INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

The Genome Rearrangement Distance Problem Using Fusion, Fission, and Transpositions with Arbitrary Weights

Z. Dias and J. Meidanis

Technical Report - IC-02-001 - Relatório Técnico

March - 2002 - Março

The Genome Rearrangement Distance Problem Using Fusion, Fission, and Transpositions with Arbitrary Weights

Zanoni Dias * João Meidanis†

Abstract

Recently we have shown that given two multi-chromossomal genomes it is easy to compute the minimum weight series of fusions, fissions, and transpositions needed to transform one genome into the other, when the weight associated to transpositions is twice as large as that associated to fusions and fissions [17]. In this work we present several results on the computation of distance when an arbitrary weight is associated to transpositions. Some variations of the problem are also studied. For instance we present a polynomial time algorithm for the problem of syntenic distance when only the events of fusions and fissions are admitted.

1 Introduction

In the last years genomic science displayed astonishing advances. Hundreds of genomes, from organisms ranging from viruses and uni-cellular organisms such as bacteria all the way up to human beings, had their gene content uncovered [12]. And the mind-boggling rhythm does not stop: every week new genomes are announced. In this context a new challenge emerges: how to relate the huge quantity of genetic information available nowadays?

The area of Genome Rearrangements has progressed a good deal trying to answer this question. In this area we compare two genomes taking into account the order of this genes, rather than the gene sequence as is done in classical sequence comparison. Some studies, for instance, the one done by Palmer and Herbon [19], show that the comparison of gene order leads to conclusions quite compatible with the real evolutionary scenario of the species.

The main rearrangement events are reversals, transpositions, block-interchange, translocations, fusions, and fissions. A reversal inverts a contiguous region of a genome, and flips the orientation of the genes, in case the genome has this information. A transposition exchanges two adjacent regions of a chromosome, while a block-interchange exchanges two arbitrary, contiguous regions of a chromosome. A translocation exchanges regions from distinct chromosomes. A fusion joins two chromosomes together and a fission divides a chromosome into two new ones.

The main results obtained in this area recently are the following. Hannenhalli and Pevzner presented the first polynomial-time algorithm for the reversal distance problem when gene orientations

*University of Campinas, Institute of Computing, P.O.Box 6167, 13084-971, Campinas, Brazil. Research supported by FAPESP

†University of Campinas, Institute of Computing, P.O.Box 6167, 13084-971, Campinas, Brazil. Research supported in part by CNPq and FAPESP

are known [15], with faster and faster algorithms closely following [2, 16, 9]. Caprara proved that the reversal distance problem is NP-Hard when no information on the orientation of the genes is given [4], and in this case, the best approximation algorithm, with a performance ratio of 1.5, was proposed by Christie [6].

The transposition distance problem has been studied by Bafna and Pevzner [1]. They presented an approximation algorithm for the problem with performance ratio of 1.5. Alternative approximation algorithms were proposed by Christie [7] and Walter, Meidanis, and Dias [21]. The complexity of the transposition distance problem still unknown.

The block-interchange distance problem was proposed and solved by Christie [5]. The distance problem involving translocation, fusion, and fission was solved by Hannenhalli and Pevzner [14].

In a previous work, we presented the first polynomial-time algorithm for distance problems involving transposition [17]. We exhibited an algorithm that finds a minimum cost series of the events fusion, fission, and transposition, when a transposition costs twice as much as a fusion or fission, needed to transform a genome in another. The idea of having a larger weight for transpositions came from the fact that experiments have shown that transpositions occurs with about half the frequency of reversals in real biological instances [3]. The main result of this last work is based on properties of permutation group.

In this work we presente several results on the fusion, fission, and transposition distance problem when the weight associated to transpositions is arbitrary. Variants of the problem are also studied, for instance, the syntenic distance problem when just fusions and fission are allowed. In this case we show a polynomial-time algorithm based on partial results on the syntenic distance using fusions, fissions, and translocations developed by Ferreti and colleagues [11], and later by DasGupta and colleagues [10].

In the following sections we define more formally the problem of computing the fusion, fission, and transposition distance, with emphasis on the case where we associate an arbitrary weight to the transposition event. We present also inequalities involving the versions of the distance with weight two and with another weight associated to transpositions. We then show that the problem of computing the fusion, fission, and transposition distance with arbitrary weight is at least as difficult as computing the transposition distance. We show also a variation of our main problem: the syntenic distance problem with just fusions and fissions. Finally, we present conclusions and ideas for future work.

