Steps Toward Accurate Reconstructions of Phylogenies from Gene-Order Data[1]

Bernard M.E. Moret†, Jijun Tang†, Li-San Wang‡, and Tandy Warnow‡

†*Department of Computer Science, University of New Mexico*
Albuquerque, NM 87131 USA
`moret,jtang@cs.unm.edu`
‡*Department of Computer Sciences, University of Texas*
Austin, TX 78712 USA
`lisan,tandy@cs.utexas.edu`

We report on our progress in reconstructing phylogenies from gene-order data. We have developed polynomial-time methods for estimating genomic distances that greatly improve the accuracy of trees obtained using the popular neighbor-joining method; we have also further improved the running time of our GRAPPA software suite through a combination of tighter bounding and better use of the bounds. We present new experimental results (that extend those we presented at ISMB'01 and WABI'01) that demonstrate the accuracy and robustness of our distance estimators under a wide range of model conditions. Moreover, using the best of our distance estimators (EDE) in our GRAPPA software suite, along with more sophisticated bounding techniques, produced spectacular improvements in the already huge speedup: whereas our earlier experiments showed a one-million-fold speedup (when run on a 512-processor cluster), our latest experiments demonstrate a speedup of one hundred million. The combination of these various advances enabled us to conduct new phylogenetic analyses of a subset of the *Campanulaceae* family, confirming various conjectures about the relationships among members of the subset and confirming that inversion can be viewed as the principal mechanism of evolution for their chloroplast genome. We give representative results of the extensive experimentation we conducted on both real and simulated datasets in order to validate and characterize our approaches.

Key Words: ancestral genome, breakpoint analysis, combinatorial optimization, distance estimator, evolutionary distance, experimental analysis, genome rearrangement, inversion distance, layered search, lower bounding, neighbor-joining

[1] Portions of this paper were presented at the Ninth International Symposium on Intelligent Systems for Molecular Biology (ISMB'01) in Copenhagen, published in *Bioinformatics* **17** (2001), S165–S173, and at the First Workshop on Algorithms in Bioinformatics (WABI'01) in Århus, published in *Lecture Notes in Computer Science* **2149** (2001), 176–190.

1. INTRODUCTION

Genome rearrangements. Modern laboratory techniques can yield the ordering and strandedness of genes on a chromosome, allowing us to represent each chromosome by an ordering of signed genes (where the sign indicates the strand). Evolutionary events can alter these orderings through rearrangements such as inversions and transpositions, collectively called genome rearrangements. Because these events are rare, they give us information about ancient events in the evolutionary history of a group of organisms. In consequence, many biologists have embraced this new source of data in their phylogenetic work [16, 26, 27, 29]. Appropriate tools for analyzing such data remain primitive when compared to those developed for DNA sequence data; thus developing such tools is becoming an important area of research, as attested by recent meetings on this topic [14, 15].

Optimization problems. A natural optimization problem for phylogeny reconstruction from gene-order data is to reconstruct an evolutionary scenario with a minimum number of the permitted evolutionary events on the tree—what is known as a *most parsimonious tree*. Unfortunately, this problem is NP-hard for most criteria—even the very simple problem of computing the median of *three* genomes under such models is NP-hard [9, 28]. However, because suboptimal solutions can yield very different evolutionary reconstructions, exact solutions are strongly preferred over approximate solutions (see [33]). Moreover, the relative probabilities of each of the rearrangement events (inversions, transpositions, and inverted transpositions) are difficult to estimate. To overcome the latter problem, Blanchette *et al.* [5] have proposed using the *breakpoint phylogeny*, the tree that minimizes the total number of *breakpoints*, where a breakpoint is an adjacency of two genes that is present in one genome but not in its neighbor in the tree. Note that constructing the breakpoint phylogeny remains NP-hard [7].

Methods for reconstructing phylogenies. Blanchette *et al.* developed the `BPAnalysis` [31] software, which implements various heuristics for the breakpoint phylogeny. We reimplemented and extended their approach in our `GRAPPA` [17] software, which runs several orders of magnitude faster thanks to algorithm engineering techniques [24]. Other heuristics for solving the breakpoint phylogeny problem have been proposed [8, 12, 13].

Rather than attempting to derive the most parsimonious trees (or an approximation thereof), we can use existing distance-based methods, such as *neighbor-joining (NJ)* [30] (perhaps the most popular phylogenetic method), in conjunction with methods for defining leaf-to-leaf distances in the phylogenetic tree. Leaf-to-leaf distances that can be computed in linear time currently include *breakpoint distances* and *inversion distances* (the latter thanks to our new algorithm [3]). We can also estimate the "true" evolutionary distance (or, rather, the expected true evolutionary distance under a specific model of evolution) by working backwards from the breakpoint distance or the minimum inversion distance, an approach suggested by Sankoff [32] and Caprara [10] and developed by us in a series of papers [23, 34, 35], in which we showed that these estimators significantly improve the accuracy of trees obtained using the neighbor-joining method.

Results in this paper. This paper reports new experimental results on the use of our distance estimators in reconstructing phylogenies from gene-order data, using both simulated and real data. We present several new results (the first two are extensions of the results we presented at ISMB'01 [23] and the third an extension of the results we presented at WABI'01 [34]):

• Simulation studies examining the relationship between the true evolutionary distance and our distance estimators. We find that our distance estimators give very good predictions of the actual number of events under a variety of model conditions (including those that did not match the assumptions).

• Simulation studies examining the relationship between the topological accuracy of neighbor-joining and the specific distance measure used: breakpoint, inversion, or one of our three distance estimators. We find that neighbor-joining does significantly better with our distance estimators than with the breakpoint or inversion distances.

• A detailed investigation of the robustness of neighbor-joining using our distance estimators when the assumed relative probabilities of the three rearrangement events are very different from the true relative probabilities. We find that neighbor-joining using our estimators is remarkably robust, hardly showing any worsening even under the most erroneous assumptions.

• A detailed study of the efficacy of using our best distance estimator, significantly improved lower bounds (still computable in low polynomial time) on the inversion length of a candidate phylogeny, and a novel way of structuring the search so as to maximize the use of these bounds. We find that this combination yields much stronger bounding in the naturally occurring range of evolutionary rates, yielding an additional speedup by one to two orders of magnitude for our `GRAPPA` code.

• A successful analysis of a dataset of *Campanulaceae* (bluebell flower) using a combination of these techniques, resulting in a *one-hundred-million-fold* speedup over the original approach—in particular, we were able to analyze the dataset on a single workstation in a few hours, whereas our previous analysis required the use of a 512-node supercluster.

Our research combines the development of mathematical techniques with extensive experimental performance studies. We present a cross-section of the results of the experimental study we conducted to characterize and validate our approaches. We used a large variety of simulated datasets as well as several real datasets (chloroplast and mitochondrial genomes) and tested speed (in both sequential and parallel implementations), robustness (in particular against mismatched models), efficacy (for our new bounding technique), and accuracy (for reconstruction and distance estimation).

2. BACKGROUND

2.1. The Nadeau-Taylor model of evolution

When each genome has the same set of genes and each gene appears exactly once, a genome can be described by an ordering (circular or linear) of these genes, each gene given with an orientation that is either positive (g_i) or negative ($-g_i$).

