A Branch-and-Bound Method for the Multichromosomal Reversal Median Problem

Meng Zhang[1], William Arndt[2], and Jijun Tang[2]

[1] College of Computer Science and Technology
Jilin University, China
zhangmeng@jlu.edu.cn

[2] Department of Computer Science and Engineering
University of South Carolina, USA
{arndtw, jtang}@cse.sc.edu

Abstract. The ordering of genes in a genome can be changed through rearrangement events such as reversals, transpositions and translocations. Since these rearrangements are "rare events", they can be used to infer deep evolutionary histories. One important problem in rearrangement analysis is to find the median genome of three given genomes that minimizes the sum of the pairwise genomic distance between it and the three others. To date, MGR is the most commonly used tool for multichromosomal genomes. However, experimental evidence indicates that it leads to worse trees than an optimal median-solver, at least on unichromosomal genomes. In this paper, we present a new branch-and-bound method that provides an exact solution to the multichromosomal reversal median problem. We develop tight lower bounds and improve the enumeration procedure such that the search can be performed efficiently. Our extensive experiments on simulated datasets show that this median solver is efficient, has speed comparable to MGR, and is more accurate when genomes become distant.

1 Introduction

Annotation of genomes with computational pipelines can yield the ordering and strandedness of genes for genomes; each chromosome can then be represented by an ordering of signed genes, where the sign indicates the strand. Rearrangement of genes under reversal (also known as inversion), transposition, and other operations such as translocations, fissions and fusions, is an important evolutionary mechanism [7]. Since genome rearrangement events are "rare", these changes of gene orders enable biologists to reconstruct evolutionary histories far back in time.

One important problem in genome rearrangement analysis is to find the median of three genomes, that is, finding a fourth genome that minimizes the sum of the pairwise genomic distances between it and the three given genomes. This problem is important since it provides a maximum parsimony solution to the smallest binary tree and thus can be used as the basis for more complex methods. However,

K.A. Crandall and J. Lagergren (Eds.): WABI 2008, LNBI 5251, pp. 14–24, 2008.

the median problem is NP-hard for genome rearrangement data [5,9] even under the simplest distance definition. To date, MGR (Multiple Genome Rearrangements) [4] is the only widely used tool that is able to handle multichromosomal genomes. However, experimental evidence indicates that MGR leads to worse trees than an optimal median solver [10], at least on small unichromosomal genomes. With more and more whole genome information available, it becomes very important to develop accurate median solvers for these multichromosomal genomes.

In this paper, we present an efficient branch-and-bound method to find the exact median for three multichromosomal genomes. We use an easy-to-compute and tight lower bound to prune bad branches and introduce a method that enumerates each genome at most once. Such median solver can be easily integrated with the existing methods such that datasets with more than three genomes can be analyzed.

2 Background and Notions

2.1 Genome Rearrangements

We assume a reference set of n genes $\{1, 2, \cdots, n\}$, and a genome is represented by an *ordering* of these genes. A gene g is assigned with an orientation that is either positive, written g, or negative, written $-g$. Specifically, we regard a multichromosomal genome as a set $A = A(1), \ldots, A(N_c)$ of N_c chromosomes partitioning genes $1, \ldots, n$; where $A(i) = \langle A(i)_1, \ldots, A(i)_{n_i} \rangle$ is the sequence of signed genes in the ith chromosome. In this paper, we also assume that each gene occurs exactly once in the genome.

In this study, we only consider *undirected* chromosomes [12], i.e. the flip of chromosomes is regarded as equivalent. We consider the following four operations on a genome: reversal, translocation, fission and fusion. Let $a = \langle a_1, \ldots, a_k \rangle$ and $b = \langle b_1, \ldots, b_m \rangle$ be two chromosomes. A *reversal* on the indices i and j $(i \leq j)$ of chromosome a produces the chromosome with linear ordering $a_1, a_2, \cdots, a_{i-1}, -a_j, -a_{j-1}, \cdots, -a_i, a_{j+1}, \cdots, a_k$. A *translocation* transforms $a = \langle E, F \rangle$ and $b = \langle X, Y \rangle$ into $\langle E, Y \rangle$ and $\langle X, F \rangle$, where E, F, X, Y are gene segments. The *fusion* of a and b results in a chromosome $c = \langle a_1, \ldots, a_k, b_1, \ldots, b_m \rangle$. A *fission* of a results in two new chromosomes $\pi = \langle a_1, \ldots, a_{i-1} \rangle$ and $\sigma = \langle a_i, \ldots, a_k \rangle$.

