

An Empirical Comparison of Phylogenetic Methods on Chloroplast Gene Order Data in Campanulaceae

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

April 19, 2000

Abstract

The first heuristic for reconstructing phylogenetic trees from gene order data was introduced by Blanchette *et al.*. It sought to reconstruct the *breakpoint phylogeny* and was applied to a variety of datasets. We present a new heuristic for estimating the breakpoint phylogeny which, although not polynomial-time, is much faster in practice than *BP-Analysis*. We use this heuristic to conduct a phylogenetic analysis of chloroplast genomes in the flowering plant family Campanulaceae. We also present and discuss the results of experimentation on this real dataset with three methods: our new method, *BPAnalysis*, and the neighbor-joining method, using breakpoint distances, inversion distances, and inversion plus transposition distances.

1 Introduction

Phylogenetic tree reconstruction is a major aspect of much biological research. This is a very difficult computational problem because most optimization tasks related to tree reconstruction are NP-hard and can require years to solve on real datasets. With the recent introduction of whole genomes for use in phylogenetic reconstruction, the problem has become even more complex. In this paper, we present a new approach for reconstructing trees from gene order data and we compare this approach to others using chloroplast genomes from the flowering plant family Campanulaceae.

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.

*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

The first heuristic for reconstructing phylogenetic trees from gene order data was introduced by Blanchette *et al.* in [4]. It sought to reconstruct the *breakpoint phylogeny* and was applied to a variety of datasets [5, 28].

A different technique for reconstructing phylogenies from gene order data was introduced by Cosner in [8]. In [9] we described a simple version of the method, which can also be described as a heuristic for the breakpoint phylogeny, although it is quite different in its technique from `BPAnalysis`. We call this 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.

In this paper, we report on an extensive phylogenetic analysis of a subset of the original dataset (12 of the original 18 genera) using a simple version of the MPBE technique. We also compare the performance of this method to two other methods for phylogeny reconstruction based upon gene order data: `BPAnalysis` (the heuristic designed and implemented by Blanchette *et al.* [4]) and the polynomial-time, distance-based method of neighbor-joining [25], using a variety of distance measures.

2 Gene order phylogenies

2.1 Definitions

We assume a fixed set of genes $\{g_1, g_2, \dots, g_n\}$. Each genome is then an ordering 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 $\mathcal{G} = \{g_1, g_2, \dots, g_6\}$ 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 ordering above. Furthermore the ordering g_1, g_2, \dots, g_n , whether linear or circular, is considered equivalent to that obtained by considering the complementary strand, i.e., to the ordering $-g_n, -g_{n-1}, \dots, -g_1$.

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 in [8].

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

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

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}, \dots, g_j and inserting it immediately after g_k . Thus the genome G above (with the additional assumption of $k > j$) is replaced by

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

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, \dots, p_k be a sequence of evolutionary events operating on G ; then $p_1, p_2, \dots, p_k(G)$ defines a genome G' . When each operation is assigned 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' . When the cost of each operation is finite, any two genomes have a finite edit distance. (Similarly, if the probability of each edit operation is given, we can define the edit sequence of maximum probability.)

The *inversion distance* between two genomes is the minimum number of inversions needed to transform one genome into another. The inversion distance between two genomes is computable in polynomial time for signed genomes [16, 19] and is available in software as `signed_dist`. 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 but, for the case of linear genomes, Bafna and Pevzner [1] have found a 1.5-approximation algorithm. When inversions and transpositions and/or transversions are allowed, nothing is known about the computational complexity or approximability of computing edit distances.

Heuristics have been developed to estimate these edit distances for different costs per edit operation; these are available in `derange2` [3]. We will refer to the edit distance calculated by `derange2` for the weighted sum of inversions, transpositions, and transversions as the **ITT** distance.

An *evolutionary tree* (or *phylogeny*) for a set S of genomes is a binary tree with $|S|$ leaves, each leaf labeled by a distinct element of S . A putative evolutionary tree is “correct” as long as this leaf-labeled topology is identical to the true evolutionary tree (which we do not know for real data sets). In the context of gene order data, a natural optimization problem is the Maximum Parsimony for Rearranged Genomes, which we now define:

Maximum parsimony for rearranged genomes (MPRG): 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, where the edit distance between two sequences is just the number of positions in which they differ, or the Hamming distance.)

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 the ITT distance). All these problems are NP-hard however, 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 [7].

