TRACTABILITY RESULTS FOR THE DOUBLE-CUT-AND-JOIN MULTICHROMOSOMAL MEDIAN PROBLEM

by

Ahmad Mahmoody-Ghaidary

B.Sc., Sharif University of Technology, 2009

A Thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science
in the School
of
Mathematics

SIMON FRASER UNIVERSITY
Summer 2011

APPROVAL

Name: Ahmad Mahmoody-Ghaidary

Degree: Master of Science

Title of Thesis: Tractability results for the Double-Cut-and-Join multichromosomal median problem

Examining Committee: Dr. Matt DeVos, Assistant Professor
Mathematics, SFU
Chair

Dr. Cedric Chauve, Associate Professor
Mathematics, SFU, Supervisor

Dr. Ladislav Stacho, Associate Professor
Mathematics, SFU, Supervisor

Dr. Bojan Mohar, Professor
Mathematics, SFU, Examiner

Dr. Pavol Hell, Professor
Computing Science, SFU, External Examiner

Date Approved: August 8th, 2011

Abstract

Genomes can be modeled by sets of adjacencies between genomic markers. There are different ways of measuring the dissimilarities between pairs of genomes, in term of genomic rearrangements. The most widely used dissimilarities are distance functions on genomes. In the present work, we consider the *Double-Cut-and-Join* (DCJ) distance model, denoted by d_{DCJ}. A DCJ median of three genomes G_1, G_2, and G_3 is a genome M which minimizes the sum $d_{\mathrm{DCJ}}(M, G_1) + d_{\mathrm{DCJ}}(M, G_2) + d_{\mathrm{DCJ}}(M, G_3)$. The problem of computing a DCJ median has been shown to be NP-hard. Currently, very few tractability result exist for this problem.

The *breakpoint graph* is a fundamental combinatorial object for modeling and studying genomes. For example, the DCJ distance of two genomes can be obtained from the following parameters of their breakpoint graph: (i) number of vertices, (ii) number of cycles, and (iii) number of odd paths (paths with an odd number of vertices). Also finding a DCJ median for three genomes is equivalent to finding a matching in their breakpoint graph which maximizes the total number of *alternating* cycles and half number of odd paths.

The maximum degree of a breakpoint graph of three genomes is at most 3. So finding such matching is NP-hard. We prove in this thesis that if the maximum degree is 2, the DCJ median problem is tractable. Therefore, hardness of the problem is due to the vertices of degree 3. Additionally, we introduce an FPT algorithm for the problem when the number of vertices of degree 3 is bounded.

Keywords: Modeling Genomes, Genomic Distance, Median Problem, Double-Cut-and-Joint (DCJ), Genomic Rearrangement

Acknowledgments

I would like to start by thanking my great supervisors Cedric Chauve and Ladislav Stacho. Cedric introduced me to the field of computational biology when I knew nothing about it, and changed my math taste. Our great discussions with Ladislav and his comments on the results were very useful. I am very grateful to them for all of their wonderful support during my masters.

I thank Matt Devos for the enjoyable time of being his student at combinatorics class and being his TA. I also thank Andrei Bulatov and Kay Wiese at department of computing science, for their nice algorithm and bioinformatics classes, where I learnt a lot.

Being at SFU was terrific! Great faculty, staff, and friends. In particular, I would like to thank Diane Pogue, our graduate secretary.

Finally, I thank my family, particularly, my beloved wife, Zainab, for their extreme love and support.

Contents

Chapter 1

Introduction

In this chapter we introduce notions related to genomic evolution and their mathematical modeling. For a comprehensive introduction to molecular evolution, we refer the reader to [14].

1.1 Genome evolution

In 1859 Charles Robert Darwin introduced his theory of *evolution* in his book "*On the origin of Species*" [11]. According to this theory, all living organisms descend from a common ancestor, and this divergence from the common ancestor follows from a process called *natural selection*. Evolution was accepted as a fact by 1870s in the scientific community. For example, in 2008, Bousseau *et al.* were able to infer computationally properties of this *last universal common ancestor* (LUCA) from the study of current genomes[6]. Here we focus on the molecular aspect of evolution, i.e., genome evolution.

1.1.1 Genomes and genes

We start with some terminology from molecular biology that will help us explain our problem. *Deoxyribose nucleic acid*, or DNA, is a *nucleic acid*, formed from four bases called *nucleotides*: **A**denine, **C**ytosine, **G**uanine, and **T**hymine (we use their first

letters as their abbreviation: **A, C, G, T**).

A *chromosome* is a molecule of DNA. It is *linear* if its bases form a linear chain whose ends are not connected (Fig. 1.1 (a), (b)), and it is *circular* if the bases form a circular chain (Fig. 1.1 (c), (d)).

We say a chromosome is *single-stranded* if it is a chain of bases (Fig. 1.1 (a), (c)). It is *double-stranded* if it consists of two chains such that each base on a chain pairs with a base on the other chain in the following way: **A** pairs with **T**, and **C** pairs with **G** (Fig. 1.1 (b), (d)). The base pairs **A-T** and **C-G** are *complementary* base pairs.

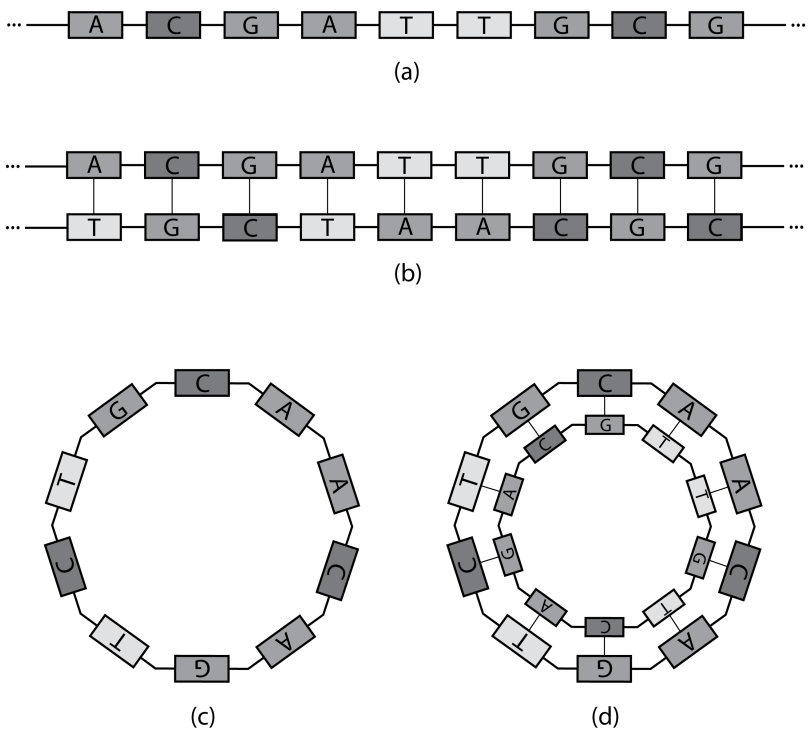

Figure 1.1: Different forms of chromosome (a) linear single-stranded, (b) linear double-stranded, (c) circular single-stranded, (d) circular double-stranded.

All the information required for the development and the functioning of a living organism are stored in its *genome*. A genome is a set of chromosomes. A genome is *uni-chromosomal* if it contains only one chromosome, otherwise it is *multi-chromosomal*.

A *circular genome* (resp. *linear genome*) is a genome such that all of its chromosomes are circular (resp. linear). A genome is *mixed* if it has both linear and circular chromosomes.

Fundamental units of information in a genome are the *genes* it contains. Each gene is a chromosome segment, and encodes the information for the synthesis of proteins in the cells. In 1920 Hans Winker introduced the term "**genome**" as a composition of the words **gen**e and chromos**ome**. In Fig. 1.2 a schematic genome is represented. Note that there are places in chromosomes which are not included in any gene.

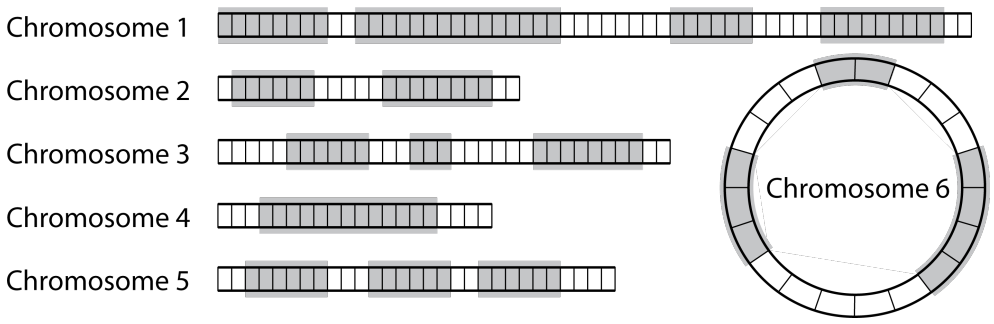

Figure 1.2: A schematic genome with five linear chromosomes and one circular. Genes are shown in gray.

1.1.2 Genome evolution

According to Darwin's theory, all living species are descendant of a common ancestor, meaning during time, new species appear by diverging from previously existing species. As it was mentioned before, this thesis focuses on the evolution of genomes.

An evolutionary event is called *speciation*. Changes in genomes that lead to a speciation span a wide spectrum:

- DNA *mutations*, which are mutations that impact short genome segments, from single nucleotides to a few tens of nucleotides.
- Genomic *rearrangements* are changes in the gene content of chromosomes and/or order of genes along the chromosomes (large-scale events). A rearrangement

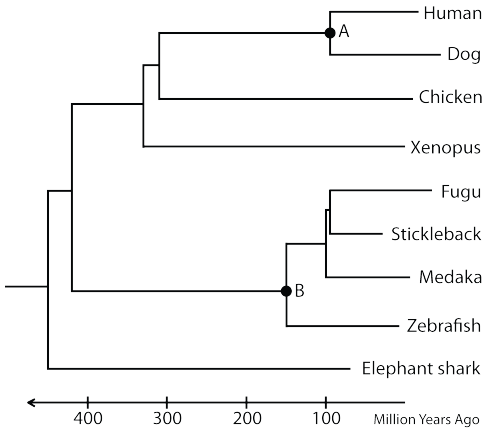

Figure 1.3: An evolutionary tree for some species. See [18] for more details.

is *balanced* if it does not modify the gene contents of the impacted genome. Otherwise, it is *unbalanced*. Examples of balanced rearrangements are reversals, translocations, chromosomes fusions/fissions, that we are describing later. Examples of unbalanced rearrangements include duplications, insertions and deletions.

In this work we only consider the balanced rearrangements.

Evolutionary relationships between a collection of species can be illustrated by a branching diagram called *evolutionary tree* or *phylogenetic tree*. In an evolutionary tree each node is the most recent common ancestor of its descendants. See Fig. 1.3.

We say that a species S_1 is an *outgroup* of species S_2 and S_3, if, in the evolutionary tree, both the paths from S_1 to S_2 and S_1 to S_3 pass through the last common ancestor of S_2 and S_3. For example, in Fig. 1.3 the node A is the last common ancestor of human and dog, and every path from chicken to human or dog passes through the node A. Hence, chicken is an outgroup for human and dog. Similarly, xenopus is an outgroup for zebrafish and fugu (where B is their last common ancestor).

1.2 Mathematical models

In this section we model genomes as discrete objects and rearrangements as discrete operators on these objects. See [24] for graph theory terminology not explained here.

1.2.1 Modeling genomes

Suppose a chromosome, like in Fig. 1.4 (a), has 4 genes. In order to distinguish these genes, we name them by distinct numbers 1, 2, 3, and 4. Also, we assign to each gene a *head* (shown by h) and a *tail* (shown by t). Head and tail of a gene are called *extremities* of that gene. The *extremities* of a genome is the set of all extremities of its genes. Now the chromosome can be defined as the set of adjacencies between its genes' extremities. So, the chromosomes in Fig. 1.4 (a), (b), and (c) can be described as:

- (a): $\{\{1_h, 2_t\}, \{2_h, 3_t\}, \{3_h, 4_t\}\}$,
- (b): $\{\{1_h, 2_h\}, \{2_t, 3_t\}, \{3_h, 4_t\}\}$,
- (c): $\{\{1_h, 2_t\}, \{2_h, 3_t\}, \{3_h, 4_t\}, \{4_h, 1_t\}\}$.