An important concept in genome rearrangement analysis is the number of breakpoints between two genomes. Given genomes A and B, a *breakpoint* is defined as an ordered pair of genes (i, j) such that i and j are adjacent in A but not in B.

2.2 Genomic Distance for Multichromosomal Genomes

We define the *edit distance* as the minimum number of operations required to transform one genome into the other. Hannenhalli and Pevzner (HP) [8] provided a polynomial-time algorithm to compute the distance (HP distance) for reversals, translocations, fissions and fusions, as well as the corresponding sequence of events. Tesler [12] corrected the HP algorithm, which was later improved by Bergeron et al. [3].

Yancopoulos et al. [13] proposed a "universal" double-cut-and-join (DCJ) operation, resulting in a new genomic distance that can be computed in linear time. Although there is no direct biological evidence for DCJ operations, these operations are very attractive because they provide a unifying model for genome rearrangement [2]. Given two genomes A and B, computing the DCJ distance between these two genomes (denote $d_{DCJ}(A, B)$) is much easier to implement than computing the HP distance (denote $d_{HP}(A, B)$).

2.3 Reversal Median Problem

The median problem on three genomes is to find a single genome that minimizes the sum of the pairwise distances between itself and each of the three given genomes[1]. It has been proven that this problem is NP-hard [5] for unichromosomal genomes using reversal distance. Specifically the *reversal median problem* (RMP) is to find a median genome that minimizes the summation of the multichromosomal HP distances on the three edges.

Several solvers have been proposed for the unichromosomal reversal median problem (including MGR), among them, the one developed by Caprara [6] is the most accurate. Caprara's median solver is exact and treats the problem in a graph model, where each permutation corresponds to a matching of a point set. As a branch-and-bound algorithm, it enumerates all possible solutions and tests them edge by edge. At first, a lower bound is computed from the graph of the given genome's matchings. In each step of testing, the graph is reduced to a smaller one according to each edge of the solution being tested. A new bound is then computed from the new graph for bound testing. If the test failed, all solutions containing the edges tested so far in the current solution are excluded.

On the other hand, when used for three genomes, MGR attempts to find a longest sequence of reversals from one of the three given genomes that, at each step in the sequence, moves closer to the other two genomes. Since it is limited to a small subset of possible paths, MGR is less accurate than Caprara's median solver. Our method presented in this paper is inspired by Caprara's solver, and to our knowledge, is the first exact solver for the multichromosomal reversal median problem.

3 Graph Model for Undirected Genome

In this section, we introduce the graph model and a lower bound on the HP distance, which will be used in our new median solver.

3.1 Capless Breakpoint Graph

We modify the breakpoint graph to deal with genomes consisting of undirected chromosomes. In [8,12], *caps* are introduced to transform the multichromosomal genomes problem to unichromosomal problem. Caps play an important role in

[1] The median problem can be generalized for q ($q \geq 3$) genomes.

deriving the rearrangement scenario; however, since they are not necessary to compute the HP distance, capping nodes are not used here. We also do not distinguish between undirected unichromosomal and multichromosomal genomes, and treat both in a uniform way. This model is equivalent to the graph model of Bergeron et al. [1].

Given a node set V, we call an edge set $M \subset \{(i,j) : i, j \in V, i \neq j\}$ a *matching* of V if each node in V is incident to at most one edge in M. If each node in V is incident to exactly one edge in M, the matching is called *perfect*, otherwise *partial*. A genome A on genes $1, \ldots, n$ can be transformed to an unsigned genome $\mathcal{A}$ on $1, \ldots, 2n$, by replacing each positive entry g with $2g-1$, $2g$ and each negative entry g with $2|g|$, $2|g|-1$.

