Constructing majority-rule supertrees

Jianrong Dong¹* , David Fernández-Baca¹* , FR McMorris²

Abstract

**Background:** Supertree methods combine the phylogenetic information from multiple partially-overlapping trees into a larger phylogenetic tree called a supertree. Several supertree construction methods have been proposed to date, but most of these are not designed with any specific properties in mind. Recently, Cotton and Wilkinson proposed extensions of the majority-rule consensus tree method to the supertree setting that inherit many of the appealing properties of the former.

**Results:** We study a variant of one of Cotton and Wilkinson’s methods, called majority-rule (+) supertrees. After proving that a key underlying problem for constructing majority-rule (+) supertrees is NP-hard, we develop a polynomial-size exact integer linear programming formulation of the problem. We then present a data reduction heuristic that identifies smaller subproblems that can be solved independently. While this technique is not guaranteed to produce optimal solutions, it can achieve substantial problem-size reduction. Finally, we report on a computational study of our approach on various real data sets, including the 121-taxon, 7-tree Seabirds data set of Kennedy and Page.

**Conclusions:** The results indicate that our exact method is computationally feasible for moderately large inputs. For larger inputs, our data reduction heuristic makes it feasible to tackle problems that are well beyond the range of the basic integer programming approach. Comparisons between the results obtained by our heuristic and exact solutions indicate that the heuristic produces good answers. Our results also suggest that the majority-rule (+) approach, in both its basic form and with data reduction, yields biologically meaningful phylogenies.

Background

**Introduction**

A supertree method begins with a collection of phylogenetic trees with possibly different leaf (taxon) sets, and assembles them into a larger phylogenetic tree, a supertree, whose taxon set is the union of the taxon sets of the input trees. Interest in supertrees was sparked by Gordon’s paper [1]. Since then, particularly during the past decade, there has been a flurry of activity with many supertree methods proposed and studied from the algorithmic, theoretical, and biological points of view. The appeal of supertree synthesis is that it can combine disparate data to provide a high-level perspective that is harder to attain from individual trees. A recent example of the use of this approach is the species-level phylogeny of nearly all extant Mammalia constructed by Bininda-Emonds [2] from over 2,500 partial estimates. Several of the known supertree methods are reviewed in the book edited by Bininda-Emonds [3] — more recent papers with good bibliographies include [4,5]. There is still much debate about what specific properties should (can), or should not (cannot), be satisfied by supertree methods. Indeed, it is often a challenging problem to rigorously determine the properties of a supertree method that gives seemingly good results on data, but is heuristic.

The well-studied consensus tree problem can be viewed as the special case of the supertree problem where the input trees have identical leaf sets. Consensus methods in systematics date back to [6]; since then, many consensus methods have been designed. A good survey of these methods, their properties, and their interrelationships is given by Bryant [7], while the axiomatic approach and the motivation from the social sciences is found in Day and McMorris’ book [8]. One of the most widely used methods is the majority-rule consensus tree [9,10], which is the tree that contains the splits displayed by the majority of the input trees. Not only does this tree always exist, but it is also unique,
can be efficiently constructed [11], and has the property of being a median tree relative to the symmetric-difference distance (also known as the Robinson-Foulds distance [12,13]). That is, the majority-rule consensus tree is a tree whose total Robinson-Foulds distance to the input trees is minimum.

The appealing qualities of the majority-rule consensus method have made it attractive to try to extend the method to the supertree setting, while retaining as many of its good characteristics as possible. Cotton and Wilkinson [14] were able to define two such extensions (despite some doubts about whether such an extension was possible [15]) and at least two additional ones have been studied since [16]. Here we study one of the latter variants, called graft-refine majority-rule (+) supertrees in [16], and here simply referred to as majority-rule (+) supertrees. These supertrees satisfy certain desirable properties with respect to what information from the input trees, in the form of splits, is displayed by them (see the Preliminaries). The key idea in this method is to expand the input trees by grafting leaves onto them to produce trees over the same leaf set. The expansion is done so as to minimize the distance from the expanded trees to their median relative to the Robinson-Foulds distance. The supertree returned is the strict consensus of the median trees with minimum distance to the expanded input trees; these median trees are called optimal candidate supertrees.

After showing that computing an optimal candidate supertree is NP-hard, we develop a characterization of these supertrees that allows us to formulate the problem as a polynomial-size integer linear program (ILP). We then describe an implementation that enables us to solve moderately large problems exactly. We show that, as a polynomial-size integer linear program (ILP). We then describe an implementation that enables us to solve moderately large problems exactly. We show that, in particular, the need to handle all possible expansions of the input trees necessitates the introduction of new techniques.

## Preliminaries

### Basic definitions and notation

Our terminology largely follows [28]. A phylogenetic tree is an unrooted leaf-labeled tree where every internal node has degree at least three. We will use “tree” and “phylogenetic tree” interchangeably. The leaf set of a tree $T$ is denoted by $L(T)$.

A profile is a tuple of trees $P = (t_1, ..., t_k)$. Each $t_i$ in $P$ is called an input tree. Let $L(P) = \bigcup_{t \in P} L(t)$, where $K$ denotes the set $\{1, ..., k\}$. An input tree $t_i$ is plenary if $L(t_i) = L(P)$. Tree $T$ is a supertree for profile $P$ if $L(T) = L(P)$.

