View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Elsevier - Publisher Connector

ELSEVIER

Discrete Applied Mathematics 71 (1996) 137–151

DISCRETE APPLIED MATHEMATICS

Polynomial-time algorithm for computing translocation distance between genomes

Sridhar Hannenhalli *

Department of Mathematics, University of Southern California, Los Angeles, CA 90089-1113, USA

Received 24 May 1995; revised 1 May 1996; accepted 20 May 1996

Abstract

With the advent of large-scale DNA physical mapping and sequencing, studies of genome rearrangements are becoming increasingly important in evolutionary molecular biology. From a computational perspective, the study of evolution based on rearrangements leads to a *rearrangement distance problem*, i.e., computing the minimum number of rearrangement events required to transform one genome into another. Different types of rearrangement events give rise to a spectrum of interesting combinatorial problems. The complexity of most of these problems is unknown. Multichromosomal genomes frequently evolve by a rearrangement event called *translocation* which exchanges genetic material between different chromosomes. In this paper we study the *translocation distance problem*, modeling the evolution of genomes evolving by translocations. The translocation distance problem was recently studied for the first time by Kececioglu and Ravi, who gave a 2-approximation algorithm for computing translocation distance. In this paper we prove a duality theorem leading to a polynomial time algorithm for computing translocation distance for the case when the orientations of the genes are known. This leads to an algorithm generating a most parsimonious (shortest) scenario, transforming one genome into another by translocations.

1. Introduction

The first computational attempt to analyze genome rearrangements in mamalian genomes was undertaken by Nadeau and Taylor in 1984 who estimated that just 178 ± 39 rearrangement events happened since the separation of lineages leading to human and mice 80 million years ago. This estimate was recently validated by Copeland et al. [4] based on a man–mouse genetic linkage map of much higher resolution compared to the one available 10 years ago. The most common rearrangement events in

* E-mail: *hannenha@hto.psu.edu*.

This work is supported by NSF Young Investigator Award, NIH grant 1R01 HG00987 and DOE grant DE-FG02-94ER61919.

0166-218X/96/$15.00 © 1996 Elsevier Science B.V. All rights reserved
SSDI 0166-218X(96)00061-3

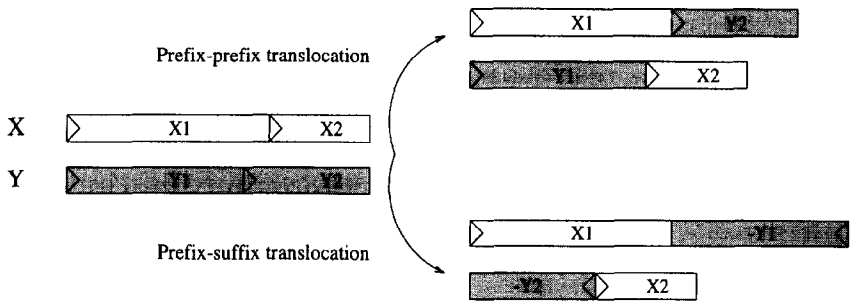

Fig. 1. Examples of translocations. Notice the change in the directions of chromosomal segments, Y_1 and Y_2, after prefix–suffix translocation.

mammalian evolution are *translocations*, which exchange genetic material between different chromosomes, and *reversals*, which rearrange genetic material within a chromosome.

A computational approach to evolutionary studies based on rearrangements was pioneered by Sankoff (see [12–14]). The study of genomes evolving by rearrangements involves a combinatorial problem of computing the minimum number of rearrangement events transforming one genome into other and finding a shortest (most parsimonious) sequence of rearrangement events transforming one genome into other.

Although *translocation* is a complicated biological process (see [11, 15] for the underlying biology), the following abstraction is adequate for our purpose. A chromosome can be represented as a sequence of genes, where each gene is represented by an integer. A translocation is said to act on chromosomes X and Y when the chromosomes are cleavaged as (X_1, X_2) and (Y_1, Y_2), respectively, and the *segments* of the chromosomes are swapped, thus transforming chromosomes X and Y into two new chromosomes. We study the most common type of translocation, viz., reciprocal translocation where each of the four segments, X_1, X_2, Y_1 and Y_2, is non-empty. A translocation is a *prefix–prefix* translocation if the prefix of one chromosome is swapped with the prefix of the other chromosome, and a translocation is a *prefix–suffix* translocation if the prefix of one chromosome is swapped with the suffix of the other chromosome (Fig. 1).

For our purposes, a genome is a set of chromosomes. A translocation on a pair of chromosomes of genome A transforms genome A into another genome. Given two genomes, A and B, the *translocation distance* between A and B, $d(A, B)$, is the minimum number of translocations required to transform A into B. We refer to any sequence of translocations transforming A into B as *evolution* of A into B.

Under most of the rearrangement events, the complexity of the rearrangement distance problem is still unknown. The importance of these problems has motivated researchers to develop approximation algorithms for rearrangement distance problems for various types of rearrangements. The first steps towards a combinatorial theory of genome rearrangements have been taken very recently. Kececioglu and Sankoff [9, 10] and Bafna and Pevzner [1] gave approximation algorithms for computing the rearrangement distance for genomes evolving by reversals. (The problem is known as

"sorting signed permutation by reversals".) Recently, Hannehalli and Pevzner [6, 7] showed that the problem of sorting signed permutations by reversals is in **P** by proving a duality theorem that gives an efficiently computable characterization of reversal distance. Recently, Kececioglu and Ravi [8] gave a 2-approximation algorithm for the rearrangement distance problem for genomes evolving by translocations and a 1.5-approximation algorithm for the rearrangement distance problem for genomes evolving by both translocations and reversals. See Bafna and Pevzner [2] and Hannenehalli et al. [5] for applications of genome rearrangement algorithms to analyze the evolution of plant organelles, mammalian X chromosomes and herpes viruses. Also, see Bafna and Pevzner [3] for a computational study of genomes evolving by another type of rearrangement event called transposition.

