Linear Programming for Phylogenetic Reconstruction Based on Gene Rearrangements *

Jijun Tang[1] and Bernard M.E. Moret[2]

[1] Dept. of Computer Science & Engineering, U. of South Carolina, Columbia, SC 29208
[2] Dept. of Computer Science, U. of New Mexico, Albuquerque, NM 87131

Abstract. Phylogenetic reconstruction from gene rearrangements has attracted increasing attention from biologists and computer scientists over the last few years. Methods used in reconstruction include distance-based methods, parsimony methods using sequence-based encodings, and direct optimization. The latter, pioneered by Sankoff and extended by us with the software suite GRAPPA, is the most accurate approach, but has been limited to small genomes because the running time of its scoring algorithm grows exponentially with the number of genes in the genome. We report here on a new method to compute a tight lower bound on the score of a given tree, using a set of linear constraints generated through selective applications of the triangle inequality (in the spirit of GESTALT). Our method generates an integer linear program with a carefully limited number of constraints, rapidly solves its relaxed version, and uses the result to provide a tight lower bound. Since this bound is very close to the optimal tree score, it can be used directly as a selection criterion, thereby enabling us to bypass entirely the expensive scoring procedure. We have implemented this method within our GRAPPA software and run several series of experiments on both biological and simulated datasets to assess its accuracy. Our results show that using the bound as a selection criterion yields excellent trees, with error rates below 5% up to very large evolutionary distances, consistently beating the baseline Neighbor-Joining. Our new method enables us to extend the range of applicability of the direct optimization method to chromosomes of size comparable to those of bacteria, as well as to datasets with complex combinations of evolutionary events.

1 Introduction

Biologists can infer the ordering and strandedness of genes on a chromosome and thus represent each chromosome by an ordering of signed genes (where the sign indicates the strand). These gene orders can be rearranged by evolutionary events such as inversions and transpositions and, because they evolve slowly, give biologists an important new source of data for phylogeny reconstruction [8, 16, 18]. Appropriate tools for analyzing such data may help resolve some difficult phylogenetic reconstruction problems. Developing such tools is thus an important area of research: the recent DCAF symposium [22] was devoted to this topic, while results are rapidly accumulating [14].

* Contact author: Jijun Tang, jtang@cse.sc.edu

A natural optimization problem for phylogeny reconstruction from gene-order data is to reconstruct a tree that minimizes the total number of evolutionary events. This problem is NP-hard for most criteria—even the very simple problem of computing the median of just *three* genomes (the median of k genomes is a genome that minimizes the sum of the pairwise distances between itself and each of the k given genomes) under such models was proved NP-hard [4, 17].

For some datasets (e.g., chloroplast genomes of land plants), biologists conjecture that rearrangement events are predominantly *inversions* (also called reversals) [6]. In other datasets (e.g., mitochondrial genomes), transpositions are viewed as more likely, but their relative preponderance with respect to inversions is unknown. Sankoff proposed the *breakpoint* distance (the number of pairwise gene adjacencies present in one genome but absent in the other) as a measure of distance between genomes that is independent of any particular mechanism of rearrangement. The *breakpoint phylogeny* [1] is then the most parsimonious tree with respect to breakpoint distances. By analogy, the *inversion phylogeny* is the most parsimonious tree with respect to inversion distances.

The main software package for reconstructing the inversion (or breakpoint) phylogeny is our `GRAPPA` [15]. Its basic optimization tool is an algorithm for computing the inversion (or breakpoint) median of three genomes. Extensive testing has shown that the trees returned by `GRAPPA` are superior to those returned by other methods used in phylogenetic reconstruction based on gene orders, such as distance-based methods and parsimony based on encodings [13, 14, 29]. The closely related software of Pevzner's group, MGR [3], is the only method that approaches its accuracy. Moreover, our extension using disk-covering, `DCM-GRAPPA` [27], runs quickly on large datasets—indeed, the number of taxa in the dataset is no longer the main issue.