2.2 Breakpoint Phylogeny

Recently, Blanchette *et al.* [5] proposed a new optimization problem for phylogeny reconstruction on gene order data. Rather than minimizing the number of evolutionary events (for example, inversions), they sought to minimize the number of breakpoints. We now define this precisely.

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).

It has long been known that the breakpoint distance is at most twice the inversion distance for any two genomes. For some datasets, however, there can be a close-to-linear relationship between the breakpoint distance and either the inversion distance or the ITT distance. 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.* [5] observed such a close-to-linear relationship in a group of metazoan genomes (the correlation coefficient between the two measures for their set was 0.9815) and went on to develop a heuristic for finding the breakpoint phylogeny.

2.3 BPAAnalysis

Computing the breakpoint phylogeny is NP-hard for the case of just three genomes [23], a special case known as the *Median Problem for Breakpoints (MPB)*. Blanchette *et al.* however, showed that the MPB reduces to the Travelling Salesman Problem (TSP) and designed special heuristics for the resulting instances of TSP. Their heuristic approach to solving the breakpoint phylogeny exactly solves numerous instances of the TSP. Specifically, their algorithm 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) and 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 $(2n - 5) \cdot (2n - 7) \cdot \dots \cdot 3$, 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 since 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* 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 [21], but will affect the running time proportionally.

2.4 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, which we originally described in [9]. This new method is derived from an earlier method developed by Cosner in [8]. Like Cosner’s technique, our method encodes the genome data as binary sequences, and seeks a maximum parsimony tree for these sequences. Unlike Cosner’s technique, our encoding is very simple and employs no biological assumptions. Furthermore, our technique has a closer relationship to the breakpoint phylogeny problem than Cosner’s technique. The other difference between our method and Cosner’s is that our method has a second phase, in which we select from the maximum parsimony trees we find the tree(s) that have minimum length, with respect to some evolutionary metric (e.g. the inversion distance, or the ITT distance). We now describe the two phases of the MPBE approach.

2.4.1 Phase I: Solving Maximum Parsimony on Binary Encodings of Genomes

We now show how we define the binary sequences. 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 [8], 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.)

Now 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 k ; the output is a tree T with leaves labelled by S and internal nodes labelled by additional binary sequences of length k in such a way as to minimize $\sum H(e)$ 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 [14], but fast heuristics exist that are widely available in standard phylogeny software packages, such as `PAUP` [29]. 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.

2.4.2 Phase II: Screening the Maximum-Parsimony Trees

Once the maximum-parsimony trees are obtained, the internal nodes are labelled by circular signed gene orders by giving the topology of the maximum-parsimony tree as a constraint to `BPAnalysis`, thus producing a labelling which (hopefully) minimizes the breakpoint distance of the tree. The labelling also allows us to score each tree for the

inversion distance (by scoring each edge using `signed_dist` [15]), or for the ITT distance (by scoring each edge using `derange2` [3]). The tree that minimizes the total cost is then returned.

2.4.3 Running Time of MPBE

The computational complexity of MPBE, while less than that of `BPAnalysis`, remains high. Evaluating a single tree topology in the search space takes polynomial time— more precisely, takes $\Theta(nk)$ time, where n is the number of genomes and k is the number of genes in each genome, but the search for the maximum-parsimony trees is based upon hill-climbing through the space of tree topologies. Thus finding the maximum parsimony trees is exponential in the number of genomes but only polynomial in the number of genes. Labelling the internal nodes of each maximum-parsimony tree by using constraint trees for `BPAnalysis` is expensive, but we generally only examine a small percentage of the space of trees and thus reduce the computational cost significantly by comparison to the exhaustive search strategy of `BPAnalysis`. Evaluating the cost of each tree with respect to inversion or ITT distances is quite fast.

2.5 MPBE as a heuristic for the Breakpoint Phylogeny

We now show that MPBE should be seen as a heuristic for the breakpoint phylogeny problem. Suppose T is the breakpoint phylogeny for the set G_1, G_2, \dots, 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 then replaced by the binary encoding, using the technique described earlier, the parsimony length of the tree (given these sequences at each node) 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, however, as 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 whose parsimony length is minimal for the input of binary sequence encodings of the genomes) to have internal nodes whose binary sequence encodings cannot be realized by circular orderings of signed genes. For example, if there are n genes in each of k genomes, then each binary sequence must have exactly n 1's and $n(k - 1)$ 0's, for the sequence to correspond to a circular ordering; and even then, only some such sequences will correspond to signed circular orderings. If the sequences in the internal nodes of an MPBE correspond to signed circular orderings, then the tree will be a breakpoint phylogeny (there can be more than one tree minimizing the number of breakpoints). However, if for each MPBE tree, the sequences do not correspond, then the MPBE trees and the breakpoint phylogenies may be disjoint.