Let G be the genome with signed ordering $g_1, g_2, \ldots, g_k$. An *inversion* between indices a and b, for $a \leq b$, produces the genome with linear ordering

$$g_1, g_2, \ldots, g_{a-1}, -g_b, -g_{b-1}, \ldots, -g_a, g_{b+1}, \ldots, g_k$$

A *transposition* on the (linear or circular) ordering G acts on three indices, a, b, c, with $a \leq b$ and $c \notin [a, b]$, picking up the interval $g_a, g_{a+1}, \ldots, g_b$ and inserting it immediately after g_c. Thus the genome G above (with the assumption of $c > b$) is replaced by

$$g_1, \ldots, g_{a-1}, g_{b+1}, \ldots, g_c, g_a, g_{a+1}, \ldots, g_b, g_{c+1}, \ldots, g_k$$

An *inverted transposition* is a transposition followed by an inversion of the transposed subsequence.

The (generalized) *Nadeau-Taylor* model [25] of genome evolution uses only genome rearrangement events, so that all genomes retain equal gene content. 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 Nadeau-Taylor model tree as a triplet $(T, \{\lambda_e\}, (\gamma_I, \gamma_T, \gamma_{IT}))$, where the the triplet $(\gamma_I, \gamma_T, \gamma_{IT})$ defines the relative probabilities of the three types of events (inversions, transpositions, and inverted transpositions). For instance, the triplet $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ indicates that the three event classes are equiprobable, while the triplet $(1, 0, 0)$ indicates that only inversions happen.

2.2. Distance-based estimation of phylogenies

Given a tree T on a set S of genomes and given any two leaves i, j in T, we denote by $P(i, j)$ the path in T between i and j. We let λ_e denote the number of events (inversions, transpositions, or inverted transpositions) on the edge e during the evolution of the genomes in S within the tree T. This is the *actual* number of events on the edge. We can then define the matrix $[\lambda_{ij}]$ of actual distances, $\lambda_{ij} = \sum_{e \in P(i,j)} \lambda_e$, which is additive. When given an additive matrix, many distance-based methods are guaranteed to reconstruct the tree T and the edge weights (but not the root). Atteson [1] showed that NJ is guaranteed to reconstruct the true tree T when given an estimate of the additive matrix $[\lambda_{ij}]$, as long as the estimate has bounded error:

THEOREM 2.1. *(From [1]) Let T be a binary tree and let λ_e and λ_{ij} be defined as described above. Let $x = \min_{e \in E(T)} \lambda_e$. Let D be any $n \times n$ dissimilarity matrix (i.e. D is symmetric and zero on the diagonal). If*

$$\max_{\{i,j\} \subset L(T)} |D_{ij} - \lambda_{ij}| < \frac{x}{2},$$

then the NJ tree, $NJ(D)$, computed for D is identical to T.

That is, NJ is guaranteed to reconstruct the true tree topology if the input distance matrix is sufficiently close to an additive matrix defining the same tree topology. Consequently, techniques that yield a good estimate of the matrix $[\lambda_{ij}]$ are of significant interest.

Distance measures. The *edit distance* between two gene orders is the minimum number of inversions, transpositions, and inverted transpositions needed to transform one gene order into the other. The *inversion distance* is the edit distance when only inversions are permitted. The inversion distance can be computed in linear time [3, 18]; the transposition distance is of unknown computational complexity [4].

Given two genomes G and G' on the same set of genes, a *breakpoint* in G is an ordered pair of genes (g_a, g_b) such that g_a and g_b appear consecutively in that order in G, but neither (g_a, g_b) nor $(-g_b, -g_a)$ appear consecutively in that order in G'. The number of breakpoints in G relative to G' is the *breakpoint distance* between G and G'. The breakpoint distance is easily calculated by inspection in linear time. See Figure 1 for an example of these distances.

Estimations of true evolutionary distances. Estimating the true evolutionary distance requires assumption about the model; in the case of gene-order evolution, the assumption

$G_0 =$	1	**2**	**3**	**4**	**5**	6	7	8	9	10
$G_1 =$	1	-5	-4	-3	-2	**6**	**7**	**8**	9	10
$G_2 =$	1	**-5**	**-4**	**-3**	**-2**	**-8**	**-7**	**-6**	9	10
$G_3 =$	1	6	7	8	2	3	4	5	9	10

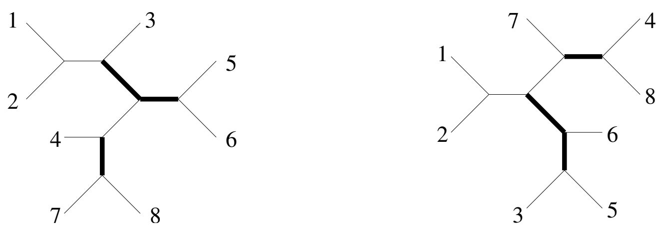
$d_T(G_0, G_3) = 1$ $\qquad$ $d_I(G_0, G_3) = 3$ $\qquad$ $d_B(G_0, G_3) = 3$

FIG. 1. Example of transposition, inversion, and breakpoint distances. We obtain G_3 from G_0 after 3 inversions (the genes in the inversion interval are highlighted at each step). G_3 can also be obtained from G_0 with one transposition: move the gene segment $(6, 7, 8)$ to the position between genes 1 and 3. d_T, d_I, and d_B are the transposition, inversion, and breakpoint distances, respectively.

is that the genomes have evolved from a common ancestor under the Nadeau-Taylor model of evolution. Sankoff's technique [32], applicable only to inversions, calculates this value exactly, while IEBP [35] and EDE [23], applicable to very general models of evolution, obtain approximations of these values, and Exact-IEBP [34] calculates the value exactly for any combination of inversions, transpositions, and inverted transpositions. These estimates can all be computed in low polynomial time.

2.3. Performance criteria

Let T be a tree leaf-labelled by the set S. Deleting some edge e from T produces a bipartition π_e of S into two sets. Let T be the true tree and let T' be an estimate of T, as illustrated in Figure 2. The *false negatives* of T' with respect to T, denoted $FN(T, T')$, are those bipartitions that appear in T that do not appear in T'. The *false negative rate* is the number of false negatives divided by the number of non-trivial bipartitions of T. Similarly, the *false positives* of T' with respect to T are defined as those bipartitions that appear in T' but not in T, and the *false positive rate* is the ratio of false positives to the number of nontrivial edges. For example, in Figure 2, the edge corresponding to the bipartition $\{1, 2, 3 \mid 4, 5, 6, 7, 8\}$ is present in the true tree, but not in the estimate, and is thus a false negative. Note that, if both trees are binary, then the number of false negatives equals the number of false positives. In reporting our results, we will use the false negative rate.



(a) True tree T $\qquad$ (b) Estimate T'

FIG. 2. False positive and false negative edges. T is the true tree, T' is a reconstructed tree, and bold edges are false negative edges in (a) and false positive edges in (b).