In this paper we prove a duality theorem characterizing the translocation distance for signed data. This leads to a polynomial algorithm which computes the shortest sequence of translocations transforming one genome into another. We restrict our discussion to the case when both *prefix–prefix* and *prefix–suffix* reciprocal translocations are allowed. The case when only prefix–prefix translocations are allowed is amenable to similar analysis and will not be discussed in this paper.

All chromosomes contain a *centromere* which is important for cell division. A translocation is *viable* if both the resulting chromosomes contain a centromere. This restricts the translocations in the course of evolution. Including centromeres in our model does not present additional difficulty. For simplicity we omit centromeres from our model.

In the following section we present the combinatorial formulation of the problem. In Section 3 we prove a lower bound on the translocation distance. In Section 4 we prove a duality theorem leading to a polynomial algorithm for computing translocation distance. In Section 5 we present an algorithm generating a most parsimonious (shortest) scenario of evolution, transforming one genome into other. And, finally, in Section 6 we briefly discuss the case of unsigned data.

2. Combinatorial formulation

For the purpose of the following discussion, a *gene* will be represented by a signed integer, where the sign models the direction of the gene, a *chromosome* is a sequence of genes and a *genome* is a set of chromosomes. We assume that the given genomes $A = ((a_{11}, a_{12}, \ldots, a_{1m_1}), (a_{21}, a_{22}, \ldots, a_{2m_2}), \ldots, (a_{N1}, a_{N2}, \ldots, a_{Nm_N}))$ and $B = ((b_{11}, b_{12}, \ldots, b_{1n_1}), (b_{21}, b_{22}, \ldots, b_{2n_2}), \ldots, (b_{N1}, b_{N2}, \ldots, b_{Nn_N}))$, contain the same set of genes and that every gene appears in each genome exactly once. For an arbitrary sequence $S = s_1, s_2, \ldots, s_k$ of genes, we will denote the reverse ordering of S by $-S$. *i.e.*, $-S = -s_k, -s_{k-1}, \ldots, -s_1$. A chromosome Y is said to be *identical* to a chromosome $X = (x_1, x_2, \ldots, x_k)$ iff either $Y = X$ or $Y = -X$. Genomes A and B are said to be identical ($A = B$) iff the sets of chromosomes corresponding to A and B are the same.

As a convention we illustrate a chromosome horizontally and read it from left to right. Since we do not distinguish a chromosome from its reverse ordering, any prefix-suffix translocation acting on X and Y can be visualized as a prefix-prefix translocation acting on X and $-Y$. For a pair of chromosomes $X = (x_1, x_2, \ldots, x_m)$ and $Y = (y_1, y_2, \ldots, y_n)$ denote translocation acting on X and Y as $\rho(X, Y, i, j), 1 < i \leqslant m, 1 < j \leqslant n$, where the cleavage occurs in X between x_{i-1} and x_i and in Y between y_{j-1} and y_j. A prefix-prefix translocation $\rho_{pp}(X, Y, i, j)$ results into chromosomes: $(x_1, \ldots, x_{i-1}, y_j, \ldots, y_n)$ and $(y_1, \ldots, y_{j-1}, x_i, \ldots, x_m)$. A prefix-suffix translocation $\rho_{ps}(X, Y, i, j)$ results into chromosomes: $(x_1, \ldots, x_{i-1}, -y_{j-1}, \ldots, -y_1)$ and $(-x_m, \ldots, -x_i, y_j, \ldots, y_n)$. For a genome A and a translocation ρ acting on a pair of chromosomes of A, we denote the resulting genome as $A \cdot \rho$. If ρ is a reciprocal translocation then the number of chromosomes in A and $A \cdot \rho$ is the same. Moreover, the set of *nodal* genes (first and the last gene of all the chromosomes) is the same for A and $A \cdot \rho$. Figure 2a shows an example of evolution of A into *target* genome B. In the following discussion we assume, w.l.o.g., that a target genome is fixed and refer to the translocation distance between A and the target genome as *translocation distance of A*, thus, $d(A) \equiv d(A, B)$. Also, we refer to the problem of finding a shortest sequence of translocations transforming A into the target genome as the problem of *sorting A by translocations.*

In the following, we introduce *cycle graph* of a genome, which is the basis of our analysis of translocation distance. In a chromosome $X = (x_1, x_2, \ldots, x_k)$, replace every positive integer $+x_i$ by ordered pair (x_i^t, x_i^h) of vertices (t stands for tail and h stands for head) and replace every negative integer $-x_i$ by ordered pair (x_i^h, x_i^t) of vertices (Fig 2b). We say that vertices u and v are *neighbors* in X if they are adjacent in the ordered list constructed in afore mentioned manner. Notice that u and v are neighbors in X iff u and v are neighbors in $-X$. We say that vertices u and v are *neighbors* in a genome if they are neighbors in some chromosome in this genome. For gene x, vertices x^t and x^h are always neighbors and for simplicity, we exclude them from the definition of "neighbors" in the following discussion. We construct the bicolored *cycle graph* $G_A \equiv G_{AB}(V, E)$ of a genome A (with respect to a fixed target genome B) as follows. The vertex set V contains the pair of vertices x^t and x^h for every gene x in A, i.e., $V = \{u : u \text{ is either } x^t \text{ or } x^h, x \text{ is a gene in } A\}$. The edges of G_A are colored either gray or black. Vertices u and v are connected by a black (solid) edge iff they are neighbors in A. Vertices u and v are connected by a gray (dotted) edge iff they are neighbors in the target genome (Fig. (2b)). Notice that x^t and x^h are not connected for any x and a pair of vertices which are neighbors in both the genomes are connected by both, a black and a gray edge. See [1, 6, 8] for similar constructions.

The number of black (equivalently, gray) edges in G_A is $n - N$ where n is the number of genes in A and N is the number of chromosomes. Clearly, each vertex is adjacent to exactly one black edge and one gray edge. Hence the graph can be uniquely decomposed into a number of disjoint cycles. We denote the number of cycles in G_A as c_A. Clearly, the number of cycles is maximized when A is identical to the target genome.