A split is a bipartition of a set. We write $A|B$ to denote the split whose parts are $A$ and $B$. The order here does not matter, so $A|B$ is the same as $B|A$. Split $A|B$ is nontrivial if each of $A$ and $B$ has at least two elements; otherwise it is trivial. Split $A|B$ extends another split $C|D$ if $A \supseteq C$ and $B \supseteq D$, or $A \supseteq D$ and $B \supseteq C$.

Phylogenetic tree $T$ displays split $A|B$ if there is an edge in $T$ whose removal gives trees $T_1$ and $T_2$ such that $A \subseteq L(T_1)$ and $B \subseteq L(T_2)$. A split $A|B$ is full with respect to a tree $T$ if $A \cup B = L(T)$; $A|B$ is partial with respect to $T$ if $A \cup B \subset L(T)$. Split $A|B$ is plenary with respect to a profile $P$ if $A \cup B = L(P)$. The set of all nontrivial full splits...
displayed by \( T \) is denoted \( \text{Spl}(T) \). It is well known that the full splits of \( T \) uniquely identify \( T \) ([28], p. 44). Let \( S \subseteq \mathcal{L}(T) \). The restriction of \( T \) to \( S \), denoted \( T|S \), is the phylogenetic tree with leaf set \( S \) such that

\[
\text{Spl}(T|S) = \{ A \cap S \mid B \cap S : A \mid B \in \text{Spl}(T) \text{ and } |A \cap S|,|B \cap S| > 1 \}.
\]

Let \( T' \) be a phylogenetic tree such that \( S = \mathcal{L}(T) \subseteq \mathcal{L}(T) \). Then, \( T \) displays \( T' \) if \( \text{Spl}(T') \subseteq \text{Spl}(T|S) \).

A set of splits is compatible if there is a tree \( T \) that displays them all. Tree \( T \) is compatible with a set of splits \( \chi' \) if there is a tree \( T' \) that displays \( T \) and \( \chi' \).

Let \( T_1 \) and \( T_2 \) be two phylogenetic trees over the same leaf set. The symmetric-difference distance, also known as Robinson-Foulds distance ([13]), between \( T_1 \) and \( T_2 \), denoted \( d(T_1, T_2) \), is defined as

\[
d(T_1, T_2) = |\text{Spl}(T_1) \setminus \text{Spl}(T_2)| \cup (|\text{Spl}(T_2) \setminus \text{Spl}(T_1)|)\). (1)
\]

The majority splits in a profile \( P = \langle t_1, t_2, \ldots, t_k \rangle \) are the splits displayed by more than \( \frac{k}{2} \) of the input trees. A majority plenary split is a plenary split that is also a majority split. Similarly, a majority partial split is a partial split that is also a majority split.

Rooted phylogenetic trees can be viewed as a special case of unrooted trees. That is, we can view a profile of rooted trees as unrooted trees, all of which have a common taxon called the root. Thus, in a split in a rooted tree, one of the two parts must contain the root; the part that does not contain the root is called a cluster (or clade, or monophyletic group). All of the above concepts (eg., compatibility and distance), as well as those introduced in the rest of this paper, directly apply to rooted trees. However, we shall not elaborate on this here.

To close this section, we examine the consensus problem, the special case of the supertree problem where the profile \( P = \langle T_1, T_2, \ldots, T_k \rangle \) consists of trees that have the same leaf set. The strict consensus of \( P \) is the tree that displays exactly the plenary splits present in every tree in the profile. The majority-rule consensus tree of \( P \), denoted \( \text{Maj}(P) \), is the tree that displays all the majority plenary splits in \( P \) [10]. For any phylogeny \( T \) with \( \mathcal{L}(T) = \mathcal{L}(P) \), define the distance from \( T \) to \( P \) as \( \text{dist}(T, P) = \sum_{i \in S} d(T_i, T) \), where \( d \) denotes the symmetric-difference distance. Any tree with leaf set \( \mathcal{L}(P) \) that minimizes \( \text{dist}(T, P) \) is called a median tree for \( P \). It is known that \( \text{Maj}(P) \) is a median tree for \( P \); indeed, it follows from [9] that \( \text{Maj}(P) \) is the strict consensus of the median trees for \( P \).

The (median) score of \( P \) is defined as \( s(P) = \min_{T: \mathcal{L}(T) = \mathcal{L}(P)} \text{dist}(T, P) \). Thus, \( s(P) = \text{dist}(\text{Maj}(P), P) \).

**Majority-rule (+) supertrees**

Here we describe a variant (suggested by Bill Day) of one of Cotton and Wilkinson’s ([14]) extensions of majority-rule consensus to the supertree setting.

The span of an input tree \( t \), denoted by \( \langle t \rangle \), is the set of all trees on \( \mathcal{L}(P) \) that display \( t \). The span of a profile \( P = \langle t_1, \ldots, t_k \rangle \), denoted \( \langle P \rangle \), is the set of all \( k \)-tuples \( R = \langle T_1, \ldots, T_k \rangle \) where \( T_\epsilon \in \langle t_\epsilon \rangle \) for each \( i \in K \). Each \( R \in \langle P \rangle \) is called a representative selection for \( P \) and \( \text{Maj}(R) \) is called a candidate supertree.