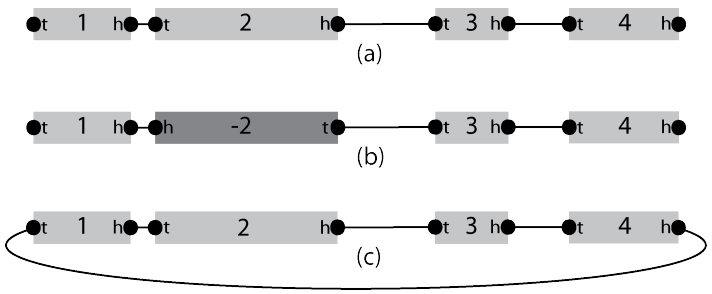

Figure 1.4: Genes' extremities and their adjacencies. (a) and (b) two linear chromosomes having gene 2 in reverse directions, (c) a circular chromosome.

A gene extremity is a *telomere* if it is not adjacent to any other extremity, e.g., in Fig. 1.4 (a) $\{1_t\}$ and $\{4_h\}$ are two telomeres. Note that the only difference between

chromosomes in Fig. 1.4 (a) and (b) is that the gene 2 appears in different (reverse) direction along these chromosomes. To express this event, we mark this gene by 2 and -2, respectively. A genome can be represented as a set of adjacencies between its extremities, by considering all of its chromosomes and their corresponding adjacencies.

1.2.2 Genome as a graph on extremities

Adjacencies of a genome can be represented by a graph called the *breakpoint graph*:

Definition Let G be a genome with gene set $\{1, 2, \ldots, n\}$. The breakpoint graph of the genome G, denoted by $B(G)$, is a graph with $\{1_h, 1_t, \ldots, n_h, n_t\}$ as its vertex set (extremities in G). Two vertices in $B(G)$ are connected by an edge if they form an adjacency in G.

Since each extremity cannot be adjacent to more than one other extremity, $B(G)$ is a matching (not necessarily perfect if some chromosomes are linear). Note that if a genome G is circular, $B(G)$ is a perfect matching. By adding the edges $\{\{i_h, i_t\}|1 \leq i \leq n\}$ we get a graph with maximum degree 2. Each connected component enables us to reconstruct the order of the genes along a chromosome. It is easy to see that a connected component is a path (resp. cycle) if its corresponding chromosome is linear (resp. circular). See Fig. 1.5.

Definition The breakpoint graph of m genomes $G_1, \ldots, G_m$ on the same set of genes $\{1, 2, \ldots, n\}$ is a graph $B(G_1, G_2, \ldots, G_m)$ whose vertex set is $\{i_h, i_t|1 \leq i \leq n\}$, and whose edge set is the **disjoint** union of the edge sets of $B(G_1), \ldots, B(G_m)$. Hence,

$$B(G_1, \ldots, G_m) = \bigcup_{i=1}^{m} B(G_i).$$

So $B(G_1, \ldots, G_m)$ can have multiple edges. To distinguish the edges of $B(G_i)$ in $B(G_1, \ldots, G_m)$, $1 \leq i \leq m$, we can consider them with different colors, i.e., we can assign the color c_i to the edges of $B(G_i)$. (See Fig. 1.6).

In the breakpoint graph $B(G_1, \ldots, G_m)$ by an (G_i, G_j)-*alternating* cycle (resp. path), $1 \leq i < j \leq m$, we mean a cycle (resp. path) whose edges are from G_i and G_j, alternatively.

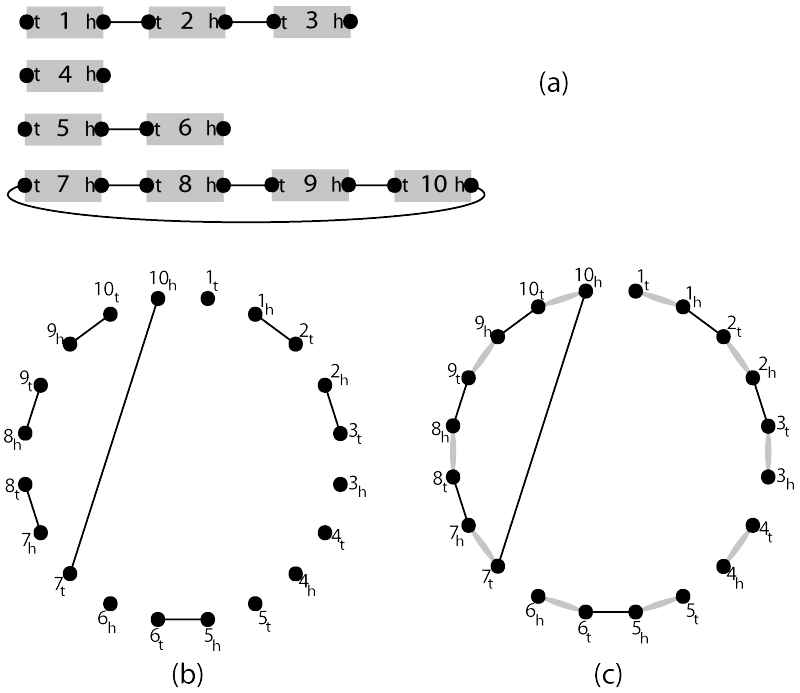

Figure 1.5: (a) A schematic genome with 10 genes, (b) its breakpoint graph, (c) connecting each gene's extremities and reconstructing the chromosomes.

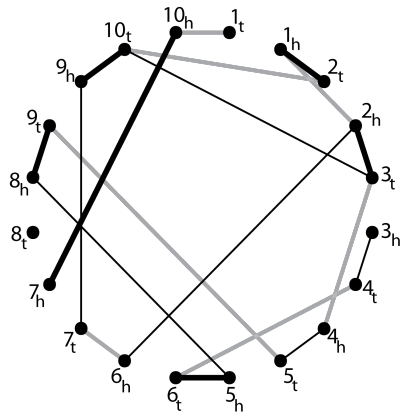

Figure 1.6: The breakpoint graph of three genomes with colors: black, bold black, and gray.

1.2.3 Modeling genome rearrangements

We consider here only rearrangements that preserve the gene content of a genome, and we describe them as discrete operators.

Definition A *double-cut-and-join*(DCJ) operation is an operation that acts on a genome adjacencies and/or telomeres as follows: Let x, y, z, and t be four extremities in the genome, then

1. Two adjacencies $\{x, y\}$ and $\{z, t\}$ can be replaced by the adjacencies $\{x, z\}$ and $\{y, t\}$, or by the adjacencies $\{x, t\}$ and $\{y, z\}$. See Fig. 1.7 (a).

2. An adjacency $\{x, y\}$ and a telomere $\{z\}$ can be replaced by the adjacency $\{x, z\}$ and the telomere $\{y\}$, or by the adjacency $\{y, z\}$ and the telomere $\{x\}$. See Fig. 1.7 (b).

3. Two telomeres $\{x\}$ and $\{y\}$ can be replaced by the adjacency $\{x, y\}$, or vice versa. See Fig. 1.7 (c).

The DCJ operation was introduced by Yancopoulos *et al.* [28]. It is a very general operation that models most rearrangements previously considered in genome rearrangement studies: *reversal*, *translocation*, *fission* and *fusion*[13]. So, we can redefine these rearrangements as DCJ operations as follows:

Definition A reversal is a DCJ operation that acts on a single chromosome by transforming two adjacencies $\{y, x\}$ and $\{z, t\}$ in one chromosome, where y, x, z, t appear in this order along the chromosome, into $\{y, z\}$ and $\{x, t\}$. It results in mirroring the genome segment located between adjacencies $\{y, x\}$ and $\{z, t\}$ and changing the direction of all the genes it contains. Reversals are also known as *inversions*. See Fig. 1.8.

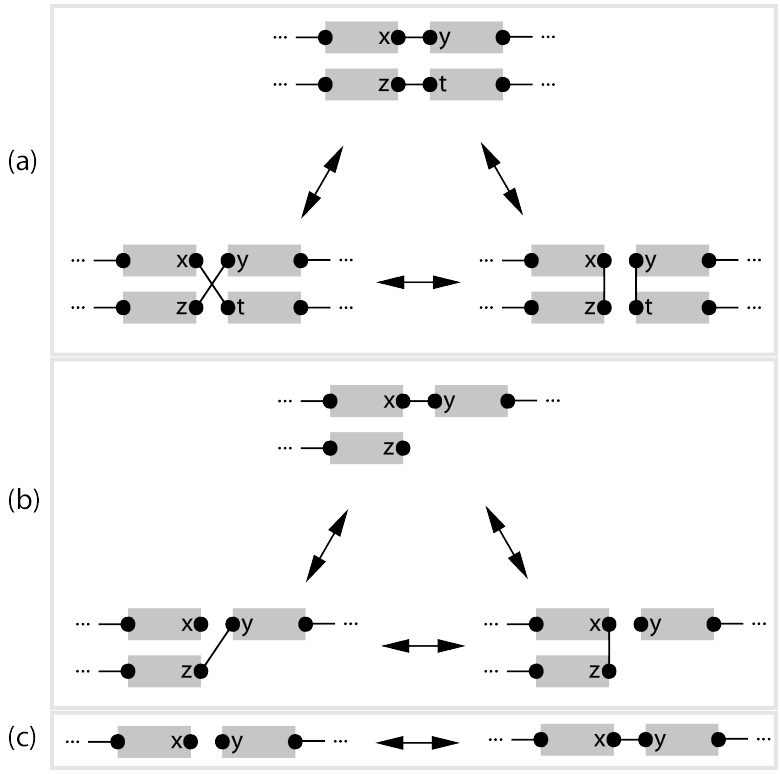

Figure 1.7: DCJ operations on adjacencies and/or telomeres, (a) Two adjacencies: cutting them and joining the resulting telomeres, (b) An adjacency and a telomere: cutting the adjacency and joining the telomere to one of the new resulting telomeres, (c) Cutting one adjacency, or joining two telomeres and creating a new adjacency.

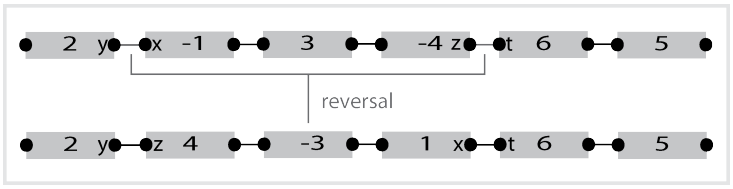

Figure 1.8: Reversal of a block. It is a DCJ operation acting on the adjacencies $\{x, y\}$ and $\{z, t\}$.

Definition A translocation is a DCJ operation which transforms two adjacencies $\{y, x\}$ and $\{z, t\}$ from different chromosomes into $\{y, z\}$ and $\{x, t\}$. See Fig. 1.9.

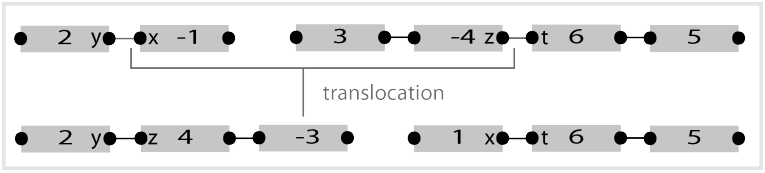

Figure 1.9: Translocation. Note that the unconnected telomeres of genes 1 and 3 remain telomeres. It is a DCJ operation acting on the adjacencies $\{x, y\}$ and $\{z, t\}$.

Definition A fission is a DCJ operation that consists in cutting an adjacency $\{x, y\}$ between two gene extremities x and y, and its inverse operation is a fusion, which joins two telomeres $\{x\}$ and $\{y\}$. See Fig. 1.10 .

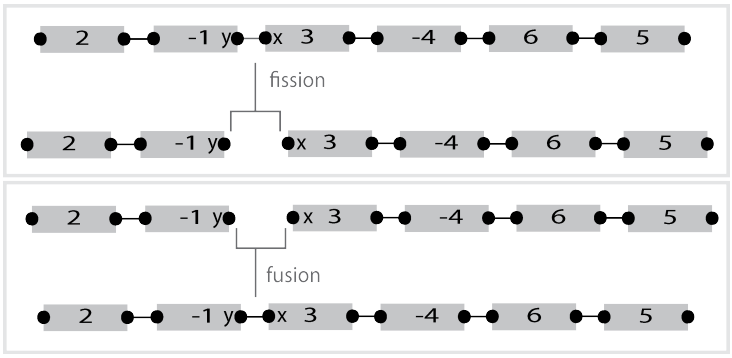

Figure 1.10: Fission and Fusion. Both of them are DCJ operation acting on $\{x, y\}$, or $\{x\}$ and $\{y\}$.

