Genome rearrangement problems accounting for duplicate genes

by

Aniket Charuchandra Mane

B.E. Mechanical Engg., BITS-Pilani, Goa Campus, 2015

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
Department of Mathematics
Faculty of Science

SIMON FRASER UNIVERSITY
Summer 2018

Approval

Name:	**Aniket Charuchandra Mane**
Degree:	**Master of Science (Mathematics)**
Title:	**Genome rearrangement problems accounting for duplicate genes**
Examining Committee:	**Chair:** Razvan Fetecau Associate Professor
	Cedric Chauve Senior Supervisor Professor
	Tamon Stephen Supervisor Associate Professor
	Leonid Chindelevitch Internal Examiner Assistant Professor School of Computing Science
Date Defended:	**August 10, 2018**

Abstract

We investigate certain genome rearrangement problems studied in relation to genome evolution. We introduce the SCJ-TD-FD rearrangement model to explain the directed evolution from an ancestor A to a descendant D, where D may contain multiple copies of genes from A. We study the *pairwise genome distance problem* that aims at finding the most parsimonious sequence of cuts, joins and single-gene duplications that transforms A to D, under this model. Next, we study the *rooted median problem* under the SCJ-TD-FD model, for which the problem is shown to be NP-hard. We provide an Integer Linear Program that, on simulated data, predicts an optimal median with high accuracy. Finally, we study the *Small Parsimony Problem* under the SCJ-TD-FD model that aims at finding the gene orders at the internal nodes of a given species tree. We define an ILP-based approach to reconstruct the ancestral gene orders and present our experiments on a data-set of *Anopheles* mosquito genomes.

Keywords: genome rearrangements, duplications, Single-Cut-or-Join, distance, parsimony, median.

Dedication

To my beloved parents.

Acknowledgements

First and foremost, I would like to thank my senior supervisor Dr. Cedric Chauve, who introduced me to the field of comparative genomics. I have learned a lot from his valuable insights and feedback. This thesis would be incomplete without his guidance and support.

I am also grateful to Dr. Tamon Stephen for his valuable guidance and advice. I would also like to take this opportunity to express my gratitude towards Dr. Razvan Fetecau and Dr. Leonid Chindelevitch, for investing their valuable time for my thesis defense.

I would like to specially thank my collaborators, Dr. Pedro Feijão and Dr. Manuel Lafond for providing crucial inputs during the course of this project.

I am thankful to my friends who have supported me at SFU.

Last but not the least, I wish to thank my family for their constant support that enabled me to focus on my studies. I will always be grateful to them for their love and patience.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Evolution is the process that is responsible for the diversity of life in nature. The genome of an organism evolves over time through a variety of mechanisms, leading to changes in the structure or size of the genome. The genome of a species consists of sets of chromosomes that are organized into an ordered sequence of genes, referred to as gene order. The change in the gene order of a genome is one of the mechanisms responsible for its evolution. Such a shuffling of gene order in the chromosome was first reported by Dobzhansky and Sturtevant when they compared the genomes of fruit flies from various parts of North America [24]. According to their observations, the changes observed in the order of genes in various species could be explained using a sequence of inversions. For instance, in Figure 1.1, the inversion of the order of genes from C to G indicates the difference between the genomes of two strains of fruit fly.

However, the shuffling of genes in the gene order is not the only mechanism of genome evolution. Among other events, the duplication of genes also plays a major role in evolution of genomes, even displaying variations in gene function for different copies of the same gene [47, 39]. The changes in the arrangement or content of genes along chromosomes are collectively referred to as *genome rearrangements*.

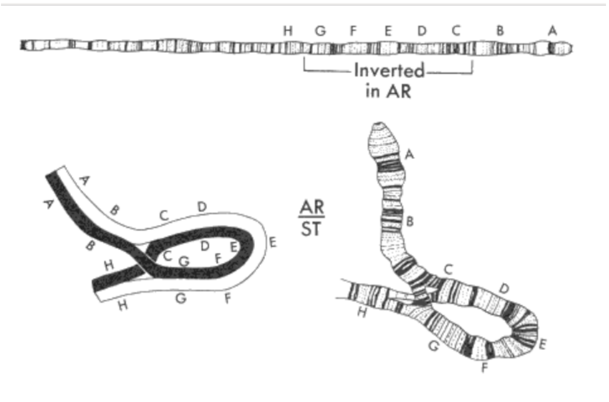


Figure 1.1: Inversions in *Drosophila* chromosomes [24]

The evolutions of a species can take place either on a nucleotide level or on a gene level. It was previously studied by comparing sequences of nucleotides or amino acids. Genome rearrangements are extremely rare as compared to nucleotide level mutations, and so provide the ability to trace back the evolutionary history to a more distant past [59, 58]. As mentioned above, the evidence of genome rearrangements as an important evolutionary mechanism was first presented in the 1930s. However, subsequent research did not consider changes in gene order as a mode of evolution until the early 1980s, due to the lack of available data in the form of assembled genomes or genome maps. The development of genome sequencing and mapping techniques and genome assembly algorithms spurred a renewed interest in the field of genome rearrangements [60, 84, 62]. In the early 1990s, the search for a structured approach to analyze genome rearrangements led to the development of computational questions that started a vibrant research direction in computational biology [73, 75].

The extent of evolution between two genomes can be determined through the number of evolutionary events that can explain the difference between them. The number of evolutionary events taking place can be estimated using the notion of evolutionary distance, which is based on the principle of "Occam's razor", also known as the principle of parsimony. While the assumption of parsimony may not lead us to the biological truth, the evolutionary hypothesis using the least number of events (genome rearrangements, in this case) is considered to be a reasonable approximation of reality, making it a reliable source of evolutionary information. Under the assumption of parsimony, the distance between two genomes is measured as the least number of genome rearrangements required to transform one genome to another. The problem of finding the distance between genomes is mainly studied in two situations. If the two genomes both belonged to extant species, the distance tells how closely related they are. On the other hand, if one of the species is an ancestor of the other, the distance measures the degree of evolution from the ancestor to the descendant; this last context, that we consider in this thesis, has received much less attention so far due to the lack of assembled ancient genomes.

Computing the distance between two given genomes with no restrictions on their gene content is not an easy problem. Hence, initial research focused on genomes with equal gene content, i.e. without the presence of duplicate genes. Under the assumption of equal gene content, finding the distance between two genomes is tractable for a wide range of models of evolution through genome rearrangements (see [35, 86] for early results and [29] for a survey dating back to 2009). Moreover, some genome rearrangement models can even handle unequal gene content resulting from gene loss and new gene creation [21, 12]. However, the addition of duplications as a possible event makes the problem of computing the distance intractable for most models (see [79][81] for recent examples). Note however that there exist some tractable models which involve only large-scale duplication events such as whole genome duplication [28] or whole chromosome duplication [88]. Yet, there were no models that could account for single-gene duplications. To answer this problem, we introduce an evolutionary model using single-gene duplications, building on the model discussed in [28], that computes an evolutionary scenario in polynomial time.

The distance problem considers only two genomes at a time. However, the notion of parsimony can be extended to three or more genomes. The median problem entails computing a common ancestor of three or more genomes that minimizes the total distance from the ancestor to each of the descendants. The median problem is the simplest instance of a problem of reconstructing the gene order of an ancestral species. The problem has been found to be tractable in specific but limited cases. On restricting the genome structure to contain only linear or circular chromosomes, the median problem is intractable for most erstwhile rearrangement models [63, 15]. However, on relaxing this structural restriction, Sankoff et al. found a polynomial time algorithm to compute the median [83] in a model called the breakpoint model. A few years later, Feijão and Meidanis introduced a related rearrangement model, called the Single-Cut-or-Join (SCJ) model, for which the median problem could also be solved in polynomial time [28].

The Small Parsimony Problem can be seen as the generalization of the median problem that considers the reconstruction of more than one ancestral gene order. In this problem, the evolutionary tree, known as the species tree, depicting the relationship between all the species is provided. While the median problem requires to compute a single ancestor, the Small Parsimony Problem aims at finding all ancestors along the species phylogeny, for a given set of extant species (see Figure 1.2). This problem is intractable for most models with the notable exception of the Single-Cut-or-Join model, under which it is tractable [28]. The SCJ model is a set theoretic model based on the notion of adjacencies that can be used to determine the gene order of a genome. A cut indicates the breaking (or removal) of an adjacency from the set while a join is the introduction of an adjacency. This problem has been studied using multiple strategies - both distance-based and otherwise [46]. We provide Integer Linear Programming based approaches to address both the median as well as the Small Parsimony Problem, using a duplication-aware version of the SCJ model.

The Small Parsimony Problem is one of the key problems investigated in relation to ancestral reconstruction. The closest approximation of ancestral reconstruction however, is the Large Parsimony Problem, which is an extension of the Small Parsimony Problem with no evolutionary tree provided. This problem has not studied comprehensively, since it is even more difficult than the Small Parsimony Problem.

The remaining parts of the thesis have been arranged as follows. Chapter 2 discusses certain biological terms that may be used frequently throughout the discussion. Chapter 3 provides a brief history of previously studied genome rearrangement models. Chapter 4 introduces the new rearrangement model that accounts for single-gene duplications. It also provides a tractable algorithm to compute the optimal parsimonious scenario. Chapter 5 uses the novel distance to compute the optimal median genome for two versions of the

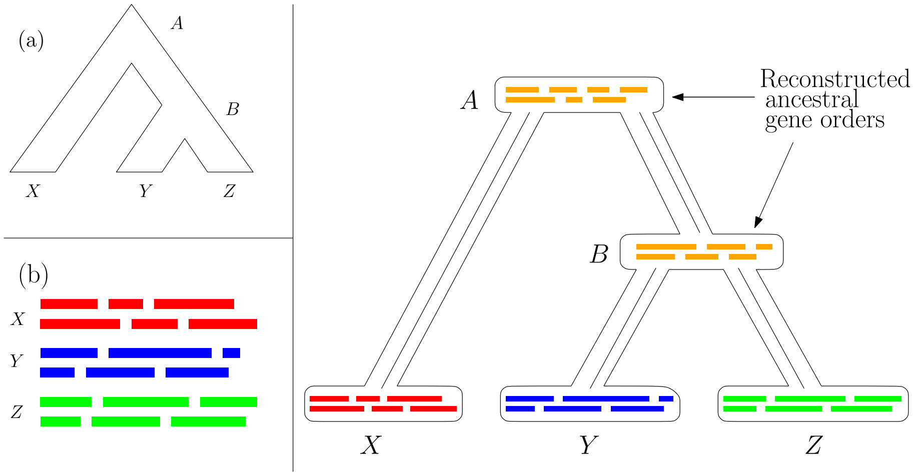


Figure 1.2: Reconstruction of ancestral gene orders A and B using (a) the species phylogeny and (b) extant gene orders

median problem. We also present our results on simulated data in both these chapters. Chapter 6 reuses the same evolutionary model to solve the Small Parsimony Problem. We then report our findings on a data set of Anopheles mosquito genomes.

Chapter 2

Basics of computational biology

This thesis introduces a genome rearrangement model and looks at some problems studied mainly in relation to the ancestral reconstruction of genomes. However, before proceeding to the mathematical part of the discussion, it serves well to establish the definitions of some important biological terms.

2.1 Genes and genomes

Deoxyribonucleic acid (DNA) is a complex organic molecule that contains the genetic instructions for the development and function of living things. DNA molecules are composed of four basic molecules called *nucleotides*, each consisting of a five carbon sugar (2'-deoxyribose), a phosphate group and one of the four bases - adenine (A), cytosine (C), guanine (G) and thymine (T) [78]. DNA molecules commonly possess a chain-like double-helix structures with two nucleotides sequences, called strands, wound against each other in opposite directions (see Figure 2.1).

Genes are the basic functional units of heredity. Each gene consists of a continuous stretch along the DNA, contributing to a specific function or trait. The existence of genes was first suggested by Gregor Mendel through his experiments regarding hereditary charac-

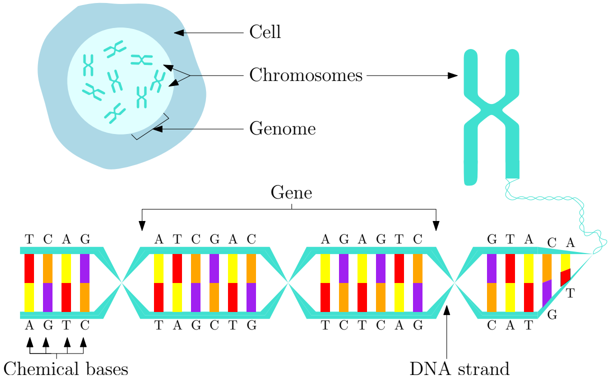


Figure 2.1: Schematic diagram of a DNA segment

teristics of pea plants. Recording certain physical traits such as color and size displayed by the child as well as its parents, Mendel noticed a pattern in the pea plants, which suggested that the peas had inherited their physical characteristics from their parents.

The DNA is molecule is packaged into compact thread-like structures called *chromosomes*, by tightly coiling around proteins known as histones. It carries the hereditary information of the organism in the form of genes, arranged in an oriented sequence. Chromosomes can be linear or circular in structure. The complete set of chromosomes in an organism is called a *genome*. Thus, the genome contains the entire hereditary information of the organism.

2.2 Evolution

Evolution is the gradual change in the genetic structure of an organism. Evolutionary information is stored in the genome, and is passed on by one generation to the next. Organisms can evolve mainly through two types of events - *point mutations* and *genome rearrangements*. Point mutations result from changes affecting a single nucleotide in the DNA sequence. These are local changes such as substitution of a nucleotide by another or insertion or deletion of nucleotide. Point mutations may be caused due to errors in the process of DNA replication during cell division. In some case, mutations also occur as a result of damage sustained by the DNA. The rate at which such mutations occur may increase due to exposure to ultaviolet (UV) radiation or mutagenic chemicals [9].

Genome rearrangements are evolutionary events that alter the structure or size of the genome on a gene level. Numerous events can bring about these changes, some of which have been illustrated in Figure 2.2. In the figure, each gene g has a head and a tail, g_h and g_t respectively. Duplications, gene creations and losses alter the content of the genomes (indicated by arrows). Inversions and translocations change the sequence of genes (indicated by segments). The study of rearrangements plays an important role in the analysis of genome evolution. Here we take a look at some of the important evolutionary events responsible for changes in the gene order.

2.2.1 Gene duplications

Gene duplications are evolutionary events in which gene segments of variable length are duplicated. In some cases, even entire chromosomes or genomes might be duplicated [88]. Duplications have been shown to be an important mechanism of genome rearrangement [39][44]. Gene duplications result from various mechanisms such as unequal cross-over, replication slippage or whole chromosome (or whole genome) duplication [89]. Unequal cross-over often leads to tandem duplications, a consecutive sequence of duplicated genes appearing immediately next to the original sequence. Replication slippage occurs due to misalignment of the two strands while DNA replication, leading to duplication of short gene sequences.

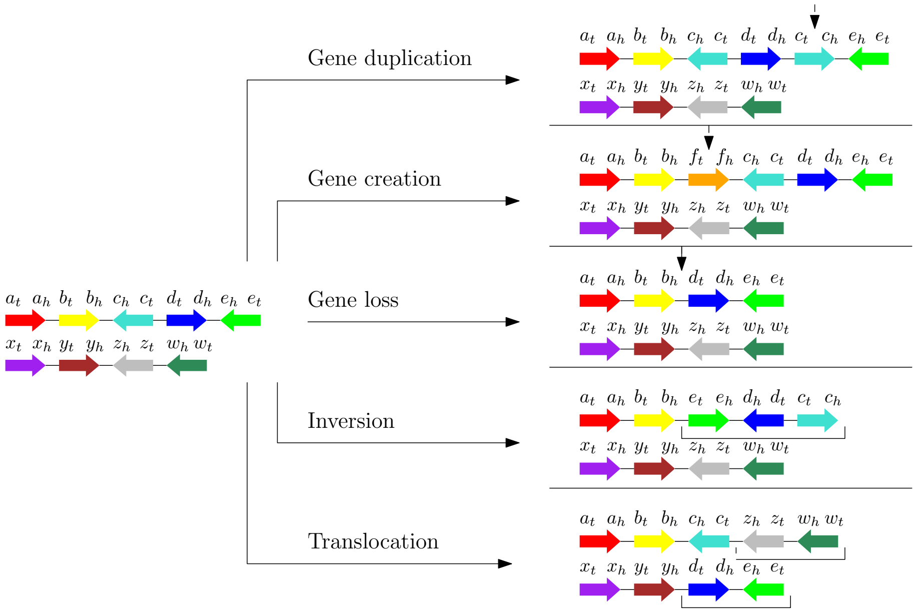

a_t a_h b_t b_h c_h c_t d_t d_h e_h e_t
x_t x_h y_t y_h z_h z_t w_h w_t
Gene duplication
a_t a_h b_t b_h c_h c_t d_t d_h c_t c_h e_h e_t
x_t x_h y_t y_h z_h z_t w_h w_t
Gene creation
a_t a_h b_t b_h f_t f_h c_h c_t d_t d_h e_h e_t
x_t x_h y_t y_h z_h z_t w_h w_t
Gene loss
a_t a_h b_t b_h d_t d_h e_h e_t
x_t x_h y_t y_h z_h z_t w_h w_t
Inversion
a_t a_h b_t b_h e_t e_h d_h d_t c_t c_h
x_t x_h y_t y_h z_h z_t w_h w_t
Translocation
a_t a_h b_t b_h c_h c_t z_h z_t w_h w_t
x_t x_h y_t y_h d_t d_h e_h e_t

Figure 2.2: Examples of genome rearrangement events

Chromosome or genome duplication result from the failure of the daughter chromosomes to separate after DNA replication. This type of duplication has been known to be common in plants [20]. After duplication, the two copies of the gene evolve independently of each other. Despite their similarity, the copies may perform altogether different functions.

2.2.2 Fusions and fissions

Fusions and fissions also play an important role in genome evolution. A *fusion* event refers to the connection of two previously disjoint gene extremities. A *fission* event denotes the separation of two previously consecutive genes in a chromosome. Fusions and fissions may be the product of various other genome rearrangement events, such as inversions, translocations or gene losses, discussed below. Fusions and fissions have been commonly observed in mammalian evolution [87, 43].

2.2.3 Inversions and translocations

An *inversion* involves a reversal of a segment of DNA within a chromosome. Thus, during an inversion event, the gene order as well as the orientation of each gene in the segment is reversed. Genome rearrangement through a sequence of inversions was one of the first problems studied in this field. Inversions can have major functional consequences and have been associated with genetic disorders [66]. A *translocation* is the relocation of part of one chromosome to another chromosome. Chromosomal translocations have been known to be a common characteristic in cancer genomes [41]. Inversions and translocations may result in the fusion or fission of chromosomes.

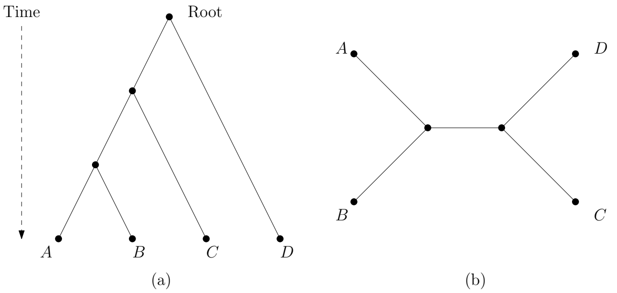


Figure 2.3: An example of (a) rooted and (b) unrooted phylogenetic X-trees

2.2.4 Gene loss

Gene losses are important in shaping the content of a genome. They can either result from a gradual loss of function or due to an abrupt mutational event. Gene losses have been shown to be a recurrent phenomenon in bacterial genomes [40][56] or in Drosophila genomes [34].

2.3 Phylogenetic trees

The evolutionary relationships between various biological entities are often described using connected acyclic graphs called *phylogenetic trees*. From an evolution perspective, the leaves (nodes of degree 1) of the phylogenetic tree represent currently existing entities, such as existing species or genes whereas interior nodes (degree 2 or more) represent their ancestors. The edges or branches between two nodes indicate the extent of evolution, usually in terms of time or number of mutations. For a set of entities X, the phylogenetic tree with the set X as its leaves is called the *phylogenetic X-tree*. The entities belonging to the set X (leaves) are called *extant* entities.

Phylogenetic trees can be *rooted* or *unrooted*. Figure 2.3 shows an example of rooted and unrooted phylogenetic X-trees with $X = \{A, B, C, D\}$. A rooted phylogenetic tree is generally used in the context of directed evolution. It has a unique node, called *root*, that represents the most recent common ancestor of the species at the leaves. All nodes of the tree are directed away from the root. Thus, by assigning appropriate directions to every branch, it can ensured that there exists a directed path from the root to each node, which is unique to the node. Depending on the direction of the branch, the source of the branch is called the *parent* while the sink is called the *child* of the parent.

A rooted tree in which all internal nodes have at most two outgoing branches is called a *rooted binary* tree. Similarly, an unrooted tree in which each internal node has degree at most three is called a *unrooted binary* tree. Binary trees provide a complete evolutionary scenario since there is a clear distinction between consecutive edges of the tree, making it possible to determine the immediate descendants for any internal node of the tree. Unless specified, we will assume all trees to be rooted binary trees.

2.3.1 Species trees

The phylogenetic X-tree defined on the set X of extant species is called the *species tree*. The set of ancestral species is represented by the internal nodes of the species tree. For any branch along a rooted species tree, the species closer to the root is the direct ancestor of the species farther from the root. Thus, the species tree contains the entire evolutionary history of a set of species. The branching at any ancestral node of the tree refers to a *speciation* event.

2.3.2 Gene trees

Similar to a species tree, a gene tree contains essential information about the history of a set of genes belonging to the same family. The phylogenetic X-tree defined on the set X of extant genes is called the *gene tree*. The internal nodes of the gene tree represent ancestral genes. The branching at each internal node is either a speciation event or a *duplication* event. The immediate descendant genes after a speciation belong to different species while those after a duplication event belong to the same species. Ancestral genes might also undergo loss or transfer. However, these events are ignored for the purposes of this discussion.
In Figure 2.4 (b), we see a gene tree inscribed inside a tube-like species tree. Species A is the root while M is its descendant. Leaves D_1 and D_2 are the descendants of M. The gene tree belongs to the blue gene family from part (a). The squares represent duplication events along the branch while dots represent speciation events. For instance, $g_1 \in M$ undergoes speciation to produce $g_1^1 \in D_1$ and then further undergoes duplication to produce $g_1^1, g_1^2 \in D_2$.

2.3.3 Reconciliation

Given a species tree and a gene tree, one of the key questions in the analysis of genome rearrangements is recognizing which gene in the tree belongs which species. However, gene trees may evolve differently as compared to a species tree due to events such as duplications,

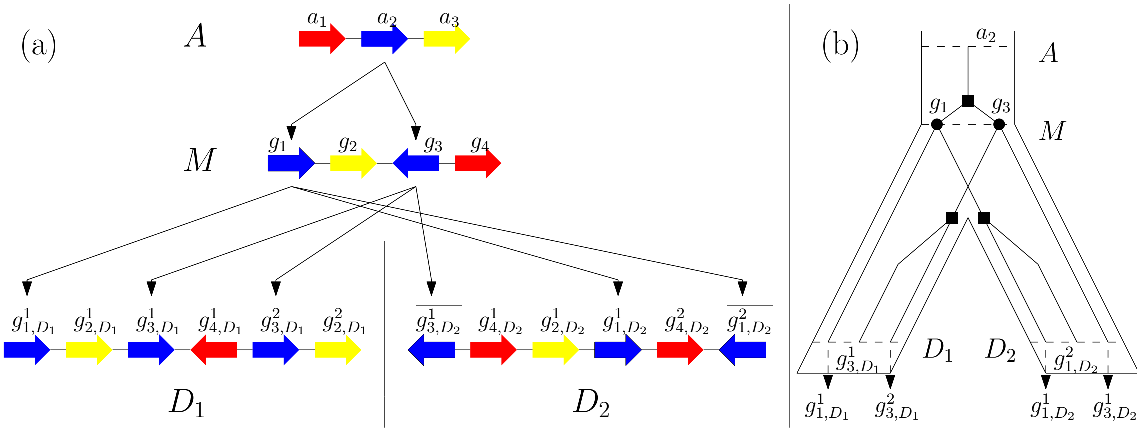


Figure 2.4: a) Orthology relations and b) corresponding reconciliations

incomplete lineage sorting or lateral gene transfers. Since the given trees may not always agree with each other, obtaining this information is not obvious. As the gene orders of the extant species are already available, it is easy to identify each extant gene with its source species. However, this is not possible for the ancestral genes and species.

As discussed before, each ancestral gene of the gene tree might undergo either speciation or duplication. Reconciliation induces a speciation or duplication event at each gene in the gene tree and at each branch in the species tree [16]. This information is necessary for understanding the relationship between different genes originating from the same ancestor in a gene tree. Reconciliation is the process of mapping the genes of a given tree with their respective species in the species tree.

