DCJ Median Problems on Linear Multichromosomal Genomes: Graph Representation and Fast Exact Solutions

Andrew Wei Xu

School of Computer and Communication Sciences
Swiss Federal Institute of Technology (EPFL)
EPFL IC LCBB, Station 14
CH-1015 Lausanne, Switzerland
wei.xu@epfl.ch

Abstract. Given a set of genomes $\mathcal{G}$ and a distance measure d, the genome rearrangement median problem asks for another genome q that minimizes $\sum_{g \in \mathcal{G}} d(q, g)$. This problem lies at the heart of phylogenetic reconstruction from rearrangement data, where solutions to the median problems are iteratively used to update genome assignments to internal nodes for a given tree. The median problem for reversal distance and DCJ distance is known to be NP-hard, regardless of whether genomes contain circular chromosomes or linear chromosomes and of whether extra circular chromosomes is allowed in the median genomes. In this paper, we study the relaxed DCJ median problem on linear multichromosomal genomes where the median genomes may contain extra circular chromosomes; extend our prior results on circular genomes—which allowed us to compute exact medians for genomes of up to 1,000 genes within a few minutes. First we model the DCJ median problem on linear multichromosomal genomes by a *capped multiple breakpoint graph*, a model that avoids another computationally difficult problem—a multi-way capping problem for linear genomes, then establish its corresponding decomposition theory, and finally show its results on genomes with up to several thousand genes.

1 Introduction

Genomes of related species contain large numbers of homologous DNA sequences, including protein-coding genes, noncoding genes, and other conserved genetic units. In the following, we shall use the term *genes* to refer to all of these sequences. These genes typically appear in different orders in different genomes, as a result of evolutionary events that rearranged the gene orders. These gene orders can thus be used to infer phylogenetic relationships; in the process, they may also be used to attempt reconstruction of ancestral gene orders.

We assume that each genome contains the same set of genes and that no gene is duplicated, so that the genes can be represented by integers, each chromosome by a sequence of integers (if a chromosome is circular, this sequence is viewed as circular), and each genome by a collection of such sequences. The integers representing genes have a sign, which denotes the strand on which the genetic information is read; we

F.D. Ciccarelli and I. Miklós (Eds.): RECOMB-CG 2009, LNBI 5817, pp. 70–83, 2009.

assume that the sign of each gene is known. A genome is linear if it only contains linear chromosome, circular otherwise. When a genome contains one chromosome, we call it unichromosomal; otherwise, we call it multichromosomal.

A breakthrough in the study of genome rearrangements was the characterization of the mathematical structure of reversals (usually called inversions by biologists) and the first polynomial-time algorithm to compute a shortest series of reversals to transform one genome into another, whether unichromosomal [5] or multichromosomal [6]. Later work yielded an optimal linear-time algorithm to compute the reversal distance [2] and a $O(n \log n)$ algorithm to sort almost all permutations by reversals [9]. While reversals have been documented by biologists since the 1930s (in the pioneering work of the fly geneticists Sturtevant and Dobzhansky), the most studied operator in the last few years is a mathematical construct that unifies reversals with translocations, fusions, and fissions, and can implement any transposition in two moves, the double-cut-and-join (DCJ) operation [16]. If we let n be the number of genes, χ denote the number of linear chromosomes, and c, p_e, and p_o denote the numbers of cycles, even-sized paths, and odd-sized paths, respectively, in the breakpoint graph between two genomes, the DCJ distance between these genomes is given by the formula:

$$d_{DCJ} = n + \chi - c - p_e - \frac{p_o}{2}. \tag{1}$$

1.1 The DCJ Median Problem

The median problem for genome rearrangements is defined as follows: given a set $\mathcal{G}$ of genomes and a distance measure d, find a genome q that minimizes $\sum_{g \in \mathcal{G}} d(q, g)$. This problem is central to both phylogenetic reconstruction using gene-order data and ancestral reconstruction of gene orders. The median problem is known to be NP-hard under most rearrangement distances proposed to date, such as breakpoint, reversal, and DCJ distances; except when the median genome can contain extra circular chromosomes, the problem becomes polynomial under the breakpoint distance [11].

DCJ operations may create intermediate genomes with one or more extra circular chromosomes in addition to the original collection. We will refer to *relaxed* and *strict* versions of the DCJ median problem depending on whether such extra circular chromosomes are, respectively, allowed or forbidden. Our focus here is on the relaxed version, in part because of its simplicity and in part because, as seen in our experimental results, the average number of extra circular chromosomes does not exceed 0.5, so that solutions for the relaxed version are frequently also solutions for the strict version and those that are not can be transformed (by merging the circular chromosomes) into good approximate solutions to the strict version.

The median problem is NP-hard for the DCJ distance, for both strict and relaxed versions, on both circular and linear genomes [4,11]. There are exact algorithms for the reversal median problem and the strict DCJ median problem [4,8,17], but limited to small sizes; and there are also heuristics [3,1,7,10] of varying speed and accuracy.