2 The Fusion, Fission, and Transposition Problem

Before attacking the problem with an arbitrary weight given to transpositions, let us formally define the original problem tacked by Meidanis and Dias [17].

Definition 2.1 *Given two genomes π and σ, denote by $d(\pi, \sigma)$ the weight of a minimum-weight series of fusions, fissions, and transpositions that transform π into σ, when we associate weight 1 to fusions and fissions, and weight 2 to transpositions.*

In their original work, the authors show that $d(\pi, \sigma)$ can be computed in polynomial time, with an $O(n)$ algorithm, where n is the number of genes in the genome π (or in σ).

A natural question is then: what can we say about this problem when an arbitrary weight ω is used instead of 2 for transpositions? The present work answers this question partially.

2.1 Using Arbitrary Weights for Transpositions

Here we consider the fusion, fission, and transposition distance problem between two genomes when the weights associated to the mutational events are 1, 2, and ω, respectively. In this case we denote by $d_\omega(\pi,\sigma)$ the distance between the two genomes.

In the sequel we show a few basic facts about the relation between $d(\pi,\sigma)$ and $d_\omega(\pi,\sigma)$.

Lemma 2.1 *For $\omega > 2$ we have:*

$$d_\omega(\pi,\sigma) = d(\pi,\sigma).$$

Proof: It is easy to see that every transposition can be replaced by a fission-fusion pair. Therefore, every optimal series of events when transpositions have weight $\omega > 2$ is composed solely by fusions and fissions. But by results of Meidanis and Dias [17], every optimal series of fusions and fissions that transform π into σ has weight $d(\pi,\sigma)$. ■

In the sequel we will show some properties of $d_\omega(\pi,\sigma)$.

Lemma 2.2 *For $0 < \omega \leq 2$ we have:*

$$d_\omega(\pi,\sigma) \leq d(\pi,\sigma).$$

Theorem 2.1 *For $0 < \omega \leq 2$ we have:*

$$d(\pi,\sigma) \leq \frac{2}{\omega} d_\omega(\pi,\sigma).$$

Proof: Let ρ_1, ρ_2, ..., ρ_k be an optimal series of events such that:

$$\rho_k \rho_{k-1} \ldots \rho_1 \pi = \sigma,$$

considering weight ω for transpositions. If $n_\tau(\rho_1,\rho_2,\ldots,\rho_k)$ is the number of transpositions in the optimal series ρ_1, ρ_2, ..., ρ_k and $n_{ff}(\rho_1,\rho_2,\ldots,\rho_k)$ is the number of fusions and fissions in this series, we can write:

$$d_\omega(\pi,\sigma) = n_{ff}(\rho_1,\rho_2,\ldots,\rho_k) + \omega n_\tau(\rho_1,\rho_2,\ldots,\rho_k)$$

However, we can also write:

$$d(\pi,\sigma) \leq n_{ff}(\rho_1,\rho_2,\ldots,\rho_k) + 2n_\tau(\rho_1,\rho_2,\ldots,\rho_k), \tag{1}$$

because, after all, ρ_1, ρ_2, ..., ρ_k is one possible series of events leading from π to σ. Multiplying Equation 1 by $\omega/2$ we get, successively,

$$\begin{aligned} \frac{\omega}{2} d(\pi,\sigma) &\leq \frac{\omega}{2} n_{ff}(\rho_1,\rho_2,\ldots,\rho_k) + \omega n_\tau(\rho_1,\rho_2,\ldots,\rho_k) \\ &\leq n_{ff}(\rho_1,\rho_2,\ldots,\rho_k) + \omega n_\tau(\rho_1,\rho_2,\ldots,\rho_k) \\ &= d_\omega(\pi,\sigma), \end{aligned}$$

since $\omega/2 < 1$. ■

From Lemma 2.2 and Theorem 2.1 we have the following result.

Theorem 2.2 *For $0 < \omega \leq 2$ we have:*

$$d_\omega(\pi,\sigma) \leq d(\pi,\sigma) \leq \frac{2}{\omega} d_\omega(\pi,\sigma)$$

Therefore, we may conclude that the algorithm for the fusion, fission, and transposition (with weight 2 for transpositions) presented by Meidanis and Dias [17] is an approximation algorithm with performance ratio $\frac{2}{\omega}$ for the problem where the weight of a transposition is ω, with $0 < \omega < 2$.

The next theorem indicates a sufficient condition for a transposition with weight ω, used on an optimal series that transforms π into σ, to be used on an optimal series that transforms π into σ when transpositions have weight 2.