Consider the node set $\mathcal{V} := 1, \ldots, 2n$, and the associated perfect matching $\mathcal{H} := \{(2i-1, 2i) : i = 1, \ldots, n\}$ (the base matching of $\mathcal{V}$). There is a correspondence between genomes composed of linear chromosomes and matchings $\mathcal{M}$ of $\mathcal{V}$ such that there are no cycles in $\mathcal{M} \cup \mathcal{H}$. These matchings are called *genome matchings*. In particular, the genome matching $\mathcal{M}(A)$ associated with a genome A is defined by

$$\mathcal{M}(A) := \big\{(\mathcal{A}(i)_k, \mathcal{A}(i)_{k+1}) : k \in \{2, 4, \ldots, 2n_i - 2\}, i \in \{1, \ldots, N_c\}\big\}.$$

The nodes in $\{\mathcal{A}(i)_1 : 1 \leq i \leq n_c\} \cup \{\mathcal{A}(i)_{2n_i} : 1 \leq i \leq n_c\}$ are called *end node*s of A, denoted by `A-ends`. The genome matching has no capping node appended and all end nodes are not incident to any edge, thus the defined genome matchings are partial matchings. The absence of caps is a crucial step to reduce the complexity of the median problem for multichromosomal genomes.

Given two genomes A and B, the capless breakpoint graph $\mathcal{G}(A, B) = (\mathcal{V}, \mathcal{M}(A) \cup \mathcal{M}(B))$ defines a set of cycles and paths whose edges are alternate in $\mathcal{M}(A)$ and in $\mathcal{M}(B)$. An example of $\mathcal{G}$ can be found in Fig. 1. $c(A, B)$ denotes the number of cycles. Paths start from an end node and terminate at another end node. According to the type of ends, all paths in the graph can be classified into three groups: AA-paths, BB-paths, and AB-paths(called *odd paths* in [2]). A node which is an end of both A and B forms an AB-path of length 0. Denote the number of AB-paths by $|AB|$, and the number of AA-paths by $|AA|$.

3.2 Lower Bound of the HP Distance

We derive the following lower bound on the HP distance for undirected genomes, using only the parameters of the number of cycles and paths.

$$d_{HP} \geq n - \big(c(A, B) + |AB|/2\big) \tag{1}$$

This bound is indeed the same as the double-cut-and-join distance formula [2]. It can be directly derived from the HP distance formula for two multichromosomal genomes [8,12] or simply from the result that $d_{DCJ} \leq d_{HP}$ [3], where $d_{DCJ} = n - \big(c(A, B) + |AB|/2\big)$ [2].

For convenience, call $c(A, B) + |AB|/2$ the *pseudo-cycle* of $\mathcal{G}(A, B)$, denoted by $\tilde{c}(A, B)$. By the aid of DCJ distance, many useful results are easy to prove. Since the DCJ distance satisfies the triangle inequality, we have

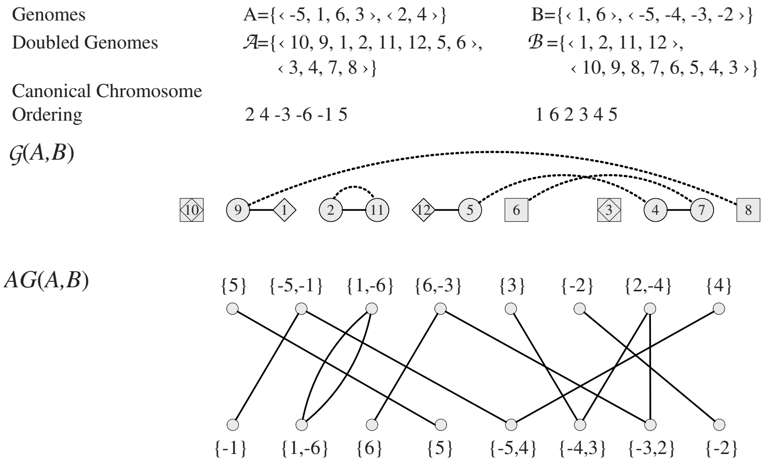
Genomes
A={‹ -5, 1, 6, 3 ›, ‹ 2, 4 ›}
B={‹ 1, 6 ›, ‹ -5, -4, -3, -2 ›}
Doubled Genomes
𝒜={‹ 10, 9, 1, 2, 11, 12, 5, 6 ›, ‹ 3, 4, 7, 8 ›}
ℬ={‹ 1, 2, 11, 12 ›, ‹ 10, 9, 8, 7, 6, 5, 4, 3 ›}
Canonical Chromosome Ordering
2 4 -3 -6 -1 5
1 6 2 3 4 5
𝒢(A,B)
10
9
1
2
11
12
5
6
3
4
7
8
AG(A,B)
{5}
{-5,-1}
{1,-6}
{6,-3}
{3}
{-2}
{2,-4}
{4}
{-1}
{1,-6}
{6}
{5}
{-5,4}
{-4,3}
{-3,2}
{-2}