3. TRUE DISTANCE ESTIMATORS

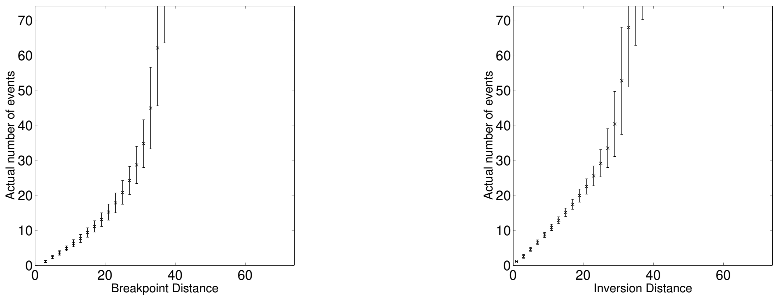
3.1. Definitions

Given two signed permutations, we can compute their breakpoint distance or one of the edit (minimum) distances (for now, the inversion distance), but the actual number of evolutionary events is not directly recoverable. All that can be done is to estimate that number under some assumptions about the model of evolution. Thus our true distance estimators return the *most likely* number of evolutionary events for the given breakpoint or inversion distance. We developed three such estimators: the IEBP estimator [35] approximates the most likely number of evolutionary events working from the breakpoint distance, using a simplified analytical derivation; the Exact-IEBP estimator [34] refines the analytical derivation and returns the exact value for that quantity; and the EDE estimator [23] uses curve fitting to approximate the most likely number of evolutionary events working from the inversion distance. All three estimators provide considerably more accurate estimates of true evolutionary distances than the breakpoint or inversion distances (at least for large distances); moreover, trees obtained by applying the neighbor-joining method to these estimators are more accurate than those obtained also using neighbor-joining, but based upon breakpoint distances or inversion distances.

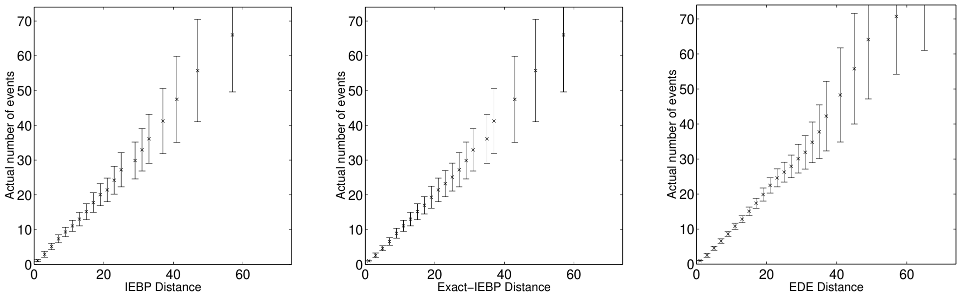
3.2. Comparison of distance estimates

We simulated the Nadeau-Taylor model of evolution under different weight settings to study the behavior of different distance estimators. The numbers of genes in the datasets are 37 (animal mitochondria [6]), and 120 (chloroplast genome in many plants [20]). For each dataset in the experiment, we chose a number between 1 and some upper bound B as the number of rearrangement events. B is chosen to be 2.5 times the number of genes, which (according to our experimental results) is enough to make the distance between two genomes similar to the distance between two random genomes. We then computed the BP (breakpoint) and INV (inversion) distances and corrected them to get IEBP, Exact-IEBP, and EDE distances. In Figures 3, 4, and 5, we plot the (unnormalized) computed distances against the actual number of events—using an inversion-only scenario for the case of 37 genes and a scenario with equally likely events as well as an inversion-only scenario for the case of 120 genes. These figures indicate that, as expected, BP and INV distances underestimate the actual number of events—although, when the number of events is low, they are highly accurate and have small variance. The linear region—the range of the x-coordinate values where the curve is a straight line—is longer for INV distances than for BP distances so that INV distances produce unbiased estimates in a larger range than do BP distances. In contrast, the three estimators provide good estimates on the average, although their variances increase sharply with the edit distance values. Exact-IEBP produces good estimates over all ranges (clearly improving on IEBP), while EDE tends to underestimate the distance unless the scenario uses only inversions.

We also plotted the difference between the actual (true) evolutionary distance and the minimum or estimated distance, under various models of evolution (mixtures of inversions, transpositions, and inverted transpositions), for two different genome sizes (37 and 120), and for various number of events (rates of evolution). Figure 6 shows the results for three different models of evolution on 37 and 120 genes, respectively; the values are plotted in a cumulative fashion: at position x along the horizontal axis, we plotted the mean absolute difference for all generated pairs with a true evolutionary distance of at most x. These figures show that Exact-IEBP is usually the best choice, but that EDE distances are

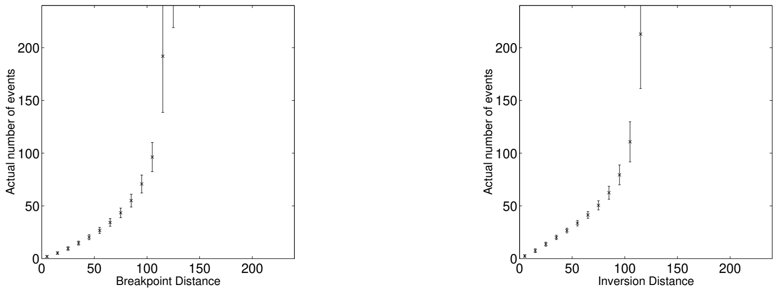


(a) BP distance *(b) INV distance*

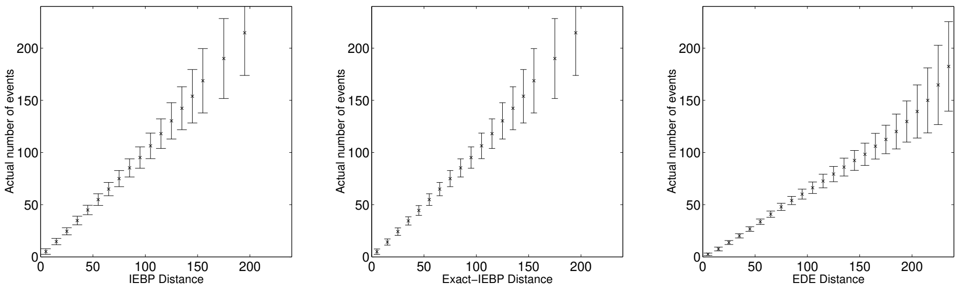


(c) IEBP distance *(d) Exact-IEBP distance* *(e) EDE distance*

FIG. 3. Mean and standard deviation plots for the two distances and three distance estimators, for 37 genes under an inversion-only scenario. The datasets are divided into bins according to their x-coordinate values (the BP or INV distance).