In previous work [15,14], we developed an decomposition approach to the relaxed DCJ median problem on circular genomes. We used the *multiple breakpoint graph* (MBG) [4] to model the median problem; and the decomposition relies on the concept

of *adequate subgraphs* of the MBG. We proved that such subgraphs enable a divide-and-conquer approach in which the MBG is decomposed, optimal solutions found recursively for its parts, and then optimal solutions for the complete instance created by combining these optimal solutions. We also showed that there are infinitely many types of adequate subgraphs [14], among which those of small sizes immediately tell us which adjacencies should exist in the median genomes. Applying this decomposition method to simulated data, we showed that instances with up to 1,000 genes with moderate numbers of rearrangement events ($\leq 0.9n$) can be solved in a few minutes.

1.2 Contributions and Presentation of Results

In this paper, we consider the relaxed DCJ median problem on linear multichromosomal genomes with equal or unequal numbers of chromosomes. Compared to its circular counterpart [15] this problem introduces a multi-way capping problem. Caps are used to delimit the ends of linear chromosomes; for a genome with χ linear chromosomes, there are $(2\chi)!$ different ways to label its 2χ caps. For a median problem containing three genomes with χ linear chromosomes each, one of these genomes can have fixed consecutive labels from 1 to 2χ, while each of the other two genomes needs to pick a labeling from $(2\chi)!$ possibilities each, for a total of $\big((2\chi)!\big)^2$ choices. For $\chi = 23$ (as in the human genome), this number is 3×10^{115}. The optimal choice is assumed to be one that yields a minimum number of evolutionary events for the median problem.

We introduce the *capped multiple breakpoint graph* (CMBG) to model the DCJ median problem on linear genomes, where a single node is used to represent all caps, following the idea in the *flower graph* [13]. In so doing, we completely avoid the capping problem. When the median genome is obtained, the optimal capping can be determined by solving pairwise capping problems between the median and each of the given genomes—and optimal pairwise cappings can be determined in time linear in the number of genes [16].

The rest of this paper is organized as follows. In Section 2, we briefly review basic concepts and results from [15,14] for the circular case. We defined the capped multiple breakpoint graph in Section 3 and its *regular adequate subgraphs* and *capped adequate subgraphs* in Section 4, where we also listed the most frequent adequate subgraphs. In Section 5 we present algorithm ASMedian-linear (similar to the one for the circular case), which finds exact solutions by iteratively detecting the existence of adequate subgraphs on CMBGs. In section 6, we present the results of tests on simulated data with varying parameters.

2 Basic Definitions and Previous Results on Circular Genomes

For the median problem on circular genomes, the breakpoint graph extends naturally to a *multiple breakpoint graph* (MBG) [4]. We call the number of given genomes $N_{\mathcal{G}}$ the *rank* of an MBG. We label the given genomes and the edges representing their adjacencies by integers from 1 to $N_{\mathcal{G}}$. Finally we define the *size* of an MBG or its subgraph as half the number of vertices it contains.

By a 1-edge, 2-edge or 3-edge, we mean an edge with colour 1, 2 or 3. Since edges of the same colour form a perfect matching of the MBG, we use i-matching to denote

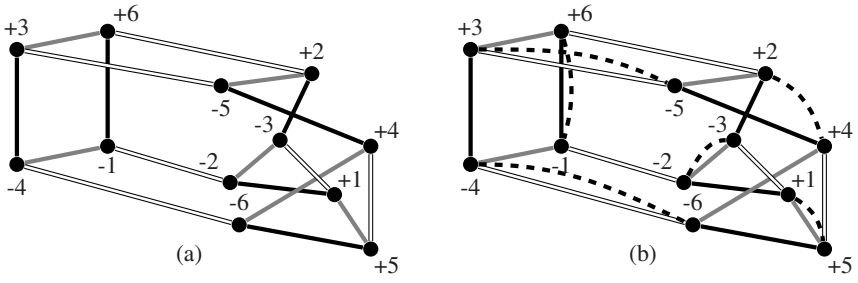

Fig. 1. Multiple breakpoint graph and median graph. Black, gray, double and dashed edges denote edges with colours 1, 2, 3 and 0 correspondingly. (a) A multiple breakpoint graph of three circular genomes, (1 2 3 4 5 6), (1 -5 -2 3 -6 -4) and (1 3 5 -4 6 -2) (b) A median graph with the median genome (1 -5 -3 2 -4 6). (and) are used to indicate the chromosomes are circular.

all edges of colour i, where $1 \leq i \leq 3$. For a candidate median genome, we use a different colour for its adjacency edges, namely colour 0; similarly we have 0-edges and 0-matchings. Adding such a candidate matching to the MBG results in the *median graph*. The set of all possible candidate matchings is denoted by $\mathcal{E}$.