Fig. 1. The $\mathcal{G}(A, B)$ is the capless breakpoint graph of genome A and B. In $\mathcal{G}(A, B)$ diamonds represent `B-ends`, squares represent `A-ends`. Squares with a diamond inside indicate nodes are both `A-ends` and `B-ends`. In this figure, $n = 6$, $c(A, B) = 1$, $|AB| = 4$, and pseudo-cycle is 3. The graph $AG(A, B)$ is defined in [2] and the concept of canonical chromosome ordering is introduced in 4.3.

Lemma 1. *Given three genomes A,B,C, $n - \tilde{c}(A, C) + n - \tilde{c}(C, B) \geq n - \tilde{c}(A, B)$.*

3.3 Contraction Operation

We define the Multi-Breakpoint (MB) graph associated with q genomes $G_1, G_2, \ldots, G_q$ as the graph $\mathcal{G}(G_1, G_2, \ldots, G_q)$ with node set $\mathcal{V}$ and edge multiset $\mathcal{M}(G_1) \cup \mathcal{M}(G_2), \ldots, \cup \mathcal{M}(G_q)$. Note that for two matchings $\mathcal{M}(G_j)$ and $\mathcal{M}(G_k)$, $j \neq k$, some edges may be common in both matchings, but they are considered distinct parallel edges in the MB graph $\mathcal{G}(G_1, G_2, \ldots, G_q)$. In this paper, q always equals three.

Let $Q := \{1, \ldots, q\}$ and let τ be a genome, $\gamma(\tau) = \sum_{k \in Q} \tilde{c}(\tau, G_k)$. By Equation 1, for any genome σ, $\delta(\sigma) = \sum_{k \in Q} d_{HP}(\sigma, G_k) \geq qn - \gamma(\sigma)$, where n is the number of genes. We introduce the *Pseudo-Cycle Median Problem* (PMP): given undirected genomes $G_1, G_2, \ldots, G_q$, find a genome τ such that $qn - \gamma(\tau)$ is minimized. We have the following theorem:

Theorem 1. *Given an RMP instance and the associate PMP instance on q undirected genomes. Let δ^* and $qn - \gamma^*$ denote the optimal solution values of RMP and PMP, then $\delta^* \geq qn - \gamma^*$.*

Proof. If σ is an optimal solution of RMP and τ an optimal solution of PMP, $\delta^* = \delta(\sigma) \geq qn - \gamma(\sigma) \geq qn - \gamma(\tau) = qn - \gamma^*$. □

Theorem 1 implies that the solution of PMP yields a lower bound on the optimal solution of RMP.