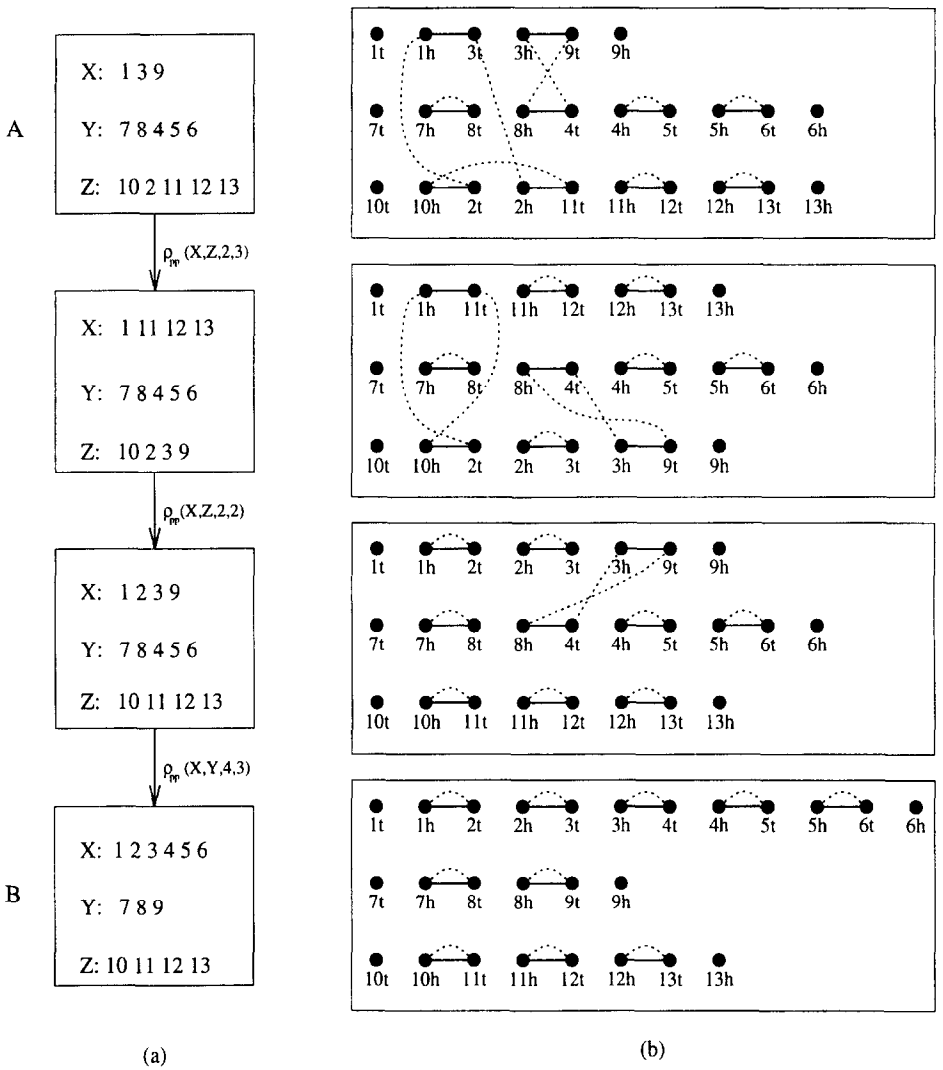

Fig. 2. (a) An example of evolution by translocations, (b) cycle graph corresponding to genomes at every stage of evolution with respect to the fixed target genome B.

Lemma 1. *$c_A = n - N$ iff A is identical to the target genome.*

For the sake of simplicity, we will refer to all the intermediate genomes in the course of evolution of A as A. Any such sequence of translocations will be reflected as changes in the associated cycle graph until the graph is left with all cycles of length 2, i.e., $c_A = n - N$ (Fig. (2b)).

Let $\rho \equiv \rho(X, Y, i, j)$ be a translocation acting on chromosomes $X = (x_1, x_2, \ldots, x_m)$ and $Y = (y_1, y_2, \ldots, y_n)$. Let $f \in \{x^t_{i-1}, x^h_{i-1}\}$ and $g \in \{x^t_i, x^h_i\}$ such that f and g are

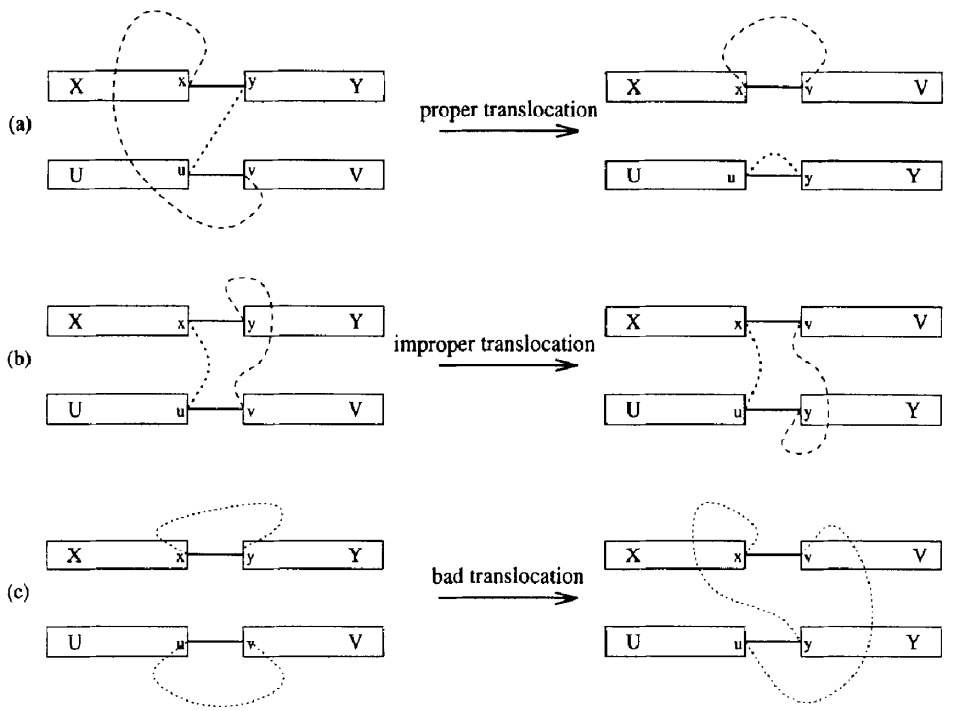

Fig. 3. Prefix–prefix translocations cutting black edges $(u\ v)$ and $(x\ y)$ affect the cycle graph G_A. (a) For a proper translocation $\Delta(c_A) = 1$. (b) For an improper translocation $\Delta(c_A) = 0$. (c) For a bad translocation $\Delta(c_A) = -1$.

