A New Fast Heuristic for Computing the Breakpoint Phylogeny and Experimental Phylogenetic Analyses of Real and Synthetic Data

Mary E. Cosner*
Dept. of Plant Biology
Ohio State University

Robert K. Jansen†
Sect. of Integrative Biology
University of Texas, Austin

Bernard M.E. Moret
Dept. of Computer Science
University of New Mexico

Linda A. Raubeson
Dept. of Biological Sciences
Central Washington University

Li-San Wang
Dept. of Computer Science
University of Texas, Austin

Tandy Warnow‡
Dept. of Computer Science
University of Texas, Austin

Stacia Wyman§
Dept. of Computer Science
University of Texas, Austin

Abstract

The *breakpoint phylogeny* is an optimization problem proposed by Blanchette *et al.* for reconstructing evolutionary trees from gene order data. These same authors also developed and implemented `BPAnalysis` [3], a heuristic method (based upon solving many instances of the travelling salesman problem) for estimating the breakpoint phylogeny. We present a new heuristic for this purpose; although not polynomial-time, our heuristic is much faster in practice than `BPAnalysis`. We present and discuss the results of experimentation on synthetic datasets and on the flowering plant family Campanulaceae with three methods: our new method, `BPAnalysis`, and the neighbor-joining method [25] using several distance estimation techniques. Our preliminary results indicate that, on datasets with slow evolutionary rates and large numbers of genes in comparison with the number of taxa (genomes), all methods recover quite accurate reconstructions of the true evolutionary history (although `BPAnalysis` is too slow to be practical), but that on datasets where the rate of evolution is high relative to the number of genes, the accuracy of all three methods is poor.

Introduction

The genomes of some organisms have a single chromosome or contain single-chromosome organelles (such as mitochondria or chloroplasts) whose evolution is largely independent of the evolution of the nuclear genome for these organisms. Many single-chromosome organisms and organelles have circular chromosomes. Given a particular strand from a single chromosome, whether linear or circular, we can infer the ordering of the genes, along with directionality of the genes, thus representing each chromosome by an ordering (linear or circular) of signed genes. Note that picking the complementary strand produces a different ordering, in which the genes appear in the reverse direction and reverse order. The evolutionary process that operates on the chromosome can thus be seen as a transformation of signed orderings of genes.

The first heuristic for reconstructing phylogenetic trees from gene order data was introduced by Blanchette *et al.* in [3]. It sought to reconstruct the *breakpoint phylogeny* and was applied to a variety of datasets [4, 29].

A different technique for reconstructing phylogenies from gene order data was introduced by Cosner in [7]. We have modified her technique so that it requires less biological input. Our approach can also be described as a heuristic for the breakpoint phylogeny, although it is quite different in its technique from `BPAnalysis`. We call our approach *Maximum Parsimony on Binary Encodings (MPBE)*. The MPBE method first encodes a set of genomes as binary sequences and then constructs maximum-parsimony trees for these sequences.

We describe MPBE and compare it with two other methods (`BPAnalysis`, the heuristic designed and implemented by Blanchette *et al.* [3], and the polynomial-time, distance-based method neighbor-joining [25]) on both real and synthetic data. We find that, when the rates of evolution are sufficiently low, all methods recover very good estimates of the evolutionary tree (although `BPAnalysis` is much slower than MPBE). However, when the rates of evolution are high, all methods recover poor estimates of the evolutionary tree.

Definitions

We assume a fixed set of genes $\{g_1, g_2, \ldots, g_n\}$. Each genome is then an ordering (circular or linear) of some multi-subset of these genes, each gene given with an orientation that is either positive (g_i) or negative ($-g_i$). The multi-subset formulation allows for deletions or duplications of a gene. A linear genome is then simply a permutation on this multi-subset, while a circular genome can be represented in the same way under the implicit assumption that the permutation closes back on itself. For example, the circular genome on gene set $\mathcal{G} = \{g_1, g_2, \ldots, g_6\}$ given by $g_1, g_2, -g_3, g_4, g_6, g_2$ has one duplication of the gene g_2, has a deletion of the gene g_5, and has a reversal of the gene g_3. That same circular genome could be represented by several different linear orderings, each given by rotating the linear

---

*Deceased

†Supported by National Science Foundation grant DEB-9982091

‡Supported by National Science Foundation grant CCR-9457800 and a David and Lucile Packard Foundation Fellowship

§Contact author; address: Department of Computer Science, University of Texas, Austin, TX 78712-1188; phone (512) 420-0511; fax (512) 471-8885; email `stacia@cs.utexas.edu`

ordering above. Furthermore, the ordering $g_1, g_2, \ldots, g_n$, whether linear or circular, is considered equivalent to that obtained by considering the complementary strand, i.e., to the ordering $-g_n, -g_{n-1}, \ldots, -g_1$.

In tracing the evolutionary history of a collection of single-chromosome genomes, we use inversions, transpositions and transversions (inverted transpositions), because these events only rearrange gene orders. A more complex set of structural changes has been considered, e.g., in [7].

Let G be the genome with signed ordering (linear or circular) $g_1, g_2, \ldots, g_n$. An *inversion* between indices i and j, for $i < j$, produces the genome with linear ordering

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

If we have $j < i$, we can still apply an inversion to a circular (but not linear) genome by simply rotating the circular ordering until the two indices are in the proper relationship—recall that we consider all rotations of the complete circular ordering of a circular genome as equivalent.

A *transposition* on the (linear or circular) ordering G acts on three indices, i, j, k, with $i < j$ and $k \notin [i, j]$, and operates by picking up the interval $g_i, g_{i+1}, \ldots, g_j$ and inserting it immediately after g_k. Thus the genome G above (with the additional assumption of $k > j$) is replaced by

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

Once again, if we have $j > i$, we can still apply the transposition to a circular (but not linear) genome by first rotating it to establish the desired index relationship.

An *edit sequence* describes how one genome evolves into another through a sequence of these evolutionary events. For example, let G be a genome and let $p_1, p_2, \ldots, p_k$ be a sequence of evolutionary events operating on G; then $p_1, p_2, \ldots, p_k(G)$ defines another genome G'. When each operation is associated with a cost, then the *minimum edit distance* between two genomes G and G' is defined to be the minimum cost of any edit sequence transforming G into G'. As long as the cost of each operation is finite, any two genomes have finite edit distance.

The *inversion distance* between two genomes is the minimum number of inversions needed to transform one genome into another. The inversion distance between two signed genomes is computable in polynomial time for signed genomes [14, 17]; this algorithm is available in software (`signed_dist`), which we modified for use in our experiments to compute only distances (and thus run very much faster). We let *I* refer to the inversion distance. The *transposition distance* between two genomes is the minimum number of transpositions needed to transform one genome into the other. Computing the transposition distance is of unknown computational complexity.

When inversions, transpositions, and transversions are allowed, nothing is known about the computational complexity or approximability of computing edit distances; however, heuristics have been developed that can estimate the minimum edit distance for weighted sums of inversions, transpositions, and transversions. One such heuristic, `derange2` [2], is available in software; both we and Sankoff *et al.* [28] have used it for tree estimation purposes. We will refer to the distances computed by `derange2` as *ITT* distances.

Another distance between genomes that is not directly an evolutionary metric is the *breakpoint distance*. Given two genomes G and G' on the same set of genes, a breakpoint in G is defined as an ordered pair of genes (g_i, g_j) such that g_i and g_j appear consecutively in that order in G, but neither (g_i, g_j) nor $(-g_j, -g_i)$ appear consecutively in that order in G'. For instance, if $G = g_1, g_2, -g_4, -g_3$ and $G' = g_1, g_2, g_3, g_4$, then there are exactly two breakpoints in G: $(g_2, -g_4)$, and $(-g_3, g_1)$; the pair $(-g_4, -g_3)$ is not a breakpoint in G' since (g_3, g_4) appear consecutively and in that order in G'. The *breakpoint distance* is the number of breakpoints in G relative to G' (or vice-versa, since the measure is symmetric).