In Caprara's solver [6], the *contraction* on breakpoint graphs for perfect matchings is introduced. We modify it to fit the MB graphs for partial matchings which are dealt with here. Given a partial matching $\mathcal{M}$ on node set $\mathcal{V}$ and an edge $e = (i, j) \in E = \{(i, j) : i, j \in \mathcal{V}, i \neq j\}$, we define $\mathcal{M}/e$, the *contraction* of e on $\mathcal{M}$ (or on the genome corresponding to $\mathcal{M}$) as follows. If $e \in \mathcal{M}$, let $\mathcal{M}/e := \mathcal{M} \setminus \{e\}$. If one of i and j is an end, say i, and (b, j) is the edge incident to j, then $\mathcal{M}/e := \mathcal{M} \setminus \{(b, j)\}$. If both i and j are ends, $\mathcal{M}/e := \mathcal{M}$. Otherwise, let (a, i), (j, b) be the two edges in $\mathcal{M}$ incident to i and j, $\mathcal{M}/e := \mathcal{M} \setminus \{(a, i), (j, b)\} \cup \{(a, b)\}$. Given an MB graph $\mathcal{G}(G_1, G_2, \ldots, G_q)$, the *contraction* of an edge e is the operation that modifies the MB graph $\mathcal{G}(G_1, G_2, \ldots, G_q)$ as follows. Edge (i, j) is removed along with nodes i and j. For $k = 1, \ldots, q$, $\mathcal{M}(G_k)$ is replaced by $\mathcal{M}(G_k)/e$, and the base matching $\mathcal{H}$ is replaced by $\mathcal{H}/e$.

4 Branch-and-Bound Algorithm

In this section, we describe a branch-and-bound algorithm for the multichromosomal reversal median problem.

4.1 A Basic Branch-and-Bound Algorithm for PMP

The following lower bound on reversal median is based on Lemma 1.

Lemma 2. *Given a PMP instance associated with genomes $G_1, \ldots, G_q$,*

$$\gamma^* \leq \frac{qn}{2} + \sum_{k=1}^{q-1} \sum_{l=k+1}^{q} \frac{\tilde{c}(G_k, G_l)}{q-1}. \tag{2}$$

The lower bound on the optimal PMP solution given by qn minus the right-hand side of (2), called LD, can be computed in $O(nq^2)$ time. For any PMP solution T, the cycles and paths in $T \cup \mathcal{M}(G_k)$ has one-to-one correspond to that in $(T/e) \cup (\mathcal{M}(G_k)/e)$ except for cycle of two copies of e. Therefore, for partial matchings and the new contraction defined on it, the same lemma in [6] holds.

Lemma 3. *Given a PMP instance and an edge $e \in E$, the best PMP solution containing e is given by $\tilde{T} \cup \{e\}$, where $\tilde{T}$ is the optimal solution of the PMP instance obtained by contracting edge e.*

According to Lemma 3, if we fix an edge e in the matching of PMP solution, an upper bound on the pseudo-cycle is given by $|\{k : e \in \mathcal{M}(G_k)\}|$ plus the upper bound (2) computed after the contraction of e. A branch-and-bound algorithm can be designed by these Lemmas. It enumerates all genomes gene by gene. In each step, it either selects a gene as an end or an inner gene in a chromosome of the solution (median) genome. In term of matching, the former operation fixes an end node in the matching of the solution and the latter one fixes an edge e in the matching. The former is called *end fixing* and the later *edge fixing*.

In edge fixing, the algorithm applies a contraction of edge e on the input genomes and computes a lower bound from these altered genomes. The number of newly generated cycles of length 0 is added to a counter. Based on these two values, the lower bound of all PMP solutions containing all edges added so far up to e can be derived. If it is greater than the current lower bound of the median problem, then e is not an edge in the matching of the best solution containing current fixed edges, thus the algorithm enumerates another gene or mark the last gene in the current solution as an end. Otherwise, the edge is fixed in the current solution.

In end fixing, no contraction operation is applied on the intermediate genomes and the lower bound is not changed. The number of newly generated paths of length 0 is added to a counter. When a complete solution is available, the lower bound that takes all the ends into account is computed by the aid of the counters and compared with the current lower bound. If they are equal the algorithm stops and the current solution is optimal.