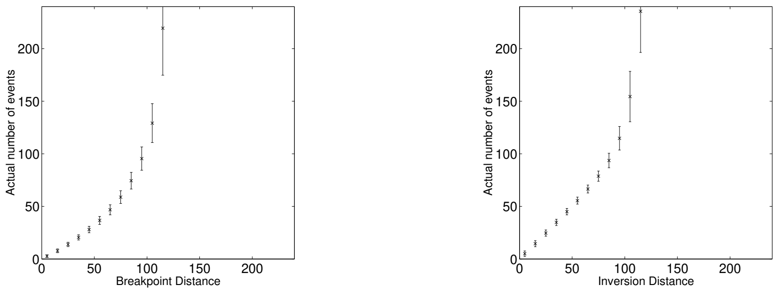


(a) BP distance *(b) INV distance*

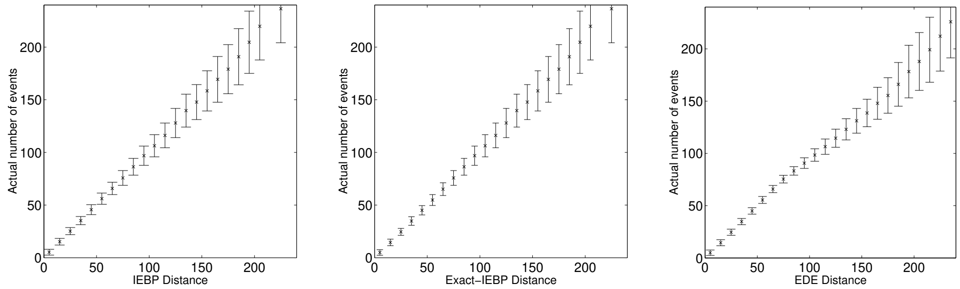


(c) IEBP distance *(d) Exact-IEBP distance* *(e) EDE distance*

FIG. 4. Mean and standard deviation plots for the two distances and three distance estimators, for 120 genes under a scenario in which all three types of events are equally likely. The datasets are divided into bins according to their x-coordinate values (the BP or INV distance).

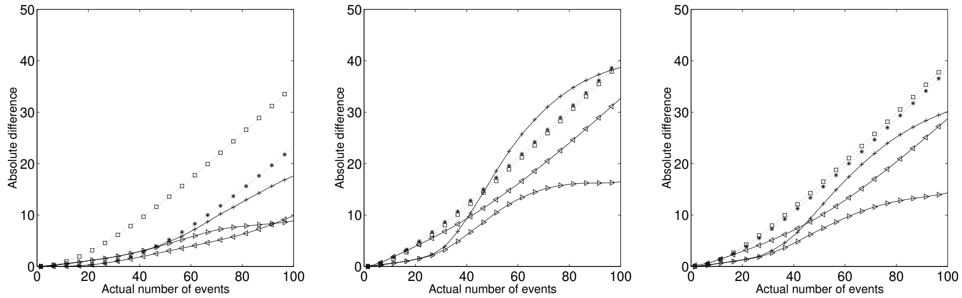


(a) BP distance *(b) INV distance*



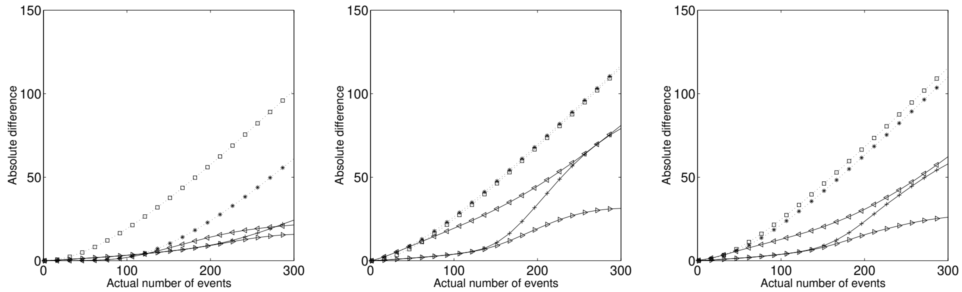
(c) IEBP distance *(d) Exact-IEBP distance* *(e) EDE distance*

FIG. 5. Mean and standard deviation plots for the two distances and three distance estimators, for 120 genes under an inversion-only scenario. The datasets are divided into bins according to their x-coordinate values (the BP or INV distance).



(a) inversions only *(b) transpositions only* *(c) equally likely events*

□ BP * INV + IEBP ▷ Exact-IEBP ◁ EDE



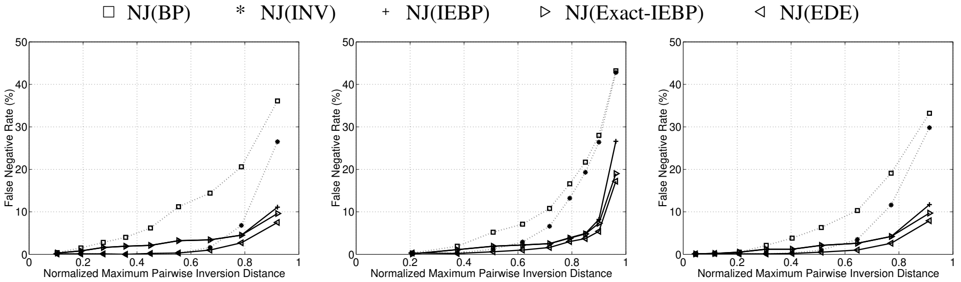
(a) inversions only *(b) transpositions only* *(c) equally likely events*

FIG. 6. The mean difference between the true evolutionary distance and our five distances estimates, under three models of evolution, plotted as a function of the tree diameter, for 37 genes (top) and 120 genes (bottom).

nearly as good (and occasionally better) when the model uses only inversions—although the variance for larger distances is high, making it difficult to draw firm conclusions. IEBP and EDE clearly improve on BP and INV.

3.3. Neighbor-joining performance

We conducted a simulation study to compare the performance of NJ using the same five distances. In Figure 7, we plot the false negative rate against the normalized pairwise inversion distances, under three different model weights settings: $(1, 0, 0)$ (inversion only), $(0, 1, 0)$ (transpositions only), and $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ (all three events equally likely). In each plot we pool the results for the same model weight but different numbers of genomes: 10, 20, 40, 80, and 160. Note that NJ(EDE) is remarkably robust: even though EDE was engineered for an inversion-only scenario, it can handle datasets with a significant number of transpositions and inverted transpositions almost as well. NJ(EDE) recovers 90% of the edges even for the nearly saturated datasets where the maximum pairwise inversion distance is close to 90% of the maximum value. That NJ(EDE) improves on NJ(IEBP), in spite of the fact that IEBP is a comparable estimator, may be attributed to the greater precision (smaller variance) of EDE for smaller distances—most of the choices made in neighbor-joining are made among small distances, where EDE is more likely to return an approximation within a small factor of the true distance. Exact-IEBP, the most expensive of our three estimators to compute, yields the second best performance, although the difference between the error rates of NJ(EDE) and NJ(Exact-IEBP) is too small to be statistically significant.

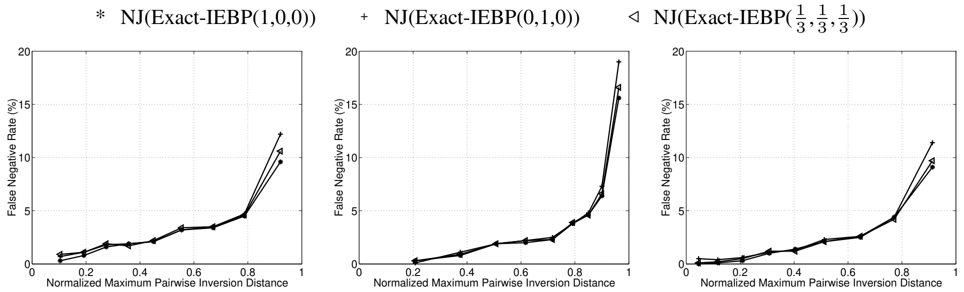


(a) inversions only *(b) transpositions only* *(c) equally likely events*

FIG. 7. False negative rates of NJ methods under various distance estimators as a function of the maximum pairwise inversion distance, for 10, 20, 40, 80, and 160 genomes. (Results for different numbers of genomes are pooled into a single figure when the model weights are identical.)