Two serious issues remain, however. Handling genomes with unequal gene content (i.e., involving duplications, insertions, and deletions of genes) remains largely unsolved, although some progress has been made in computing pairwise distances in such cases [9, 10, 25, 28]. Handling large genomes within `GRAPPA` is very expensive, because the median computation takes time exponential in the size of the genomes; this problem prevents its application to organismal genomes with thousands of genes. It is this second problem that we tackle here.

2 Definitions

A *phylogeny* for a set S of N genomes is a (preferably) binary tree with N leaves, with each leaf labeled by a distinct element of S. Let G be the genome with signed ordering of $g_1, g_2, \ldots, g_n$. An *inversion* between indices i and j ($i \leq j$), produces the genome with linear ordering

$$g_1, g_2, \cdots, g_{i-1}, -g_j, -g_{j-1}, \cdots, -g_i, g_{j+1}, \cdots, g_n$$

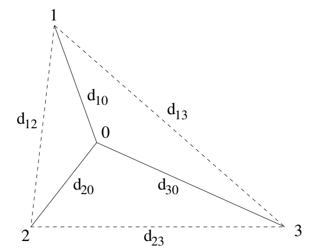


Fig. 1. The median problem: given gene orders 1, 2, and 3, find a gene-order 0 that minimizes $\sum_{i=1}^{3} d_{0,i}$.

A *transposition* on genome G acts on three indices i, j, k, with $i \leq j$ and $k \notin [i, j]$, picking up the interval $g_i, g_{i+1}, \ldots, g_j$ and inserting it immediately after g_k. Thus genome G is replaced by (assume $k > j$):

$$g_1, \cdots, g_{i-1}, g_{j+1}, \cdots, g_k, g_i, g_{i+1}, \cdots, g_j, g_{k+1}, \cdots, g_n$$

The *edit distance* between two genomes is the minimum number of evolutionary events required to transform one genome into the other. When only inversions are allowed, the edit distance is the *inversion distance*. The *score* of a tree is the sum of the costs of its edges, where the cost of an edge is the distance (breakpoint or edit distance) between the two genomes that label the endpoints of the edge.

For N genomes $\{G_1, G_2, \ldots, G_N\}$, each with identical set of genes, the *median problem* is to find a signed gene order G_0 that minimizes $\sum_{i=1}^{N} d_{0,i}$, where $d_{0,i}$ is the distance between G_0 and G_i. Because our various genomic distances are all metrics, we must have

$$\sum_{i=1}^{N} d_{0,i} \geq \frac{1}{2}\left(d_{N,1} + \sum_{i=1}^{N-1} d_{i,i+1}\right)$$

When this inequality is an equality, we call the corresponding median a *perfect median*. Finding the gene order G_0 is NP-hard even for the case $N = 3$, the case (illustrated in Fig. 1) of most use in phylogenetic reconstruction [4, 17].

In the following, we will consider only genomes with *equal gene content*, that is, with the same number of genes and where each gene appears exactly once. Clearly, this restriction is unrealistic in biological practice; however, we use it here mostly for clarity of exposition: all results presented in this paper can be readily applied to genomes with unequal gene contents as long as a pairwise distance between such genomes can be computed, as in [9, 25].

3 GRAPPA

GRAPPA is based on the approach pioneered by Sankoff and Blanchette in the software package BPAnalysis [21], but uses various algorithmic techniques to improve its accuracy and speed. GRAPPA is an exhaustive search method, moving systematically through the space of all $(2N-5)!!$ possible trees on N genomes. For each tree, the program tests a lower bound to determine whether the tree is worth scoring; if so, then the program will iteratively solve the median problems at internal nodes until convergence, as outlined in Fig. 2. Since the scoring

```
Initially label all internal nodes with gene orders
Repeat
    For each internal node v, with neighbors A, B and C, do
    Solve median problem on A, B, C to yield m
    If relabeling v with m improves the tree score, then do it
Until no change occurs
```

Fig. 2. The GRAPPA scoring procedure

procedure of GRAPPA involves solving numerous instances of median problems, a fast median solver is crucial. Two inversion median solvers are available, both using a branch-and-bound strategy. Caprara's solver [5] is based on an extension of the breakpoint graph; that developed by Siepel and Moret [24] runs a direct search. Although they are both fast when the pairwise distances among the three given genomes are relatively small, they can become extremely slow when the distances become larger. For example, in the worst case, Siepel's algorithm needs to check n^d gene orders, where d is $min(d_{12}+d_{13}, d_{21}+d_{23}, d_{13}+d_{23})$ and n is the number of genes (see Fig. 1). Indeed, for large genomes, a single median problem can take anywhere from seconds to days of computation. It is thus crucial to avoid scoring trees unnecessarily and thus to use tight lower bounds.