The 0-i *cycles* in a median graph with a 0-matching E, numbering $c_{0,i}$ in all, are the cycles where 0-edges and i-edges alternate. Let $c_E^{\Sigma} = \sum_{1 \leq i \leq 3} c_{0,i}$. Then $c_{\max}^{\Sigma} = \max\{c_E^{\Sigma} : E \in \mathcal{E}\}$ is the maximum number of cycles that can be formed from the MBG. For circular genomes, since the DCJ distance is determined by the number of cycles in the induced breakpoint graph, we have:

Lemma 1. *[15] Minimizing the total DCJ distance in the median problem on circular genomes is equivalent to finding an optimal 0-matching E, i.e., with $c_E^{\Sigma} = c_{\max}^{\Sigma}$.*

A connected MBG subgraph H of size m is an *adequate subgraph* if $c_{\max}^{\Sigma}(H) \geq \frac{1}{2}mN_{\mathcal{G}}$; it is *strongly adequate* if $c_{\max}^{\Sigma}(H) > \frac{1}{2}mN_{\mathcal{G}}$. For the median of three problem where the rank is $N_{\mathcal{G}} = 3$, an adequate subgraph is a subgraph with $c_{\max}^{\Sigma}(H) \geq \frac{3m}{2}$ and a strongly adequate subgraph is one with $c_{\max}^{\Sigma}(H) > \frac{3m}{2}$.

The existence of an adequate subgraphs on an MBG gives a proper decomposition of the MBG into two subproblems, where the optimal solution of the original problem can be found by combing solutions from the two subproblems, as stated in the following theorem.

Theorem 1. *[15]*[1] *The existence of an adequate subgraph gives a proper decomposition from which* an *optimal solution can be found by combining solutions from the two subproblems. The existence of a strongly adequate subgraph gives a proper decomposition from which* all *optimal solutions can be found by combining solutions from the two subproblems.*

Remark 1. Subproblems induced by adequate subgraphs of small sizes are easy to solve. When we only use these small adequate subgraphs (as our algorithms do), we

[1] This theorem has been rephrased, in order to avoid the concept of a *decomposer*, whose definition requires several other concepts.

can encode their solutions into algorithms, so that whenever their existences are detected, adjacencies representing solutions of these subproblems are immediately added into the median genome.

An intuitive understanding of this theorem follows from the definition of an adequate subgraph. For any subgraph H of size m, the theoretical largest value for $c^{\Sigma}_{max}(H)$ is $mN_{\mathcal{G}}$, which is only possible when edges of different colours coincide. The ability for an adequate subgraph to form half the maximum number or more color alternating cycles, indicates that the decomposition by this subgraph is optimal, as it probably forms more cycles than any other alternative choice.

3 Capped Multiple Breakpoint Graph—A Graph Representation of the Median Problem on Linear Multichromosomal Genomes

In the rest of the paper we study the relaxed DCJ median problem on linear multichromosomal genomes. Here in this section, we introduce its first graph representation. The idea follows a *flower graph* [13], a variant of the breakpoint graph on linear genomes with a single *cap* node delimiting all ends of linear chromosomes. But first let us quickly review the traditional model of using a pair of caps for each linear chromosome [6,12], and show why the problem is simplified when the DCJ distance measure is used.

The process of adding caps is called *capping*. For a pair of genomes with χ linear chromosomes, there are $(2\chi)!$ different ways of capping. Different cappings may lead to different breakpoint graphs, and hence different pairwise distances. The capping problem for two genomes is to identify pairs of telomeres, one telomere from each genome, evolving from the same telomeres in their common ancestor genome. This orthologous relationship is represented, presumably, by an optimal capping, the one giving the smallest distance. Given an optimal capping, since vertices (representing endpoints of genes or telomeres) are incident to two adjacencies, one from each genome, the induced breakpoint graph consists of $c' = c + p_e + \frac{p_o}{2}$ color alternating cycles. As for circular genomes the DCJ distance is determined by its number of cycles c, for linear genomes the DCJ distance is determined by the number of cycles c', in the breakpoint graph induced by an optimal capping. For pairwise genomes under the DCJ distance, optimal cappings can be easily determined [16].

The traditional model of using a pair of caps for each linear chromosome lead to different induced breakpoint graphs, which are all equivalent in the amount of information they carry. We proposed in [13] a succinct graph representation, namely a *flower graph*, in which by allowing cycles and paths to be freely arranged, caps are merged into a single node. Flower graph gives a unique graph representation for pairs of linear genomes, as illustrated by Fig. 2.(a). In determining the pairwise DCJ distance, the significance of introducing the flower graph is limited to giving a mathematically succinct and unique graph representation; an optimal capping is easy to find and its induced breakpoint graph can be thought of as the representative breakpoint graph. However, when there are more than two genomes, such as in the median problem, optimal cappings may be hard to find; any procedure that first requires to find optimal cappings can be computationally very costly. As for the median problem with three given linear multichromosomal genomes, there are $((2\chi)!)^2$ ways of capping; as each capping induces

