

Multi-break rearrangements and chromosomal evolution

Max A. Alekseyev*, Pavel A. Pevzner

Department of Computer Science and Engineering, University of California at San Diego, La Jolla, CA 92093-0114, USA

Abstract

Most genome rearrangements (e.g., reversals and translocations) can be represented as *2-breaks* that break a genome at 2 points and glue the resulting fragments in a new order. Multi-break rearrangements break a genome into multiple fragments and further glue them together in a new order. While multi-break rearrangements were studied in depth for $k = 2$ breaks, the k -break distance problem for arbitrary k remains unsolved. We prove a duality theorem for multi-break distance problem and give a polynomial algorithm for computing this distance.

© 2008 Elsevier B.V. All rights reserved.

Keywords: Multi-break; Reversal; Translocation; Transposition; Genome rearrangement; Breakpoint graph; Genomic distance

1. Introduction

Rearrangements are genomic “earthquakes” that change the chromosomal architectures. The fundamental question in molecular evolution is whether there exist “chromosomal faults” (*rearrangement hot-spots*) where rearrangements are happening over and over again. The *Random Breakage Model* (RBM), proposed by Susumu Ohno in 1970, postulates that rearrangements happen at “random” genomic positions, and thus there are no rearrangement hot-spots in mammalian genomes. RBM was embraced by biologists (due to its prophetic prediction power) and has become the *de facto* theory of chromosome evolution [1,2]. However, Pevzner and Tesler, 2003 [3] recently refuted RBM and suggested an alternative Fragile Breakage Model of chromosome evolution. Murphy et al., 2005 [4] and a variety of other studies further argued for the existence of fragile regions in mammalian genomes [5–10].¹

The standard rearrangement operations (reversals/translocations/fusions/fissions) can be modelled by making 2-breaks in a genome and gluing the resulting fragments in a new order. Most biologists believe that k -break rearrangements are unlikely for $k > 3$ and relatively rare for $k = 3$ (at least in mammalian evolution). Indeed, biophysical limitations and selective constraints are already severe for $k = 2$, let alone for $k > 2$. However, 3-break rearrangements (e.g., transpositions) undoubtedly happen in evolution, although it is still unclear how frequent they are in mammalian evolution. Therefore, it would be useful to generalize the Pevzner–Tesler arguments against RBM for the case of k -breaks (and 3-breaks in particular). Also, in radiation biology, chromosome aberrations for $k > 2$

* Corresponding author.

E-mail addresses: maxal@cs.ucsd.edu (M.A. Alekseyev), ppevzner@cs.ucsd.edu (P.A. Pevzner).

¹ While the rebuttal of RBM caused a controversy [11], recent study [12] revealed an important flaw in the arguments supporting RBM [11].

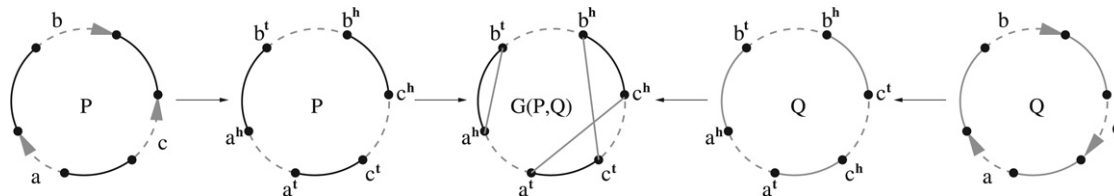


Fig. 1. The breakpoint graph $G(P, Q)$ of unichromosomal genomes $P = +a + b - c$ and $Q = +a + b + c$ represented as a black-obverse cycle and a gray-obverse cycle correspondingly.

(indicative of chromosome damage rather than evolutionary viable variations) may be more common, e.g., complex rearrangements in irradiated human lymphocytes [13–16]. Thus, both the analysis of rearrangement hot-spots and radiation/cancer biology call for studies of k -break rearrangements for $k > 2$.

In this paper we initiate studies of k -break rearrangements. We prove a duality theorem for the k -break distance between genomes with n genes that shows how to compute it. In particular, we present a dynamic programming algorithm with the running time $O(n^{\lfloor k/2 \rfloor - 2}) + O(n)$ that is practical for small values of k . We also show how one can compute the k -break distance in linear time in n for an arbitrary k that requires preliminary computations that are exponential in k but independent of n . The applications of these results for studying rearrangements between human and mouse genomes and for analyzing “FBM vs. RBM” alternative is described in [17].

2. Multi-break distance problem

We will find it convenient to represent a circular² chromosome with genes x_1, \dots, x_n as a cycle (Fig. 1) composed of n directed labelled edges (corresponding to genes) and n undirected unlabelled edges (connecting adjacent genes). The directions of the edges correspond to *signs* (strand) of the genes. We label the tail and head of a directed edge x_i as x_i^t and x_i^h respectively. Vertex x_i^t is called the *obverse* of vertex x_i^h , and vice versa. Vertices in a chromosome connected by an undirected edge are called *adjacent*. We represent a genome as a collection of disjoint cycles (chromosomes) with edges of two *alternating* colors: one color (black) reserved for undirected edges and the other (obverse³) color reserved for directed edges. We do not explicitly show the directions of edges since they are defined by superscripts “*t*” and “*h*” (Fig. 1).

Let P be a genome represented as a collection of alternating black-obverse cycles (a cycle is alternating if colors of its edges alternate). For any two black edges (u, v) and (x, y) in the genome (graph) P we define a *2-break* rearrangement as replacement of these edges with either a pair of edges (u, x) , (v, y) , or a pair of edges (u, y) , (v, x) (Fig. 2). 2-breaks correspond to standard rearrangement operations of reversals (Fig. 2a), fissions (Fig. 2b), or fusions/translocations⁴ (Fig. 2c). 2-break rearrangements can be generalized as follows. Given k black edges forming a matching on $2k$ vertices, define a *k-break* as replacement of these edges with a set of k black edges forming another matching in on the same set of $2k$ vertices. Note that a 2-break is a particular case of a 3-break (as well as of a k -break for $k > 3$), in which case only two edges are replaced and the third one remains the same.