4 Lower Bounding with Perfect Medians

Siepel and Moret [24] reported that almost all medians found by their algorithms were perfect medians, i.e., they obeyed

$$\sum_{i=1}^{3} d_{0,i} = \frac{d_{1,2}+d_{1,3}+d_{2,3}}{2}$$

When the pairwise distances were about $0.1n$, all medians were perfect; as the distances increased, the proportion of perfect medians decreased slowly—for instance, over 97% of the medians remained perfect with pairwise distances around $0.3n$ [23]. Moreover, even when the medians were not perfect, their

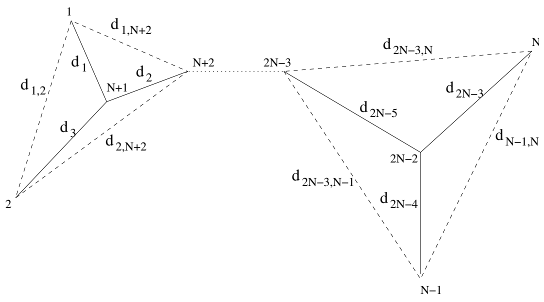


Fig. 3. Perfect medians for internal nodes

scores exceeded the lower bound by only one inversion. These findings indicate that an assumption that all medians are perfect may lead to a tight lower bound.

Label a phylogenetic tree with N leaves as follows; the leaves are labeled from the set $\{1,2,\ldots,N\}$, while the internal nodes (of which there are at most $N-2$) are labeled from the set $\{N+1,N+2,\ldots,2N-2\}$; we number the edges from 1 to $2N-3$ and denote the length of edge i by d_i, as illustrated in Fig. 3.

Theorem 1. *The sum of the perfect median scores (over all internal tree nodes) is a lower bound on the tree score; that is, we have (see Fig. 3):*

$$w(T)=\sum_{i=1}^{2N-3} d_i \geq \frac{(d_{1,2}+d_{1,N+2}+d_{2,N+2})+\ldots+(d_{N-1,N}+d_{2N-3,N-1}+d_{2N-3,N})}{2}$$

Because most medians are perfect, this lower bound is very close—or perhaps even equal—to the tree score. Of course, we cannot compute this lower bound exactly without first solving the median problems, which would defeat the entire purpose of bounding. So we settle for a close approximation of that bound through mathematical programming.

5 Setting up the Integer Linear Program

An integer linear program is composed of a linear objective function and of a collection of constraints, most taking the form of linear equations or inequations and some requiring that certain variables assume only integer values. The scoring function for our method is just the length of the tree; that is, our objective is to minimize the sum of the tree edge lengths:

$$\min \sum_{i=1}^{2N-3} d_i \tag{1}$$

This function introduces a total of $2N-3$ variables.

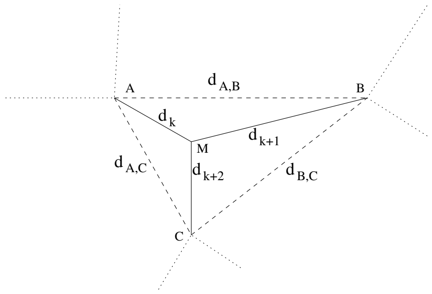


Fig. 4. Equation setup for any internal nodes

Given any internal node M, its three neighbors A, B. and C, and their associated tree edges k, $k+1$, and $k+2$ (see Fig. 4), we can write (if M is a perfect median):