3.4. Robustness of distance estimators

As discussed earlier, estimating the true evolutionary distance requires assumptions about the model parameters. In the case of EDE, we assume that evolution proceeded through inversions only—so how well does NJ(EDE) perform when faced with a dataset produced through a combination of transpositions and inverted transpositions? In the case of the two IEBP methods, the computation requires values for the respective rates of inversion, transposition, and inverted transposition, respectively, which obviously leaves a lot of room for mistaken assumptions. We ran a series of experiments under conditions similar to those shown earlier, but where we deliberately mismatched the evolutionary parameters used in the production of the dataset and those used in the computation of the distance estimates used in NJ. Figure 8 shows the results for the Exact-IEPB estimator (results for



(a) inversions only *(b) transpositions only* *(c) equally likely events*

FIG. 8. Robustness of the Exact-IEBP method with respect to model parameters. Triples in the legend indicate the model values used in the Exact-IEBP method.

the other estimators are similar), indicating that our estimators, when used in conjunction with NJ, are remarkably robust in the face of erroneous model assumptions.

4. MAXIMUM PARSIMONY AND TOPOLOGICAL ACCURACY

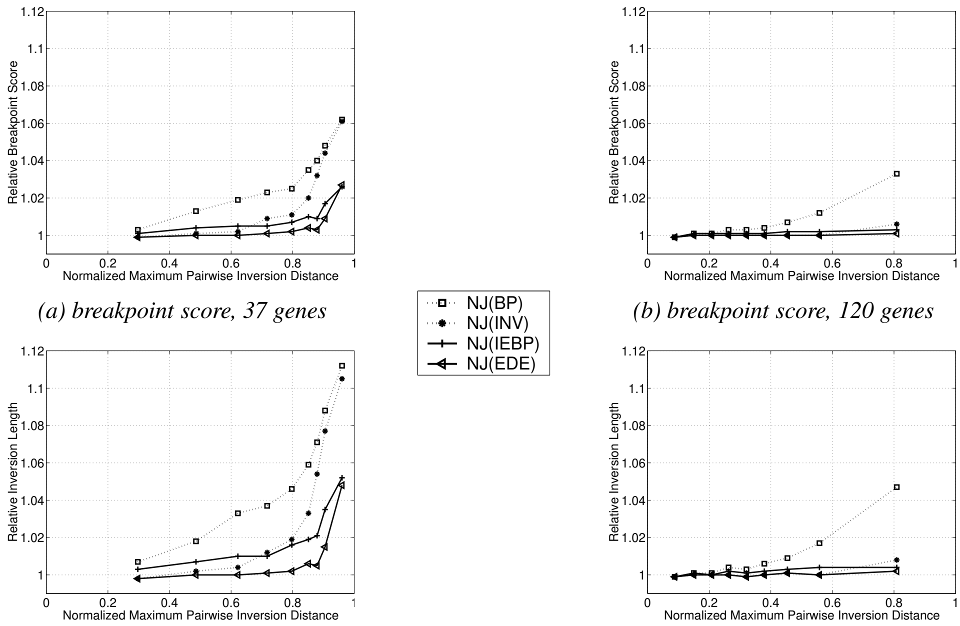
The main goal of phylogeny reconstruction is to produce the correct tree topology. Two basic approaches are currently used for phylogeny reconstruction from whole genomes: distance-based methods such as NJ applied to techniques for estimating distances and "maximum parsimony" (MP) approaches, which attempt to minimize the "length" of the tree, for a suitably defined measure of the length.

We examine two specific MP problems in this section: the breakpoint phylogeny problem, where we seek to minimize the total number of breakpoints over all tree edges, and the inversion phylogeny problem, where we seek to minimize the total number of inversions. We want to determine, using a simulation study, whether topological accuracy is improved by reducing the number of inversions or the number of breakpoints. If possible, we also want to determine whether the breakpoint phylogeny problem or the inversion phylogeny problem are topologically more accurate under certain evolutionary conditions, and if so, under which conditions.

We ran a large series of tests on model trees to investigate the hypothesis that minimizing the total breakpoint distance or inversion length of trees would yield more topologically accurate trees. We ran NJ on a total of 209 datasets with both inversion and breakpoint distances. Each test consists of at least 12 data points, on sets of up to 40 genomes. We used two genome sizes (37 and 120 genes, representative of mitochondrial and chloroplast genomes, respectively) and various ratios of inversions to transpositions and inverted transpositions, as well as various rates of evolution. For each dataset, we computed the total inversion and breakpoint distances and compared their values with the percentage of errors (measured as false negatives).

We used the nonparametric *Cox-Stuart test* [11] for detecting trends—i.e., for testing whether reducing breakpoint or inversion distance consistently reduces topological errors. Using a 95% confidence level, we found that over 97% of the datasets with inversion distance and over 96% of those with breakpoint distance exhibited such a trend. Indeed, even at the 99.9% confidence level, over 82% of the datasets still exhibited such a trend.

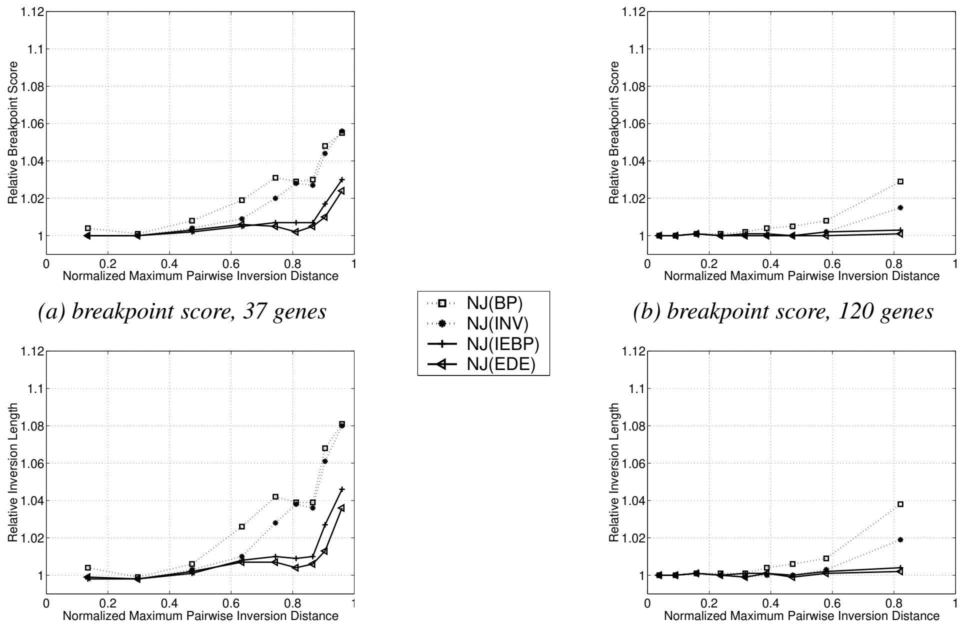
Figures 9 and 10 show the results of scoring the different NJ trees under the two optimization criteria: breakpoint score and inversion length of the tree. In general, the relative ordering and trend of the curves agree with the curves of Figure 7, suggesting that de-



(a) breakpoint score, 37 genes *(b) breakpoint score, 120 genes*

(c) inversion length, 37 genes *(d) inversion length, 120 genes*

FIG. 9. Scoring NJ methods under various distance estimators as a function of the maximum pairwise inversion distance for 10, 20, and 40 genomes. Plotted is the ratio of the NJ tree score to the model tree score (breakpoint or inversion) on an inversion-only model tree.