As we saw, the DCJ operation can model the genome rearrangements reversal, translocation, fission, and fusion. But the DCJ operation has more capability. For example, a DCJ operation can create a circular chromosome, which is impossible for the other aforementioned operations (see Fig. 1.11).

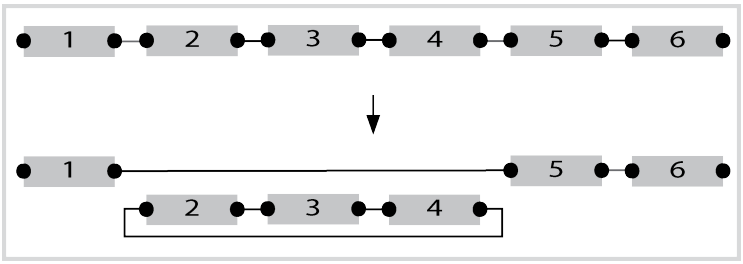

Figure 1.11: A DCJ operation creating circular chromosomes.

Also, *transpositions* can be modeled with two DCJ operations. A transposition consists in moving some consecutive genes, a.k.a. a *genomic block*, from a place to another place in the genome. In Fig. 1.12 a transposition is modeled by two DCJ operations.

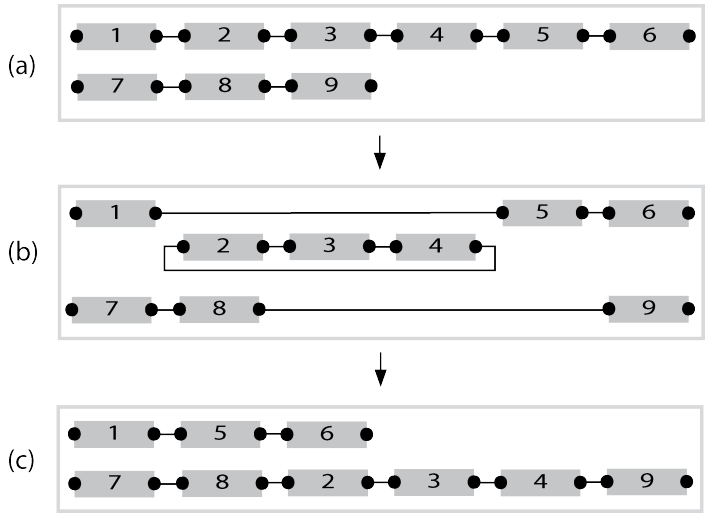

Figure 1.12: Transposition in two DCJ operation. (a) Two chromosomes, (b) taking out the genomic block (2, 3, 4) and creating three chromosomes, (c) inserting the circular chromosome between genes 8 and 9.

These observations show the generality of the DCJ operations, which is currently the most widely used model for genomic rearrangements. In the next chapter we study the distance and the median problems in DCJ and other models.

Chapter 2

Genomic Distance and Median Problem

In this chapter, we introduce the notions of genomic distances, and median problems.

2.1 Distances

In Section 1.2.1 we stated that genomes can be modeled as discrete objects. A natural question is: How can we measure the similarity/dissimilarity of two genomes? Genomic distances are good candidates for dissimilarity functions. Computing the distances allows us to generate distance matrices between genomes, and infer parsimonious species trees (see Fig. 1.3). This problem was launched in a paper by David Sankoff *et al.* [22].

Genomic distance can be the minimum number of allowed operations to transform a genome to the other one, or only a function to show the dissimilarity between the genomes. The former can be formalized as follows:

> Given two genomes G_1 and G_2 on the same set of genes, a set of allowed rearrangements $\mathcal{R}$, and an *optimality criterion* $\mathcal{C}$. An *evolutionary scenario* $\mathcal{S} = s_1, s_2, \ldots, s_k$ is a sequence of rearrangements $s_i \in \mathcal{R}$, and $\mathcal{C}$ is a function from all possible scenarios to real numbers. What is an optimal

scenario $\mathcal{S}$, i.e., it minimizes $\mathcal{C}(\mathcal{S})$?

The answer to this question depends on our choice for the model $(\mathcal{R}, \mathcal{C})$. It can be tractable or intractable by assuming different models. *Parsimony* is a widely used criterion in which we try to minimize the number of rearrangements, i.e., among all evolutionary scenarios $\mathcal{S} = s_1, s_2, \ldots, s_k$ we want to minimize k. In this work, we always assume parsimony as the optimality criterion.

For example, suppose $G_1 = (-3, -4, 5, -6, -1, 2)$ and $G_2 = (6, -4, 5, 3, -1, 2)$, and $\mathcal{R} = \{\text{reversals}\}$. One can easily check that the scenario in Fig. 2.1 is optimal to transform G_1 to G_2.

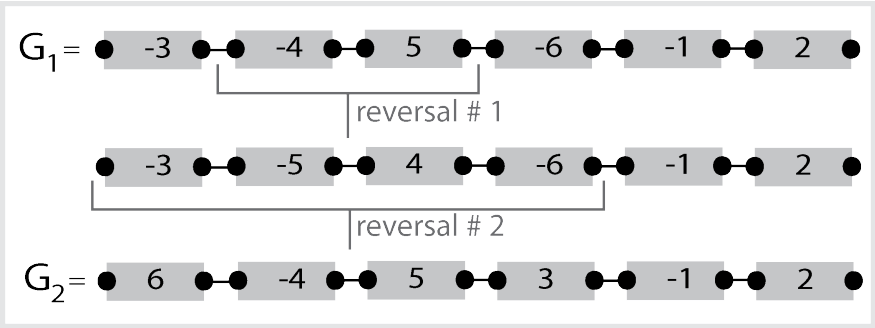

Figure 2.1: Transforming G_1 to G_2 by two reversals.

Note that in the genomes $G_1 = (-3, -4, 5, -6, -1, 2)$ and $G_2 = (6, -4, 5, 3, -1, 2)$ we can always rename the genes as long as their order and directions (signs) are consistent. For example, the gene 3 in G_1 is at the first place and has negative sign. In G_2 it is at the fourth place with positive sign (opposite of the sign in G_1). So if we rename the gene 3 in G_1 by a we should consider G_1 and G_2 as $(a, -4, 5, -6, -1, 2)$ and $(6, -4, 5, -a, -1, 2)$, respectively. Also, by taking $G_2 = (1, 2, 3, 4, 5, 6)$, G_1 becomes $(-4, 2, 3, -1, 5, 6)$.

Hence, the distance is equal to the minimum number of reversals needed to sort the elements of $G_1 = (-4, 2, 3, -1, 5, 6)$. This is known as the *sorting by reversal problem* [2].

2.1.1 Breakpoint distance

The breakpoint distance is not a distance based on genome rearrangements but a measure of dissimilarity between two genomes, based on the number of common adjacencies, and the number of common telomeres. An adjacency $\{x, y\}$ is common between two genomes if the extremities x and y are adjacent in the both genomes. Similarly, $\{x\}$ is a common telomere if the extremity x is telomere in the both genomes.

The breakpoint distance between multi-chromosomal genomes can be defined in various ways. In [23] Tannier *et al.* introduced a general approach using more information between two genomes. They considered the breakpoint distance to be a linear combination of the number of genes, number of common adjacencies, common telomeres, and the number of chromosomes: Suppose G_1 and G_2 are two genomes on n genes with c_1 and c_2 chromosomes, respectively. Let $a(G_1, G_2)$ be the number of common adjacencies in G_1 and G_2, and $e(G_1, G_2)$ be the number of their common telomeres. They considered the following formula as the general form of the breakpoint distance of G_1 and G_2:

$$d_{\mathrm{BP}}(G_1, G_2) = n - \alpha a(G_1, G_2) - \beta e(G_1, G_2) + \gamma(c_1 + c_2) + \delta|c_1 - c_2|,$$

where d_{BP} is the breakpoint distance function, α, β, γ are non-negative reals, and δ is a real number. Since d_{BP} is supposed to be a distance function it should satisfy the following conditions:

1. $d_{\mathrm{BP}}(G_1, G_2) = 0$ if and only if $G_1 = G_2$,
2. $d_{\mathrm{BP}}(G_1, G_2) = d_{\mathrm{BP}}(G_2, G_1)$,
3. $d_{\mathrm{BP}}(G_1, G_2) \leq d_{\mathrm{BP}}(G_1, G_3) + d_{\mathrm{BP}}(G_2, G_3))$, where G_3 is a third genome.

If we assume that fission and fusion change the breakpoint distance by 1 (since they break or repair an adjacency), it implies that $\gamma = \delta = 0$. Also by the first condition of distance function, considering $G_1 = G_2$ implies that $\alpha = 1$ and $\beta = \frac{1}{2}$. Therefore we have:

$$d_{\mathrm{BP}}(G_1, G_2) = n - a(G_1, G_2) - \frac{1}{2}e(G_1, G_2).$$

Note that the breakpoint distance d_{BP} is the distance of two multi-chromosomal genomes on the same set of genes.

2.1.2 Double-Cut-and-Join (DCJ) distance

The *DCJ distance* between two genomes on the same set of genes is equal to the minimum number of DCJ operations needed to transform one of the genomes to the other one. Currently, DCJ distance is the most widely used genomic distance.

In [4] Bergeron *et al.* showed that DCJ distance is computable in linear time. Suppose G_1 and G_2 are two genomes on the same set of n genes. Since $B = B(G_1, G_2)$ is the disjoint union of two matchings $B(G_1)$ and $B(G_2)$, B is a graph with maximum degree at most 2, and each connected component of B is a path or a cycle. Let $c(G_1, G_2)$ be the number of cycles in B, and $p(G_1, G_2)$ be the number the *odd paths* (paths with an odd number of vertices; so a single vertex is an odd path) in B. We denote the DCJ distance between G_1 and G_2 by $d_{\mathrm{DCJ}}(G_1, G_2)$. In [4] Bergeron *et al.* proved the following theorem:

Theorem 2.1.1 *The DCJ distance between two genomes G_1 and G_2 on the same set of n genes is*

$$d_{DCJ}(G_1, G_2) = n - c(G_1, G_2) - \frac{p(G_1, G_2)}{2}.$$

Proof Note that all cycles in $B = B(G_1, G_2)$ are *even* (having even number of vertices) because their edges are colored with two colors; each from one genome, and these colors alternate. So the total number of vertices in all cycles of B is equal to $2k$ for some integer $k \geq 0$. Therefore, $k \geq c(G_1, G_2)$, because each cycle has at least 2 vertices. Now, since B has $2n$ vertices, the total number of vertices in all paths of B (even or odd paths) is equal to $2(n - k)$, and since some of these vertices are in odd paths and each odd path has at least one vertex, we have $p(G_1, G_2) \leq 2(n - k)$. Hence,

$$2n - 2c(G_1, G_2) - p(G_1, G_2) \geq 2n - 2k - p(G_1, G_2) \geq 2n - 2k - 2(n - k) = 0,$$

hence, we have

$$n - c(G_1, G_2) - \frac{p(G_1, G_2)}{2} \geq 0 \Longleftrightarrow n \geq c(G_1, G_2) + \frac{p(G_1, G_2)}{2}. \tag{2.1}$$

Therefore, the equality holds in (2.1) if and only if all the cycles have length 2 and all of the remaining vertices form paths with one vertex. However, this happens if and only if $G_1 = G_2$ because they both have exactly the same set of adjacencies (as well as the same set of telomeres).

Now we claim that each DCJ operation changes at most one of the following items:

- number of (even) cycles in B by one, or
- number of odd paths in B by two.

In Fig. 2.2, different effects of DCJ operations on the number of cycles and odd paths are illustrated, so we have

$$d_{\mathrm{DCJ}}(G_1, G_2) \geq n - c(G_1, G_2) - \frac{p(G_1, G_2)}{2}.$$

Note that in each DCJ operation, removed edges should have the same color. So all cycles in B are even (see black frames in Fig. 2.2).

Let δ and π be the current numbers of cycles and odd paths, respectively. Now consider the following procedure: If there is a (even) cycle $C = c_1 c_2 \dots c_{2r} c_1$ in B, $r \geq 2$, we can cut the adjacencies $\{c_{2r}, c_1\}, \{c_2, c_3\}$ (both have the same color) and add adjacencies $\{c_1, c_2\}, \{c_3, c_{2r}\}$ (with the same color), and obviously we have increased δ by one and not changed π. Let P be a path (odd or even) $P = p_1 \dots p_s$. If $s > 2$ we cut $\{p_2, p_3\}$ and add adjacency $\{p_1, p_2\}$ (with color of $\{p_2, p_3\}$), which increases δ by one and does not change π. If $s = 2$, we just cut $\{p_1, p_2\}$ which only increases π by 2.