$$d_k + d_{k+1} + d_{k+2} = \frac{d_{A,B} + d_{A,C} + d_{B,C}}{2} \tag{2}$$

Each of the $N-2$ internal nodes gives rise to one such equation, which we use as an equality constraint in the integer linear program. (This approach was previously used in the sequence alignment program GESTALT [12] to help in deriving good sequence assignments for internal nodes.) Equation 2 introduces three new variables per node (the pairwise distances $d_{A,B}$, $d_{A,C}$, and $d_{B,C}$), unless two of the node's neighbors are leaves, in which case only two new variables are introduced (because one of the pairwise distances is then known). Thus, overall, Equations 1 and 2 (the latter applied at each internal node) introduce from $4N$ to $5N$ variables, depending on the tree shape. Therefore we must add many constraints (beyond our $N-2$ equalities) in order to obtain a solution.

A good selection of constraints derived from triangle inequalities in the tree should meet the following criteria:

- Since the only available data are in the distance matrix, we should use every pairwise distance in the set of constraints.
- Because LP solvers take more time when the number of constraints increases and because many constraints will be redundant, the number of constraints should be kept as small as possible.
- In order to obtain robust solutions, every variable added by Equations 1 and 2 should appear in at least one constraint.

Consider again Fig. 4: in addition to the edges of the tree (drawn with uninterrupted lines), we have shown the pairwise connections among the three neighbors of an internal node (drawn with dashed lines). By considering the graph resulting from the addition of these edges connecting neighbors of internal nodes to the the set of edges of the original tree, we get a graph with many alternate paths. We base our additional constraints on the distances between pairs of tree leaves, as measured directly and as measured along paths in the extended graph. Specifically, for each pair (i, j) of leaves, we can write the inequality

$$d_{i,j} \leq \sum_{e \in \text{path}} d(e)$$

where $d_{i,j}$ is the genomic distance between the two gene orders i and j and where $d(e)$ is the length of edge e on the selected path from leaf i to leaf j in the extended graph. Since there are $\binom{n}{2}$ pairs of leaves, we immediately get a large number of constraints.

However, a number of edges in the extended graph may remain unused in any constraint; to remedy this problem, we generate *two* inequalities for each pair of leaves that are not siblings: one each for the two paths between these leaves with the fewest edges. Finding the k shortest paths between two vertices in a graph is a thoroughly studied problem [11] and is easily done, especially when the length is just the number of edges. Using the two shortest paths will produce $N(N-2)$ inequality constraints, which works well in our context, for three reasons:

- Each pairwise distance shows up in at least one inequality constraints.
- For 20 genomes (the largest problem size we need to solve within the DCM approach), we get on the order of 100 variables and 400 constraints, an instance of quite modest size for a modern LP solver.
- There are roughly N^2 constraints and at most $5N$ variables, so that, on average, a variable will appear in $\frac{N}{5}$ constraints. Thus the probability that a variable does not appear in any constraint is very small.

6 Implementation Details

We implemented this bounding computation within our `GRAPPA` platform. We do not solve the integer linear program, but only its linear programming relaxation: the ILP itself is already an approximation and would take too long to solve exactly and many techniques (such as randomized rounding) are readily available to use LP solutions in order to obtain a good integer solution. We used a standard non-commercial package for linear programming, `lp_solve` (version 4.0).

The new bound can be used in two different contexts: (i) we can use it to improve the pruning rate of `GRAPPA`; or (ii) we can use it directly as a selection criterion. The current version of `GRAPPA` (2.0) already has very fast and strong pruning (often eliminating 99.9999% of the candidate trees); the LP bound, which is noticeably more expensive to compute than the bounds currently used in GRAPPA, can be used to filter the remaining trees in order further to reduce the number of trees that must be scored.

The LP bound offers (so far) no help in scoring a tree: the problem with medians of large genomes remains unaffected. Thus, while the first application of the LP bound does provide additional speedup for GRAPPA, it does not yet enable us to handle large genomes. The second application enables us to process trees quickly: we retain the pruning strategy of GRAPPA and simply compute a "score" (the LP lower bound) for each remaining tree, retaining those trees with the best score. With this approach, the size of the genomes no longer presents any computational problem, so that we can handle datasets of up to 16 arbitrarily large genomes with GRAPPA and much larger datasets with DCM-GRAPPA.

