Improving Genome Rearrangement Phylogeny Using Sequence-Style Parsimony

Jijun Tang[1] and Li-San Wang[2]

[1] Department of Computer Science and Engineering, University of South Carolina
jtang@cse.sc.edu
[2] Department of Biology, University of Pennsylvania
lswang@mail.med.upenn.edu

Abstract. The study of genome rearrangements, the evolutionary events that change the order and strandedness of genes within genomes, presents new opportunities for discoveries about deep evolutionary events. The best software so far, GRAPPA, solves breakpoint and inversion phylogenies by scoring each tree topology through iterative improvements of internal node gene orders. We find that the greedy hill-climbing approach means the accuracy is limited because of multiple local optima. To address this problem, we propose integration GRAPPA with MPME, a string encoding of gene adjacency relationships whose optimal internal node assignments can be determined globally in polynomial time, to provide better initializations for GRAPPA. In simulation studies, the new algorithm yields shorter tree lengths and better accuracy in phylogeny reconstruction.

1 Introduction

Genome rearrangement evolution. Modern techniques can yield the ordering and strandedness of genes for genomes, particularly for smaller ones such as the single chromosome of organelles (mitochondria and chloroplasts). Each chromosome can then be represented by an ordering of signed genes, where the sign indicates the strand. Rearrangement of genes under inversion, transposition, and perhaps other operations, is an important evolutionary mechanism [10]. Reconstructing phylogenies from gene-order data has been studied intensely since the pioneering papers of Sankoff [2, 3, 23]. Because they capture the complete genome, gene-order data do not suffer from the gene tree vs. species tree problem; and because rearrangements of genes are rare genomic events [20], gene-order data enable the reconstruction of evolutionary events far back in time. In consequence, many biologists have embraced this new source of data in their phylogenetic work [10, 18, 19]. Past simulation studies [15, 24] confirm that gene-order data lead to very accurate reconstructions.

We examine the following rearrangement events, which has been intensively studied by biologists. An *inversion* on a gene order produces a new gene order in which a substring of genes has been replaced by their reverse complement substring—for instance, an inversion from the second to the fourth position in the gene order $(1, 2, 3, 4, 5, 6)$ produces the new order $(1, 2, -4, -3, 5, 6)$. A *transposition* on a gene order produces a new gene order in which a substring of genes has been moved from one location to another—for instance, a transposition of the substring defined by the second and fourth positions to the sixth position transforms the gene order $(1, 2, 3, 4, 5, 6, 7)$ into the new order $(1, 2, 5, 3, 4, 6, 7)$.

The *edit distance* between two genomes is the minimum number of evolutionary events needed to transform one genome into the other. The *inversion distance* between two genomes is the minimum number of inversions between two genomes, which can be computed in linear time for signed orderings [1]. Given two signed orderings of genes, a *breakpoint* is an adjacency (read on either strand) present in one genome, but not the other (for example, adjacency $(2,3)$ is in genome $(1,2,3,4,5,6)$ but not in $(1,2,-4,-3,5,6)$); the *breakpoint distance* [23] between the two genomes is just the number of breakpoints.

We use the *Generalized Nadeau-Taylor* model [27] to characterize the stochastic process of genome rearrangement evolution. The model assumes that the number of each of the three types of events obeys a Poisson distribution on each edge, that the relative probabilities of each type of event are fixed across the tree, and that events of a given type are equiprobable. Thus we can represent a GNT model tree as a triple $(T, \{\lambda_e\}, (\gamma_{\text{inv}}, \gamma_{\text{transp}}, \gamma_{\text{inv.transp}}))$, where the three γ's define the relative probabilities of the three types of events. Note that inversions and transpositions cannot affect the gene contents of genomes, only their order; thus we assume throughout the rest of this paper that all genomes have the same gene contents and that each gene appears exactly once in each genome (no duplicates).

Maximum parsimony. In this paper we focus on the maximum parsimony criterion [11], which can be defined as follows. Let X be the set of admissible states (for example, the set of all DNA sequences of length k, or the set of all possible gene orders with n distinct genes). Let $S = \{s_1, \ldots, s_m\}$ be the collection of input data, where each member s_i is an element of X. Let $d : X \times X \to R$ be a cost function; for simplicity we assume d is a metric[1]. Let T be any unrooted binary tree topology with m leaves $\{1, \ldots, m\}$ and $m-2$ internal nodes $\{m+1, \ldots, 2m-2\}$, such that each leaf i of T is labeled by s_i. The *parsimony score* (or *length*) of T is defined as

$$\min_f \sum_{(u,v)\in E(T)} d(f(u), f(v)),$$

where f is an *assignment*, a function that maps internal nodes and leaves of T to members of X, and $f(i) = s_i$, $1 \le i \le m$. The problem of maximum parsimony is to find all tree topologies with smallest parsimony scores. If X is the set of all DNA sequences of the same length and d is the Hamming distance between two sequences (i.e. the number of mismatched sites), the Fitch algorithm [11] computes the score of any tree in $O(mk)$ time. However, finding trees with the smallest parsimony score in this case is NP-hard [9].