In the above procedure the value of $\delta + \frac{\pi}{2}$ always increases by 1. Since $n \geq \delta + \frac{\pi}{2}$ this procedure can be continued until $n = \delta + \frac{\pi}{2}$. This happens exactly in $n - c(G_1, G_2) - \frac{p(G_1, G_2)}{2}$ steps. Therefore we have,

$$d_{\mathrm{DCJ}}(G_1, G_2) = n - c(G_1, G_2) - \frac{p(G_1, G_2)}{2}. \quad \blacksquare$$

Note that when at least one of the genomes G_1 or G_2 is circular then $p(G_1, G_2) = 0$, and $d_{\mathrm{DCJ}}(G_1, G_2) = n - c(G_1, G_2)$.

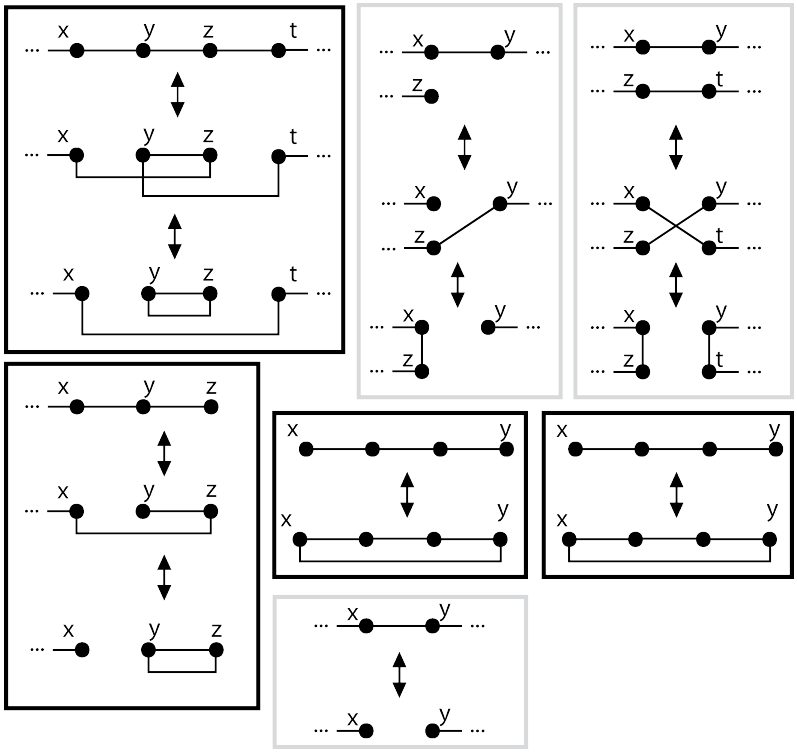

Figure 2.2: **Black frames:** when only the number of (even) cycles may change by 1, **gray frames:** when only the number of odd paths may change by 2.

2.1.3 Reversal distance

For the reversal distance, we assume the only permitted genomic rearrangement is the reversal. So we have to consider only uni-chromosomal genomes.

We saw that finding the reversal distance of two genomes is equivalent to solving the problem of sorting by reversals. In 1999, Hannenhalli and Pevzner [15] proved that the problem of sorting by reversals is tractable [1]. They also showed that the minimum number of reversals for sorting a genome G is:

[1]Only when genes are signed (having direction). Otherwise, if the reversals reverse the order of genes but not their signs, the problem is NP-hard [8].

$$d_{\mathrm{r}}(G) = \begin{cases} b(G) + c(G) + h(G) + 1, & \text{if } G \text{ is a fortress,} \\ b(G) + c(G) + h(G), & \text{otherwise,} \end{cases}$$

where the parameter $b(G)$ is the number of breakpoints of G, $c(G)$ is the maximum number of edge-disjoint alternating cycles of the breakpoint graph of G, and $h(G)$ is the number of hurdles in G (all of these parameters are defined precisely in [15]; also see [13, 3]). All of these parameters and the condition of being fortress are computable in time $O(n^4)$. In 2001, Bader *et al.* presented an algorithm which computes the reversal distance of genomes in linear time [1].

2.1.4 Hannenhalli-Pevzner (HP) distance

Prior to the DCJ model, the first tractable distance model for multi-chromosomal genomes was introduced by Hannenhalli and Pevzner [16].

The set of allowed operations is $\mathcal{R} = \{\text{reversals, translocations, fission, fusion}\}$. In this model the HP distance function, d_{HP}, is the minimum number of rearrangements from $\mathcal{R}$ to transform a genome to another one. Hannenhalli and Pevzner showed that the distance function d_{HP} is computable in polynomial time. However, unlike the DCJ distance, that is very easy to compute, the HP distance is very intricate to compute. The initial proof of Hannenhalli and Pevzner contained several flaws that were fixed by Ozery-Flato *et al.* [20], and by Jean *et al.* [17].

In [5] Bergeron *et al.* proved that HP distance can be obtained from DCJ distance plus an *extra cost*. More precisely, if G_1 and G_2 are two genomes on the same set of genes, then

$$d_{\mathrm{HP}}(G_1, G_2) = d_{\mathrm{DCJ}}(G_1, G_2) + t(G_1, G_2),$$

where $t(G_1, G_2)$ is the extra cost and can be computed in linear time. Since DCJ distance is computable in linear time (see Subsection 2.1.2), HP distance is also computable in linear time. Note that $t(G_1, G_2) \geq 0$, because all operations in $\mathcal{R}$ are instances of DCJ operations and $d_{\mathrm{HP}} \geq d_{\mathrm{DCJ}}$.

2.2 The median problem

Given more than two genomes and an evolutionary tree, a natural question is the architecture of the ancestral genomes (see [10, 19]). This leads to an important question known as *the median problem*.

Definition A median for a given set of genomes $G_1, \ldots, G_m$ is a genome M which minimizes

$$\sum_{i=1}^{m} d(M, G_i).$$

The most studied case, and the one we study, is when $m = 3$, because almost all the evolutionary trees are assumed to be binary. When $m = 3$, the median problem is equivalent to the ancestral genome reconstruction problem, as M can be seen as the last common ancestor of G_1 and G_2, with G_3 acting as an *outgroup*. Note that the median problem does not necessarily have a unique solution.

There are various versions of the median problem, depending on the following features:

1. The choice of the distance function.
2. Whether the genomes G_1, G_2, G_3, M are uni-chromosomal or multi-chromosomal.
3. Whether the genomes G_1, G_2, G_3 are circular, linear, or mixed.
4. Whether the median M is circular, linear, or mixed.

For example, consider this version of the median problem: (1) using the DCJ distance, (2) assuming all the genomes are multi-chromosomal, (3) letting G_1, G_2, G_3 be mixed genomes, and (4) looking for a circular median. We denote this version with (d_{DCJ}, multi, mixed, circular). Similarly, for other versions we can describe them by a similar notation.

DCJ median. In [23] Tannier *et al.* explored several versions of the median problem. In particular, they showed that computing a *DCJ median* (using the DCJ distance) is NP-hard, for all kinds of genomes. The focus in the DCJ median problem is on multi-chromosomal genomes. If the genomes are uni-chromosomal then a DCJ operation is just a reversal. See [9] and *HP median* below. From now on, by DCJ median problem we mean (d_{DCJ}, multi, mixed, circular) version of the median problem. The main reason is that when the median M is circular, then its distance to any genome G is equal to $n - c(M, G)$, where n is the number of genes in M and G. This assumption makes the calculations easier, and the median problem will be equivalent to the problem of maximizing $c(M, G_1) + c(M, G_2) + c(M, G_3)$. Note that if we consider only circular medians, then the problem does not lie in a metric space.

Breakpoint median. Tannier *et al.* also gave a polynomial time algorithm for computing the *breakpoint median* (using the breakpoint distance) when genomes are all circular or all mixed, i.e., the version (d_{BP}, multi, circular (resp. mixed), circular (resp. mixed)). This was a surprising result, as most other versions of the median problem are known to be NP-hard. The showed that allowing circular chromosomes in the median gives the flexibility needed to fall in the tractable area. It implies that even if the input genomes are unichromosomal, allowing for a multichromosomal mixed median allows to find a median. The idea of the proof is based on expanding the breakpoint graph of the genomes to another graph, B', such that finding a breakpoint median is equivalent to finding a maximum weighted matching in B', which is doable in polynomial time.

HP median. In [9] Caprara showed that finding an RT median (using Reversal-Translocation distance) is NP-hard, when genomes are unichromosomal, and genomes and median are all either circular or linear.

In Table 2.1 the tractability results of the median problem for Breakpoint, DCJ, and Reversal-Translocation distances are summarized.

Different versions of the median problem	**Tractability**
(d_{BP}, uni, circular (resp. linear), circular (resp. linear))	NPC [21, 7]
(d_{BP}, multi, circular (resp. mixed), circular (resp. mixed))	P [23]
(d_{BP}, multi, linear, linear)	NPC [23]
(d_{DCJ}, uni, circular (resp. linear), circular (resp. linear))	NPC [9]
(d_{DCJ}, multi, circular (resp. mixed), circular (resp. mixed))	NPC[23]
(d_{DCJ}, multi, linear, linear)	Open
(d_{HP}, uni, circular (resp. linear), circular (resp. linear))	NPC [9]
(d_{HP}, multi, linear/circular/mixed, linear/circular/mixed)	Open

Table 2.1: Summary of tractability results on the median problem with Breakpoint, DCJ, and Reversal-Translocation distances.

2.3 Hardness of the DCJ median problems

In [23] Tannier *et al.* proved the following theorem:

Theorem 2.3.1 *DCJ median problem for multi-chromosomal genomes is NP-hard, even for circular genomes.*

Sketch of Proof. We say a graph is *bicolored* if its edges are colored by blue and red, and it is *balanced* if

1. all of its vertices have degree 2 or 4,
2. at each vertex the number of red edges is equal to number of blue edges at that vertex, and
3. there is no cycle with only blue or only red edges.

The *breakpoint graph decomposition* (BGD) problem for a balanced bicolored graph G, is to find a partition of its edges into the maximum number of edge-disjoint cycles with edges alternating between red and blue. In [9] Caprara proved that BGD is NP-hard.

The idea of the proof of Theorem 2.3.1 is based on the reduction of the BGD problem to the median problem. We transform a balanced bicolored graph G to a breakpoint graph G' of three genomes G_1, G_2, G_3 as follows (see Figure 2.3) :

1. Replace each vertex u of degree 4 in G with four extremities u_t^x, u_h^x, u_t^y, and u_h^y such that u_t^x, u_t^y are attached to red edges of u, and u_h^x, u_h^y are attached to blue edges of u. Add the blue and red edges to G_3, $\{u_t^x, u_h^x\}, \{u_t^y, u_h^y\}$ to G_1, and $\{u_t^x, u_h^y\}, \{u_t^y, u_h^x\}$ to G_2.

2. Replace each vertex v of degree two with two extremities v_t^x, v_h^x where v_t^x is connected to the red edges of v and v_h^x to the blue one. Add two edges between v_t^x, v_h^x, one in G_1 and one in G_2, and consider the blue and red edges in G_3.

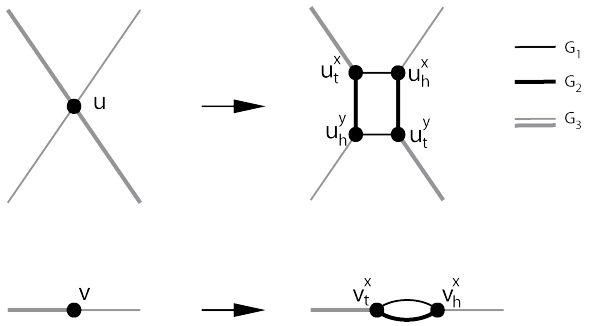

Figure 2.3: Reduction from BGD to DCJ median problem.