Lowest Common Ancestor mapping: For a tree T, let $V(T)$ and $E(T)$ denote its vertices and edges, respectively. For a subset of leaves X, the *lowest common ancestor* of X, denoted by $lca_T(X)$ is the internal node farthest from the root of T, that is ancestor of every leaf $l \in X$. Let T_v be the subtree of T with an internal node v as the root. Further for an internal node $v \in T$, let $L_T(v)$ denote the set of extant entities in T_v. Thus, for $v \in T$, $v = lca_T(L_T(v))$. Consider a gene tree G and a species tree S. The leaves of G are already labeled with the species to which they belong. A *reconciliation of G with S* is a mapping $M : V(G) \rightarrow V(S)$ such that for $g \in G$ and $s \in S$, $M(g) = s$ if $s = lca_S(L_G(g))$ [17]. Let $c(g)$ denote a child of g in G. If $M(c(g)) = s$, then there is a duplication at g, otherwise there is a speciation at g.

Orthology relations: Consider two genes g_1 and g_2 and let $lca_G(\{g_1, g_2\}) = g$. If the event at g is a speciation then g_1 and g_2 are orthologs while if it is a duplication, then they are paralogs. Orthology relations help in determining which ancestry of a gene [32]. For instance, in Figure 2.4, the genes g^1_{1,D_1} and g^1_{1,D_2} are orthologs while g^1_{1,D_1} and g^1_{3,D_1} are paralogs.

In the current context, reconciliation will be used to obtain the orthology relations between genes, as will be discussed subsequently. However, it also has applications in inferring a species tree from discordant gene trees [71] or reconstructing the gene trees using others [68]. Numerous softwares have been proposed for reconciliation, relying on approaches based on parsimony (minimize the number of evolutionary events) [36] or probability (maximum likelihood) [33].

2.4 Genome representation

A genome consists of a set of *chromosomes* which are maximal contiguous sequences of genes. A chromosome with k genes can have either $k - 1$ adjacencies, in which case it is a *linear* chromosome, or k adjacencies, in which case it is a *circular* chromosome.

Definition 2.4.1. A gene is an oriented piece of DNA, identified by a *head* and a *tail*, both of which are called *gene extremities*.

In this representation, gene x is represented using a pair of gene extremities (x_t, x_h), x_t denotes the tail of the gene x and x_h denotes its head.

Definition 2.4.2. The sequential ordering of oriented genes along a chromosome is referred to as *gene order.*

Each chromosome can be linear or circular. In our examples, a circular chromosome is represented using round brackets (*e.g.* $(a, \overline{b}, c)$) while a linear chromosome is represented using square brackets (*e.g.* $[a, \overline{b}, c]$), where a gene b in reverse orientation is denoted by $\overline{b}$. Alternatively, a genome can be represented by a set of *gene extremity adjacencies.*

Definition 2.4.3. An *adjacency* is an unordered pair of gene extremities that are adjacent in a genome.

For example $(a, \overline{b}, c)$ is encoded by the set of adjacencies $\{a_h b_h, b_t c_t, c_h a_t\}$ and $[a, \overline{b}, c]$ by $\{a_h b_h, b_t c_t\}$.

We assume that a given gene a can have multiple copies in a genome, with its number of occurrences being called its *copy number.* A genome in which every gene has copy number 1 is a *trivial genome* [85]. In this context, a non-trivial genome sometimes cannot be represented unambiguously by a set of adjacencies unless we distinguish the copies of each gene, for example by denoting the copies of a gene a with copy number k by $a^1, \ldots, a^k$. For example, the genome $(a^1, \overline{b}, c^1, a^2), [\overline{a^3}, d, c^2]$, with two duplicated genes of respective copy numbers 3 and 2, is represented by $\{a^1_h b_h, b_t c^1_t, c^1_h a^2_t, a^2_h a^1_t, a^3_t d_t, d_h c^2_t\}$. We call a *gene family* the set of all copies of a gene that is present in both considered genomes. A gene family is trivial if it has exactly one copy in both genomes. From now, we identify a genome with its multi-set of adjacencies.

Let A and D be the respective adjacency sets for the genomes $[a, b, \bar{c}, d]$ and $[a, b, \bar{c}, \overline{d}][b, \bar{c}]$. Then the *multiset difference* between the two sets is denoted by $A - D$ (similarly $D - A$). Thus, if $A = \{a_h b_t, b_h c_h, c_t d_t\}$ and $D = \{a_h b_t, b_h c_h, c_t d_h, b_h c_h\}$ then $A - D = \{c_t d_t\}$ whereas $D - A = \{c_t d_h, b_h c_h\}$.

Finally, we describe two rearrangement events that will be used frequently throughout this discussion. Both events are closely related to the notion of an adjacency set, described above.

Definition 2.4.4. A *cut* is an event that deletes an adjacency from a genome.

Definition 2.4.5. A *join* is an event that creates an adjacency by joining two gene extremities that were previously not adjacent to any other extremity in the genome.

Figure 2.5 illustrates the cut and join events from A to D. The genome A and D can be respectively represented by their adjacency sets as $\{a_h b_h, b_t c_t, c_h d_t\}$ and $\{b_t c_t, c_h a_t, a_h d_h, \}$. The adjacencies in A that are missing from D are cuts while adjacencies present in D but not in A are joins. If a linear chromosome undergoes a cut, it gets split into two linear

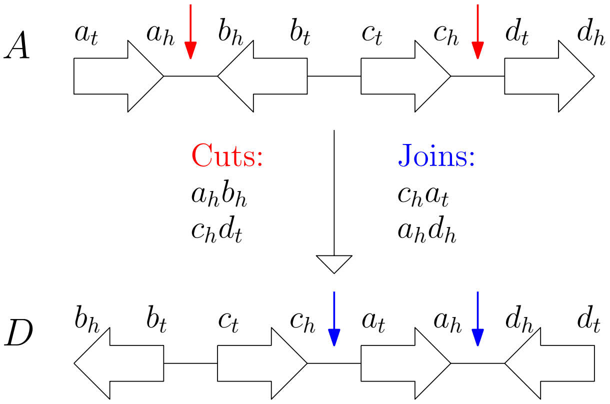


Figure 2.5: Cut and join operations on genome A leading to genome D

chromosomes while a cut in a circular chromosome results in a linear chromosome. Joining two distinct linear chromosomes yields a single linear chromosome while joining the free extremities of a linear chromosome results in a circular chromosome.

This chapter provided a background on the key aspects of computational biology that will be useful in the upcoming discussion on genome rearrangements. It also introduced some of the important evolutionary events that result in the change of gene order. In the next chapter, we will take a look at the main computational problems that are motivated by genome rearrangements. We will also review the important methods designed to solve these problems, in order to study the evolution of genomes through genome rearrangements.

Chapter 3

Genome rearrangement problems: A brief overview

The analysis of genome rearrangements started almost 80 years ago, when Dobzhansky and Sturtevant [24] observed that the evolution of certain Drosophila species could be explained using a sequence of reversals. In 1988, Jeffrey Palmer observed some interesting patterns in the evolution of plant organelles [62]. He compared the mitochondrial genomes of cabbages and turnips. About 99.9% of the genes were identical in both the genomes. However, it was noted that the gene orders of both these vegetables were considerably different. These discoveries along with similar findings suggested that genome rearrangements might play an important role in genome evolution [60, 50]. Up until then, evolution was traditionally explained through nucleotide-level changes in the DNA sequence. However, in the light of newly found evidence, pioneered by David Sankoff, novel approaches based on comparison of gene sequences were investigated [73, 75].

These events can be viewed as evolutionary "earthquakes" leading to chromosomal faults, ultimately resulting in disruption of gene order. In comparison to point mutations, genome rearrangements are rare events [69]. However, they can accumulate over time, prompting a clear distinction between the gene orders of the original and evolved genomes. As a result, the similarity between the gene orders of two species can reveal their proximity to each other. Thus, genome rearrangements act as good phylogenetic markers. Inevitably, combinatorial problems posed by genome rearrangements have attracted significant interest over the years. In this section, we provide a brief history of these problems.

3.1 Computational problems on genome rearrangements

The study of genome rearrangements involves solving a combinatorial "puzzle" to find the shortest sequence of rearrangements that can transform one genome into another. We are provided a set of genomes as input. Each genome is defined by the order of genes along the chromosomes. Considering certain structural constraints, each puzzle aims at finding the most parsimonious rearrangement scenario to explain the evolution between the genomes. Here we take a look at some important computational problems studied in relation to genome rearrangements.

Pairwise genome rearrangement distance: Given the gene orders of two input genomes, this problem aims at finding the most parsimonious sequence of genome rearrangements that can transform one genome into the other. For a particular instance of the problem, the set of genome rearrangements to be used to obtain such an evolutionary scenario is also specified. In the context of directed evolution, the distance is always computed from the ancestor to the descendant. In general, the distance between two genomes X and Y will be denoted as $d(X, Y)$.

Genome median: Given the gene orders of k genomes, G_1, G_2, ..., G_k, the genome median problem aims at computing their common ancestor M such the sum $\sum_{i=1}^{k} d(M, G_i)$ is minimized. The median problem is of theoretical interest as it can be used as an iterative step for computing the ancestral gene orders for the Small Parsimony Problem (discussed below).

Small Parsimony (SPP): In this problem, we are provided a species tree, with the leaves and internal nodes of the tree representing the extant and ancestral species, respectively. Given the gene orders of the extant species, the gene content of the ancestral species and the orthology relations for each gene family, the aim is to reconstruct the gene orders of the ancestral species while minimizing the sum of the genome rearrangement distances along the branches of the tree. The overall distance is called the SPP score.

The median problem forms the simplest instance of the Small Parsimony Problem, since it requires to compute only one ancestral gene order. The median problem can also be used to solve the SPP using an iterative approach [76]. In this approach, each ancestral node defines an instance of the median problem (the node itself representing the median M of its neighbors). Initially, each ancestral node is assigned a random gene order, governed by the gene content of the node. In each iteration, the gene order of an ancestral node is updated as the median of its three neighbors, if such an update improves the overall distance. This iterative technique has often been used for ancestral reconstruction under various frameworks [57].

The SPP has also been solved using other optimization-based approaches which reconstruct ancestors using scaffolding techniques that preserve contiguous genome segments or distance-based techniques (without the iterative process). These techniques will be discussed in detail in Chapter 6.

Large Parsimony: This problem is a harder version of the SPP as the species tree is not provided. The aim of this problem is to find a species tree that minimizes the score of the small parsimony problem associated with the tree. For n given extant genomes, the possible number of trees is $(2n-5)!!$ which renders the Large Parsimony Problem NP-hard in general.

3.2 Genome rearrangement models

We have seen that various events such as inversions, translocations and gene-duplication can alter the gene order. However, traditional frameworks under which the genome rearrangement problems are studied include only a few of these events at a time in their analysis. These frameworks, referred to as *genome rearrangement models*, are defined by a set of evolutionary events that explain the transformation of a given genome into another. Such a restriction enables us to solve the problem from a combinatorial perspective.

3.2.1 Reversal model

Palmer's discovery of a novel pattern in the evolution of plant organelles led to the observation that the transformation between the mitochondrial genomes of the two vegetables (cabbages and turnips) could be achieved by a series of inversions [62]. Subsequent research was motivated towards finding the least number of inversions to transform a genome into another. In one of his pioneering works, Sankoff imagined gene orders as permutations, with numbers used to denote genes [73, 75]. Thus, a *reversal* or *inversion* of a segment $(\pi_i, ..., \pi_j)$, applied to a gene order $(\pi_1, ..., \pi_n)$, $1 \leq i < j \leq n$ results in the gene order $(\pi_1, ...\pi_{i-1}, \pi_j, ..., \pi_i, \pi_{j+1}, ..., \pi_n)$.

Reversal distance: Given two permutations, each representing a gene order, the *reversal model* explains the rearrangement from one gene order to the other using a sequence of reversals. The *reversal distance problem* involves finding the shortest sequence of reversals that can convert a gene order into another [37]. The number of reversals in the shortest such sequence is the *reversal distance* between the two genomes in question.

Each gene in a permutation can be assigned an orientation or sign, depending on the DNA strand on which they are located. However, in initial reversal models, permutations were considered without the knowledge of the DNA strand on which the genes were located and were referred to as *unsigned* permutations. Further, the genomes consisted of a single chromosome without any duplicate genes. This limited the analysis to smaller genomes. Under these conditions, the reversal distance problem was deemed to be intractable [37]. For unsigned permutations, Caprara later confirmed the NP-hardness of computing the reversal distance [14]. There were, however, plenty of approximation results for the problem [38][3][18].

However, gene orders could be better represented through the notion of *signed* permutations, by assigning an orientation to each gene. A *signed reversal* of a segment $(\pi_i, ..., \pi_j)$, applied to a gene order $(\pi_1, ..., \pi_n)$, $1 \leq i < j \leq n$ would result in the gene order

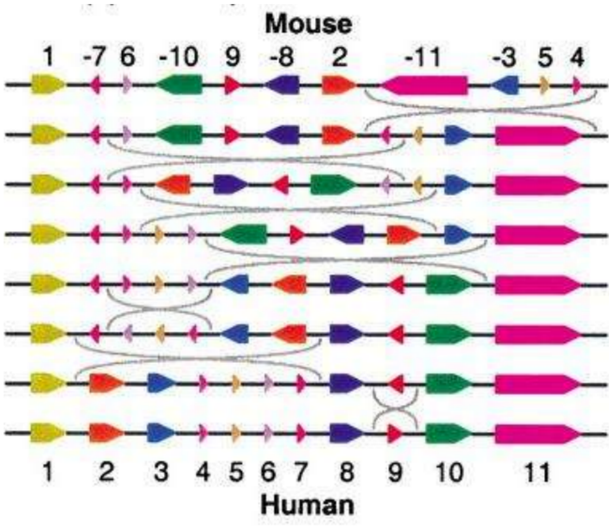


Figure 3.1: A sequence of inversions transforming a segment of a mouse genome to that of a human genome [64]

$(\pi_1, ...\pi_{i-1}, -\pi_j, ..., -\pi_i, \pi_{j+1}, ..., \pi_n)$. An example of genome rearrangement using signed reversals is illustrated in Figure 3.1. Although this change led to a larger search space for possible intermediate genomes, the orientations of the genes provided a better signal for choosing good reversals.

Breakpoint graph: Under this model, Hannenhalli and Pevzner proved that the reversal distance problem could be solved in $O(n^4)$ time for signed unichromosomal [35]. They used a concept called the breakpoint graph. This was one of the first tractability results in this field.

A breakpoint graph is defined for a pair of permutations. Consider a permutation $\pi_1, ..., \pi_n$ of genes $1, ..., n$. Two vertices $2i$ and $2i-1$ are introduced for each gene $i \in \{1, 2, ..., n\}$. Further, two extra vertices $\pi_0 = 0$ and $\pi_{n+1} = 2n+1$ are introduced. The breakpoint graph is then defined on the these $2n+2$ vertices. A *breakpoint* is defined by a pair (π_i, π_{i+1}) of consecutive genes in the permutation. A pair of vertices $(2\pi_{i-1}, 2\pi_i - 1)$ is joined by a black edge while a pair $(2i, 2i+1)$ is joined by a gray edge (see Figure 3.2). For the graph thus defined, the distance between the two permutations, $(1, 2, ..., n)$ and $(\pi_1, \pi_2, ..., \pi_n)$ is given by

$$d = n - c + h + f$$

where c is the number of cycles with alternate black and gray edges, h is the number of hurdles and f a binary variable called a fortress. The latter two (hurdles and fortress) are combinatorial constructs determined using the partial order defined by the permutations.

Subsequent algorithms have yielded a best known complexity of $O(n\log n)$ for the signed unichromosomal reversal distance problem [82]. In case of multichromosomal genomes, reversals alone may be insufficient to transform a genome into another. However, reversals

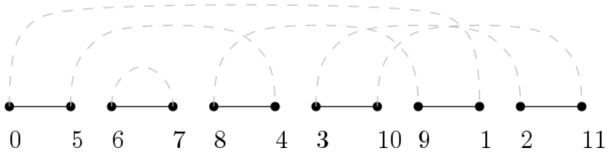


Figure 3.2: A breakpoint graph between two permutations, one denoted by black edges, the other by gray edges

Model	Pairwise distance	Genome median
Unsigned, unichromosomal	NP-hard[14]	NP-hard[15]
Signed, unichromosomal	$O(n\text{log}n)$ [82]	NP-hard[15]

Table 3.1: Complexity of problems under the reversal model

along with translocations, fissions and fusions could explain the transformation between the two genomes. Aided by a new rearrangement model (discussed in the next subsection), efficient algorithms ($O(n)$) were provided to compute the optimal number of rearrangements in case of multichromosomal genomes [6].

Reversal median: The median problem is the easiest phylogenetic questions posed by genome rearrangements. In case of unichromosomal, the reversal median problem is NP-hard [15]. The problem is still open for the case of multichromosomal reversals.

3.2.2 Double-Cut-and-Join (DCJ)

The positive results for the reversal distance were limited to instances where the number of chromosomes in both genomes were equal. Further, it was necessary that homologous genes are located in the same chromosome for both genomes. In practical, the number of chromosomes and the location of the genes may vary in both genomes. In such a case, when the number of chromosomes in both the genomes varied, explaining the evolutionary scenario solely using reversals was not viable. Other rearrangements events such as translocations, fusions and fissions were required to exactly determine the evolutionary scenario. Moreover, there was no obvious biological intuition behind the use of combinatorial constructs like hurdles and fortresses. In order to simplify matters, the *Double-Cut-and-Join model* was introduced [86].

In a typical genome rearrangement event, the genome is cut at a maximum of two places and the cuts are suitably repaired. A DCJ event mirrors this mechanism. A *Double-Cut-and-Join* operation applied to two adjacencies ab and cd involves replacing the two adjacencies with either 1) ac and bd or 2) ad and bc. Here a, b, c and d are gene extremities. If the genome is cut at less than 2 places, it can still be modeled as a DCJ operation using empty chromosome, which is simply a hypothetical construct of a temporary circular chromosome consisting of two adjacent telomeres [5]. In this manner, rearrangement events

Model	Pairwise distance	Genome median
Unichromosomal	$O(n)$[5]	NP-hard[83]
Multichromosomal, circular/mixed	$O(n)$[5]	NP-hard [83]
Multichromosomal, linear	$O(n)$[86]	Open

Table 3.2: Complexity of problems under the DCJ model

such as reversals, translocations, fusions and fissions can all be modeled as a DCJ event. In Figure 3.3, in each example red arrows indicates locations of cuts while black arrows indicate joins. An empty chromosome is used to model fusion and fission events.

Adjacency graph: The adjacency graph is a reformulation of the breakpoint graph. The graph is bipartite with a set of vertices pertaining to either genomes. Each set of vertices consists of either adjacencies or individual extremities (which are not adjacent to any other extremity in the genome) belonging to the genome. Two vertices are joined by an edge if and only if they share an extremity. An illustration of the graph is shown in Figure 3.4. The *DCJ distance* can then be computed as

$$d = n - c - i/2$$

where c is the number of cycles in the graph and i is the number of paths of odd length. The DCJ distance problem was proved to be solvable in linear time for unichromosomal as well as multichromosomal genomes [5, 86]. However, this is the case in which the genes in both genomes are assumed to be unique. For genomes containing duplicate genes, even if the gene content of both genomes is equal, computing the DCJ distance is NP-hard [70].

DCJ median: Although computing the DCJ distance is possible in linear time, the general median problem under the DCJ model was proved to be NP-hard, even in the simplest case

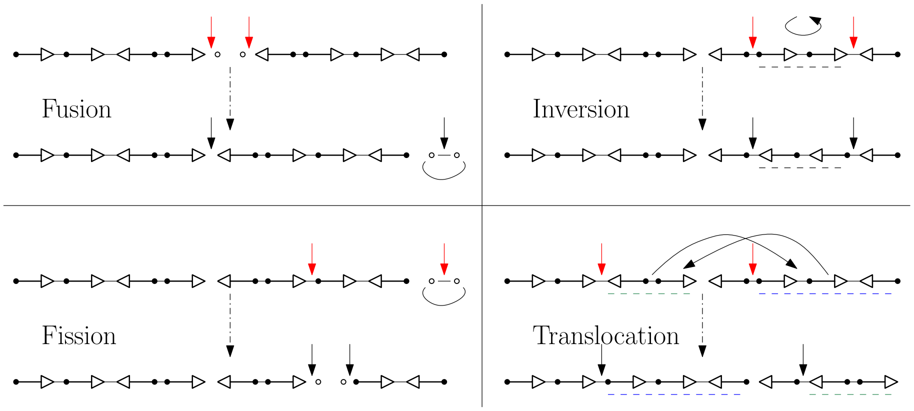


Figure 3.3: A DCJ operation used to model various rearrangement events

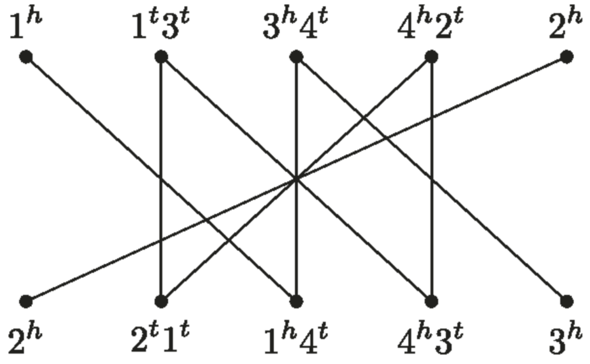


Figure 3.4: An example of an adjacency graph [53]

with $k = 3$. However, the problem is still open in case the genomes are forced to be linear [83].

3.2.3 Breakpoint model

The notion of breakpoint was introduced in some of the earlier studies on genome rearrangements [84, 60] . Given two genomes G_1 and G_2, a *breakpoint* in G_1 is defined as an adjacency that is not seen in G_2. Genome rearrangement events result in the disruption of gene orders, through a series of breaks and repairs. Subsequent studies debated if specific regions in the genome are prone to breakage. Through their analysis of conserved segments - segments in the genome that have not been disrupted, Nadeau and Taylor [60] observed that the breakpoints were randomly distributed throughout the genome. Pevzner and Tesler [64], using the breakpoint graph for the human-mouse distance, reported that rearrangements are more likely to happen within the chromosome. Moreover, it was also noted in [64] that the breakpoints in certain fragile regions in the genome are used repeatedly, which was in contrast with earlier results in [60].

Breakpoint reuse: Pevzner and Tesler defined a *reuse* statistic r based on the number of breakpoints, as a measure to infer the breakpoint prone regions in the genome. Computing the distance d according to the Hannenhalli-Pevzner theory,

$$r = \frac{2d(G_1, G_2)}{b}$$

where b denoted the number of breakpoints in G_1. The value of r ranges between 1 to 2 with the endpoints indicating minimum or maximum reuse [77].

However, this formula assumed that every rearrangement results in 2 cuts. This does not always hold true, since some DCJ operations may lead to one or even no cut in the genome. To address this discrepancy, the reuse statistic was redefined as

$$r' = \frac{c}{b}$$

Model	Pairwise distance	Genome median
Unichromosomal	$O(n)$[83]	NP-hard[63]
Multichromosomal, circular/mixed	$O(n)$[83]	$O(n^4)$[83], [45]
Multichromosomal, linear	$O(n)$[83]	NP-hard [83]

Table 3.3: Complexity of problems under the breakpoint model

where c measures the exact number of cuts in the scenario while b refers to the number of vertices in G_2 that fell on long paths.

Breakpoint distance: Based on the conservation of adjacencies, the breakpoint distance between G_1 and G_2 is given by

$$d_b p(G_1, G_2) = n - a(G_1, G_2) - \frac{e(G_1, G_2)}{2}$$

where n is the number of genes, $a(G_1, G_2)$ is the number of adjacencies common to both genomes whereas $e(G_1, G_2)$ is the number of telomeres common to both genomes. Notice that this distance can easily be translated into the DCJ distance through the adjacency graph. Each common adjacency corresponds to a 2-cycle in the adjacency graph while each common telomere corresponds to a one path. Similar to the DCJ distance, the breakpoint distance problem is also solvable in linear time.

Breakpoint median: The breakpoint median problem was proved to be NP-hard for linear, unichromosomal genomes [63]. However, if the median is allowed to have circular chromosomes, the problem of computing the median can be solved in polynomial time by reformulating the genome median as a maximum weight matching problem [83]. A graph is defined on vertices $V_1 \cup V_2$, where V_1 denotes the set of gene extremities of M while the set V_2 consists of one vertex t_x for each gene extremity x. All edges are weighted with the following weight scheme. An edge between two extremities $x, y \in V_1$ is weighted by the number of genomes G_i containing the adjacency xy. An edge between two extremities $x, y \in V_2$ is weighted 0. An edge xt_x is weighted by half the number of genomes G_i with x as a telomere, while an edge xt_y is weighted 0. Given such a graph, any matching on the graph defines a median genome M. Moreover, a maximum weight matching in this graph defines an optimal median M, thus making the problem tractable with circular genomes.