To understand this point, it may be useful to rephrase the the breakpoint phylogeny problem as follows. Suppose we say that a binary sequence is a “circular genome sequence” if it is the binary encoding of a circular genome, under some pre-specified representation method. Then the breakpoint phylogeny problem is to find the tree of minimum parsimony length, where the leaves are labelled by the binary encodings of the circular genomes, and where all internal nodes are 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 select binary sequences which are not circular genome sequences as labels. When this happens, it will fail to find feasible solutions to the breakpoint phylogeny problem.

Thus, MPBE is a heuristic for the breakpoint phylogeny, but the internal nodes must be relabelled by circular genome sequences (using `BPAnalysis` or other such techniques) so that the true breakpoint distance of the trees can be computed. This is why we have included Phase II in our method.

The other issues are straightforward and have to do with computational effort. 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.

2.6 Distance-Based Methods for Reconstructing Trees

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 minimum edit distances, based upon some cost function for

each of the allowed operations (inversions, transpositions, etc.). Given the distance matrix d , the method computes an edge-weighted tree whose leaf-to-leaf distances closely fit the distance matrix. Since almost all optimization problems related to tree reconstruction are NP-hard, the most frequently used distance-based methods are polynomial-time methods such as neighbor-joining [25]; these do not explicitly seek to optimize any criterion, but can have good performance in empirical studies. In particular, neighbor-joining has had excellent performance in studies based upon simulating biomolecular sequence evolution and is probably the most popular distance-based method.

There has been little use of distance-based methods for reconstructing phylogenies from gene order data. However, in a recent publication, Blanchette *et al.* [5] evaluated two of the most popular polynomial-time distance-based methods for phylogenetic reconstruction, neighbor-joining and Fitch-Margoliash [13], 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. The trees obtained by these methods were 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 [26].

3 The Campanulaceae cpDNA Dataset

We tested MPBE on the Campanulaceae dataset, and compared it to `BPAnalysis` and to the popular distance-based method, neighbor joining. As we will show, all the methods produced very similar trees, and differed primarily in terms of their running times.

Chloroplast DNA is generally highly conserved in nucleotide sequence, gene order and content, and genome size [22]. The genomes contain approximately 120 genes which are 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 usually rare. Therefore, they are extremely useful as phylogenetic markers because they are easily polarized and exhibit very little homoplasy when properly characterized [10]. In groups in which more than one gene order change has been detected, the order of events is usually readily determined (e.g., [20, 17]). Chloroplast DNA gene order changes have been useful in phylogenetic reconstruction in many plant groups (see [10]). These changes have considerable potential to resolve phylogenetic relationships and they provide valuable insights into the mechanisms of cpDNA evolution.

In earlier work [8], Cosner obtained detailed restriction site and gene maps for 18 genera of the Campanulaceae and tobacco, and used a complicated variant of the first phase of the MPBE analysis described above to obtain a phylogenetic analysis of these genera. Her analysis suggested an incredible diversity of genome rearrangements, including inversions, insertions, deletions, gene duplications, and putative transpositions. Transpositions in particular have only rarely been hypothesized for chloroplast evolution, therefore the inference of these events for the Campanulaceae was surprising. Also interesting were the extensive contractions and expansions of the inverted repeats, and the disruption of highly conserved operons. The variety of rearrangements far exceeds that reported in any group of land plants, making it challenging to determine the exact numbers and the evolutionary sequence of rearrangement events. Several of these events are of particular interest because they have not been encountered elsewhere, or because they are common in the Campanulaceae.

The analysis technique used by Cosner differs from our technique in a few significant ways. First, she inferred additional binary characters based upon gene losses, gene duplications obtained by expansions of the inverted repeats, and large insertions; also, in some analyses her characters were weighted in accordance with the relationships between different events. Consequently, her analysis employs greater biological input than our analysis.