4.2 Genome Enumeration

In the beginning, the algorithm enumerates all partial matchings by fixing, in turn, either 1, or 2, $\ldots$, or $2n$ as an end in the solution matching, which corresponds to one end of a chromosome, but the other end of this chromosome is not chosen yet. We call this chromosome an *opening chromosome* and this end an *open end*. Recursively, if the last operation is an edge fixing of (x, j), we proceed the enumeration by fixing in the solution, in turn, edge (k, l) where k is the other end of the edge (j, k) in $\mathcal{H}$ incident to j, for all l with no incident edge fixed so far or fix k as an end. Two cases exist if the last operation is an end fixing of x:

1. x is the other end of the current opening chromosome. We call x a *closed end* and the chromosome is closed. If there exist nodes that are not fixed in the solution, we enumerate by fixing in the solution, in turn, all available nodes as an open end in the solution.
2. x is an open end. We enumerate the cases as follows: each edge (y, l) is fixed in the solution one at a time, over all edges such that y is the node incident to x in $\mathcal{H}$ and l has no incident edge fixed so far; or we take y as a closed end.

4.3 Improved Branch-and-Bound Algorithm for PMP

The above scheme checks all concatenates of a genome thus will enumerate a genome more than once, which costs considerable computing time. We define the *canonical chromosome ordering* to overcome this problem. Let chromosome $X = \langle X_1, \ldots, X_M \rangle$. The *canonical flipping* of X is: $\langle X_1, \ldots, X_j, \ldots, X_M \rangle$, if $|X_1| < |X_M|$; otherwise the flipping is $\langle -X_M, \ldots, -X_j, \ldots, -X_1 \rangle$. For single-gene chromosomes, say $\langle g \rangle$, the canonical flipping is $\langle |g| \rangle$.

For a signed gene g, if $g > 0$, $l(g) = 2g - 1, r(g) = 2g$; if $g < 0$, $l(g) = 2|g|, r(g) = 2|g| - 1$. Let $\langle X_1, \ldots, X_M \rangle$ be the canonical flipping of chromosome X, the *smaller end* of X is $l\big(X_1\big)$. The *larger end* of X is $r\big(X_M\big)$. After the canonical flipping of each chromosome, we order the chromosomes by their smaller ends in increasing order to obtain the canonical chromosome ordering of a genome. Obviously, one genome has only one form of canonical ordering, and it can be uniquely represented by the canonical ordering along with the markers of start and end points. Fig. 1 shows an example.

We improve the basic branch-and-bound algorithm to enumerate each genome only once by using the canonical ordering. There are two states in the algorithm: (1) open chromosome and (2) build chromosome. The algorithm is in state "open chromosome" when it starts. It enumerates all the possible smaller ends of the first chromosome by fixing, in turn, either node 1, or 2,..., or $2n - 2$ as an end in the solution. When a smaller end is selected, the state is changed to "build chromosome". Recursively, in state "build chromosome", let the last edge fixed in the current partial solution be (i, j) and the edge in $\mathcal{H}$ incident to j be (j, k). There exist two branches if k can be a larger end of the current opening chromosome:

1. "closed chromosome": it fixes k as a larger end and closes the current opening chromosome. The state is changed to "open chromosome".
2. "build chromosome": it proceeds the enumeration by fixing in the solution, in turn, edge (k, l), for all l not be fixed in solution so far.

If k cannot be the larger end then only the "build chromosome" branch is permitted. If the state is "open chromosome", it will enumerate all the available smaller ends of the next chromosome by fixing, in turn, all available smaller ends.

We proceed in a depth-first order again. With this scheme we can perform the lower bound test after each edge fixing. When a node is fixed as an end, no operation is applied on the input genomes, thus the lower bound equals that of the previous step. At each end fixing, we record the number of newly generated SG_k-path, $k \in Q$, of length 0, where S denotes the partial solution. Thus when a complete solution is available, we will also know the total SG_k-paths in $\mathcal{M}(S) \cup \mathcal{M}_1, \ldots, \mathcal{M}(S) \cup \mathcal{M}_q$.

In the implementation, the initial lower bound LD is computed form the input genomes. The branch-and-bound starts searching for a PMP solution of target value $T = LD$ and tries another gene as soon as the lower bound for the current partial solution is greater than T. If a solution of value T is found, it is optimal and we stop, otherwise there is no solution of value LD. The algorithm then restarts with an increased target value $T = LD + 1$, and so on. The search stops as soon as we find a solution with the target value. All the partial solutions tested under target value T will be reconsidered under $T + 1$. The computation with $T + 1$ is typically much longer than the previous one, therefore the running time is dominated by the running time of the last target value.