3.3 Single-Cut-or-Join (SCJ)

Unlike the DCJ model, which can be explain the distance using a series of genome rearrangement events, the breakpoint model is not mechanistic. Feijão and Meidanis proposed a mechanistic version of the breakpoint distance using a set theoretic distance [28]. One of

Model	Pairwise distance	Genome median	Small parsimony
Unichromosomal	$O(n)$[28]	Open	Open
Multichromosomal, circular/mixed	$O(n)$[28]	$O(n)$[28]	$O(n^2 l)$[28]
Multichromosomal, linear	$O(n)$[28]	$O(n)$[28]	Open

Table 3.4: Complexity of problems under the SCJ model. Here, n is the number of genes in an genome and l is the number of leaves in the phylogeny.

the more notable features of the SCJ model is the tractability of the median as well as the Small Parsimony Problem under this model.

SCJ distance: In the SCJ model, each rearrangement is either a cut or a join. For a given pair of genomes G_1 and G_2, represented by their adjacency sets, the SCJ distance between the two is given by:

$$d_{SCJ}(G_1, G_2) = |G_1 - G_2| + |G_2 - G_1|$$

Note that the model requires both genomes to have equal gene content with each gene distinctly identified. Using the above formula, the SCJ distance can be computed in linear time.

SCJ median: Along with the pairwise distance problem, the SCJ median problem was also solved in polynomial for linear, circular or mixed multichromosomal genomes [28]. It was proved that minimizing the total distance $\sum_{i=1}^{k} w_i d_{SCJ}(G_i, M)$ is equivalent to maximizing the score of the median $s(M) = 2\sum_{i=1}^{k} w_i|M \cap G_i| - \sum_{i=1}^{k} w_i|M|$. Consider a median that consists solely of an adjacency $a = xy$. The score $s(a)$ can be computed as:

$$\begin{aligned} s(a) &= 2\sum_{i=1}^{k} w_i|a \cap G_i| - \sum_{i=1}^{k} w_i|a| \\ &= 2\sum_{i \in S_a} w_i - \sum_{i=1}^{k} w_i \qquad\qquad \text{where } S_a = \{i|a \in G_i\} \end{aligned}$$

The score $s(a)$ represents the contribution of an adjacency a towards the score of the median. Clearly, if this contribution is negative, it would be desirable to leave a out of the optimal median. Moreover, if $s(a) > 0$ for some $a = xy$, then for any other extremity $z \neq y$, $s(xz) < 0$ thus avoiding any conflicts with other adjacencies. Thus, the median M defined by the set of adjacencies with $s(a) > 0$ is an optimal median.

To obtain an optimal median solution consisting only of linear chromosomes, it suffices to remove the least weighted edge from an existing circular chromosome.

SCJ Small Parsimony: The SCJ rearrangement model is the only model under which the Small Parsimony Problem is can be solved in polynomial time. In the SCJ model, a genome

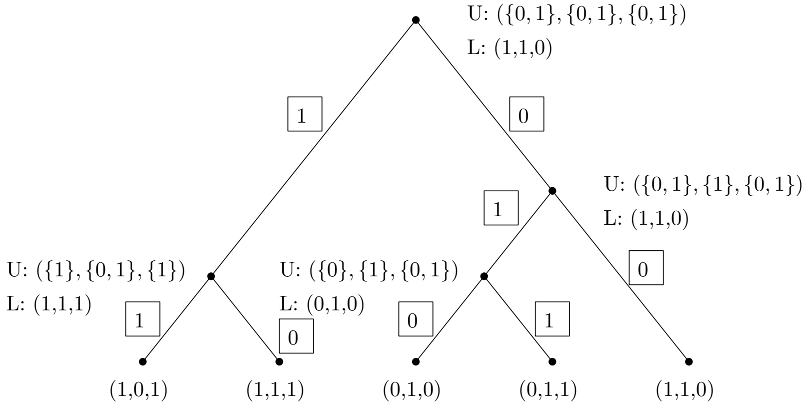


Figure 3.5: Assignment of characters to internal nodes, with the cost of transition along each branch (in boxes)

can be represented as a set of adjacencies. The small parsimony problem can be solved by using the Fitch algorithm individually for each adjacency [30, 28].

In this approach, each adjacency is treated as a binary state 0 or 1 determining its absence or presence in the genome. The algorithm follows the principle of parsimony and thus, tries to avoid the transition of states (from 0 to 1 or vice versa). This directly translates to minimizing the number of cuts and joins along the tree. For each adjacency, the algorithm is carried out in two passes. The first, an upward pass determines the set of all possible states at a node in the tree. The reverse pass then chooses a state from these sets such that the overall number of state transitions is minimized. An illustration of the Fitch algorithm is provided in Figure 3.5. At each internal node, U denotes the sets of possible states for each adjacency, generated during the upward pass while L denotes the assignment of parsimonious states for each adjacency. For a particular ancestral node in the tree, the set of adjacencies with state 1 defines the genome at the node.

3.4 The SCJ-TD-FD model

The SCJ-TD-FD model is an extension of the SCJ model, consisting of two types of evolutionary events: genome rearrangements and duplications. Genome rearrangements are modeled by *Single-Cut-or-Join* (SCJ) operations, that either delete (cut) or create (join) an adjacency in a genome. For duplication events, we consider two types of duplications, both creating an extra copy of a single gene: *Tandem Duplications* (TD) and *Floating Duplications* (FD).

Definition 3.4.1. A *tandem duplication* of an existing gene g is the event in which a new copy of g, say g' is introduced immediately next to the original gene g in the chromosome.

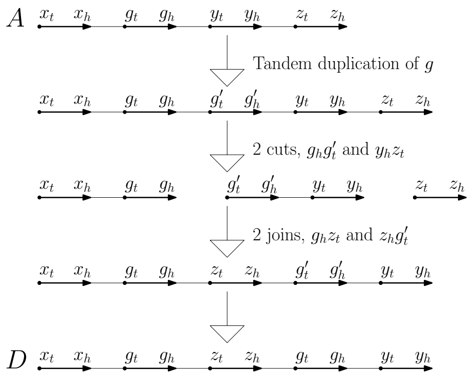

A
x_t x_h g_t g_h y_t y_h z_t z_h
Tandem duplication of g
x_t x_h g_t g_h g'_t g'_h y_t y_h z_t z_h
2 cuts, $g_h g'_t$ and $y_h z_t$
x_t x_h g_t g_h g'_t g'_h y_t y_h z_t z_h
2 joins, $g_h z_t$ and $z_h g'_t$
x_t x_h g_t g_h z_t z_h g'_t g'_h y_t y_h
D
x_t x_h g_t g_h z_t z_h g_t g_h y_t y_h

Figure 3.6: An example of a tandem duplicate g' of of g, used to transform A to D.

A tandem duplication thus results in the addition of an adjacency $g_h g'_t$. If there was an adjacency $g_h x$, it gets replaced by the adjacency $g'_h x$. An example of a tandem duplication is shown in Figure 3.6. In this example, the adjacency $g_h y_t$ has been replaced by $g'_h y_t$ and an adjacency $g_h g'_t$ has been introduced. Note that the number of cut and join operations is dependent on the adjacencies of the gene g in A and D.

Definition 3.4.2. A *floating duplication* of a gene g is the event in which a new copy of g, say g' is introduced as a single-gene circular chromosome.

A floating duplication results in the addition of the adjacency $g'_h g'_t$. An example of a floating duplication is illustrated in Figure 3.7.

The motivation for this type of duplication is that gene insertions and gene deletions have been modeled with artificial circular chromosomes before, greatly simplifying how to deal with such type of operations. For instance, in the Double-Cut-and-Join (DCJ) model, a deletion of a gene can be seen as a DCJ operation that applies two cuts to remove the given gene from a chromosome, followed by two joins to "repair" the broken chromosome and

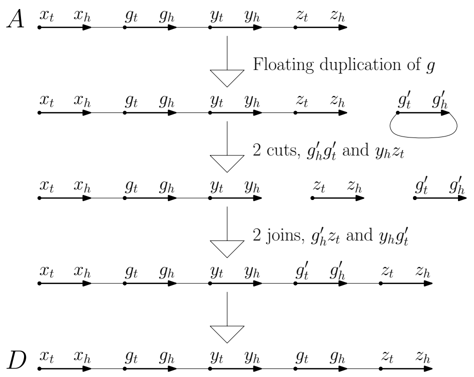

A
x_t x_h g_t g_h y_t y_h z_t z_h
Floating duplication of g
x_t x_h g_t g_h y_t y_h z_t z_h g'_t g'_h
2 cuts, $g'_h g'_t$ and $y_h z_t$
x_t x_h g_t g_h y_t y_h z_t z_h g'_t g'_h
2 joins, $g'_h z_t$ and $y_h g'_t$
x_t x_h g_t g_h y_t y_h g'_t g'_h z_t z_h
D
x_t x_h g_t g_h y_t y_h g_t g_h z_t z_h

Figure 3.7: An example of a floating duplicate g' of of g, used to transform A to D.

to circularize the deleted gene. A gene insertion is the inverse of this operation. This idea was effectively used in the DCJ indel model by Compeau [19]. We discuss the possibility of using a single-gene linear chromosome instead of a circular one at the end of Section 4.3.2.

In this chapter, we were introduced to the important problems investigated in relation to genome rearrangement. It also presented a review of the various frameworks under which genome rearrangement problems have been studied in the past. It discussed the SCJ distance, based on a set theoretic evolutionary model, using which the median and small parsimony problems can be solved in polynomial time. Finally, it discussed a variant of the SCJ model that can handle single-gene duplications. In the subsequent chapters, we will analyze the distance, median and small parsimony problems under this model. Contrary to previously studied models, we will see that some of problems are tractable despite the presence of gene duplication as an evolutionary event.

Chapter 4

A tractable variant of the Single Cut or Join distance with duplicated genes

The previous chapter highlighted various evolutionary models that have been used to study genome rearrangements. We also got acquainted with the SCJ-TD-FD model that uses single gene duplications in addition to single cuts and joins.

The following chapter highlights the key results on the genome distance problem, published in proceedings of the *RECOMB-CG*, 2017 satellite conference [27]. We use the SCJ-TD-FD model to solve the genome distance problem. The results presented in this chapter are a product of joint collaboration with Dr. Pedro Feijão. The SCJ-TD-FD model is an extension of his work in [28].

4.1 Introduction

Genome rearrangement problems aim at finding a plausible evolutionary scenario to explain the evolution of a genome. Under parsimony assumptions, this translates to finding a scenario with the least number or cost of genome rearrangement events. The *pairwise genome rearrangement distance* problem aims at finding a most parsimonious or most likely sequence of genome rearrangements, within a given evolutionary model, that transforms one given genome into another given genome, thus giving a possible evolutionary scenario between the two given genomes. In the absence of duplicated genes, computing the pairwise distance is tractable for most rearrangement models. However, when gene duplication is allowed as an evolutionary event, most rearrangement distance problems become NP-hard. Although there exist polynomial time algorithms that handle large-scale duplications [28, 88], erstwhile approaches have been unable to handle single-gene duplication events. Here, we present an exception to this trend. We prove that the rearrangement distance with duplicated genes can be computed in polynomial time, under the SCJ-TD-FD model.

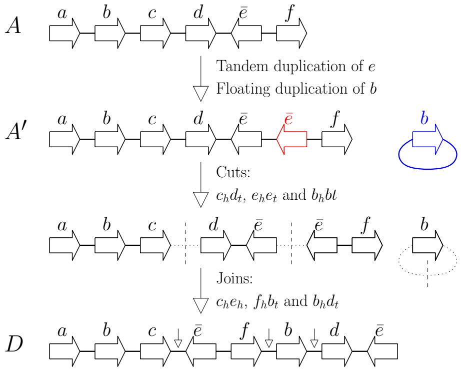


Figure 4.1: The transformation from A to D using cuts, joins and single gene duplications.

Given an ancestor A and a descendant D, we use the SCJ-TD-FD model to propose an evolutionary scenario from a duplication-free ancestor A to descendant D that may contain duplicated genes. This setting is inspired by the development of algorithms to reconstruct ancestral gene orders along a given species phylogeny using reconciled gene trees that provide, for each gene family within the set of considered genomes, one-to-one or one-to-many orthology relations between each ancestral gene and its descendant gene(s), if any. This general framework was introduced by Sankoff and El-Mabrouk in [74] (see also [17]). It was later implemented in the DeCo* family of algorithms [25] to reconstruct ancestral gene orders in a duplication-aware evolutionary model from data including extant gene orders and reconciled gene trees. In this context, the genome A represents an ancestral genome, reconstructed for example with DeCo*, the genome D represents a descendant of A and we are interested in computing a directed distance, from an ancestor to its descendant, where all members of a same gene family present in genome A are considered as distinguishable thanks to the information provided by the reconciled gene tree of this family. In the evolutionary model we consider, rearrangements are either Single Cuts or Single Joins, while duplications can only be single gene duplications, but of two different types, Tandem Duplications (TD) or Floating Duplications (FD) in which a new copy is introduced as a circular chromosome (refer Figure 4.1). It is shown that in this model the distance problem can be simply reduced to deciding, for each gene family with duplicates in D, the length of a tandem array of duplicates to introduce in A and we provide a polynomial time algorithm for this problem.

The remaining chapter is organized as follows: in Section 4.2, we discuss the main problem statement. In Section 4.3 we present our theoretical results, a closed equation for the SCJ distance with duplications and a linear time algorithm to find an optimal scenario. Finally, we provide preliminary experimental results in Section 4.4.

4.2 Problem statement

The pairwise distance problem. We consider the case of directed evolution from a trivial ancestral genome A to a descendant genome D. The evolutionary model excludes gene loss and de-novo gene creation, so we assume that every gene a in A has at least one descendant in D and conversely every gene D has a unique ancestor gene in A. If so, we say that A and D have the same *gene families set*.

The directed SCJ-TD-FD (d-SCJ-TD-FD) distance problem. Let A be a trivial genome and D be a non-trivial genome, such that no gene family is absent from either A or D. Compute the minimum number of SCJ, TD and FD operations needed to transform A into D, denoted by $d_{\mathrm{DSCJ}}(A, D)$.

Note that if D is a trivial genome, the usual SCJ distance, denoted by $d_{\mathrm{SCJ}}(A, D)$ is defined by the symmetric differences of the adjacencies sets of A and D: $d_{\mathrm{SCJ}}(A, D) = |A - D| + |D - A|$ where the first term accounts for the number of cuts and the second term for the number of joins.

4.3 Algorithmic results

In this section, we show that, after a preprocessing step of removing obvious TD and FD in D, the d-SCJ-TD-FD distance can be calculated with the symmetric difference between the adjacency (multi)sets of the input genomes, with an extra factor to account for the gene duplications. We first focus on the preprocessing. Next we describe a linear time algorithm to compute a parsimonious scenario.

4.3.1 The directed SCJ-TD-FD distance

An *observed duplication* in D is defined as an adjacency of the form $g_h g_t$, that defines either a single-gene circular chromosome or a *tandem array* of two (or more) copies of a gene g that occur consecutively and with the same orientation. We denote by t the number of such adjacencies in D and by D' the genome obtained from D by removing first all genes but one from each tandem arrays, and then all single-circular chromosomes for genes from non-trivial families but one if all genes of the family are in such circular chromosomes. D can obviously be obtained from D' by $t(D)$ duplications and the following lemma is immediate:

Lemma 1. $d_{\mathrm{DSCJ}}(A, D) = d_{\mathrm{DSCJ}}(A, D') + t(D)$.

As a consequence, we assume from now on that D has been preprocessed as described above and does not contain any tandem array or any extra copy of a non-trivial family that is in a single-gene circular chromosome. We say that D is *reduced.* Note that single-gene linear chromosomes are not impacted by this preprocessing as, in our setting, if the considered gene is from a non-trivial family, the linear chromosome it forms required at least a cut to be created.

Theorem 1. Given a trivial genome A and a reduced non-trivial genome D such that no gene family is absent from either A or D and where D has $\delta(A,D)$ more genes than A, the d-SCJ-TD-FD distance between A and D is given by

$$d_{\mathrm{DSCJ}}(A,D) = |A-D| + |D-A| + 2\delta(A,D).$$

Proof. First, we show that $d_{\mathrm{DSCJ}}(A,D) \geq |A-D| + |D-A| + 2\delta(A,D)$. To obtain D from A, we need exactly $\delta(A,D)$ gene duplications. Each duplication of a gene g will create the adjacency $g_h g_t$, regardless of the type of the duplication or the timing of the duplication event. Therefore, $\delta(A,D)$ adjacencies of the type $g_h g_t$ will have to be cut, as D is reduced and has no adjacency of this type. In addition, any adjacency in $A-D$ and $D-A$ defines an unavoidable cut or join respectively. Therefore, we can not transform A into D with less than $|A-D| + |D-A| + 2\delta(A,D)$ operations.

Now, we show that $d_{\mathrm{DSCJ}}(A,D) \leq |A-D|+|D-A|+2\delta(A,D)$, by induction on $\delta(A,D)$. For the base case $\delta(A,D) = 0$, the result follows immediately as both genomes are trivial and $d_{\mathrm{DSCJ}}(A,D) = d_{\mathrm{SCJ}}(A,D)$.

We now assume that $\delta(A,D) > 0$, and pick a gene g with one copy in A and more than one copy in D. Depending on how the adjacencies of g are conserved or not in D, we have a few different subcases to consider. However, in each subcase the general strategy remains the same, as follows. We build a genome A_2 from A by applying one duplication (FD or TD) and also relabeling the original copy g as g', creating an adjacency $g_h g_t$ in the case of an FD or $g'_h g_t$ in the case of a TD. Then we build a genome D_2 from D by also relabeling one copy of g to g', thus creating a new trivial gene family and an instance of the d-SCJ-TD-FD problem with exactly $\delta(A,D)-1$ duplicated gene copies. We can apply the induction hypothesis, leading to the inequality

$$d_{\mathrm{DSCJ}}(A_2,D_2) \leq |A_2-D_2| + |D_2-A_2| + 2(\delta(A,D)-1).$$

Also, as D and D_2 are identical but for the relabeling of g, there is a scenario from A to D, going from A to A_2 and then to D, resulting in the upper bound

$$d_{\mathrm{DSCJ}}(A,D) \leq d_{\mathrm{DSCJ}}(A,A_2) + d_{\mathrm{DSCJ}}(A_2,D_2) = 1 + d_{\mathrm{DSCJ}}(A_2,D_2).$$

We will then show that we can build A_2 and D_2 in a way that they satisfy

$$|A-D| + |D-A| = |A_2-D_2| + |D_2-A_2| - 1,$$

where the -1 term is due to the extra $g_h g_t$ adjacency on A_2 created with the duplication. Together with the above inequalities this will lead to

$$d_{\mathrm{DSCJ}}(A, D) \leq 1 + d_{\mathrm{DSCJ}}(A_2, D_2) \leq |A - D| + |D - A| + 2\delta(A, D)$$

and the result follows. To show that we can build A_2 and D_2 that satisfy the above conditions, we will consider three subcases.

Case (i): Assume that g is not a telomere (and so there are two adjacencies involving g in A, say xg_t and $g_h y$) and there is a copy of g in D whose extremities for also adjacencies xg_t and $g_h y$. We say that the context of g is *strongly conserved* between A and D. Note that x and y do not need to belong to trivial gene families and there might be several copies of x, y, g in D that conserve the context of g in A.

In this case, we build A_2 by applying an FD to create an extra copy of g and relabel the original copy of g in A as g'; we also relabel g' an arbitrary copy of g in D that has the same context than g in A, to obtain D_2 (see Fig. 4.2. Comparing the adjacency sets of A and D with A_2 and D_2, we can see that from A to A_2 two adjacencies where renamed from xg_t and $g_h y$ to xg'_t and $g'_h y$, and exactly the same change happened from D to D_2. Also, the adjacency $g_h g_t$ was added in A_2. As a result, $A_2 = A - \{xg_t, g_h y\} + \{xg'_t, g'_h y, g_h g_t\}$. Similarly, $D_2 = D - \{xg_t, g_h y\} + \{xg'_t, g'_h y\}$. Therefore, we have that $|A - D| + |D - A| = |A_2 - D_2| + |D_2 - A_2| - 1$. Note that this relabeling only works if we introduce a an extra copy of g in A with an FD here; if instead we introduce it with a TD, it would not be possible to get adjacencies xg'_t and $g'_h y$ in D_2, as the copy of g involved in both adjacencies would be different.

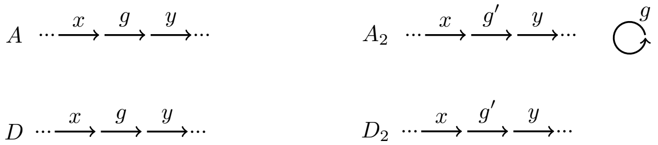


Figure 4.2: The context of g is strongly conserved between A and D (Case (i)).

Case (ii): Assume that g is not a telomere in A, its context is not strongly conserved between A and D, but both adjacencies involving g, xg_t and $g_h y$, are present in D *on different copies* of g. We say that the context of g is *weakly conserved* between A and D. Again x and y need not to be trivial gene families and there might be several occurrences of adjacencies xg_t and $g_h y$ in D.

In this case, we build A_2 by applying a TD on g, relabeling the gene g that has the adjacency xg_t as a new gene g' in both A_2 and D_2, as shown on Fig. 4.3. Comparing the adjacency sets of A and A_2, we notice that the adjacency xg_t changes to xg'_t, and $g_h g_t$ is added. Thus, $A_2 = A - \{xg_t\} + \{xg'_t, g_h g_t\}$. From D to D_2 we also have the same change, and possibly one more, depending if g'_h is a telomere in D (no change) or if g'_h has an adjacency g'_hw. In the former case, $D_2 = D - \{xg_t\} + \{xg'_t\}$. Otherwise, $D_2 = D - \{xg_t, g_hw\} + \{xg'_t, g'_hw\}$. In either case, the possible adjacency g'_hw does not exist in A or A_2. Consequently, the equality $|A - D| + |D - A| = |A_2 - D_2| + |D_2 - A_2| - 1$ holds.

Note also that in this case an FD would not be optimal, because it would force the labeling of the adjacency g_hy to g'_hy, and since the adjacency g_hy on D cannot have the label g'_hy, this would force an extra pair of SCJ operations.

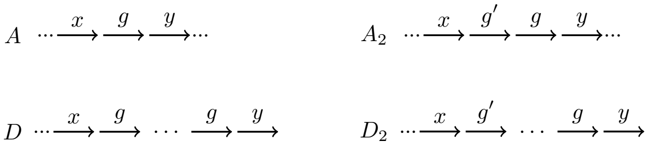


Figure 4.3: The context of g is weakly conserved between A and D (Case (ii)).

Case (iii) : We assume now that the context of g in A is neither strongly nor weakly conserved, and so at most one adjacency of g in A is also present in D.

This case is similar to case (i), if we assume that either xg_t or g_hy, are present in D, or neither. In the same way, we apply an FD on g, labeling the original copy as g', as shown in Fig. 4.4. On D, we pick a gene g that has an adjacency xg_t or g_hy if any or, if no adjacency involving g is conserved in D, we pick an arbitrary g, and relabel it as g'.

Now, any adjacencies that were conserved between A and D will remain conserved between A_2 and D_2, and no new conserved adjacencies have been created. Since, as before, A_2 has a new g_hg_t adjacency, the equality $|A - D| + |D - A| = |A_2 - D_2| + |D_2 - A_2| - 1$ holds.

These three cases cover all possible configurations for g, so the theorem is proved.

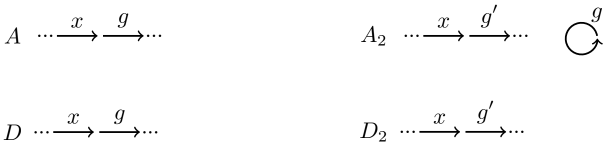


Figure 4.4: At most one adjacency of g is conserved (Case (iii)).

□

4.3.2 Computing a parsimonious scenario

It follows from Lemma 1 and Theorem 1 that computing the d-SCJ-TD-FD distance can be done in linear time in the size of the considered genomes A and D. Moreover, they define a simple algorithm that computes a parsimonious scenario in terms of duplications, cuts and joins from A to D, described in Algorithm 1 below.