Let P and Q be two genomes on the same set of genes \mathcal{G} . The *breakpoint graph* $G(P, Q)$ is defined on the set of vertices $V = \{x^t, x^h \mid x \in \mathcal{G}\}$ with edges of three colors: obverse, black, and gray (Fig. 1). Edges of each color form a matching on V : *obverse matching* (pairs of obverse vertices), *black matching* (adjacent vertices in P), and *gray matching* (adjacent vertices in Q). Every pair of matchings forms a collection of alternating cycles in $G(P, Q)$, called *black-gray*, *black-obverse*, and *gray-obverse* cycles respectively. The chromosomes of the genome P (resp. Q) can be read along black-obverse (resp. gray-obverse) cycles. The black-gray cycles in the breakpoint graph play an important role in analyzing rearrangements [18] (see Chapter 10 of [19] for background information on genome rearrangements).

Every k -break in the genome P corresponds to a transformation of the breakpoint graph $G(P, Q)$. Since the breakpoint graph of two identical genomes is a collection of *trivial* black-gray cycles of length 2 (the *identity*

² In this paper we deal with circular chromosomes. Extension of the present results to the case of linear chromosomes is described in [40].

³ We have chosen rather unusual name “obverse” for the color to be consistent with previous papers on genome rearrangements.

⁴ This definition of elementary rearrangement operations follows the standard definitions of reversals, translocations, fissions, and fusions for the case of circular chromosomes. For circular chromosomes fusions and translocations are not distinguishable.

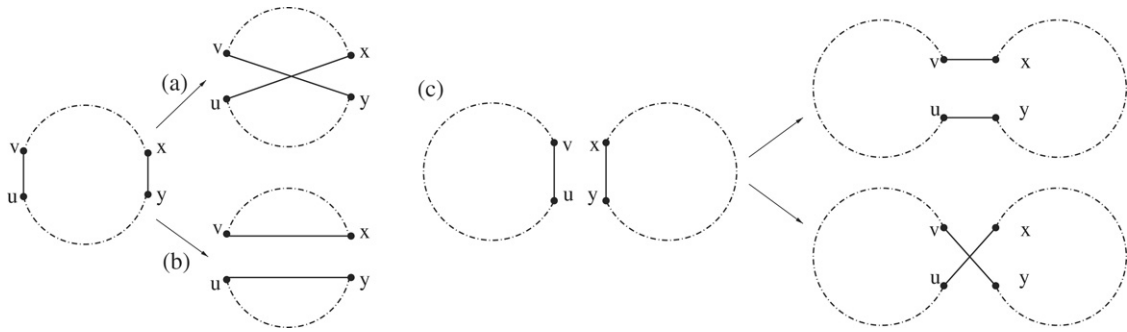


Fig. 2. A 2-break on edges (u, v) and (x, y) corresponding to: (a) Reversal: the edges belong to the same black–obverse cycle that is rearranged after 2-break; (b) Fission: the edges belong to the same black–obverse cycle that is split by 2-break; (c) Translocation/fusion: the edges belong to different black–obverse cycles that are joined by 2-break.

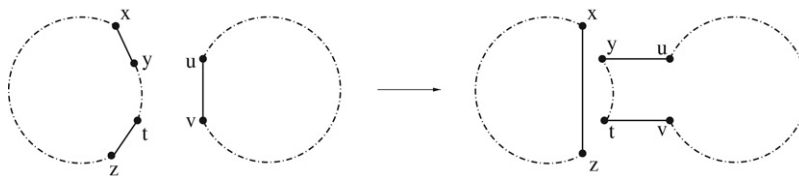


Fig. 3. An example of a 3-break on edges (u, v) , (x, y) and (z, t) corresponding to transposition of a segment $y \dots t$ from one chromosome to another. A transposition cuts off a segment of one chromosome and inserts it into the same or another chromosome. A transposition of a segment $\pi_i \pi_{i+1} \dots \pi_j$ of a chromosome $\pi_1 \pi_2 \dots \pi_i \pi_{i+1} \dots \pi_j \dots \pi_k \pi_{k+1} \dots \pi_m$ into a position k of the same chromosome results a chromosome $\pi_1 \pi_2 \dots \pi_{i-1} \pi_{j+1} \dots \pi_k \pi_i \pi_{i+1} \dots \pi_j \pi_{k+1} \dots \pi_m$. For chromosomes $\pi = \pi_1 \pi_2 \dots \pi_i \pi_{i+1} \dots \pi_j \dots \pi_m$ and $\sigma = \sigma_1 \sigma_2 \dots \sigma_n$ a transposition of a segment $\pi_i \pi_{i+1} \dots \pi_j$ of chromosome π into a position k in the chromosome σ results in chromosomes $\pi_1 \pi_2 \dots \pi_{i-1} \pi_{j+1} \pi_{j+2} \dots \pi_m$ and $\sigma_1 \sigma_2 \dots \sigma_{k-1} \pi_i \pi_{i+1} \dots \pi_j \sigma_k \dots \sigma_n$.

breakpoint graph), the problem of transforming the genome P into the genome Q by k -breaks can be formulated as the problem of transforming the breakpoint graph $G(P, Q)$ into the identity breakpoint graph. This is equivalent to the following problem:

k -break distance problem. Given two perfect matchings (black and gray) in a graph, find a shortest series of k -breaks that transforms one matching into the other.

In difference from the Genomic Distance Problem [20–22] (for linear multichromosomal genomes), the 2-break distance problem for circular multichromosomal genomes is trivial (compare to [23]). For the sake of completeness, we reproduce a simple theorem for computing 2-break distance from [24]:

Theorem 1. The 2-break distance between a black matching P and a gray matching Q is $|P| - c(P, Q)$ where $c(P, Q)$ is the number of black–gray cycles in $G(P, Q)$.

