Theoretical Computer Science 491 (2013) 127–135

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Sorting genomes by generalized translocations

Xiao Yin [a], Daming Zhu [b,*]

[a] School of Management Science and Engineering, Shandong University of Finance and Economics, Jinan, 250014, China
[b] School of Computer Science and Technology, Shandong University, Jinan, 250101, China

ARTICLE INFO

Article history:
Received 2 August 2009
Received in revised form 21 January 2011
Accepted 9 January 2013
Communicated by A. Apostolico

Keywords:
Algorithm
Genome rearrangement
Translocation

ABSTRACT

Translocation is a prevalent rearrangement event in the evolution of multi-chromosomal species which exchanges ends between two chromosomes. A translocation is reciprocal if none of the exchanged ends is empty; otherwise, non-reciprocal. The problem of sorting by translocations asks to find a shortest sequence of translocations transforming one genome into another. The problem of sorting by reciprocal translocations can be solved in polynomial-time. Several algorithms have been developed for reciprocal translocation sorting. They can only be applied to a pair of genomes having the same set of chromosome ends. Such a restriction can be removed if non-reciprocal translocations are also allowed. In this paper, we show how to extend the algorithm for sorting by reciprocal translocations to include non-reciprocal translocations, allowing us to compare genomes containing different chromosome ends. We call this problem sorting by generalized translocations. We present a polynomial algorithm for this problem. At a conceptual level, there is some similarity between our algorithm and the algorithm developed by Hannenhalli which is used to sort genomes by reversals and translocations.

1. Introduction

Genome rearrangement is a common mode of molecular evolution in biological species [1–4]. Although the rearrangement process is very complicated, there are three basic operations: *reversal*, *translocation* and *transposition*. In this paper, we study the translocation operations. Translocation is a prevalent rearrangement event in the evolution of multi-chromosomal species which exchanges ends between two chromosomes. A translocation is *reciprocal* if none of the exchanged ends is empty; otherwise, *non-reciprocal*.

The problem of *sorting by translocations* is defined as follows: given two signed multi-chromosomal genomes A and B, find a shortest sequence of translocations transforming A into B. The length of this sequence is called the *translocation distance* between A and B. This problem was first introduced by Kececioglu and Ravi [4]. Hannenhalli designed the first $O(n^3)$ algorithm [5] for *sorting by reciprocal translocations* (abbreviated as SRT) which only allows reciprocal translocations. Bergeron et al. [6] pointed out an error in Hannenhalli's algorithm and gave a new $O(n^3)$ algorithm for SRT. Zhu et al. [7] presented an $O(n^2 logn)$ algorithm for SRT. Wang et al. [8] presented an $O(n^2)$ algorithm for SRT. Recently, the time complexity was improved to $O(n^{3/2}\sqrt{\log(n)})$ by Ozery-Flato and Shamir [9], which is currently the fastest algorithm for SRT.

All of these algorithms for SRT allows only reciprocal translocations. Thus, two genomes compared are assumed to have the same set of chromosome ends, which rarely happens in biology. This restriction is removed if SRT is extended to allow non-reciprocal translocations. In this paper, we study the more general problem of sorting by translocations in which both reciprocal and non-reciprocal translocations are allowed. We call this problem *sorting by generalized translocations*

* Corresponding author. Tel.: +86 531 88390006; fax: +86 531 88392498.
E-mail addresses: yx_018@126.com (X. Yin), dmzhu_of_sdu@hotmail.com, dmzhu@sdu.edu.cn (D. Zhu).

0304-3975/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2013.01.004

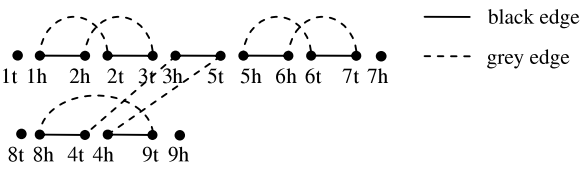

Fig. 1. The cycle graph $G(A, B)$.

(abbreviated as SGT). It was conjectured in [10] that SGT could be solved in polynomial time. Hannenhalli [2] presented an algorithm for sorting genomes by reversals and translocations, using a reduction to the problem of sorting by reversals. Taking a similar approach, we reduce SGT to SRT and present a polynomial-time algorithm for SGT.

2. Preliminaries

This section we provide a basic background for SRT. It follows to a large extent the notation of [5]. In the model, a genome is a set of chromosomes and a chromosome is a sequence of genes. Each gene is identified by an integer with a sign of '+' or '−' which denotes its direction. For example, $\{(3, -5), (2, 4, -6), (-1, 7)\}$ is a genome with three chromosomes and seven genes.

Given a sequence of genes $I = x_1, x_2, \ldots, x_k$, the *reverse* of I is $-I = -x_k, -x_{k-1}, \ldots, -x_1$. A chromosome is orientation-less. Therefore, *flipping* a chromosome X into $-X$ does not affect the chromosomes it represents. Hence, a chromosome X is *identical* to a chromosome Y if either $X{=}Y$ or $X{=}-Y$. Genomes A and B are *identical* if they have the same set of chromosomes.