4.4 Branch-and-Bound Algorithm for RMP

The above PMP algorithm can easily be modified to find the optimal RMP solutions. There are two modifications. First, the initial target value of the median score is computed as $\frac{\sum_{k=1}^{q-1}\sum_{l=k+1}^{q} d_{HP}(G_k,G_l)}{q-1}$. Second, when a complete solution whose lower bound is not greater than T is available, we compute the sum of HP distances between this solution and the input genomes. If the sum equals T, the algorithm stops, and the current solution is optimal. If there is no such solution, no genome exists whose sum of HP distances between it and the input genomes is better than $T+1$. We increase the target value by 1 and start the algorithm again. Though some non-optimal solutions can pass the lower bound test, they can not pass the HP distance test. But any optimal genome that passes the lower bound test will also pass the HP distance test according to Theorem 1. The first optimal genome (there may be several) encountered will be outputted as the optimal RMP solution.

5 Experimental Results

We have implemented the algorithm and conduct simulations to assess its performance. Our implementation is based on Caprara's unichromosomal median solver and uses MGR's code for multichromosomal reversal distance computation.

In our simulation study, each genome has 100 and 200 genes, with 2 and 4 chromosomes respectively. We create each dataset by first generating a tree topology with three leaves, assigning it with different edge lengths. We assign a genome G_0 to the root, then evolve the signed permutation down the tree, applying along each edge a number of operations equal to the assigned edge length. We test a large range of evolutionary rates: letting r denote the expected number of evolutionary events along an edge of the model tree, we used values of r in the range of 4 to 32 for datasets with 100 genes, and 4 to 40 for datasets with 200 genes. The actual number of events along each edge is sampled from a uniform distribution on the set $\{1, 2, \ldots, 2r\}$. We compare our new method with MGR and use two criteria to assess the accuracy: the median score which can be computed by summing the three edge lengths, and the multichromosomal reversal distance from the inferred median to the true ancestor which is known in our simulation.

Table 1 shows the median score for these two methods. When the genomes are closed (smaller r values), both methods return the same score. When r increases,

Table 1. Comparisons of the average median scores for 100 genes/2 chromosomes (top) and 200 genes/4 chromosomes (bottom)

	r=4	r=8	r=12	r=16	r=20	r=24	r=28	r=32
Our Method	11.1	22.6	39.0	55.2	63.4	72.6	76.1	84.2
MGR	11.1	22.6	39.0	55.2	63.5	73.3	77.4	86.5

	r=4	r=8	r=16	r=24	r=32	r=40
Our Method	11.8	22.0	45.2	78.0	98.8	111.6
MGR	11.8	22.0	45.2	78.0	98.8	112.2

Table 2. The average reversal distances from the inferred median to the true ancestor, for 100 genes/2 chromosomes (top) and 200 genes/4 chromosomes (bottom)

	r=4	r=8	r=12	r=16	r=20	r=24	r=28	r=32
Our Method	0	0.1	0	0.6	1.0	3.1	1.8	4.4
MGR	0	0.1	0	0.6	1.7	3.4	3.7	5.4

	r=4	r=8	r=16	r=24	r=32	r=40
Our Method	0	0.1	0	0	0	0.8
MGR	0	0.2	0	0	0	1.8

Table 3. The average time (in seconds) used for 100 genes/2 chromosomes (top) and 200 genes/4 chromosomes (bottom)

	r=4	r=8	r=12	r=16	r=20	r=24	r=28	r=32
Our Method	<1	<1	<1	12	6	149	203	411
MGR	<1	2	6	17	27	32	46	95

	r=4	r=8	r=16	r=24	r=32	r=40
Our Method	<1	<1	11.0	115.2	361.5	866.2
MGR	< 1	1.2	11.1	57.6	134.3	184.4

our method performs better by returning solutions with smaller scores. Although the difference of median scores seems small, in genome rearrangement analysis based on parsimony, such difference will have a big impact on the accuracy of phylogenies [11].

Table 2 shows the reversal distance of the inferred median to the true ancestor. Both MGR and our new method return solutions that are very close to the true ancestors, especially for datasets with 200 genes. Our new method is superior to MGR when the genomes become distant (for example, $r \geq 16$ for 100 genes).