Inversion and Breakpoint Phylogeny. Reconstruction methods from gene-order data include methods based strictly on pairwise distances such as neighbor-joining (NJ) [21], methods based on remapping gene orders into sequences [7, 26], and methods based on parsimony, such as our software suite `GRAPPA` [17]. The last has proved the most accurate in all of our tests to date [14, 15]. `GRAPPA` is based on the approach pioneered by Sankoff and Blanchette in the software package `BPAnalysis` [23], but runs 1–5 million times faster and is more accurate.

[1] The dissimilarity d is a metric if d is symmetric, satisfies triangular inequality, $d(x,y) \ge 0$ for all x and y, and $d(x,y) = 0$ if and only if $x = y$

The basic approach of GRAPPA is to examine every possible tree topology in turn, score the tree if it is potentially good, and retain the tree(s) with the lowest score. The score is obtained by assigning signed gene orders to the internal node of a tree and calculating the length of each edge, then summing these lengths. Since computing the tree length is extremely costly, GRAPPA employs a pruning strategy by first obtaining an easy-to-compute lower bound before actually computing the length of any tree T [16]; if the lower bound is higher than the length of the best tree so far, we can discard T without actually scoring it. To ensure a good tree is obtained initially, GRAPPA uses neighbor joining on EDE distance [16].

When computing the score of a tree, GRAPPA improves the assignment of gene orders by iteratively improving the assignment of gene orders to internal nodes using median-of-three-genome solvers, one node at a time (see Section 2.1 for details). Computing the median (under either breakpoint or inversion distance) is itself an NP-hard optimization problem, but instances of the size produced by chloroplast genomes can be solved almost exactly within reasonable time [6]. Of the two choices, the inversion median invariably leads to better solutions [14].

In [24], the authors integrated GRAPPA with DCM (disk-covering method, a divide-and-conquer approach for phylogeny reconstruction) to handle up to more than a thousand genomes in simulation, though each edge is short (20 events or less per edge) – for longer edges, the median solver seldom finishes.

MPME and MPBE. Despite their accuracy, both BPAnalysis and GRAPPA are computationally costly, making analyzing large datasets impractical with the capabilities of current computational hardware. Two heuristics, MPBE and MPME, are available to approximate the breakpoint phylogeny. Both methods transform adjacency pairs from the signed permutation into sequence-like strings. These transformed encodings are then inputs to the ordinary sequence parsimony software, where the scoring of a tree topology (and the corresponding optimal sequences for the internal nodes) can be done in low polynomial time using a dynamic programming algorithm [11].

The *Maximum Parsimony on Binary Encodings* (MPBE [8]) algorithm has running time exponential in the number of genomes but linear in the number of genes. In MPBE, each gene ordering is translated into a binary string, where each site from the binary string corresponds to a pair of genes. (The ordering of the sites is immaterial in this encoding.) For the pair (g_i, g_j), the string has a 1 at the corresponding site if g_i is immediately followed by g_j in the gene ordering and a 0 otherwise (note that g_i and g_j can be negative and that, since (g_i, g_j) and $(\text{-}g_j, \text{-}g_i)$ denote the same adjacency, we need only one site for both). There are $\binom{n}{2}$ pairs, where n is the number of genes in each genome, but we drop the sites where every string has the same value.

Bryant [5] proposed an encoding method, based on an earlier characterization approach of Boore [4], that we have used to develop a new character scoring method that we call *Maximum Parsimony on Multistate Encodings* (MPME) in [26]. Let n be the number of genes in each genome; then each gene order is translated into a string of length $2n$. For every i, $1 \leq i \leq n$, site i takes the value of the gene immediately following gene i and site $n+i$ takes the value of the gene immediately following gene -i. Figure 1 contains examples of the three encodings.