(a) breakpoint score, 37 genes *(b) breakpoint score, 120 genes*

(c) inversion length, 37 genes *(d) inversion length, 120 genes*

FIG. 10. Scoring NJ methods under various distance estimators as a function of the maximum pairwise inversion distance for 10, 20, and 40 genomes. Plotted is the ratio of the NJ tree score to the model tree score (breakpoint or inversion) on a model tree where the three classes of events are equiprobable.

creasing the number of inversions or breakpoints leads to an improvement in topological accuracy. The correlation is strongest for the 120-gene case; this may be because, for the same number of events but a larger number of genes, the rate of evolution effectively goes down and overlap of events becomes less likely. Finally, this trend still holds under the other evolutionary models (such as when only transpositions occur).

5. SEARCHING FOR MAXIMUM PARSIMONY TREES

5.1. The lower bound and its use

The following theorem is well known:

THEOREM 5.1. *Let d be a $n \times n$ matrix of pairwise distances between the taxa in a set S; let T be a tree leaf-labelled by the taxa in S; and let w be an edge-weighting on T, so that we have $w_{ij} = \sum_{e \in P_{ij}} w(e) \geq d_{ij}$. Set $w(T) = \sum_{e \in E(T)} w(e)$. If $1, 2, \ldots, n$ is a circular ordering of the leaves of T, under some planar embedding of T, then we have $2w(T) \geq d_{1,2} + d_{2,3} + \ldots + d_{n,1}$.*

This corollary immediately follows:

COROLLARY 5.1. *Let d be the matrix of minimum distances between every pair of genomes in a set S, let T be a fixed tree on S, and let $1, 2, \ldots, n$ be a circular ordering of leaves in T under some planar embedding of T. Then the length of T is at least $\frac{1}{2}(d_{1,2} + d_{2,3} + \ldots + d_{n,1})$.*

This corollary forms the basis of the old "twice around the tree" heuristic for the TSP based on minimum spanning trees [19]. Note that the theorem and its corollary hold for any distance measure that obeys the triangle inequality.

In earlier work [22, 24], we used these bounds in a simple manner to reduce the cost of searching tree space.

- We obtain an initial upper bound on the minimum achievable inversion length by using NJ with inversion distances. This upper bound is updated every time the search finds a better tree.
- Each tree we examine is presented in the standard nested-parentheses format (the nexus format [21]); this format defines a particular circular ordering of the leaves. We use that ordering to compute the lower bound of Corollary 5.1, again using inversion distances. If the lower bound exceeds the upper bound, the tree can be discarded.

This bounding may reduce the running time substantially, because the bound can be computed very efficiently, whereas scoring a tree with a tool like `GRAPPA` [17] involves solving numerous TSP instances. However, when the rate of evolution is high for the size of the genome, the bound often proves loose—the bound is exact when distances are additive, but high rates of evolution produce distances that are much smaller than the additive value.

We thus set about improving the bound as well as how it is used in the context of `GRAPPA` to prune trees before scoring them. Our new results come from three separate ideas: (i) tighten the lower bound; (ii) return more accurate scores for trees with large edge lengths; and (iii) process the trees so as to take better advantage of the bounds.

5.2. Tightening the bound

The particular circular order defined by the tree description is only one of a very large number of circular orderings compatible with that tree: since the tree has no internal ordering (i.e., no notion of left *vs.* right child), swapping two subtrees does not alter the phylogeny, but does yield a different circular ordering. Any of these orderings defines a valid lower bound, so that we could search through all orderings and retain the largest bound produced in order to tighten the bound. Unfortunately, the number of compatible circular orderings is exponential in the number of leaves, so that a full search is too expensive. We developed and tested a fast greedy heuristic, *swap-as-you-go*, that provides a high-quality approximation of the optimal bound. Our heuristic starts with the given tree and its implied circular ordering. It then traverses the tree in preorder from an arbitrarily chosen initial leaf, deciding locally at each node whether or not to swap the children by computing the score of the resulting circular ordering and moving towards larger values, in standard greedy fashion. With incremental computations, such a search takes linear time, because each swap only alters a couple of adjacencies, so that the differential cost of a swap can be computed in constant time. In our experiments, the resulting bound is always very close, or even equal, to the optimal bound and much better than the original value in almost all cases.

We tested a related bounding technique proposed by Bryant [8], but found it to be very slow (in order to yield reasonably tight bounds, it requires the introduction of Lagrangian variables and the solution of a system of linear equations to determine their values) and always (in our experiments) dominated by our algorithm. Since we use bounding strictly to reduce the total amount of work, it is essential that any lower bound be computable with very little effort.

5.3. Accurate scoring of long edges

Long edges in the tree suffer from the same problem as large leaf-to-leaf distances: they are seriously underestimated by an edit distance computation. Thus we decided to use the same remedy presented in the first part of this paper, by correcting edit distances with one of our true distance estimators. We chose the EDE estimator, because it offers the best tradeoff between accuracy and computational cost of our three estimators and used it within `GRAPPA` wherever distances are computed: in computing the distance matrix for NJ, in computing circular lower bounds from that matrix, and, more importantly, in computing the distance along each edge and in computing the median-of-three that lies at the heart of the `GRAPPA` approach [24]. Because a distance estimator effectively stretches the range of possible values and because that stretch is most pronounced when the edit distances are already large (the worst case for our circular bounds), using a distance estimator yields significant benefits—a number of our more challenging instances suddenly became very tractable with the combination of EDE distances and our improved circular bound.

5.4. Layered search

We added a third significant improvement to the bounding scheme. Since the bound itself can no longer be significantly improved, obtaining better pruning requires better use of the bounds we have. Our original approach to pruning [24] was simply to enumerate all trees, keeping the score of the best tree to date as an upper bound and computing a lower bound for each new tree to decide whether to prune it or score it. In this approach, each tree is generated once, bounded once, and scored at most once, but the upper bound in use through the computation may be quite poor until close to the end if optimal and near-optimal trees appear only toward the end of the enumeration—in which case the program must score nearly every tree. To remedy this problem, we devised a novel and radically different approach, which is motivated by the fact that generating and bounding a tree is very inexpensive, whereas scoring one, which involves solving potentially large numbers of instances of the Travelling Salesperson Problem, is very expensive.