We consider all vertices $v_t^x, v_h^x, v_t^y, v_h^y$ as extremities of genomes G_1, G_2, G_3, and the resulting graph, G', is their breakpoint graph. Let w_2 and w_4 be the number of vertices of degree 2 and 4 in G, respectively. Finally, there exists a genome M such that $d_{\text{DCJ}}(M, G_1) + d_{\text{DCJ}}(M, G_2) + d_{\text{DCJ}}(M, G_3) \leq w_2 + 3w_4 - k$ if and only if there exist at least k edge-disjoint cycles in G with alternating red/blue edges, which implies the theorem. ∎

As we saw in this chapter while the distance problem is tractable in many models, the median problem is computationally intractable in most models. Recent progress in understanding the properties of the breakpoint graphs of genomes, and specifically the family of *adequate subgraphs*, led Xu to design algorithms to compute DCJ median genomes which are efficient on real data, but do not define well-characterzied classes of tractable instances in general [27, 26, 25]. An adequate subgraph H with m vertices is a subgraph such that $\text{cyc}(H) \geq \frac{3}{4}m$. Xu *et al.* showed that every adequate subgraph

is 0-independent. Their algorithm is based on finding adequate subgraphs in the breakpoint graph, consider a median for them, and reduce the problem to a smaller breakpoint graph.

In the next chapter we present some tractability results and a FPT algorithm for DCJ median problem[2] in some special cases.

[2](d_{DCJ}, multi-chromosomal, mixed, circular) version.

Chapter 3

Tractable Instances for the DCJ Median Problem

As mentioned in the previous chapter, we only consider the DCJ median problem of mixed multi-chromsomal genomes and a circular median. So a *circular DCJ median* for three genomes G_1, G_2, and G_3 on the same set of n genes, is a circular genome M which minimizes

$$\sum_{i=1}^{3} d_{DCJ}(G_i, M) = 3n - c(G_1, M) - c(G_2, M) - c(G_3, M),$$

(see Section 2.1.2) and it is equivalent to

$$\text{maximize} \sum_{i=1}^{3} c(G_i, M). \tag{3.1}$$

Therefore, a genome M that maximizes the total number of (M, G_i)-alternating cycles (for $i \in \{1, 2, 3\}$) is a circular DCJ median. In this work, we consider the maximization case. An *alternating cycle* is a (M, G_i)-alternating cycle for some i, $1 \leq i \leq 3$. For the sake of simplicity, by *median* we always mean the circular DCJ median.

We can generalize the concept of breakpoint graph to any 3-edge-colored graph, and define the median problem as follows:

Definition A breakpoint graph $B(G_1, G_2, G_3)$ is a 3-edge-colored graph with color

classes G_1, G_2, and G_3. A median of B is a matching M on vertices of B, which maximizes $\sum_{i=1}^{3} c(G_i, M)$.

3.1 Preliminaries

Let $B = B(G_1, G_2, G_3)$ be the breakpoint graph, and let M be a median of B. The graph $B_M(G_1, G_2, G_3) = B \cup M$ is called the *median graph* of B with the median genome M. Also, by $\text{cyc}(B)$ we mean the number of alternating cycles in the median graph B_M.

Remark The edges of a breakpoint graph are colored, and cyc is a function from 3-edge-colored graphs to integer numbers. Hence, $\text{cyc}(B)$ does not depend only on the topology of B, but also on the colors of its edges.

Remark Note that for different medians M of B, B_M has the same number of alternating cycles.

From now on, for a given median graph $B_M(G_1, G_2, G_3)$, edges in $G_1 \cup G_2 \cup G_3$ are called *colored edges*, and edges in M are called *median edges*. When the context is clear, we only use B_M for $B_M(G_1, G_2, G_3)$. By a k-cycle in a 3-edge-colored graph, we mean an alternating cycle with k vertices.

Let H be a subgraph of the breakpoint graph B. An *H-crossing edge* in the median graph B_M is a median edge which connects a vertex in $V(H)$ to a vertex in $V(B) - V(H)$. By an *H-crossing cycle* we mean an alternating cycle which contains at least one H-crossing edge. The subgraph H is *k-independent* if there is a median M for B such that the number of H-crossing edges in B_M is at most k. We denote by C_k (resp. P_k) a subgraph of B that is isomorphic to a cycle (resp. path) with k vertices.

The following operation was defined in [27]: *shrinking* a pair of vertices $\{u, v\}$ or an edge with vertices u and v consists of: *(1)* removing all edges between u and v, *(2)* identifying the edges with same color incident to u and v, *(3)* removing u and v. We denote the resulting graph by $B \cdot \{u, v\}$ (Fig. 3.1).

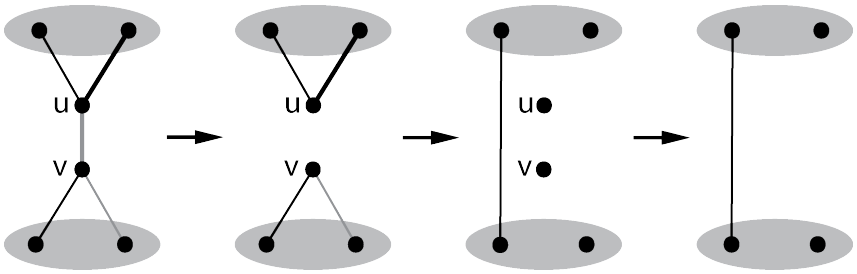

Figure 3.1: Shrinking a pair $\{u, v\}$

Proposition 3.1.1 *Let B be the breakpoint graph of genomes $G_1, \ldots, G_m$, and $u, v \in V(B)$. Suppose that there are k colored edges between u and v. If there exists a median containing uv, then $cyc(B) = cyc(B \cdot \{u, v\}) + k$.*

Proof Consider a median M which contains the edge uv. Let $B' = B \cdot \{u, v\}$, $M' = M - \{u, v\}$, and C be an alternating cycle in B_M. If C does not contain $\{u, v\}$, then it is easy to see that since $\{u, v\}$ is in M, C does not contain any of the k edges between u and v. Thus, C remains in $B' \cup M'$, without change. So assume that C contains uv. If its length is larger than 2, shrinking $\{u, v\}$ results in a cycle with smaller length in $B' \cup M'$ (2 units smaller). Otherwise, if it has length 2, it disappears in $B' \cup M'$. Thus the number of alternating cycles which disappear in $B' \cup M'$ is k, since there are k edges between u and v. Therefore, $\text{cyc}(B) \leq \text{cyc}(B \cdot \{u, v\}) + k$.

Now suppose N is a median for B'. By a similar argument, if $N' = N \cup \{u, v\}$, then $B \cup N'$ has $\text{cyc}(B \cdot \{u, v\}) + k$ alternating cycles. So, $\text{cyc}(B) \geq \text{cyc}(B \cdot \{u, v\}) + k$, and we have $\text{cyc}(B) = \text{cyc}(B \cdot \{u, v\}) + k$, which means that M' (resp. N') is a median for B' (resp. B). ∎

3.2 A class of tractable instances

Our main result covers a large class of tractable instances for the median problem. Obviously, the median problem for three genomes involves a breakpoint graph with maximum degree 3. We hint here that the hardness of the problem is due to the

vertices of degree 3.

Theorem 3.2.1 *Let G_1, G_2, and G_3 be three genomes. If there exists a median of $B(G_1, G_2, G_3)$ with at most ℓ edges whose both end-vertices are of degree 3, then computing such a median can be done in time $O(n^3 \cdot (\ell + 1) \cdot (3^m \cdot m^{2\ell} + 1))$, where m is the number of vertices of degree 3, and n is the number of genes in $B(G_1, G_2, G_3)$.*

Note that, as corollaries of this theorem, we have:

1. if m is bounded, then computing a median is tractable,
2. if ℓ is bounded, then computing a median is FPT with parameter m,
3. if m is not bounded, we can remove some edges incident to vertices of degree 3, so that in the new instance m' (the number of vertices of degree 3 in the newer graph) is bounded. Now, by the case 1, there is a polynomial time algorithm which computes the median of the new instance. We approximate the median of the original problem by this median.

Informally, to prove Theorem 3.2.1, we first consider all possibilities (configurations) for matching vertices of degree 3. For each configuration, we reduce the breakpoint graph by shrinking and removing some edges to obtain a graph with maximum degree 2 whose connected components will be paths or cycles. Next, we compute a median of the remaining breakpoint graph in polynomial time. This will follow since we show that there exists at least one median M such that: (1) every even component of $B(G_1, G_2, G_3)$ has no crossing edge, so they can be considered independently and (2) odd components $B(G_1, G_2, G_3)$ are matched with each other. See Fig 3.2.

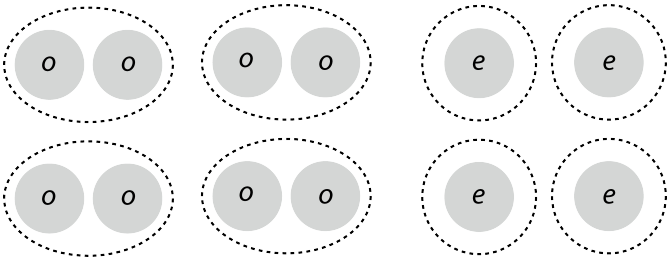

Figure 3.2: **(e)** Each even component has no crossing edge, **(o)** For each odd component H_1, there is exactly another odd component H_2, such that $(H_1 \cup H_2)$ has no crossing edge.

From now, G_1, G_2, and G_3 are mixed multi-chromosomal genomes on n genes, and M is a circular median of these genomes, unless otherwise specified. We denote their breakpoint graph by B, and the median graph by B_M.

3.2.1 Independence of even cycles and paths

In this section we show that if a connected component of B is isomorphic to an even cycle or even path, it is 0-independent (i.e. each can be processed independently from the other connected components) and if it is isomorphic to an odd path it is 1-independent. We first state two auxiliary results.

Lemma 3.2.2 *If B is isomorphic to P_k or C_{2k}, for $k \geq 1$, then for every subgraph $H \subseteq B$, $cyc(H) \geq \frac{|E(H)|}{2}$.*

Proof Consider the path $P_k = u_1 u_2 \dots u_k$. Let M be the matching consisting of the edges $u_1u_2, u_3u_4, \dots$, and $u_{t-1}u_t$, where $t = 2\lfloor \frac{k}{2} \rfloor$. Obviously, the number of alternating cycles in $P_k \cup M$ is $\lfloor k/2 \rfloor$, so $\text{cyc}(P_k) \geq \frac{k}{2} \geq \frac{|E(P_k)|}{2}$. Similarly $\text{cyc}(C_{2k}) \geq k = \frac{|E(C_{2k})|}{2}$. see Fig. 3.3.

Any proper subgraph $H \subset P_k$ or C_{2k} is a union of disjoint paths. If we take the union of matchings described above for each of these paths and call it M, there are at least $|E(H)|/2$ alternating cycles in $H \cup M$. Therefore for any subgraph $H \subseteq B$, $\text{cyc}(H) \geq \frac{|E(H)|}{2}$. ∎

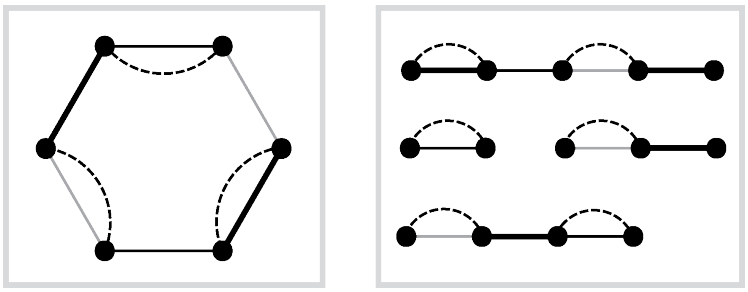

Figure 3.3: Median edges (dashed) for cycles and union of disjoint paths.

Suppose S and T are two subgraphs of a breakpoint graph. So both of them are 3-edge-colored graphs, and they are smaller breakpoint graphs. We say that T is an *alternating-subdivision* of S if we can obtain T from S as follows: subdivide each edge $e = \{a, b\}$ by an even (possibly zero) number of vertices resulting in a path $av_1v_2 \dots v_{2k}b$, then remove every second edge, i.e., $v_1v_2, v_3v_4, \dots, v_{2k-1}v_{2k}$. We denote the set of all removed edges by $\mathrm{Rem}(S)$. See Fig. 3.4.