An *evolutionary tree* (or *phylogeny*) for a set S of genomes is a binary tree with $|S|$ leaves, each leaf labelled by a distinct element of S. A putative evolutionary tree is "correct" as long as this leaf-labelled topology is identical to the true evolutionary tree. The true phylogeny is unknown for real data. Synthetic data can be created with simulations using a given model tree; for such data, the "true" tree is the model tree and is thus known. Studies using such synthetic data are standard in the phylogenetics literature because they enable one to test the reliability of different methods. We now describe two approaches currently in favor for genome phylogeny reconstruction.

Maximum parsimony: Assume that we are given a tree in which each node is labelled by a genome. We define the cost of the tree to be the sum of the costs of its edges, where the cost of an edge is one of the edit distances between the two genomes that label the endpoints of the edge. Finding the tree of minimum cost for a given set of genomes and a given definition of the edit distance is the problem of *Maximum Parsimony for Rearranged Genomes (MPRG)*; the optimal trees are called the maximum-parsimony trees. (The MPRG problem is related to the more usual maximum-parsimony problem for biomolecular sequences, defined later.)

Distance-based methods: Distance-based methods for tree reconstruction operate by first computing all pairwise distances between the taxa in the dataset, thus computing a representation of the input data as a distance matrix d. In the context of genome evolution, this calculation of distances is done by computing breakpoint (*BP*) distances, or minimum edit (e.g. *I* or *ITT*) distances. Given the distance matrix d, the method computes an edge-weighted tree whose leaf-to-leaf distances closely fit the distance matrix. The most frequently used distance-based methods are polynomial-time methods such as neighbor-joining [25]. These methods do not explicitly seek to optimize any criterion, but can have good performance in empirical studies. In particular, neighbor-joining has shown excellent performance in studies based upon simulating biomolecular sequence evolution and is probably the most popular distance-based method.

Previous Phylogenetic Methods

Distance-Based Methods

There has been little use of distance-based methods for reconstructing phylogenies from gene order data. Blanchette *et al.* [4] recently evaluated two of the most popular polynomial-time distance-based methods for phylogenetic reconstruction, neighbor-joining and Fitch-Margoliash [11], for the problem of reconstructing the phylogeny of metazoans. They calculated a breakpoint distance matrix for inferring the metazoan phylogeny from mitochondrial gene order data. They found the trees obtained by these methods unacceptable because they violated assumptions about metazoan evolutionary history. Later, they examined a different dataset and found the result to be acceptable with respect to evolutionary assumptions about that dataset [27].

Computational Complexity of MPRG

MPRG seems to be the optimization criterion of choice; indeed, most approaches to reconstructing phylogenetic trees from gene order data have explicitly sought to find the maximum-parsimony tree with respect to some definition of genomic distances (inversion distances or a weighted sum of inversions, transpositions, and transversions). However, all these problems are NP-hard or of unknown computational complexity. Even the fundamental problem of computing optimal labels (genomes) for the internal nodes is very difficult: when only inversions are allowed, it is NP-hard, even for the case where there are only three leaves [6].

Breakpoint Phylogeny

Blanchette *et al.* [4] recently proposed a new optimization problem for phylogeny reconstruction on gene order data. In this problem, the tree sought is that with the minimum number of breakpoints rather than that with the minimum number of evolutionary events. It has long been known that the breakpoint distance is at most twice the inversion distance for any two genomes [14]. For some datasets, however, there can be a close-to-linear relationship between the breakpoint distance and either the inversion distance or the weighted sum of inversions and transpositions. When a linear relationship exists, the tree with the minimum number of breakpoints is also the tree with the minimum number of evolutionary events. Consequently, when a close-to-linear relationship exists, the tree with the minimum number of breakpoints may be close to optimal with respect to the number of evolutionary events. Blanchette *et al.* [4] observed such a close-to-linear relationship in a group of metazoan genomes (we computed the correlation coefficient between the two measures for their set and obtained a very high value of 0.9815) and went on to develop a heuristic for finding the breakpoint phylogeny.

Computing the breakpoint phylogeny is NP-hard for the case of just three linear signed genomes [23], a special case known as the *Median Problem for Breakpoints (MPB)*. Blanchette *et al.* showed that the MPB reduces to the travelling salesman problem (TSP) [26] and designed special heuristics for the resulting instances of TSP. Their overall heuristic for the breakpoint phylogeny considers each tree topology in turn. For each tree, it fills in internal nodes by computing medians of triplets of genomes iteratively (until no change occurs) using the TSP reduction, then scores the resulting tree. The best tree is returned at the end of the procedure. This heuristic is computationally intensive on several levels. First, the number of unrooted binary trees on n leaves is exponential in n (specifically it is $(2n-5)\cdot(2n-7)\cdot\ldots\cdot 3$), so that the outer loop is exponential in the number of genomes. Secondly, the inner loop itself is computationally intensive, since computing the median of three genomes is NP-hard [23] and because the technique used by Blanchette *et al.* involves solving many instances of TSP in a reduction where the number of cities equals the number of genes in the input. Finally, the number of instances of TSP can be quite large, since the procedure iterates until no further change of labelling occurs within the tree. Thus the computational complexity of the entire algorithm is exponential in *each* of the number of genomes and the number of genes.

The accuracy of `BPAnalysis` for the breakpoint phylogeny problem depends upon the accuracy of its component heuristics. While it evaluates every tree, the labelling given to each tree is only locally optimal: although it solves TSP exactly at each node, it labels nodes with an iterative method that can easily be trapped at a local optimum. In our experiments, we have found that `BPAnalysis` often needed to be run on several different random starting points in order to score a given tree accurately. This is typical of hill-climbing heuristics, but will affect the running time proportionally.

Our New Method: Maximum Parsimony on Binary Encodings of Genomes (MPBE)

In this section we describe a new approach to reconstructing phylogenies from gene order data. This new method is derived from an earlier method developed by Cosner in [7]. Like Cosner's technique, our method encodes the genome data as binary sequences and seeks a maximum-parsimony tree for these sequences, although our encoding is very simple and uses no biological assumptions. However, our method has a second phase, in which we select, from the maximum-parsimony trees we find, the trees that have minimum length with respect to some evolutionary metric (such as the inversion distance or the *ITT* distance). We now describe the two phases of the MPBE approach.

Phase I: solving maximum parsimony on binary encodings of genomes We begin by defining the binary encoding. We note all ordered pairs of signed genes (g_i, g_j) that appear consecutively in at least one of the genomes. Each such pair defines a position in the sequences (the choice of index is arbitrary). If (g_i, g_j) *or* $(-g_j, -g_i)$ appear consecutively in a genome, then that genome has a 1 in the position for this ordered pair, and otherwise it has a 0. These "characters" can also be weighted. (In this study, we did not weight any characters; however, in the study reported in [7], character weighting was used, along with other characters such as gene segment insertions and deletions, duplications of inverted repeats, etc. Thus, the method can be extended to allow for evolutionary events more complex than gene order changes.)

Let $H(e)$ be the Hamming distance between the sequences labelling the endpoints of the edge e—the Hamming distance between two sequences is the number of positions in which they differ. We define the *Binary Sequence Maximum Parsimony (BSMP)* problem as follows:

the input consists of a set S of binary sequences, each of length L; the output is a tree T with leaves labelled by S and internal nodes labelled by additional binary sequences of length L in such a way as to minimize $\Sigma H(e)$ as e ranges over the edges of the tree. The trees with the minimum score are called maximum-parsimony trees.

Our first phase then operates as follows. First, each genome is replaced by a binary sequence. The BSMP problem is then solved exactly or approximately, depending upon the dataset size. BSMP is NP-hard [12], but fast heuristics exist that are widely available in standard phylogeny software packages, such as `PAUP*` [30]. Although no study has been published on the accuracy of these heuristics on large datasets, it is generally believed that these heuristics usually work well on datasets of size up to about 40 genomes. Moreover, exact solutions on datasets of up to about 20 genomes can be obtained through branch-and-bound techniques in reasonable amounts of time; consequently, BSMP has been solved exactly in some cases.

Phase II: screening the maximum-parsimony trees Once the maximum-parsimony trees are obtained, the internal nodes are then re-labelled by circular signed gene orders (recall that the labelling of internal nodes obtained in the first phase of MPBE is with binary sequences, not with circular signed genomes). The relabelling is obtained by giving the maximum-parsimony tree as a constraint to `BPAnalysis`, thus producing a labelling of each internal node with circular signed gene orders which (hopefully) minimizes the breakpoint distance of the tree. The labelling also allows us to score each tree for the *I* or *ITT* distance. The tree that minimizes the total cost is then returned.