We analyzed the same dataset studied by Cosner. However, in order to apply the MPBE method, we had to remove two incompletely mapped genera from the dataset (these lack certain gene segments). We also had to eliminate the repeated regions in order to represent each genome by a signed circular ordering of 105 gene segments. This had the consequence of making certain pairs of genera indistinguishable, since they differed only in duplications and insertions, and in the inverted repeat. Consequently, our dataset size was reduced from the original 19 genera (18 Campanulaceae and tobacco) to 13.

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

4 Chloroplast Data Analysis

4.1 The Binary Encoding of the Data

We used gene maps to encode each of the 13 genera as a circular ordering 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 compactly 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 1 gives the compact representations of the genomes for the 13 genera.

4.2 The phylogenetic analysis of the data

We analyzed the dataset of 13 circular genomes using `BPAnalysis`, neighbor joining, and MPBE.

We used these 13 circular orderings as input to `BPAnalysis`. The program spent over 43 hours of computation time without completing. The best breakpoint score found in those 43 hours was 96.

We also computed neighbor-joining trees (using `Phylip` [11]) on three different distance matrices: the inversion distance matrix (computed using `signed_dist`, the ITT distance 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 2; the other distance matrices are available on our web page [30]. Neighbor joining finished in less than a second, with the trees shown in Figures 3 below.

In the MPBE analysis, we encoded these circular orderings with our binary encoding technique and conducted a

parsimony analysis of the resulting binary sequences. Because the dataset size was small enough, we were able to exactly solve maximum parsimony using the branch-and-bound procedure of PAUP*. We obtained four maximum parsimony trees from this dataset. (The sequences, as well as the four MP trees, are available on our web page[30]. They can also be calculated directly from the gene order data. Note that the ordering of the columns of the datamatrix does not affect the parsimony analysis.) We then inferred circular orderings of signed gene segments for each internal node by giving each of the four binary maximum parsimony trees 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 (using the labels assigned by BPAnalysis) for the ITT distance (using weights of 2.1 for transpositions and transversions and 1 for inversions) and the breakpoint distance. Each of the four trees we obtained had only 89 breakpoints, but they differed significantly in terms of their ITT distances.

The first tree has a total of 40 inversions and 12 transpositions and transversions; the second has 48 inversions and 18 transpositions and transversions; the third has 40 inversions and 12 transpositions and transversions; and the fourth has 67 inversions and 32 transpositions. See our web page for figures with edge weights labelled [30]. 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: 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 [8] on the original 19 genera (18 Campanulaceae and the outgroup tobacco) when restricted to the smaller subset. Thus, the full character set used by Cosner (which represent insertions, deletions, duplications, contractions/expansions of the inverted repeat, etc.) is compatible with this analysis, but gives additional resolution.

4.3 A third phase

Finally, we took the four MPBE trees and three neighbor joining trees, and computed the strict consensus. We gave this to tree (which was partially unresolved, see Figure 4) as a constraint tree to BPAnalysis. This makes BPAnalysis search the set of trees that refine this constraint tree for the tree(s) with smallest breakpoint lengths. This search returned six trees with 89 breakpoints, four of which were the MPBE trees, and two new trees. We scored those trees under maximum parsimony (using binary sequence encodings of the leaves) and observed that these trees differed in parsimony length from our optimal trees by at most three steps. This suggests that MPBE should be modified to examine not only the optimal parsimony trees obtained in Phase I, but also the near-optimal trees.

4.4 Comparison of different methods on this dataset

Since MPBE and BPAnalysis both seek the breakpoint phylogeny, we will compare them with respect to the breakpoint scores obtained. We will also compare the trees obtained by neighbor joining and MPBE topologically, to see how different they are.

The best trees we found using MPBE had breakpoint scores of 89, but the best trees found by BPAnalysis after 43 hours had 96 breakpoints. This difference in score is quite significant. In order to compare the methods on this dataset, we need to quantify the differences in tree reconstructions; we do this using standard techniques in phylogenetics, as follows.

The False Positive and False Negative Rate 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 $C(T) = \{\pi_e : e \in E(T)\}$ uniquely defines the tree T ; this characterization is called the *character encoding* of T . Given a collection of trees T_1, T_2, \dots, T_k , each leaf-labelled by S , we define the *strict consensus* of the trees to be that unique tree T_{sc} defined by $C(T_{sc}) = \cap_i C(T_i)$. 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