neighbors in X. Let $u \in \{y^t_{j-1}, y^h_{j-1}\}$ and $v \in \{y^t_j, y^h_j\}$ such that u and v are neighbors in Y. We say that ρ *cuts* black edges $(f\ g)$ and $(u\ v)$.

A prefix–prefix translocation cutting black edges $(u\ v)$ and $(f\ g)$ (read from the left to the right) is *proper* if there is a cycle $(u\ v \ldots f\ g \ldots u)$ in G_A. In Fig. 2, the translocation $\rho_{pp}(X, Z, 2, 3)$ is proper since the black edges $(2^h\ 11^t)$ and $(1^h\ 3^t)$ cut by this translocation belong to the cycle $(2^h\ 11^t\ 10^h\ 2^t\ 1^h\ 3^t\ 2^h\ )$. A prefix–suffix translocation cutting black edges $(u\ v)$ and $(f\ g)$ (read from the left to the right) is *proper* if there is a cycle $(u\ v \ldots g\ f \ldots u)$ in G_A. Notice that for every pair of black edges on different chromosomes but belonging to the same cycle in the cycle graph, there is proper translocation (prefix–prefix or prefix–suffix) cutting the two black edges. We call a translocation *improper*, if it cuts black edges belonging to the same cycle but is not proper. We call a translocation *bad* if it cuts black edges belonging to different cycles (Fig. 3).

In the following we study the effect of a translocation on the structure of the cycle graph and describe the parameters that play a key role in determining the translocation distance.

3. Lower bound on translocation distance

Let ψ_A be a parameter ψ associated with genome A (or G_A). For a translocation ρ on A we denote the increase in ψ as $\Delta(\psi)$, i.e., $\Delta(\psi) \equiv \psi_{A\cdot\rho} - \psi_A$.

Lemma 2. *For a translocation ρ, $\Delta(c_A) = 1$ iff ρ is proper, $\Delta(c_A) = 0$ iff ρ is improper and $\Delta(c_A) = -1$ iff ρ is bad (Fig. 3).*

Lemmas 1 and 2 imply

Theorem 3. *For an arbitrary genome A, $d(A) \geqslant n - N - c_A$.*

As it turns out, there are additional parameters associated with a genome which are important in computing the translocation distance. In particular, if a set of genes occur close together within a chromosome in both the genomes but not in the same order, then reordering them necessitates a translocation that decreases the number of cycles. This leads to the notion of a subpermutation described in the following. Define *segment* as an interval $I = x_i, x_{i+1}, \ldots, x_j$ within a chromosome $X = x_1, x_2, \ldots, x_m$ in A. Let V_I be the set of vertices induced by the genes in I, i.e., $V_I = \{u : u \text{ is either } x_k^t \text{ or } x_k^h,\ i \leqslant k \leqslant j\}$. We refer to the left vertex corresponding to x_i and the right vertex corresponding to x_j as $LEFT(I)$ and $RIGHT(I)$, respectively. In Fig. 4, for the interval $I = 2,4,3,5$, $LEFT(I) = 2^t$ and $RIGHT(I) = 5^h$. Define $IN(I) = V_I \backslash \{LEFT(I) \cup RIGHT(I)\}$. An edge $(u\ v) \in G_A$ is said to be *inside* the interval I if $u, v \in IN(I)$. A *subpermutation* (SP) is an interval of genes $x_i, x_{i+1}, \ldots, x_j$ within a chromosome X in genome A such that there exists a segment x_i, *permutation*$(x_{i+1}, \ldots, x_{j-1}), x_j$ within some chromosome Y of target genome B and *permutation*$(x_{i+1}, \ldots, x_{j-1}) \neq x_{i+1}, \ldots, x_{j-1}$. Equivalently, SP is an interval I within some chromosome of A such that (i) there exists no edge $(u\ v)$ such that $u \in IN(I)$ and $v \notin IN(I)$ and (ii) there is at least one *long cycle* (of size > 2) involving edges inside I. A *minimal subpermutation* (*minSP*) is a SP not containing any other SP. The size of a SP is the number of genes in the SP. In Fig. 4 the interval (2 4 3 5) is a *minSP* of size 4 contained inside the SP (1 2 4 3 5 6) of size 6.

Notice that for an arbitrary partition of a SP into a non-empty prefix segment L and a non-empty suffix segment R, there must be a gray edge $(u\ v)$ such that $u \in V_L, v \in V_R$ (the cycle containing the black edge ($RIGHT(L)$ $LEFT(R)$) must contain such a gray edge). We refer to such an edge as a *connecting* gray edge from L to R.