Proof. It is easy to see that every non-trivial black–gray cycle can be split into two by a 2-break. Since no 2-break can increase the number of black–gray cycles by more than 1, the 2-break distance between P and Q is $|P| - c(P, Q)$. \square

While 2-breaks correspond to standard rearrangements, 3-breaks add transposition-like operations (transpositions and inverted transpositions) as well as 3-way fissions to the set of rearrangements (Fig. 3). In difference from standard rearrangements (modelled as 2-breaks), transpositions introduce 3 breaks in the genome, making them notoriously difficult to analyze. Computing the minimum number of transpositions transforming one genome into another is called *sorting by transpositions*. After Bafna and Pevzner, 1995 [25] gave a first 1.5-approximation algorithm for sorting by transpositions, a number of faster algorithms with the same approximation ratio were proposed [26–28] culminating in a recent 1.375-approximation algorithm by Elias and Hartman [29]. A number of researchers considered transpositions in conjunction with other rearrangement operations [30–36]. The complexity of sorting by transpositions remains unknown.

Let $c^{\text{odd}}(P, Q)$ be the number of black–gray cycles in the breakpoint graph $G(P, Q)$ with an odd number of black edges (*odd cycles*). The 3-break distance theorem has a simple proof that is very similar to the arguments in [25]:

Theorem 2. *The 3-break distance between a black matching P and a gray matching Q is $\frac{|P| - c^{\text{odd}}(P, Q)}{2}$.*

Proof. A trivial black–gray cycle is a cycle with a single black edge. If $Q = P$, the breakpoint graph $G(P, Q)$ is a set of $|P|$ trivial cycles that are odd cycles (each with a single black edge). It is easy to see that as soon as there is a non-trivial black–gray odd cycle, it can be split into 3 odd cycles by a 3-break, thus increasing the number of odd cycles by 2. On the other hand, if there exists a black–gray even cycle, it can be split into two odd cycles, thus again increasing the number of odd cycles by 2. Since no 3-break can increase the number of black–gray cycles by more than 2, the 3-break distance is $\frac{|P| - c^{\text{odd}}(P, Q)}{2}$. \square

Alekseyev and Pevzner [24] further illustrate the theoretical advantages of considering the 3-break distance (as compared to the transposition distance) by showing that some very difficult problems can be solved if one moves from transpositions to 3-breaks.

Below we prove the duality theorem for the k -break distance for an arbitrary k . A black–gray cycle is called an i_k -cycle if it has i modulo $k - 1$ black edges. A subset of cycles in a breakpoint graph $G(P, Q)$ is called *breakable* if the total number of black edges in these cycles equals 1 modulo $k - 1$. Let $s_k(P, Q)$ be the maximum number of disjoint breakable subsets in $G(P, Q)$. For example, for $k = 3$, every odd cycle forms a breakable subset and $s_3(P, Q) = c^{\text{odd}}(P, Q)$. Let $c_k^i(P, Q)$ be the number of black–gray i_k -cycles in $G(P, Q)$. For $k = 4$, every 1_4 -cycle forms a breakable subset and every pair of 2_4 -cycles forms a breakable subset, implying that $s_4(P, Q) = c_4^1(P, Q) + \lfloor c_4^2(P, Q)/2 \rfloor$. Below we prove that the k -break distance is $d_k(P, Q) = \lceil \frac{|P| - s_k(P, Q)}{k-1} \rceil$.

We introduce a few definitions. A k -break β and a cycle c are called *compatible* if β either does not use edges of c or uses all its black edges. Otherwise β and c are called *incompatible*. Given a k -break β , we define $\text{def}(\beta)$ as the number of cycles in $G(P, Q)$ that are incompatible with β . Obviously, a k -break β may increase the number of trivial cycles by at most $k - \text{def}(\beta)$. A k -break β is called *optimal* if it is compatible with all cycles in $G(P, Q)$ and if it increases the number of trivial cycles by k . A k -break β with $\text{def}(\beta) = 1$ is called *semi-optimal* if it increases the number of trivial cycles by $k - 1$.

Lemma 3. *A set S of non-trivial black–gray cycles with m black edges can be transformed into m trivial cycles with $\frac{m-1}{k-1}$ k -breaks if S is breakable and with $\lceil \frac{m}{k-1} \rceil$ k -breaks otherwise.*

Proof. We first prove that any set S of non-trivial black–gray cycles with m black edges can be transformed into m trivial cycles with a series of $\lceil \frac{m}{k-1} \rceil$ k -breaks. It is easy to see that if $m > k$ then either an optimal or a semi-optimal k -break exists. Indeed, let c_1, \dots, c_t be a set of non-trivial cycles in S containing at least k black edges while c_1, \dots, c_{t-1} contains less than k black edges. If c_1, \dots, c_t contain exactly k black edges then there exists an optimal k -break using all black edges of these cycles. If c_1, \dots, c_t contain more than k black edges then there exists a semi-optimal k -break using all black edges of c_1, \dots, c_{t-1} and some black edges of c_t . In either case, the number of trivial cycles is increasing by at least $k - 1$ with every k -break. To complete the proof (for non-breakable sets) it is sufficient to notice that every set of cycles with k or less black edges can be transformed into trivial cycles by a single k -break.

We showed above how to transform a set S into m trivial cycles with a series of optimal and semi-optimal k -breaks (with a possible exception of the last k -break). If one of these k -breaks is optimal, the bound $\lceil \frac{m}{k-1} \rceil$ turns into $\lceil \frac{m-1}{k-1} \rceil$ (since each optimal k -break creates k trivial cycles as compared to $k - 1$ trivial cycles for semi-optimal k -breaks). It is easy to see that for a breakable set S there exists at least one optimal k -break in the series. \square

Theorem 4. *The k -break distance between a black matching P and a gray matching Q is $\lceil \frac{|P| - s_k(P, Q)}{k-1} \rceil$.*

Proof. We first prove that there exists a series of $\lceil \frac{|P| - s_k(P, Q)}{k-1} \rceil$ k -breaks transforming $G(P, Q)$ into a set of trivial cycles. Let S be a collection of $s_k(P, Q)$ disjoint breakable subsets of black–gray cycles in $G(P, Q)$ and M be the total number of black edges in S . Lemma 3 implies that every breakable set with m black edges can be decomposed into trivial cycles with $\frac{m-1}{k-1}$ k -breaks. Therefore, all $s_k(P, Q)$ breakable sets from S can be decomposed into M trivial

cycles with $\frac{M-s_k(P, Q)}{k-1}$ k -breaks. Lemma 3 also implies that all remaining cycles (i.e., cycles that do not belong to elements of \mathcal{S}) with $|P| - M$ black edges in total can be broken into trivial cycles by $\lceil \frac{|P|-M}{k-1} \rceil$ k -breaks. Therefore, all cycles can be transformed into trivial cycles by $\frac{M-s_k(P, Q)}{k-1} + \lceil \frac{|P|-M}{k-1} \rceil = \lceil \frac{|P|-s_k(P, Q)}{k-1} \rceil$ k -breaks.

