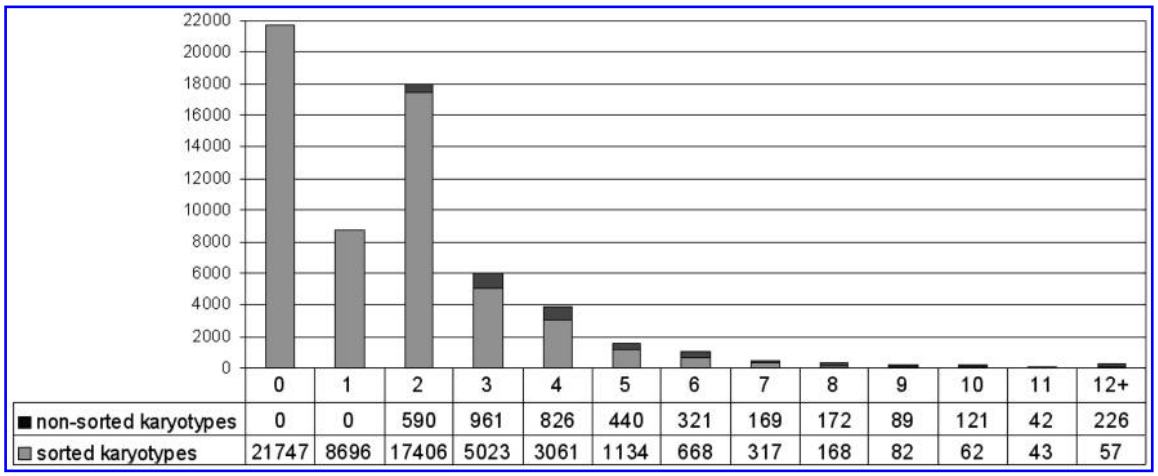
In this section, we present the results of sorting real cancer karyotypes, using Algorithm 1, combined with the improvement heuristic in Procedure 2.

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