A translocation *cuts* a segment S iff it cuts a black edge inside S. A translocation ρ *destroys* a SP S in A if S is not a SP in $A \cdot \rho$. What makes SPs interesting is that in order to destroy a SP we must do a bad translocation since there is atleast one long cycle in a SP and any translocation cutting a black edge inside a SP must cut a black edge belonging to a different cycle. *minSP*s are specially interesting in this respect since destroying a *minSP* S destroys all the SPs containing S. We can destroy at most 2 *minSP*s on different chromosomes in a single bad translocation by choosing

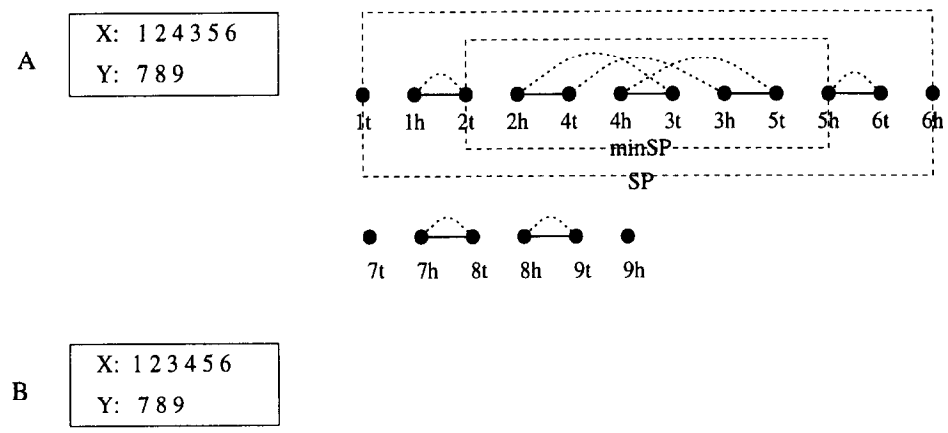

Fig. 4. Examples of subpermutations.

a translocation cutting both the *minSP*s. In the process any *SP* containing either of the *minSP*s is also destroyed. However a pair of *minSP*s on the same chromosome cannot be destroyed simultaneously in a single bad translocation. Whenever we destroy any *minSP* (1 or 2), the translocation must cut black edges from two different cycles and hence $\Delta(c_A) = -1$. If s_A is the number of *minSP*s in A then

Lemma 4. *For any translocation* $\Delta(c_A - s_A) \leqslant 1$.

This leads to a slightly improved lower bound, $d(A) \geqslant n - N - c_A + s_A$, since $s_A = 0$ if A is identical to the target genome. In the following we show that this bound is very tight by proving that $d(A) \leqslant n - N - c_A + s_A + 2$. The bound, $d(A) \geqslant n - N - c_A + s_A$, assumes destroying two *minSP*s in a single translocation since if we destroy exactly one *minSP* in a translocation then $\Delta(c_A - s_A) = 0$. Sometimes it may be impossible to destroy two *minSP*s in a single bad translocation and, thus, unavoidable to make a translocation with $\Delta(c_A - s_A) \leqslant 0$. If the number of *minSP*s is odd then we cannot avoid such a translocation. Let $o_A = 1$ if the number of *minSP*s is odd and $o_A = 0$ otherwise. Clearly, $\Delta(c_A - s_A) = 1$ implies that $\Delta(o_A) = 0$. One could verify that

Lemma 5. *For any translocation* $\Delta(c_A - s_A - o_A) \leqslant 1$.

Genome A has an *even-isolation* if

(i) all the *minSP*s of A reside on a single chromosome,

(ii) s_A is even and

(iii) all the *minSP*s are contained within a single *SP*.

Notice that if A has an *even-isolation*, we must perform a translocation with $\Delta(c_A - s_A - o_A) \leqslant -1$. Consider the first translocation ρ destroying the even-isolation. If $\Delta(s_A) > 0$ (ρ creates one or two new *minSP*s) then $\Delta(s_A + o_A) = 2$, hence $\Delta(c_A - s_A - o_A) \leqslant -1$. One could verify that if a translocation ρ destroys a *minSP* then there cannot be a new *minSP* in $A \cdot \rho$. Moreover, due to property (iii) in the definition of an

even-isolation there cannot be a proper translocation distributing the *minSP*s over two chromosomes. Hence, if $\Delta(s_A) = 0$ then ρ must be a bad translocation distributing the existing *minSP*s over two chromosomes, thus destroying even-isolation. In this case $\Delta(s_A) = \Delta(o_A) = 0$, hence $\Delta(c_A - s_A - o_A) = -1$. If $\Delta(s_A) = -1$ (ρ destroys one *minSP*) then ρ must be a bad translocation. In this case $\Delta(c_A) = -1$, $\Delta(o_A) = 1$, hence $\Delta(c_A - s_A - o_A) = -1$.

Let $i_A = 1$ if A has an even-isolation and $i_A = 0$ otherwise. One could verify that

Lemma 6. *For any translocation* $\Delta(c_A - s_A - o_A - 2 \cdot i_A) \leqslant 1$.

Notice that $o_A = i_A = 0$ if A is identical to the target genome. This gives us an improved lower bound.

Theorem 7. *For an arbitrary genome A,* $d(A) \geqslant n - N - c_A + s_A + o_A + 2 \cdot i_A$.

4. Duality theorem for translocation distance

We call a translocation *valid* if $\Delta(c_A - s_A - o_A - 2 \cdot i_A) = 1$. In this section we prove the existence of a valid translocation for arbitrary genome A, implying that $d(A) = n - N - c_A + s_A + o_A + 2 \cdot i_A$. We say that a proper translocation ρ *acts* on the gray edge $(u\ v)$ if it cuts the black edges incident on u and v. In the following, we consider only the proper translocations acting on gray edges. Denote an arbitrary non-empty prefix (suffix) of a segment S by $pref(S)$ ($suff(S)$). Denote a segment formed by concatenating segments L and R (in that order) as $[L\ R]$.

Lemma 8. *For an arbitrary partition of a minSP S into prefix segment L and suffix segment R, there exists a gray edge connecting L to R different from* $(RIGHT(L)\ LEFT(R))$.