Running time of MPBE The computational complexity of MPBE, while less than that of `BPAnalysis`, remains high. The maximum-parsimony evaluation of a single tree in the search space takes polynomial time (the precise time is $\Theta(nk)$, where n is the number of genomes and k is the number of genes in each genome). Thus, the first phase is exponential in the number of genomes, but polynomial in the number of gene segments. However, we have the option of doing hill-climbing through tree space (rather than exhaustive search) and thus can reduce the computational effort by comparison to the exhaustive search strategy of `BPAnalysis`. In the second phase, we give the maximum-parsimony trees to `BPAnalysis` as constraint trees. Thus we also call `BPAnalysis`, (which is exponential in both the number of genomes and the number of gene segments), but only on a (typically small) subset of the possible trees. Finally, we compute the cost of each node-labelled tree with respect to *I* or *ITT* distances. Computing *ITT* distances is fast, although `derange2` can be inexact. Computing inversion distances with the original `signed_dist` is fairly slow because the program also returns inversions, but fast when it is modified to compute only distances. Overall, Phase II is more computationally expensive than Phase I.

MPBE as a Heuristic for the Breakpoint Phylogeny Suppose T is the breakpoint phylogeny for the set $G_1, G_2, \ldots, G_n$ of genomes. Each node in T is labelled by a circular ordering of signed genes and the number of breakpoints in the tree is minimized. If each node in the tree is replaced by its binary encoding, the parsimony length of the tree is exactly twice the number of breakpoints in the tree. Thus, seeking a tree with the minimum number of breakpoints is exactly the same as seeking a tree (based upon binary encodings) with the minimum parsimony length, *provided that* each binary sequence can be realized by a circular ordering of signed genes.

This last point is significant, because not all binary sequences are derivable from signed circular orderings on genomes! In other words, it is possible for the MPBE tree (that is, the tree with minimal parsimony length for the binary sequence encodings of the genomes) to have internal nodes whose binary sequence encodings cannot be realized by circular orderings of signed genes. If the sequences in the internal nodes of an MPBE correspond to signed circular orderings, then the tree will be a breakpoint phylogeny. If they do not, then the MPBE trees and the breakpoint phylogenies may be disjoint.

Consider rephrasing the breakpoint phylogeny problem as follows. We say that a binary sequence is a "circular genome sequence" if it is the binary encoding of a circular genome under a given representation method. The breakpoint phylogeny problem is to find the tree of minimum parsimony length, with leaves labelled by the binary encodings of the circular genomes and internal nodes labelled by "circular genome sequences." Since MPBE does not restrict the labels of internal nodes to circular genome sequences, it searches through a larger space for the the labels of internal nodes and thus may assign labels to nodes that are not circular genome sequences. When this happens, MPBE will fail to find feasible solutions to the breakpoint phylogeny problem.

MPBE is thus a heuristic for breakpoint phylogeny, but it produces labellings of the internal nodes that are binary sequences; as we discussed, these may not correspond to circular orderings of signed gene segments. Therefore we must relabel the internal nodes by circular genome sequences (using `BPAnalysis` or other such techniques) so that the breakpoint distance of the trees can be computed. This is why we have included Phase II in our method.

Since each of the problems we solve (maximum parsimony on binary sequences, the median problem for breakpoints, and the *ITT*) is either known or conjectured to be NP-hard, the accuracy of the heuristics will determine whether we find globally optimal or only locally optimal solutions.

Chloroplast Data Analysis

Chloroplast DNA is generally highly conserved in nucleotide sequence, gene order and content, and genome size [22]. The genomes contain approximately 120 genes involved in photosynthesis, transcription, translation, and replication. Major changes in gene order, such as inversions, gene or intron losses, and loss of one copy of the inverted repeat, are rare. These genes are very useful as phylogenetic markers because they are easily polarized and exhibit very little homoplasy when properly characterized [9]. In groups in which more than one gene order change has been detected, the order of events is usually readily determined (e.g., [15, 18]). Chloroplast DNA gene order changes have been useful in phylogenetic reconstruction in many plant groups (see [9]). These changes have considerable potential to resolve phylogenetic relationships and provide valuable insights into the mechanisms of cpDNA evolution.

The Campanulaceae cpDNA Dataset We have used the chloroplast genomes of the flowering plant family Campanulaceae for a test case of our technique. In earlier work [7], Cosner obtained detailed restriction site and gene maps for 18 genera of the Campanulaceae and the outgroup tobacco. (An "outgroup" is a taxon selected so that any two other members of the set are more closely related than either is to the outgroup; the use of outgroups in phylogenetic analysis allows us to root the tree). She then used a variant of the MPBE analysis described above to obtain a phylogenetic analysis of these genera. We analyzed the same dataset, but, in order to apply the MPBE method, had to remove two incompletely mapped genera from the dataset. We also removed the repeated regions, causing certain pairs of genera (which differ only in terms of insertions and deletions of gene segments or expansions and contractions of the inverted repeat) to become indistinguishable, reducing our dataset to 13 genera from the original 19.

Data Analysis We used gene maps to encode each of the 13 genera as a circular ordering of signed gene segments. The result is shown in Figure 1.

We used these 13 circular orderings as input to `BPAnalysis`. The program spent over 43 hours of computation time without completing. We also encoded these orderings with our binary encoding technique and conducted an analysis of the resulting binary sequences under maximum parsimony using the branch-and-bound procedure of `PAUP*`. (These sequences are available on our web page [32], but can also be calculated directly from the gene order data; the parameters used in the parsimony analysis with `PAUP*` are also available there.) We obtained four maximum-parsimony trees from this dataset. We inferred circular orderings of signed gene segments for each internal node by giving each binary tree as a constraint tree to `BPAnalysis`. This produces a tree in which each node (internal and leaf) is represented by circular signed orderings on genes, potentially minimizing the number of breakpoints in the tree. (An actual minimization is not guaranteed, because `BPAnalysis` uses hill-climbing on each fixed-tree and thus may find only a local minimum.) We then scored each tree for the number of breakpoints. Interestingly, the labelling of internal nodes obtained by `BPAnalysis` produced the same number of breakpoints on all four trees, namely 89.

We note that the best breakpoint score obtained in 43 hours of computation by `BPAnalysis` from the original orderings was 96—much larger than the breakpoint score obtained by our parsimony analysis of binary sequences.

We then scored each tree (using the labels assigned by `BPAnalysis`) for the *I* distance using our modified `signed_dist` and for the *ITT* distance using `derange2` with relative weights of 2.1 for transpositions and transversions *vs.* 1 for inversions. Using this weighting scheme, the first tree has a total of 40 inversions and 12 transpositions/transversions; the second has 48 inversions and 18 transpositions/transversions; the third has 40 inversions and 12 transpositions/transversions; and the fourth has 67 inversions and 43 transpositions/transversions. Thus, the first and third trees are superior (under this analysis) to the second and fourth. We then evaluated the first and third trees with respect to the inversion distance, given the labelling on internal nodes obtained by `BPAnalysis`:

Trachelium
(1–15)(76–56)(53–49)(37–40)(35–26)(44–41)(45–48)
(−36)(25–16)(90–84)(77–83)(91–96)(55–54)(105–97)

Campanula
(1–15)(76–49)(39–37)(40)(35–26)(44–41)(45–48)
(−36)(25–16)(90–84)(77–83)(91–96)(55–54)(105–97)

Adenophora
(1–15)(76–49)(39–37)(29–35)(40)(26–27)(44–41)(45–48)
(−36)(25–16)(90–84)(77–83)(91–96)(55–54)(105–97)

Symphyandra
(1–15)(76–56)(39–37)(49–53)(40)(35–26)(44–41)(45–48)
(−36)(25–16)(90–84)(77–83)(91–96)(55–54)(105–97)

Legousia
(1–15)(76–56)(27–26)(44–41)(45–48)(36–35)(25–16)
(90–84)(77–83)(91–96)(5–8)(55–53)(105–98)(28–34)
(40–37)(49–52)(−97)