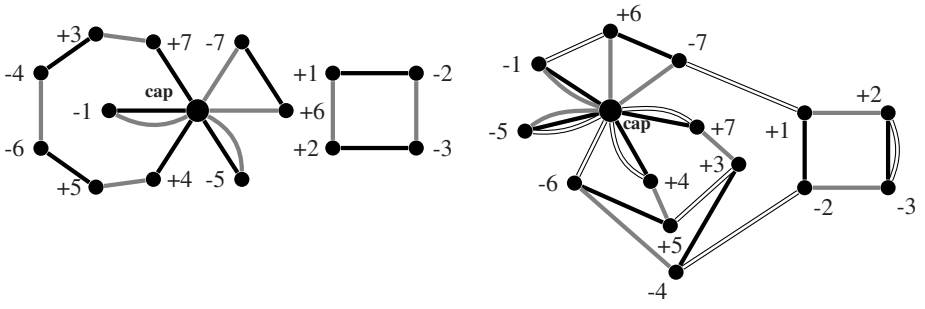

Fig. 2. (a) Flower graph for genomes {1 2 3 4; 5 6 7} and {1 -2 3 -7; 5 -4 6}, where we use black and gray edges to distinguish adjacency edges from different genomes. We have $n = 7, \chi = 2, c = 1, p_o = 2, p_e = 2$, so their DCJ distance is 5. (b) Capped multiple breakpoint graph for genomes {1 2 3 4; 5 6 7}, {1 -2 3 -7; 5 -4 6} and {5 -3 -2 4; 6 1 7}. Black, gray and double edges represent edges of colours 1 2 and 3 correspondingly. The cap is incident to 6 edges.

a different multiple breakpoint graph and an equivalent instance of the median problem on circular genomes, to find optimal cappings one would need to solve $((2\chi)!)^2$ instances of the median problem on circular genomes.

To model the median problem on linear multichromosomal genomes, following the idea of flower graphs and multiple breakpoint graphs, we propose the *capped multiple breakpoint graph* (CMBG). A CMBG is constructed as follows: each gene is represented by a pair of ordered vertices, a single node named *the cap* is added to delimit ends of all linear chromosomes; adjacencies between genes are represented by coloured edges connecting their corresponding endpoints, and for genes residing at ends of chromosomes, coloured edges are added connecting the cap and their endpoints. Edges representing adjacencies from the same genome are labeled with the same colour. For an instance of the median of three problem, where each genome contains n genes and χ linear chromosomes, the corresponding CMBG has $2n$ regular vertices of degree 3 each, and the cap vertex of degree 6χ. Fig.2.(b) shows a CMBG for the median of three problem with genomes {1 2 3 4; 5 6 7}, {1 -2 3 -7; 5 -4 6} and {5 -3 -2 4; 6 1 7}.

When the genomes contain different numbers of linear chromosomes, a few null chromosomes are added to equalize the number of chromosomes. A null chromosome consists of two telomeres and no genes; in CMBG, they correspond to edges looping around the cap. When the context is clear, these looping edges can be omitted from the graph.

The definition of the *size* of an MBG as half the number of vertices no longer applies to a CMBG or its subgraphs, as the cap node actually represents a number of telomeres—counting it as one vertex does not reflect the actual number of its adjacent edges. In this case for a CMBG or its subgraph, the *size* is defined as the total number of 0-edges to be added. For a subgraph of a CMBG, it is possible to have different interpretations of its size; however this does not impose much difficulty in defining *adequate subgraphs* for a CMBG—the definition holds if the requirement is satisfied by any interpretation and furthermore such an interpretation is always unique and obvious.

Similar to Lemma 1 we have the following statement for the median problem on linear multichromosomal genomes.

Lemma 2. *Finding the DCJ median on genomes with n genes and χ linear chromosomes is equivalent to finding a set of $n + \chi$ 0-edges for the capped multiple breakpoint graph, satisfying the following properties:*

1. *each regular vertex is incident to E exactly once;*
2. *the cap node is incident to E exactly 2χ times;*
3. *E maximizes the total number of cycles $c'^{\Sigma}_{max} = \max\{\sum_{1\leq i\leq 3} c'_{0,i}$: for all possible sets of 0-edges$\}$, where $c'_{0,i}$ is equal to the quantity $c + p_e + \frac{p_o}{2}$ between a candidate median genome whose adjacencies represented by a set of 0-edges and the ith given genome.*

Using a single cap node to represent all telomeres for the median problem is more than just for a succinct graph representation; it completely avoids the computationally costly capping problem and allows us to identify the orthologous relationships for telomeres in different genomes in the process of constructing the median genome.

Meanwhile this new representation poses new challenges in finding the median genome; compared to the counterpart problem for circular genomes, the existence of the cap node requires special considerations. The first problem arises in representing the problem when part of the median genome is known; the second problem is about finding the median genome efficiently.

When a partial solution is known for the problem on circular genomes, we can perform shrink operations[4,15]; the resultant multiple breakpoint graph has a smaller size but represents the same problem, which we can further decompose into smaller problems. However on the CMBG, if the 0-edge is incident to the cap node (we call it a capped 0-edge), the shrink operation is not defined as the cap node is incident to many edges and we do not know which edges to choose in order to perform such a shrink operation. One choice, as used in the current paper, is to just keep these capped 0-edges, although this will bring some complications to the graph representation and the implementation of data structures for the algorithms. In the full version of this paper, we will introduce an elegant graph representation with another node called the "cup" which collects the vertices incident to the capped 0-edges; the resultant graphs also consist of regular vertices, one cap node, one cup node and non-0-edges.

To improve the efficiency in finding the median genome, we need to increase the frequency of decompositions carried on the CMBGs. The adequate subgraphs defined in [15] apparently do not apply to any subgraph with the cap node; we need to establish parallel theorems on subgraphs with the cap (capped subgraphs).