Theorem 2.3 *Let π and σ be two genomes, and τ a transposition. If $d_\omega(\pi,\sigma) = d_\omega(\tau\pi,\sigma) + \omega$, then $d(\pi,\sigma) = d(\tau\pi,\sigma) + 2$ when $d(\pi,\sigma) < \frac{1}{(2-\omega)}$ and $1 < \omega < 2$.*

Proof: Assume that $d_\omega(\pi,\sigma) \geq d_\omega(\tau\pi,\sigma) + \omega$. We have then:

$$\begin{aligned} d(\pi,\sigma) &\geq d_\omega(\pi,\sigma) \\ &\geq d_\omega(\tau\pi,\sigma) + \omega \\ &\geq \frac{\omega d(\tau\pi,\sigma)}{2} + \omega \end{aligned}$$

Rewriting,

$$2d(\pi,\sigma) \geq \omega d(\tau\pi,\sigma) + 2\omega$$
$$d(\pi,\sigma) \geq \omega d(\tau\pi,\sigma) - d(\pi,\sigma) + 2\omega$$
$$d(\pi,\sigma) \geq d(\tau\pi,\sigma) + (\omega-1)d(\tau\pi,\sigma) - d(\pi,\sigma) + 2\omega$$

But, by hypothesis, we have:

$$d(\pi,\sigma) < \frac{1}{(2-\omega)}$$
$$(2-\omega)d(\pi,\sigma) < 1$$
$$(\omega-2)d(\pi,\sigma) > -1$$
$$(\omega-2)d(\pi,\sigma) + 2 > 1$$
$$((\omega-1)d(\pi,\sigma) - d(\pi,\sigma)) + (2\omega - 2(\omega-1)) > 1$$
$$(\omega-1)(d(\pi,\sigma)-2) - d(\pi,\sigma) + 2\omega > 1$$

But for every transposition τ the following holds:

$$d(\tau\pi,\sigma) \geq d(\pi,\sigma) - 2$$

Then:

$$(\omega-1)d(\tau\pi,\sigma) - d(\pi,\sigma) + 2\omega > 1$$

and, since $d(\pi,\sigma)$ is an integer, we have:

$$\begin{aligned} d(\pi,\sigma) &\geq d(\tau\pi,\sigma) + (\omega-1)d(\tau\pi,\sigma) - d(\pi,\sigma) + 2\omega \\ &> d(\tau\pi,\sigma) + 1 \\ &\geq d(\tau\pi,\sigma) + 2 \end{aligned}$$

■

3 Relationship Between Evolutionary Distance Problems

In this section we will show a relationship between the distance problem involving fusion, fission, and transpositions when the weight associated to transpositions is part of the input, and the (pure) transposition distance problem. The transposition problem has been studied intensely in the last years [1, 18, 7, 20], but its computational complexity is still unknown.

Theorem 3.1 *The distance problem involving fusions, fissions, and transpositions when the weight of transpositions is part of the input is NP-Hard if the transposition distance problem is NP-Hard.*

Proof: We will show that it is possible to polynomially reduce the transposition distance problem to the distance problem involving fusion, fission, and arbitrarity-weighted transposition (DPFFWT).

Given an instance of the transposition distance problem consisting of two genomes π and σ, we will build an instance for the DPFFWT as follows. We use the very genomes π and σ as part of the input, and select $\omega = 1/n$, as the weight to given to transpositions, where n is the number of genes in π (or in σ).

We known that given two genomas A and B with one n-gene chromosome each, it is possible to transform one into the other with at most $n - 1$ transpositions [1]. Therefore, we can solve our instance using transpositions alone at a cost $d_\omega(\pi, \sigma) < 1$, and no fusions or fissions will be used.

The series of events obtained for the DPFFWT is also an optimal series for the transposition distance problem, because of the value chosen for ω. The value of the transposition distance can be obtained as follows: $d_\tau(\pi, \sigma) = nd_\omega(\pi, \sigma)$.

Therefore, we may conclude that if the transposition distance problem is NP-Hard, so is the DPFFWT. ■

Two observations are in order with respect to the value of ω. First, notice that any series of events transforming a mono chromosomal genome π into another monochromosomal genome σ using at least a fusion or a fission will have weight at least 2. Hence, we could have chosen $\omega = 2/n$. Second, if the conjecture by Meidanis, Walter e Dias [18] about the transposition diameter is correct, we would have $D_\tau = \lfloor (n-1)/2 \rfloor + 1$, and hence any value for ω such that $0 < \omega < 4/(n+1)$ would suffice.