Asyneuma
(1–15)(76–57)(27–26)(44–41)(45–48)(36–35)(25–16)(89–84)
(77–83)(90–96)(105–98)(28–34)(40–37)(49–52)(−97)

Triodanus
(1–15)(76–56)(27–26)(44–41)(45–48)(36–35)(25–16)
(89–84)(77–83)(90–96)(55–53)(105–98)(28–34)(40–37)
(49–52)(−97)

Wahlenbergia
(1–11)(60–49)(37–40)(35–28)(12–15)(76–61)(27–26)
(44–41)(45–48)(−36)(54)(25–16)(90–84)(77–83)(91–96)
(−55)(105–97)

Merciera
(1–10)(49–53)(28–35)(40–37)(60–56)(11–15)(76–61)
(27–26)(44–41)(45–48)(−36)(54)(25–16)(90–85)(77–84)
(91–96)(−55)(105–97)

Codonopsis
(1–8)(36–18)(15–9)(40)(56–60)(37–39)(44–41)(45–53)
(16–17)(54–55) (61–76)(96–77)(105–97)

Cyananthus
(1–8)(29)(36–26)(40)(56–60)(37–39)(25–9)(44–48)
(55–49)(61–96)(105–97)

Platycodon
(1)(8)(2–5)(29–36)(56–50)(28–26)(9)(49–45)(41–44)
(37–40)(16–25)(10–15)(57–59)(6–7)(60–96)(105–97)

Tobacco
(1–105)

Figure 1: 12 genera of Campanulaceae and the outgroup Tobacco, as circular orderings of signed gene segments. We represent each circular ordering as a linear ordering, beginning at gene segment 1. In order to conserve space (and make the rearrangements easier to observe), we have represented each ordering in a compact representation by noting the maximal intervals of consecutive gene segments with the same orientation. Thus the sequence 1, 2, −4, −3, 5, 6, 7, 10, 8, 9 would be represented as (1–2)(4–3)(5–7)(10)(8–9). Tobacco has the "unrearranged" ordering 1, 2, . . . , 105, which we represent as (1–105).

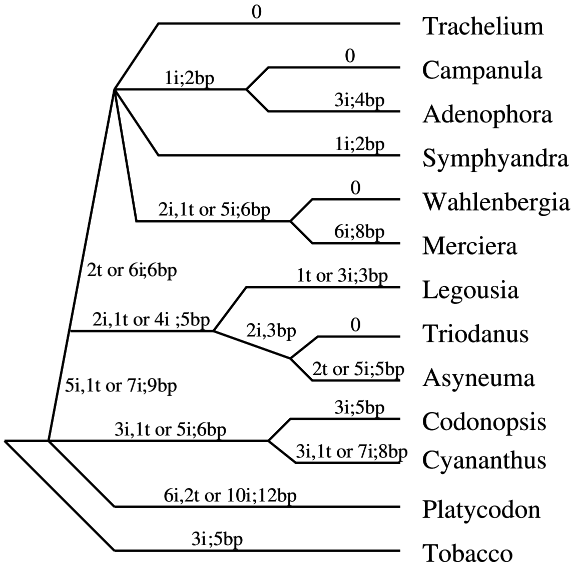


Figure 2: The reconstructed phylogeny of 12 genera of Campanulaceae and the outgroup tobacco based upon an MPBE analysis of 185 binary characters. Above each edge are given the number of inversions and transpositions/transversions, the number of inversions in an inversion-only scenario, and the number of breakpoints.

the first tree has a total number of 68 inversions, while the third has 67. Both trees have zero-length edges (i.e., the endpoints of some edges have the same gene orderings). When these edges are contracted, the two trees are identical. The contracted tree is shown in Figure 2. Interestingly, that tree is also a contraction of each of the trees obtained by the Cosner analysis [7] on the original 19 genera, and then restricted to the subset of 13 genera. Thus our restricted subset of characters is compatible with the more biologically rich analysis performed by Cosner, in which insertions, deletions, duplications, contractions/expansions of the inverted repeat, etc., were also used.

We computed neighbor-joining trees (using `Phylip` [10]) on three different distance matrices: the *I* matrix computed using our modified `signed_dist`, the *ITT* matrix computed with `derange2` with relative weights of 1, 2.1, and 2.1, and the breakpoint matrix computed using `BPAnalysis`. We show the `derange2` distance matrix in Table 1; the other distance matrices are on our web page [32].

The three neighbor-joining trees have identical topologies, differing only in their edge-weights, while the MPBE trees differ from the NJ trees by at most 2 edges; see Table 2. The similarity between all reconstructed trees indicates a high level of confidence in the the accuracy of the common features of the phylogenetic reconstructions (see our web page for the strict consensus tree).

The conditions under which these genomes evolved (low rates of evolution and a large number of gene segments) are probably responsible for this high level of similarity, which is observable at various levels. For instance, the breakpoint distance and the *ITT* distance (using relative costs of 1, 2.1, and 2.1) are very closely related, as illustrated in Figure 3. (The high correlation coefficient ρ indicates that the two distances stand in a nearly linear relationship to each other.)

Table 1: The *ITT* distance matrix for the Campanulaceae dataset, computed using `derange2` and a 2.1 weight ratio

	Tra	Cam	Ade	Sym	Leg	Asy	Tri	Wah	Mer	Cod	Cya	Pla	Tob
Tra	0.0	1.0	4.0	1.0	8.3	10.4	8.3	4.1	8.1	15.2	14.1	19.2	10.0
Cam	1.0	0.0	3.0	2.0	9.3	11.4	9.3	5.1	9.2	15.1	15.2	20.2	11.2
Ade	4.0	3.0	0.0	5.1	12.1	14.3	12.1	8.1	11.2	16.2	15.2	20.2	13.1
Sym	1.0	2.0	5.1	0.0	9.2	11.4	9.3	5.1	9.1	14.2	13.3	20.2	11.1
Leg	8.3	9.3	12.1	9.2	0.0	8.4	4.1	12.2	14.3	18.1	16.1	23.2	14.2
Asy	10.4	11.4	14.3	11.4	8.4	0.0	4.2	12.4	16.2	18.2	16.2	21.1	12.2
Tri	8.3	9.3	12.1	9.3	4.1	4.2	0.0	12.2	14.4	18.2	15.2	21.2	12.2
Wah	4.1	5.1	8.1	5.1	12.2	12.4	12.2	0.0	6.0	18.1	16.2	23.1	14.2
Mer	8.1	9.2	11.2	9.1	14.3	16.2	14.4	6.0	0.0	17.2	16.3	24.1	16.1
Cod	15.2	15.1	16.2	14.2	18.1	18.2	18.2	18.1	17.2	0.0	8.3	18.2	10.2
Cya	14.1	15.2	15.2	13.3	16.1	16.2	15.2	16.2	16.3	8.3	0.0	16.3	10.2
Pla	19.2	20.2	20.2	20.2	23.2	21.1	21.2	23.1	24.1	18.2	16.3	0.0	13.3
Tob	10.0	11.2	13.1	11.1	14.2	12.2	12.2	14.2	16.1	10.2	10.2	13.3	0.0

Table 2: The number of missing edges (i.e. false negatives) out of 10 possible, for various reconstruction methods on the Campanulaceae data of Figure 1. MPBE1 through MPBE4 are the four most parsimonious trees by the first phase of the MPBE method. NJ refers to the tree obtained by neighbor-joining on the three distance matrices (these were identical).

	NJ	MPBE1	MPBE2	MPBE3	MPBE4
NJ	0	1	2	1	2
MPBE1	1	0	1	1	2
MPBE2	2	1	0	2	1
MPBE3	1	1	2	0	1
MPBE4	2	2	1	1	0

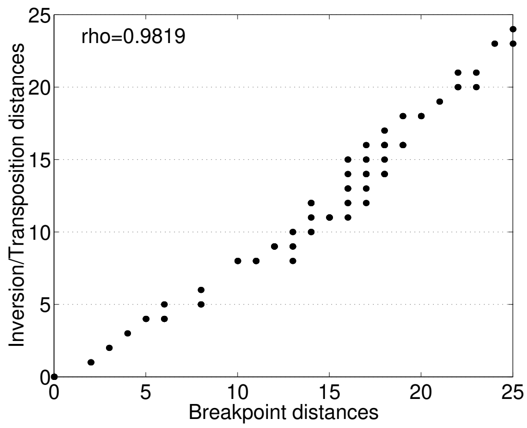


Figure 3: Comparison of distance calculations on the Campanulaceae dataset

These observations suggest that this dataset forms an easy case for phylogeny reconstruction. We therefore began an experimental investigation into the performance of methods for phylogenetic reconstruction from gene order data to determine whether all methods continue to perform well under a larger range of model conditions and whether there are model conditions under which some methods consistently outperform others.