4 The Decomposition Theorem and Adequate Subgraphs

For the median problem on circular genomes, [15] shows that the existences of adequate subgraphs allow us to decompose the problem into two subproblems from which the optimal solution of the original problem can be found by combining solutions to the two subproblems. In this section, we will develop the parallel results for the problem

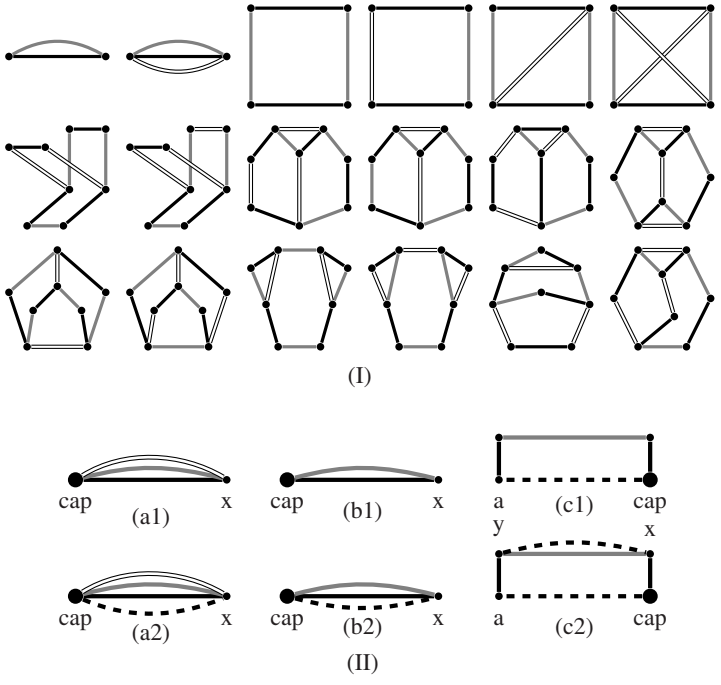

Fig. 3. The most frequent (I) regular adequate subgraphs [15] and (II) capped adequate subgraphs on CMBGs. Black, gray and double edges represent adjacency edges from extant genomes and dashed edges represent adjacency edge in the median genome.

on linear multichromosomal genomes. The cap node in the capped multiple breakpoint graph requires us to distinguish two types of subgraphs: regular subgraphs—the ones not containing the cap node; capped subgraphs—the ones containing the cap node. Parallel to these definitions, we have two types of adequate subgraphs: regular adequate subgraphs and capped adequate subgraphs. The following theorem, with no surprise, states that regular adequate subgraphs defined on CMBGs are identical to adequate subgraphs defined on MBGs. Figure 3.(I) shows the most frequent ones [15].

Theorem 2. *As long as the cap does not involve, regular adequate subgraphs are applicable to capped multiple breakpoint graphs, giving proper decompositions.*

Proof. If we use 2χ caps to delimiter χ linear chromosomes in a traditional way and treat these caps as regular vertices (as these caps are all of degree 2), each CMBG can be transformed into $((2\chi)!)^2$ MBGs. Suppose a regular adequate subgraph exists on the CMBG, then it must also exist in every MBG, since the transformation between the CMBG and the MBGs does not change regular vertices and edges connecting them. This adequate subgraph then decomposes every MBG into two parts, one of which is the adequate subgraph itself. Then the same decomposition induced by this adequate subgraph must happen on the original CMBG.

For a decomposition of a CMBG induced by a capped adequate subgraph, by combining the solutions from the two subproblems, we can find optimal solutions of the following three categories:

1. all optimal solutions with all possible optimal cappings;
2. some optimal solutions with all possible optimal cappings;
3. some optimal solutions with some optimal capping.

In this paper we define capped adequate subgraphs correspond to the first two categories; they are similar to strongly adequate subgraphs and adequate subgraphs on MBGs respectively. Before giving their definitions, we first quickly review some related definitions. Recall that the size of a subgraph (denoted by m) is defined as the number of 0-edges to be added; the rank ($N_{\mathcal{G}}$) of CMBGs is equal to the number of extant genomes involved in the median problem. For a subgraph, we define γ as the number of its edges incident to the cap node. A connected CMBG capped subgraph H of size m is a capped adequate subgraph if $c'^{\Sigma}_{max}(H) \geq \frac{1}{2}(mN_{\mathcal{G}}+\gamma-1)$. It is a capped strongly adequate subgraph if $c'^{\Sigma}_{max}(H) > \frac{1}{2}(mN_{\mathcal{G}}+\gamma-1)$. For the median of three problem, this critical number in above definitions becomes $\frac{1}{2}(3m+\gamma-1)$. It is worth to note that γ for any capped adequate subgraph must be greater than 1, for vertex degrees on any adequate subgraphs are at least 2 [14]. Fig 3.(II) shows the most frequent capped adequate subgraphs.

Theorem 3. *The existence of a capped adequate subgraph on a CMBG gives a proper decomposition from which* an *optimal solution can be found by combining solutions from the two subproblems. The existence of a capped strongly adequate subgraph gives a proper decomposition from which* all *optimal solutions can be found by combining solutions from the two subproblems.*

5 An Exact Algorithm and Lower and Upper Bounds

In this section, we give a high-level description of our algorithm *ASMedian-linear*, which finds exact solutions to the relaxed DCJ median problem on linear multichromosomal genomes. Similar to the algorithm for the circular case, this algorithm iteratively detects existences of regular adequate subgraphs and capped adequate subgraphs. Upon their existences, one or a few adjacencies are added into the median genome. In situations where existences of adequate subgraphs can not be detected (either they do not exist or they have too large sizes to be detected efficiently), this algorithm looks all possible ways of constructing next adjacency edge.