Algorithm 1 Compute an SCJ-TD-FD parsimonious scenario between a trivial genome A and a genome D

```
Reduce $D$ into a reduced genome $D'$
Let $A' = A$ and $i = 1$
while $(A', D')$ has a non trivial gene family do
    Let $g$ be an arbitrary gene from a non trivial family in $A'$; relabel $g$ by $g^i$.
    if the context of $g$ is strongly conserved then
        relabel the corresponding copy of $g$ in $D'$ by $g^i$
        add to $A'$ a single-gene circular chromosome $g$.
    else if the context of $g$ is weakly conserved then
        create an extra copy of $g^i$ with a TD
        relabel a copy of $g$ involved in adjacency $xg_t$ in $D'$ by $g^i$.
    else if one adjacency of $g$ is conserved in $D'$ then
        relabel the corresponding copy of $g$ in $D'$ by $g^i$
        add to $A'$ a single-gene circular chromosome $g^i$.
    else
        relabel an arbitrary copy of $g$ in $D'$ by $g^i$
        add to $A'$ a single-gene circular chromosome $g^i$.
    $i = i + 1$
Compute an SCJ scenario from $A'$ to $D'$.
Recreate in $D'$, the tandem arrays and single-gene circular chromosomes removed when
reducing $D$ into $D'$.
```

Theorem 2. Given a trivial genome A with n_A genes and a possibly non-trivial genome D on the same set of gene families and with n_D genes, Algorithm 1 computes a parsimonious SCJ-TD-FD scenario that transforms A into D and can be implemented to run in time and space $O(n_D)$.

The correctness of the algorithm follows immediately from the fact that it implements exactly the rules described to compute the SCJ-TD-FD distance (Lemma 1 and Theorem 1). The linear time and space complexity follows from the fact that these rules are purely local and ask only to check for the conservation of adjacencies in both considered genomes.

Every iteration of the while loop in Algorithm 1 takes place only if there is a non-trivial gene family left in D'. The maximum number of iterations is the number of duplicates genes, $\delta(A, D) = n_D - n_A$ which is $O(n_D)$ when $n_D \geq n_A$. In each iteration, we check if the context of the chosen gene g is strongly conserved, weakly conserved or not conserved. This involves trying to match the adjacencies of g in A with those in the adjacency set of D' that involve a copy of g. This can be done in constant time, with a linear time preprocessing of the data. Hence, the worst-case time complexity is $O(n_D)$.

Remark 1. We have discussed in Section 4.2 the rationale to create duplicate genes with a FD creating a circular single-gene chromosome. However if the evolutionary model of the FD event created a linear single-gene chromosome, this would introduce a dissymetry between TD and FD (namely no adjacency is created with an FD), while in our model each created copy induces a cost of two due to the necessary break of the created adjacency required in the process of obtaining the reduced genome D'. We conjecture that the use of linear chromosomes would affect the choice of duplication event (FD or TD) only when the context is not conserved, which would result in a more complicated distance formula.

4.3.3 Relation to the exemplar distance framework.

Sankoff [72] introduced the notion of Exemplar Breakpoint (EBP) distance, where an *exemplar* of a non-trivial genome is obtained by keeping exactly one gene copy from each gene family. In the directed evolution setting, an exemplar can be assumed to be the original gene from A having evolved into a gene now present in D, all other copies having been created by duplications. So the EBP distance problem aims to find an exemplar for each group of duplicates in D such that the trivial genome that results from deleting all non-exemplar copies minimizes the breakpoint distance to A. The notion of exemplar distance can naturally be used in conjunction with the SCJ distance instead of the BP distance, a problem we denote the ESCJ distance. The EBP distance problem has been shown to be NP-hard even in the directed evolution case where every duplicated gene has exactly two copies in D [13], and it is immediate to extend this hardness result to the directed ESCJ distance problem.

Intuitively, the directed ESCJ distance and the d-SCJ-FD-TD distance problems seem very similar. For example in the case of duplicated genes having exactly two copies in D, the later aims at deciding which copy in D is exemplar (*i.e.* evolved from the original copy in A) and then, for the second copy, if it originates from a TD or a FD, thus resulting in a matching between two genomes with two copies of each duplicate, opposed to the ESCJ setting where the matching is between genomes with one copy of each gene.

It is interesting to notice that both problems, although similar, have opposed properties in terms of tractability, and that the d-SCJ-TD-FD distance problem is tractable despite considering a larger solution space. Moreover, one can ask if there is a strong correlation between the distance obtained in both settings. It is not difficult to find examples that show that both distances can be quite different: the ESCJ distance between $A = [a, b, c, d]$ and $D = [a, c, b, d, c, a, d, b]$ is 0, whereas the SCJ-TD-FD distance between the same two genomes is 18 (4 duplications, 7 cuts, 7 joins). However, although the difference between both distances can be arbitrarily large, tight bounds can be derived.

Lemma 2. Let A be a trivial genome, D be an arbitrary genome on the same set of gene families than A, $d_{\text{DSCJ}}(A, D)$ and $d_{\text{ESCJ}}(A, D)$ denote the d-SCJ-FD-TD and the ESCJ distances, respectively. Let k be the difference between the number of genes in D and the number of genes in A. The following bounds

$$k \leq d_{\text{DSCJ}}(A, D) - d_{\text{ESCJ}}(A, D) \leq 5k$$

are tight.

Proof. First, we obtain the genome A' by applying $d_{\text{ESCJ}}(A, D)$ SCJ operations on A in a way that the duplicated genes in D are in the same order as the corresponding matched genes in A', as given by an optimal exemplar matching. In the SCJ-FD-TD model, we need to apply at least k duplications on A' to obtain D, so $d_{\text{DSCJ}}(A, D) \geq d_{\text{ESCJ}}(A, D) + k$ (otherwise $d_{\text{ESCJ}}(A, D)$ would not be optimal). To show that this bound is tight, we can see that the trivial case of no duplications holds. But, whenever A and D differ by only k tandem duplications, the bound is tight, since in this case $d_{\text{ESCJ}}(A, D) = 0$ and $d_{\text{DSCJ}}(A, D) = k$.

Now, from A', we can apply k free duplications, followed by k cuts on these duplications. Also, perform at most k cuts, between any two genes on A' if both have more than one copy on D. Since A' was ordered in relation to its corresponding copies on D, it is possible to join the "fragments" of A' that were created with the previous $2k$ cuts with $2k$ joins in a way to transform A' in D, and therefore we built a d-SCJ-FD-TD scenario from A to D with $d_{\text{ESCJ}}(A, D) + 5k$ operations.

Any pair of circular genomes $A = (1, 2, \ldots, n)$ and $D = (1, n, 2, 1, \ldots, i, i-1, \ldots, n, n-1)$ satisfies the tight bound. □

4.4 Experimental results

We ran experiments on simulated instances with the aim to evaluate the ability of the d-SCJ-TD-FD distance to correlate with the true number of syntenic events. We followed a simulation protocol inspired from [80]. The code itself was programmed in Python and is available via github[1]. We first describe the simulation protocol, followed by the results we obtained.

We started from a genome A composed of a single linear chromosome containing 1000 single-copy genes. Then, we transformed A genome into a genome D through a sequence of random segmental duplications and inversions. We fixed the number N of evolutionary events (from 50 to 500 by steps of 50) and the probability P that a given event is a segmental duplication (from 0 to 0.5 by steps of 0.1). A segmental duplication is defined by three parameters: the position of the first gene of the duplicated segment, the length of the duplicated segment, and the breakpoint where the duplicated segment is transposed into; we considered two models of segmental duplications, one with fixed segment length L (with L taking values in $\{1, 2, 5\}$) and one where for each segment, L is picked randomly (under the uniform distribution) in $\{1, 2, 5, 10\}$. Inversion breakpoints were chosen randomly, again under the uniform distribution. For each array of parameters, we ran 50 replicates.

For each instance, we compared two quantities to the true number of cuts and joins in the scenario transforming A into D, which is roughly four times the number of inversions plus

[1] https://github.com/acme92/SCJTDFD

three times the number of segmental duplications: first we compared the full SCJ-TD-FD distance, defined as stated in Theorem 1 and the number of cuts and joins ($|A-D|+|D-A|$). Figure 4.5 illustrates the results we obtained.

We can make several observations from these results. The first one is a general trend that both measured quantities (the number of cuts and joins and the full SCJ-TD-FD distances) scale linearly with the true number of cuts and joins. The second observation is that, as expected, the slope and a y-intercept of the graphs depend from both the frequency of duplications and the length of the duplicated segments. This leaves open the question of using the SCJ-TD-FD distance as an estimator of the number of cuts and joins in an evolutionary model where the probability of duplication compared to rearrangements (that can be estimated for example from reconciled gene trees and adjacency forests [25]) is given and the length of duplicated segments is expected to follow a well defined distribution.

4.5 Conclusion

In this work, we introduced a simple variant of the SCJ model that accounts for duplications, and showed that, in this model, computing a directed parsimonious genomic distance from a trivial ancestral genome to a non-trivial descendant genome can be done in linear time. The tractability stems mostly from the combination of assuming that one genome is trivial and of a simplified model of duplication where gene duplication are single-gene events. However we believe it is interesting to push the tractability boundaries of the SCJ models toward augmented models of evolution (here accounting for duplications). Moreover, our work is motivated by the increasing performance of ancestral gene order reconstruction methods, that can now account for complex gene histories using reconciled gene trees and motivate the directed distance approach, and provides an additional positive result along the line of the research program outlined in [74]. For example, our algorithm will allow to extend the small parsimony algorithm PhySca introduced in [46] to a duplication-aware framework by allowing to score exactly and quickly an ancestral gene order configuration within a species phylogeny.

There are several avenues to extend the results we presented in this chapter. It will likely be easy to modify our algorithm to work in an extended the evolution model to integrate the loss of gene families and de-novo creation of genes. Our main result provides a simple algorithm that computes a parsimonious scenario, however it is likely one among a large number of parsimonious scenarios, and it is open to see if the results of [54] about counting and sampling SCJ parsimonious scenarios can be extended to our model. An important open question toward a more realistic model of evolution concerns the possibility to include larger scale duplications as unit-cost events. The case of a single whole genome duplication and of whole-chromosome duplications have been shown to be tractable [28, 88], but to the best of our knowledge there is no known result including segmental duplications in

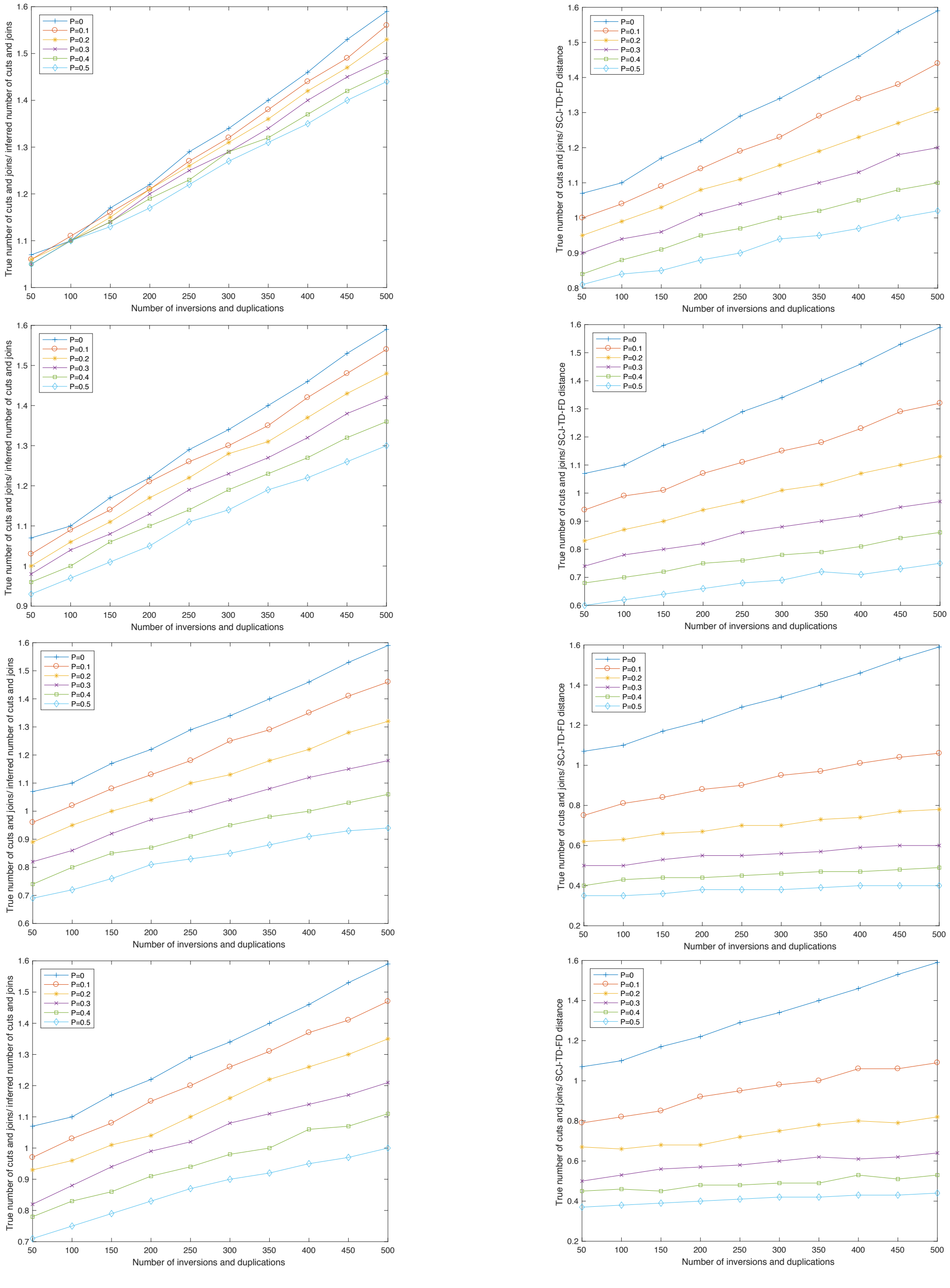


Figure 4.5: Experimental results, for four duplications parameters – single-gene segmental duplication (top row), two-genes segmental duplication (second row), five-genes segmental duplications (third row), variable length segmental duplications (bottom row) – and two measured quantities – inferred cuts and joins (left column) and SCJ-TD-FD distance (right column).

which a contiguous segment of genes is duplicated either in tandem or appearing as a single chromosome. It also remains to be seen if the directed SCJ-TD-FD distance can be used toward the computation of an estimated distance in a more realistic evolutionary distance, similarly to the use of the breakpoint distance to estimate the true DCJ distance [11]; our experimental results suggest this is a promising avenue, although it might be difficult to obtain analytical results in models mixing rearrangements and duplications. Finally, the question of the tractability of the small parsimony problem in our model is also, to the best of our knowledge, still open. It is known to be tractable in the pure SCJ model (with no duplications) due to the independence of adjacencies; this assumption does not hold anymore here and the small parsimony problem is thus likely more difficult in our model.

Chapter 5

Median problems with single gene duplications

The previous chapter discussed the problem of finding the distance between two genomes, using the SCJ-TD-FD model. It is proved that computing the SCJ-TD-FD distance and even an evolutionary scenario is possible in linear time.

The following chapter presents some interesting results on the median problem, accepted for the *RECOMB-CG*, 2018 satellite conference [51]. We discuss two genome median problems under the SCJ-TD-FD model. We see how the introduction of an ancestral genome in one of the problems leads to contrasting hardness results. Under the SCJ-TD-FD distance, the directed (unrooted) median problem is found to be tractable whereas the rooted median problem is NP-hard. The analysis of the tractability results for the unrooted and rooted versions of the SCJ-TD-FD median problem is credited to Dr. Pedro Feijão and Dr. Manuel Lafond, respectively.

5.1 Introduction

The reconstruction of ancestral genomes for a given species phylogeny is an important problem in computational biology [61, 55]. One of the important problems, studied in the context of genome rearrangements, is the Small Parsimony Problem, which aims to reconstructing ancestral gene orders of the given species tree while minimizing the overall genome rearrangement distance along the branches of the tree. The simplest instance of this problem is the median problem, where the given phylogeny contains a single ancestral node whose gene order is to be reconstructed.

As seen in chapter 3, the median problem is NP-hard for most evolutionary models. The exception to this rule is the Single-Cut-or-Join (SCJ) rearrangement model under which, both the SCJ median problem and the SCJ SPP are tractable [28]. In the presence of duplications as possible evolutionary events, even the distance problem is NP-hard for most rearrangement models. The only exception is an evolutionary model involving Single-Cut-or-Join events with whole genome duplications [88].

Here, we focus on two versions of the median problem, namely the directed median problem and the rooted median problem. We prove that, under the SCJ-TD-FD model, both versions of the median problem lead to different tractability results. In the first variant, we consider k descendant genomes $D_1, ..., D_k$. The median M minimizing the sum of the directed SCJ-TD-FD distances from M to each D_i can be computed in polynomial time. Interestingly, the other variant in which an ancestral genome A is provided in addition to given descendant genomes, is found to be intractable. In this variant, it is required to find M that minimizes the sum of the directed SCJ-TD-FD distances from M to each D_i and from A to M. In both cases, we are provided with the gene content of M and the orthology relations along each branch. The rest of the chapter is outlined as follows. In section 4.3, we prove that the median problem under the SCJ-TD-FD distance is tractable. In Section 5.4, we prove that the rooted median problem is NP-hard even when $k = 2$. In Section 5.5, we describe a simple Integer Linear Program (ILP) for this problem, based on a reduction to a colored MWM problem. We provide in Section 5.6 experimental results on simulated data.

5.2 Problem Statements

Recall that the **directed SCJ-TD-FD (d-SCJ-TD-FD) distance problem**, introduced in chapter 4 asks to compute the minimum number of SCJ, TD and FD operations needed to transform A into D, denoted by $d_{\text{DSCJ}}(A, D)$. The problem has been shown to be tractable and the distance can be computed using a simple set-theoretical formula, extending naturally the distance formula for the SCJ with no duplication model.

The directed median problem is the natural extension of the pairwise directed distance problem towards the Small Parsimony Problem.

The directed SCJ-TD-FD (d-SCJ-TD-FD) median problem. Let $k \geq 2$ and $D_1, \dots, D_k$ (possibly) non-trivial genomes, such that no gene family is absent from any D_i. Compute a trivial genome A on the same set of gene families as the non-trivial genomes, that minimizes $\sum_{i=1}^{k} d_{\text{DSCJ}}(A, D_i)$.

We also introduce the a variation of this problem as follows:

The rooted SCJ-TD-FD (r-SCJ-TD-FD) median problem. We are given $k + 1 \geq 3$ genomes, A, $D_1, \dots, D_k$ such that A is a trivial genome, ancestor to the D_i's. The goal of the rooted median problem is to find a genome M which is a descendant of A and an ancestor of $D_1, \dots, D_k$, minimizing the sum of its distance to A and to the $D_i' s$. Following the approach introduced in the previous chapter, we assume we are given the *gene content* Γ of M and the *orthology relations* between A and M, as well as between M and the $D_i' s$. This implies that every gene of M (resp. $D_1, \dots, D_k$) has a unique ancestor in A (resp. in M), so M is a trivial genome compared to the $D_i' s$ but might not be compared to A (see

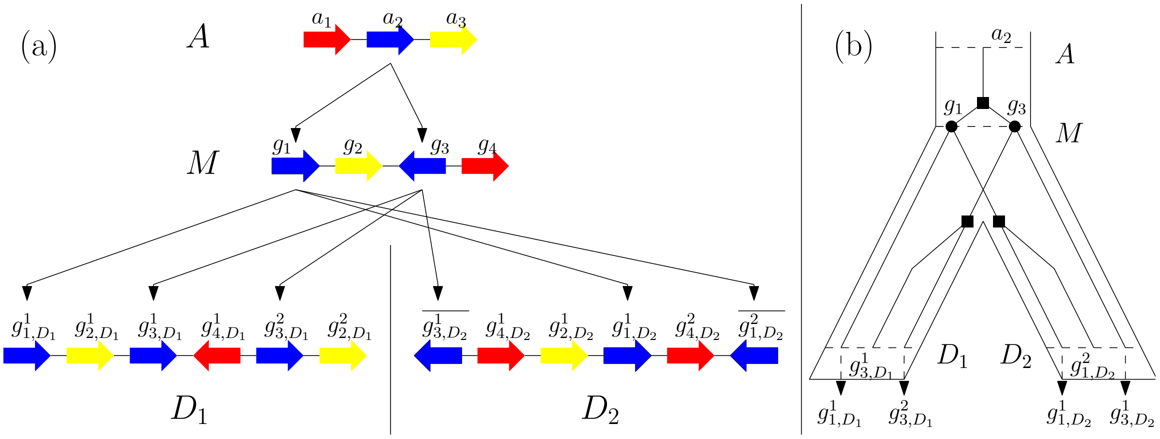

(a)
A
a_1 a_2 a_3
M
g_1 g_2 g_3 g_4
g^1_{1,D_1} g^1_{2,D_1} g^1_{3,D_1} g^1_{4,D_1} g^2_{3,D_1} g^2_{2,D_1}
$\overline{g^1_{3,D_2}}$ g^1_{4,D_2} g^1_{2,D_2} g^1_{1,D_2} g^2_{4,D_2} $\overline{g^2_{1,D_2}}$
D_1
D_2
(b)
a_2
A
g_1 g_3
M
D_1 D_2
g^1_{3,D_1}
g^2_{1,D_2}
g^1_{1,D_1} g^2_{3,D_1} g^1_{1,D_2} g^1_{3,D_2}

Figure 5.1: In part (a), each color represents a gene family from A. Notice that each gene in D_1 and D_2 can be traced to a unique gene in M whereas a gene from A might have multiple daughters in M. Part (b) displays the gene tree of the gene family in blue (indicated by arrows in part (a)). Since the gene a_2 undergoes duplication (dark squares) to form g_1 and g_3 in M, M is not trivial w.r.t A.

Fig. 5.1 for an illustration). To formally handle this difference, we assume that all copies of a gene g of A in M (i.e. the genes of M whose ancestor in A is gene g) are distinguishable (e.g. labeled, say $g_1, \ldots, g_k$) and, for a given gene g_i of M, we denote its ancestor in A by $a(g_i)$. Then for a given genome M on Γ, we denote by M_a the genome where every gene g is relabeled by $a(g)$. The goal of the rooted median problem is to find a genome M that minimizes the following function:

$$d_{\text{DSCJ}}(A, M_a) + \sum_{i=1}^{k} d_{\text{DSCJ}}(M, D_i). \tag{5.1}$$

5.3 The Directed Median Problem

Let us remind that under the SCJ-TD-FD evolutionary model, the *directed median problem* asks, given k non-trivial genomes $D_1, \ldots, D_k$, $k \geq 2$, with the same gene families, to find a trivial common ancestor A, such that $\sum_{i=1}^{k} d_{\text{DSCJ}}(A, D_i)$ is minimized.

We first assume that the genomes $D_1, \ldots, D_k$ are reduced. We define the *score* $s(A)$ of a genome A as

$$s(A) = \sum_{i=1}^{k} d_{\text{DSCJ}}(A, D_i) = \sum_{i=1}^{k} (|A - D_i| + |D_i - A| + 2n_{d_i})$$

where n_{d_i} is the number of extra gene copies in D_i compared to A, for $i = 1, \ldots, k$. Using the fact that $|A - D| + |D - A| = |A| + |D| - 2|A \cap D|$ we derive

$$s(A) = N_d - \left(2 \sum_{i=1}^{k} |A \cap D_i| - k|A| \right)$$

where $N_d = \sum_{i=1}^{k} (2n_{d_i} + |D_i|)$, and does not depend from A. Therefore, minimizing $s(A)$ is equivalent to maximizing $2 \sum_{i=1}^{k} |A \cap D_i| - k|A|$.

For a given adjacency a, let $\delta_i(a)$ be 1 if $a \in D_i$, and 0 otherwise. The score of a genome with a single adjacency a is $s(\{a\}) = N_d - \left(2 \sum_{i=1}^{k} \delta_i(a) - k \right)$. This motivates the following approach, similar to the breakpoint median algorithm of [83]. Build a graph G where the vertices are defined as the extremities (head and tail) of a unique copy for each gene family in the considered genomes D_i (so a gene family a induces two vertices a_h and a_t), and weighted edges are defined as follows: for any edge $e = (x, y)$ such that x and y form an adjacency in at least one of the genomes D_i, the weight of e is $w(e) = 2 \sum_{i=1}^{k} \delta_i(e) - k$. Any matching M on G defines a trivial genome A_M, having the adjacencies corresponding to the edges in the matching M. Also, if $W(M)$ denotes the weight of the matching M, that is the sum of the weights of the edges in M we have that

$$\begin{aligned} s(A_M) &= N_d - \left(2 \sum_{i=1}^{k} |A_M \cap D_i| - k|A_M| \right) \\ &= N_d - \sum_{e \in M} \left(2 \sum_{i=1}^{k} \delta_i(e) - k \right) \\ &= N_d - W(M) \end{aligned}$$

Therefore, solving a maximum weight matching problem on G solves the directed median problem. To handle the case when some D_i is not reduced, we can rely on Lemma 1 that implies that the genomes can be reduced first without impacting the optimality of a trivial genome obtained by a maximum weight matching. Combined with the tractability of computing a maximum weight matching [65], this proves our last theorem.