Our Experimental Investigation

We developed a simple simulator that, given a model tree and parameters, mimics the evolutionary history of a genome and produces a set of genomes. Using both actual and synthetic model trees, we then reconstruct the putative phylogeny using the various methods proposed to date as well as our new method (only through Phase I). These putative phylogenies are then compared to the model tree.

We computed *BP* distances ourselves, *I* distances using our modified `signed_dist`, and *ITT* distances using `derange2`. Since we generate the synthetic data ourselves, we can observe the actual process that happens during the simulation. In particular, we can note when no evolutionary event (inversion, transposition, or transversion) takes place on an edge, enabling us to derive a better estimate of the quality of a reconstruction, since no reconstruction method can recover an edge (other than by guessing) when no evolutionary event happens on it.

Terminology

Let T be a tree leaf-labelled by the set S. Given an edge e in T, the deletion of the edge from T produces a bipartition π_e of G into two sets. The set $C(T) = \{\pi_e \colon e \in E(T)\}$ uniquely defines the tree T; this characterization is called the *character encoding* of T. Given a collection of trees $T_1, T_2, \ldots, T_k$, each leaf-labelled by S, we define the *strict consensus* of the trees to be that unique tree T_{sc} defined by $C(T_{sc}) = C(T_1) \cap C(T_1) \cap ... \cap C(T_n)$. This is the maximally resolved tree which is a common contraction of each tree T_i. Character encodings are used to compare trees and to evaluate the performance of a phylogenetic reconstruction method. Let T be the "true" tree and let T' be the estimate of T. Then the *false negatives* of T' with respect to T are those edges e that obey $\pi_e \in C(T) - C(T')$, i.e., edges in the true tree that the method fails to infer. The *false positives* of T' with respect to T are those edges e that obey $\pi_e \in C(T') - C(T)$, i.e., edges in the inferred tree that do not exist in the true tree and should not have been inferred. Note that every trivial bipartition (induced by the edge incident to a leaf) exists in every tree. Consequently, false positives and false negatives are calculated only with respect to the internal edges of the tree. These are sometimes expressed as a percentage of the number of internal edges.

Experimental Setup

The simulator: The *Nadeau-Taylor* [20] is the standard model of genome evolution; it assumes that only inversions occur during the evolutionary history of a set of genomes, that all inversions are equally likely, and that the number of inversions on each edge obeys a Poisson distribution. We designed a simulator to enable us to generate gene orders under the Nadeau-Taylor model, as well as under more complex models in which transpositions and transversions also occur. The input to the simulator is the topology of a rooted tree T (which determines the number of genomes), the number k of genes in the genomes, the expected number λ_e of inversions on each edge e, and a constant C denoting the relative cost of inversions to transpositions and transversions. The number of each of these events is a random variable obeying a Poisson distribution. Thus, we generate a random leaf-labelled tree, randomly assign lengths (chosen uniformly from various ranges) to each edge to represent the expected number of inversions per edge, and feed the result to the simulator.

The simulator generates signed circular orderings of the genes as follows. The root is assigned the identity gene ordering $g_1, g_2, \ldots, g_k$. When traversing an edge e with expected number of inversions λ_e, three random numbers are generated. The first determines the actual number of inversions on that edge; the second the actual number of transpositions; and the third the actual number of transversions. Once the number of each event is determined, the order of these events is randomly selected. This process produces a set of circular signed gene orders for each genome at the leaves of the model tree. The simulator also produces other information for use in performance studies: the gene orders computed at each internal and leaf node, the actual number of inversions, transpositions, and transversions that occurred during that run of the simulator on each edge, and the "true distance matrix" D between every pair of leaves in the tree. (Given the actual number of inversions, transpositions, and transversions that occur on each leaf-to-leaf path, the distance between the two leaves is the number of inversions plus the weighted cost of the transpositions and transversions.) Note that this matrix defines the model tree, with each edge weighted according to the weighted cost of the events on that edge. As long as every edge has at least one event, standard distance methods (such as neighbor-joining [25]), when applied to the matrix D are *guaranteed* to recover the true tree topology (see [31]).

Phylogenetic methods: For each dataset generated by the simulator, we computed the *BP*distance and at least one of the *I* or *ITT* distances. We computed neighbor-joining trees (as implemented in `Phylip` [10]) on these distance matrices. We denote the neighbor-joining trees for the different distance matrices by NJ(*BP*), NJ(*ITT*), and NJ(*I*). The MPBE heuristic is only computed through Phase I, so that we return the strict consensus of all maximum-parsimony trees we compute and do not perform any additional screening.

We wrote software to obtain binary sequence representations of the signed circular gene orderings. We solved maximum parsimony exactly on datasets of up to 20 taxa using the branch-and-bound program of `PAUP*` and heuristically for larger datasets; naturally, when we use a heuristic to "solve" maximum parsimony, we are not guaranteed to find globally optimal solutions, only locally optimal ones. We used the TBR (tree-bisection-reconnection) branchswapping heuristic of PAUP*, with 100 initial starting points (trees obtained by optimizing the sequential placement of taxa, randomly ordered, into the tree). We kept up to 10,000 trees in memory and included auto-increment in the analysis. As these searches often returned hundreds or thousands of local optima, we computed the strict consensus and majority consensus trees of the local optima. In the following, we denote these trees by MPBE, "maximum-parsimony tree(s) for the binary encoding of the genome data."

We labelled internal nodes of each tree with circular orderings of signed genes using BPAnalysis, and scored the resultant node labelled trees under breakpoint distances (ourselves), *I* distances (using signed_dist) and *ITT* distances (using derange2).

We were unable to run BPAnalysis to completion on our datasets because of its computational complexity; however, we did use BPAnalysis in a restricted search, by providing it with the strict consensus of the trees we obtained using our other techniques as a "constraint" tree. This way of using BPAnalysis makes it evaluate all binary trees that resolve the constraint tree. Since all trees we found using other methods will be in the set of refinements of the constraint tree, this strategy enables BPAnalysis to evaluate these trees and to find other, potentially better, trees.

Experiment 1: Neighbor-Joining on Synthetic Data

The first round of experiments focussed on the performance of neighbor-joining under a variety of model conditions. We generated three random model trees. T_A had 20 genomes and 20 genes, with high rates of change (3 to 10 inversions per edge on average), T_B had 20 genomes and 20 genes, but low rates of change (1 to 3 inversions per edge on average), and T_C had 20 genomes and 105 genes, with low rates of change (1 to 3 inversions per edge on average). In each of the 50 runs of this experiment, we ran our simulator on each random tree with relative costs of 1, 2.1, and 2.1 for inversions, transpositions, and transversions. This simulation generated gene orders for the 20 genomes at the leaves. Each run thus gives rise to three matrices: D, *BP*, and *ITT* (true distances, breakpoint distances, and *ITT* distances). The matrix D is determined during the simulation, the matrix *BP* can be calculated exactly in linear time, but the matrix *ITT* is estimated using derange2, perhaps with significant errors. We constructed the neighbor-joining trees on the *BP* and *ITT* matrices, thus producing trees NJ(*BP*) and NJ(*ITT*) (see our earlier discussion). These were then compared with the model tree, scoring the comparison in terms of false negatives (since all trees are binary, false positive and false negative rates are identical). Note that, on trees with low rates of evolution (T_B and T_C), slightly more than 3 edges per tree have no changes; in these cases, a false negative rate of around 3 would indicate complete success, so that all false negative rates should be scaled down accordingly. (3 edges represents 18% of the interior edges of T_B and T_C; thus false error rates should be decreased by about 12% to make up for the zero-length edges and the expected accuracy of a guessed resolution of an unresolved tree.) The results are summarized in Table 3.