In the next section we treat the problem where transpositions have zero weight. Notice that in this case any transformation affecting only one chromosome has cost zero. Therefore, we are in fact interested in guaranteeing that the two genomes have the same sets of genes as chromosomes, regardless of the order of these genes in the set. In other words, we are talking about the synteny problem using only the events of fusion and fission.

4 The Syntenic Distance Problem

Ferretti and coworkers [11] proposed a distance measure with a high degree of abstraction, where the order of genes in a particular chromosome is unknown or ignored. The genome of a species is then just a collection of gene sets. Each set correspond to a chromosome. In this synteny context a gene may occur several times in a genome. We define two types of operation: fusion and fission. Fusion correspond to set union, and fission to division of a set A into B and C such that $A = B \cup C$.

Notice that B and C may have genes in common. Originally the problem was proposed with a third operation, translocation, which exchanges subsets of two chromosomes.

The *syntenic distance* between two genomes in our context is the minimum number of fusions and fissions necessary to transform the genome of a species into the genome of the other species. We denote by $d_{syntenic}(\pi, \sigma)$ the syntenic distance between genomes π and σ.

Observe that, given two genomes, it is always possible to transform one into the other using only fusions and fissions, when both genomes have the same gene set. The justification of this model is as follows: for many organisms the information that specifies the gene order in a chromosome (physical map) is not known, but the distribution of genes in each chromosome is. Even with such incomplete information it is important to have a precise definition of an evolutionary distance based on genomic events, and the syntenic distance, or just synteny, provides this definition.

DasGupta and colleagues [10] studied the synteny problem when fusions, fissions, and translocations are permitted. They have proved that this problem is NP-Hard, and showed an approximation algorithm with a factor 2. They have also proved that the median problem for three genomes using synteny with the three operations mentioned above is NP-Hard, and that in this case it is possible to obtain approximation algorithms with factor $4 + \epsilon$ for any $\epsilon > 0$. Several of the results presented here on synteny are an adaptation of results from DasGupta et. al. [10] for the problem when only fusions and fission are allowed.

Lemma 4.1 *Let π and σ be two genomes with the same gene set. Then we have $d_{syntenic}(\pi, \sigma) = d_{syntenic}(\sigma, \pi)$.*

Proof: Given a series of events transforming π into σ it is easy to revert each operation (the reverse of a fusion is a fission and vice-versa) to obtain a series of operations from σ into π. Hence, the minimum series has the same length in both directions. ■

4.1 The Compact Representation

Ferretti, Nadeau and Sankoff [11] defined a compact representation for the synteny problem. Given two genomes π and σ it is possible to obtain a compact representation of the problem with respect to π using the following method. For each chromosome π_i of the genome π create chromosome $\pi'_i = \{i\}$ in π'. For each chromosome σ_j of σ create $\sigma'_j = \bigcup_{x \in \sigma_j} \{y | x \in \pi_y\}$.

For instance, let $\pi = \{\pi_1, \pi_2, \pi_3\}$ and $\sigma = \{\sigma_1, \sigma_2\}$ with:

$$\pi_1 = \{a, b, c\}, \pi_2 = \{b, d, e\}, \pi_3 = \{f, g\}$$

$$\sigma_1 = \{b, c, d, f, g\}, \sigma_2 = \{a, b, e\}$$

The compact representation of the problem with respect to π is:

$$\pi' = \{\{1\}, \{2\}, \{3\}\}$$

$$\sigma' = \{\{1, 2, 3\}, \{1, 2\}\}$$

Analogously we can define the compact representation with respect to σ. For each chromosome π_i of π, create $\pi''_i = \bigcup_{x \in \pi_i} \{y | x \in \sigma_y\}$. For each chromosome σ_j of σ create $\sigma''_j = \{j\}$ in σ''. The problem above becomes:

$$\pi'' = \{\{1,2\},\{1,2\},\{1\}\}$$

$$\sigma'' = \{\{1\},\{2\}\}$$

The following results have been proved by DasGupta et. al. [10].

Lemma 4.2 *Let π' and σ' be the two genomes that form the compact representation of π and σ with respect to π. There is a $1-1$ mapping between each operation (fusion or fission) used to transform π into σ and each operation used to transform π' into σ', that is, $d_{syntenic}(\pi,\sigma) = d_{syntenic}(\pi',\sigma')$.*

Given two genomes π and σ, the problem involving the compact representation with respect to π and the problem involving the compact representation with respect to σ are called **dual problems**. The following result shows the relationship between dual problems:

Lemma 4.3 *Let π and σ be two genomes over the same set of genes. Let π' and σ' be their compact representation with respect to π, and π'', σ'' be their compact representation with respect to σ. Then $d_{syntenic}(\pi',\sigma') = d_{syntenic}(\pi'',\sigma'')$.*

DasGupta and colleagues [10] have shown also a algorithm to construct the compact representation with respect to a given genome.