An optimal representative selection is a representative selection \( R \) with minimum score \( s(R) \) over all \( R \in \langle P \rangle \). We refer to \( \text{Maj}(R) \) as the optimal candidate supertree associated with \( R \). The majority-rule (+) supertree of profile \( P \), denoted by \( \text{Maj}+(P) \), is the strict consensus of all the optimal candidate supertrees. We have shown elsewhere ([16]) that \( \text{Maj}+(P) \) satisfies the following appealing properties (originally conjectured by Cotton and Wilkinson).

(CW1) \( \text{Maj}+(P) \) displays all of the majority plenary splits in \( P \).

(CW2) \( \text{Maj}+(P) \) is compatible with each majority partial split in \( P \).

(CW3) Each split in \( \text{Maj}+(P) \) is compatible with a majority of the trees in \( P \).

(CW4) Every plenary split in \( \text{Maj}+(P) \) extends at least one input tree full split.

We should note that majority-rule (+) supertrees, as defined above, do not generalize majority-rule consensus. That is, when used in the consensus setting, \( \text{Maj}+(P) \) is not, in general, the same as \( \text{Maj}(P) \). Nevertheless, majority-rule (+) consensus trees have a simple characterization that yields an efficient algorithm for computing them (see Theorem 1 of the Methods).

The majority-rule (+) supertrees we study differ from other variants in the way the span of an input tree is defined. Cotton and Wilkinson originally defined the span of a tree \( t \) as the set of all plenary binary trees that display \( t \) ([14]). This version does not generalize majority-rule consensus and does not satisfy (CW4) ([16]). In a more recent version, suggested by Wilkinson (personal communication), the span of \( t \) is the set of all plenary trees \( T \) such that \( T|\mathcal{L}(t) = t \). This definition of span prohibits refinement of any original polytomies (nodes of degree at least four) in \( t \). It can be shown that the supertree method that results from using this definition generalizes majority-rule consensus, and that it satisfies properties (CW1)-(CW4) ([16]). Nonetheless, we have preferred Day’s variant for two reasons. First, we have found it computationally easier to deal with than the others. More importantly, it can be argued that a strict generalization of majority-rule consensus might not be the ideal approach for supertree construction: In practice, one often encounters profiles where different trees “specialize” in different groups of taxa, leaving other groups largely unresolved or unrepresented. In a combined analysis, each input tree should contribute its own specialized information so that, jointly, the trees lead to
a well-resolved supertree. A strict generalization of majority rule would disallow this, since the method discards minority information. In contrast, the majority-rule (+) supertrees presented here preserve this fine-grained information, unless it were substantially contradicted by the remaining trees (the sense in which this is true can be gleaned from Theorem 1 of the Methods).

Methods

Constructing optimal candidate supertrees

We first consider the consensus version of the problem. Let $P = (T_1, ..., T_k)$ be a profile of trees over the same leaf set. Given a plenary split $X = A|B$, define

$$K_X(P) = \{i \in K : X \text{ is displayed by } T_i\}$$

and

$$K_{\bar{X}}(P) = \{i \in K : X \text{ is not compatible with } T_i\}.$$

The theorem below, proved elsewhere (Dong, Fernández-Baca, McMorris, and Powers: Majority-rule (+) consensus trees, unpublished), characterizes the majority-rule (+) consensus tree of a profile and implies that this tree can be computed in polynomial time.

**Theorem 1.** For any profile $P$, $\text{Maj}^+(P)$ is precisely the tree that displays every split $X$ such that $\vert K_X(P) \vert > \vert K_{\bar{X}}(P) \vert$. Furthermore, $\text{Maj}^+(P)$ is an optimal candidate supertree for $P$, as well as the strict consensus of all optimal candidate trees for $P$.

On the other hand, the next result suggests that finding the majority-rule (+) supertree for a profile of trees with partially overlapping leaf sets may be hard.

**Theorem 2.** There is no polynomial-time algorithm to construct an optimal candidate supertree unless $P = \text{NP}$.

**Proof.** We show that if there is a polynomial time algorithm to compute an optimal candidate supertree, then there exists a polynomial-time algorithm for the quartet compatibility problem, which is known to be NP-complete [29]. The quartet compatibility problem asks whether, given a collection $Q$ of trees on four leaves, there exists a single tree that displays them all. If the answer is “yes”, we say that $Q$ is compatible. Let $Q$ be an instance of quartet compatibility. Construct a profile $P$ that simply consists of the trees in $Q$ in some arbitrary order. We claim that $Q$ is compatible if and only if $P$ has an optimal candidate supertree with a score of zero. Suppose first that $Q$ is compatible and that $T$ is any tree that displays each element of $Q$. Then, for each tree $t$ in $P$, $T \in \langle t \rangle$, because all the splits in $T$ must be compatible with $t$, so any split in $T$ that is not in $t$ can be added to $t$. Hence, $T$ is a candidate tree for $P$ with a score of zero, and thus $T$ is also an optimal candidate supertree. Conversely, if $P$ has an optimal candidate supertree with zero score, it can be seen that $T$ displays all the quartets in $Q$; i.e., $Q$ is compatible.

In the next sections, we show that despite the above result, moderately large majority-rule (+) supertree problems can be solved using integer linear programming. For this, we need to address a potential complication: Since the definition of $\langle t \rangle$ allows refinement of multifurcations in $t$, a tree $T \in \langle t \rangle$ can contain many more non-trivial splits than $t$; indeed, we cannot predetermine the number of nontrivial splits $T$ will contain. We circumvent this potential problem by defining a restricted version of the span.