In Figures 4, 5, and 6, we compare the distances *BP* and *ITT* to the true distances, on trees T_A, T_B, and T_C, respectively. We also give the correlation coefficient between the two measurements in each figure—a statistical measure of the degree to which the two distances are linearly related. Note how closely correlated the breakpoint and *ITT* distances are in the second and third cases (and, to a lesser extent, in the first case), indicating a linear or nearly linear relationship. In contrast, the true distance shows no particular correlation to the other two distances in the first two trees. In tree T_C, all three distances are closely correlated, reflecting the relative lack of evolution and overall simplicity of that tree.

Table 3: Average false negatives of the neighbor-joining trees from the matrices *BP* and *ITT*. Values in parentheses are the percentages over the 17 nontrivial bipartitions in each model tree.

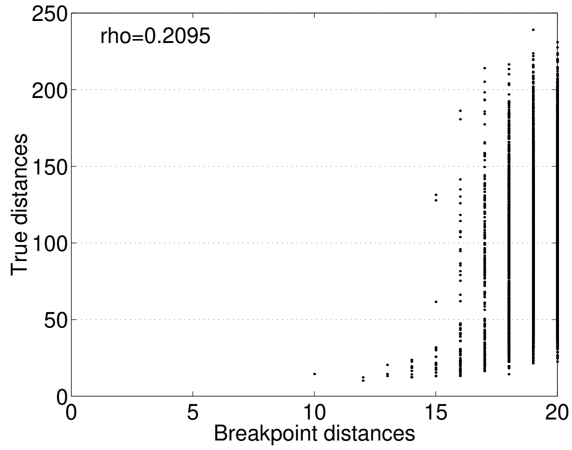
	Avg. Number (Avg. %) of False Negatives	
model tree	NJ(*BP*) *vs.* model	NJ(*ITT*) *vs.* model
T_A	14.84 (87.29%)	15.34 (90.24%)
T_B	8.22 (48.35%)	7.70 (45.29%)
T_C	2.24 (13.18%)	1.90 (11.18%)

Neighbor-joining does quite well on the third tree T_C, but poorly on T_B and very poorly on T_A. Furthermore, its performance does not appear to depend upon the choice of edit distance, but it does correlate well with the accuracy of the edit distance calculation (*BP* or *ITT*) with respect to the true distance D. This accuracy in turn seems to depend upon the rate of evolution relative to the number of genes in the genomes.

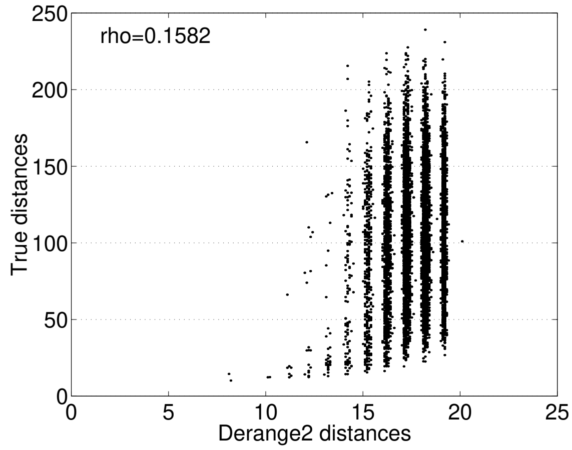
Experiment 2—All Methods on Synthetic Data

In this experiment, we simulated only inversion events and so used the Nadeau-Taylor model of evolution. We varied the number of genomes, the number of genes, and rates of evolution. We computed *BP* distances, *ITT* distances, and *I* distances and then calculated neighbor-joining trees NJ(*BP*), NJ(*ITT*) (and sometimes NJ(*I*)) for these distance matrices. We computed the strict consensus of the trees obtained during Phase I of the MPBE method; in some cases we also computed trees using BPAnalysis with the strict consensus of various recovered trees given as a constraint tree (see the discussion above). We compared each tree to the model tree and computed false negatives and false positives. Our results are summarized in Table 4. As the model trees and neighbor-joining trees are always binary, we only report false negative rates for neighbor-joining trees. On the other hand, we report both false negatives and false positives for the MPBE strict consensus trees.

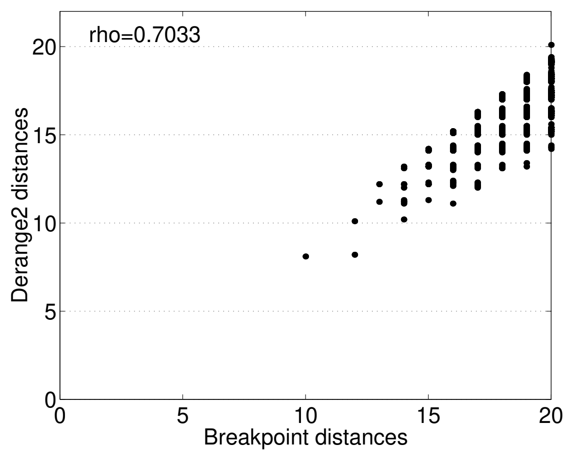
These results indicate that the various methods (neighbor-joining on *BP* and *ITT* distances and maximum parsimony on binary encodings of gene order data) have the same qualitative performance on all model conditions we examined. That is, we cannot as yet identify a model condition under which one method will outperform the others. However, one other trend is clear: all methods do well when the rate



(a) breakpoint vs. true distances

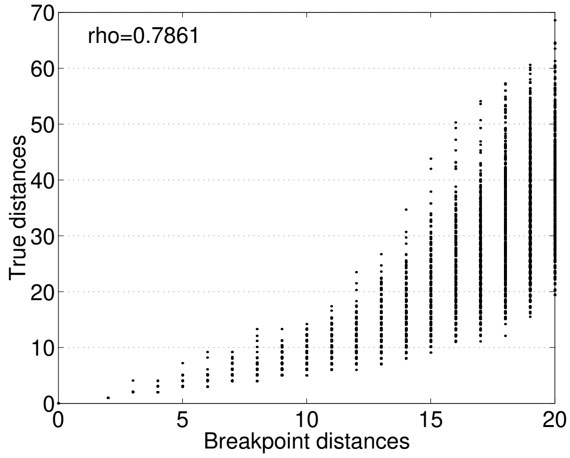


(b) ITT vs. true distances

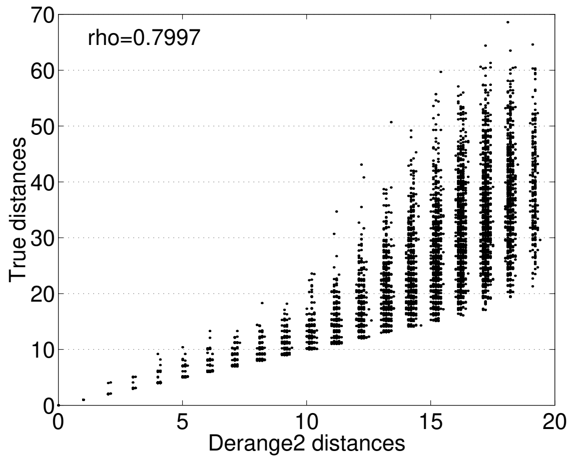


(c) breakpoint vs. ITT distances

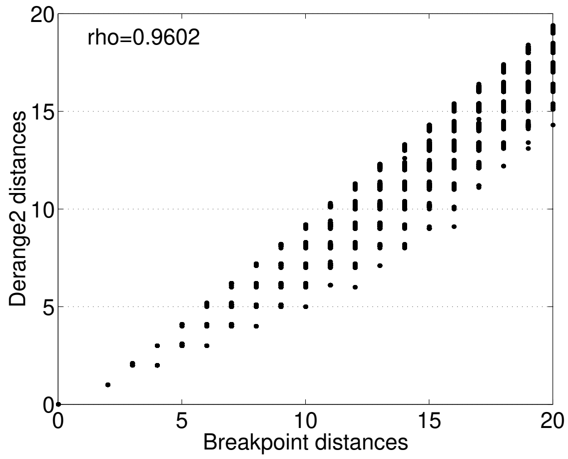
Figure 4: Comparison of distances on model tree T_A.