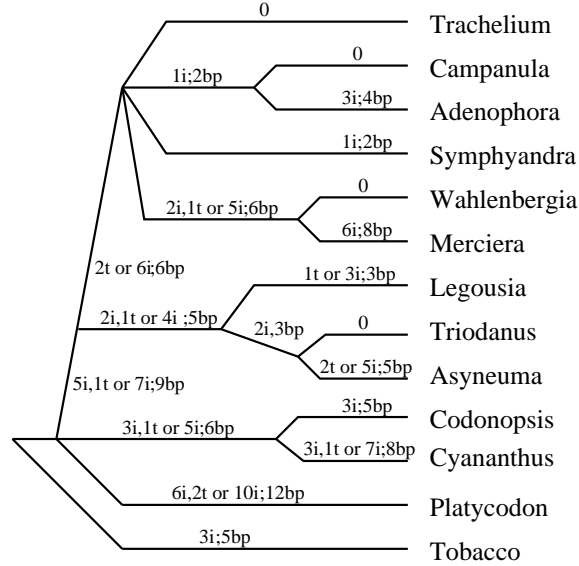


Figure 2: The reconstructed phylogeny of 12 genera of Campanulaceae and the outgroup tobacco based upon an MPBE analysis of 185 binary characters. The number of inversions and transpositions is given above each edge followed by the number of inversions in an inversion-only scenario; the number of breakpoints is given last.

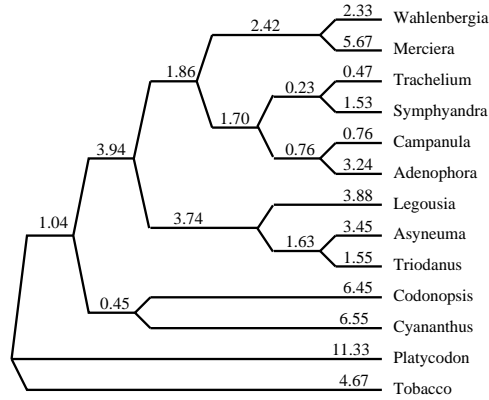
respect to T are those edges e that obey $\pi_e \in C(T') - C(T)$, i.e., edges in the inferred tree that do not exist in the true tree and should not have been inferred. Note that every trivial bipartition (induced by the edge incident to a leaf) exists in every tree. Consequently, false positives and false negatives are calculated only with respect to the internal edges of the tree and expressed as a percentage of the number of internal edges.

All of the trees we reconstructed (whether by neighbor joining or maximum parsimony) are fairly similar, differing pairwise in at most two edges. Table 1 shows the false negative rate for the 7 trees: 3 neighbor-joining trees and 4 maximum parsimony trees. The similarity between all the trees reconstructed indicates a high level of confidence in the accuracy of the common features of the phylogenetic reconstructions. This is illustrated by the strict consensus tree in Figure 4 which displays the common features of the 7 trees.

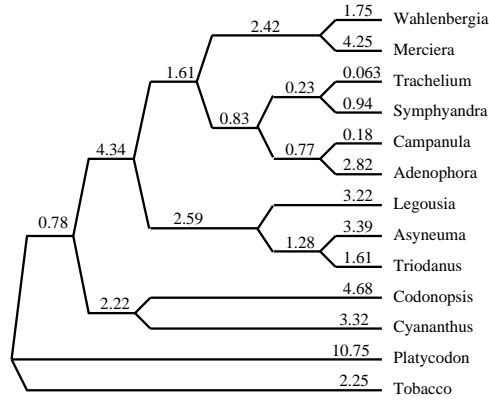
The similarity of the trees reconstructed by MPBE and neighbor joining on this dataset is perhaps to be expected, according to our earlier study [9] where we explored the performance of the MPBE and neighbor joining method on synthetic datasets, under a wide range of model conditions. We showed that when the rate of evolution on each edge is low relative to the total number of genes, then the two methods obtain very good estimates of the true tree topology (with error rates that are acceptable), but when the rate of evolution increases, both methods decrease in accuracy. Although we did not calculate any specific cutoff point at which the rate of evolution is too high for these methods to

Table 1: The false negative rate matrix of trees from various reconstruction methods on the Campanulaceae data in Figure 1. MPBE1 through MPBE4 are the four most parsimonious trees by the MPBE method. NJ(BP), NJ(INV), and NJ(ITT) are the neighbor joining trees from the distance matrix using MPBE estimation, `derange2` with inversion only, and `derange2` with cost ratio *inversion : transposition* = 1 : 2.1.