Lower bounds and upper bounds of the total DCJ distance are used to prune obviously bad solutions. The lower bound is derived from the metric property of the DCJ distance (many other distance measures such as reversal distance also have this property),

$$l' = \frac{d_{1,2} + d_{2,3} + d_{1,3}}{2}, \tag{2}$$

where $d_{i,j}$ is the pairwise DCJ distance between genome i and genome j.

Upper bound can be obtained by taking one of the extant genomes which gives the smallest total distance to the remaining genomes as the median genome,

$$u' = d_{1,2} + d_{2,3} + d_{1,3} - \max\{d_{1,2}, d_{2,3}, d_{1,3}\}. \quad (3)$$

When part of the median genome is known, $\tilde{c}$ is used to denote the number of cycles (including paths) formed between existing 0-edges and the given CMBG. Lower/upper bounds of subproblems are denoted by l' and u', whose values are determined by Equations 2 and 3, as each subproblem is viewed as an instance of the median problem. Then the bounds for the original problem are $l = \tilde{c} + l'$ and $u = \tilde{c} + u'$.

It is observed that tightness of lower bound is directly related to algorithms' performance. Any improvement on it shall have a great impact on algorithms' efficiency. An initial value for $d^{\Sigma} = \sum d_{0,i}$ obtained from a fast heuristic algorithm can be used to improve the pruning efficiency in the exact algorithm. At this moment, we use an adequate subgraph based heuristic, which arbitrarily constructs an adjacency edge if adequate subgraphs can not be detected.

Algorithm 1. ASMedian-linear

Input: three genomes with equal or unequal numbers of linear chromosomes
Output: the median genome and the smallest total DCJ distance d^{Σ}

1 run a heuristic algorithm to get an initial value for d^{Σ};
2 construct the capped multiple breakpoint graph, and push it into $\mathcal{L}$, the unexamined list of CMBG or intermediate CMBGs with partial solutions;
3 **while** *$\mathcal{L}$ is not empty and the smallest lower bound l in $\mathcal{L}$ is smaller than d^{Σ}* **do**
4 pop out a (intermediate) CMBG with the smallest lower bound l;
5 **if** *an adequate subgraph (regular or capped) H is detected on this (intermediate) CMBG* **then**
6 add one or a few 0-edges which are guaranteed to exist in an optimal solution, perform shrinking operation for newly added regular 0-edges and push the resultant intermediate CMBG into $\mathcal{L}$;
7 **else**
8 select the vertex v with the smallest label, create a set of intermediate CMBGs by adding one 0-edge incident to v to each of them and shrinking regular 0-edges if there is any, and add them into $\mathcal{L}$;
9 make necessary update for d^{Σ} (the smallest total DCJ distance obtained so far) with upper bounds u derived from newly created intermediate CMBGs ;
10 **return** d^{Σ} as the minimum total distance and the median genome;

6 Performance on Simulated Data

Our algorithm *ASMedian-linear* is implemented in Java, which runs serially on a single CPU. In order to test its performance, we generated sets of simulation data, with varying parameters. In rest of the section, we use n for number of genes in each genome, χ for number of linear chromosomes, r for the total number of reversals used to generate each instance. Three extant genomes in each instance is generated by applying

$r/3$ random reversals of random size on the identity genome (where each chromosome contains roughly the same number of genes, whose labels are consecutive). Each data set contains 100 instances except that the ones in Subsection 6.1 contain 10 instances each.

6.1 Speedups Due to Using Adequate Subgraphs

The program by Zhang et al. [17] is the only published exact solver for the strict DCJ median problem on linear multichromosomal genomes, which exhaustively searches the solution space with a branch-and-bound approach. We compare this program to our algorithm to see gains in speed by using our adequate subgraph based decomposition method. Running times for the program by Zhang et al. are used to to estimate the running times for a relaxed DCJ median solver using a branch-and-bound approach only, as the two problems are closely related with small differences—the relaxed version has a much larger solution space which may require the algorithms to search more solutions for an optimal solution is found; on the other hand the relaxed version may have a smaller optimal total DCJ distances which may let the algorithms terminate earlier compared to the counterpart algorithms for the strict version.

We generated simulations on genomes containing 40 or 50 genes, with varying number of linear chromosomes and varying number of reversals, as shown in Table 1, with 10 instances in each data set. Average running times of the two programs are reported in seconds, together with speedups of our program over Zhang et al's [17] exhaustive search one and average numbers of extra circular chromosomes in the median genomes produced by our algorithm.

The speedups range from 10^1 to 10^8 or even more, increasing along as the numbers of genes and chromosomes increase. Comparisons are carried only on small genomes, otherwise Zhang et al's program can not finish within reasonable time ($\gg$ 400 hours). One can expect much larger speedups on large genomes using our decomposition method. It is safe to say that our adequate subgraph based decomposition method achieves dramatic speedups.