(a) breakpoint vs. true distances

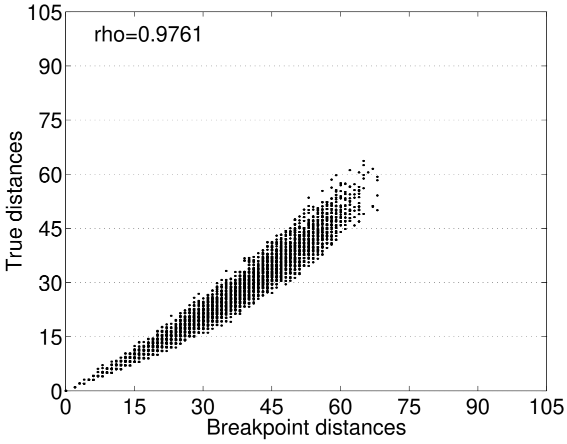


(b) ITT vs. true distances

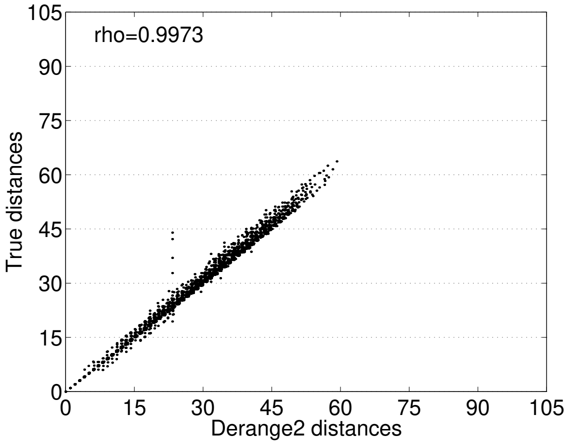


(c) breakpoint vs. ITT distances

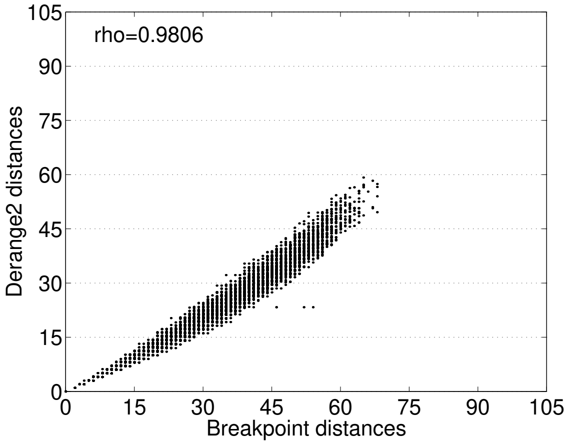
Figure 5: Comparison of distances on model tree T_B.



(a) breakpoint vs. true distances



(b) ITT vs. true distances



(c) breakpoint vs. ITT distances

Figure 6: Comparison of distances on model tree T_C.

Table 4: The false negative rates (in %) with respect to the true tree of various reconstruction methods for various model trees and rates of evolution.

Genomes	Genes	Inv./Edge[a]	NJ(*BP*)[b]	NJ(*ITT*)[c]	NJ(*I*)[d]	MPBE[e]
10	105	9–11	0	0	0	0 / 0
25	105	1–5	9.09	4.55		9.09/ 4.55[f]
25	105	4–6	0	0		0 / 0
25	105	1–10	9.09	0		4.55/ 4.55
40	105	1–5	13.51	10.81		10.81/ 2.70[f,g]
40	105	1–10	16.22	0		2.70/ 2.70[g]
25	37	1–5	22.73	9.09	4.55	27.27/ 9.09[f]
25	37	1–10	9.09	13.64	13.64	31.82/13.63[f]
40	37	1–5	37.84	10.81	18.92	35.14/ 2.70[f,g]
40	37	1–10	32.43	32.43	32.43	48.65/24.32[f,g]
20	20	3–10	49.41	60.00	60.00	65.88/20.00[f]
60	20	3–5	66.66	68.42		75.43/57.89[f,g]

[a] the expected number of inversions per edge
[b] neighbor-joining on the breakpoint distance matrix
[c] neighbor-joining on the *ITT* distance matrix computed by `derange2`
[d] neighbor-joining on the inversion distance matrix computed by `signed_dist`
[e] maximum parsimony on the binary encoding of the genomes; includes both false negative and false positive rates
[f] the strict consensus of all maximum-parsimony trees
[g] dataset too large for branch-and-bound parsimony, heuristic used instead

of change on an edge is low relative to the number of genes, while their performance decreases as this rate increases. What is surprising is that the rate at which their performance decreases appears to be the same.

We then examined the performance of `BPAnalysis` with respect to solving the breakpoint phylogeny problem. We were also interested in determining whether the model tree is one of the breakpoint phylogenies (and hence determine whether solving the breakpoint phylogeny is a good approach to reconstructing trees from gene order data). However, our results for `BPAnalysis` are limited, because of the extreme slowness of the program; we found that the trees obtained by `BPAnalysis` were almost always the same trees found by using Phase I of the MPBE method, provided that we let `BPAnalysis` run long enough. Therefore, `BPAnalysis` seems to be doing a reasonably effective job at solving the breakpoint phylogeny problem.

It seems that the breakpoint phylogeny may not always be a good estimate of the model tree. In our experiments, the breakpoint phylogeny is a good estimate of the model tree only when the rates of evolution on each edge are low relative to the number of genes. In these cases, the model tree is one of the breakpoint phylogenies or is close to optimal. In other cases, the breakpoint score of the model tree is significantly larger than the breakpoint scores found by either MPBE or `BPAnalysis`. This discrepancy suggests that, for model conditions in which the rates of evolution are high, breakpoint phylogenies are unlikely to be accurate estimates of the true evolutionary tree.

Software Issues

Running time is always important in comparing phylogenetic methods. While neighbor-joining runs in polynomial time, neither MPBE nor BPAnalysis does.

We timed each method on the Campanulaceae dataset, using a Sun E5500 with 2GB of memory running Solaris 2.7. The first phase of MPBE took 0.15 seconds to complete on the Campanulaceae dataset (finding the four maximum-parsimony trees with PAUP* took 0.15 seconds on a Macintosh G4). The second phase took somewhat longer. Labelling the internal nodes with BPAnalysis took 0.38 seconds for each tree. Computing inversion distances on each edge using our modified signed_dist took 0.02 seconds and computing *ITT* distances on each edge using derange2 took 0.01 seconds. The second phase of MPBE thus took about 4.5 seconds in all. Hence the complete MPBE analysis ran in under 5 seconds.

We also attempted to time BPAnalysis on the real dataset, but it did not complete its search, so we had to estimate the amount of time it took per tree and extrapolate. Our experiments suggest that BPAnalysis evaluates 120 trees a minute; at this rate, since the number of trees on 13 leaves is 13,749,310,575, BPAnalysis would take well over 200 years to complete its search of tree space for our problem. Blanchette *et al.* did complete their analysis of the metazoan dataset, which has 11 genomes on a set of 37 genes. This is a much easier problem, as there are far fewer trees to examine (only 2,027,025) and as scoring each tree involves solving a smaller number of TSP instances on a much smaller number of cities (37 rather than 105). Overall, it is clear that datasets of sizes such as ours are currently too large to be fully analyzed by BPAnalysis.

In view of these observations, our new method stands as a good compromise between speed and accuracy. Neighbor-joining is faster (guaranteed polynomial-time), but returns only one tree and thus tells us little about the space of near-optimal trees, while BPAnalysis is quite slow. Furthermore, our results confirm that our new method returns results as good as any of the other methods and does so within very reasonable times, even on datasets on which BPAnalysis cannot run to completion.

Conclusions

Our initial study on real and synthetic data containing a single chromosome suggests that, for some conditions (when the rate of inversions per edge is low relative to the number of genes), many of the proposed methods for reconstructing small phylogenetic trees from gene order data *can recover* highly accurate tree topologies. Further, under model conditions with low evolutionary rates, the breakpoint phylogeny seems to be a good candidate for the true evolutionary tree. Consequently, under these conditions, methods that seek the breakpoint phylogeny offer real promise. However, the methods can be distinguished in terms of the computational effort involved, in which respect the MPBE method is a significant improvement over BPAnalysis for at least some moderate to large datasets.