Lemma 4.4 *If π and σ have n and m chromosomes, respectively, and if each chromosome is a subset of $\{1,2,\ldots,k\}$, then it is possible to construct the compact representation with respect to π (or with respect to σ) in time $O((k+nm)\alpha(k,n+m))$, where $\alpha(x,y)$ is the inverse of Ackerman's function [8].*

The function $\alpha(x,y)$ grows very slowly, and therefore it is reasonable to expect the algorithm to exhibit a $O(k+nm)$ behaviour in practice.

We define the synteny problem using the compact representation as follows:

Definition 4.1 *Let π be a genome with k chromosomes with each chromosome being a subset of $\{1,2,\ldots,n\}$. The synteny problem is to compute the minimum number of fusion and fission, denoted by $d_{syntenic}(\pi)$, needed to transform π into the genome $\{\{1\},\{2\},\ldots,\{n\}\}$.*

4.2 The Canonical Order

The synteny distance problem has an important characteristic, rarely found in genome rearrangement problems: a canonical order for the mutation events.

Lemma 4.5 *Let ρ_1,ρ_2, …, ρ_k be any series of events that transforms π into σ. Then there is a series transforming π into σ, using the same number of fusions and the same number of fissions as ρ_1,ρ_2, …, ρ_k, but where all fusions occur before the fissions.*

Proof: Let ρ_1,ρ_2, ..., ρ_k an sequence of events that transforms π into σ. If all fusions occur before the fissions there is nothing to be done. Let us then assume that there is $i < k$ such that i is the largest integer with ρ_i being a fission and ρ_{i+1} being a fusion. We will construct a new optimal series ρ_1,ρ_2, ..., ρ_{i-1}, ρ'_i, ρ'_{i+1}, ρ_{i+2}, ..., ρ_k, with ρ'_i being a fusion and ρ'_{i+1} being a fission. Repeating this procedure as many times as needed we obtain the desired series.

Suppose that fission ρ_i transforms chromosome A of genome $\rho_{i-1}\rho_{i-2}\ldots\rho_1\pi$ into A' e A'', with $A = A' \cup A''$. Likewise, suppose that fusion ρ_{i+1} transforms chromosomes B' and B'' of the genome $\rho_i\rho_{i-1}\ldots\rho_1\pi$ into B, with $B = B' \cup B''$. We have three cases:

- If each of the two chromosomes A' and A'' created by fission ρ_i is different from both B' and B'', then we take $\rho'_i = \rho_{i+1}$ and $\rho'_{i+1} = \rho_i$, since the two events are interchangeable.
- If A' and A'' are the same as B' and B'', then ρ_i and ρ_{i+1} are inverses of each other and, again, are interchangeable.
- In the remaining case one of the chromosomes A', A'' is equal to one of B', B''. Without loss of generality, suppose $A' = B'$. We have then that the net effect of ρ_i plus ρ_{i+1} is to transform chromosomes A and B'' of genome $\rho_{i-1}\rho_{i-2}\ldots\rho_1\pi$ into chromosomes A'' and B of genome $\rho_{i+1}\rho_i\ldots\rho_1\pi$. This effect can be also obtained by taking as ρ'_i th fusion that transforms A and B'' into $A \cup B''$, and as ρ'_{i+1} the fission that transforms $A \cup B''$ into A'' and B. This last fission is a valid operation because $A'' \cup B$ = $A'' \cup B' \cup B''$ = $A'' \cup A' \cup B''$ = $A \cup B''$.

■

4.3 Lower Bound

In this section we will show a lower bound for the synteny problem using a data structure named synteny graph.

Definition 4.2 *Given a genome π, the synteny graph $G_{syntenic}(\pi)$ has one vertex for each chromosome of π. Two vertices are adjacent if and only if the corresponding chromosomes have a nonempty intersection.*

Lemma 4.6 *Let π be an arbitrary genome with n genes and p the number of connected components of $G_{syntenic}(\pi)$. Then at least $n - p$ fissions are necessary to transform π into ι_n.*

Proof: By definition $G_{syntenic}(\iota_n)$ has n connected components, and therefore any series of events that transforms π into ι_n must increase the number of connected components by $n - p$. We need to determine how the mutation events affect the synteny graph. A fusion merges two vertices into a single one. If the two vertices are in the same connected component, the number of connected components does not change. If the vertices are in distinct components, then the number of connected components will decrease by one. A fission is the opposite of a fusion, and therefore the number of connected components will either remain the same or increase by one. We conclude that at least $n - p$ fissions are necessary to transform $G_{syntenic}(\pi)$ into a graph with n connected components.
■

Lemma 4.7 *Consider a genome π with n genes and c chromosomes. Let p be the number of connected components of the graph $G_{syntenic}(\pi)$. Then at least $c - p$ fusions are needed to transform π into ι_n.*

Proof: Let $\rho_1, \rho_2, \ldots, \rho_k$ be any series of events transforming π into ι_n with l fusions. According to Lemma 4.5, it can be chosen so that the first l events are fusions and the remaining $k - l$ events are fissions.