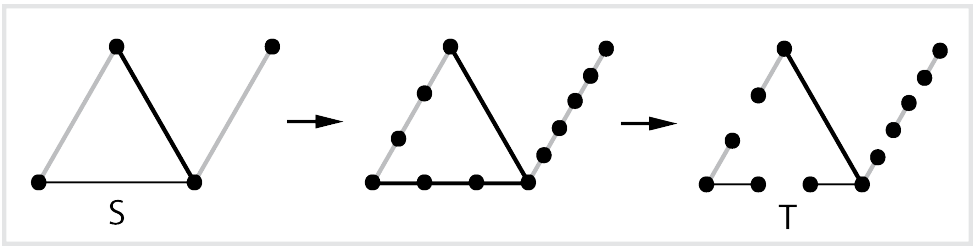

Figure 3.4: Obtaining T as alternating-subdivision of S

Lemma 3.2.3 *If T is an alternating-subdivision of S, then $cyc(T) \geq cyc(S)$.*

Proof Let M be median of S and $M' = M \cup \mathrm{Rem}(S)$. Note that M' is a matching on T, and each cycle in $S \cup M$ corresponds to a cycle in $T \cup M'$. Obviously this correspondence is one-to-one, so $S \cup M$ and $T \cup M'$ have $\mathrm{cyc}(S)$ alternating cycles, and we have $\mathrm{cyc}(T) \geq \mathrm{cyc}(S)$ (see Fig. 3.5). ∎

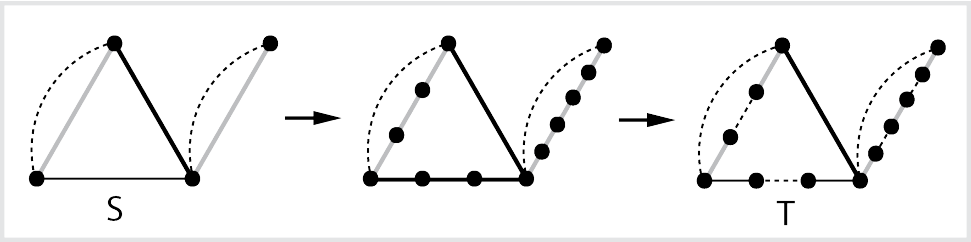

Figure 3.5: Obtaining a matching for T from the median of S (dashed edges are the median edges)

Proposition 3.2.4 *Let H be a connected component of B. If H is isomorphic to P_{2k} or C_{2k}, for $k \geq 1$, then H is 0-independent.*

Proof Let M be a median of B. Suppose M has ℓ H-crossing edges in B_M. If $\ell = 0$, then we are done, so assume that $\ell > 0$. Since H has an even number of vertices, ℓ is even and $\ell \geq 2$. Because H is a connected component in B, each H-crossing cycle contains an even number of H-crossing edges.

Let $\mathrm{C}(M)$ be the set of all H-crossing cycles in the median graph B_M, and $\mathrm{Cr}(M)$ be the set of all H-crossing edges. Let $\mathrm{X}(M)$ be the set of colored edges in all cycles of $\mathrm{C}(M)$, and $\mathrm{Y}(M)$ be the set of all H-crossing edges in all cycles of $\mathrm{C}(M)$.

If there is no H-crossing cycle, i.e., $\mathrm{C}(M) = \mathrm{Y}(M) = \emptyset$, we modify M by removing all H-crossing edges, and re-matching the vertices inside of H together and outside of H together. Since ℓ is even, this is always possible and we get a median with no H-crossing edge. So assume that there exists at least one H-crossing cycle.

We introduce a transformation on M such that after each step we obtain another median of B with fewer H-crossing edges (decreasing $|\mathrm{Cr}(M)|$) or the same number of H-crossing edges but with fewer edges in H-crossing cycles (implies the decrement in $|\mathrm{X}(M)|$). After this transformation, we obtain a median with no H-crossing cycle. Each step of the transformation is as follows: from each H-crossing cycle we pick one colored edge in H incident to an H-crossing edge in that cycle arbitrarily. Let S be the subgraph of B induced by these picked colored edges. Since every colored edge is in at most one alternating cycle and since two edges of the same color are independent,

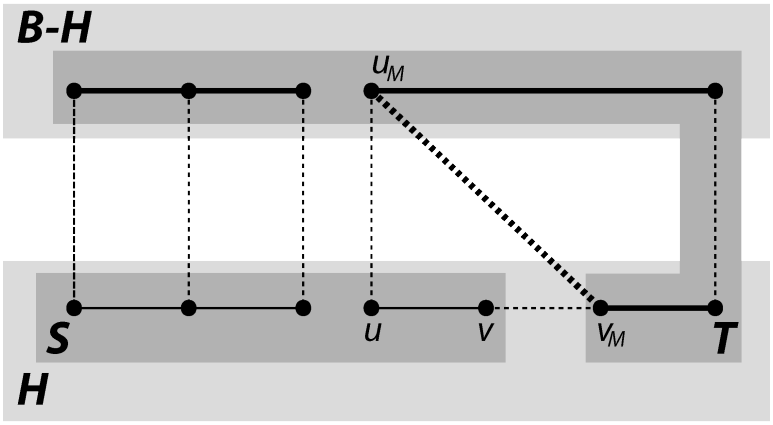

Figure 3.6: Dashed lines are the median edges (the bold dashed edge is in M' as in the proof of Proposition 3.2.4). The edges of S and T are shown by solid and bold solid edges, respectively (in dark gray areas). Note that T can have edges in H and $B - H$.

$|E(S)| = |\mathrm{C}(M)|$. Also $S \subseteq H$, and by Lemma 3.2.2, $\mathrm{cyc}(S) \geq |E(S)|/2$. See Fig. 3.6.

Let $T = \mathrm{X}(M) - S$. We claim that T is an alternating-subdivision of S. For a vertex $x \in V(S)$ let x_M be the neighbor of x in M. If $u, v \in V(S)$ and $uv \in E(S)$ then, by definition, uv is a colored edge of an H-crossing cycle in H which is incident to an H-crossing edge. Therefore, there is an alternating path from u_M to v_M, with alternating colored and median edges in that cycle. If this path has t colored edges, we subdivide the edge uv with $2t - 2$ vertices and remove every second edge. If we do this for every edge $uv \in E(S)$ we obtain the alternating-subdivision T of S. By Lemma 3.2.3, $\mathrm{cyc}(T) \geq \mathrm{cyc}(S) \geq |E(S)|/2 = |\mathrm{C}(M)|/2$.

Now we remove all the edges in $\mathrm{Y}(M)$: the total number of alternating cycles decreases by $|\mathrm{C}(M)|$. Let M_S be a median of S and M_T the matching obtained from M_S as described in Lemma 3.2.3. Considering M_S and M_T defines at least $|\mathrm{C}(M)|/2 + |\mathrm{C}(M)|/2 = |\mathrm{C}(M)|$ alternating cycles. Hence, the new matching $M' = (M - \mathrm{Y}(M)) \cup M_S \cup M_T$ is still a median of B.

Since M_S and M_T both give at least $|\mathrm{C}(M)|/2$ cycles, and M is median we must have $\mathrm{cyc}(S) = \mathrm{cyc}(T)$. If there exists $e \in \mathrm{X}(M') - \mathrm{X}(M)$ then there would be at

least one H-crossing cycle induced by M' which is not induced by M_S or M_T, so the number of alternating cycles in $B_{M'}$ would be more than the number of alternating cycles in B_M, which is a contradiction, because B_M and $B_{M'}$ have the same number of alternating cycles. Regarding the fact that $S \neq \emptyset$, we have $\mathrm{X}(M') \subset \mathrm{X}(M)$, since $E(S) \subset \mathrm{X}(M)$ and $E(S) \cap \mathrm{X}(M') = \emptyset$ (the vertices in S are matched to themselves).

We now show that $|\mathrm{Cr}(M')| \leq |\mathrm{Cr}(M)|$. An edge e in M' is H-crossing edge only if $e = u_M v_M \in M_T$, where $u, v \in V(S)$, $v_M \in H$, $u_M \in B - H$, and $uv \in M_S$ (see Lemma 3.2.3, how we get M_T from M_S). This implies that uu_M is an H-crossing edge in M. Now by assigning uu_M to $u_M v_M$ we obtain a one-to-one function from $\mathrm{Cr}(M')$ to $\mathrm{Cr}(M)$, since uu_M and $u_M v_M$ are incident and M' is a matching and different edges in $\mathrm{Cr}(M')$ are assigned to different edges in $\mathrm{Cr}(M)$. Hence $|\mathrm{Cr}(M')| \leq |\mathrm{Cr}(M)|$. We have $|\mathrm{X}(M')| + |\mathrm{Cr}(M')| < |\mathrm{X}(M)| + |\mathrm{Cr}(M)|$, and by iterating all steps above we get a median without any H-crossing edge. ∎

Remark The transformation introduced in the proof of Proposition 3.2.4 can be applied as long as there are at least two H-crossing edges. Because when there is no H-crossing cycle and there are at least two H-crossing edges, we can remove two of them and match their end-vertices in H together and other end-vertices together. Otherwise, if there is at least one H-crossing cycle we can define S and T as before and continue our transformation steps.

Lemma 3.2.5 *Let H be a connected component of B. If H is isomorphic to P_{2k-1}, for $k \geq 1$, then H is 1-independent. Moreover, there exists a median in which the H-crossing edge is incident to one of the terminal vertices (vertices of degree one) of H.*

Proof From Lemma 3.2.2, we know that for every subgraph $H' \subseteq P_{2k-1}$ we have $\mathrm{cyc}(H') \geq |E(H')|/2$. Now by using the transformation introduced in the proof of Proposition 3.2.4 and by previous remark, we can obtain a median with exactly 1 H-crossing edge. Note that if there is only one H-crossing edge it cannot be in any H-crossing cycle, and by Proposition 3.2.7 below (which shows that $\mathrm{cyc}(P_n)$ is independent from edge colors in P_n) we can connect that crossing edge to a terminal vertex of H. ∎

Lemma 3.2.6 *For the breakpoint graph B there exists a median in which even components have no crossing edge, and each odd path has exactly one crossing edge.*

Proof It is easy to see that for each even/odd path or even cycle we can do the transformation in Proposition 3.2.4 on a current median and reduce the crossing edges for each of them, without increasing the number of crossing edges in other components. ∎

3.2.2 Computing cyc for paths and even cycles

The results of the previous section open the way to computing a median of a breakpoint graph with maximum degree 2 by considering each path or even cycle independently (for odd paths we consider only one crossing edge connected to one of its terminal vertices). Here, first we show that computing a median of an even connected component or an odd path is tractable. In the following let H be a connected component of B.

Proposition 3.2.7 *If H is isomorphic to P_k, for $k \geq 1$, then $cyc(H) = \lfloor \frac{k}{2} \rfloor$.*

Proof From Lemma 3.2.2 $\text{cyc}(H) \geq \lfloor \frac{k}{2} \rfloor$. To prove the equality we use induction on k. It obviously holds for $k = 1$. So we assume that $k \geq 2$, and consider a median M for H. If there is no 2-cycle (i.e., $E(H) \cap M = \emptyset$; a 2-cycle is a cycle with length two which is consist of two parallel edges, denoted by C_2), each alternating cycle has length ≥ 4; since in each alternating cycle there are at least 2 colored edges, $\text{cyc}(H) \leq \lfloor \frac{|E(H)|}{2} \rfloor = \lfloor \frac{k-1}{2} \rfloor \leq \lfloor \frac{k}{2} \rfloor$.

Now assume that the median M contains a 2-cycle. So there is an edge $uv \in E(H) \cap M$. Shrinking $\{u, v\}$ results in H' that is either a single path with $k-2$ vertices or two paths with p and q vertices such that $p + q = k - 2$. In both cases, using induction and the fact that all paths are 0-independent or 1-independent, we obtain that in the case:

- **of one remaining path** $\text{cyc}(H') \leq \lfloor \frac{k-2}{2} \rfloor + 1 = \lfloor \frac{k}{2} \rfloor$.
- **of two remaining paths** $\text{cyc}(H') \leq \lfloor \frac{p}{2} \rfloor + \lfloor \frac{q}{2} \rfloor + 1 \leq \lfloor \frac{k}{2} \rfloor$.

This completes the proof and it is easy to see that the time needed to find a median of P_k is $O(k)$. ∎

Remark Note that if H is isomorphic to P_k, then $\text{cyc}(H)$ is independent from the edge coloring of H.

Lemma 3.2.8 *If H is isomorphic to C_{2k}, for $k \geq 1$, then either $cyc(H) = k$ or $k+1$.*

Proof Taking the edges of H, alternatively as a matching M, results in a graph with k 2-cycles, so $\text{cyc}(H) \geq k$. Note that when $k = 4$, $\text{cyc}(C_4) = 3 = \frac{4}{2} + 1$. Also $\text{cyc}(C_2) = 2 = \frac{2}{2} + 1$.