Table 1. Running time comparison between two exact DCJ solvers: our ASMedian-linear for the relaxed version and the one by Zhang et al. for the strict version, on small genomes with varying number of linear chromosomes. For each choice of parameters, results are averaged over 10 simulated instances. Running times for the program by Zhang et al. are used to to estimate the running times for a relaxed DCJ median solver using branch-and-bound approach only, as the two problems are closely related. The table shows our program ASMedian-linear achieves dramatic speedups.

n, r	40, 8			50, 10		
χ	2	4	8	2	4	8
Zhang's	1.9×10^{-2}	1.6×10^{2}	$>1.6\times10^{5}$	2.1×10^{-2}	3.6×10^{3}	$>1.6\times10^{5}$
ASMedian-linear	1.0×10^{-3}	1.0×10^{-3}	1.0×10^{-3}	2.0×10^{-3}	2.0×10^{-3}	2.0×10^{-3}
average # circular chromosomes	0.1	0.0	0.1	0.1	0.0	0.1
speedup	10^1	10^5	$>10^8$	10^2	10^6	$>10^8$

6.2 Performance on Large Genomes

Sets of data on large genomes (with n ranging from 100 to 5000, and χ equal to 2 or 10) are also generated, 100 instances each. The total number of reversals used to generate these data sets is proportional to the number of genes, where this coefficient ranges from 0.3 to 0.9. Average running times are reported in seconds if every instance finishes in 10 minutes; otherwise number of finished instances is reported in parentheses. The last column reports numbers of extra circular chromosomes in the median genomes averaged over all finished instances with the same genome size.

Table 2 and 3 show running times on genomes with 2 or 10 linear chromosomes respectively. All instances with r/n no larger than 0.78 can finished within 1 second for

Table 2. Results for simulated genomes with 2 linear chromosomes. For each data set, 100 instances are simulated and if every instance finishes in 10 minutes, then their average running time is shown in seconds; otherwise number of finished instances is shown with parenthesis. Average numbers of extra circular chromosomes in the median genomes for instances with the same genome size are reported in the last column. As these numbers are no larger than 0.5, our exact solver for the relaxed DCJ median either gives optimal solutions or near-optimal solutions to the strict DCJ median problem on linear multichromosomal genomes.

n	r/n					average # circular chromosomes
	0.3	0.6	0.78	0.84	0.9	
100	$<1\times10^{-3}$	$<1\times10^{-3}$	1×10^{-3}	1×10^{-3}	1×10^{-3}	0.30
200	$<1\times10^{-3}$	1×10^{-3}	4×10^{-3}	7×10^{-3}	1.5×10^{0}	0.40
500	3×10^{-3}	5×10^{-3}	1.0×10^{-2}	1.3×10^{-1}	(98)	0.40
1000	1.4×10^{-2}	1.7×10^{-2}	4.0×10^{-2}	2.5×10^{0}	(80)	0.48
2000	5.7×10^{-2}	6.9×10^{-2}	1.5×10^{-1}	6.9×10^{0}	(21)	0.34
5000	3.7×10^{-1}	4.5×10^{-1}	9.9×10^{-1}	(73)	(0)	0.34

Table 3. Results for simulated genomes with 10 linear chromosomes. For each data set, 100 instances are simulated and if every instance finishes in 10 minutes, then their average running time is shown in seconds; otherwise the number of finished instances is shown with parenthesis. Average numbers of extra circular chromosomes in the median genomes for instances with the same genome size are reported in the last column. As these numbers are no larger than 0.15, our exact solver for the relaxed DCJ median gives optimal solutions in most of the cases, and in the remaining cases it gives near-optimal solutions to the strict DCJ median problem on linear multichromosomal genomes.

n	r/n					average # circular chromosomes
	0.3	0.6	0.78	0.84	0.9	
100	$<1\times10^{-3}$	1×10^{-3}	4×10^{-3}	8×10^{-3}	1.9×10^{-2}	0.08
200	1×10^{-3}	1×10^{-3}	2.8×10^{-2}	4×10^{-3}	(98)	0.13
500	4×10^{-3}	5×10^{-3}	2.3×10^{-2}	2.7×10^{-2}	(73)	0.14
1000	1.4×10^{-2}	1.7×10^{-2}	5.0×10^{-2}	2.7×10^{0}	(52)	0.11
2000	5.5×10^{-2}	6.9×10^{-2}	2.0×10^{-1}	(91)	(20)	0.11
5000	3.7×10^{-1}	4.5×10^{-1}	9.6×10^{-1}	(58)	(0)	0.10

both cases. When r/n is no larger than 0.6, the data sets only differing in χ have almost the same running time.

While as r/n increases, average running time increases quickly and many instances can not finish within 10 minutes. Comparison of Table 2 to Table 3 shows that, instances with 10 linear chromosomes take more time than the ones with 2 linear chromosomes. This is not surprising, because the multichromosomal case is associated with a three way capping problem, whose solution space is $\left((2\chi)!\right)^2$, which increases dramatically as χ increases.

Notice that the reported average numbers of extra circular chromosomes in the median genomes are very small (≤ 0.5). This means that on more than half of the instances, our algorithm gives optimal solutions to the problem of the strict version, and on the remaining instances, our algorithm provides near-optimal solutions after merging the extra circular chromosomes.