Then $\pi_l = \rho_l \rho_{l-1} \ldots \rho_1 \pi$ is a genome that can be transformed into ι_n using fissions only, that is, π_l cannot contain chromosomes A and B with $A \cap B \neq \emptyset$. It follows that every connected component of $G_{syntenic}(\pi_l)$ is composed of a single, isolated vertex. Since fusions do not increase the number of components, the number of connected components in $G_{syntenic}(\pi_l)$ is at most p.

A fusion always decreases the number of vertices by one. We start with c vertices and, after all fusions are applied, we end up with at most p vertices (one vertex per component). It follows that $l \geq c - p$. ■

Combining Lemmas 4.6 and 4.7 we can state the following theorem proposing a lower bound for the syntenic distance using the events of fusion and fission.

Theorem 4.1 *Consider a genome π with n genes and c chromosomes. Let p be the number of connected components of $G_{syntenic}(\pi)$. Then $d_{syntenic}(\pi) \geq n + c - 2p$.*

4.4 The Polynomial-Time Algorithm

In the sequel we exhibit an algorithm that computes the syntenic distance and yields an optimal series of events that transforms a genome π with n genes into ι_n.

Theorem 4.2 *Consider a genome π with n genes and c chromosomes. Let p be the number of connected components of the graph $G_{syntenic}(\pi)$. Then $d_{syntenic}(\pi) = n + c - 2p$ and it is possible to obtain an optimal series of events transforming π into ι_n in $O(n^2 + nc\alpha(nc, n))$ time.*

Proof: According to previous results, the algorithm in Figure 1 produce a series of events that transforms π into ι_n. Let us compute the number of fusions and fissions used. Each fusion decreases the number of chromosomes by one. Initially, π contains c chromosomes and after all fusions are applied we end up with exactly p chromosomes (one for each component). Therefore we used $c - p$ fusions. Regarding the fissions, each one creates a new chromosome. Because at the end of the algorithm we have n chromosomes, the number of fissions is $n - p$, which implies that $d_{syntenic}(\pi) \leq n + c - 2p$. Using Theorem 4.1, we conclude that this is in an exact algorithm for the synteny problem.

The most time-consuming step of the algorithm is the one that determines the connected components of the synteny graph (line 4), without explicitly constructing the graph. This step can be implemented with a union-find structure, using union-by-rank and path compression, in time $O(nc\alpha(nc, n))$ [8]. We construct the chromosome lists of each component in time proportional to the sum of the component sizes, that is, in $O(nc)$ time. Each iteration of the loop on lines 5-11 takes $O(n)$ time, and therefore the entire loop takes $O(nc)$ time, since it is executed $O(c)$ times. Likewise, each iteration of the last loop (lines 12-18) takes $O(n)$ and the total time is $O(n^2)$ because it is executed $O(n)$ times. We conclude that the algorithm runs in $O(n^2 + nc\alpha(nc, n))$ time. ■

```
SYNTENIC DISTANCE()
Input: π, n
n_fusions ← 0
n_fissions ← 0
Determine the connected components C_1, C_2, ..., C_p of G_syntenic(π)
for i ← 1 to p
do while |C_i| > 1
    do ρ ← any fusion involving two chromosomes X and Y of C_i
        π ← ρπ
        n_fusions ← n_fusions + 1
        Print ρ
        Remove X and Y and add X ∪ Y to C_i
for j ← 1 to p
do while C_j has a chromosome X with more then one gene
    do ρ ← any fission of X into disjoint parts A and B
        π ← ρπ
        n_fissions ← n_fissions + 1
        Print ρ
        Remove X and add A and B to C_j
Output: n_fusions + n_fissions
```

Figure 1: An algorithm for syntenic distance.