Theorem 3. Let $k \geq 2$ and $D_1, \ldots, D_k$ be k genomes on the same set of n gene families, having respectively $n_1, \ldots, n_k$ adjacencies. The directed SCJ-TD-FD median problem for these genomes can be solved in time and space $O(n(n_1 + \cdots + n_k) \log(n_1 + \cdots + n_k))$.

Remark 2. In the case of the median of two genomes D_1 and D_2, note that the only edges with strictly positive weight in the graph are defined by adjacencies that appear in both D_1 and D_2, while edges appearing just once have weight 0. So a median genome can be defined as a maximum matching over the unweighted graph defined only by adjacencies that appear

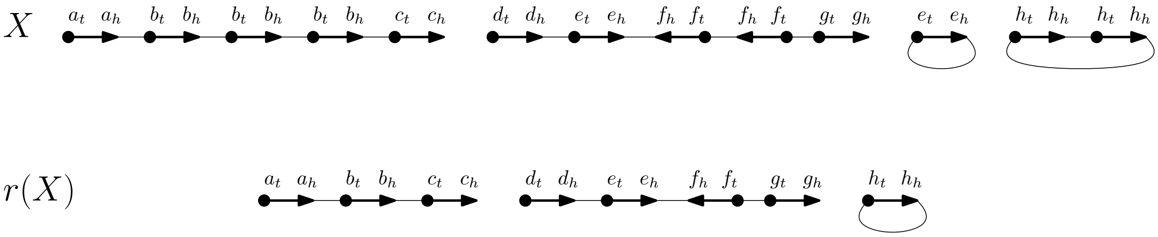

X
a_t a_h b_t b_h b_t b_h b_t b_h c_t c_h d_t d_h e_t e_h f_h f_t f_h f_t g_t g_h e_t e_h h_t h_h h_t h_h
r(X)
a_t a_h b_t b_h c_t c_h d_t d_h e_t e_h f_h f_t g_t g_h h_t h_h

Figure 5.2: An example of the reduced genome $r(X)$, of the genome X. Note that an instance of $h_h h_t$ is retained so that $r(X)$ contains at least one representative of gene family h. All observed duplications are removed in $r(X)$. Here, $t(X) = |X - r(X)| = 5$.

in both genomes, and given such a median, it can be augmented by any subset of edges appearing just once that do not re-use a gene extremity already used in the matching.

Remark 3. It is interesting to observe that if we assume there is no duplication from A to M, i.e. both have the same gene content. Consequently, the MWM algorithm discussed above for the directed median problem applies to the rooted median problem and the problem is thus tractable. So the difficulty in solving the rooted median problem is to account for duplications from A to M.

Modifying the pairwise distance formula. Given a gene $g \in \Gamma$, we call a *g-tandem array* a sequence of consecutive adjacencies $g_h g_t$; if this sequence forms a circular chromosome, it is called a *g-chromosome*. Given a genome X, we call an adjacency $g_h g_t$ an *observed duplication* if g has more than one copy in X. Observed duplications are part of a g-tandem array or a g-chromosome. Let $r(X)$ be the genome obtained from X by successively deleting an observed adjacency from X, chosen arbitrarily, until there remains no observed adjacency. Note that this corresponds to deleting every $g_h g_t$ adjacency, except that we keep one in the special case that all copies of g are organized in g-chromosomes, as shown in Fig. 5.2. We call $r(X)$ the *reduced* genome of X. We define $t(X) = |X - r(X)|$, the number of adjacencies to delete to transform X into $r(X)$. Formally, the multi-set difference $X - Y$ between two multi-sets X and Y of adjacencies is the multi-set obtained as follows: it contains k copies of a given adjacency if and only if X contains exactly k more occurrences of this adjacency than Y (with $k = 0$ being possible).

The SCJ-TD-FD distance between an ancestral genome A and a descendant genome D is given by (from chapter 4:

$$d_{\text{DSCJ}}(A, D) = |A - r(D)| + |r(D) - A| + 2\delta(A, r(D)) + t(D) \tag{5.2}$$

where $\delta(A, r(D))$ is the difference between the number of genes of $r(D)$ and the number of genes of A (i.e. the number of duplications from A to $r(D)$). We introduce a slightly different formulation of d_{DSCJ} that will be useful in our hardness proof:

$$d_{\mathrm{DSCJ}}(A, D) = |A - r(D)| + |r(D) - A| + 2\delta(A, D) - t(D) \tag{5.3}$$

Proof. The original pairwise distance formula (eq. (5.2)) is

$$d_{\mathrm{DSCJ}}(A, D) = |A - r(D)| + |r(D) - A| + 2\delta(A, r(D)) + t(D)$$

and we want to prove it is equivalent to

$$d_{\mathrm{DSCJ}}(A, D) = |A - r(D)| + |r(D) - A| + 2\delta(A, D) - t(D).$$

Notice that the $2\delta(A, r(D))$ term from the original formula was switched for the $2\delta(A, D)$ term. Consider the difference in the number of genes from D to $r(D)$. Each time we remove a $g_h g_t$ observed duplication from D while reducing it, it corresponds to removing a copy of g from D. Thus D has $t(D)$ more genes than $r(D)$, so that $2\delta(A, D) = 2\delta(A, r(D)) + 2t(D)$. This implies $2\delta(A, D) - t(D) = 2\delta(A, r(D)) + t(D)$. □

Remark 4. For $d_{\mathrm{DSCJ}}(M, D_i)$, the value of $t(D_i)$ does not depend on our choice of M, for $i = 1, \ldots, k$. We will therefore assume that the $D_i's$ are reduced (hence we may refer to $r(D_i)$ as simply D_i instead). However $t(M_a)$ has an impact on $d_{\mathrm{DSCJ}}(A, M_a)$, and so we will not assume that M is reduced.

5.4 The rooted median problem is NP-hard

We show that finding the optimal gene order for M is NP-hard even for $k = 2$, by reduction from the 2P2N-3SAT problem [8][1].

In 2P2N-3SAT, we are given n variables $x_1, \ldots, x_n$ and m clauses $C_1, \ldots, C_m$, each containing exactly 3 literals. Each x_i variable appears as a positive literal in exactly 2 clauses, and as a negative literal in exactly 2 clauses. Note that since each variable occurs in exactly 4 clauses and each clause has 3 literals, $m = 4n/3$. An example of a 2P2N-3SAT instance is shown in Figure 5.3.

We now describe how we transform the x_i variables and C_j clauses into an instance of the rooted median. The genes of M are

$$\Gamma = \{g_1^+, \gamma_1^+, g_1^-, \gamma_1^-, \ldots, g_n^+, \gamma_n^+, g_n^-, \gamma_n^-, c_1, \ldots, c_m, \alpha_1, \ldots, \alpha_{2n-m}\}$$

[1] This problem is sometimes called the (3,B2)-SAT problem, where B2 indicates that the literals are *balanced* with two occurrences each.

The genes $g_i^+, \gamma_i^+, g_i^-, \gamma_i^-$ correspond to the x_i variable, and c_j to the clause C_j. The purpose of the $2n - m = 2n/3$ special α_i genes will become apparent later.

To simplify matters, every adjacency in our reduction is between the tails of two genes. Hence, the heads of each gene of A, D_1 and D_2 are telomeres (linear chromosomes extremities), so that all chromosomes are linear and have at most 2 genes. From now, we will omit the t subscript from the extremities for these adjacencies, with the understanding that every adjacency is between tails; for instance, we may write $g_i^+\gamma_i^+$ for the adjacency $g_{i,t}^+\gamma_{i,t}^+$.

We can now describe A, D_1 and D_2. The genes of A are $g_1', \gamma_1', , ..., g_n', \gamma_n', c_1', ..., c_m'$, $\alpha_1', ..., \alpha_{2n-m}'$. The genes g_i^+ and g_i^- (resp. γ_i^+ and γ_i^-) are duplicates of g_i' (resp. γ_i'), and there are no other duplications in M compared to A. Formally, for each $i \in [n]$, put $a(g_i^+) = a(g_i^-) = g_i'$, $a(\gamma_i^+) = a(\gamma_i^-) = \gamma_i'$ and for each $j \in [m]$, put $a(c_j) = c_j'$. Finally, for each $i \in [2n - m]$, put $a(\alpha_i) = \alpha_i'$. The adjacencies of A are $\{g_i'\gamma_i' : i \in [n]\}$.

The genomes D_1 and D_2 are identical, i.e. they contain the same set of genes and of adjacencies. We simply describe the set of adjacencies of D_1 and D_2 with the understanding that if an extremity, say x, appears in two adjacencies xy and xz, then the two x are the tails of two distinct copies of the same gene on two distinct chromosomes.

The adjacencies of D_1 and D_2 are described as follows.

- For each $i \in [n]$, add to D_1 and D_2 the adjacencies $g_i^+\gamma_i^+$ and $g_i^-\gamma_i^-$.

- For each $i \in [n]$, let C_{j_1}, C_{j_2} be the two clauses in which x_i occurs positively and let C_{k_1}, C_{k_2} be the two clauses in which x_i occurs negatively. Add to D_1 and D_2 the adjacencies $g_i^+c_{j_1}$ and $\gamma_i^+c_{j_2}$. Similarly, add to D_1 and D_2 the adjacencies $g_i^-c_{k_1}$ and $\gamma_i^-c_{k_2}$[2].

- Finally, for each $i \in [n]$ and each $j \in [2n - m]$, add to D_1 and D_2 the adjacencies $g_i^+\alpha_j, g_i^-\alpha_j, \gamma_i^+\alpha_j$ and $\gamma_i^-\alpha_j$.

This completes our construction. The intuition behind our hardness proof is that for each $i \in [n]$, we need to pick one of $g_i^+\gamma_i^+$ or $g_i^-\gamma_i^-$ in M, as we will show. Simultaneously, we would like to include as many adjacencies that are in both D_1 and D_2. It will possible to choose the positive and negative adjacencies *and* match all the c_j and α_j if and only if the 2P2N-3SAT instance is satisfiable.

It will be useful to think of D_1 (and D_2) as the set of adjacencies that are allowed to belong to M, as stated in the following.

Lemma 3. Let a be an adjacency in M, such that $a \notin D_1$ (equivalently, $a \notin D_2$). Then $M - a$ achieves a smaller total distance to A, D_1 and D_2 than M.

[2] Intuitively, these adjacencies represent using a literal to satisfy a specific clause. For instance, the adjacency $g_i^+c_{j_1}$ represents “setting x_i to true and satisfying C_{j_1}”.

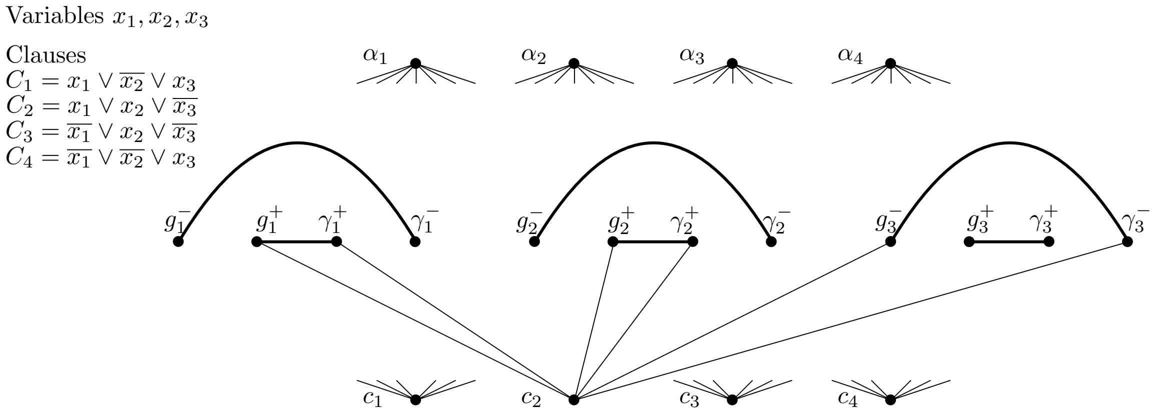

Variables x_1, x_2, x_3
Clauses
$C_1 = x_1 \vee \overline{x_2} \vee x_3$
$C_2 = x_1 \vee x_2 \vee \overline{x_3}$
$C_3 = \overline{x_1} \vee x_2 \vee \overline{x_3}$
$C_4 = \overline{x_1} \vee \overline{x_2} \vee x_3$
α_1
α_2
α_3
α_4
g_1^-
g_1^+
γ_1^+
γ_1^-
g_2^-
g_2^+
γ_2^+
γ_2^-
g_3^-
g_3^+
γ_3^+
γ_3^-
c_1
c_2
c_3
c_4

Figure 5.3: An example of a 2P2N-3SAT instance, with an illustration of the genes of M (only the gene tails are shown) and the adjacencies that are allowed by D_1 and D_2. The fat edges represent pairs of adjacencies of which at least one must be present according to Proposition 3. Among the c_j extremities, only the adjacencies for c_2 are shown.

Proof. By cutting a, we increase the distance to A by at most 1, but decrease the distance to D_1 and D_2 by 1 each. This is because $|(M-a)-D_1|+|D_1-(M-a)| = |M-D_1|-1+|D_1-M|$, the value of $\delta(M, D_1)$ is unchanged and $t(D_1) = 0$ by assumption (and the same holds for D_2). Therefore removing a from M yields a better median genome. □

Therefore, we may assume that every adjacency of a median M belongs to D_1 and D_2. Note that this implies that M contains no observed duplications (with respect to A), as no such adjacency is in D_1 and D_2. Thus we will ignore the $t(M_a) = 0$ term in $d_{\text{DSCJ}}(A, M_a)$ (eq. (5.3)), and we will not make a distinction between M_a and $r(M_a)$, as these are equal.

Another property of M is that it must contain at least one "positive" or one "negative" adjacency for each $i \in [n]$.

Lemma 4. For $i \in [n]$, M contains at least one of $g_i^+\gamma_i^+$ and $g_i^-\gamma_i^-$.

Proof. Suppose that for some i, M contains none of $g_i^+\gamma_i^+$ or $g_i^-\gamma_i^-$. Note that M does not contain $g_i^+\gamma_i^-$ nor $g_i^-\gamma_i^+$, by Lemma 3. This implies that $g_i'\gamma_i' \notin M_a$, as we have excluded all the four possibilities of having this adjacency in M_a.

Consider the median M' obtained from M by adding $g_i^+\gamma_i^+$, cutting the adjacencies that g_i^+ and γ_i^+ were contained in, if needed. If g_i^+ and γ_i^+ are both telomeres in M, then it is easy to check that $M' = M + g_i^+\gamma_i^+$ (M augmented by the adjacency $g_i^+\gamma_i^+$) attains a better distance than M since $g_i^+\gamma_i^+ \in D_1, D_2$ and $a(g_i^+)a(\gamma_i^+) = g_i'\gamma_i' \in A$ (this decreases the distance by 3).

Suppose that $g_i^+x \in M$ for some x, and that γ_i^+ is a telomere in M. By Lemma 3, g_i^+x is in both D_1 and D_2, which implies that $x = c_j$ or $x = \alpha_j$ for some j. This implies in turn that $a(g_i^+)a(x) \notin A$. We can argue that $M' = M - g_i^+x + g_i^+\gamma_i^+$ is better. To see this, observe that $|M' - D_1| = |M - D_1|$ and $|D_1 - M'| = |D_1 - M|$ (and the same with D_2). On the other hand, recalling that $g_i' \gamma_i' \notin M_a$, we have $|M_a' - A| = |M_a - A| - 1$ (because $a(g_i^+)a(x) \notin A$ and $a(g_i^+)a(\gamma_i^+) \in A$) and $|A - M_a'| = |A - M_a| - 1$ (because $a(g_i^+)a(\gamma_i^+) \in A$). We have thus decreased the distance by 2. The same argument applies if g_i^+ is a telomere but γ_i^+ is not.

Finally, suppose that $g_i^+ x$ and $\gamma_i^+ y$ are adjacencies of M. As we argued above, $a(g_i^+)a(x) \notin A$ and $a(\gamma_i^+)a(y) \notin A$. Letting $M' = M - g_i^+ x - \gamma_i^+ y + g_i^+ \gamma_i^+$, we find that $|M' - D_1| = |M - D_1|$ and $|D_1 - M'| = |D_1 - M| + 1$. As the same holds with D_2, we have increased the distance to D_1 and D_2 by 2. On the other hand, $|A - M_a'| = |A - M_a| - 1$ and $|M_a' - A| = |M_a - A| - 2$. To sum up, the total distance decreases by 1. □

We now formally prove the hardness of computing the SCJ-TD-FD median.

Theorem 4. The rooted SCJ-TD-FD median problem is NP-hard.

Proof. Let $x_1, \ldots, x_n$ and $C_1, \ldots, C_m$ be a 2P2N-instance, and let A, D_1, D_2 and the genes Γ of M be the corresponding instance of the SCJ-TD-FD median genome problem. We will show that the given 2P2N-3SAT instance is satisfiable if and only there exists a median genome M satisfying $d_{\text{DSCJ}}(A, M_a) + d_{\text{DSCJ}}(M, D_1) + d_{\text{DSCJ}}(M, D_2) \leq 2|D_1| - 2n + 4\delta(M, D_1)$.

($\Rightarrow$) Suppose that the 2P2N-3SAT can be satisfied by an assignment of the x_i variables to true or false. Construct a median genome using the following steps.

1. For each $i \in [n]$, if x_i is set to true, then add $g_i^- \gamma_i^-$ to M, and if instead x_i is set to false, add $g_i^+ \gamma_i^+$ to M.

2. Then, add to M these adjacencies in an algorithmic fashion: for each $j = 1, 2, \ldots, m$, consider clause C_j and let x_i be any variable satisfying C_j.

   - If x_i is set to true, then note that g_i^+ and γ_i^+ have not been matched in Step 1. Add $g_i^+ c_j$ to M if g_i^+ is not part of an adjacency of M yet, or add $\gamma_i^+ c_j$ to M otherwise.
   - If instead x_i is set to false, then g_i^- and γ_i^- have not been matched in Step 1. Add $g_i^- c_j$ if g_i^- is not part of an adjacency in M yet, or add $\gamma_i^- c_j$ to M otherwise.

   Note that since each x_i can satisfy at most two clauses, it will always be possible to find an extremity to match c_j with.

3. Finally, observe that so far each of the g_i^+, g_i^-, γ_i^+ and γ_i^- extremities are in an adjacency M, except $4n - 2n - m = 2n - m$ of them. Associate each such extremity g with a distinct α_j extremity arbitrarily, and add each $g\alpha_j$ to M, noting that there are just enough α_j genes to do so.

Note that M contains $n + m + 2n - m = 3n$ adjacencies in total, exactly n of which correspond to an adjacency of A (those included in Step 1). Also, every adjacency of M occurs in both D_1 and D_2. We have

$$\begin{aligned} d_{\text{DSCJ}}(A, M_a) &= |A - M_a| + |M_a - A| + 2\delta(A, M_a) - t(M_a) \\ &= 0 + 2n + 2n - 0 = 4n \end{aligned}$$

As for D_1 and D_2,

$$\begin{aligned} d_{\text{DSCJ}}(M, D_1) = d_{\text{DSCJ}}(M, D_2) &= |D_1 - M| + |M - D_1| + 2\delta(M, D_1) \\ &= |D_1| - 3n + 0 + 2\delta(M, D_1) \end{aligned}$$

Therefore the total distance is $4n + 2(|D_1| - 3n + 2\delta(M, D_1)) = 2|D_1| - 2n + 4\delta(M, D_1)$, as we predicted.

($\Leftarrow$) Suppose that there exists a median genome M of total distance at most $2|D_1| - 2n + 4\delta(M, D_1)$. By Lemma 3, we may assume that every adjacency of M is present in both D_1 and D_2.

With the next two claims, we will prove that M has exactly $3n$ adjacencies, of which exactly n are adjacencies corresponding to those in A.

Claim 1. $|M| \leq 3n$, and $|M| = 3n$ only if every c_j and α_j extremity is in some adjacency of M.

Proof. Call an extremity e of a gene in Γ *matchable* if there exists an adjacency of D_1 that contains e. By Lemma 3, the adjacencies of M only contain matchable extremities. The g_i^+, g_i^-, γ_i^+ and γ_i^- extremities account for $4n$ matchable extremities. The c_j genes account for m matchable extremities and the α_j genes for $2n - m$ matchable extremities . Thus there are $4n + m + 2n - m = 6n$ matchable extremities. Because an adjacency contains 2 extremities, there can be at most $3n$ adjacencies in M. The second part of the claim follows from the fact that we have to assume that every c_j and α_j is matched to attain this bound. □

For the rest of the proof, denote by q the number of distinct adjacencies $ab \in A$ for which there exists $xy \in M$ such that $a(x)a(y) = ab$.

Claim 2. $|M| = 3n$ and $q = n$.

Proof. By the definition of q, we have $|A - M_a| = n - q$ and $|M_a - A| = |M| - q$. It follows that

$$\begin{aligned} d_{\text{DSCJ}}(A, M_a) &= |A - M_a| + |M_a - A| + 2\delta(A, M_a) - t(M_a) \\ &= n - q + |M| - q + 2n - 0 \\ &= |M| + 3n - 2q \end{aligned}$$

Using Lemma 3, we also have $d_{\text{DSCJ}}(M, D_1) = |M - D_1| + |D_1 - M| + 2\delta(M, D_1) = 0 + |D_1| - |M| + 2\delta(M, D_1)$. Thus the sum of the 3 distances is

$$|M| + 3n - 2q + 2|D_1| - 2|M| + 4\delta(M, D_1) \leq 2|D_1| - 2n + 4\delta(M, D_1)$$

(this inequality is due to our initial assumption on the total distance of M). After simplifying, this gives $5n \leq |M| + 2q$. By Claim 1, $|M| \leq 3n$ and because A has n adjacencies, $q \leq n$. Hence, this inequality is only possible if $|M| = 3n$ and $q = n$. □

Because $q = n$, Claim 2 implies that for each $i \in [n]$, (at least) one of $g_i^+\gamma_i^+$ and $g_i^-\gamma_i^-$ is in M. This lets us define as assignment for our 2P2N-3SAT instance: for each $i \in [n]$, set x_i to *true* if $g_i^-\gamma_i^-$ is in M, and otherwise set x_i to $false$. We claim this this assignment satisfies every clause.

To see this, let C_j be a clause and let c_j be its corresponding extremity in M. By Claim 2, every extremity that is part of some adjacency in D_1 must be part of an adjacency in M, including c_j. Thus there is some e such that $c_je \in M$. By Lemma 3, the adjacency c_je must also be in D_1, and by construction either (1) $e \in \{g_i^+, \gamma_i^+\}$ for some x_i that occurs positively in C_j, or (2) $e \in \{g_i^-, \gamma_i^-\}$ for some x_i that occurs negatively in C_j. Suppose that case (1) applies. Then $c_jg_i^+$ or $c_j\gamma_i^+$ being in M means that $g_i^+\gamma_i^+ \notin M$, implying in turn that $g_i^-\gamma_i^-$ is in M. In this situation, we have set x_i to *true* and we satisfy C_j. Suppose instead that case (2) applies. Then $g_i^-\gamma_i^- \notin M$, in which case we have set x_i to false and satisfy C_j. As the argument applies to any clause C_j, this concludes the proof.

□

Remark 5. In the reduction above, none of the considered genomes contain a g-tandem array or a g-chromosome. So our result also implies the hardness of the rooted median problem where the distance between two genomes A and D, where A is an ancestor of D, is defined in a simpler way as $|A - D| + |D - A| + 2\delta(A, D)$, i.e. does not contain a term related to reducing the descendant genome.

5.5 An Integer Linear Program

We now describe a simple Integer Linear Program (ILP) to solve the rooted median problem. The key idea, already used in previous median problems [83], including the directed median problem discussed above, is to convert the rooted median problem into an instance of a MWM problem, albeit with certain additional constraints. More precisely, in this approach we define a complete graph G on the extremities g_h and g_t of every gene g in Γ. A pair of distinct extremities defines an edge and thus a potential adjacency in M, which is thus defined by a matching in G. Each edge is assigned a weight that reflects the number of descendant genomes that contain the corresponding adjacency. Further, each edge is assigned a color that reflects its corresponding adjacency in the ancestral genome, if any, and the number of colors of the selected edges also contributes to the weight of the matching defining the median M.

An alternative formulation for the distance. We first introduce an alternative formula to compute the directed distance, denoted by $d_{\mathrm{DSCJ}}(u, v)$, from an ancestor u to a descendant v. For the rooted median problem, the pair (u, v) can represent either the pair (A, M_a) or any pair (M, D_i). The new formulation is easier to handle in an ILP framework than Eq. (5.3). We denote by $n_v(g)$ the number of copies of gene g in v, by $n_v(g_hg_t)$ the number of occurrences of adjacency g_hg_t in v, and by $t_v(g)$ the number of observed duplications of gene g in v. Note that $t_v(g) \in \{n_v(g_hg_t)-1, n_v(g_hg_t)\}$, the case $t_v(g) = n_v(g_hg_t)-1$ occurring when adjacencies g_hg_t form only g-chromosomes. Further, let $t(v) = \sum_{g\in\Gamma_u} t_v(g)$ denote the total number of observed duplications in v, where Γ_u is the set of genes of u and also the alphabet of genes of v.