Given an input tree $t$ in a profile $P$, the restricted span of $t$, denoted $\langle t \rangle_r$, is the set of all plenary trees $T$ such that every nontrivial split in $T$ extends a distinct nontrivial split in $t$. Thus, $\vert \text{Spl}(T) \vert = \vert \text{Spl}(t) \vert$. Note that $T$ is obtained from $t$ by filling in each of $t$’s splits, by adding zero or more taxa to each part, to make them plenary splits in such a way that the resulting splits are compatible. Note also that $\langle t \rangle_r \subseteq \langle t \rangle$. The restricted span of a profile $P = (t_1, ..., t_k)$, denoted $\langle P \rangle_r$, is the set of all $R = (T_1, ..., T_k)$ for $P$ such that $T_i \in \langle t_i \rangle_r$ for each $i \in K$. Each $R \in \langle P \rangle_r$ is called a restricted representative selection for $P$. Since $\langle P \rangle_r \subseteq \langle P \rangle$, the restricted span represents an intermediate level between the input profile and the original definition of span. The restricted span is more manageable than the original span because it does not allow any refinement of input trees. In the rest of this section, we will show how to obtain majority-rule (+) supertrees directly from the restricted span.

Before presenting the first of the two main results of this section, we need to introduce some new concepts. An optimal candidate supertree $T$ for a profile $P$ is minimal if contracting any edge in $T$ yields a tree that is not an optimal candidate supertree. Let $R = (T_1, ..., T_k)$ and $R' = (T'_1, ..., T'_k)$ be two representative selections for a profile $P$. We say that $R$ displays $R'$ if $T_i$ displays $T'_i$ for every $i \in K$. Theorem 1 motivates the next definition.

The completion of a representative selection $R = (T_1, ..., T_k)$ for a profile $P$ is the representative selection $\hat{R} = (\hat{T}_1, ..., \hat{T}_k)$ obtained as follows: For every $i \in K$, $\hat{T}_i$ is the tree constructed by inserting into $T_i$ each plenary split $X = A|B$ compatible with $T_i$ such that $\vert K_X(R) \vert > \vert K_{\bar{X}}(R) \vert$.

**Theorem 3.** Let $T$ be a minimal optimal candidate supertree for a profile $P$ and let $R \in \langle P \rangle_r$ be such that $T = \text{Maj}(R)$. Consider any $G \in \langle P \rangle$, such that $G$ is displayed by $R$. Then, $R$ is the completion of $G$ and $T = \text{Maj}^+(G)$.

**Proof.** We begin by proving that $T$ is an optimal candidate supertree for $G$. Assume the contrary. Then, there exists another candidate tree $T'$ for $G$ such that (i) $T' = \text{Maj}(R')$ for some $R' \in \langle G \rangle$ and (ii) $s(R') < s(R)$. But then, since $\langle G \rangle \subseteq \langle P \rangle_r$, we have $R' \in \langle P \rangle$, and thus (ii) contradicts the optimality of $T$ for $P$. 

http://www.almob.org/content/5/1/2
Next, we argue that $T$ is a minimal optimal candidate supertree for profile $G$. Suppose this is not true. Then, $T$ displays an optimal candidate supertree $T'$ for $G$ such that $T \neq T'$. Consider any $R' \in \langle G \rangle$ such that $T' = \text{Maj}(R)$. Since $T$ and $T'$ are both optimal for $G$, $s(R) = s(R')$. Since $R'$ displays $P$, we have $R' \in \langle P \rangle$. Hence, $T'$ is also an optimal candidate supertree for $P$. This, however, contradicts the assumption that $T$ is a minimal optimal candidate tree for $P$.

By Theorem 1, $\text{Maj}^+(G)$ is an optimal candidate supertree for $G$, as well as the strict consensus of all optimal candidate supertrees for $G$. Therefore, $\text{Maj}^+(G)$ is the only minimal optimal candidate supertree for $G$. Hence $T = \text{Maj}^+(G)$.

Suppose $R = (T_1, ..., T_k)$ and let $\hat{R} = (\hat{T}_1, ..., \hat{T}_k)$ be the completion of $G$. We claim that $\hat{R} = R$.

Assume, on the contrary, that there is some $i \in K$ such that $\hat{T}_i \neq T_i$.

That is, $\mathcal{X} \cup \mathcal{Y} \neq \emptyset$, where $\mathcal{X} = \text{Splt}(\hat{T}_i) \setminus \text{Splt}(T_i)$ and $\mathcal{Y} = \text{Splt}(T_i) \setminus \text{Splt}(\hat{T}_i)$. Set $\mathcal{X}$ consists of splits $X$ such that $|K_Y(G)| > |K_X(G)|$ and $\mathcal{Y}$ consists of splits $Y$ such that $|K_Y(G)| > |K_X(G)|$. By Theorem 1, $T = \text{Maj}^+(G)$ displays all splits $X$ such that $|K_Y(G)| > |K_X(G)|$. Thus, $d(T, \hat{T}_i) < d(T, T_i)$.