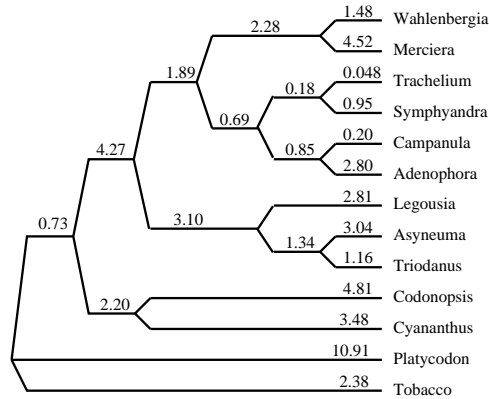
	NJ(BP)	NJ(INV)	NJ(ITT)	MPBE1	MPBE2	MPBE3	MPBE4
NJ(BP)	0	0	0	1	2	1	2
NJ(INV)	0	0	0	1	2	1	2
NJ(ITT)	0	0	0	1	2	1	2
MPBE1	1	1	1	0	1	1	2
MPBE2	2	2	2	1	0	2	1
MPBE3	1	1	1	1	2	0	1
MPBE4	2	2	2	2	1	1	0



(a) *NJ(BP)*, the neighbor joining tree on the breakpoint distance matrix.



(b) *NJ(INV)*, the neighbor joining tree on the distance matrix from signed_dist, inversion only.



(c) *NJ(ITT)*, the neighbor joining tree on the distance matrix from derange2, cost ratio inversion : trans-position = 1 : 2.1.

Figure 3: The reconstructed phylogeny of 12 genera of Campanulaceae and the outgroup tobacco using neighbor joining on various distance estimation methods.

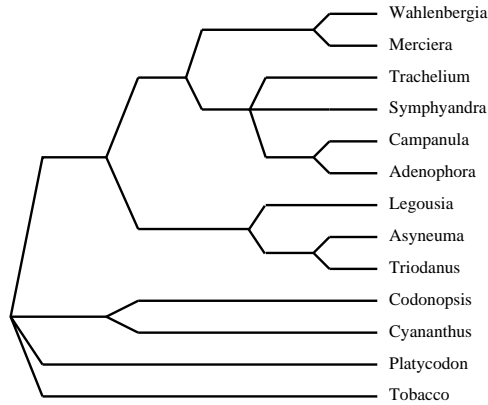


Figure 4: The strict consensus tree of the four MPBE trees and three neighbor joining trees from the 12 genera dataset of Campanulaceae and the outgroup tobacco. The tree has 8 internal edges, out of at most 10 possible internal edges for a 13-taxon phylogeny.

be accurate, we observed that accuracy in these methods was high when the breakpoint distances and ITT distances are close to linearly correlated. We compared the breakpoint distances and the evolutionary distances for our dataset (using relative costs of 1, 2.1, and 2.1); see Figure 5, and calculated the correlation coefficient, which is .98. These distances are very close to linearly related, thus predicting that the 13 genera dataset should be accurately analyzed by both neighbor joining and MPBE.

4.5 Running time

We timed each method on the chloroplast dataset. The neighbor joining trees were computed in less than a second for each dataset. Indeed, neighbor joining is a polynomial time method ($O(n^3)$).

MPBE was also reasonably fast (though not as fast as neighbor joining). Finding the four maximum-parsimony trees with PAUP took 0.15 seconds on a Macintosh G4. Labelling the internal nodes (by a call to `BPAAnalysis` giving each binary tree as a constraint) took 0.38 seconds for each tree. Computing the length of each tree, given the sequences labelling the nodes, involves calls to `signed_dist` (for inversion distances) and `derange2` (for ITT distances). These took 45.65 seconds per edge 0.01 seconds per edge, respectively. In all, MPBE took less than a minute to complete.

Table 2: The distance matrix computed with `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

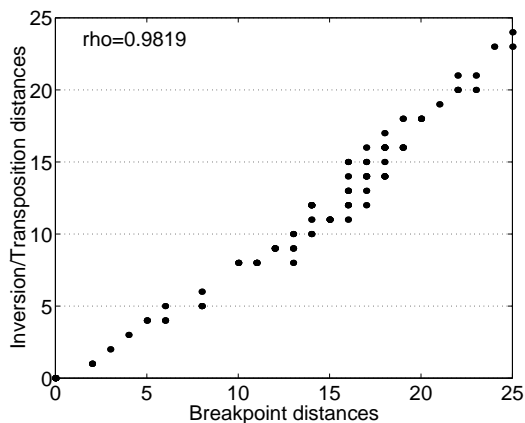


Figure 5: Comparison of distance calculations on the Campanulaceae Chloroplast dataset with a correlation coefficient of $\rho = 0.9819$.