Genome	Circular gene order and the equivalent reversed representation
A	1 2 3 4 5 6 = -6 -5 -4 -3 -2 -1
B	1 2 -5 -4 3 6 = -6 -3 4 5 -2 -1
C	1 -6 -5 -4 -3 -2 = 2 3 4 5 6 -1

(a) Three signed circular genomes

Genome	Signed genes											
	1	2	3	4	5	6	-1	-2	-3	-4	-5	-6
A	2	3	4	5	6	1	-6	-1	-2	-3	-4	-5
B	2	-5	6	5	-2	1	-6	-1	4	3	-4	-3
C	-6	3	4	5	6	-1	2	1	-2	-3	-4	-5

(b) MPME

Genome	Adjacencies										
	(1,2)	(2,3)	(3,4)	(4,5)	(5,6)	(6,1)	(2,-5)	(-4,3)	(3,6)	(1,-6)	(-2,1)
A	1	1	1	1	1	1	0	0	0	0	0
B	1	0	0	1	0	1	1	1	1	0	0
C	0	1	1	1	1	0	0	0	0	1	1

(c) MPBE

Fig. 1. Examples of the two sequence-style encodings of genome rearrangements, MPBE and MPME. See Section 1 for details.

One can regard MPME and MPBE as relaxed versions of the original breakpoint phylogeny problem, since if two MPME (MPBE) strings correspond to actual gene orders, their Hamming distance is exactly twice the breakpoint distance between the corresponding gene orders. Bryant showed in [5] that the MPME score of any binary tree T is a tighter lower bound of the breakpoint length of T than the MPBE score of T, and in our previous study [26] MPME yields more accurate phylogenies than MPBE. Therefore we will focus on MPME in this paper.

Outline of the paper. In the next section, we present our update for the GRAPPA algorithm using MPME information. We briefly review the GRAPPA algorithm for breakpoint and inversion phylogenies, and the Fitch algorithm for maximum parsimony. We then present our modifications to the GRAPPA algorithm by showing (1) how to compute a closest gene order to any MPME sequence, and (2) how to choose an MPME string out of all possible choices for each internal node. In section three, we evaluate our new algorithm using simulated datasets. Finally, we conclude our paper with discussions and future research directions.

2 Finding Tree Length Using MPME Information

In this section we present our improvements for GRAPPA by incorporating MPME.

2.1 Tree length computations in GRAPPA

We begin by reviewing the algorithm for computing the length of any single phylogeny in GRAPPA. Given any phylogeny T, GRAPPA computes the breakpoint length of T in the following manner:

1. For each internal node in T, assign a gene order using some initialization procedures.
2. Repeat the following procedure until no improvements can be made:
   (a) Pick an internal node x in T with neighbors a, b, and c, solve the median problem of G_a, G_b, G_c to yield G_M.
   (b) If the new median improves the tree score, then assign G_M to x.

3. Return the tree score.

Two median solvers are available in GRAPPA: the breakpoint (BP) median solver uses the strategy of reduction to TSP devised by Sankoff [2]; whereas the inversion (INV) median solver developed by Caprara [6] is based on an extension of the breakpoint graph. Internal genomes can be initialized trivially, by giving each internal node a random gene order. However, other complex procedures yield better results. So far, the best initialization method is *Nearest Neighbor*, which assigns each internal node the median solution from its three nearest leaves, using either BP or INV median solver.

First proposed in [23], this approach is a hill-climbing greedy algorithm. Thus, depending on how we assign gene orders to internal nodes initially and in what order we update the internal nodes, the algorithm could be trapped at some local optimum. As we will show in our simulation studies, this is often the case, and can have an adverse effect on the accuracy of reconstructed trees.

2.2 The new algorithm

Our strategy for improvement is to use the "globalness" of the optimal internal node strings in MPME, so we have better initial gene orders for internal nodes in the original GRAPPA algorithm. Since optimal internal node MPME strings constitute a global optimum for a relaxed version of the breakpoint phylogeny, an intuitive approach is to use a set of closest gene orders (not necessarily unique) to each of the optimal internal node strings in the MPME parsimony. Hopefully by doing this we (1) obtain an assignment that leads to a lower tree score, (2) find the length of the trees faster because of the better initialization, and (3) obtains more accurate phylogenies due to better tree length computations.

We need to address two issues in our new algorithm: firstly, given any MPME string X, we need to find a gene order closest to X. Secondly, the optimal MPME strings for internal nodes may not be unique (there is ambiguity).

Finding a closest gene order to an MPME string. We define the following problems.

Definition 1. *(***MPBE-CGO** *and* **MPME-CGO***) Given an MPBE (MPME) string X for n genes, find a gene order G of the n genes such that the Hamming distance between MPBE(G) (MPME(G)) and X is minimized.*

We do not know the complexity of MPME-CGO, though MPBE-CGO is NP-complete (by reduction from the Hamiltonian Cycle problem, proof omitted).