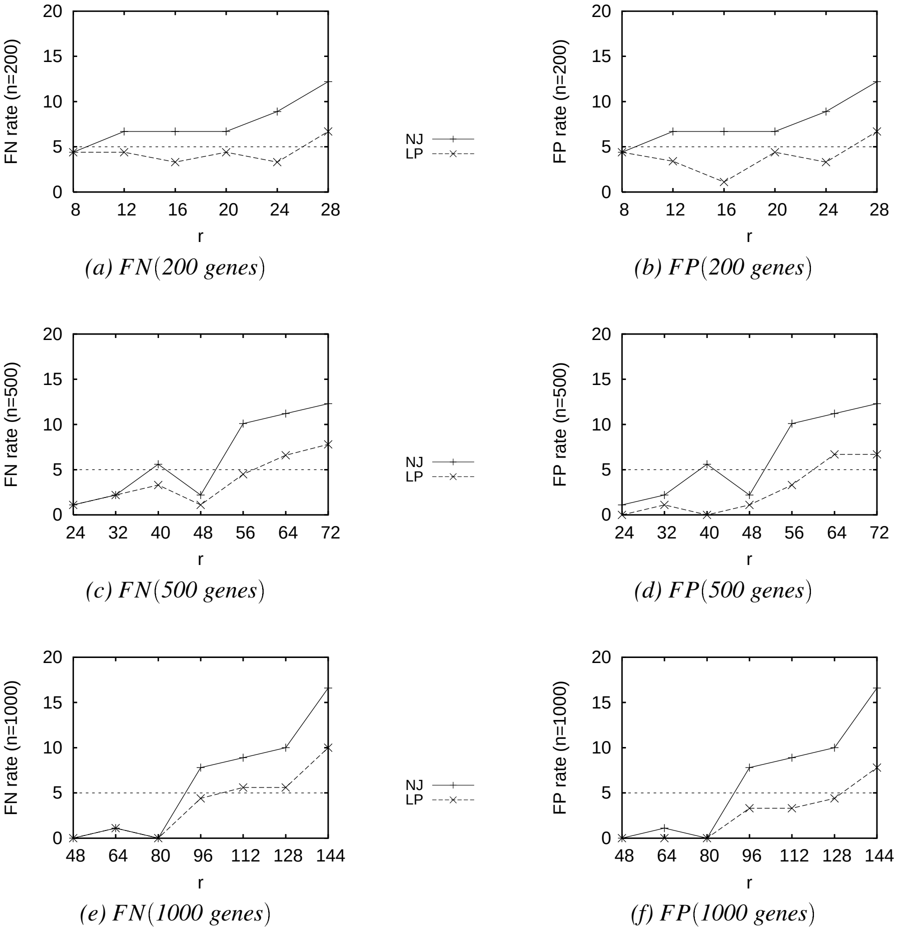
7 Experimental Results

We set out to test the accuracy of the second approach described above: using the LP bound as a direct selection criterion to choose among the trees not pruned away by GRAPPA. For this purpose, we generated datasets of 12 genomes each (datasets of that 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 [27]) and chose genomes of 200, 500, and 1,000 genes, spanning the range from large organelles (such as plant mitochondria) to small bacteria (such as symbiotic bacteria). Our model trees are birth-death trees, but the number of evolutionary events along an edge is set independently of the birth-death process to avoid any semblance of a molecular clock. We used a large range of evolutionary rates, including very high 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 5% to 15% of the number of genes. The actual number of events along each edge is sampled from a uniform distribution on the set $\{1, 2, \ldots, 2r\}$. While all our distance computations are based on inversion distances, we generated the data with a deliberate model mismatch to test the robustness of our bounding, using a mix of 80% inversions and 20% transpositions. For each combination of parameter settings, we ran 10 datasets and averaged the results.

Given an inferred tree (reconstructed phylogeny), we can assess the topological accuracy *false positive* and *false negative* [19] with respect to the true tree. If an edge in the true tree is missing in the inferred tree, this edge is then called a *false negative* (FN). Similarly, a *false positive* edge (FP) is an edge in the inferred tree but not in the true tree.