Proof. If $(RIGHT(L)\ LEFT(R))$ is the only gray edge connecting L to R, then L (and/or R) will qualify for a *SP* or S did not have any long cycle inside it, implying that S was not a *minSP*, a contradiction. □

Lemma 9. *For an arbitrary partition of a minSP S into prefix segment L, middle segment M and suffix R where all three of the segments are non-empty, there must be a gray edge g such that g either connects L to M or g connects M to R.*

Proof. Since M does not contain nodal elements, there must be a vertex $v \in V_M$ which is a neighbor of some vertex $u \notin V_M$ in genome B. Clearly, $u \in V_L \cup V_R$. □

Theorem 10. *If there exists a proper translocation in A then there exists a proper translocation* σ *in A such that* $A \cdot \sigma$ *does not have any new minSP.*

Proof. Assume that every proper translocation leads to the creation of a new *minSP*. Let ρ be the prefix–prefix (w.l.o.g.) translocation creating a *smallest* such *minSP*. Notice that ρ could create at most two new *minSP*s. Let $[L\ R]$ be the (smallest) new *minSP* in the genome $A \cdot \rho$ where segments L and R belong to different chromosomes in genome A (Fig. (5a)). Our goal is to find an alternative proper translocation in A cutting L and R which either does not create a new *minSP* or creates a smaller one. Since $(RIGHT(L)\ LEFT(R))$ is a gray edge, by Lemma 8, there must be a gray edge $(l\ r)$ in $A \cdot \rho$ (hence in A) such that $l \in IN(L)$, $r \in IN(R)$. A proper translocation σ (acting on $(l\ r)$) cuts L and R. Assume that the translocation σ *breaks up* L into non-empty segments L_1 and L_2 and breaks up R into non-empty segments R_1 and R_2 (Fig. 5(b) and (c)).

Case 1: σ *is a prefix-prefix translocation* (*Fig.* (5b)). We will now argue that the resulting genome $A \cdot \sigma$ either does not have any new *minSP* or has a smaller one. We need to concentrate only on the subgraph induced by the pair of chromosomes involved in the translocation since the rest of the graph remains unchanged. Clearly, any new *minSP* must involve parts of either L or R.

A new *minSP* in $A \cdot \sigma$ cannot be

(i) $[suff(P)\ [L1\ R2]\ pref(V)]$ since there must be a gray edge $(u\ v)$ such that $u \in V_{[L1\ R2]}$, $v \in V_{[R1\ L2]}$ (Lemma 9).

(ii) $[suff(U)\ [R1\ L2]\ pref(Q)]$ (similar to (i)).

(iii) $[suff(P)\ pref([L1\ R2])]$, since, by Lemma 8, there must be a connecting gray edge between $suff(P)$ and $pref([L1\ R2])$ different from $(RIGHT(P)\ LEFT([L1\ R2]))$ in $A \cdot \sigma$ (and hence in $A \cdot \rho$), implying that $[L\ R]$ is not a *minSP* in $A \cdot \rho$, a contradiction.

(iv) $[suff([L1\ R2])\ pref(V)]$ (similar to (iii)).

(v) $[suff(U)\ pref([R1\ L2])]$ (similar to (iii)).

(vi) $[suff([R1\ L2])\ pref(Q)]$ (similar to (iii)).

So any *minSP* involving parts of either L or R must be within $[L1\ R2]$ or $[R1\ L2]$, hence smaller than $[L\ R]$, a contradiction.

Case 2: σ *is a prefix – suffix translocation* (*Fig.* (5c)). Arguments for this case are very similar to the previous case. Notice that there is a gray edge between $LEFT(-L2)$ $(RIGHT(L2))$ and $RIGHT(-R1)$ $(LEFT(R1))$ (ρ acts on this edge in A). This prevents the creation of a new *minSP* $[suff(P)\ [L1\ -R1]\ pref(-U)]$ or $[suff(-Q)\ [-L2\ R2]\ pref(V)]$ in $A \cdot \sigma$. There cannot be any new *minSP* in $A \cdot \sigma$ involving parts of either L or R by the same arguments as for the previous case. Hence we conclude that any *minSP* involving parts of either L or R is within $[L1\ -R1]$ or $[-L2\ R2]$, hence smaller that $[L\ R]$, a contradiction to the assumption that ρ created the smallest *minSP*. □

Let S_1 and S_2 be two *minSP*s within chromosome X of A such that S_1 is to the left of S_2. A gray edge $(u\ v)$ *separates* S_1 and S_2 if the vertex v belongs to a chromosome different from X and the vertex u is in between the vertices $RIGHT(S_1)$ and $LEFT(S_2)$ on X, i.e., u is to the right of $RIGHT(S_1)$ and to the left of $LEFT(S_2)$ in the ordered