As we are assuming that there is at least one such $i \in K$, we have $\Sigma_{j \in K} d(T, \hat{T}_i) < \Sigma_{j \in K} d(T, T_j)$, contradicting the fact that $T$ is a minimal optimal candidate supertree for $G$.

Motivated by Theorem 3, we define the adjusted score of a representative selection $R$ for a profile $P$, denoted $\hat{s}(R)$, to be the score of the completion $\hat{R}$ of $R$; i.e., $\hat{s}(R) = s(\hat{R})$. Recall that $s(R) = \text{dist}(\text{Maj}(R), R)$.

**Theorem 4.** Let $P$ be a profile. Define $G = \{G \in \langle P \rangle : \hat{s}(G) \text{ is minimum} \}$ and $S = \{T = \text{Maj}^+(G) : G \in G \}$. Then, $\text{Maj}^+(P)$ is the strict consensus of $S$.

**Proof:** Let $\emptyset$ be the set of all optimal candidate supertrees for $P$ and let $M$ be the set of all minimal optimal candidate supertrees of $P$. In what follows, we show that $M \subseteq S \subseteq \emptyset$. This immediately implies the theorem, because not only is (by definition) $\text{Maj}^+(P)$ the strict consensus of $\emptyset$, but it must also be the strict consensus of $S$.

Suppose $T \in M$. We claim that $T \in S$ and, therefore, that $M \subseteq S$. Let $R$ be a representative selection for $P$ such that $T = \text{Maj}(R)$. Let $G$ be any restricted representative selection for $P$ displayed by $R$. By Theorem 3, $T = \text{Maj}^+(G)$ and $R$ is the completion of $G$. We claim that $G \in G$; i.e., $\hat{s}(G)$ is minimum. Assume, by way of contradiction, that there is another $G' \in \langle P \rangle$ such that $\hat{s}(G') < \hat{s}(G)$. Let $R'$ be the completion of $G'$. Then, $s(R') = s(G') < s(G) = s(R)$, which contradicts the assumption that $T$ is optimal. Therefore, $\hat{s}(G)$ is minimum and $T \in S$.

Suppose $T \in S$. We claim that $T \in \emptyset$ and, therefore, that $S \subseteq \emptyset$. Let $G \in \langle P \rangle$, be such that $T = \text{Maj}^+(G)$ and the adjusted score $\hat{s}(G)$ is minimum. Let $R$ be the completion of $G$. Assume, by way of contradiction, that $T \notin S$. Then there is a $T' \in M$ such that, if $R'$ is a representative selection for $P$ where $T' = \text{Maj}(R')$, then $s(R') < s(R)$. By Theorem 3, there is a $G' \in \langle P \rangle$ such that $T' = \text{Maj}^+(G')$ and $\hat{s}(G') = s(R')$. Then $\hat{s}(G') = s(R') < s(R) = \hat{s}(G)$. This contradicts the assumption that $\hat{s}(G)$ is minimum.

**ILP formulation**

In this section we first describe an ILP formulation of the optimal candidate supertree problem based on Theorem 4. The optimum solution to this ILP is a $G \in \langle P \rangle$, with minimum adjusted score. For ease of exposition, we divide the variables of our ILP into three categories: fill-in variables, which represent the way taxa are added to the input trees to create $G$; objective function variables, which are used to express $\hat{s}(G)$; and auxiliary variables, which are used to establish a connection between the fill-in and objective function variables. All variables are binary. After presenting our ILP model, we discuss how to use it to generate $\text{Maj}^+(P)$.

**Fill-in variables**

At the core of our ILP formulation is a matrix representation of the input trees similar to that used in MRP [18,19]. Let $P = (t_1, ..., t_k)$ be a profile where $|\mathcal{L}(P)| = n$. Assume input tree $t_i$ has $m_i$ nontrivial splits, which are assumed to be ordered in some fixed but arbitrary way. A matrix representation of $t_i$ is a $n \times m_i$ matrix $M(t_i)$ whose columns are in one to one correspondence with the nontrivial splits of $t_i$.

Suppose column $i$ of $M(t_i)$ corresponds to split $A \mid B$ in $t_i$ and let $x$ be a taxon in $\mathcal{L}(P)$. Then, $M_{n \times m}(t_i(t_j) = 1$ if $x \in A$, $M_{n \times m}(t_i(t_j) = 0$ if $x \in B$, and $M_{n \times m}(t_i(t_j) = ?$ otherwise. We note that for unrooted trees the assignment of 1 to the $A$ side of the split and of 0 to the $B$ side is arbitrary. For rooted trees, all taxa in the side of a split that contains the root are assigned a 1.

Let $m = \Sigma_{j \in K} m_j$. A matrix representation of $P$, denoted $M(P)$, is a $n \times m$ matrix $M(P)$ obtained by concatenating matrices $M(t_1), ..., M(t_k)$.

A fill-in of matrix $M(P)$ is a matrix representation for a restricted representative selection $G$ for $P$. Note that $M(G)$ has no question marks and that, for every taxon $x$ and split $i$ such that $M_{n \times m}(P(t_j)(x) \in \{0, 1\}$, we have $M_{n \times m}(P(t_j)(x) = M_{n \times m}(P(t_j)(x)$. To represent fill-ins of $M(P)$, the ILP associates a fill-in variable $F_{x \mid i}$ with each $x$ and $i$. If $M_{n \times m}(P(t_j)(x) = 0$, then $F_{x \mid i} = M_{n \times m}(P(t_j)(x)$. We note that for unrooted trees the assignment of a value of 0 or 1, representing an assignment of taxon $x$ to one of the two sides of split $i$. Our ILP has constraints (described below) to ensure that each value assignment to the $F$-variables corresponds to a restricted
representative selection for \( P \). That is, there must exist a \( G \in \langle P \rangle \), such that \( M_G(G) = F_x \) for every \( x \) and \( i \).