Table 3 shows the average run time. Surprisingly, our method is faster when the genomes are not distant. However, it is much slower when the edge lengths increase and it cannot finish many datasets with r larger than the values in this experiment. For datasets with very large evolutionary rates, the edit distance will severely under-estimate the true distance, hence all median-based approaches will become unreliable.

6 Conclusions

In this paper we present a new branch-and-bound method for the multichromosomal reversal median problem. Our extensive experiments show that this method is more accurate than existing methods. However, this method is still primitive needs further improvements. In recent years, the double-cut-and-join (DCJ) distance has attracted much attention. We find that the lower bound used in this paper is indeed very similar to the DCJ distance [2], thus it may be relatively easy to extend our work and develop a new DCJ median solver.

Acknowledgments

WA and JT are supported by US National Institutes of Health (NIH grant number R01 GM078991-01). MZ is supported by NSF of China No.60473099.

References

1. Bergeron, A., Mixtacki, J., Stoye, J.: On sorting by translocations. In: Miyano, S., Mesirov, J., Kasif, S., Istrail, S., Pevzner, P.A., Waterman, M. (eds.) RECOMB 2005. LNCS (LNBI), vol. 3500, pp. 615–629. Springer, Heidelberg (2005)
2. Bergeron, A., Mixtacki, J., Stoye, J.: A unifying view of genome rearrangements. In: Bücher, P., Moret, B.M.E. (eds.) WABI 2006. LNCS (LNBI), vol. 4175, pp. 163–173. Springer, Heidelberg (2006)
3. Bergeron, A., Mixtacki, J., Stoye, J.: Hp distance via double cut and join distance. In: Ferragina, P., Landau, G.M. (eds.) CPM 2008. LNCS, vol. 5029, Springer, Heidelberg (2008)
4. Bourque, G., Pevzner, P.: Genome-scale evolution: reconstructing gene orders in the ancestral species. Genome Research 12, 26–36 (2002)
5. Caprara, A.: Formulations and hardness of multiple sorting by reversals. In: Proc. 3rd Ann. Int'l Conf. Comput. Mol. Biol. (RECOMB 1999), pp. 84–93. ACM Press, New York (1999)
6. Caprara, A.: On the practical solution of the reversal median problem. In: Gascuel, O., Moret, B.M.E. (eds.) WABI 2001. LNCS, vol. 2149, pp. 238–251. Springer, Heidelberg (2001)
7. Downie, S.R., Palmer, J.D.: Use of chloroplast DNA rearrangements in reconstructing plant phylogeny. In: Soltis, P., Soltis, D., Doyle, J.J. (eds.) Plant Molecular Systematics, pp. 14–35. Chapman and Hall, Boca Raton (1992)
8. Hannenhalli, S., Pevzner, P.A.: Transforming mice into men (polynomial algorithm for genomic distance problems). In: Proc. 36th Ann. IEEE Symp. Foundations of Comput. Sci. (FOCS 1995), pp. 581–592. IEEE Press, Piscataway (1995)
9. Pe'er, I., Shamir, R.: The median problems for breakpoints are NP-complete. Elec. Colloq. on Comput. Complexity 71 (1998)
10. Swenson, K.W., Arndt, W., Tang, J., Moret, B.M.E.: Phylogenetic reconstruction from complete gene orders of whole genomes. In: Proc. 6th Asia Pacific Bioinformatics Conf. (APBC 2008), pp. 241–250 (2008)
11. Tang, J., Wang, L.: Improving genome rearrangement phylogeny using sequence-style parsimony. In: Proc. 5th IEEE Symp. on Bioinformatics and Bioengineering BIBE 2005. IEEE Press, Los Alamitos (2005)
12. Tesler, G.: Efficient algorithms for multichromosomal genome rearrangements. J. Comput. Syst. Sci. 63(5), 587–609 (2002)
13. Yancopoulos, S., Attie, O., Friedberg, R.: Efficient sorting of genomic permutations by translocation, inversion and block interchange. Bioinformatics 21(16), 3340–3346 (2005)