To rewrite $d_{\mathrm{DSCJ}}(u, v)$, we introduce an indicator variable $\alpha_{g,uv}$, where $\alpha_{g,uv} = 1$ if g_hg_t is common to both u and v, but all occurrences were removed while reducing v. Formally, $\alpha_{g,uv} = 1$ if $g_hg_t \in u \cap v$ and $g_hg_t \notin r(v)$; otherwise $\alpha_{g,uv} = 0$. It is then relatively straightforward to show that

$$d_{\mathrm{DSCJ}}(u, v) = |u - v| + |v - u| + 2\delta(u, v) - 2t(v) + 2\sum_{g\in\Gamma_u} \alpha_{g,uv} \tag{5.4}$$

Proof. From eq. (5.3), we have $d_{DSCJ}(u, v) = |u - r(v)| + |r(v) - u| + 2\delta(u, v) - t(v)$. However, it is easier to express the distance without the reduced genome terms. Hence, we eliminate the need for computing the reduced genomes by replacing $|u - r(v)|$ and $|r(v) - u|$ by suitable expressions as follows. We show that (1) $|u - r(v)| = |u - v| + \sum_{g\in\Gamma_u} \alpha_g$, and (2) $|r(v) - u| = |v - u| - t(v) + \sum_{g\in\Gamma_u} \alpha_g$. Substituting the terms in eq. (5.3) yield eq. (5.4).

(1) Consider first the difference between $u - r(v)$ and $u - v$. Suppose that $xy \in u - v$ but $xy \notin u - r(v)$. Then $xy \in r(v)$ but $xy \notin v$, which is not possible. Thus the difference can only be due to some $xy \in u - r(v)$ such that $xy \notin u - v$. This means that $xy \notin r(v)$ and $xy \in v$, which only happens when $xy = g_hg_t$ for some gene g. As we have $xy = g_hg_t \in u \cap v$ and $g_hg_t \notin r(v)$, we also have $\alpha_g = 1$, by definition. Since only one such adjacency is possible for each gene g (because u is trivial), $u - r(v)$ and $u - v$ differ only by adjacencies on genes for which $\alpha_g = 1$. We have shown that $|u - r(v)| = |u - v| + \sum_{g\in\Gamma_u} \alpha_g$.

(2) Now consider the difference between $r(v) - u$ and $v - u$. Note that there are $t(v)$ adjacencies in v not in $r(v)$, all observed duplications of the type $g_h g_t$. Let $g \in \Gamma_u$. If $g_h g_t \notin u$, then all of the $t(g)$ observed duplications in g are counted in $v - u$ but not in $r(v) - u$. This is also true when $g_h g_t \in u$ and $g_h g_t \in r(v)$. In these cases, $\alpha_g = 0$. However when $g_h g_t \in u \cap v$ but $g_h g_t \notin r(v)$, there are $t(g) - 1$ of the $g_h g_t$ adjacencies counted in $v - u$ not counted in $r(v) - u$ (this is because exactly one $g_h g_t$ adjacency of v can be matched with the $g_h g_t$ adjacency in u, and $r(v)$ has no such adjacency). This case occurs precisely when $\alpha_g = 1$. This shows that $|r(v) - u| = |v - u| - \sum_{g \in \Gamma_u}(t(g) - \alpha_g) = |v - u| - t(v) + \sum_{g \in \Gamma_u} \alpha_g$. □

This formulation is interesting due to the fact it does not rely on the notion of reduced genome. We will discuss later how variables $\alpha_{g,uv}$ and $t_v(g)$ can be handled simply in an ILP framework.

Reformulating the objective function. We now use Eq. (5.4) to reformulate the objective function of the rooted median problem.

Claim 3. Minimizing the function eq. (5.1) defining the evolutionary cost of a median M is equivalent to maximizing the following expression:

$$\sum_{i=1}^{k}\left(2|M \cap D_i| - 2\sum_{g \in \Gamma_M} \alpha_{g,MD_i}\right) + 2|A \cap M_a| + 2t(M_a) - 2\sum_{g \in \Gamma_A} \alpha_{g,AM_a} - (k+1)|M| \quad (5.5)$$

where Γ_A and Γ_M are the set of genes of A and M, respectively, and so also the gene alphabets for M and the D_is, and variables α_{g,AM_a} and α_{g,MD_i} are defined as $\alpha_{g,uv}$ above.

Proof. By eq. (5.4), we know that

$$d_{\mathrm{DSCJ}}(A, M_a) = |A - M_a| + |M_a - A| + 2\delta(A, M_a) - 2t(M_a) + 2\sum_{g \in \Gamma_A} \alpha_{g,AM_a}$$

$$d_{\mathrm{DSCJ}}(M, D_i) = |M - D_i| + |D_i - M| + 2\delta(M, D_i) - 2t(D_i) + 2\sum_{g \in \Gamma_M} \alpha_{g,MD_i}$$

where Γ_A and Γ_M are the set of genes in the gene orders of A and M, respectively, and so also the genes alphabets for M and the D_is. Variables α_{g,AM_a} and α_{g,MD_i} are defined as $\alpha_{g,uv}$ above.

For any two adjacency sets X and Y, we use the identity $|X - Y| + |Y - X| = |X| + |Y| - 2|X \cap Y|$ to obtain

$$d_{\mathrm{DSCJ}}(A, M_a) = |A| + |M_a| - 2|A \cap M_a| + 2\delta(A, M_a) - 2t(M_a) + 2\sum_{g \in \Gamma_A} \alpha_{g,AM_a},$$

$$d_{\mathrm{DSCJ}}(M, D_i) = |M| + |D_i| - 2|M \cap D_i| + 2\delta(M, D_i) - 2t(D_i) + 2\sum_{g \in \Gamma_M} \alpha_{g,MD_i}.$$

This eliminates the need to count the actual number of cut and join events along every branch. Instead, it suffices to compute the common adjacencies in the parent and child genomes (using the terms $|A \cap M_a|$ and $|M \cap D_i|$) for each branch (A, M_a) and (M, D_i).

For a median M, let $s(M) = d_{\text{DSCJ}}(A, M_a) + \sum_{i=1}^{k} d_{\text{DSCJ}}(M, D_i)$ be the *score* of M. It follows easily from above that

$$\begin{aligned} s(M) = &\left[|A| + 2\delta(A, M_a) + \sum_{i=1}^{k} (|D_i| + 2\delta(M, D_i)) \right] \\ &- \left[\sum_{i=1}^{k} \left(2|M \cap D_i| + 2t(D_i) - 2 \sum_{g \in \Gamma_M} \alpha_{g,MD_i} \right) \right. \\ &\left. + 2|A \cap M_a| + 2t(M_a) - 2 \sum_{g \in \Gamma_A} \alpha_{g,AM_a} - (k+1)|M| \right] \end{aligned}$$

Let $N = |A| + 2\delta(A, M_a) + \sum_{i=1}^{k} \big(|D_i| + 2\delta(M, D_i) + 2t(D_i)\big)$. Given that N depends only on A and D_i and not on M, it is constant (note that $\delta(A, M_a)$ and $\delta(M, D_i)$ are constant as the gene content of M is an input to the problem). Thus in order to minimize the score $s(M)$, we only need to maximize the term:

$$\sum_{i=1}^{k} \left(2|M \cap D_i| - 2 \sum_{g \in \Gamma_M} \alpha_{g,MD_i} \right) + 2|A \cap M_a| + 2t(M_a) - 2 \sum_{g \in \Gamma_A} \alpha_{g,AM_a} - (k+1)|M|$$

which is negated in $s(M)$, as required in eq. (5.5). □

Such a reformulation of the objective function enables us to translate the problem as an instance of a *colored MWM problem*, as will be made clear in the subsequent paragraphs.

An interpretation as a colored MWM problem. The terms $\alpha_{g,uv}$ and $t(M_a)$ in Eq. (5.5) account for the presence of observed duplications. In the absence of observed duplications however, solving the rooted median problem requires to find a matching in G that maximizes the sum of the weight of the selected edges and of the number of colors represented by the matching edges. The matching edges weight is partly accounted for by the term $|M \cap D_i|$, while on the other hand, $|A \cap M_a|$ determines the number of colors used in the matching. Using the intersection terms in the objective function, we now interpret the notion of *weight* and *color* of an edge in terms of decision variables of an ILP.

In order to compute $|M \cap D_i|$, we introduce the variable $\gamma_i(e)$ denoting the existence of a potential adjacency e of M in a genome D_i: we put $\gamma_i(e) = |e \cap D_i|$, i.e. $\gamma_i(e) = 1$ if $e \in D_i$ and 0, otherwise. For each adjacency e in the graph G, the weight $w(e)$ of e is determined using the weight function $w : E(G) \rightarrow \mathbb{N}$:

$$w(e) = 2\left(\sum_{i=1}^{k} \gamma_i(e)\right) - (k+1)$$

Since M is trivial w.r.t. every D_i, the weights for edges $e \in M$ will account for the term $\sum_{i=1}^{k} 2|M \cap D_i| - (k+1)|M|$ in Eq. (5.5). However, this principle does not work with A. Indeed, it is possible that $x_1y_1 \in M$ and $x_2y_2 \in M$ such that $a(x_1)a(y_1) = a(x_2)a(y_2) \in A$. In this situation, only one of x_1y_1 or x_2y_2 can contribute to $|A \cap M_a|$, but both $|x_1y_1 \cap A|$ and $|x_2y_2 \cap A|$ equal to 1. In other words, we cannot simply sum the adjacencies of M_a that are in A.

To address this issue, we introduce the notion of a *color family*. Let m_A be the number of adjacencies in A. Each number from the set $\{1, 2, ..., m_A\}$ represents a distinct color. We arbitrarily assign a distinct color from this set to each adjacency in A. If $E(G)$ is the edge set of G, representing all possible adjacencies in M, then every adjacency in $E(G)$ is assigned a color from $\{1, 2, ..., m_A\} \cup \{0\}$, consistent with the orthology relations: the adjacency $xy \in M$ receives color $i \neq 0$ if the adjacency $a(x)a(y)$ is present in A and was assigned color i, and color 0 if $a(x)a(y)$ is not present in A. The set of adjacencies having the same color i form a color family, represented by E_i. We denote by C the coloring function $E(G) \rightarrow \{0, 1, ..., m_A\}$ defined as described above. Notice that a color i contributes exactly once to the term $|A \cap M_a|$ if there exists at least one adjacency in M that belongs to the color family i.

Reducing the size of the ILP. The size of the ILP we are about to describe is polynomial in the sum of the considered genomes. As the total number of adjacencies is quadratic in the number of genes in M, it can reach large values when dealing with large genomes, thus making the ILP challenging to solve in practice. We show that the set of decision variables can be restricted to specific adjacencies, that we call *candidate adjacencies.*An adjacency xy is a *candidate adjacency* for the median if at least $\left\lfloor \frac{k+1}{2} \right\rfloor + 1$ genomes from the set $\{A, D_1, D_2, ..., D_k\}$ contain xy (where here A contains xy if $a(x)a(y) \in A$). Lemma 5 shows that the number of adjacencies to consider in an ILP is linear in the sum of the sizes of the input genomes.

Lemma 5. There exists an optimal median consisting of only candidate adjacencies. Furthermore, when k is even, an adjacency which is not a candidate adjacency can not be a part of any optimal median.

Proof. To prove this lemma, we start with a median containing a non-candidate adjacency. For odd values of k, we prove that removing the non-candidate adjacency results in another median of the same cost whereas for even k, it is shown that the resultant median (on removing the non-candidate adjacency) is better. We temporarily ignore the influence of reduced genomes for this proof.

Consider an adjacency xy that is not a candidate. Recall that since xy is not a candidate it is present in at most $\left\lfloor \frac{k+1}{2} \right\rfloor$ genomes from $\{A, D_1, ..., D_k\}$. Assume that M is a median genome and xy is present in M. Further, assume that M is optimal. Thus, the sum of the distances $d_{\text{DSCJ}}(A, M_a) + \sum_{i=1}^{k} d_{\text{DSCJ}}(M, D_i)$ should be the least over all medians. Let M' be the genome obtained by removing xy from M.

Let $D_{xy} \subseteq \{D_1, ..., D_k\}$ be the set of descendant genomes that contain xy, and let $\overline{D_{xy}}$ be the set of those that do not. For any $D_i \in D_{xy}$, the adjacency need not be cut along (M, D_i), however it has to be added along (M', D_i), introducing an extra cost of 1 to the total distance. Thus, $d_{\text{DSCJ}}(M, D_i) = d_{\text{DSCJ}}(M', D_i) - 1$, for all $D_i \in D_{xy}$. On the other hand, if $D_i \notin D_{xy}$, then it does not contain xy. Consequently, for all such D_i, the adjacency has to be cut along (M, D_i) but not along (M', D_i) (since M' does not contain it in the first place). Thus, for all $D_i \notin D_{xy}$, $d_{\text{DSCJ}}(M, D_i) = d_{\text{DSCJ}}(M', D_i) + 1$.

Further if A contains $a(x)a(y)$, it need not be cut along (A, M_a) but may need to be cut along (A, M'_a) thereby introducing a possible extra cost of 1 (note here the possibility that some $x^*y^* \in M$ distinct from xy such that $a(x^*)a(y^*) = a(x)a(y)$). Thus, $d_{\text{DSCJ}}(A, M_a) \geq d_{\text{DSCJ}}(A, M'_a) - 1$. If instead, A does not contain xy then it has to be joined along (A, M_a) and not along (A, M'_a). Unlike the previous case, the cost of the join is unavoidable. Hence, $d_{\text{DSCJ}}(A, M_a) = d_{\text{DSCJ}}(A, M'_a) + 1$.

Case 1: A contains xy. Then $|D_{xy}| \leq \left\lfloor \frac{k+1}{2} \right\rfloor - 1$.

$$\begin{aligned}
d_{\text{DSCJ}}(A, M_a) &\geq d_{\text{DSCJ}}(A, M'_a) - 1 & \\
d_{\text{DSCJ}}(M, D_i) &= d_{\text{DSCJ}}(M', D_i) - 1 & \forall D_i \in D_{xy} \\
d_{\text{DSCJ}}(M, D_i) &= d_{\text{DSCJ}}(M', D_i) + 1 & \forall D_i \notin D_{xy}
\end{aligned}$$

Summing over all the input genomes, we get

$$\begin{aligned}
d_{\text{DSCJ}}(A, M_a) + \sum_{D_i \in D_{xy}} d_{\text{DSCJ}}(M, D_i) \geq d_{\text{DSCJ}}(A, M'_a) + \sum_{D_i \in D_{xy}} d_{\text{DSCJ}}(M', D_i) \\
+ |\overline{D_{xy}}| - (|D_{xy}| + 1)
\end{aligned}$$

We know that $|D_{xy}| + 1 \leq \left\lfloor \frac{k+1}{2} \right\rfloor$. If k is even, $|\overline{D_{xy}}| > |D_{xy}| + 1$. Hence,

$$d_{\text{DSCJ}}(A, M_a) + \sum_{D_i \in D_{xy}} d_{\text{DSCJ}}(M, D_i) > d_{\text{DSCJ}}(A, M'_a) + \sum_{D_i \in D_{xy}} d_{\text{DSCJ}}(M', D_i)$$

Thus, the cost of M' is better than that of the optimal median M and we have a contradiction. If k is odd, then $|\overline{D_{xy}}| = |D_{xy}| + 1$ and hence both M and M' incur the same overall cost. In other words, the removal of a non-candidate adjacency does not increase the cost of the optimal median. Thus, iteratively removing all such adjacencies will yield an optimal median that consists solely of candidate adjacencies.

Case 2: A does not contain xy. Then $|D_{xy}| \leq \left\lfloor \frac{k+1}{2} \right\rfloor$.

$$\begin{aligned} d_{\text{DSCJ}}(A, M_a) &= d_{\text{DSCJ}}(A, M'_a) + 1 & & \\ d_{\text{DSCJ}}(M, D_i) &= d_{\text{DSCJ}}(M', D_i) - 1 & & \forall D_i \in D_{xy} \\ d_{\text{DSCJ}}(M, D_i) &= d_{\text{DSCJ}}(M', D_i) + 1 & & \forall D_i \notin D_{xy} \end{aligned}$$

The analysis in this case is similar to Case 1. On adding all the equations and using $|D_{xy}| \leq \left\lfloor \frac{k+1}{2} \right\rfloor$, once again we reach a contradiction when k is even. When k is odd, both M and M' yield the same overall distance. Thus, we can still obtain the optimal median by iteratively removing non-candidate adjacencies.

Thus, when k is odd, there exists at least one optimal median consisting only of candidate adjacencies. However, when k is even, the optimal median must consist only of candidate adjacencies. □

Remark 6. The difficulty of the rooted median problem stems from the fact that duplication from M to the D_is can create conflicting adjacencies, where a median gene extremity belongs to several candidate adjacencies. It is interesting to observe that this can happen only due to convergent evolution, i.e. the fact that the same adjacency is created independently in several D_is. This suggests that in the practical context of a limited level of convergent evolution, the rooted median problem is easy to solve.

The ILP for the rooted median problem. We can now provide the complete ILP formulation to solve the rooted SCJ-TD-FD median problem. Let $x(e)$ be a binary decision variable denoting the inclusion of edge (candidate adjacency) $e \in E(G)$ in M. Also, let c_i be a binary decision variable indicating if at least one edge with color i belongs to M. From the previous paragraph, one can write the objective function as
Maximize:

$$\sum_{e \in E(G)} w(e)x(e) + 2\sum_{i=1}^{m_A} c_i + 2t(M_a) - 2\sum_{g \in \Gamma_A} \alpha_{g,AM_a} - 2\sum_{i=1}^{k} \sum_{g \in \Gamma_M} \alpha_{g,MD_i}$$

We now describe the constraints of the ILP. The first set of constraints concern the *consistency* of the set of chosen adjacencies, that ensures that each gene extremity in M belongs to at most one adjacency, or in other words that M is a matching for the graph G (these are the first two sets of constraints below). Next, we use an additional set of constraints to determine the values of c_i, $i = \{1, 2, ..., m_A\}$. If at least one adjacency of color i is present in the median, $c_i = 1$, otherwise $c_i = 0$. The following inequalities define these color constraints:

$$\sum_{e=(y_h,z)} x(e) \leq 1 \qquad \forall y \in \Gamma_M \tag{5.6}$$

$$\sum_{e=(y_t,z)} x(e) \leq 1 \qquad \forall y \in \Gamma_M \tag{5.7}$$

$$c_i = \left\lceil \frac{\sum_{C(e)=i} x(e)}{|E_i|} \right\rceil \qquad \forall i \in \{1, 2, ..., m_A\} \tag{5.8}$$

Note that for c_i above, the constraints of the type $x = \lceil y \rceil$ are not linear, but if x is restricted to be in $\{0, 1\}$, it can be replaced by the constraint $y \leq x \leq y + \epsilon$, where ϵ is very close to 1, say 0.999. A similar trick can be used for floor functions.

In order to compute $\alpha_{g,uv}$ for every pair (u, v) – where either $u = A$, $v = M_a$ or $u = M, v = D_i$ for some i – and every gene $g \in \Gamma_u$, we use some additional constraints. Let $p_v(e)$ be the binary variable denoting if the adjacency e exists in v. We use an indicator variable $\lambda_{g,uv}$ such that $\lambda_{g,uv} = 1$ if and only if all copies of g are involved in $g_h g_t$ adjacencies. Consequently, $\lambda_{g,uv} = 1$ ensures the existence of the $g_h g_t$ adjacency in $r(v)$. Thus, $\lambda_{g,uv} = \left\lfloor \frac{n_v(g_h g_t)}{n_v(g)} \right\rfloor$. Further, we use $\Lambda_{g,uv}$ to indicate if at least one instance of $g_h g_t$ has been observed in v. Thus, we can represent $\Lambda_{g,uv}$ as $\left\lceil \frac{n_v(g_h g_t)}{n_v(g)} \right\rceil$. Since we already know the gene orders of A and each D_i, the values of $p_A(e)$ and $p_{D_i}(e)$ are known. Further, $p_M(e) = x(e)$. Thus, we obtain the following constraints for every gene g and branch (u, v):

$$\lambda_{g,uv} = \left\lfloor \frac{n_v(g_h g_t)}{n_v(g)} \right\rfloor \tag{5.9}$$

$$\Lambda_{g,uv} = \left\lceil \frac{n_v(g_h g_t)}{n_v(g)} \right\rceil \tag{5.10}$$

$$\alpha_{g,uv} = \min(p_u(g_h g_t), \Lambda_{g,uv} - \lambda_{g,uv}) \tag{5.11}$$

$$_v t(g) = n_v(g_h g_t) - \lambda_{g,uv} \tag{5.12}$$

We use the fact that if $g_h g_t \notin v$ for some g then $g_h g_t \notin r(v)$. Thus, if $g_h g_t \notin v$, $\lambda_{g,uv} = 0$ thereby ensuring the correctness of constraints to find $\alpha_{g,uv}$. Again, note that the min function is not linear, but that a constraint $x = \min(y, z)$ can be replaced by $x \geq y$ and $x \geq z$, assuming that $x, y, z \in \{0, 1\}$.

5.6 Experimental results

We ran experiments on simulated data in order to evaluate the ability of the ILP to correctly predict the gene order of the median genome. The input for the program, including gene orders for the ancestor genome A and the descendant genomes D_i, along with the orthology relations, generated using the ZOMBI genome simulator [23]. The ILP was solved using the Gurobi solver version 7.5.2. The code was written in Python.

Simulations parameters. Our input genomes consisted of one ancestor A and two descendants D_1 and D_2. We started with the ancestral genome A as a single circular chromosome consisting of 1000 genes, belonging to different gene families (so without duplicate genes). The genome A evolved into the median genome M using duplications, inversions and translocations. The genome M was further evolved along two independent branches to yield the descendant genomes, D_1 and D_2. The total number of rearrangements (inversions + translocations) from A to M and from M to D_i was varied from 100 to 500, in steps of 100. The parameter for duplication events was kept constant throughout the experiments. The average number of duplicated genes, over all three branches collectively, was found to be 362.8 with a standard deviation of 82 genes. Considering the number of duplication events, the mean and standard deviation of segmental duplications over the three branches was 72.6 and 15.8 respectively. The lengths of segmental duplications, inversions and translocations were controlled using specific extension rates. These extension rates (all between 0 and 1) are the parameters of a geometric distribution that dictated the respective lengths. Thus, the length of the segment being acted upon would be 1 if the extension rate parameter is set to 1 and would increase as the parameter value reduces. In our experiments, the inversion, translocation and duplication extension rates were 0.05, 0.3 and 0.2 respectively. For each setting (number of rearrangements) we ran 40 simulations.

Results. For each simulation, we compared the optimal median according to the ILP to the actual median generated by the simulator. For each group, we measured the average precision and recall statistics. The ILP predicts the median genome in the form of its adjacency set. Thus, in this context, precision refers to the ratio of number of correctly predicted adjacencies to the total number of adjacencies in the computed optimal median. On the other hand, recall represents the ratio of the correctly predicted adjacencies to the total number of adjacencies in the actual median. For each instance, we measured the number of candidate adjacencies used in the ILP. Additionally, to evaluate the effectiveness of our approach, we also measured the number of adjacencies in the solution that were common to all genomes (A, D_1 and D_2) and those common to only two of the three.

An overview of the results is given in Table 5.1. The ILP rarely predicts an erroneous adjacency to be a part of the optimal median, with a near-perfect precision. This property is observed throughout the experiments irrespective of the number of rearrangement events. On the other hand, the ILP predicts more than 90% of the median for lower rates of rearrangement and a decreasing trend is observed as the number of rearrangement events increase. This can be partly attributed to the decrease in the number of candidate adjacencies. In general, the number of candidate adjacencies is lower than the true number of

Events	Adj. in true median	Cand. adj.	Adj. in ILP median	Precision	Recall	% Adj. common to all genomes	% Adj. common to two genomes	No. of optimal solutions	Avg. time per run (in sec)
100	1514	1503	1493	0.9998	0.9859	86.43	13.57	2.3	53
200	1107	1062	1044	0.9991	0.9428	69.49	30.51	15.8	29
300	1312	1192	1155	0.9985	0.8758	52.94	47.06	40.3	38
400	1151	985	961	0.9981	0.8329	49.44	50.56	393.7	51
500	1430	1174	1132	0.9972	0.7897	46.68	53.32	3682.6	84

Table 5.1: Statistics of the ILP median experiment on simulated data.

adjacencies in the median, as including other adjacencies may result in a non-optimal median. This, however, emphasizes the practicality of Lemma 5, as the number of adjacency variables is significantly reduced. It can also be observed that the number of adjacencies common to all genomes decreases with increase in rearrangements. These adjacencies will be preferred by the ILP on account of higher weight.