Our baseline is the basic neighbor-joining (NJ) method [20]. We considered all trees given the minimum score by our LP procedure and took their strict consensus. Therefore, the trees returned by our procedure need not be fully resolved and will tend to have somewhat better rates for false positives (FP) than for false negatives (FN). Thus we report FP and FN rates separately rather than as a single Robinson-Foulds score. We attempted to run `GRAPPA` on these datasets, but, for all but the lowest of the r values, some median computations took days or weeks. In consequence, we could not complete the runs, so that our comparisons are limited to NJ and our new LP-based method.

Fig. 5 shows our results for each genome size; we placed a line at the 5% error level, the typical threshold of acceptability for accuracy in phylogenetic reconstruction [26]. Note that, while the NJ trees almost always exceed the 5% error rate in both FP and FN, our LP-based method only exceeds that threshold for very large evolutionary rates; for instance, with 200 genes and $r = 28$, some



(a) $FN(200\ genes)$ *(b) $FP(200\ genes)$*

(c) $FN(500\ genes)$ *(d) $FP(500\ genes)$*

(e) $FN(1000\ genes)$ *(f) $FP(1000\ genes)$*

Fig. 5. Average FN and FP rates as a function of r for 200, 500, and 1,000 genes.

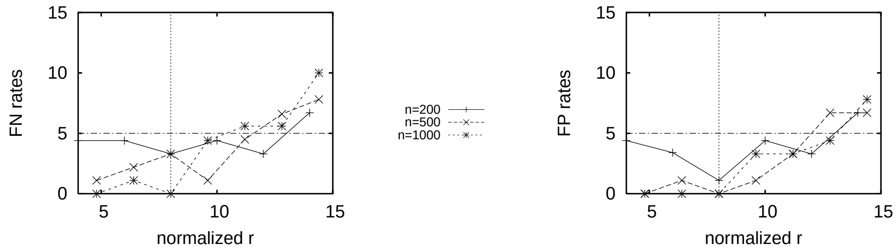


Fig. 6. Average FN (left) and FP (right) rates as a function of the normalized $\frac{r}{n}$ (the vertical line indicates when LP becomes significantly better than NJ).

edges will have up to 56 inversions and transpositions, resulting in well over 100 breakpoints. In Fig. 6, we show the FP and FN rates for all three genome sizes, as a function of the ratio $\frac{r}{n}$—the expected percentage of events per edge in terms of genome size. Again, note that, when the normalized r is 15%, the breakpoint distance could reach 70% of the genome size—values at which all past reconstruction methods based on distance fail as badly as NJ; in contrast, our new LP bound still selects good trees at that rate.

We also ran our procedure on small chloroplast datasets that we have used in the past (the Campanulaceae [7] and some land plants and algae [28]), with similar results.

8 Conclusions

We have used mathematical programming techniques to derive tight bounds on the total length (score) of a phylogenetic tree and have used these bounds as a selection criterion to reconstruct small phylogenies based on gene-order data for genomic sizes that had been out of reach of existing reconstruction tools. Our experiments show that the method works well, returning trees with accurate (better than 95%) topologies up to very large evolutionary rates. While we have not applied the method to the reconstruction of ancestral genomes (and the solution of the median problem for genomes), we expect that knowledge of the edge lengths (provided by the LP solution) will enable us to speed up such reconstruction dramatically, thereby removing the last scaling problem left in phylogenetic reconstruction from gene-order data.

Acknowledgments

This work is supported by the US National Science Foundation under grants ANI 02-03584, EF 03-31654, IIS 01-13095, IIS 01-21377, and DEB 01-20709, and by the Dept. of Computer Science and Engineering at U. of South Carolina.