Let $X{=}(X_1, X_2)$ and $Y{=}(Y_1, Y_2)$ be two chromosomes, where X_1, X_2, Y_1, and Y_2 are sequences of genes. A *translocation* cuts X into X_1 and X_2 and Y into Y_1 and Y_2 and exchanges segments between the chromosomes. There are two types of translocations. A *prefix–prefix* translocation switches X_1 with Y_1 resulting in (Y_1, X_2), (X_1, Y_2). A *prefix–suffix* translocation switches X_1 with Y_2 resulting in $(-Y_2, X_2)$, $(Y_1, -X_1)$. A translocation is *reciprocal* if none of X_1, X_2, Y_1 and Y_2 are empty. Otherwise, it is *non-reciprocal*. There are three types of non-reciprocal translocations: *fusion*, *fission* and *fission–fusion*. They can be regarded as special cases of reciprocal translocations where one or two segments exchanged are empty. A fusion of X and Y connects X and Y into one chromosome (X, Y). It can be viewed as the translocation between $(X, \emptyset)$ and $(\emptyset, Y)$, resulting in (X, Y) and a null chromosome $(\emptyset, \emptyset)$. A fission of X cuts X into two chromosomes (X_1) and (X_2). It can be viewed as the translocation between (X_1, X_2) and $(\emptyset, \emptyset)$. A *fission–fusion* of X and Y cuts X into X_1 and X_2, and then pastes one segment to Y, resulting in (X_1) and (Y, X_2), or (X_1) and $(-X_2, Y)$, or (X_1, Y) and (X_2), or $(Y, -X_1)$ and (X_2). It can be viewed as the translocation between (X_1, X_2) and $(Y, \emptyset)$ or between (X_1, X_2) and $(\emptyset, Y)$.

For a chromosome $X = (x_1, x_2, \ldots, x_m)$, define $\mathit{Tails}(X) = \{x_1, -x_m\}$. x_1 and $-x_m$ are *tails* of X. For a genome A, define $\mathcal{T}(A){=}\bigcup_{X\in A} \mathit{Tails}(X)$. Genomes A and B are *co-tailed* if $\mathcal{T}(A){=}\mathcal{T}(B)$. Note that SRT is solvable only for co-tailed genomes. Readers are referred to [5] for a thoughtful understanding of SRT.

2.1. Cycle graph

Let A and B be a pair of co-tailed genomes. Let n and N be the number of genes and chromosomes in A (equivalently B). We will always assume that both A and B consist of genes $\{1, 2, \ldots, n\}$. For a chromosome $X{=}(x_1, x_2, \ldots, x_m)$, replace each gene x_i by a pair of ordered vertices $(l(x_i), r(x_i))$. If the sign of x_i is '+', then $l(x_i){=}x^t$, $r(x_i){=}x^h$. If the sign of x_i is '−', then $l(x_i){=}x^h$, $r(x_i){=}x^t$. As a result, X corresponds to an ordered list of vertices as

$$l(x_1)r(x_1)l(x_2)r(x_2)\ldots l(x_m)r(x_m).$$

Vertices u and v are *neighbors* in A (B) if they are adjacent in the ordered list of a chromosome in A (B) constructed by the aforementioned method. For any gene x, x^t and x^h are always neighbors. For simplicity, they are excluded from the definition of "neighbors". The bicolored *cycle graph* of A and B, denoted $G(A, B)$, is defined as follows. The set of vertices is $\bigcup_{i=1}^{n}\{i^t, i^h\}$. Vertices u and v are connected by a *black* edge if they are neighbors in A and are connected by a *gray* edge if they are neighbors in B. An example of $G(A, B)$ is given in Fig. 1 where $A = \{(1, -2, 3, 5, -6, 7), (8, 4, 9)\}$, and $B = \{(1, 2, 3, 4, 5, 6, 7), (8, 9)\}$.

A gray edge (u, v) in $G(A, B)$ is *external* if u, v belong to different chromosomes of A, otherwise it is *internal*. Each vertex in $G(A, B)$ has degree 0 or 2, where vertices of degree 0 belong to $\mathcal{T}(A)$ (equivalently, $\mathcal{T}(B)$). Therefore, $G(A, B)$ is uniquely decomposed into vertex-disjoint cycles with alternating black and gray edges. A cycle is *long* if it contains at least two black edges, otherwise *short*. If $A = B$, then all cycles in $G(A, B)$ are short.

2.2. MSP and even-isolation

Considering a sequence $I = x_i, x_{i+1}, \ldots, x_{j-1}, x_j$ in a chromosome of A. Let $V(I) = \bigcup_{i\le k\le j}\{x_k^t, x_k^h\}$, $IN(I) = V(I)\backslash\{l(x_i), r(x_j)\}$. I is a *sub-permutation* (*SP*) if there exists a sequence $I' = x_i, \mathit{permutation}(x_{i+1}, \ldots, x_{j-1}), x_j$ in some chromosome of B and $\mathit{permutation}(x_{i+1}, \ldots, x_{j-1}) \neq (x_{i+1}, \ldots, x_{j-1})$. A *minimal sub-permutation* (*MSP*) is a *SP* not containing any other *SP*. A *SP* I can be viewed as a subgraph of $G(A, B)$ containing the vertex set $IN(I)$ such that: (1) there is no edge (u, v) with