**Objective function variables**

The objective is to minimize \( \hat{d} (G) \) over all \( G \in \langle P \rangle \), where each \( G \) is represented by a fill-in of \( M(P) \). By definition, \( \hat{d} (G) = \text{dist}(\text{Maj}^+(G), R) \), where \( R = (\hat{T}_1, \ldots, \hat{T}_k) \) is the completion of \( G = (T_1, \ldots, T_k) \). We do not, however, construct \( \text{Maj}^+(G) \) and \( R \) explicitly. Instead, we proceed indirectly, using the fact that, by Theorems 1 and 3, all splits in \( \text{Maj}^+(G) \) and \( R \) are already in \( G \). Indeed, those theorems and the definition of Robinson-Foulds distance (Equation 1) imply that

\[
\text{dist}(\text{Maj}^+(G), R) = \sum_{j \in K} | \text{Spl}(\text{Maj}^+(G)) \setminus \text{Spl}(T_j) | + \sum_{j \in K} | \text{Spl}(T_j) \setminus \text{Spl}(\text{Maj}^+(G)) |.
\]

The next result, which follows Theorems 1 and 3, allows us to count directly from \( G \) the contribution of each split \( X \in \text{Spl}(\text{Maj}^+(G)) \cup \text{Spl}(\hat{T}_j) \) to \( d(\text{Maj}^+(G), \hat{T}_j) \).

**Lemma 1.** Let \( P \) be a profile and suppose \( G \in \langle P \rangle \). Then, for each \( j \in K \),

(i) \( X \in \text{Spl}(\text{Maj}^+(G)) \setminus \text{Spl}(\hat{T}_j) \) if and only if \( | K_X(G) | = | K_{\hat{T}_j}(G) | \) and \( j \in K_{\hat{T}_j}(G) \).

(ii) \( X \in \text{Spl}(\hat{T}_j) \setminus \text{Spl}(\text{Maj}^+(G)) \) if and only if \( | K_X(G) | \leq | K_{\hat{T}_j}(G) | \) and \( j \in K_{\hat{T}_j}(G) \).

Suppose we have a fill-in for \( M(P) \) that corresponds to some \( G = (T_1, \ldots, T_k) \in \langle P \rangle \). Our ILP has two kinds of objective function variables. The first group of variables are denoted \( w_1, \ldots, w_m \), where \( w_i \) corresponds to the \( i \)th column of \( M(G) \). This column corresponds to split \( X \) in tree \( T_j \); thus, \( j \in K_X(G) \). Our ILP has constraints such that \( w_i = 1 \) if and only if \( | K_X(G) | \geq | K_{\hat{T}_j}(G) | \). Thus, \( w_i = 0 \) means that \( | K_X(G) | \leq | K_{\hat{T}_j}(G) | \), which, together with Lemma 1 (ii), implies that \( \sum_{i=1}^m (1 - w_i) = \sum_{j \in K} | \text{Spl}(T_j) \setminus \text{Spl}(\text{Maj}^+(G)) | \).

The second group of variables are denoted \( z_{ij} \), \( 1 \leq i \leq m \), \( 1 \leq j \leq k \). Suppose column \( j \) of \( M(P) \) corresponds to split \( X \). Our ILP has constraints such that \( z_{ij} = 1 \) if and only if \( w_i = 1 \) (i.e., \( | K_X(G) | \geq | K_{\hat{T}_j}(G) | \) and \( j \in K_{\hat{T}_j}(G) \) and \( j = \min \{ \ell : \ell \in K_{\hat{T}_j}(G) \} \). Thus, by Lemma 1 (i),

\[
\sum_{i=1}^m \sum_{j=1}^k z_{ij} = \sum_{j \in K} | \text{Spl}(\text{Maj}^+(G)) \setminus \text{Spl}(T_j) |.
\]

The objective function can now be expressed as

\[
\text{minimize } \sum_{i=1}^m \sum_{j=1}^k z_{ij} + \sum_{i=1}^m (1 - w_i).
\]

**Auxiliary variables and constraints**

As mentioned earlier, all variables, including the auxiliary ones, are Boolean. We take advantage of this, expressing the constraints relating the variables as Boolean expressions in terms of the “and,” “or,” “exclusive or,” and “if and only if” operators (denoted by the usual symbols, and, \( \land \), \( \lor \), and \( \Leftrightarrow \), respectively). We then convert these expressions into equivalent linear inequalities on zero-one variables using standard techniques [[30], pp. 231-244].