Our new approach, which we call a *layered search*, still bounds each tree once and still scores it at most once, but it typically examines the tree more than once; it works as follows.

- In a first phase, we compute the NJ tree (using EDE distances) and score it to obtain an initial upper bound, as in the original code.
- In a second phase, every tree in turn is generated, its lower bound computed as described above, and that bound compared with the cost of the NJ tree. Trees not pruned away are stored, along with their computed lower bound, in buckets ordered by the value of the lower bound.[3]
- We then begin the layered search itself, which proceeds on the principle that the lower bound of a tree is correlated with the actual parsimony score of that tree. The search looks at each successive bucket of trees in turn, scoring trees that cannot be pruned through their lower bound and updating the upper bound whenever a better score is found.

This search technique is applicable to any class of optimization problems where the cost of evaluating an object in the solution space is much larger than the cost of generating that object and works well whenever there is a good correlation between the lower bound and the value of the optimal solution.

Our experiments indicate that, unless the interleaf distances are all nearly maximal (for the given number of genes), the correlation between our lower bound and the parsimony score is quite strong, so that our layered search strategy very quickly reduces the upper bound to a score that is optimal or nearly so, thereby enabling drastically better pruning—as detailed below, we frequently observed pruning rates of over 99.999%.

5.5. An experimental assessment of bounding

We measured the percentage of trees that are pruned through bounding (and thus not scored) as a function of the three model parameters: number of genomes, number of genes, and number of inversions per edge. We used an inversion-only scenario as well as one with approximately half inversions and half transpositions or inverted transpositions. Our data consisted of two collections of 10 datasets each for a combination of parameters. The number of genomes was 10, 20, 40, 80, and 160, the number of genes was 10, 20, 40, 80, and 160, and the rate of evolution varied from 2 to 8 events per tree edge, for a total of 75 parameter combinations and 1,500 datasets. We used EDE distances to score edges and our "swap-as-you-go" bounding computation, but not layered search—because layered search requires enumerating all trees up front, something that is simply not possible for 20 or more genomes.

For each data set with 10 genomes, `GRAPPA` tested all trees, scored and updated the upper bound if necessary, and kept statistics on the pruning rate. For 20 or more genomes, the number of trees is well beyond the realm of enumeration—with 20 genomes, we have $35!! \approx 2 \cdot 10^{20}$ trees! For these cases, we began by running `GRAPPA` for 6 hours to score as many trees as possible, then used the best score obtained in this first phase as an upper bound in a second phase where we ran another 6 hours during which a random selection

[3] The required storage can exceed the memory capacity of the machine, in which case we store buckets on disk in suitably-sized blocks. The cost of secondary memory access is easily amortized over the computation.

	$r = 2$					$r = 4$				$r = 8$				r value
	10	20	40	80	160	10	20	40	80	10	20	40	80	# genomes
10	95	100	5	1	1	71	5	0	1	70	0	0	1	
20	100	100	96	1	1	94	90	0	0	90	0	0	0	
40	100	100	100	100	100	100	100	0	0	100	0	0	0	
80	100	100	100	100	100	100	100	100	0	100	0	0	0	
160	100	100	100	100	100	98	100	100	100	-	-	-	-	
# genes														

FIG. 11. Percentage of trees eliminated through bounding for various numbers of genes and genomes and three rates of evolution.

of trees are bounded using our method and their bounds compared to the upper bound obtained in the first phase. The result is a (potentially very) pessimistic estimate of the pruning rate.

Figure 11 shows the percentage of trees pruned away by the circular lower bound; in the table, the parameter r denotes the expected number of inversions per edge used in the simulated evolution. (We could not run enough tests for the setting of 160 genes and $r = 8$, because merely scoring such a tree accurately can easily take days of computation—the instances of the TSP generated under these circumstances are very time-consuming.) In comparison with the similar table for our earlier approach [23], our new approach shows dramatic improvement, especially at high rates of evolution. We found that most circular orderings in datasets of up to 20 genomes were eliminated. However the table also shows that the bounding does not eliminate many trees in datasets with 40 or more genomes—not unless these datasets have large numbers of genes. The reason is clear: the number of genes dictates the range of values for the pairwise distances and thus also for the tree score—in terms of inversion distances, for instance, we have roughly mn possible tree scores, where m is the number of genes and n the number of genomes; yet the number of distinct trees is $(2n-5)!!$, a number so large that, even when only a very small fraction of the trees are near optimal, that fraction contains so many trees that they cannot efficiently be distinguished from others with only mn buckets. Of course, GRAPPA in 6 hours can only examine a vanishingly small fraction of the tree space, so that the upper bound it uses is almost certainly much too high. In contrast with these findings, our layered method gave us pruning rates of 90% in the 10-genome case for $r = 8$ and 10 or 20 genes, a huge improvement over the complete failure of pruning by the normal search method.

6. A TEST OF OUR METHODS ON REAL DATA

We repeated our analysis of the *Campanulaceae* dataset, which consists of 13 chloroplast genomes, one of which is the outgroup Tobacco, but this time to reconstruct the EDE (as opposed to the breakpoint or inversion) phylogeny. Each of the 13 genomes has 105 gene segments and, though highly rearranged, has what we consider to be a low rate of evolution. In our previous analysis, we found that GRAPPA, with our first bounding approach, pruned about 85% of the trees, for a substantial speedup (on the order of 5–10) over a version without pruning. By using EDE distances and our improved bound computation, we increased this percentage to over 95%, for another substantial speedup of 5–10. By adding layered search, however, we managed to prune almost all of the 13.75 billion trees—all but a few hundred thousand, which were quickly scored and dispatched, for a further speedup of close to 20. The pruning rate was over 99.99%, reducing the number of trees that had to be scored by a factor of nearly 800. As a result, the same dataset that required a couple of hours on a 512-processor supercluster when using our first bounding strategy [2, 22, 24] can now be run in a few hours on a single workstation—and in one minute on that same cluster. In terms of our original comparison to the `BPAnalysis` code, we have now achieved a speedup (on the supercluster) of *one hundred million*! We also confirmed the results of our previous analysis—that is, the trees returned in our new analysis, which uses EDE distances, match those returned in an analysis that had used minimum inversion distances, a pleasantly robust result.

The speedup obtained by bounding depends on two factors: the percentage of trees that can be eliminated by the bounding and the difficulty of the TSP instances avoided by using the bounds. As Table 11 shows, when the rate of evolution is not too high, close to 100% of the trees can be eliminated by using the bounds. However, the TSP instances solved in `GRAPPA` can be quite small when the evolutionary rate is low, due to how we compress data (see [24]). Consequently, the speedup also depends on the rate of evolution, with lower rates of evolution producing easier TSP instances and thus smaller speedups. The *Campanulaceae* dataset is a good example of a dataset that is quite easy for `GRAPPA`, in the sense that it produces easy TSP instances—but even in this case, a significant speedup results. More generally, the speedup increases with larger numbers of genomes and, to a point, with higher rates of evolution. When one is forced to exhaustively search tree space, these speedups represent substantial savings in time.