Now assume that $\text{cyc}(H) > k$. We claim that there is at least one 2-cycle in any median graph. As in Proposition 3.2.7, if all alternating cycles in the median graph H_M have length at least 4, then the number of alternating cycles is at most k, so there must exist at least one 2-cycle. Let $\{u, v\}$ be the colored edge in that 2-cycle: $\text{cyc}(H) = \text{cyc}(H \cdot \{u, v\}) + 1$ (Proposition 3.1.1). Obviously, $H \cdot \{u, v\}$ is a path, or a cycle, and it is a cycle if and only if the two edges incident to $\{u, v\}$ have the same color. If it is a path, Proposition 3.2.7 implies that $\text{cyc}(H \cdot \{u, v\}) = k - 1$ and $\text{cyc}(H) = k$, which contradicts the assumption that $\text{cyc}(H) > k$. So $H \cdot \{u, v\}$ is a cycle and the edges incident to uv have same color. By induction we can find a median of $H \cdot \{u, v\}$ with $\text{cyc}(H \cdot \{u, v\}) = k - 1 + 1$ or $\text{cyc}(H \cdot \{u, v\}) = k - 1$, alternating cycles. Hence, $\text{cyc}(H) = k + 1$ or $\text{cyc}(H) = k$. ∎

Remark By previous theorem, if H is isomorphic to C_{2k}, then $\text{cyc}(H)$ dependents on the edge coloring of H.

We say that a cycle C_{2k} is of the *first kind* if $\text{cyc}(C_{2k}) = k$, and it is of the *second kind* if $\text{cyc}(C_{2k}) = k + 1$ (see Fig. 3.7). We show below how to decide in polynomial time if an even cycle is of the first or second kind.

In a cycle C, the *signature* of a vertex is an ordered pair (a, b) such that a and b are the colors of the edges incident to that vertex. We define signature of vertices of a cycle with respect to a fixed orientation of the cycle; a will be the color of the incoming edge and b the color of the outgoing edge.

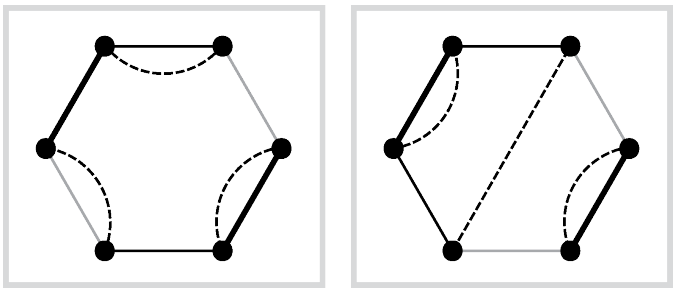

Figure 3.7: Dashed edges are median edges: (Left) a cycle C_6 of the first kind — (Right) a cycle C_6 of the second kind; the matched vertices are diagonal

Two vertices u and v are *diagonal* if their signatures are of the form (a, b) and (b, a) (see Fig. 3.7). Now let M be a median of an even cycle C. Let uv and $u'v'$ be edges in M. We say that uv and $u'v'$ *cross* if u, u', v, v' appear in this order along C. A *cross-free diagonal* matching for C is a matching whose edges connect pairs of diagonal vertices and no two edges cross.

We first give an auxiliary lemma regarding cycles of the second kind.

Lemma 3.2.9 *Let C be an even cycle of the second kind, and M be a median of C.*

1. *Each edge in M joins two diagonal vertices.*

2. *The edges in M do not cross.*

Proof Let $C = C_k$. We first prove point (1). For the sake of contradiction, assume that $uv \in M$ and two vertices u and v are not diagonal. Also suppose that signatures of u and v are (a, b) and (c, d), respectively. So by our assumption $(c, d) \neq (b, a)$. We have the following cases:

- $(a, b) = (c, d)$: In this case by shrinking the pair $\{u, v\}$ we get a smaller cycle C_{n-2} and by Proposition 3.1.1, $\mathrm{cyc}(C) = \mathrm{cyc}(C_k \cdot \{u, v\}) = \mathrm{cyc}(C_{k-2}) \leq \frac{k-2}{2} + 1 = \frac{k}{2}$ which is a contradiction, since C is of the second kind. Note that in this case u and v cannot be consecutive vertices on C.

- $(a, b) \neq (c, d)$: Now by shrinking the pair $\{u, v\}$, the remaining graph can be one of the following:
  - A path with $k-2$ vertices,
  - A cycle and a path, together with $k-2$ vertices.

  In the first case, vertices u and v must be consecutive on C. But now $\text{cyc}(C) = \text{cyc}(C_k \cdot \{u, v\}) + 1 = \text{cyc}(P_{k-2}) + 1 = \frac{k-2}{2} + 1 < \frac{k}{2} + 1$, which is a contradiction, since C is of the second kind. In the second case $\text{cyc}(C) = \text{cyc}(C_k \cdot \{u, v\}) = \text{cyc}(C_\ell) + \text{cyc}(P_m) \leq \frac{\ell}{2} + 1 + \frac{m}{2} \leq \frac{k-2}{2} + 1 < \frac{k}{2} + 1$, since paths are either 0- or 1-independent and $\ell + m = k - 2$ (note that in the latter case u and v cannot be consecutive). This is again a contradiction since C is of second kind.

We now prove point (2). Since the cycle C is of the second kind, it follows from the proof of Lemma 3.2.8, there is a 2-cycle with colored edge $\{u', v'\}$ (which is also an edge in M), and by point (1), vertices u' and v' are diagonal. So $C_k \cdot \{u', v'\}$ is the C_{k-2} cycle and it must be of the second kind, as otherwise $\text{cyc}(C_k) = \text{cyc}(C_{k-2}) + 1 = \frac{k-2}{2} + 1 = \frac{k}{2} < \frac{k}{2} + 1$. Obviously, $\{u', v'\}$ does not cross with any median edges of M. By shrinking this pair and using the induction on $C_k \cdot \{u', v'\}$ the proof is complete. ∎

Lemma 3.2.10 *An even cycle C is of the second kind if and only if there exists a matching M of C that is cross-free diagonal.*

Proof Assume $C = C_k$. The necessity follows from Lemma 3.2.9. Now, assume that there exists a cross-free diagonal matching M on vertices of C. It is easy to see that M contains edge $\{u, v\}$ where u and v are consecutive on C (note that M is a perfect matching, since we only consider circular medians). If we shrink the pair $\{u, v\}$, the resulting graph is C_{k-2} and remaining edges of M are a cross-free diagonal matching for C_{k-2}. We can complete the proof by induction on k, since $\text{cyc}(C_k) = 1 + \text{cyc}(C_{k-2}) = 1 + \frac{k-2}{2} + 1 = \frac{k}{2} + 1$, and the statement of the lemma is obviously true for $k = 2$. ∎

Lemma 3.2.11 *Let C be an even cycle. Deciding if C admits a cross-free diagonal matching is tractable.*

Proof Let $C_k = v_1 v_2 \dots v_k$, and k be even. We use the following greedy algorithm to compute a median M (in fact a classical algorithm for deciding if a parenthesis word is balanced):

1. $M = \emptyset$ ($v_1, v_2, \dots, v_k$ are not matched)
2. For $j = 1$ to k
   (a) if there exists i ($1 \leq i < j$), such that v_i and v_j are diagonal, v_i is not matched, and i is the maximum number with this property, then add $\{v_i, v_j\}$ to M (match v_i and v_j).
3. If all vertices are matched, C has a cross-free diagonal matching, otherwise it does not.

The time complexity of this algorithm is $O(k^2)$: we iterate the loop k times and for each j in the loop we check previous vertices to find the proper i. One can easily see that this can be done in linear time as follows:

1. $S = \emptyset$ is a stack.
2. For $j = 1$ to k
   (a) if the top element of S is diagonal with v_j, pop it from the stack S.
   (b) else, push v_j on S.
3. If S is empty, C has a cross-free diagonal matching, otherwise it does not.

Proposition 3.2.12 *Let C be an even cycle of size k ($k \geq 2$). Computing $cyc(C)$ can be done in time $O(k)$.*

Proof Immediate consequence of Lemma 3.2.8, Lemma 3.2.10, and Lemma 3.2.11. ∎

3.3 Proof of Theorem 3.2.1

We first prove that finding a median of a breakpoint graph of maximum degree two with only odd components can be done in polynomial time. We need the following lemma.

Lemma 3.3.1 *If B has maximum degree 2 and consists of two odd connected components, then computing a median of B is tractable.*

Proof Let H_1 and H_2 be the two odd connected components of B. Obviously, a median M contains an edge e between H_1 and H_2 (e is a H_1-crossing edge). If one of H_1 and H_2 is a path, then, by Lemma 3.2.5, we can assume e is connected to its terminal vertex. In either case, by shrinking e we get an even connected component or two even connected components, whose medians can be computed independently in polynomial time. There are at most $|V(H_1)| \times |V(H_2)|$ possible candidates for e, and for each candidate in time $O(|V(H_1)| + |V(H_2)|)$ we calculate the median. Hence computing a median of B is tractable in time $O(|V(H_1)| \cdot |V(H_2)| \cdot (|V(H_1)| + |V(H_2)|))$. ∎

Lemma 3.3.2 *If B has maximum degree 2, then there exists a median of B such that every odd connected component of B is connected by median edges to exactly one other odd connected component.*

Proof Let M be the median described in Lemma 3.2.6, and from Lemma 3.2.5 we can assume that for each odd path there exists exactly one crossing edge of M connected to its terminal vertex. Now consider an odd component H and one of its crossing edges, say e, that connects H to another odd component H'. By shrinking e we get $(H \cup H') \cdot e$ which is a set of even components and it is then independent.

So M can be modified into a median without any edge leaving $H \cup H'$. This means that H and H' are linked together and nothing else. Using this argument for other odd components and the fact that the number of odd components is even (because the number of vertices in the breakpoint graph is even), we arrive to a median that satisfies the lemma (see Fig 3.2). ∎

Proposition 3.3.3 *If B is a breakpoint graph with $2n$ vertices and consisting of only odd cycle(s)/path(s), then computing a median of B can be done in $O(n^3)$.*

Proof We first define a complete edge-weighted graph K_B as follows:

1. each connected component C defines a vertex v_C;
2. each edge $\{v_C, v_D\}$ has weight $\text{cyc}(C \cup D)$

By Lemma 3.3.1, K_B is computable in polynomial time. We claim it is computable in $O(n^3)$. Suppose B has t components and $n_1, \ldots, n_t$ are the number of vertices in each component. So we have $n_1 + \ldots + n_t = 2n$. The time to construct K_B is of order

$$\sum_{i<j} n_i \cdot n_j \cdot (n_i + n_j) = \sum_{i<j} n_i^2 \cdot n_j + n_i \cdot n_j^2 = \frac{1}{3}((2n)^3 - (n_1^3 + \ldots + n_t^3)) \leq \frac{8}{3}n^3.$$

Finally, by Lemma 3.3.2 we only need to find a maximum weight matching for K_B, which can be done in $O(n^3)$ by using Edmonds's algorithm [12] (one can consider any algorithm for finding the maximum matching without changing the total complexity, as long as it is in $O(n^3)$, since constructing K_B is in $O(n^3)$). ∎

If the breakpoint graph B has maximum degree 2, its connected components are paths and cycles. By Lemma 3.2.6 and Proposition 3.2.12 we can find the median edges for even components independently. Finally for odd components we find the median edge by Proposition 3.3.3.

Handling vertices of degree 3. We now assume that $B(G_1, G_2, G_3)$ has maximum degree 3. We consider all possibilities for matching the vertices of degree 3. A vertex u of degree 3 can be matched in two ways:

- to another vertex of degree 3. By shrinking these two vertices, obtain a smaller graph with fewer vertices of degree 3;
- to another vertex of degree less than 3. This implies that one of the edges incident to u is not in any alternating cycle, and we can remove this edge and transform u into a vertex of degree 2 (Fig. 3.8).

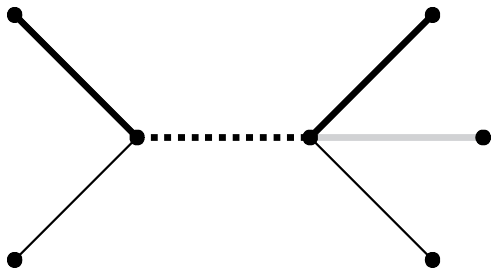

Figure 3.8: The dashed edge is a median edge. The gray edge cannot be in any alternating cycle.