We now prove that a k -break on $G(P, Q)$ can reduce the value of $\lceil \frac{|P|-s_k(P, Q)}{k-1} \rceil$ by at most 1, or equivalently, that every k -break can increase $s_k(P, Q)$ by at most $k - 1$. Every k -break can create at most k “new” cycles, implying that $s_k(P, Q)$ can increase by at most k . Assume that a k -break β increases $s_k(P, Q)$ by k . Let \mathcal{S} be a maximum set of disjoint breakable subsets of black–gray cycles after performing the k -break β (i.e., $|\mathcal{S}| = s_k(P, Q) + k$). The k -break β may be viewed as a replacement of some “old” cycles c'_1, \dots, c'_t in $G(P, Q)$ with k “new” cycles c_1, \dots, c_k . Therefore, the total number of black edges in these cycles is the same: $\sum_{i=1}^k b(c_i) = \sum_{i=1}^t b(c'_i)$ where $b(\cdot)$ denotes the total number of black edges in a subgraph g .

Note that if for each “new” cycle c_i ($i = 1, \dots, k$) we remove from \mathcal{S} a breakable subset contains c_i , then the remaining breakable subsets will contain only cycles from $G(P, Q)$, implying that the number of remaining subsets cannot exceed $s_k(P, Q)$. Therefore, each “new” cycle c_i ($i = 1, \dots, k$) must belong to a distinct breakable subset $\mathcal{B}_i \in \mathcal{S}$ with $e_i + b(c_i)$ black edges in total, where $e_i = b(\mathcal{B}_i \setminus \{c_i\})$. Since $e_i + b(c_i)$ equals 1 modulo $k - 1$, $\sum_{i=1}^k e_i + \sum_{i=1}^t b(c'_i) = \sum_{i=1}^k e_i + b(c_i)$ equals 1 modulo $k - 1$ as well. Therefore, the cycles c'_1, \dots, c'_t together with the cycles from all $\mathcal{B}_i \setminus \{c_i\}$ form a breakable subset \mathcal{B}' . Then the set $(\mathcal{S} \setminus \{\mathcal{B}_1, \dots, \mathcal{B}_k\}) \cup \{\mathcal{B}'\}$ consists of $s_k(P, Q) + 1$ disjoint breakable subsets of black–gray cycles in $G(P, Q)$, a contradiction to the definition of $s_k(P, Q)$. It proves that every k -break can increase $s_k(P, Q)$ by at most $k - 1$. \square

Theorem 4 and the formula for $s_4(P, Q)$ imply a formula for the 4-break distance.

Corollary 5. *The 4-break distance between a black matching P and a gray matching Q is*

$$d_4(P, Q) = \left\lceil \frac{|P| - c_4^1(P, Q) - \lfloor c_4^2(P, Q)/2 \rfloor}{3} \right\rceil.$$

Similarly, one can derive a formula for the 5-break distance, which we state below without a proof.

Corollary 6. *The 5-break distance between a black matching P and a gray matching Q is $d_5(P, Q) = \left\lceil \frac{|P|-s_5(P, Q)}{4} \right\rceil$ where*

$$s_5(P, Q) = c_5^1(P, Q) + \min\{c_5^2(P, Q), c_5^3(P, Q)\} + \left\lceil \frac{\max\{0, c_5^3(P, Q) - c_5^2(P, Q)\}}{3} \right\rceil.$$

For $k > 5$, a formula for the k -break distance becomes more complicated, e.g., $d_6(P, Q) = \left\lceil \frac{|P|-s_6(P, Q)}{5} \right\rceil$ where

$$s_6(P, Q) = c_6^1(P, Q) + \left\lceil \frac{c_6^3(P, Q)}{2} \right\rceil + \min\{c_6^2(P, Q), c_6^4(P, Q)\} + \left\lceil \frac{\max\{0, c_6^2(P, Q) - c_6^4(P, Q)\}}{3} \right\rceil + \left\lceil \frac{\max\{0, c_6^4(P, Q) - c_6^2(P, Q)\}}{4} \right\rceil + \delta$$

and δ is either 0 or 1, and $\delta = 1$ iff (i) $c_6^3(P, Q)$ is odd, (ii) $c_6^4(P, Q) > c_6^2(P, Q)$, and (iii) $c_6^4(P, Q) - c_6^2(P, Q)$ equals 2 or 3 modulo 4.

From the algorithmic perspective, while the k -break distance between genomes with n genes can be computed in $O(n)$ time for small k (e.g., for $k \leq 10$), it is unclear whether one can compute $d_k(P, Q)$ in linear time for arbitrary k . In the next Section we address this problem by establishing the relationship between the k -break distance and the Gröbner basis of an appropriately constructed polynomial ideal.

3. Algorithms for computing multi-break distance

In this section we present two approaches to computing the k -break distance between genomes with n genes. We start with a dynamic programming algorithm with the running time $O(n^{\lfloor k/2 \rfloor - 2}) + O(n)$ that is practical for small values of k . We further show how one can derive closed-form formulas for the k -break distance via computing the set of so-called extremal breakable vectors. While these formulas lead to linear-time algorithms for a wider range of k , it is not clear how to generalize this approach for an arbitrary k . Finally, we show how to compute the k -break distance in linear time in n for an arbitrary k (with preliminary computations that are exponential in k but independent of n). While the latter algorithm is linear in theory, the high cost of the preliminary computations makes it less practical than the former algorithms.