The running time of `BPAnalysis` is harder to evaluate, because it did not complete its search during the four days of our experiment. However, we were able to approximate the amount of time it would take to complete, on the basis of how long `BPAnalysis` took to evaluate a single tree. Our calculation suggest that `BPAnalysis` evaluated approximately 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.

MPBE is less computationally intensive than `BPAnalysis`, because it is exponential only in the number of genomes, while `BPAnalysis` is exponential also in the number of gene segments. To see this, consider the cost of labelling the internal nodes of a tree under maximum parsimony, and then compare that to the cost of computing the labelling that minimizes the number of breakpoints. The first problem (maximum parsimony on a fixed tree) is polynomial time, using a very simple dynamic programming algorithm [12]; the second problem (breakpoint phylogeny on a fixed tree) is NP-hard [23]. Furthermore, the heuristic used by `BPAnalysis` to label the internal nodes of a fixed tree uses many calls to heuristics for the NP-hard Travelling Salesman Problem. This may explain the difference in running time, at least in part, on our dataset.

However, we note that 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`.

Table 3: The distance matrix computed with `signed_dist`, inversion only.

	Tra	Cam	Ade	Sym	Leg	Asy	Tri	Wah	Mer	Cod	Cya	Pla	Tob
Tra	0	1	4	1	8	10	8	5	8	15	14	19	10
Cam	1	0	3	2	9	11	9	6	9	15	15	20	11
Ade	4	3	0	5	12	14	12	8	11	16	15	20	13
Sym	1	2	5	0	9	11	9	6	9	14	13	20	11
Leg	8	9	12	9	0	9	5	12	14	18	16	23	14
Asy	10	11	14	11	9	0	5	12	16	18	16	21	12
Tri	8	9	12	9	5	5	0	12	14	18	15	21	12
Wah	5	6	8	6	12	12	12	0	6	18	16	23	14
Mer	8	9	11	9	14	16	14	6	0	17	16	24	16
Cod	15	15	16	14	18	18	18	18	17	0	8	18	10
Cya	14	15	15	13	16	16	15	16	16	8	0	16	10
Pla	19	20	20	20	23	21	21	23	24	18	16	0	13
Tob	10	11	13	11	14	12	12	14	16	10	10	13	0

Computational issues arise in other aspects of gene order data analysis as well. Even calculating the distance matrix between every pair of signed circular genomes in a large data set is computationally challenging: `derange2` is fast, but inexact (because it heuristically computes the distance between two genomes by using inversions, transpositions, and inverted transpositions (transversions) using a greedy strategy, it only allows an operation if that operation decreases the breakpoint distance between the two genomes). Consequently, it can miss minimal edit sequences, as we observed in our tests. Hannenhalli’s software `signed.dist` for pairwise distances runs in slow polynomial-time ($\Theta(k^5)$ to compute distances between a pair of genomes on k genes); in order to compute all pairwise distances, it requires $\Theta(n^2 k^5)$ time. For our dataset, $k = 105$ and $n = 13$.

5 Conclusions

There are three basic observations we can make as a result of this study. First, the breakpoint phylogeny seems to be an excellent approach for reconstructing phylogenetic trees from gene order data, at least when the rates of evolution are sufficiently low. Second, `BPAnalysis` is too slow to analyze some datasets, and in particular was unable to analyze our chloroplast dataset in a reasonable amount of time. Third, our new MPBE method shows promise as a heuristic to solve the breakpoint phylogeny problem, and does so significantly faster than `BPAnalysis`.

Our future work will explore faster methods for solving the breakpoint phylogeny problem. Since MPBE depends upon `BPAnalysis` in order to label internal nodes with circular genomes, and upon `derange2` to score these trees for ITT distances, one direction our work will take is to speed up and improve the accuracy of both `BPAnalysis` and `derange2`. 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.

As a final note, we point out that in our studies neighbor joining has performed as well as MPBE in terms of topological accuracy, and is polynomial time. Therefore, an important question remains as to whether these more computationally intensive approaches are merited. However, one advantage of both MPBE and `BPAnalysis` is that these methods tell us more about the space of optimal and near-optimal trees than neighbor joining does, and hence also identify alternative hypotheses. Still, until we can identify regions of the parameter space in which MPBE or `BPAnalysis` outperform neighbor joining in topological accuracy, there is also an argument to be made in favor of the fast neighbor joining method. Indeed, we conjecture that such regions do exist (as other studies based upon biomolecular sequence evolution show [24, 18]), and our research will also seek to determine whether this is true for gene order data as well.