Theorem 1. *The* **MPBE-CGO** *problem is NP-complete.*

Since any MPME string can be translated into an MPBE string (but not vice versa), solving the MPBE-CGO problem will be sufficient. We will reduce the MPBE-CGO problem to an instance of the Traveling Salesperson Problem (TSP) [12]. We create the graph $K(X)$ with vertices $\{\pm 1, \ldots, \pm n\}$ as follows:

1. There is a *gene* edge between $+i$ and $-i$, $1 \leq i \leq n$.
2. For any adjacency (a, b) such that the state in X is 1, we add an *adjacency* edge to $K(X)$ between $-a$ and b.

We now create a complete graph $K'(X)$ based on $K(X)$.

1. Let $V(K'(X)) = \{\pm 1, \ldots, \pm n\}$.
2. We set the cost for each gene edge $(+i, -i)$ to be some large negative number $-n^2$, $1 \leq i \leq n$.
3. For each adjacency edge in $K(X)$, we set the cost of the same edge in $K'(X)$ to be 0.
4. For the rest of the edges in $K'(X)$, we set the cost to be 1.

We have the following result (proof omitted):

Theorem 2. *Any solution to the TSP problem with $K'(X)$ as input corresponds to a solution for the `MPBE-CGO` problem (and the `MPME-CGO` problem). Let the cost for the TSP problem be c; then the cost for the `MPBE-CGO` problem (and the `MPME-CGO` problem) is $|X| - n + 2c$, where $|X|$ is the number of 1's in X, and n is the number of genes.*

Removing ambiguity in an `MPME` string. We first review the Fitch algorithm [11] for solving the parsimony score for a fixed tree T. Assume X is the set of strings of length k; let A be the alphabet. Let T be the (unrooted) tree topology to be scored, and let T' be the resulting rooted tree by first bisecting an edge of T, then setting the new node as the root.

Our goal is to construct the optimal assignment f for all internal nodes of T. Let d be the Hamming distance between any two strings of X; one can show that in this case, no matter how we root the T, the score will always be the same. Moreover, we can construct f independently for each site in the strings. Let $f_j(i)$ be the state of the j'th site in $f(i)$, $1 \leq i \leq 2m-2$, $1 \leq j \leq k$. The Fitch algorithm is divided into two stages:

1. Initially, we set $B_j(i) = \{(s_i)_j\}, 1 \leq j \leq k, 1 \leq i \leq m$ ($(s_i)_j$ is the state of the j'th site in s_i).
2. In the first (also called forward) stage, we recursively compute $B_j(i)$ for all internal nodes i: let u and v be the two child nodes of i in T', then
$$B_j(i) = \begin{cases} B_j(u) \cap B_j(v), & \text{if } B_j(u) \cap B_j(v) \neq \emptyset \\ B_j(u) \cup B_j(v), & \text{otherwise} \end{cases}$$
A postorder (bottom-up) traversal of T' suffices.
3. In the second (also called backward) stage, we can compute $f_j(i)$ (by no means exhaustive) for all internal nodes i through a preorder (top-down) traversal. If i is the root of T', then choose an element from $B_j(i)$ as $f_j(i)$. Otherwise, let u be the parent node of i.
   (a) If $f_j(u) \in B_j(i)$ then set $f_j(i) = f_j(u)$.
   (b) Otherwise, choose an element from $B_j(i)$ to be $f_j(i)$.

We see f need not be unique. Moreover, since we treat the sites independently, it is possible that $f_j(i) = f_{j'}(i)$ for some node i and sites $j \neq j'$. When X is the set of `MPME` sequences, $f(i)$ does not correspond to a gene order if and only if the above happens.

```
For each topology $T$, arbitrarily root $T$ to obtain $T'$.
  1. Forward stage: compute $B_j(i)$, the set of candidate states, for each internal
     node $i$ and site $j$ (as in the forward stage of the Fitch algorithm).
  2. Backward stage: compute $f(i)$ for each internal node $i$ of $T'$ in preorder:
     (a) Set $P = \emptyset$.
     (b) For $j = 1$ to $2n$ do
           i. Let $Q_j(i) = B_j(i) - P$.
          ii. If $Q_j(i) \neq \emptyset$, choose $f_j(i)$ from $B_j(i) - P$, and add $f_j(i)$ to $P$.
         iii. Otherwise choose $f_j(i)$ randomly from $B_j(i)$.
  3. Convert the MPME string $f(i)$ for each internal node $i$ to the closest gene order
     by reduction to TSP.
  4. Update the gene orders for the internal nodes by repeatedly invoking median
     solvers (as in the original GRAPPA), until no improvement can be made.
Return the set of phylogenies $\{T\}$ having the lowest tree length.
```

Fig. 2. The proposed `GRAPPA/MPME` algorithm.