3.1. Dynamic programming algorithms

First we reformulate the k -break distance as a multi-dimensional packing problem. Since a breakable subset remains breakable after removing all 0_k -cycles, without loss of generality we assume that breakable subsets do not contain 0_k -cycles. Then every breakable subset \mathcal{B} is characterized by a *breakable vector* $v = (v_1, \dots, v_{k-2})$ where v_i is the number of i_k -cycles in \mathcal{B} .

For genomes P and Q , let $c = (c_1, \dots, c_{k-2})$ where $c_i = c_k^i(P, Q)$. Finding $s_k(P, Q)$ amounts to finding the maximum number of breakable vectors v^1, \dots, v^t such that $v^1 + \dots + v^t \leq c$ (component-wise). Note that we can limit our search only to the set V of all *proper* breakable vectors v with $v_j < k - 1$ for all $j = 1, \dots, k - 2$. Since the first coordinate of a proper breakable vector $v = (v_1, \dots, v_{k-2})$ is uniquely defined by the others as $v_1 = 1 - 2 \cdot v_2 - \dots - (k - 2) \cdot v_{k-2} \pmod{(k - 1)}$, the total number of proper breakable vectors is $|V| = (k - 1)^{k-3}$.

For a vector u with $k - 2$ components, define $s(u)$ as the maximum number of elements of V (each element may appear several times) with the sum not exceeding u . Then $s_k(P, Q) = s(c)$. We will use this formula and [Theorem 4](#) to come up with an algorithm for computing the k -break distance for an arbitrary k .

Theorem 7. For genomes P and Q with n genes, $d_k(P, Q)$ can be computed in $O(n^{k-2}) + O(n)$ time.

Proof. It is easy to see that $s(u) = \max_{v \in V, v \leq u} s(u - v) + 1$. This formula leads to a dynamic programming algorithm for computing $s_k(P, Q) = s(c)$ via computing $s(u)$ for all $u \leq c$. We need to fill up a dynamic programming table of size $(c_1 + 1) \times \dots \times (c_{k-2} + 1) = O((n/k)^{k-2})$. Note that the time-complexity of computing each $s(u)$ depends on k but not on n . Therefore, the total time to compute $s_k(P, Q)$ (and $d_k(P, Q)$) is $O(n^{k-2}) + O(n)$, where the term $O(n)$ accounts for time needed to construct the breakpoint graph $G(P, Q)$ and to compute the vector c . \square

The following theorem describes a faster version of the dynamic programming approach.

Theorem 8. For genomes P and Q with n genes, $d_k(P, Q)$ can be computed in $O(n^{\lfloor k/2 \rfloor - 2}) + O(n)$ time.

Proof. Let \mathcal{S} be a maximum set of disjoint breakable subsets of black–gray cycles in $G(P, Q)$. An i_k -cycle and a $(k - i)_k$ -cycle ($i = 2, \dots, k - 2$) are called *paired* in \mathcal{S} if they form an element of \mathcal{S} . We will show how to transform the set \mathcal{S} into a maximum set \mathcal{S}' of disjoint breakable subsets of black–gray cycles in $G(P, Q)$ such that for every $i = 2, \dots, k - 2$, either all i_k -cycles are paired or all $(k - i)_k$ -cycles are paired in \mathcal{S}' .

Suppose that for some i there is a non-paired i_k -cycle p (belonging to a breakable subset \mathcal{B}_1) and a non-paired $(k - i)_k$ -cycle q (belonging to a breakable subset \mathcal{B}_2) in \mathcal{S} . If $\mathcal{B}_1 = \mathcal{B}_2$ then we replace this subset in \mathcal{S} with a breakable subset $\{p, q\}$. If $\mathcal{B}_1 \neq \mathcal{B}_2$ then we replace \mathcal{B}_1 and \mathcal{B}_2 in \mathcal{S} with breakable subsets $\{p, q\}$ and $(\mathcal{B}_1 \cup \mathcal{B}_2) \setminus \{p, q\}$. Note that this operation transforms \mathcal{S} into a maximum set of disjoint breakable subsets and increases the number of paired cycles. Therefore, after a number of steps we will arrive at a maximum set \mathcal{S}' of disjoint breakable subsets with the required property.

It is easy to see that the number of breakable subsets in \mathcal{S}' formed by an i_k -cycle and a $(k - i)_k$ -cycle equals $p_i = \min\{c_k^i(P, Q), c_k^{k-i}(P, Q)\}$ for $i \neq k/2$ and (for k even) $p_{k/2} = \lfloor c_k^{k/2}(P, Q)/2 \rfloor$. Let

$c' = (0, c_k^2(P, Q) - p_2, \dots, c_k^{k-2}(P, Q) - p_{k-2})$, except that for even k , the $(k/2)$ th component $c'_{k/2} = c_k^{k/2}(P, Q) - 2p_{k/2} = c_k^{k/2}(P, Q) \bmod 2$. Then

$$s_k(P, Q) = |S'| = s(c') + c_k^1(P, Q) + \sum_{i=2}^{\lfloor k/2 \rfloor} p_i.$$

Note that at least $\lfloor (k-3)/2 \rfloor$ coordinates of the vector c' are zero while the $(k/2)$ th coordinate (for even k) is at most 1. Therefore, the dynamic programming table for computing $s(c')$ in Theorem 7 is of size $O(n^{\lfloor k/2 \rfloor - 2})$, reducing the overall complexity of the algorithm to $O(n^{\lfloor k/2 \rfloor - 2}) + O(n)$. \square

The big-O notation in both dynamic programming algorithms hides a large constant (directly related to the size of the set V) that is exponential in k . Below we describe how one can significantly reduce this constant.

A vector v dominates a vector u if $u \leq v$. The vectors that dominate other vectors can be safely removed from the set V to compute the k -break distance more efficiently. This results in a set of extremal breakable vectors V' . In the next section we show how the set of extremal breakable vectors can be efficiently computed using Hilbert bases, and explore their relation to an explicit formula for $s_k(P, Q)$. While we are unaware of any theoretical bounds on the number of extremal breakable vectors $|V'|$, the numerical results suggest that it is small as compared to the number of proper breakable vectors $|V|$. Replacing the set of proper breakable vectors V with the set of extremal breakable vectors V' in the dynamic programming algorithms reduces the time-complexity in roughly $|V|/|V'|$ times.