We first describe the variables and constraints that are used to ensure that the settings of the fill-in variables (the \( F \) variables) correspond to a restricted representative selection. That is, the assignments to the \( F \) variables must be such that, for each input tree \( t_j \), the resulting plenary splits associated with the tree are pairwise compatible, so that they yield a plenary tree \( T_j \in \langle t_j \rangle \). For this purpose, we define variables \( C_{pq} \), \( 1 \leq p, q \leq m \) and add constraints linking these variables and the \( F \) variables such that \( C_{pq} = 1 \) if and only if columns \( p \) and \( q \) are compatible under the fill-in represented by the \( F \) variables. To guarantee that the assignment to the \( F \) variables corresponds to a restricted representative selection, we require that \( C_{pq} = 1 \) for every two column indices \( p, q \) that correspond to splits in the same input tree. We note that the constraints relating the fill-in variables \( F \) and the \( C \)-variables closely resemble the ones used by Gusfield et al. [26]. One difference is that for our problem we need “if and only if” relationships, whereas Gusfield et al. require only one direction of the implication.

The constraints on the \( C \)-variables use the fact that splits \( p \) and \( q \) are incompatible if and only if \( 00, 01, 10 \), and \( 11 \) all appear in some rows of columns \( p \) and \( q \) (the “four gametes condition”). The presence or absence of these patterns for columns \( p \) and \( q \) is indicated by the settings of variables \( B_{ab} \), \( a, b \in \{0, 1\} \), where \( B_{ab} = 1 \) if and only if there is a taxon \( r \) such that \( F_{rp} = a \) and \( F_{rq} = b \). The \( B_{ab} \) are determined from the settings of variables \( \Gamma_{r_{pq}} \), where \( r \) ranges over the taxa (i.e., the rows of \( M(P) \)). The \( \Gamma \) variables satisfy \( \Gamma_{r_{pq}} \Leftrightarrow ((F_{rp} = a) \land (F_{rq} = b)) \). This condition is expressed by the following constraints.

\[
(1) F_{rp} + (1) F_{rq} + \Gamma_{r_{pq}} \geq 1 - a - b,
\]

\[
(2) F_{rp} + (1) F_{rq} + 2\Gamma_{r_{pq}} \leq 2 - a - b.
\]

We have that \( B_{pq} \Leftrightarrow \lor, \Gamma_{r_{pq}} \), which is expressed by the inequalities below.

\[
\sum_i \Gamma_{r_{pq}} + B_{pq} \leq 0,
\]

\[
\sum_i \Gamma_{r_{pq}} - nB_{pq} \leq 0
\]
Observe that $\neg C_{pq} \iff B_{pq}^{(00)} \land B_{pq}^{(01)} \land B_{pq}^{(10)} \land B_{pq}^{(11)}$. Equivalently we have the constraints below.

$$
\begin{align*}
B_{pq}^{(00)} + B_{pq}^{(01)} + B_{pq}^{(10)} + 4C_{pq} & \geq 4, \\
B_{pq}^{(00)} + B_{pq}^{(01)} + B_{pq}^{(11)} + C_{pq} & \leq 4.
\end{align*}
$$

(5)

We now consider the variables and constraints that enable us to express the objective function variables. There are three main sets of variables:

- For $1 \leq p \leq m$, $D_p$ equals 1 if and only if column $p$ represents the same split as some column with smaller index.
- For $1 \leq i \leq m$, $S_{ij}^{(1)}$, equals 1 if and only if split $i$ is in tree $j$.
- For $1 \leq i \leq m$, $1 \leq j \leq k$, $S_{ij}^{(2)}$ equals 1 if and only if split $i$ is compatible with tree $j$.

As we shall see, the values of the $w$ and the $z$ variables in the objective function are determined, respectively from the $S_{ij}^{(1)}$ and $S_{ij}^{(2)}$ variables, and from the $w$, $S_{ij}^{(2)}$, and $D$ variables.

The value of $E_{pq}$ depends on the patterns that appear in columns $p$ and $q$, which can be deduced from the values of $B_{pq}^{(ab)}$ for different choices of $a$ and $b$ as follows.

- For rooted trees, $E_{pq} \iff \neg B_{pq}^{(01)} \land \neg B_{pq}^{(10)}$. This is expressed as follows.

$$
\begin{align*}
B_{pq}^{(01)} + B_{pq}^{(10)} + 2E_{pq} & \leq 2, \\
B_{pq}^{(01)} + B_{pq}^{(10)} + E_{pq} & \geq 1.
\end{align*}
$$

(6)

- For unrooted trees, we introduce two auxiliary variables $\delta_{pq}^{(1)}$ and $\delta_{pq}^{(2)}$ such that

$$
\delta_{pq}^{(1)} \iff \neg B_{pq}^{(01)} \land \neg B_{pq}^{(10)} \quad \text{and} \quad \delta_{pq}^{(2)} \iff \neg B_{pq}^{(00)} \land \neg B_{pq}^{(11)}.
$$

Then,

$$
E_{pq} \iff \delta_{pq}^{(1)} \oplus \delta_{pq}^{(2)}.
$$

These logical constraints are expressed by the following inequalities.

$$
\begin{align*}
B_{pq}^{(01)} + B_{pq}^{(10)} + 2S_{pq}^{(1)} & \leq 2, \\
B_{pq}^{(01)} + B_{pq}^{(10)} + \delta_{pq}^{(1)} & \geq 1, \\
B_{pq}^{(00)} + B_{pq}^{(11)} + 2S_{pq}^{(2)} & \leq 2, \\
B_{pq}^{(00)} + B_{pq}^{(11)} + \delta_{pq}^{(2)} & \geq 1, \\
E_{pq} - \delta_{pq}^{(1)} - \delta_{pq}^{(2)} & = 0.
\end{align*}
$$