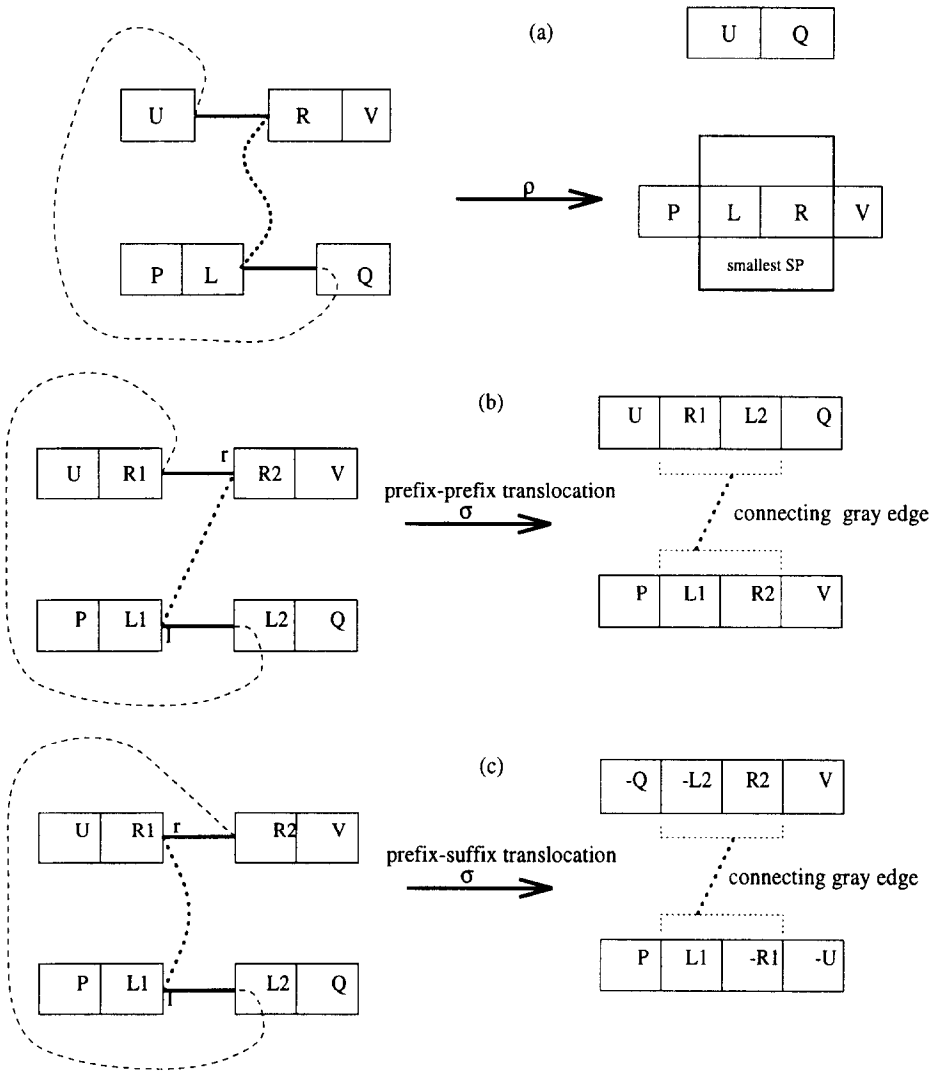

Fig. 5. (a) Proper prefix–prefix translocation ρ creates a new *minSP*. (b),(c) Finding an alternate proper translocation not creating any new *minSP*.

list of vertices induced by genes of X. A translocation, acting on $(u\ v)$, *separates* S_1 and S_2 if $(u\ v)$ separates S_1 and S_2.

Theorem 11. *If there is a gray edge separating minSPs S_1 and S_2 in genome A, then there exists a valid translocation σ separating S_1 and S_2.*

Proof. By Theorem 10, if a translocation ρ acting on gray edge $(u\ v)$ creates a *minSP* $[L\ R]$ (proof of Theorem 10), there exists an alternative proper translocation σ connecting L to R which does not create a *minSP* (following the arguments in the proof of Theorem 10). Notice that if ρ separates *minSP*s S_1 and S_2 then σ separates S_1 and

S_2, implying that $A \cdot \sigma$ does not have an even-isolation. Hence σ is a proper valid translocation separating S_1 and S_2. □

Theorem 12. *If there exists a proper translocation in A then there exists a valid translocation in A.*

Proof. Notice that any proper translocation cannot destroy any of the existing *minSP*s. If there exists a proper translocation separating two *minSP*s , by theorem 11, there exists a proper valid translocation. On the other hand if there is no gray edge separating two *minSP*s then, by theorem 10, there is a proper translocation not creating any new *minSP*. Any such proper translocation ρ cannot create an even-isolation, since an even-isolation in $A \cdot \rho$ implies an even-isolation in A. Hence ρ is valid. □

Theorem 13. *For every genome there exists a valid translocation.*

Proof. If there is a proper translocation in A then, by Theorem 12, there exists a valid translocation in A. Assume that there is no proper translocation in A. Clearly, in this situation, there is no gray edge going across two different chromosomes and hence G_A is a collection of *SP*s distributed over the chromosomes. Let v be the number of chromosomes containing at least one *minSP*.

Case 1: $v = 1$. If $s_A = 1$ then we can destroy the only *minSP* by choosing a translocation cutting the *minSP* and an arbitrary black edge in some other chromosome. In this case $\Delta(c_A - s_A - o_A - 2 \cdot i_A) = (-1 - (-1) - (-1) - 0) = 1$.

If $s_A > 1$ then choose a translocation ρ that destroys the *minSP* second from the left using the same technique. Notice that, either two chromosomes in $A \cdot \rho$ contain atleast one *minSP* or $s_{A \cdot \rho} = 1$. In either case $A \cdot \rho$ does not have even-isolation.

If s_A is odd, then $\Delta(c_A - s_A - o_A - 2 \cdot i_A) = (-1 - (-1) - (-1) - 0) = 1$. If s_A is even (A has even-isolation), then $\Delta(c_A - s_A - o_A - 2 \cdot i_A) = (-1 - (-1) - 1 - 2 \cdot (-1)) = 1$.

Case 2: $v = 2$. If $s_A = 2$ then choose a translocation ρ that destroys both of the *minSP*s. If $s_A > 2$ then choose a translocation ρ that destroys the leftmost *minSP* within one chromosome and the second *minSP* within the other chromosome. Since $A \cdot \rho$ cannot have even-isolation, $\Delta(c_A - s_A - o_A - 2 \cdot i_A) = (-1 - (-2) - 0 - 0) = 1$.

Case 3: $v = 3$. If $s_A = 3$ then choose a translocation ρ that destroys any two *minSP*s. If $s_A > 3$ then choose a translocation ρ acting on chromosomes X and Y where X has atleast two *minSP*s within it. Since $A \cdot \rho$ cannot have even-isolation, $\Delta(c_A - s_A - o_A - 2 \cdot i_A) = (-1 - (-2) - 0 - 0) = 1$.

Case 4: $v \geqslant 4$. In this case every translocation ρ, destroying two *minSP*s, is valid. □

Theorems 10, 12 and 13 imply the following duality theorem providing a characterization of translocation distance.

Theorem 14. *For an arbitrary genome A,* $d(A) = n - N - c_A + s_A + o_A + 2 \cdot i_A$, *i.e.,*

$$d(A) = \begin{cases} n - N - c_A + s_A + 2 & \text{if } A \text{ has an even-isolation,} \\ n - N - c_A + s_A + 1 & \text{if } A \text{ has an odd number of minSPs,} \\ n - N - c_A + s_A & \text{otherwise.} \end{cases}$$

5. Algorithm for sorting by translocations

Theorems 12 and 13 motivate the algorithm *Translocation_Sort* generating a shortest sequence of translocations transforming genome A into the target genome.