Below we show how to compute the set of extremal breakable vectors via computing a certain Hilbert basis. We further use the set of extremal breakable vectors to interpret the problem of computing the k -break distance in terms of algebraic varieties. Then we employ Gröbner bases to come up with an algorithm for computing the k -break distance (for a fixed k) between two genomes with n genes in $O(n)$ time.

3.2. Extremal breakable vectors and closed-form formulas for multi-break distance

Consider an embedding $f : V \rightarrow C$ of the set V of all proper breakable vectors into a cone:

$$C = \{x \in \mathbb{Z}_+^k \mid a \cdot x = 0\}, \quad a = (-1, 1, 2, \dots, k-3, k-2, -(k-1))$$

such that

$$V \ni (v_1, \dots, v_{k-2}) \xrightarrow{f} \left(1, v_1, \dots, v_{k-2}, \frac{\sum_{i=1}^{k-2} i v_i - 1}{k-1} \right) \in C.$$

Let H be a Hilbert basis of the cone C , i.e., the minimal set of vectors such that any point in C can be expressed as an integral non-negative linear combination of vectors in H .

Theorem 9. *The set of extremal breakable vectors is $f^{-1}(f(V) \cap H)$.*

Proof. Let $H' = f(V) \cap H$ and $V' = f^{-1}(H')$. It can be easily verified that H' consists of all vectors in H with the first coordinate equal to 1.

Let $v \in V$ and S be a set of elements of the Hilbert basis H that appear in the expansion of $f(v)$ with positive coefficients. Since the first coordinate of $f(v)$ is 1, S contains exactly one element h from H' , and thus $f^{-1}(h) \leq v$. If v is an extremal vector then $f^{-1}(h) = v$, implying that $f^{-1}(H')$ contains all extremal vectors of V . On the other hand, if v is not extremal then $f(v) \neq h$, implying that the set of all extremal breakable vectors is $f^{-1}(H')$. \square

We have computed the Hilbert basis H of the cone C (for $k \leq 20$) using the algorithm from [37], and applied Theorem 9 to obtain a set of extremal breakable vectors V' . The size of H and V' is listed in Table 1.

For small k , the terms in the formula for $s_k(P, Q)$ can be mapped to the set of extremal breakable vectors V' . For example, for $k = 6$, the set of extremal breakable vectors is

$$V' = \{(1, 0, 0, 0), (0, 0, 2, 0), (0, 1, 0, 1), (0, 0, 1, 2), (0, 3, 0, 0), (0, 0, 0, 4)\}$$

Table 1

The size of the set V of all proper breakable vectors, of the Hilbert basis H of the cone C , of the set V' of extremal vectors, and of the reduced Gröbner basis GB

k	$ V = (k - 1)^{k-3}$	$ H $	$ V' $	$ GB $
3	1	3	1	1
4	3	7	2	3
5	16	13	3	9
6	125	27	6	43
7	1296	39	8	125
8	16807	83	16	1117
9	262144	117	22	8227
10	4782969	205	37	
11	100000000	291	53	
12	2357947691	555	92	
13	61917364224	634	110	
14	1792160394037	1277	201	
15	56693912375296	1567	260	
16	1946195068359375	2368	376	
17	72057594037927936	3315	519	
18	2862423051509815793	5740	831	
19	121439531096594251776	6228	963	
20	5480386857784802185939	11404	1592	

and it is mapped to the terms of the formula for $s_6(P, Q)$ as follows⁵

$$\begin{array}{cccccc}
 (1, 0, 0, 0) & (0, 0, 2, 0) & (0, 1, 0, 1) & (0, 0, 1, 2) & (0, 3, 0, 0) & (0, 0, 0, 4) \\
 \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
 c_6^1(P, Q) & \left\lfloor \frac{c_6^3(P, Q)}{2} \right\rfloor & \min\{c_6^2(P, Q), c_6^4(P, Q)\} & \delta & \left\lfloor \frac{\max\{0, c_6^2(P, Q) - c_6^4(P, Q)\}}{3} \right\rfloor & \left\lfloor \frac{\max\{0, c_6^4(P, Q) - c_6^2(P, Q)\}}{4} \right\rfloor.
 \end{array}$$

This may give a hope for a “simple” formula for $s_k(P, Q)$ that would allow one to compute $d_k(P, Q)$ efficiently. While we indeed were able to achieve it for $k < 10$ (via the Hilbert basis approach), the complexity of such formulas grows very fast with k (e.g., see how the term “ δ ” in the formula for $s_6(P, Q)$ is defined).

3.3. Computing multi-break distance in linear time

For a field \mathcal{K} , consider a polynomial ring $\mathcal{P} = \mathcal{K}[x, y_1, \dots, y_{k-2}, z_1, \dots, z_m]$ where $m = |V'|$ is the number of extremal breakable vectors.⁶ Let I be an ideal of \mathcal{P} generated by binomials $xy_1^{v_1^i} \dots y_{k-2}^{v_{k-2}^i} - z_i, i = 1, \dots, m$ where v^1, \dots, v^m are the elements of V' . Let GB be a reduced Gröbner basis of the ideal I w.r.t. the degree of x and the graded reverse lexicographical ordering of the variables $y_1, \dots, y_{k-2}, z_1, \dots, z_m$. The following theorem shows how to compute $s(c)$ in constant time using the Gröbner basis GB .