We propose the following heuristic for disambiguation. To score any (arbitrarily-rooted) tree T and find `MPME` sequences for the ancestral `MPME` strings, we compute the forward stage for every one of the $2n$ sites. We then compute $f(i)$ for each internal node in a preorder traversal (backward stage) by considering all sites at once. We iteratively determine the state for $f(i)$: for each site j, when we determine its state, we will mark the corresponding gene $f_j(i)$ as used. When we determine the state of the next site $j+1$, we will randomly pick a state from the list of possible states $B_{j+1}(i)$ and check if the corresponding gene is used. If that gene is used, we will randomly pick another state from the possible states. This is a heuristic and cannot guarantee to remove the ambiguity, but in our experience using this heuristic gives us better gene order assignments for the internal nodes. Please see Figure 2 for details.

3 Evaluation

In this section we evaluate our algorithm through three simulation studies. We first define our measure for the accuracy of reconstructed trees, which will be used in Studies 2 and 3. Given an inferred tree, we compare its "topological accuracy" by computing "false negatives" with respect to the "true tree" [13]. During the evolutionary process, some edges of the model tree may have no change (i.e. evolutionary events) on them. Since reconstructing such edges is at best guesswork, we are not interested in these edges. Hence, we define the true tree to be the result of *contracting* those edges in the model tree on which there is no change.

For every tree there is a natural association between every edge and the bipartition on the leaf set induced by deleting the edge from the tree. Let T be the true tree and let T' be the inferred tree. An edge e in T is "missing" in T' if T' does not contain an edge defining the same bipartition; such an edge is called a *false negative* (FN). Note that external edges (i.e. edges incident to a leaf) are trivial in the sense that they are present in every tree with the same set of leaves. The *false negative rate* is the number of false negative edges in T' with respect to T divided by the number of internal edges in T. The *false positive*

Q25%/Median/Q75% (corr)	b=10	b=20
a=10	6 / 9 / 13 (0.866)	19 / 23 / 27 (0.846)
a=20	8 / 11 / 15 (0.771)	22 / 25 / 30 (0.717)
a=40	11 / 14.5 / 19 (0.690)	26.75 / 31 / 36 (0.588)
a=60	14 / 18 / 22 (0.605)	28 / 35 / 39.5 (0.608)

Fig. 3. Results of Simulation Study 1. The underlying simulation model is GNT with weights (50% inversions and 50% transpositions). The model tree has one root and three leaves; one external edge has length a, and the other two have length b. Each cell in the table begins with three numbers separated by slashes – the 25% quantile/Median/75% quantile of the breakpoint distance between G_M (the closest gene order to the MPME sequence of the internal node) and G_0 (the true median), and then followed by the correlation (see Section 3.1 for the definition of the correlation).

(FP) rate is defined similarly but with T and T' swapped. The *Robinson-Foulds* (RF) rate is defined as the average of the FN and FP rates.

3.1 Study 1: MPME sequence as a median solver

We first examine the potential of MPME-CGO in finding good internal gene orders. In our simulation study, each genome has 100 genes. We create each dataset by first generating a tree topology with three leaves, assigning it edge lengths (one edge has length a, and the other two have identical lengths b). We assign the identity permutation 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, with the operations chosen according to the model of gene-order evolution (in our case, the Generalized *Nadeau-Taylor* (GNT) model where inversions, transpositions, and inverted transpositions all occur). We then compute an optimal MPME string M and its closest gene order G_M, and compute the correlation between $d(M, \texttt{MPME}(G_M))$ (the Hamming distance between M and MPME(G_M)), and BP(G_M,G_0) (the breakpoint distance between G_M and G_0). For each setting we repeat 50 times, and compute the correlation between the two distances, and the quartiles of BP(G_M,G_0). The results are in Figure 3.

We make two observations. First, the idea of using MPME-CGO does not give us a good median solver: in no case does the closest gene order G_M agree with G_0, though the error (median of BP(G,G_0)) seems to be linearly correlated with the edge length a when $b = 10$ and $b = 20$. However, we do see the error of MPME-CGO is highly correlated with $d(M, \texttt{MPME}(G_M))$ (which we can always compute without knowledge of the actual median gene order). Therefore we can think of $d(M, \texttt{MPME}(G_M))$ as the *quality* of the MPME string: the smaller $d(M, \texttt{MPME}(G_M))$ is, the more accurate G_M is as an estimate to the actual median.

Though the idea of MPME-CGO as the median solver does not fare well, the next two simulation studies show that our new algorithm consistently outperforms the original GRAPPA in both scoring a tree topology and finding the best tree.