Algorithm *Translocation_Sort*(A)
1. **while** A is not identical to the target genome
2. **if** there is a proper translocation in A
3. select a valid proper translocation ρ (Theorem 12)
4. **else** select a valid bad translocation ρ (Theorem 13)
5. $A \leftarrow A \cdot \rho$
6. **endwhile**

The cycle graph G_A can be constructed in $O(n)$ time where n is the number of genes in A. A data structure to maintain the list of gray edges leading to proper translocations and the list of *minSP*s can be initialized in $O(n^3)$. Clearly, there are at most $O(n)$ iterations. Step 3 may require searching among at most $O(n)$ proper translocations since every alternative choice of a proper translocation reduces the size of the new *minSP* created (proof of Theorem 10). Checking the validity of any such translocation takes $O(n)$ time. In the absence of any proper translocation Theorem 13 suggests a way to find a valid bad translocation in constant time. Performing the valid translocation in step 5 involves updating the data structures which can be done in at most $O(n^2)$ time. Therefore the overall running time of *Translocation_Sort* is $O(n^3)$.

6. The case of unsigned data

Physical maps usually do not provide information about directions of genes, thus leading to the problem of computing the rearrangement distance for unsigned data. We can construct the cycle graph for unsigned data by assuming arbitrary direction for each gene, constructing the cycle graph as described earlier and then collapsing the vertices x^t and x^h for every gene x. Notice that the resulting graph has an equal number of black and gray edges incident on every vertex, hence the graph can be decomposed into alternating cycles (cycles whose edges alternate colors). Any such decomposition can be viewed as assigning a direction to every gene. W.l.o.g., all genes in the target genome B have positive orientation. An assignment of directions to the genes in the

source genome A dictated by the decomposition of the cycle graph is defined as a *spin* of A. Let $\hat{A}$ be the set of all spins of A. It is not hard to show that

$$d(A) = \min_{\vec{A} \in \hat{A}} d(\vec{A}).$$

Refer to Hannenhalli and Pevzner [7] for similar arguments. Hence the problem of computing the translocation distance for unsigned data is equivalent to the problem of computing an optimal spin of A, i.e., a spin that minimizes the translocation distance. This equivalent characterization could be used to approximate the translocation distance for unsigned data. It can be shown that any decomposition of the cycle graph that attempts to maximize the number of cycles leads to an approximation of the translocation distance. At this point, the existence of a polynomial algorithm to compute the translocation distance for unsigned data remains an open problem when both prefix–prefix and prefix–suffix reciprocal translocations are allowed.

Acknowledgments

The author is very thankful to Pavel Pevzner for many helpful suggestions, and for pointing out a few mistakes in the earlier versions of this paper. The author also wishes to thank the referees for their comments.

References

[1] V. Bafna and P. Pevzner, Genome rearrangements and sorting by reversals, in: Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science (1993) 148–157; SIAM J. Comput., to appear.

[2] V. Bafna and P. Pevzner, Sorting by reversals: genome rearrangements in plant organelles and evolutionary history of X chromosome, Mol. Biol. Evol 12 (1995) 239–246.

[3] V. Bafna and P. Pevzner, Sorting by transpositions, in: Proceedings of the 6th Annual ACM–SIAM Symposium on Discrete Algorithms (1995) 614–623.

[4] N.G. Copeland, N.A. Jenkins, D.J. Gilbert, J.T. Eppig, L.J. Maltals, J.C. Miller, W.F. Dietrich, A. Weaver, S.E. Lincoln, R.G. Steen, L.D. Steen, J.H. Nadeau and E.S. Lander, A genetic linkage map of the mouse: current applications and future prospects, Science 262 (1993) 57–65.

[5] S. Hannenhalli, C. Chappey, E. Koonin and P. Pevzner, Genome sequence comparison and scenarios for gene rearrangements: a test case, in: Genomics, Vol. 30 (1995) 299–311.

[6] S. Hannenhalli and P. Pevzner, Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals), in: Proceedings of the 27th Annual ACM Symposium on the Theory of Computing (1995) 178–189.

[7] S. Hannenhalli and P. Pevzner, Transforming men into mice (polynomial algorithm for genomic distance problem), in: Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science (1995) 581–592.

[8] J. Kececioglu and R. Ravi, Of mice and men: evolutionary distances between genomes under translocation, in: Proceedings of the 6th Annual ACM–SIAM Symposium on Discrete Algorithms (1995) 604–613.

[9] J. Kececioglu and D. Sankoff, Exact and approximation algorithms for the inversion distance between two permutations, in: Combinatorial Pattern Matching, Proceedings of the 4th Annual Symposium (CPM'93), Lecture Notes in Computer Science, Vol. 684 (Springer, Berlin, 1993) 87–105; extended version has appeared in Algorithmica 13 (1995) 180–210.

[10] J. Kececioglu and D. Sankoff, Efficient bounds for oriented chromosome inversion distance, in: Combinatorial Pattern Matching, Proceedings of the 5th Annual Symposium (CPM'94), Lecture Notes in Computer Science, Vol. 807 (Springer, Berlin, 1994) 307–325.

[11] B. Lewin, Genes V (Oxford Univ. Press, Oxford, 1994).

[12] D. Sankoff, Edit distance for genome comparison based on non-local operations, in: Combinatorial Pattern Matching, Proceedings of the 3rd Annual Symposium (CPM'92), Lecture Notes in Computer Science, Vol. 644 (Springer, Berlin, 1992) 121–135.

[13] D. Sankoff, R. Cedergren and Y. Abel, Genomic divergence through gene rearrangement, in: Molecular Evolution: Computer Analysis of Protein and Nucleic Acid Sequences, Ch. 26 (Academic Press, New York, 1990) 428–438.

[14] D. Sankoff, G. Leduc, N. Antoine, B. Paquin, B.F. Lang and R. Cedergren, Gene order comparisons for phylogenetic inference: evolution of the mitochondrial genome, Proc. Nat. Acad. Sci. USA 89 (1992) 6575–6579.

[15] E. Therman and M. Susman, Human Chromosomes, Structure, Behavior, and Effects (Springer, Berlin, 1993).