Theorem 10. *Let N be an integer such that $s(c) \leq N$ (e.g., $N = \sum_{i=1}^{k-2} c_i$), $f = x^N y_1^{c_1} \dots y_{k-2}^{c_{k-2}}$ be a polynomial in \mathcal{P} , and f' be a normal form of f with respect to the Gröbner basis GB . Then*

$$f' = x^{N-s(c)} y_1^{d_1} \dots y_{k-2}^{d_{k-2}} z_1^{e_1} \dots z_m^{e_m}$$

where $d_1, \dots, d_{k-2}, e_1, \dots, e_m$ are some non-negative integers. Moreover, $e_1 + \dots + e_m = s(c)$ and the multiset of vectors $\{(v^1)^{e_1}, \dots, (v^m)^{e_m}\}$ from V' is of the maximum cardinality with the sum of elements not exceeding c .

Proof. It follows from the Buchberger algorithm (see [38] for background information on Gröbner bases) that the reduced Gröbner basis of an ideal generated by binomials consists of binomials. Hence, the normal form of the

⁵ While knowing V' provides an intuition and facilitates the proof of the formulas for k -break distance, we are not aware of an algorithm to automatically translate V' into a formula for k -break distance.

⁶ We note that the running time of computing a Gröbner basis is highly sensitive to the number of variables. Hence, using the set of extremal breakable vectors V' instead of the set of proper breakable vectors V dramatically reduces the complexity of the Gröbner basis computing.

monomial f is a monomial. Suppose that $f' = x^{N'} y_1^{d_1} \dots y_{k-2}^{d_{k-2}} z_1^{e_1} \dots z_m^{e_m}$ where $N', d_1, \dots, d_{k-2}, e_1, \dots, e_m$ are some non-negative integers.

The definition of the function $s(\cdot)$ implies that there exist non-negative integers t_1, \dots, t_m such that $t_1 \cdot v^1 + \dots + t_m \cdot v^m \leq c$ and $t_1 + \dots + t_m = s(c)$. Then the polynomial $x^{N-s(c)} y_1^{t_1} \dots y_{k-2}^{t_{k-2}} z_1^{t_1} \dots z_m^{t_m}$ belongs to $f + I$ where $u = c - t_1 \cdot v^1 - \dots - t_m \cdot v^m$.

Since GB is a Gröbner basis of the ideal I , the polynomial f' is minimal in $f + I$. Hence, $N' \leq N - s(c)$. On the other hand, it is easy to see that $e_1 \cdot v^1 + \dots + e_m \cdot v^m \leq c$ and, thus $N - N' = e_1 + \dots + e_m \leq s(c)$ by the definition of $s(\cdot)$. Therefore, $s(c) = N - N'$. \square

For a given k , computing the reduced Gröbner basis GB may take time exponential in k . But as soon as GB is found, computing the k -break distance between genomes P and Q with n genes takes time linear in n . In particular, it takes linear time in n to construct the breakpoint graph $G(P, Q)$ and the vector c to obtain the polynomial f . Then it takes constant time (depending on k) w.r.t. n to compute a normal form of f w.r.t. GB and to obtain the distance between P and Q . For k up to 9, we have computed the reduced Gröbner basis GB using computer algebra system SINGULAR version 3.0.2 [39] (see Table 1).

4. Conclusions

In this paper we initiated studies of multi-break rearrangements in chromosomal evolution. For $k = 2$, k -breaks are similar to reversals and translocation while for $k = 3$, k -breaks are similar to transpositions and inverted transposition. However, in difference from the previously studied “standard” rearrangement operations, the k -breaks are easier to analyze and the corresponding k -break distance is easier to compute. Therefore, the k -breaks may serve as a reasonable substitute for standard rearrangement operations in various bioinformatics problems related to computing the rearrangement distance between genomes. In particular, Alekseyev and Pevzner [24] succeeded in solving the 3-Break Genome Halving problem while there is currently no solution known to the similar problem involving transpositions. The k -breaks also allow one to explicitly count the individual breaks (important to analyzing “RBM vs. FBM” alternative) as was demonstrated in [17].

Acknowledgements

We are grateful to Glenn Tesler, George Andrews, Vikas Bansal, Tzvika Hartman, and Alex Zelikovsky for insightful comments.

References

- [1] S. Ohno, *Evolution by Gene Duplication*, Springer, Berlin, 1970.
- [2] J.H. Nadeau, B.A. Taylor, Lengths of chromosomal segments conserved since divergence of man and mouse, *Proceedings of the National Academy of Sciences* 81 (3) (1984) 814–818.
- [3] P.A. Pevzner, G. Tesler, Human and mouse genomic sequences reveal extensive breakpoint reuse in mammalian evolution, *Proceedings of the National Academy of Sciences* 100 (2003) 7672–7677.
- [4] W.J. Murphy, D.M. Larkin, A.E. van der Wind, G Bourque, G. Tesler, L. Auvil, J.E. Beever, B.P. Chowdhary, F. Galibert, L. Gatzke, C. Hitte, C.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, H.A. Lewin, Dynamics of mammalian chromosome evolution inferred from multispecies comparative map, *Science* 309 (5734) (2005) 613–617.
- [5] A.E. van der Wind, S.R. Kata, M.R. Band, M. Rebeiz, D.M. Larkin, R.E. Everts, C.A. Green, L. Liu, S. Natarajan, T. Goldammer, J.H. Lee, S. McKay, J.E. Womack, H.A. Lewin, A 1463 gene cattle-human comparative map with anchor points defined by human genome sequence coordinates, *Genome Research* 14 (7) (2004) 1424–1437.
- [6] J. Bailey, R. Baertsch, W. Kent, D. Haussler, E. Eichler, Hotspots of mammalian chromosomal evolution, *Genome Biology* 5 (4) (2004) R23.
- [7] S. Zhao, J. Shetty, L. Hou, A. Delcher, B. Zhu, K. Osoegawa, P. de Jong, W.C. Nierman, R.L. Strausberg, C.M. Fraser, Human, Mouse, and Rat Genome Large-Scale Rearrangements: Stability Versus Speciation, *Genome Research* 14 (2004) 1851–1860.
- [8] C. Webber, C.P. Ponting, Hotspots of mutation and breakage in dog and human chromosomes, *Genome Research* 15 (12) (2005) 1787–1797.
- [9] H. Hinsch, S. Hannenhalli, Recurring genomic breaks in independent lineages support genomic fragility, *BMC Evolutionary Biology* 6 (2006) 90.
- [10] A. Ruiz-Herrera, J. Castresana, T.J. Robinson, Is mammalian chromosomal evolution driven by regions of genome fragility? *Genome Biology* 7 (2006) R115.
- [11] D. Sankoff, P. Trinh, Chromosomal breakpoint re-use in the inference of genome sequence rearrangement, in: *Proceedings of the Eighth Annual International Conference on Computational Molecular Biology, RECOMB, 2004*, pp. 30–35.