4.5 The Synteny Problem with Indistinguishable Genes

In the sequel we define a variation for the synteny problem. Here we do not know the order of the genes, nor do we have sufficient information to identify which genes are in which chromosomes. All we know is the number of genes in each chromosome. A chromosome will be represented simply by an integer, and a genome π will be a set of integers (with multiplicity), with $|\pi|$ indicating the number of chromosomes.

A fusion acting on two chromosomes with r and s genes, transforms them into a new chromosome with $t = r + s$ genes. A fission acting on a chromosome with t genes transforms it into two new chromosomes with r and s genes ($t = r + s$).

This model, where we do not have qualitative information on the genes, but only their quantity, is compatible with hybridization experiments involving promoters [13]. These experiments are a fast and simple way of obtaining a good idea on the number of genes in a chromosome.

We define syntenic distance problem between two genomes π and σ with indistinguishable genes as the smallest number of mutation events (fusions and fissions) that transform π into σ, and we denote this distance by $\overline{d}_{syntenic}(\pi, \sigma)$.

Lemma 4.8 *Given two arbitrary genomes π and σ, we have $\overline{d}_{syntenic}(\pi, \sigma) \geq | \, |\pi| - |\sigma| \, |$.*

Proof: If $|\pi| > |\sigma|$ at least $|\pi| - |\sigma|$ fusions are needed to transform π into σ, since each fusion decreases the number of chromosomes by one. Likewise, if $|\pi| > |\sigma|$ then at least $|\sigma| - |\pi|$ fissions are needed to transform π into σ, since each fission increases the number of chromosomes by one. ■

Lemma 4.9 *Given two arbitrary genomes π and σ, we have $\overline{d}_{syntenic}(\pi,\sigma) \leq |\pi| + |\sigma| - 2$.*

Proof: We can easily transform π into σ using the following algorithm: merge all chromosomes of π into one single chromosome. With $|\pi| - 1$ fusions we can accomplish this transformation. Then apply a series of fission so that the chromosomes of σ are created. For this task $|\sigma| - 1$ fissions suffice. In this way it is possible to transform π into σ using $|\pi| + |\sigma| - 2$ events. ■

Theorem 4.3 *Given two genomes π and σ, we have:*

$$|\,|\pi| - |\sigma|\,| \leq \overline{d}_{syntenic}(\pi,\sigma) \leq |\pi| + |\sigma| - 2.$$

Theorem 4.4 *The syntenic distance problem with indistinguishable genes is NP-Hard.*

Proof: We will show that we can reduce the partition problem polynomially to the syntenic distance problem with indistinguishable genes.

The partition problem can be defined as follows: given a set $A = \{a_1, a_2, \ldots, a_n\}$ of positive integers, determine whether there exists a way of partitioning A into two subsets $B = \{b_1, b_2, \ldots, b_l\}$ and $C = \{c_1, c_2, \ldots, c_m\}$ such that $\sum_{i=1}^{l} b_i = \sum_{j=1}^{m} c_j$.

If $\sum_{i=1}^{n} a_i$ is odd the problem becomes trivial since it is impossible to obtain a suitable partition, but if the sum is even the problem is NP-Hard.

Let $A = \{a_1, a_2, \ldots, a_n\}$ be a set such that $\sum_{i=1}^{n} a_i$ is even. We can construct an instance of the syntenic distance problem with indistinguishable genes as follows: take $\pi = A$ and $\sigma = \{\sigma_1, \sigma_2\}$ where $\sigma_1 = \sigma_2 = (\sum_{i=1}^{n} a_i)/2$.

By Theorem 4.3, we have $n - 2 \leq \overline{d}_{syntenic}(\pi,\sigma) \leq n$. Observe that $\overline{d}_{syntenic}(\pi,\sigma) \neq n - 1$, because Lemma 4.8 says that $n - 2$ fusions are necessary; if the extra event is a fusion, we end up with 1 chromosome; if it is a fission, we end up with 3 chromosomes; but σ has 2 chromosomes.

The final part of the proof is to show that a solution for this instance of the syntenic distance problem with indistinguishable genes corresponds to a solution of the original partition problem. Notice that, in this context, the ability to partition A into two subsets of equal sum is equivalent to being able to transform π into σ using fusions only.

If $\overline{d}_{syntenic}(\pi,\sigma) = n-2$ then it is possible to partition A as desired, since π can be transformed into σ using fusions only. In contrast, if $\overline{d}_{syntenic}(\pi,\sigma) = n$, it is impossible to find a suitable partition, because a fission was necessary to transform π into σ. ■

5 Conclusion

We have shown in this work results about the rearrangement distance using fusion, fissions, and transpositions when an arbitrary weight is associated to transpositions (see Table 1). We have proved that the distance problem using fusions, fissions, and transpositions with the transposition weight given as input is at least as hard as the transposition distance problem, which is still open. Finally, we have determined the complexity of two variations on the syntenic distance problem.