Another notable observation is the increase in the number of optimal solutions with larger rates of rearrangement. This correlates naturally with the decrease in the number of adjacencies that are common to all genomes. For only 100 rearrangements, the ILP outputs a unique optimal median in most runs, with an overall average of 2.3 solutions. However, the average number of optimal solutions exceeded 3000 in case of 500 rearrangements. Despite a pool of optimal solutions, the SCJ distance between the actual median and an optimal median does not vary by much. If the SCJ distance between the actual median and a randomly chosen optimal median is D, then the distance between the actual median and any other optimal median was observed to stay within the range $(D-2, D+2)$.

5.7 Conclusion

In this chapter, we introduced two versions of the median problem, namely the directed median and the rooted median problem. We studied both the problems under the SCJ-TD-FD model. We proved that computing the median with the most parsimonious directed distance for an ancestor A and descendants D_i, $i = 1$ to k is NP-hard by reduction from the 2P2N-3SAT problem. This contrasts with the directed median problem that does not involve an ancestral genome A. An interesting feature of our hardness proof is that it relies on two identical descendant genomes, showing a sharp tractability boundary between the directed pairwise distance problem and the rooted median of three problem. Similarly to other SCJ-related median problem, our rooted median problem aims at selecting adjacencies among candidate adjacencies that are seen in a majority of the given input genomes; nevertheless the possibility of conflicting median adjacencies due to convergent evolution is at the heart of the intractability of the problem (Remark 6). To address this intractability, we provide a simple Integer Linear Program that computes an optimal median. Unsurprisingly, we observe that our ILP outputs a more reliable estimate of the median in case of lower rates of rearrangements. Moreover, we observe that despite having many more optimal solutions for higher rates of rearrangement, the distance of a random solution from the actual median does not deviate by much.

Our work can be commented with regard to the Small Parsimony Problem under the directed SCJ-TD-FD model. The hardness result of the rooted median problem likely implies the corresponding SPP problem is also NP-hard. This motivates our current work about extending the rooted median ILP toward the SPP. It is worth noting that our median ILP can also be used to solve the SPP by iterative application from an initial assignment of ancestral gene orders, similarly to the early SPP solvers for genome rearrangements such as GRAPPA [57]. Considering the multiplicity of the solutions, it also remains to be investigated if the sampling and subsequent analysis of co-optimal evolutionary scenarios, in a similar manner as [46], is possible within this framework.

Chapter 6

The Small Parsimony Problem with single-gene duplications

The rooted median problem is motivated by iterative approaches for ancestral reconstruction [76]. In the previous chapter, we have seen how the adjacency sets of descendant genomes and the root genome can be used to solve this problem.

In this chapter, we provide another ILP-based approach for reconstructing ancestral gene orders by solving the Small Parsimony Problem. The problem aims at finding the most parsimonious assignment of gene orders for the internal nodes of the tree, given a species tree topology and the gene orders of the extant species. The ILP described in this chapter was developed in collaboration with Dr. Pedro Feijão.

6.1 Introduction

Approaches designed to address the Small Parsimony Problem can be widely divided into two types: homology-based and parsimony-based. Homology-based approaches do not consider genome rearrangements directly. Instead, they rely on the use of *conserved intervals* as a measure of similarity between two genomes [7]. These structures are a generalized notion of a conserved adjacency, defining genome segments that may be preserved. Aided by set theoretic operations on permutations, these approaches provide a set of possible gene orders (as sets of permutation intervals) for each ancestral species in a given species tree [4]. Moreover, for most homology-based methods, this is achieved in linear time. Other approaches based on conserved structures (also referred to as Contiguous Ancestral Regions or CARs) have also been subsequently proposed [49].

On the other hand, parsimony-based approaches are guided by minimizing the evolutionary cost using a set of genome rearrangement events. These methods were inspired by a multitude of results on the pairwise genome rearrangement distance problem [29]. The aim of these methods is to reconstruct all ancestral genomes so that the rearrangement scenarios between successive gene orders along the species tree can be explained using the minimum number of evolutionary events. They are specific to the evolutionary model used, as has been illustrated in [1] and [90] through the use of breakpoint model and the Double-Cut-or-Join model respectively. A more general version of the problem, known as the *Multiple Genome Rearrangement Problem* aims at finding a phylogenetic tree inferring the most likely rearrangement scenario for a set of species [76]. However, even when restricted to its simplest instance, which translates to the median problem, the problem has been proved to be NP-hard for most cases [63, 15, 83]. In case of the Small Parsimony Problem, with the exception of the Single-Cut-or-Join distance (SCJ)[28], the problem is NP-hard for most rearrangement distances [83]. To combine the two approaches, Luhmann et al provided PhySca algorithm, reconstructing gene orders using an optimality criterion that is a linear combination of conserved synteny blocks and the SCJ distance [46]. The importance given to the individual criteria was decided using the convexity parameter $\alpha \in [0, 1]$, $\alpha = 0$ being the case that favored minimizing the SCJ distance. It was proved that considering both the approaches simultaneously, the Weighted SCJ Labeling problem is NP-hard for $33/34 < \alpha < 1$, while the problem is otherwise open [46].

Duplication events play an important role in genome evolution [26]. For instance, mosquito genomes have been known to exhibit high levels of duplication [2]. Under most rearrangement models, finding the median itself is NP-hard. As a result, the SPP is also intractable. However, the problem of ancestral genome reconstruction with duplications has been studied using homology-based approaches that conserve synteny blocks [48, 67]. While a holistic approach that considers maximizing the weights of selected adjacencies as well as minimizing the rearrangement cost has been provided under the SCJ distance, a similar method is yet to be provided in the duplication-aware framework.

The previous chapters discussed the distance and median problems under the SCJ-TD-FD distance, in the context of directed evolution from ancestral genomes to descendant genomes. It is assumed that the descendant may contain multiple copies of a gene in the ancestor. However, the ancestral genome can contain only one parent of a gene in the descendant. The relations between genes along any branch are obtained through orthology relations, obtained using reconciled gene trees.

In this chapter, we study the Small Parsimony Problem (SPP) accounting for duplicate genes in the descendant genomes. This is a natural extension of the idea introduced in [46], which discussed the problem under the SCJ distance. Research in this direction has been motivated, among other reasons, by the study of mosquito genome evolution. Consequently, the methods described in subsequent sections are used on a data set of Anopheles mosquito genomes [61]. We use the modified the SCJ-TD-FD distance from chapter 5 to accommodate for the possible discrepancies arising from single-gene duplications.

6.2 Problem statements

The Small Parsimony is an important problem studied in relation with ancestral reconstruction. In this problem, we are provided as input, a phylogenetic tree with extant genomes at its leaves. The gene content of every genome as well as the gene orders for the extant genomes are also available. Furthermore, the orthology relations between each ancestral gene and its descendants for every branch can be obtained using reconciled trees. Given such a setting the general version of the SPP under the SCJ-TD-FD model can be stated as follows:

Mixed SCJTDFD Small Parsimony Problem: Let a phylogenetic species tree T and the extant genomes (at the leaves) be provided. Any pair of genomes u, v, v being the child of u in T, is allowed to have unequal gene content (with respect to gene families and copy numbers). Compute the genomes at the ancestral nodes such that the total SCJTDFD distance over the tree is minimized.

However, we focus on the linear version of the problem, in which the gene orders are a set of linear chromosomes. We use an Integer Linear Program (ILP) that provides a solution minimizing the total evolutionary cost over the phylogenetic tree. The number of variables in this context increases with the number of candidate adjacencies. This presents the possibility of a substantially large matrix. To avoid this issue, we also provide as input, a list of candidate adjacencies with their weights, which is significantly shorter than the list of all possible adjacencies. The list of candidate adjacencies is obtained using the DeCoSTAR software [25]. Accounting for the aforementioned conditions, we address the following problem:

Weighted Linear SCJTDFD Small Parsimony Problem: Given the same setting as the Mixed SCJTDFD SPP and a list of weighted candidate adjacencies for every genome, compute the genomes at each node such that the total SCJTDFD distance over the tree is minimized and each chromosome in every genome is linear.

6.3 ILP for Small Parsimony Problem

Consider a phylogenetic tree $T = (V, E)$. Let $v \in V$ be a node in T and u be the parent of v. For every node $v \in V$, we define an adjacency graph on the extremities of all the genes. An edge between two extremities represents a possible adjacency. Using the definitions from [46], $p_{v,a} = 1$ if an adjacency a exists in species v and 0 otherwise. For every adjacency a in v, weight $w_{v,a} \in [0, 1]$ represents the confidence measure of the existence of the adjacency a in the species v.

Given this setting, the *Weighted SCJ-TD-FD labeling problem* aims at minimizing:

$$\sum_{v \in V} \left(\alpha \left(\sum_{a \in v} (1 - p_{v,a}) w_{v,a} \right) + (1 - \alpha) d_{DSCJ}(u, v) \right)$$

Let $ext(u)$ and $ext(v)$ be the sets of extremities of genes in the respective genomes. We define A_u, the adjacency set for genome u as

$$A_u = \{a = xy | x, y \in ext(u), x \neq y\}$$

A_v is defined in a similar manner as above, replacing u by v.

Consider A (respectively, D) as the set of all adjacencies a in A_u (respectively A_v) such that $p_{u,a}$ (respectively $p_{v,a}$) equals 1. Thus, $d_{DSCJ}(u, v)$ can be defined as

$$|A - D| + |D - A| + 2\delta(u, v) + 2\eta(u, v) - 2t(v) + 2 \sum_{g \in G_u} \alpha_g$$

where $\delta(u, v)$ is the number of duplicate genes in v compared to u, $\eta(u, v)$ is the number of newly created genes and $t(v)$ is the observed number of g-adjacencies in v.

We call the adjacency $xy \in A_u$ as the parent adjacency of an adjacency $x'y' \in A_v$ if x' originated from x and y' from y. Conversely, $x'y'$ is the child (adjacency) of xy. Clearly, when duplications are allowed from u to v, we might have multiple children of the same parent adjacency. Hence, for every adjacency $a_{par} = xy \in A_u$, we denote by F_a the set of all adjacencies $a = x'y' \in A_v$.

Further, using change variables $c_{u,v,a_{par}}$ and $c_{v,u,a_{par}}$ the number of rearrangements along the branch (u, v) can be computed as:

$$|A - D| = \sum_{a_{par} \in u} c_{u,v,a_{par}}$$

$$|D - A| = \sum_{a_{par} \in u} c_{v,u,a_{par}}$$

where

$$c_{u,v,a_{par}} = \max\{0, p_{u,a_{par}} - \sum_{a \in F_a} p_{v,a}\}$$

and

$$c_{v,u,a_{par}} = \max\{0, \sum_{a \in F_a} p_{v,a} - p_{u,a_{par}}\}$$

Let $K_v = \gamma \sum_{a \in v} (1 - p_{v,a}) w_{v,a}$. Since $\delta(u, v)$ are constant for a given pair (u, v), minimizing $d_{DSCJ}(u, v)$ is equivalent to minimizing $\sum_{a_{par} \in u} c_{u,v,a_{par}} + \sum_{a_{par} \in u} c_{v,u,a_{par}} - 2t(v) + 2 \sum_{g \in G_u} \alpha_g$. Thus, our problem reduces to minimizing:

$$\sum_{v \in V} \left\{ K_v + (1 - \gamma) \left(\sum_{a_{par} \in u} c_{u,v,a_{par}} + \sum_{a_{par} \in u} c_{v,u,a_{par}} + 2\delta(u, v) - 2t(v) + 2 \sum_{g \in G_u} \alpha_g \right) \right\}$$

subject to:

$$
\begin{aligned}
c_{u,v,a_{par}} &\geq 0 & \forall a_{par} \in u && (6.1)\\
c_{u,v,a_{par}} &\geq p_{u,a_{par}} - \sum_{a \in F_a} p_{v,a} & \forall a_{par} \in u && (6.2)\\
c_{v,u,a_{par}} &\geq 0 & \forall a_{par} \in u && (6.3)\\
c_{v,u,a_{par}} &\geq \sum_{a \in F_a} p_{v,a} - p_{u,a_{par}} & \forall a_{par} \in u && (6.4)\\
\sum_{a=(x_t,y)} p_{v,a} &\leq 1 & \forall x, \forall v && (6.5)\\
\sum_{a=(x_h,y)} p_{v,a} &\leq 1 & \forall x, \forall v && (6.6)
\end{aligned}
$$

The constraints (6.1-6.4) ensure that the values for $c_{u,v,a_{par}}$ and $c_{v,u,a_{par}}$ is chosen correctly according to the $p_{v,a}$ values. The constraints (6.5-6.6) ensure the consistency of the genome. In other words, they ensure that each extremity takes part in at most one adjacency.

6.3.1 Linearization of chromosomes

Using the above formulation, it is possible to obtain the exact optimal solution to the Small Parsimony Problem under the SCJ-TD-FD model. In general, the ILP solves the mixed version of the problem, which does not eliminate the possibility of circular chromosomes in the genomes. However, as mentioned in the problem statement, we require our solution to consist of linear chromosomes only. This requirement stems from the structure of mosquito genomes, which is known to be linear. In previously published methods [52], the problem of linearization of chromosomes has been addressed in a greedy manner. The method involves the removal of the least weighted adjacency contained in an existing circular chromosome. However, this approach does not guarantee a globally optimal solution. Moreover, removing an adjacency might introduce rearrangement events that, in turn can increase the DSCJ distance. Potentially, this can output a solution that is far from optimal.

Delayed constraint generation method: In a LP framework, the removal of a circular chromosome corresponds to the addition of a new constraint that prevents the chromosome in question from being circular. These constraints will be referred to as *linearity constraints*. An obvious way to obtain an optimal solution containing no circular chromosomes would be to introduce a linearity constraint each for all possible circular chromosomes. However, in doing so, we get an exponential number of constraints. Thus, introducing all the constraints at once will be too expensive, in terms of both time and space. Furthermore, it is possible that many of the constraints might be redundant and will never impact the feasibility region.

To overcome this issue, we use the *delayed constraint generation method* [10]. Typically, the use of this method involves a given linear programming (LP) instance with a large num-

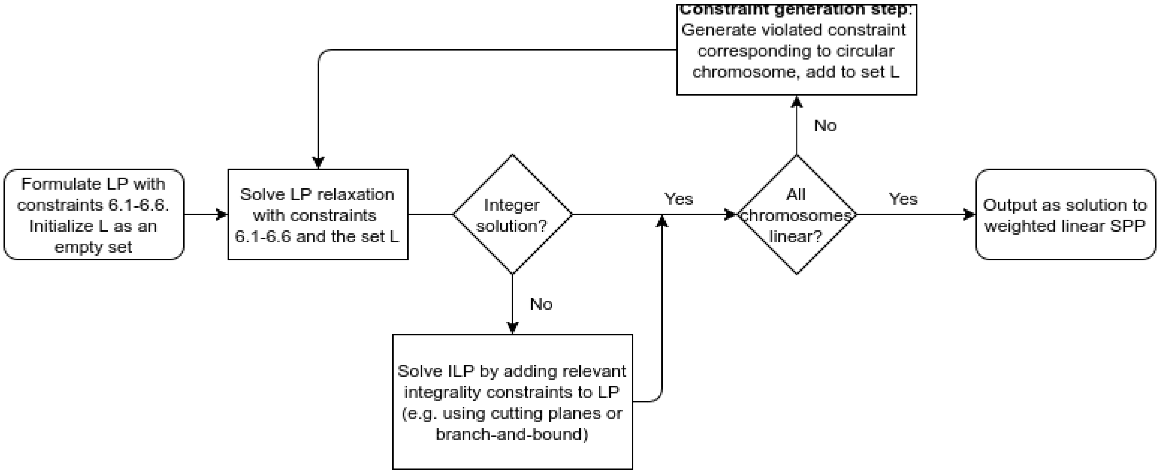

Constraint generation step: Generate violated constraint corresponding to circular chromosome, add to set L
No
Formulate LP with constraints 6.1-6.6. Initialize L as an empty set
Solve LP relaxation with constraints 6.1-6.6 and the set L
Integer solution?
Yes
All chromosomes linear?
Yes
Output as solution to weighted linear SPP
No
Solve ILP by adding relevant integrality constraints to LP (e.g. using cutting planes or branch-and-bound)

Figure 6.1: Process for solving the weighted linear SPP using constraint generation

ber of constraints. Instead of dealing with all the constraints at once, we initially introduce only a subset of the constraints. The problem defined only using this subset of constraints is called the *restricted* problem. In our case, the restricted problem consists only of the constraints (6.1-6.6) from the above formulation. Note that we may have a non-integer solution to the restricted problem. It may also contain circular chromosomes. If we add to the restricted problem, an integrality constraint for each variable, we can obtain the solution to the *mixed* version of the SPP (one that permits circular chromosomes).

For the weighted linear SPP, we start the initial optimization process with the restricted problem, using constraints (6.1-6.6) only. We add appropriate integrality constraints if required. However, the linearity constraints are left out. We initialize an empty set L of constraints. The set L is meant for linearity constraints to be generated when the corresponding circular chromosomes are chosen to be linearized. Let S be the optimal solution obtained from solving the restricted problem with the integrality constraints. If the solution S to the restricted problem consists only of linear chromosomes, then it is feasible for the original problem, with all the constraints. S is then reported as the optimal solution for the weighted linear SPP. However, if it does not, then S is infeasible, since it violates at least one linearity constraint. A circular chromosome from the current solution is then chosen at random. The corresponding constraint prevents the adjacencies involved to form the circular chromosome. This is accomplished simply by ensuring that the sum of the adjacency variables ($p_{v,a}$'s) is less than the length of chromosome itself. The new constraint is added to the set L. The restricted ILP is then solved along with constraints from the set L. This process is iterated till all chromosomes in the optimal solution are found to be linear. The process has been illustrated in Figure 6.1. The introduction of a linearity constraint is equivalent to the introduction of a hyperplane that possibly separates the optimal solution of the restricted problem from the feasibility region.

In the absence of integrality constraints, upon addition of the new constraint, the optimization process does not start from scratch. Instead, it uses the last solution (which violates the newly added constraint) as its starting solution. The corresponding solution for the dual of the original primal formulation is also feasible for the dual of the modified one. Thus, it is possible to use dual simplex method to the modified problem. The initial simplex tableau for the new problem can be constructed immediately constructed from the previous solution. However, this technique does not work in an ILP framework. First, the solution to the LP relaxation is found using the primal simplex method. In case of a non-integer optimal solution, the relevant integrality constraints are added using branch-and-bound or cutting plane method [42, 31]. The resulting integer solution may contain a circular chromosome. If so, the violated constraint is added to the LP relaxation and the problem is first optimized without the integrality constraints. As a result, the simplex tableau of the previous optimal solution (which contains some of the integrality constraints) can not be used as a starting point for the new optimization process.

During the process of iteratively removing circular chromosomes, a possible scenario in this approach is an exponential number of iterations. To avoid this, we make a provision to bound the number of iterations. Once, this bound is reached, the remaining circular chromosomes, if any, are handled using a greedy approach [52], in which the least weight adjacency from each circular chromosome is discarded, thus linearizing each genome.

We use Gurobi Optimizer to solve the ILP, which has the following advantage. Although the solver has to start the optimization process from scratch, the problem size is reduced by removing redundant constraints as well as variables, identification and merging of constraints that form cliques and appropriate rounding of bounds of integer variables. Such presolving techniques significantly improve the efficiency of the optimization process.

Remark 7. The prevention of circular chromosomes is similar to the removal of subtours in the Traveling Salesman Problem (TSP) [22]. Thus, each constraint preventing a circular chromosome is formed using the same logic as the *loop* or *subtour elimination* constraints for the TSP. There is, however, a slight difference in the two approaches. Consider the TSP on a set of nodes C. For the subtour elimination constraints, a set $C' \subseteq C$ is identified. The total number of edges $e = ij$, $i, j \in C'$ is restricted to $|C'| - 1$, thus preventing the possibility of *any* cycle involving all the elements of C'. On the other hand, for the prevention of circular chromosomes, we find that it suffices to add the constraint that prevents only the specific order of adjacencies that forms the observed circular chromosome.

6.4 Experimental results

To evaluate the performance of the ILP, we carried out experiments on a data set of Anopheles mosquito genomes, used in [2]. The code itself was written in Python and solved using Gurobi optimization solver, version 7.5.2.

Most of the data used in this project was produced by the Anopheles 16 Genomes project [61]. The species tree consists of 18 extant species, the gene orders of which are available to us through [2]. The phylogeny is based on the X-chromosome genes. The gene content and weighted adjacencies of ancestral genomes were also obtained through [2]. The weight $w_{v,a} \in [0,1]$ assigned to each adjacency indicates the confidence level that an adjacency a actually occurs in a genome v. The gene trees for each gene family were constructed in [2] using RaxML and refined using ProfileNJ. In total, this resulted in 14,940 gene trees, consisting of a total of 394195 genes, 183680 of which belong to extant genomes. *Anopheles gambiae* is the only fully assembled genome in the data set while the remaining genomes are assembled into scaffolds. The extant genomes in our data are highly fragmented with roughly 33500 scaffolds to start with. The mosquito genomes exhibit significant gene loss, with more than 10% genes over the course of evolution. In total, our ILP consisted of 3187979 integer variables, 2218787 of which were binary and 5850114 constraints. The matrix was highly sparse (sparsity of ~$5 * 10^{-7}$) with only 9115423 non-zero entries.

We then toggle the parameter $\alpha \in [0,1]$, starting at 0 and incrementing it by 0.25. Recall that the objective function of the ILP is a linear combination of the SCJ distance and cost of unselected adjacencies. Thus, $\alpha = 0$ corresponds to the case when the cost is measured purely in terms of the SCJ distance while $\alpha = 1$ ignores the SCJ distance completely. Our model does not yet handle gene losses. Hence, the event of gene loss affects the SCJ-TD-FD distance, which is mainly visible through the number of cuts resulting from gene losses.

Based on our experiments, we report the following observations. The number of adjacencies selected shows a increasing trend as α is increased. Thus, for $\alpha = 0$, the ILP selects around 290000 adjacencies from the available pool of 461605 adjacencies. This number increases to 350000 as α approaches 1. Our method is also meant to improve the assembly of extant genomes. Thus, we measured the number of scaffolds in the extant genomes after implementing the ILP. As α increases, a decreasing trend is observed in the number of scaffolds. Thus, the case $\alpha = 0$ yields the least improvement. However, even in this scenario, it produces significant improvement ($> 65\%$) with 11235 scaffolds. The cases $\alpha = 0.75$ and $\alpha = 1$ show the least number of scaffolds after the ILP with less than 8200 scaffolds (see Figure 6.2).

This is not surprising since the case $\alpha = 0$ prefers parsimony-guided reconstruction. Thus, it tries to generate the ancestral gene orders resulting in the lowest overall distance. On the other hand, $\alpha = 1$ is more likely to conserve adjacencies even if they result in a higher SCJ-TD-FD distance. Thus, the assembly of extant genomes gains more than 23000 adjacencies (15.78% of the initial number of adjacencies) when $\alpha = 1$, but while when $\alpha = 0$ the gain is less than 16500 (11.21%), as shown in Figure 6.3 (a).

The correlation between number of selected adjacencies and the corresponding SCJ-TD-FD distance can be observed by the contrasting trends in both parts of Figure 6.3. As α increases, the SCJ-TD-FD distance also increases. Interestingly, the number of cuts tends

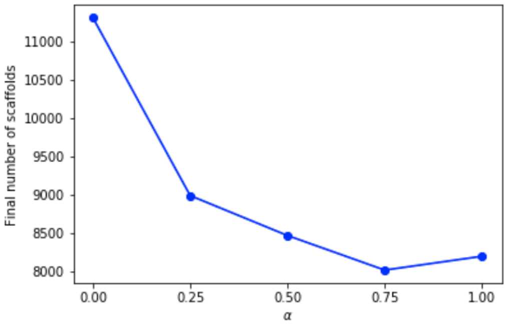
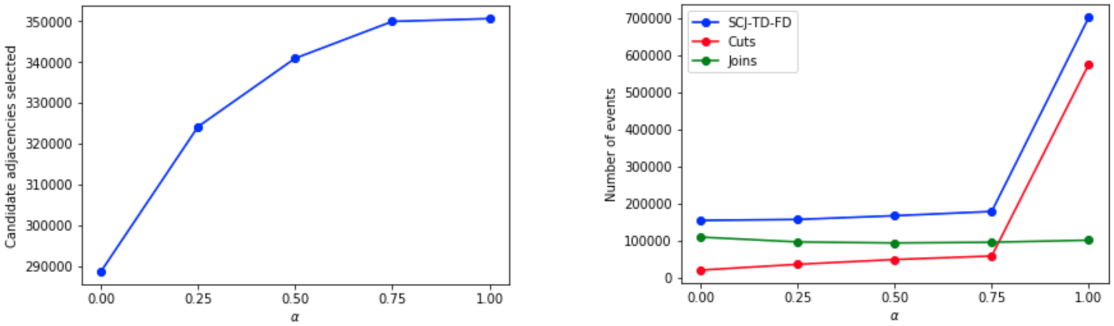


Figure 6.2: The final number of scaffolds in extant genomes, changing with respect to α



(a) Total number of selected adjacencies

(b) Total number of events

Figure 6.3: Inversion correlation between (a) total number of adjacencies chosen across all genomes and (b) the overall SCJ-TD-FD distance

to increase with α whereas the number of joins does not (see Fig. 6.3 (b)). This can be explained by the high rates of gene loss in the genomes. The ancestral genomes may involve adjacencies between genes that are lost in the descendant, leading to a higher ratio of cuts to joins.