- [12] Q. Peng, P.A. Pevzner, G. Tesler, The fragile breakage versus random breakage models of chromosome evolution, *PLoS Computational Biology* 2 (2006) e14.
- [13] R.K. Sachs, D. Levy, P. Hahnfeldt, L. Hlatky, Quantitative analysis of radiation-induced chromosome aberrations, *Cytogenetic and Genome Research* 104 (2004) 142–148.
- [14] D. Levy, M. Vazquez, M. Cornforth, B. Loucas, R.K. Sachs, J. Arsuaga, Comparing DNA damage-processing pathways by computer analysis of chromosome painting data, *Journal of Computational Biology* 11 (2004) 626–641.
- [15] M. Vazquez, et al., Computer analysis of mFISH chromosome aberration data uncovers an excess of very complicated metaphases, *International Journal of Radiation Biology* 78 (12) (2002) 1103–1115.
- [16] R.K. Sachs, J. Arsuaga, M. Vazquez, L. Hlatky, P. Hahnfeldt, Using graph theory to describe and model chromosome aberrations, *Radiation Research* 158 (2002) 556–567.
- [17] M.A. Alekseyev, P.A. Pevzner, Are There Rearrangement Hotspots in the Human Genome? *PLoS Computational Biology* 3 (11) (2007) e209. doi:10.1371/journal.pcbi.0030209.
- [18] V. Bafna, P.A. Pevzner, Genome rearrangement and sorting by reversals, *SIAM Journal on Computing* 25 (1996) 272–289.
- [19] P.A. Pevzner, *Computational Molecular Biology: An Algorithmic Approach*, The MIT Press, Cambridge, 2000.
- [20] S. Hannenhalli, P. Pevzner, Transforming men into mouse (polynomial algorithm for genomic distance problem), in: *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, 1995, pp. 581–592.
- [21] G. Tesler, Efficient algorithms for multichromosomal genome rearrangements, *Journal of Computer and System Sciences* 65 (2002) 587–609.
- [22] M. Ozery-Flato, R. Shamir, Two notes on genome rearrangement, *Journal of Bioinformatics and Computational Biology* 1 (2003) 71–94.
- [23] S. Yancopoulos, O. Attie, R. Friedberg, Efficient sorting of genomic permutations by translocation, inversion and block interchange, *Bioinformatics* 21 (2005) 3340–3346.
- [24] M.A. Alekseyev, P.A. Pevzner, Whole Genome Duplications, Multi-Break Rearrangements, and Genome Halving Theorem, in: *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA, 2007, pp. 665–679.
- [25] V. Bafna, P.A. Pevzner, Sorting permutations by transpositions, *SIAM Journal on Discrete Mathematics* 11 (1998) 224–240.
- [26] D.A. Christie, *Genome Rearrangement Problems*, Ph.D. Thesis, University of Glasgow (1999).
- [27] M.E. Walter, L. Reginaldo, A.F. Curado, A.G. Oliveira, Working on the problem of sorting by transpositions on genome rearrangements, *Lecture Notes in Computer Science* 2676 (2003) 372–383.
- [28] T. Hartman, A simpler 1.5-approximation algorithm for sorting by transpositions, *Lecture Notes in Computer Science* 2676 (2003) 156–169.
- [29] I. Elias, T. Hartman, A 1.375-approximation algorithm for sorting by transpositions, *Lecture Notes in Computer Science* 3692 (2005) 204–214.
- [30] M. Bader, E. Ohlebusch, Sorting by weighted reversals, transpositions, and inverted transpositions, in: *Proceedings of the 10th Conference on Research in Computational Molecular Biology*, RECOMB, 2006, pp. 563–577.
- [31] Q.P. Gu, S. Peng, H. Sudborough, A 2-approximation algorithm for genome rearrangements by reversals and transpositions, *Theoretical Computer Science* 210 (1999) 327–339.
- [32] T. Hartman, R. Sharan, A 1.5-approximation algorithm for sorting by transpositions and transreversals, *Lecture Notes in Computer Science* 3240 (2004) 50–61.
- [33] G.H. Lin, G. Xue, Signed genome rearrangements by reversals and transpositions: models and approximations, *Theoretical Computer Science* 259 (2001) 513–531.
- [34] Y.C. Lin, C.L. Lu, H.-Y. Chang, C.Y. Tang, An efficient algorithm for sorting by block-interchanges and its application to the evolution of vibrio species, *Journal of Computational Biology* 12 (2005) 102–112.
- [35] A.J. Radcliffe, A.D. Scott, E.L. Wilmer, Reversals and transpositions over finite alphabets, *SIAM Journal on Discrete Mathematics* 19 (2005) 224–244.
- [36] M.E. Walter, Z. Dias, J. Meidanis, Reversal and transposition distance of linear chromosomes, in: *String Processing and Information Retrieval: A South American Symposium*, SPIRE, 1998, pp. 96–102.
- [37] D.V. Pasechnik, On computing the Hilbert bases via the Elliott–MacMahon algorithm, *Theoretical Computer Science* 263 (2001) 37–46. implementation: <http://stuwwww.uvt.nl/~dpasech/software.html>.
- [38] D. Cox, J. Little, D. O’Shea, *Ideals, Varieties, and Algorithms*, Springer-Verlag, 1996.
- [39] G.-M. Greuel, G. Pfister, H. Schönemann, *Singular 3.0.2*. Website: <http://www.singular.uni-kl.de>.
- [40] M.A. Alekseyev, Multi-break rearrangements: from linear to circular genomes, *Lecture Notes in Bioinformatics* 4751 (2007) 1–15. doi:10.1007/978-3-540-74960-8-1.