6 Recommendations

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 and/or more 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, but that these breakpoint phylogenies be scored under ITT distances. In order to find these breakpoint phylogenies, we recommend that MPBE be used. An alternative approach suggested by our studies is to compute the strict consensus of the trees obtained by various methods (for example, MPBE trees and neighbor joining), and then give this to `BPAnalysis` as a constraint tree. Although this technique did not produce better trees on our dataset, it may so on other datasets, and is worth exploring.

References

- [1] Bafna, V., and Pevzner, P.A., “Sorting by transpositions,” *SIAM J. Discrete Math.* **11**, 2 (1998), 224–240.
- [2] Berman, P., and Karpinski, M. “On some tighter inapproximability results.” ECCO Report No. 29 (1998), University of Trier.
- [3] Blanchette, M., `derange2`, software available under [www.cs.washington.edu](http://www.cs.washington.edu/~blanchem/software.html) at address `homes/blanchem/software.html`.

- [4] 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.
- [5] Blanchette, M., Kunisawa, T., and Sankoff, D., "Gene order breakpoint evidence in animal mitochondrial phylogeny," *J. Mol. Evol.* **49** (1999), 193–203.
- [6] Caprara, A., "Sorting by reversals is difficult," *Proc. 1st Conf. Computational Molecular Biology RECOMB97*, ACM Press, New York (1997), 75–83.
- [7] Caprara, A., "Formulations and hardness of multiple sorting by reversals," *Proc. 3rd Conf. Computational Molecular Biology RECOMB99*, ACM Press, New York (1999), 84–93.
- [8] Cosner, M.E. "Phylogenetic and molecular evolutionary studies of chloroplast DNA variations in the Campanulaceae." Doctoral Dissertation (1993), Ohio State University, Columbus OH.
- [9] Cosner, M.E., Jansen, R.K., Moret, B.M.E., Raubeson, L.A., Wang, L.-S., Warnow, T., and Wyman, S., "A New Fast Heuristic for Computing the Breakpoint Phylogeny and Experimental Phylogenetic Analyses of Real and Synthetic Data," to appear *Proceedings of the 8th Intl. Conf. on Intel. Sys. for Mol. Bio. ISMB'00*.
- [10] 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 and Hall, New York (1992), 14–35.
- [11] Felsenstein, J., "PHYLIP—Phylogeny Inference Program," a software suite available on-line from evolution.genetics.washington.edu at address phylip/phylip.html
- [12] Fitch, Wm. "Toward defining the course of evolution: minimum change for a specified tree topology," *Syst. Zool.*, **1971**, 20:406-416.
- [13] Fitch, W., and Margoliash, E. "Construction of phylogenetic trees." *Science* **1955** (1967), 279–284.
- [14] Foulds, L.R., and Graham, R.L. "The steiner tree problem in phylogeny is NP-Complete," *Advances in Appl. Math.* **3** (1982), 43–49.
- [15] Hannenhalli, S., "Software for computing inversion distances between signed gene orders," available at www-hto.usc.edu/plain/people/Hannenhalli.html
- [16] 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.
- [17] 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.
- [18] 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, Saarbrücken, Germany, 1998.
- [19] Kaplan, H., Shamir, R., and Tarjan, R.E., "Faster and simpler algorithm for sorting signed permutations by reversals," *Proc. 8th Ann. ACM-SIAM Symp. on Discrete Algorithms SODA97*, ACM Press (1997), 344–351.
- [20] 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.
- [21] Maddison, D.R., "The Discover and Importance of Multiple Islands of Most-parsimonious Trees," *Systematic Zoology* **40** (1991), 315–328.
- [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] Sankoff, D., private communication, February 2000.
- [27] Sankoff, D., and Blanchette, M., "Multiple genome rearrangement and breakpoint phylogeny," *J. Computational Biology* **5** (1998), 555–570.
- [28] 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.

- [29] Swofford, D.L. "PAUP*: Phylogenetic Analysis under Parsimony and Other Methods," version 4.0. Sinauer Associates, Sunderland, Massachusetts.
- [30] <http://www.cs.utexas.edu/users/stacia/dcaf>.