References

1. 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.
2. M. Berkelaar, K. Eikland, and P. Notebaert. `lp_solve` Available at `www.geocities.com/lpsolve/`.
3. G. Bourque and P. Pevzner. Genome-scale evolution: reconstructing gene orders in the ancestral species. *Genome Research*, 12:26–36, 2002.
4. A. Caprara. Formulations and hardness of multiple sorting by reversals. In *Proc. 3rd Ann. Int'l Conf. Comput. Mol. Biol. (RECOMB'99)*, pages 84–93. ACM Press, New York, 1999.
5. 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.
6. Cosner, M.E., R.K. Jansen, J.D. Palmer and S.R. Downie SR (1997) The highly rearranged chloroplast genome of Trachelium caeruleum (Campanulaceae): multiple inversions, inverted repeat expansion and contraction, transposition, insertions/deletions, and several repeat families. *Curr Genet 1997*, 31:419-429.
7. 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'00)*, pages 104–115, 2000.
8. 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.
9. J. Earnest-DeYoung, E. Lerat, and B.M.E. Moret. Reversing gene erosion: reconstructing ancestral bacterial genomes from gene-content and gene-order data. In *Proc. 4th Int'l Workshop Algs. in Bioinformatics (WABI'04)*, volume 3240 of *Lecture Notes in Computer Science*, pages 1–13. Springer Verlag, 2004.
10. N. El-Mabrouk. Genome rearrangement by reversals and insertions/deletions of contiguous segments. In *Proc. 11th Ann. Symp. Combin. Pattern Matching (CPM'00)*, volume 1848 of *Lecture Notes in Computer Science*, pages 222–234. Springer Verlag, 2000.
11. D. Eppstein. Finding the k shortest paths. *SIAM J. on Computing*, 28(2):652–673, 1998.
12. G. Lancia and R. Ravi, GESTALT: GEnomic STeiner ALignmenTs. In *Proc. 10th Ann. Symp. Combin. Pattern Matching (CPM'99)*, volume 1645 of *Lecture Notes in Computer Science*, pages 101–114. Springer Verlag, 1999.
13. 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.
14. B.M.E. Moret, J. Tang, and T. Warnow. Reconstructing phylogenies from gene-content and gene-order data. In O. Gascuel, editor, *Mathematics of Evolution and Phylogeny*, pages 321–352. Oxford University Press, 2005.
15. 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.
16. J.D. Palmer. Chloroplast and mitochondrial genome evolution in land plants. In R. Herrmann, editor, *Cell Organelles*, pages 99–133. Springer Verlag, 1992.
17. I. Pe'er and R. Shamir. The median problems for breakpoints are NP-complete. *Elec. Colloq. on Comput. Complexity*, 71, 1998.
18. L.A. Raubeson and R.K. Jansen. Chloroplast DNA evidence on the ancient evolutionary split in vascular land plants. *Science*, 255:1697–1699, 1992.
19. D.R. Robinson and L.R. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53:131–147, 1981.

20. N. Saitou and M. Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, 4:406–425, 1987.
21. D. Sankoff and M. Blanchette. Multiple genome rearrangement and breakpoint phylogeny. *J. Comput. Biol.*, 5:555–570, 1998.
22. D. Sankoff and J. Nadeau, editors. *Comparative Genomics.* Kluwer Academic Pubs., Dordrecht, Netherlands, 2000.
23. A.C. Siepel. Exact algorithms for the reversal median problem. Master's thesis, U. New Mexico, Albuquerque, NM, 2001. Available at `www.cs.unm.edu/∼acs/thesis.html`.
24. A.C. Siepel and B.M.E. Moret. Finding an optimal inversion median: experimental results. In *Proc. 1st Int'l Workshop Algs. in Bioinformatics (WABI'01)*, volume 2149 of *Lecture Notes in Computer Science*, pages 189–203. Springer Verlag, 2001.
25. K.M. Swenson, M. Marron, J.V. Earnest-DeYoung, and B.M.E. Moret. Approximating the true evolutionary distance between two genomes. In *Proc. 7th Workshop on Alg. Engineering & Experiments (ALENEX'05)*, Vancouver (2005), SIAM Press.
26. D.L. Swofford, G. Olson, P. Waddell, and D.M. Hillis. Phylogenetic inference. In D.M. Hillis, C. Moritz, and B. Mable, editors, *Molecular Systematics, 2nd ed.*, chapter 11. Sinauer Associates, 1996.
27. 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.
28. 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.
29. 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.