7. CONCLUSIONS AND FUTURE WORK

We have described new theoretical and experimental results that have enabled us to analyze significant datasets in terms of inversion events and that also extend to models incorporating transpositions. This work is part of an ongoing project to develop fast and robust techniques for reconstructing phylogenies from gene-order data. The distance estimators we have developed clearly outperform straight distance measures in terms of both accuracy (when used in conjunction with a distance-based method such as neighbor-joining) and efficiency (when used with a search-based optimization method such as implemented in our `GRAPPA` software suite). Our current software suffers from several limitations, particularly its exhaustive search of all of (constrained) tree space. However, the bounds we have described can be used in conjunction with branch-and-bound (based on inserting leaves into subtrees or extending circular orderings) as well as in heuristic search techniques.

ACKNOWLEDGMENTS

We thank R. Jansen for introducing us to this research area, and D. Sankoff and J. Nadeau for inviting us to the DCAF meeting, during which some of the ideas in this paper came to fruition. This work is supported in part by National Science Foundation grants ACI 00-81404 (Moret), DEB 01-20709 (Moret and Warnow), EIA 01-13095 (Moret), EIA 01-13654 (Warnow), EIA 01-21377 (Moret), and EIA 01-21680 (Warnow), and by the David and Lucile Packard Foundation (Warnow).

REFERENCES

1. K. Atteson. The performance of the neighbor-joining methods of phylogenetic reconstruction. *Algorithmica*, 25(2/3):251–278, 1999.

2. D.A. Bader and B.M.E. Moret. GRAPPA runs in record time. *HPC Wire*, 9(47), 2000.

3. D.A. Bader, B.M.E. Moret, and M. Yan. A linear-time algorithm for computing inversion distance between signed permutations with an experimental study. *J. Comput. Biol.*, 8(5):483–491, 2001.
4. V. Bafna and P. Pevzner. Sorting permutations by transpositions. In *Proc. 6th Annual ACM-SIAM Symp. on Disc. Alg. SODA95*, pages 614–623. ACM Press, 1995.
5. 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, Tokyo, 1997.
6. M. Blanchette, T. Kunisawa, and D. Sankoff. Gene order breakpoint evidence in animal mitochondrial phylogeny. *J. Mol. Evol.*, 49:193–203, 1998.
7. D. Bryant. The complexity of the breakpoint median problem. Centre de Recherches Mathématiques, Technical Report CRM2579, Université de Montréal, 1998.
8. D. Bryant. A lower bound for the breakpoint phylogeny problem. In *Lecture Notes in Computer Science, Vol.1848: Proc. 11th Annual Symp. Combinatorial Pattern Matching (CPM)*, pages 235–247. Springer-Verlag, 2000.
9. A. Caprara. Formulations and hardness of multiple sorting by reversals. In *Proc. 3rd Int'l Conf. on Comput. Mol. Biol. RECOMB99*, pages 84–93. ACM Press, 1999.
10. A. Caprara and G. Lancia. Experimental and statistical analysis of sorting by reversals. In D. Sankoff and J.H. Nadeau, editors, *Comparative Genomics*, pages 171–184. Kluwer Acad. Pubs., 2000.
11. W.J. Conover. *Practical Nonparametric Statistics, 3rd ed.* John Wiley & Sons, 1999.
12. 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 Acad. Pubs., 2000.
13. M.E. Cosner, R.K. Jansen, B.M.E. Moret, L.A. Raubeson, L. Wang, T. Warnow, and S.K. Wyman. A new fast heuristic for computing the breakpoint phylogeny and experimental phylogenetic analyses of real and synthetic data. In *Proc. 8th Int'l Conf. on Intelligent Systems for Mol. Biol. ISMB-2000*, pages 104–115, 2000.
14. Workshop on Gene Order Dynamics, Comparative Maps and Multigene Families (DCAF). Montreal, Canada, August 2000.
15. DIMACS Workshop on Whole Genome Comparison. Piscataway, New Jersey, USA, February 2001.
16. 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.
17. GRAPPA: Genome Rearrangements Analysis under Parsimony and other Phylogenetic Algorithms. `http://www.cs.unm.edu/~moret/GRAPPA/`.
18. S. Hannenhalli and P. Pevzner. Transforming cabbage into turnip (polynomial algorithm for genomic distance problems). In *Proc. 27th Symp. on Theory of Comp. STOC95*, pages 178–189. ACM Press, 1995.
19. M. Held and R.M. Karp. The travelling salesman problem and minimum spanning trees. *Oper. Res.*, 18:1138–1162, 1970.
20. R.K. Jansen. Personal communication, October 2000.
21. D.R. Maddison, D.L. Swofford, and W.P. Maddison. Nexus: An extensible file format for systematic information. *Sys. Biol.*, 46:590–621, 1997.
22. B.M.E. Moret, D.A. Bader, and T. Warnow. High-performance algorithm engineering for computational phylogenetics. In *Proc. 2001 Int'l Conf. Computational Science*, volume 2073–2074 of *Lecture Notes in Computer Science*, San Francisco, CA, 2001. Springer Verlag.
23. B.M.E. Moret, L.-S. Wang, T. Warnow, and S. Wyman. New approaches for reconstructing phylogenies based on gene order. In *Proc. 9th Intl. Conf. on Intel. Sys. for Mol. Bio. ISMB 2001*, volume 17 of *Bioinformatics*, pages S165–S173, 2001.
24. 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. Biocomputing PSB 2001*, pages 583–594. World Scientific Pub., 2001.
25. J.H. Nadeau and B.A. Taylor. Lengths of chromosome segments conserved since divergence of man and mouse. *Proc. Nat'l Acad. Sci. USA*, 81:814–818, 1984.
26. R.G. Olmstead and J.D. Palmer. Chloroplast DNA systematics: a review of methods and data analysis. *Amer. J. Bot.*, 81:1205–1224, 1994.
27. J.D. Palmer. Chloroplast and mitochondrial genome evolution in land plants. In R. Herrmann, editor, *Cell Organelles*, pages 99–133. Springer Verlag, 1992.

28. I. Pe'er and R. Shamir. The median problems for breakpoints are NP-complete. *Elec. Colloq. on Comput. Complexity*, 71, 1998.

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

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

31. D. Sankoff and M. Blanchette. Multiple genome rearrangement and breakpoint phylogeny. *J. Comp. Biol.*, 5:555–570, 1998.

32. D. Sankoff and M. Blanchette. Probability models for genome rearrangements and linear invariants for phylogenetic inference. In *Proc. 3rd Int'l Conf. on Comput. Mol. Bio. RECOMB99*, pages 302–309. ACM Press, 1999.

33. D. Swofford, G. Olson, P. Waddell, and D. Hillis. Phylogenetic inference. In D. Hillis, C. Moritz, and B. Mable, editors, *Molecular Systematics, 2nd ed.*, chapter 11. Sinauer Associates, 1996.

34. L.-S. Wang. Exact-IEBP: a new technique for estimating evolutionary distances between whole genomes. In *Proc. 1st Workshop Algs. Bioinformatics WABI'01*, volume 2149 of *Lecture Notes in Computer Science*, pages 176–190, Århus, Denmark, 2001. Springer Verlag.

35. L.-S. Wang and T. Warnow. Estimating true evolutionary distances between genomes. In *Proc. 33th Annual Symp. Theory Comp. STOC'01*, pages 637–646. ACM, 2001.