7 Conclusion

In this paper, in order to solve the relaxed DCJ median problem on linear multichromosomal genomes efficiently, we introduce capped multiple breakpoint graphs and their adequate subgraphs. By applying our adequate subgraph based decomposition method, we design a relative efficient algorithm *ASMedian-linear* which quickly gives exact solutions to most instances with number of genes up to thousands and with moderate number of evolution events as in real biology problems. Although the solutions may contain some extra circular chromosomes, which is generally considered to be undesirable, these numbers are either zero or very small. So we actually obtain optimal or near-optimal solutions for the strict DCJ median problems.

Since the median problem is NP-hard (for DCJ distance with relaxed version or strict version, or for reversal distance) and there is a need to solve instances with tens of thousands or even more genes (plus other conserved genetic units) in mammal genomes, highly efficient and accurate heuristics should be considered.

Acknowledgments

We would like to thank Jijun Tang for providing their solver for the strict DCJ median problem on linear multichromosomal genomes. I also want to thank anonymous referees, Bernard Moret and Vaibhav Rajan for their help and suggestions in writing this paper.

References

1. Adam, Z., Sankoff, D.: The ABCs of MGR with DCJ. Evol. Bioinformatics 4, 69–74 (2008)
2. Bader, D., Moret, B., Yan, M.: A fast linear-time algorithm for inversion distance with an experimental comparison. J. Comput. Biol. 8(5), 483–491 (2001)
3. Bourque, G., Pevzner, P.: Genome-scale evolution: Reconstructing gene orders in the ancestral species. Genome Res. 12, 26–36 (2002)

4. Caprara, A.: The reversal median problem. INFORMS J. Comput. 15, 93–113 (2003)
5. Hannenhalli, S., Pevzner, P.: Transforming cabbage into turnip: Polynomial algorithm for sorting signed permutations by reversals. In: Proc. 27th ACM Symp. on Theory of Computing STOC 1995, pp. 178–189. ACM, New York (1995)
6. Hannenhalli, S., Pevzner, P.: Transforming men into mice (polynomial algorithm for genomic distance problem). In: Proc. 43rd IEEE Symp. on Foudations of Computer Science FOCS 1995, pp. 581–592. IEEE Computer Soc., Los Alamitos (1995)
7. Lenne, R., Solnon, C., Stützle, T., Tannier, E., Birattari, M.: Reactive stochastic local search algorithms for the genomic median problem. In: van Hemert, J., Cotta, C. (eds.) EvoCOP 2008. LNCS, vol. 4972, pp. 266–276. Springer, Heidelberg (2008)
8. Siepel, A., Moret, B.: Finding an optimal inversion median: Experimental results. In: Gascuel, O., Moret, B.M.E. (eds.) WABI 2001. LNCS, vol. 2149, pp. 189–203. Springer, Heidelberg (2001)
9. Swenson, K., Rajan, V., Lin, Y., Moret, B.: Sorting signed permutations by inversions in o(nlogn) time. In: Batzoglou, S. (ed.) RECOMB 2009. LNCS, vol. 5541, pp. 386–399. Springer, Heidelberg (2009)
10. Swenson, K., To, Y., Tang, J., Moret, B.: Maximum independent sets of commuting and noninterfering inversions. In: Proc. 7th Asia-Pacific Bioinformatics Conf. APBC 2009, vol. 10 (suppl. 1), p. S6 (2009)
11. Tannier, E., Zheng, C., Sankoff, D.: Multichromosomal median and halving problems. In: Crandall, K.A., Lagergren, J. (eds.) WABI 2008. LNCS (LNBI), vol. 5251, pp. 1–13. Springer, Heidelberg (2008)
12. Tesler, G.: Efficient algorithms for multichromosomal genome rearrangements. J. Comput. Syst. Sci. 65(3), 587–609 (2002)
13. Xu, A.: The distance between randomly constructed genomes. In: Proc. 5th Asia-Pacific Bioinformatics Conf. APBC 2007. Advances in Bioinformatics and Computational Biology, vol. 5, pp. 227–236. Imperial College Press, London (2007)
14. Xu, A.: A fast and exact algorithm for the median of three problem—A graph decomposition approach. In: Nelson, C.E., Vialette, S. (eds.) RECOMB-CG 2008. LNCS (LNBI), vol. 5267, pp. 184–197. Springer, Heidelberg (2008)
15. Xu, A.W., Sankoff, D.: Decompositions of multiple breakpoint graphs and rapid exact solutions to the median problem. In: Crandall, K.A., Lagergren, J. (eds.) WABI 2008. LNCS (LNBI), vol. 5251, pp. 25–37. Springer, Heidelberg (2008)
16. Yancopoulos, S., Attie, O., Friedberg, R.: Efficient sorting of genomic permutations by translocation, inversion and block interchange. Bioinformatics 21, 3340–3346 (2005)
17. Zhang, M., Arndt, W., Tang, J.: An exact median solver for the DCJ distance. In: Proc. 14th Pacific Symposium on Biocomputing PSB 2009, pp. 138–149. World Scientific, Singapore (2009)