3.2 Study 2: GRAPPA/MPME on Uniformly Random Trees

We test the combinations of three initializations (MPME, BP, INV) and two median solvers (BP, INV) on uniformly random trees with 12 genomes (datasets of that

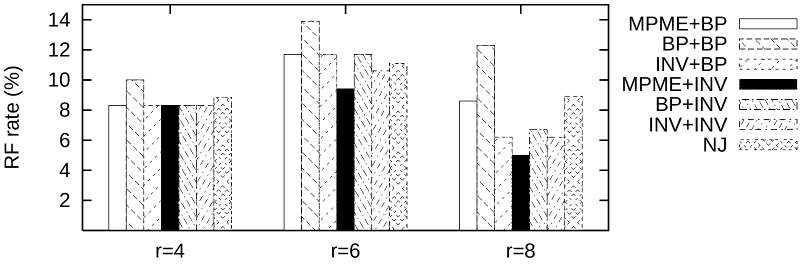

Fig. 4. Robinson-Foulds (RF) rate of each method on uniform trees (`X+Y` means we use `X` to initialize and use `Y` to score a tree).

size form the bulk of the subproblems solved in the DCM approach to reconstruction from gene-order data when working on datasets of 1,000 genomes [24]) and chose genomes of 100 genes (the size range of chloroplast genomes). We generate uniformly random tree by randomly picking a tree from all possible trees—there are $(2N-5) \times (2N-7) \times \cdots \times 3$ trees for N taxa. The number of events on each edge is sampled from a uniform distribution on the set $\{0, 2, \ldots, 2.5r\}$, where r is the expected evolutionary rate. We choose $r = 4$, 6 and 8 to test these methods on datasets with various level of difficulty, where $r = 8$ is considered very hard for gene-order data. We use a mix of 80% inversions and 20% transpositions. For each way of setting the various parameters, we run 20 datasets and averaged the results. For each dataset, we use the `BP` and `INV` median solvers and `MPME` to initialize the gene orders on all internal nodes. However, since `MPME-CGO` is not a reliable median solver, we only use `BP` and `INV` median solvers to compute the score of a tree. We also examine `NJ(EDE)`, the best neighbor joining tree for genome rearrangement data[16].

For all datasets we test, the final tree scores rely only on the scoring method and the scores are always identical regardless of which initialization method is used. However, the final tree topologies depend on both the initialization and scoring methods (Fig 4). Among all these methods (including Neighbor-joining as the baseline), `MPME+INV` (using `MPME` to initialize and `INV` to score) is the most accurate, whereas `BP`+`BP` gives the worst result. Although `MPME-CGO` is not good at solving medians, using the inherent global information to initialize the internal nodes avoids the trap of local optima, hence the higher accuracy.

Running time is similar for all datasets, roughly in the range of 1 to 10 minutes for all methods on a Pentium-4/Linux workstation. The speed of `GRAPPA` is determined by both the median solver and the bounding technique to discard bad trees [15]. Although there is almost no difference in the final tree score, most of the time using `MPME+INV` lowers the score of the NJ tree, hence better initial upper bound for pruning subsequent trees in `GRAPPA`; as a result, `MPME+INV` is the fastest among all combinations.

3.3 Study 3: GRAPPA/MPME on Yule-Harding Trees

We use the r8s software [22] to generate Yule-Harding trees. We then multiply each edge length by a factor s drawn from a random distribution (where $\ln s$

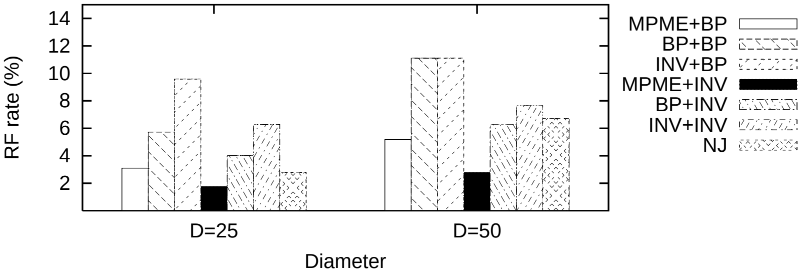

Fig. 5. Robinson-Foulds (RF) rate of each method on Birth-Death trees (X+Y means we use X to initialize and use Y to score a tree).

is uniformly distributed between $-\ln 1.5$ and $\ln 1.5$) to deviate the tree from ultrametricity. We test only two diameters (25 and 50); no method (except NJ) can finish for diameters larger than 75. Although these diameters seem small, the Yule-Harding trees we generate are quite difficult because these trees have highly diversified edge lengths. For example, for all datasets with diameter 25, there is at least one edge with more than 40 events, about 2 edges with $15-40$ events, but more than 10 edges with fewer than 4 events. This diversity presents great difficulty for all reconstruction methods based on median computation; nonetheless, GRAPPA using MPME initialization is still the most accurate among all the methods we test (including NJ).