Now for each i, $0 \leq i \leq \ell$, we can select $2i$ vertices among all m vertices of degree 3, and consider a perfect matching on these $2i$ vertices in $O(m^{2i})$, and remove an incident edge to each remaining vertex of degree 3 in $O(3^m)$ ways (more precisely, $O(3^{m-2i})$). Finally for the remaining graph, after removing all vertices of degree 3 it has maximum degree 2 and we can find a median in time $O(n^3)$. So in this case we can handle vertices of degree 3 in time $O(n^3 \cdot (\ell + 1) \cdot (3^m \cdot m^{2\ell} + 1))$. Note that m can be equal to 0. ∎

Chapter 4

Conclusion

In this work, we specified a large class of tractable instances for the DCJ median problem (with circular median and mixed genomes). In fact, we proved that only the vertices of degree 3 make the problem intractable. Also, by removing k edges from the breakpoint graph and decreasing its maximum degree, score of its median is not smaller than k units less than the score of the main median (i.e. the current score + k is an upper bound for the score of the main median). Finally, we showed there is an FPT algorithm for the DCJ median problem, if there always exists a median such that the number of its edges connecting two vertices of degree 3 is bounded.

From a theoretical point of view, it raises several interesting questions. First, it leaves open the possibility that the DCJ median problem is FPT. The next obvious problem is to extend our approach to the case of a mixed or linear median. This would require to understand better the combinatorics of odd paths in the breakpoint graphs.

Another interesting question is about expanding the breakpoint distance toward the DCJ distance: As we saw in Chapter 2, for two genomes G_1 and G_2 on n genes, their breakpoint distance is equal to

$$d_{\mathrm{BP}}(G_1, G_2) = n - a(G_1, G_2) - \frac{1}{2}e(G_1, G_2).$$

The parameters $a(G_2, G_2)$ and $e(G_1, G_2)$ are also equal the number of 2-cycles and 1-paths (P_1) in the breakpoint graph $B(G_1, G_2)$, respectively. The DCJ distance of

these genomes is:

$$d_{\mathrm{DCJ}}(G_1, G_2) = n - c(G_1, G_2) - \frac{p(G_1, G_2)}{2},$$

where $c(G_1, G_2)$ and $p(G_1, G_2)$ are the number of (even) cycles and odd paths in the $B(G_1, G_2)$, respectively. This motivates us to define a dissimilarity function as follows:

$$d_{(i,j)}(G_1, G_2) = n - c_i(G_1, G_2) - \frac{1}{2}p_j(G_1, G_2),$$

where $c_i(G_1, G_2)$ is the number of (even) cycles with at most $2i$ vertices, and $p_j(G_1, G_2)$ is the number of odd paths with at most $2j - 1$ vertices.

By considering this dissimilarity, the median problem is tractable when $i = j = 1$, since $d_{(1,1)} = d_{\mathrm{BP}}$. By taking $i = j = \infty$ we have $d_{(\infty,\infty)} = d_{\mathrm{DCJ}}$, and the median problem would be intractable. A natural question is "how much i and/or j can be increased such that the median problem is still tractable?"

We have also tried to extend our result to the *DCJ halving problem* [23]. It seems that the techniques like finding independent subgraphs can help us solve this problem. We have not been able to solve the problem yet.

Appendix A

A Practical Heuristic for Finding a Median

In Chapter 2, we saw that computing a DCJ median is NP-hard. A natural question is: "Is there any practical heuristic for computing a median, and/or a criterion to see how good is a practical heuristic?" The answer to the first question is **yes**.

Definition For a given breakpoint graph $B = B(G_1, G_2, G_3)$, the *cost* of a circular genome X on B is:

$$\text{cost}(X, B) = d_{\text{DCJ}}(X, G_1) + d_{\text{DCJ}}(X, G_2) + d_{\text{DCJ}}(X, G_3).$$

So by definition, X is a median for B if and only if it has the minimum cost on B.

In the median graph $B_M = B_M(G_1, G_2, G_3)$ each colored edge can be in at most one alternating cycle, since the edges in one genome form a matching. So, by removing a colored edge, e, the number of alternating cycles decreases at most by 1. Since we only consider circular medians, this implies that if M' is a median for the new breakpoint graph $B' = B(G_1, G_2, G_3) - e$, then

$$\text{cost}(M', B) - 1 \leq \text{cost}(M, B) \leq \text{cost}(M', B).$$

More generally, if B'' is a breakpoint graph obtained from B by removing k edges and M'' is a median for B'', then

$$\text{cost}(M'', B) - k \leq \text{cost}(M, B) \leq \text{cost}(M'', B).$$

Since d_{DCJ} is a distance function, we have

$$d_{\text{DCJ}}(M, G_1) + d_{\text{DCJ}}(M, G_2) \geq d_{\text{DCJ}}(G_1, G_2),$$

$$d_{\text{DCJ}}(M, G_1) + d_{\text{DCJ}}(M, G_3) \geq d_{\text{DCJ}}(G_1, G_3),$$

$$d_{\text{DCJ}}(M, G_2) + d_{\text{DCJ}}(M, G_3) \geq d_{\text{DCJ}}(G_2, G_3).$$

So by adding these inequalities we obtain the following lower bound for the cost of the median:

$$cost(M, B) \geq \frac{1}{2}(d_{\text{DCJ}}(G_1, G_2) + d_{\text{DCJ}}(G_1, G_3) + d_{\text{DCJ}}(G_2, G_3)).$$

Hence, we have the following practical heuristic:

1. Let T be the induced subgraph on the vertices of degree 3 in B.
2. Find the maximum matching of T, and remove the edges in the matching. and call the resulting graph by B_1.
3. For each vertex of degree 3 in B_1, remove one of its incident edges, randomly, and name the resulting graph by B_2.
4. Compute a median for B_2, its cost, the number of edges have been removed from B to obtain B_2, and the lower bound for the cost of a median of B.

The following table represents some experimental results of this algorithm for mammalian genomes: human (1), common chimpanzee (2), bornean orangutan (3), rhesus monkey (4), house mouse (5), rat (6), dog (7), cow (8), and horse (9):

- **1st column:** the triple of genomes (by their indices),
- **2nd column:** the cost of the approximated median,
- **3rd column:** the number of removed edges from the initial breakpoint graph,
- **4th column:** the lower bound for the cost of a median,

- **5th column:** the percentage of the maximum error of the cost our approximated median, i.e. (the cost of the answer - the cost of a median)/(the cost of a median)×100:

Triple	**Cost**	**# removed edges**	**lower bound**	**max % of the error**
(1, 5, 9)	379	19	360	5.3
(1, 6, 9)	282	11	271	4.1
(1, 6, 7)	317	21	296	7.1
(2, 6, 7)	322	21	301	6.9
(2, 5, 9)	385	20	365	5.5
(2, 6, 9)	286	10	276	3.6
(3, 5, 7)	415	29	386	7.5
(3, 5, 8)	450	33	417	7.9
(3, 5, 9)	372	13	359	3.6
(3, 6, 9)	277	11	266	4.1
(4, 5, 9)	372	16	356	4.5
(4, 6, 9)	275	13	262	4.9
(4, 6, 7)	311	24	287	8.4

As we see, this algorithm has small error on these data, and in practice it can be useful, since we are always aware of the maximum error of the answer. This algorithm has been implemented, and is accessible from

`https://sites.google.com/a/brown.edu/ahmad/thesis/DCJmedian.zip`.

Bibliography

[1] D. A. Bader, B. M. E. Moret, and M. Yan. A linear-time algorithm for computing inversion distance between signed permutations with an experimental study. *Journal of Computational Biology*, 8(5):483–491, 2001.

[2] V. Bafna and P. Pevzner. Sorting by reversals: Genome rearrangements in plant organelles and evolutionary history of x chromosome. *Molecular Biology and Evolution*, 12(2):239–246, 1995.

[3] A. Bergeron, J. Mixtacki, and J. Stoye. *Mathematics of Evolution and Phylogeny (édité par O. Gascuel)*, chapter The inversion distance problem, pages 262–290. Oxford University Press, 2005.

[4] A. Bergeron, J. Mixtacki, and J. Stoye. A unifying view of genome rearrangements. In P.Bucher and B.M.E. Moret, editors, *Algorithms in Bioinformatics, 6th International Workshop, WABI 2006, Zurich, Switzerland, September 11-13, 2006, Proceedings*, volume 4175 of *Lecture Notes in Computer Science*, pages 163–173. Springer, 2006.

[5] A. Bergeron, J. Mixtacki, and J. Stoye. A new linear time algorithm to compute the genomic distance via the double cut and join distance. *Theoretical Computer Science*, 410(51):5300–5316, 2009.

[6] B. Boussau, S. Blanquart, A. Necsulea, N. Lartillot, and M. Gouy. Parallel adaptations to high temperatures in the archaean eon. *Nature*, 456(1):942–945, 2008.

[7] D. Bryant. *The complexity of the breakpoint median problems.* Technical Report CRM-2579 Centre de recherches mathematiques, Universite de Montreal, 1998.

[8] A. Caprara. Sorting by reversals is difficult. In *RECOMB*, pages 75–83, 1997.

[9] A. Caprara. The reversal median problem. *INFORMS Journal on Computing*, 15(1):93–113, 2003.

[10] C. Chauve and E. Tannier. A methodological framework for the reconstruction of contiguous regions of ancestral genomes and its application to mammalian genomes. *PLoS Computational Biology*, 4(11):e1000234, 2008.

[11] C. Darwin. *On the origin of species*. Oxford University Press, 1998.

[12] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

[13] G. Fertin, A. Labarre, I. Rusu, E. Tannier, and S. Vialette. *Combinatorics of Genome Rearrangements*. The MIT Press, 2009.

[14] D. Graur and W. H. Li. *Fundamentals of Molecular Evolution*. Sinauer Associates, Inc., second edition, 2000.

[15] S. Hannenhalli and P. Pevzner. Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing, 29 May-1 June 1995, Las Vegas, Nevada, USA*, pages 178–189. ACM, 1995.

[16] S. Hannenhalli and P. Pevzner. Transforming men into mice (polynomial algorithm for genomic distance problem). In *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, 23-25 October 1995.*, pages 581–592. IEEE Computer Society Press, 1995.

[17] G. Jean and M. Nikolski. Genome rearrangements: a correct algorithm for optimal capping. *Information Processing Letters*, 104(1):14–20, 2007.

[18] A. P. Lee, S. Y. Kerk, Y. Y. Tan, S. Brenner, and B. Venkatesh. Ancient vertebrate conserved noncoding elements have been evolving rapidly in teleost fishes. *Molecular Biology and Evolution*, 28(3):1205–1215, 2011.

[19] M. Muffato and H. R. Crollius. Paleogenomics in vertebrates, or the recovery of lost genomes from the mist of time. *BioEssays*, 30:122–134, 2008.

[20] M. Ozery-Flato and R. Shamir. Two notes on genome rearrangement. *Journal of Bioinformatics and Computational Biology*, 1:71–94, 2003.

[21] I. Pe'er and R. Shamir. The median problems for breakpoints are NP-complete. *Electronic Colloquium on Computational Complexity (ECCC)*, 5(71), 1998.

[22] 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. *Proceedings of the National Academy of Sciences of the United States of America*, 89:6575–6579, 1992.

[23] E. Tannier, C. Zheng, and D. Sankoff. Multichromosomal median and halving problems under different genomic distances. *BMC Bioinformatics*, 10:120, 2009.

[24] D. B. West. *Introduction to Graph Theory*. Prentice Hall, second edition, 1996.

[25] A.W. Xu. A fast and exact algorithm for the median of three problem: a graph decomposition approach. *Journal of computational biology*, 16(10):1–13, 2009.

[26] A.W. Xu. DCJ median problems on linear multichromosomal genomes: Graph representation and fast exact solutions. In F. Ciccarelli and I. Miklós, editors, *Comparative Genomics, International Workshop, RECOMB-CG 2009, Budapest, Hungary, September 27-29, 2009. Proceedings*, volume 5817 of *Lecture Notes in Computer Science*, pages 70–83. Springer, 2009.

[27] A.W. Xu and D. Sankoff. Decompositions of multiple breakpoint graphs and rapid exact solutions to the median problem. In K. A. Crandall and J. Lagergren, editors, *Algorithms in Bioinformatics, 8th International Workshop, WABI 2008, Karlsruhe, Germany, September 15-19, 2008. Proceedings*, volume 5251 of *Lecture Notes in Computer Science*, pages 25–37. Springer, 2008.

[28] S. Yancopoulos, O. Attie, and R. Friedberg. Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinformatics*, 21(16):3340–3346, 2005.