Additionally, the ILP iteratively linearizes the circular chromosomes in all the genomes. We record the randomly selected circular chromosome for each iteration. We then add the corresponding constraint to remove the circular chromosome and re-optimize the updated ILP. Thus, the number of circular chromosomes linearized also gives the number of iterations. Even though the number of possible circular chromosomes may be exponential, our results clearly suggest that the number of chromosomes actually required to be linearized is not too large. As a result, an individual run in our experiments could be completed in under 8 minutes. A clear trend is not visible in the number of iterations required to linearize all the genomes. However, there is a marked increase in the number of iterations for $\alpha = 1$ which is expected since the case tends to select much more adjacencies in its solution than other values of α. Surprisingly, the time required for the optimization step in each iteration,

α	Cuts due to gene loss	#Circular chromosomes linearized	Avg. time (sec) per iteration
0	12088	9	24.11
0.25	24684	15	24.60
0.5	38899	13	23.70
0.75	50683	13	22.50
1	68280	46	6.12

Table 6.1: Statistics for the SPP ILP

when $\alpha = 1$, was close to 6 seconds whereas for all other cases the time per iteration was over 20 seconds.

6.5 Conclusion

In this chapter, we studied the Small Parsimony Problem that accounts for duplicate genes under the SCJ-TD-FD model. Following the results from chapter 5 on the NP-hardness of the rooted median problem, the SPP is also expected to be NP-hard. Hence, we provide an integer linear program for solving the problem following a similar approach used in [46], based on an optimality criterion that uses a linear combination of the SCJ-TD-FD distance and conserved adjacencies. Our experiments indicate a clear correlation between the number of conserved adjacencies and the associated SCJ-TD-FD distance. They also show significant improvements in the scaffolding of extant genomes with over 11% improvement even when the objective function is adjusted to focus solely on minimizing the SCJ-TD-FD distance. It is further observed that the number of circular chromosomes in the initial optimal solution is small enough and can be easily managed by the ILP through additional constraints.

Future research on the SCJ-TD-FD model could proceed in several directions. Mosquito genomes tend to display high rates of gene loss. The gene loss also has a substantial impact on the estimation of the SCJ-TD-FD distance. The handling of gene losses in the SCJ-TD-FD model is not immediate and thus, needs to be investigated further. Another potential line of inquiry is the weighting of ancestral adjacencies using the candidate adjacency selection technique from chapter 5. This approach may be used to choose candidate adjacencies in order to decrease the effect of gene loss on the overall objective function. Whereas this strategy may lead to fragmented assemblies for ancestral genomes, it is expected to provide a more reliable estimate of the distance. Finally, the multiplicity of co-optimal solutions and the dissimilarity between the various solutions is another avenue worth investigating.

Concluding remarks

In this thesis, we have discussed computational techniques to address some important genome rearrangement problems used for analyzing the evolution of genomes, in a duplication-aware framework.

The evolution of genomes through genome rearrangements was discovered in the 1930s. However, genome rearrangements problems were first presented as combinatorial problems by Sankoff in the early 1990s. This pioneered the research on genome rearrangements through various rearrangement models and techniques. The problem of computing the distance between two genomes is tractable under most models. However, in most cases, these tractability results are not extended to the median problem. The introduction of the SCJ model, a mechanistic counterpart of the breakpoint model, provided some positive results. Both the median problem and the SPP could be solved in polynomial time under the SCJ model.

The only downside of this model was that it works only when genome content is equal for both genomes. In chapter 4, we saw a solution to this problem. We introduced a model known as the SCJ-TD-FD model handling single gene duplications, cuts and joins. Each copy in the ancestor genome was assumed to be unique. Further the descendant contained at least one and possibly more copies from each gene in the ancestor. Secondly, we assumed that the duplication events take place only through specific mechanisms namely, tandem duplications and floating duplications (single-gene circular chromosomes). Given this setting, we proved that in the context of directed evolution from a trivial ancestor to a possibly non-trivial descendant, the SCJ-TD-FD can be computed in linear time.

This work is motivated by increasing availability of ancestral gene orders along a given species tree through algorithms that use reconciled gene trees to obtain the orthology relations for each gene family. This problem also acts an important stepping stone in the bigger picture of ancestral reconstruction techniques. The SCJ-TD-FD model allows the extension of the median and SPP problems to a duplication-aware framework.

In chapter 5, we discussed two variations of the median problem under the SCJ-TD-FD model. The directed median problems aims to compute the median of a set of k descendant genomes. It is proved that this can be achieved in polynomial time by redefining the problem as a maximum weight matching problem. On the other hand, the rooted median problem, which aims to find the median of an ancestor and k descendants is NP-hard. This has been proved by reduction from the 2P2N-3SAT problem. We provided an Integer Linear Program to solve the rooted median problem. The problem can be viewed as a colored maximum weight matching problem, wherein adjacencies are assigned colors in addition to weights. It is intended to compute a matching that maximizes the number of colors chosen in addition to the total weight of chosen edges.

Our experiments also indicated that the impact of convergent evolution, leading to conflicting adjacencies in the median, may be the main reason behind the hardness of the problem. It is also the reason behind multiple co-optimal solutions for the median. Our experiments on simulated data showed that the ILP outputs a reliable estimate of the median. The high accuracy of the reconstructed median genomes suggest that the ILP can also be used in iterative approaches to solve the Small Parsimony Problem.

In chapter 6, we discussed a more direct approach to solve the Small Parsimony Problem. Considering the NP-hardness of the rooted median problem, the SPP is also implied to be NP-hard. We provided an ILP to solve the problem, that accounts for duplication events. The ILP is motivated by the approach used in [46] that optimizes a linear combination conserved adjacencies and the SCJ-TD-FD distance. We applied our method on a data set consisting of Anopheles mosquito genomes, which consist of linear chromosomes only. The linearization is achieved iteratively by adding an extra constraint for a randomly chosen circular chromosome and re-optimizing. This is motivated by the constraint generation methods, which use the duality of the linear program and optimize the new ILP (with the extra constraint). Our results indicate that the assembly of the extant genomes is considerably improved by the ILP. As expected, the fragmentation of the genomes decreases when the parameter α is shifted to favor conservation of adjacencies over parsimony of genome rearrangement events.

The analysis of the three widely studied genome rearrangement problems presents various avenues for future research. The mosquito genomes exhibit high rates of gene loss which may impact the SCJ-TD-FD distance estimate. The incorporation of gene loss events in the current model remains to be investigated. The multiplicity and sampling of co-optimal scenarios for the SPP, in a similar manner as in [46], is also a possible future line of research. Lastly, the iterative use of the rooted median problem to solve the SPP, similar to [57], in a duplication-aware framework is also worth investigating.

Bibliography

[1] Max Alekseyev and Pavel Pevzner. Breakpoint graphs and ancestral genome reconstructions. *Genome Research*, 5:943–957, 2009.

[2] Yoann Anselmetti, Wandrille Duchemin, Eric Tannier, Cedric Chauve, and Sèverine Bérard. Phylogenetic signal from rearrangements in 18 anopheles species by joint scaffolding extant and ancestral genomes. *BMC Genomics*, 19(2):96, May 2018.

[3] Vineet Bafna and Pavel A. Pevzner. Genome rearrangements and sorting by reversals. *SIAM Journal on Computing*, 25(2):272–289, 1996.

[4] Anne Bergeron, Mathieu Blanchette, Annie Chateau, and Cedric Chauve. Reconstructing ancestral gene orders using conserved intervals. In *Algorithms in Bioinformatics, 4th International Workshop, WABI 2004, Bergen, Norway*, Lecture Notes in Computer Science, pages 14–25, 2004.

[5] Anne Bergeron, Julia Mixtacki, and Jens Stoye. A unifying view of genome rearrangements. In *Algorithms in Bioinformatics, 6th International Workshop, WABI 2006, Zurich, Switzerland*, Lecture Notes in Computer Science, pages 163–173, 2006.

[6] Anne Bergeron, Julia Mixtacki, and Jens Stoye. A new linear time algorithm to compute the genomic distance via the double cut and join distance. *Theor. Comput. Sci.*, 410(51):5300–5316, 2009.

[7] Anne Bergeron and Jens Stoye. On the similarity of sets of permutations and its applications to genome comparison. In *Computing and Combinatorics, 9th Annual International Conference, COCOON 2003, Big Sky, MT, USA*, Lecture Notes in Computer Science, pages 68–79, 2003.

[8] Piotr Berman, Marek Karpinski, and Alex D. Scott. Approximation hardness of short symmetric instances of MAX-3SAT. *Electronic Colloquium on Computational Complexity (ECCC)*, (049), 2003.

[9] John S. Bertram. The molecular biology of cancer. *Molecular Aspects of Medicine*, 21(6):167–223, 2000.

[10] Dimitris Bertsimas and John Tsitsiklis. *Introduction to Linear Optimization.* Athena Scientific, 1st edition, 1997.

[11] Priscila Biller, Laurent Guéguen, and Eric Tannier. Moments of genome evolution by Double Cut-and-Join. *BMC Bioinformatics*, 16(Suppl 14):S7, 2015.

[12] Marília D. V. Braga, Eyla Willing, and Jens Stoye. Double cut and join with insertions and deletions. *Journal of Computational Biology*, 18(9):1167–1184, 2011.

[13] David Bryant. A lower bound for the breakpoint phylogeny problem. *Journal on Discrete Algorithms*, 2(2):229–255, 2004.

[14] Alberto Caprara. Sorting by reversals is difficult. In *Proceedings of the First Annual International Conference on Research in Computational Molecular Biology, RECOMB 1997*, pages 75–83, 1997.

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

[16] Cedric Chauve, Jean-Philippe Doyon, and Nadia El-Mabrouk. Gene family evolution by duplication, speciation, and loss. *Journal of Computational Biology*, 15(8):1043–1062, 2008.

[17] Cedric Chauve, Nadia El-Mabrouk, Laurent Gueguen, Magali Semeria, and Eric Tannier. Duplication, rearrangement and reconciliation: A follow-up 13 years later. In *Models and Algorithms for Genome Evolution*, pages 47–62, 2013.

[18] David 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, 1998.

[19] Phillip E. C. Compeau. DCJ-Indel sorting revisited. *Algorithms for Molecular Biology*, 8:6, 2013.

[20] Karen D. Crow and Günter P. Wagner. What is the role of genome duplication in the evolution of complexity and diversity? *Molecular Biology and Evolution*, 23(5):887–892, 2006.

[21] Poly H. da Silva, Raphael Machado, Simone Dantas, and Marília D. V. Braga. DCJ-indel and DCJ-substitution distances with distinct operation costs. *Algorithms for Molecular Biology*, 8:21, 2013.

[22] George B. Dantzig, D. Ray Fulkerson, and Selmer M. Johnson. Solution of a large-scale traveling-salesman problem. *Operations Research*, 2(4):393–410, 1954.

[23] Adrian A. Davin, Theo Tricou, Eric Tannier, Damien M. de Vienne, and Gergely J. Szöllosi. ZOMBI: A simulator of species, genes and genomes that accounts for extinct lineages. *bioRxiv*, 2018.

[24] Theodosius Dobzhansky and Alfred H. Sturtevant. Inversions in the chromosomes of drosophila pseudoobscura. *Genetics*, 23(1):28–64, 1938.

[25] Wandrille Duchemin, Yoann Anselmetti, Murray Patterson, Yann Ponty, Sèverine Bérard, Cedric Chauve, Céline Scornavacca, Vincent Daubin, and Eric Tannier. DeCoSTAR: Reconstructing the ancestral organization of genes or genomes using reconciled phylogenies. *Genome biology and evolution*, 9(5):1312–1319, May 2017.

[26] Evan Eichler and David Sankoff. Structural dynamics of eukaryotic chromosome evolution. *Science*, 301:793–797, 2003.

[27] Pedro Feijão, Aniket C. Mane, and Cedric Chauve. A tractable variant of the Single Cut or Join distance with duplicated genes. In *Comparative Genomics - 15th International Workshop, RECOMB CG 2017*, volume 10562 of *Lecture Notes in Computer Science*, pages 14–30. Springer, 2017.

[28] Pedro Feijão and João Meidanis. SCJ: A breakpoint-like distance that simplifies several rearrangement problems. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(5):1318–1329, 2011.

[29] Guillaume Fertin, Anthony Labarre, Irena Rusu, Eric Tannier, and Stéphane Vialette. *Combinatorics of Genome Rearrangements.* Computational molecular biology. MIT Press, 2009.

[30] Walter M. Fitch. Toward defining the course of evolution: Minimum change for a specific tree topology. *Systematic Biology*, 20(4):406–416, 1971.

[31] Ralph E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64(5):275–278, 1958.

[32] Morris Goodman, John Czelusniak, G. William Moore, A. E. Romero-Herrera, and Genji Matsuda. Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences. *Systematic Biology*, 28(2):132–163, 1979.

[33] Pawel Górecki, J. Gordon Burleigh, and Oliver Eulenstein. Maximum likelihood models and algorithms for gene tree evolution with duplications and losses. volume 12, page S15, 2011.

[34] Matthew W Hahn, Mira V Han, and Sang-Gook Han. Gene family evolution across 12 drosophila genomes. *PLOS Genetics*, 3(11):1–12, 11 2007.

[35] Sridhar Hannenhalli and Pavel A. 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*, pages 178–189, 1995.

[36] Edwin Jacox, Cedric Chauve, Gergely J. Szöllosi, Yann Ponty, and Céline Scornavacca. ecceTERA: comprehensive gene tree-species tree reconciliation using parsimony. *Bioinformatics*, 32(13):2056–2058, 2016.

[37] John D. Kececioglu and David Sankoff. Exact and approximation algorithms for the inversion distance between two chromosomes. In *Combinatorial Pattern Matching, 4th Annual Symposium, CPM 93*, Lecture Notes in Computer Science, pages 87–105, 1993.

[38] John D. Kececioglu and David Sankoff. Exact and approximation algorithms for sorting by reversals, with application to genome rearrangement. *Algorithmica*, 13(1/2):180–210, 1995.

[39] Fyodor A. Kondrashov. Gene duplication as a mechanism of genomic adaptation to a changing environment. *Proceedings of the Royal Society of London B: Biological Sciences*, 279(1749):5048–5057, 2012.

[40] Victor Kunin and Christos Ouzonis. The balance of driving forces during genome evolution in prokaryotes. *Genome research*, 13:1589–1594, 2003.

[41] Ralf Küppers and Riccardo Dalla-Favera. Mechanisms of chromosomal translocations in B cell lymphomas. *Oncogene*, 10:5580–5594, 2001.

[42] Ailsa H. Land and Alison G. Doig. An automatic method for solving discrete programming problems. In *50 Years of Integer Programming 1958-2008 - From the Early Years to the State-of-the-Art*, pages 105–132. 2010.

[43] Allison N. Lau, Lei Peng, Hiroki Goto, Leona Chemnick, Oliver A. Ryder, and Kateryna D. Makova. Horse domestication and conservation genetics of Przewalski's horse inferred from sex chromosomal and autosomal sequences. *Molecular Biology and Evolution*, 26(1):199–208, 2009.

[44] Anthony Levasseur and Pierre Pontarotti. The role of duplications in the evolution of genomes highlights the need for evolutionary-based approaches in comparative genomics. *Biology Direct*, 6(1):11, 2011.

[45] László Lovász and Michael Plummer. *Matching Theory*, volume 29 of *Annals of Discrete Mathematics, Amsterdam: North Holland.* 1986.

[46] Nina Luhmann, Manuel Lafond, Annelyse Thèvenin, Aïda Ouangraoua, Roland Wittler, and Cedric Chauve. The SCJ small parsimony problem for weighted gene adjacencies. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, early access, 2017.

[47] Michael Lynch and John S. Conery. The evolutionary fate and consequences of duplicate genes. *Science*, 290(5494):1151–1155, 2000.

[48] Jian Ma, Aakrosh Ratan, Brian J. Raney, Bernard B. Suh, Louxin Zhang, Webb Miller, and David Haussler. DUPCAR: reconstructing contiguous ancestral regions with duplications. *Journal of Computational Biology*, 15(8):1007–1027, 2008.

[49] Jian Ma, Louxin Zhang, Bernard B. Suh, Brian J. Raney, Richard Burhans, W. James Kent, Mathieu Blanchette, David Haussler, and Webb Miller. Reconstructing contiguous regions of an ancestral genome. *Genome research*, 16:1557–1565, 2006.

[50] Christopher A. Makaroff and Jeffrey D. Palmer. Mitochondrial DNA rearrangements and transcriptional alterations in the male-sterile cytoplasm of ogura radish. *Molecular and cellular biology*, 8:1474–1480, 1988.

[51] Aniket C. Mane, Manuel Lafond, Pedro Feijão, and Cedric Chauve. The rooted scj median with single gene duplications. In *Comparative Genomics - 16th International Workshop, RECOMB CG 2018 Proceedings*, 2018.

[52] Ján Maňuch, Murray Patterson, Roland Wittler, Cedric Chauve, and Eric Tannier. Linearization of ancestral multichromosomal genomes. *BMC Bioinformatics*, 13(19):S11, 2012.

[53] Fábio Viduani Martinez, Pedro Feijão, Marília D. V. Braga, and Jens Stoye. On the family-free DCJ distance. pages 174–186, 2014.

[54] István Miklós, Sándor Z. Kiss, and Eric Tannier. Counting and sampling SCJ small parsimony solutions. *Theoretical Computer Science*, 552:83–98, 2014.

[55] Ray Ming, Robert VanBuren, Ching M. Wai, et al. The pineapple genome and the evolution of CAM photosynthesis. *Nature Genetics*, 47(12):1435–1442, 2015.

[56] Alex Mira, Howard Ochman, and Nancy A. Moran. Deletional bias and the evolution of bacterial genomes. *Trends in Genetics*, 17(10):589 – 596, 2001.

[57] Bernard Moret, Stacia Wyman, David Bader, Tandy Warnow, and Mi Yan. A new implementation and detailed study of breakpoint analysis. In *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 583–594. 2001.

[58] Bernard M. E. Moret, Yu Lin, and Jijun Tang. Rearrangements in phylogenetic inference: Compare, model, or encode? In *Models and Algorithms for Genome Evolution*, pages 3147–171. 2013.

[59] Bernard M.E. Moret and Tandy Warnow. Advances in phylogeny reconstruction from gene order and content data. In *Molecular Evolution: Producing the Biochemical Data*, volume 395 of *Methods in Enzymology*, pages 673 – 700. Academic Press, 2005.

[60] Joseph Nadeau and Benjamin Taylor. Lengths of chromosomal segments conserved since divergence of man and mouse. *Proceedings of the National Academy of Sciences, USA*, 81:814–818, 1984.

[61] Daniel E. Neafsey, Robert M. Waterhouse, Mohammad R. Abai, et al. Highly evolvable malaria vectors: The genomes of 16 Anopheles mosquitoes. *Science*, 347(6217), 2015.

[62] Jeffrey Palmer and Laura Herbon. Plant mitochondrial DNA evolves rapidly in structure, but slowly in sequence. *Journal of Molecular Evolution*, 28:87–97, 1988.

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

[64] Pavel Pevzner and Glenn Tesler. Genome rearrangements in mammalian evolution: Lessons from human and mouse genomes. *Genome Research*, 13:37–45, 2003.

[65] Lovász L. Plummer M.D. *Matching Theory.* Elsevier, 1986.

[66] Marta Puig, Sónia Casillas, Sergi Villatoro, and Mario Cáceres. Human inversions and their functional consequences. *Briefings in Functional Genomics*, 14(5):369–379, 2015.

[67] Ashok Rajaraman and Jian Ma. Reconstructing ancestral gene orders with duplications guided by synteny level genome reconstruction. *BMC Bioinformatics*, 17(S-14):201–212, 2016.

[68] Matthew D. Rasmussen and Manolis Kellis. A bayesian approach for fast and accurate gene tree reconstruction. *Molecular Biology and Evolution*, 28(1):273–290, 2011.

[69] Antonis Rokas and Peter W.H. Holland. Rare genomic changes as a tool for phylogenetics. *Trends in Ecology and Evolution*, 15(11):454–459, 2000.

[70] Diego P. Rubert, Pedro Feijão, Marília Dias Vieira Braga, Jens Stoye, and Fábio Viduani Martinez. Approximating the DCJ distance of balanced genomes in linear time. *Algorithms for Molecular Biology*, 12(1):3:1–3:13, 2017.

[71] Michael J. Sanderson and Michelle M. McMahon. Inferring angiosperm phylogeny from est data with widespread gene duplication. *BMC Evolutionary Biology*, 7:273–290, 2007.

[72] David Sankoff. Genome rearrangement with gene families. *Bioinformatics*, 15(11):909–917, 1999.

[73] David Sankoff, Robert Cedergren, and Yvon Abel. Genomic divergence through gene rearrangement. *Molecular Evolution: Computer Analysis of Protein and Nucleic Acid Sequences*, 26:428–438, 1990.

[74] David Sankoff and Nadia El-Mabrouk. Duplication, rearrangement, and reconciliation. *Comparative Genomics: Computational Biology*, 1, 2000.

[75] David Sankoff, Guillaume Leduc, Natalie Antoine, Bruno Paquin, B. Franz Lang, and Robert Cedergren. Gene order comparisons for phylogenetic inference: evolution of the mitochondrial genome. *Proceedings of the National Academy of Sciences*, 89(14):6575–6579, 1992.

[76] David Sankoff, Gopal Sundaram, and John D. Kececioglu. Steiner points in the space of genome rearrangements. *International Journal of Foundations of Computer Science*, 7(1):1–9, 1996.

[77] David Sankoff and Phil Trinh. Chromosomal breakpoint reuse in genome sequence rearrangement. *Journal of Computational Biology*, 12(6):812–821, 2005.

[78] João Carlos Setubal and João Meidanis. *Introduction to computational molecular biology.* PWS Publishing Company, 1997.

[79] Mingfu Shao, Yu Lin, and Bernard M. E. Moret. An exact algorithm to compute the DCJ distance for genomes with duplicate genes. In *Research in Computational Molecular Biology - 18th Annual International Conference, RECOMB 2014*, Lecture Notes in Computer Science, pages 280–292, 2014.

[80] Mingfu Shao and Bernard M. E. Moret. A fast and exact algorithm for the exemplar breakpoint distance. *Journal of Computational Biology*, 23(5):337–346, 2016.

[81] Mingfu Shao and Bernard M. E. Moret. On computing breakpoint distances for genomes with duplicate genes. *Journal of Computational Biology*, 24(6):571–580, 2017.

[82] Krister M. Swenson, Vaibhav Rajan, Yu Lin, and Bernard M. E. Moret. Sorting signed permutations by inversions in $O(n\log n)$ time. *Journal of Computational Biology*, 17(3):489–501, 2010.

[83] Eric Tannier, Chunfang Zheng, and David Sankoff. Multichromosomal median and halving problems under different genomic distances. *BMC Bioinformatics*, 10, 2009.

[84] Hall T.E. Watterson G.A., Ewens W.J. and Morgan A. The chromosome inversion problem. *Journal of Theoretical Biology*, 99:1–7, 1982.

[85] Zhexue Wei, Daming Zhu, and Lusheng Wang. A dynamic programming algorithm for (1, 2)-exemplar breakpoint distance. *Journal of Computational Biology*, 22(7):666–676, 2015.

[86] Sophia Yancopoulos, Oliver Attie, and Richard Friedberg. Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinformatics*, 21(16):3340–3346, 2005.

[87] JJ Yunis and O Prakash. The origin of man: a chromosomal pictorial legacy. *Science*, 215(4539):1525–1530, 1982.

[88] Ron Zeira and Ron Shamir. Sorting by cuts, joins, and whole chromosome duplications. *Journal of Computational Biology*, 24(2):127–137, 2017.

[89] J. Zhang. Evolution by gene duplication: an update. *Trends in Ecology and Evolution*, 18:292–298, 2003.

[90] Chunfang Zheng and David Sankoff. On the PATHGROUPS approach to rapid small phylogeny. *BMC Bioinformatics*, 12(S-1):S4, 2011.