Problem	**Result**
Fusion, Fission, and Transposition Distance ($\omega = 2$)	$O(n^2)$ [17]
Fusion, Fission, and Transposition Distance ($0 < \omega < 2$)	Factor $\frac{2}{\omega}$ approx. [HERE]
Synteny with Fusion, Fission, and Translocation	NP-Hard + factor 2 approx. [10]
Synteny with Fusion and Fission (distinguishable genes)	$O(nc\alpha(nc, n))$ [HERE]
Synteny with Fusion and Fission (indistinguishable genes)	NP-Hard [HERE]

Table 1: Problems and result related to the present work.

References

[1] V. Bafna and P. A. Pevzner. Sorting by transpositions. *SIAM Journal on Discrete Mathematics*, 11(2):224–240, May 1998.

[2] P. Berman and S. Hannenhalli. Fast sorting by reversal. In D. S. Hirschberg and E. W. Myers, editors, *Proceedings of Combinatorial Pattern Matching (CPM'96), 7th Annual Symposium*, volume 1075 of *Lecture Notes in Computer Science*, pages 168–185, Laguna Beach, USA, 1996. Springer.

[3] M. Blanchette, T. Kunisawa, and D. Sankoff. Parametric genome rearrangement. *Journal of Computational Biology*, 172:11–17, 1996.

[4] A. Caprara. Sorting permutations by reversals and eulerian cycle decompositions. *SIAM Journal on Discrete Mathematics*, 12(1):91–110, 1999.

[5] D. A. Christie. Sorting permutations by block-interchanges. *Information Processing Letters*, 60(4):165–169, 1996.

[6] D. A. Christie. A 3/2-approximation algorithm for sorting by reversals. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 244–252, San Francisco, USA, 1998.

[7] D. A. Christie. *Genome Rearrangement Problems*. PhD thesis, Glasgow University, 1998.

[8] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. McGraw-Hill, 1990.

[9] B. M. E. Moret D. A. Bader and M. Yan. A linear-time algorithm for computing inversion distance between signed permutations with an experimental study. In *Proceedings of the Seventh Workshop on Algorithms and Data Structures (WADS'01)*. Springer Verlag, 2001.

[10] B. DasGupta, T. Jiang, S. Kannan, M. Li, and E. Sweedyk. On the complexity and approximation of syntenic distance. *Discrete Applied Mathematics, Second Special Issue on Computational Biology*, 88:59–82, 1998.

[11] V. Ferretti, J. H. Nadeau, and D. Sankoff. Original synteny. In D. S. Hirschberg and E. W. Myers, editors, *Proceedings of Combinatorial Pattern Matching (CPM'96), 7th Annual Symposium*, volume 1075 of *Lecture Notes in Computer Science*, pages 159–167, Laguna Beach, USA, 1996. Springer.

[12] TIGR The Institute for Genomic Research, October 2001. Website: www.tigr.org.

[13] D. Frishmana, A. Mironov, and M. Gelfand. Starts of bacterial genes: Estimating the reliability of computer predictions. *Gene*, 234:257–265, 1999.

[14] S. Hannenhalli and P. A. Pevzner. Transforming men into mice (polynomial algorithm for genomic distance problem). In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS'95)*, pages 581–592, Los Alamitos, USA, October 1995. IEEE Computer Society Press.

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

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

[17] J. Meidanis and Z. Dias. Genome rearrangements distance by fusion, fission, and transposition is easy. Technical Report IC-01-07, Institute of Computing - University of Campinas, 2001.

[18] J. Meidanis, M. E. Walter, and Z. Dias. Transposition distance between a permutation and its reverse. In R. Baeza-Yates, editor, *Proceedings of the 4th South American Workshop on String Processing (WSP'97)*, pages 70–79, Valparaiso, Chile, 1997. Carleton University Press.

[19] J. D. Palmer and L. A. Herbon. Plant mitochondrial dna evolves rapidly in structure, but slowly in sequence. *Journal of Molecular Evolution*, 27:87–97, 1988.

[20] M. E. Walter. *Algoritmos para Problemas em Rearranjo de Genomas*. PhD thesis, University of Campinas, Brazil, 1999. In Portuguese.

[21] M. E. Walter, Z. Dias, and J. Meidanis. A new approach for approximating the transposition distance. In *Proceedings of the String Processing and Information Retrieval: A South American Symposium (SPIRE'2000)*, 2000.