Figure 5 shows the comparison of RF rates of these methods. The best initialization/median-solver combination is MPME+INV: when $D = 50$, its RF rate is about half of the RF rates of the second-best combination, MPME+BP. Both MPME+INV and MPME+BP have lower RF rates than other methods (except that NJ is slightly better than MPME+BP when $D = 25$). Another interesting observation is, if we fix the median solver (INV or BP), the order of initializations in terms of RF rate is always MPME $<$ BP $<$ INV (BP =INV when $D = 50$ and we use BP median solver). MPME is the best initialization as expected, but it is surprising BP outperforms INV, despite past experience that inversion phylogeny is a more accurate criterion than breakpoint phylogeny.

Unlike the experiments on uniform trees, the running time is highly variable and ranges from 1 minute to about a day. MPME+INV is about 2–3 times slower compared to the fastest method (BP+INV). However, the higher accuracy and shorter tree lengths make the additional time worthwhile.

4 Discussion

In this paper we proposed the use of MPME, a sequence-style heuristic for breakpoint phylogeny, to improve the accuracy of genome rearrangement phylogeny. We find that GRAPPA, the best software for inversion and breakpoint phylogenies to date, suffers from the problem of multiple local optima. To address this problem, we propose the use of MPME, a string encoding of gene adjacency relationships whose optimal internal node assignments can be determined globally in polynomial time, to provide better initializations of GRAPPA.

We discussed the problems of finding closest gene orders to MPME and MPBE strings (MPBE-CGO and MPME-CGO), and showed how to solve the problems by reduction to TSP. We then showed how we can use the inherent "globalness" of MPME to better initialize the GRAPPA tree length computation algorithm. Though MPME-CGO does not give us good median solvers, MPME-CGO assigns good initial gene orders to internal nodes in GRAPPA, thereby avoiding being trapped at some inferior local optimum. In simulation, the breakpoint and inversion scores of the best tree almost always improve, and the error of the most parsimonious tree dropped by up to 50%.

The next step for our research is to improve the our algorithm so it is faster and yields lower tree scores. First, it is desirable to find more efficient approaches for MPME-CGO other than using TSP, as well as find out the computational complexity of the problem. Moreover, we use a simple greedy, randomized heuristic to choose optimal MPME strings for all internal nodes. Fine tuning the algorithm may further improve its accuracy and computational efficiency. We will also extend MPME to cope with more evolutionary events, such as deletions and duplications. This improvement, along with our extension of GRAPPA for unequal gene content [25], will eventually give us an accurate tool to analyze datasets with arbitrary gene content.

References

[1] D.A. Bader, B.M.E. Moret, and M. Yan. A fast linear-time algorithm for inversion distance with an experimental comparison. *J. Comput. Biol.*, 8(5):483–491, 2001.

[2] M. Blanchette, G. Bourque, and D. Sankoff. Breakpoint phylogenies. In S. Miyano and T. Takagi, editors, *Genome Informatics 1997*, pages 25–34. Univ. Academy Press, 1997.

[3] M. Blanchette and D. Sankoff. The median problem for breakpoints in comparative genomics. In *Proc. 3rd Int'l Conf. Computing and Combinatorics (COCOON'97)*, volume 1276 of *Lecture Notes in Computer Science*, pages 251–263. Springer Verlag, 1997.

[4] J.L. Boore, T. Collins, D. Stanton, L. Daehler, and W.M. Brown. Deducing the pattern of arthropod phylogeny from mitochondrial DNA rearrangements. *Nature*, 376:163–165, 1995.

[5] D. Bryant. A lower bound for the breakpoint phylogeny problem. In R. Giancarlo and D. Sankoff, editors, *Proc. 11th Ann. Symp. Combinatorial Pattern Matching CPM'00*, pages 235–247. Springer, 2000.

[6] A. Caprara. On the practical solution of the reversal median problem. In *Proc. 1st Int'l Workshop Algs. in Bioinformatics (WABI'01)*, volume 2149 of *Lecture Notes in Computer Science*, pages 238–251. Springer Verlag, 2001.

[7] M.E. Cosner, R.K. Jansen, B.M.E. Moret, L.A. Raubeson, L. Wang, T. Warnow, and S.K. Wyman. An empirical comparison of phylogenetic methods on chloroplast gene order data in Campanulaceae. In D. Sankoff and J.H. Nadeau, editors, *Comparative Genomics*, pages 99–122. Kluwer Academic Publishers, 2000.