Our results suggest that all of the methods we evaluated have unacceptable levels of errors on trees in which the inversion rate on the edges is high relative to the number of genes. Thus, new methods need to be developed for these types of genome evolution problems and current approaches to phylogenetic analyses based upon gene orderings should be restricted to cases with low rates of evolution. These findings apply to neighbor-joining based upon various ways of calculating genome distances, maximum-parsimony analyses of binary sequences derived from genome data, and breakpoint phylogenies. Indeed, it may be that any approach for solving the breakpoint phylogeny will perform poorly in the presence of high evolutionary rates relative to the number of genes. In such cases, approaches that explicitly seek to minimize the total number of evolutionary events may be required, but no such method currently exists.

Future Work and Recommendations

Faster methods are needed for solving the breakpoint phylogeny problem, as well as to score trees with respect to evolutionary distances (*ITT* and *I*). Since MPBE depends upon BPAnalysis in order to label internal nodes with circular genomes, and upon derange2 and signed_dist to score these trees for *ITT* and *I* distances, a first step should be to speed up BPAnalysis and signed_dist, and improve the accuracy of BPAnalysis and derange2 (since these find local optima but not necessarily global optima). More effective implementations of the basic concept in BPAnalysis, such as hill-climbing or branch-and-bound through the tree space and abandoning strict optimality in solving the TSP instances in favor of a fast and reliable heuristic (such heuristics abound in the TSP literature), could make the method run fast enough to be applicable to datasets comparable to ours.

We note that in our studies the polynomial-time method of neighbor-joining has performed as well as MPBE in terms of topological accuracy, bringing into question whether the more computationally intensive approaches deserve consideration. One clear advantage of both MPBE and BPAnalysis is that they tell us more about the space of optimal and near-optimal trees than neighbor-joining does and hence help us identify alternative hypotheses. The task remains to identify regions of the parameter space in which MPBE or BPAnalysis outperform neighbor-joining in topological accuracy. We conjecture that such regions do exist (as other studies based upon biomolecular sequence evolution show [24, 16]).

Given the rapid increase in the availability of complete genome sequences, the current limitation in reconstructing phylogenies from gene order data for datasets containing many genomes or genes is of major concern. Until improved methods are developed, we recommend that phylogenetic analyses of gene order data seek to obtain the breakpoint phylogenies and that these breakpoint phylogenies then be scored under *ITT* distances, for some appropriate weighting of the events. We also recommend that MPBE be used, until BPAnalysis can be made competitively fast.

References

[1] Berman, P., and Karpinski, M., "On some tighter inapproximability results," ECCC Report No. 29 (1998), University of Trier.

[2] Blanchette, M., derange2 at URL www.cs.washington.edu/homes/blanchem/software.html.

[3] Blanchette, M., Bourque, G., and Sankoff, D., "Breakpoint phylogenies," in *Genome Informatics 1997*, Miyano, S., and Takagi, T., eds., Universal Academy Press, Tokyo, 25–34.

[4] Blanchette, M., Kunisawa, T., and Sankoff, D., "Gene order breakpoint evidence in animal mitochondrial phylogeny," *J. Mol. Evol.* **49** (1999), 193–203.

[5] Bowman, C.M., Baker, R.F., and Dyer, T.A., "In wheat cpDNA, segments of ribosomal protein genes are dispersed repeats, probably conserved by nonreciprocal recombination," *Curr. Genet.* **14** (1988), 127–136.

[6] Caprara, A., "Formulations and hardness of multiple sorting by reversals," *Proc. 3rd Conf. Computational Molecular Biology RECOMB99*, ACM Press, New York (1999), 84–93.

[7] Cosner, M.E. "Phylogenetic and molecular evolutionary studies of chloroplast DNA variations in the Campanulaceae." Ph.D. Dissertation (1993), Ohio State U., Columbus OH.

[8] Cosner, M.E., Jansen, R.K., Palmer, J.D., and Downie, S.R., "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.* **31** (1997), 419–429.

[9] Downie, S.R., and Palmer, J.D., "Use of chloroplast DNA rearrangements in reconstructing plant phylogeny," in *Plant Molecular Systematics*, Soltis, P., Soltis, D., and Doyle, J.J., eds., Chapman & Hall, New York (1992), 14–35.

[10] Felsenstein, J., "PHYLIP—Phylogeny Inference Program," at URL evolution.genetics.washington.edu/phylip/phylip.html

[11] Fitch, W., and Margoliash, E. "Construction of phylogenetic trees." *Science* **1955** (1967), 279–284.

[12] Foulds, L.R., and Graham, R.L. "The steiner tree problem in phylogeny is NP-Complete," *Advances in Appl. Math.* **3** (1982), 43–49.

[13] Hannenhalli, S., "Software for computing inversion distances between signed gene orders," at URL www-hto.usc.edu/plain/people/Hannenhalli.html

[14] Hannenhalli, S., and Pevzner, P.A., "Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals)," *Proc. 27th Ann. ACM Symp. on Theory of Computing*, ACM Press (1995), 178–189.

[15] Hoot, S.B., and Palmer, J.D., "Structural rearrangements, including parallel inversions, within the chloroplast genome of Anemone and related genera," *J. Mol. Evol.* **38** (1994), 274–281.

[16] D. Huson, S. Nettles, K. Rice, T. Warnow, and S. Yooseph. "The Hybrid tree reconstruction method." To appear, *The Journal of Experimental Algorithms*, special issue for selected papers from The Workshop on Algorithms Engineering, Saarbrucken, Germany, 1998.

[17] Kaplan, H., Shamir, R., and Tarjan, R.E., "Faster and simpler algorithm for sorting signed permutations by reversals," *Proc. 8th ACM-SIAM Symp. on Discrete Algorithms SODA97*, ACM Press (1997), 344–351.

[18] Knox, E.B., Downie, S.R., and Palmer, J.D., "Chloroplast genome rearrangements and the evolution of giant lobelias from herbaceous ancestors," *Mol. Biol. Evol.* **10** (1993), 414–430.

[19] Milligan, B., Hampton, J., and Palmer, J.D., "Dispersed repeats and structural reorganization in subclover chloroplast DNA," *Mol. Biol. Evol.* **6** (1989), 355–368.

[20] Nadeau, J.H., and Taylor, B.A., "Lengths of chromosome segments conserved since divergence of man and mouse," *Proc. Nat'l Acad. Sci. USA* **81** (1984), 814–818.

[21] Ogihara, Y., Terachi, T., and Sasakuma, T., "Intramolecular recombination of the chloroplast genome mediated by short direct-repeat sequences in wheat species," *Proc. Nat'l Acad. Sci. USA* **85** (1988), 8573–8577.

[22] Palmer, J.D., "Plastid chromosomes: structure and evolution," in *The Molecular Biology of Plastids, Vol. 7A*, Bogorad, L., and Vasil, I.K., eds., Academic Press, New York (1991), 5–53

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

[24] K. Rice and T. Warnow, "Parsimony is Hard to Beat!," Proceedings, Third Annual International Conference of Computing and Combinatorics (COCOON), Shanghai, China, 1997, pp. 124-133. T. Jiang and D.T. Lee, Eds.

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

[26] Reinelt, G. *The Traveling Salesman: Computational Solutions for TSP Applications*. LNCS **840** (1994), Springer Verlag, Berlin.

[27] Sankoff, D., private communication, February 2000.

[28] Sankoff, D., and Blanchette, M., "Multiple genome rearrangement and breakpoint phylogeny," *J. Computational Biology* **5** (1998), 555–570.

[29] Sankoff, D., Leduc, G., Antoine, N., Paquin, B., Lang, B.F., and Cedergren, R., "Gene order comparisons for phylogenetic inference: evolution of the mitochondrial genome," *Evolution* **89** (1992), 6575–6579.

[30] Swofford, D.L. "PAUP*: Phylogenetic Analysis under Parsimony and Other Methods," version 4.0. Sinauer Associates, Sunderland, Massachusetts.

[31] Warnow, T., "Some combinatorial problems in phylogenetics", *Proc. Int'l Colloquium on Combinatorics and Graph Theory*, Balatonlelle, Hungary, 1996.

[32] www.cs.utexas.edu/users/stacia/ismb2000/.