[8] M.E. Cosner, R.K. Jansen, B.M.E. Moret, L.A. Raubeson, L.-S. Wang, T. Warnow, and S.K. Wyman. A new fast heuristic for computing the breakpoint phylogeny and a phylogenetic analysis of a group of highly rearranged chloroplast genomes. In *Proc. 8th Int'l Conf. on Intelligent Systems for Mol. Biol. (ISMB'00)*, pages 104–115, 2000.

[9] W.H.E. Day. Computationally difficult parsimony problems in phylogenetic systematics. *Journal of Theoretical Biology*, 103:429–438, 1983.

[10] S.R. Downie and J.D. Palmer. Use of chloroplast DNA rearrangements in reconstructing plant phylogeny. In P. Soltis, D. Soltis, and J.J. Doyle, editors, *Plant Molecular Systematics*, pages 14–35. Chapman and Hall, 1992.

[11] W.M. Fitch. Toward defining the course of evolution: minimum change for a specific tree topology. *Systematic Zoology*, 20:406–416, 1971.

[12] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NPCompleteness*. Freeman Publisher, 1979.

[13] S. Kumar. Minimum evolution trees. *Mol. Biol. Evol.*, 15:584–593, 1996.

[14] B.M.E. Moret, A.C. Siepel, J. Tang, and T. Liu. Inversion medians outperform breakpoint medians in phylogeny reconstruction from gene-order data. In *Proc. 2nd Int'l Workshop Algs. in Bioinformatics (WABI'02)*, volume 2452 of *Lecture Notes in Computer Science*, pages 521–536. Springer Verlag, 2002.

[15] B.M.E. Moret, J. Tang, L.-S. Wang, and T. Warnow. Steps toward accurate reconstructions of phylogenies from gene-order data. *J. Comput. Syst. Sci.*, 65(3):508–525, 2002.

[16] B.M.E. Moret, L.-S. Wang, T. Warnow, and S.K. Wyman. New approaches for reconstructing phylogenies from gene-order data. In *Proc. 9th Int'l Conf. on Intelligent Systems for Mol. Biol. (ISMB'01)*, volume 17 of *Bioinformatics*, pages S165–S173, 2001.

[17] B.M.E. Moret, S.K. Wyman, D.A. Bader, T. Warnow, and M. Yan. A new implementation and detailed study of breakpoint analysis. In *Proc. 6th Pacific Symp. on Biocomputing (PSB'01)*, pages 583–594. World Scientific Pub., 2001.

[18] J.D. Palmer. Chloroplast and mitochondrial genome evolution in land plants. In R. Herrmann, editor, *Cell Organelles*, pages 99–133. Springer Verlag, 1992.

[19] L.A. Raubeson and R.K. Jansen. Chloroplast DNA evidence on the ancient evolutionary split in vascular land plants. *Science*, 255:1697–1699, 1992.

[20] A. Rokas and P.W.H. Holland. Rare genomic changes as a tool for phylogenetics. *Trends in Ecol. and Evol.*, 15:454–459, 2000.

[21] N. Saitou and M. Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, 4:406–425, 1987.

[22] M. Sanderson. `r8s` software package. http://loco.ucdavis.edu/r8s/r8s.html.

[23] D. Sankoff and M. Blanchette. Multiple genome rearrangement and breakpoint phylogeny. *J. Comput. Biol.*, 5:555–570, 1998.

[24] J. Tang and B.M.E. Moret. Scaling up accurate phylogenetic reconstruction from gene-order data. In *Proc. 11th Int'l Conf. on Intelligent Systems for Mol. Biol. (ISMB'03)*, volume 19 of *Bioinformatics*, pages i305–i312. Oxford U. Press, 2003.

[25] J. Tang, B.M.E. Moret, L. Cui, and C.W. dePamphilis. Phylogenetic reconstruction from arbitrary gene-order data. In *Proc. 4th IEEE Symp. on Bioinformatics and Bioengineering BIBE'04*, pages 592–599. IEEE Press, Piscataway, NJ, 2004.

[26] L.-S. Wang, R.K. Jansen, B.M.E. Moret, L.A. Raubeson, and T. Warnow. Fast phylogenetic methods for genome rearrangement evolution: An empirical study. In *Proc. 7th Pacific Symp. on Biocomputing (PSB'02)*, pages 524–535. World Scientific Pub., 2002.

[27] L.-S. Wang and T. Warnow. Estimating true evolutionary distances between genomes. In *Proc. 33rd Ann. ACM Symp. Theory of Comput. (STOC'01)*, pages 637–646. ACM Press, New York, 2001.