(7)

We are now ready to give the constraints for the $D$, $S_{ij}^{(1)}$ and $S_{ij}^{(2)}$ variables. Observe that $D_i = 0$ and that, for $1 < p \leq m$, $D_p \iff \bigvee_{i=1}^{p-1} E_{ip}$. Equivalently we have

$$
\begin{align*}
D_p - \sum_{i=1}^{p-1} E_{ip} & \leq 0, \\
\sum_{i=1}^{p-1} E_{ip} - p \cdot D_p & \leq 0.
\end{align*}
$$

(8)

In describing the constraints for the $S_{ij}^{(1)}$ and $S_{ij}^{(2)}$ variables, we adopt the convention that the splits of the $j$th tree correspond to columns $j_1, \ldots, j_d$ of $M(P)$. Then, $S_{ij}^{(1)} \iff E_{j_1} \oplus \cdots \oplus E_{j_d}$. This translates into the equality constraint

$$
S_{ij}^{(1)} - \sum_{r=1}^{d} E_{j_r} = 0.
$$

(9)

On the other hand, $S_{ij}^{(2)} \iff C_{j_1} \land \cdots \land C_{j_d}$. This is equivalent to the two constraints below.

$$
\begin{align*}
d \cdot S_{ij}^{(2)} - \sum_{r=1}^{d} C_{j_r} & \leq 0, \\
1 - d - S_{ij}^{(2)} + \sum_{r=1}^{d} C_{j_r} & \leq 0.
\end{align*}
$$

(10)

Finally, we describe how the objective function variables relate to the auxiliary variables. For each $i$, $w_i = 1$ if and only if $\sum_{j=1}^{k} S_{ij}^{(1)} > k - \sum_{j=1}^{k} S_{ij}^{(2)}$. This is expressed by the following two constraints.

$$
\begin{align*}
k \cdot w_i + 1 - \sum_{j=1}^{k} S_{ij}^{(1)} - k \cdot \sum_{j=1}^{k} S_{ij}^{(2)} & \leq 0, \\
\sum_{j=1}^{k} S_{ij}^{(1)} + \sum_{j=1}^{k} S_{ij}^{(2)} - k - k \cdot w_i & \leq 0.
\end{align*}
$$

(11)

It follows from the definition of the $z$ variables that, for every $i$, $j$, $z_{ij} \iff w_i \land \neg S_{ij}^{(1)} \land \neg D_i$. Equivalently we have the following.

$$
\begin{align*}
-2 - w_i + S_{ij}^{(2)} + D_i + 3 \cdot z_{ij} & \leq 0, \\
w_i - S_{ij}^{(2)} - D_i - z_{ij} & \leq 0.
\end{align*}
$$

(12)

Table 1 summarizes the number of variables of each kind in our integer programming formulation. Here, as usual, $n$ is the total number of taxa and $m$ is the total number of splits in the input trees. As can be seen,
there are a total of $O(nm^2)$ variables; this number is dominated by the $\Gamma$ variables. The total number of constraints for the unrooted case, broken down by constraint type, is given by the following expression.

$$\frac{4nm(m-1) + 4m(m-1) + m(m-1) + 5m(m-1)}{2} + 2(m-1) + mk + 2mk + 2m + 2mk = O(nm^2).$$

The number of constraints for the rooted case is slightly smaller, but of the same order of magnitude. It should be noted that the expressions given in Table 1 assume that all the variables listed are indeed variables. In reality, the values of many of the $F$ variables are fixed because they correspond to non-question-mark entries in $M(P)$. This in turn fixes the values for several $\Gamma$ variables, as well as those of other variables. As a consequence, the number of true variables in the ILP formulation is typically much smaller than the worst case estimates in Table 1. In general, the larger the number of question marks in matrix $M(P)$, the closer the problem size will be to the worst case estimates.

**Building Maj$^+(P)$**

The ILP model just outlined allows us to find a $G \in \langle P \rangle_r$ corresponding to some optimal candidate supertree $T^*$. To build Maj$^+(P)$ we need, in principle, the set of all such $G$. While there are ways to enumerate this set [31], we have found that an alternative approach works much better in practice. The key observation is that, since Maj$^+(P)$ is the strict consensus of all optimal candidate supertrees, each split in Maj$^+(P)$ must also be in $T^*$. Thus, once we have $T^*$, we simply need to verify which splits in $T^*$ are in Maj$^+(P)$ and which are not. To do this, for each split $A|B$ in $T^*$, we put additional constraints on the original ILP requiring that the optimal tree achieve an objective value equal or smaller than that of $T^*$ and not display split $A|B$. The resulting ILP has only $O(mn)$ more variables and constraints than the original one. If the new ILP is feasible, then $A|B \notin \text{Spl(Maj}^+(P))$; otherwise, $A|B \in \text{Spl(Maj}^+(P))$. We have found that detecting infeasibility is generally much faster than finding an optimal solution.

**A data reduction heuristic**

The ILP formulation described in the previous section allows us to solve supertree problems of moderate size. Here we describe a data reduction heuristic that allows us to extend the range of our method significantly in practice, by exploiting the structure that is present in certain supertree problems. Our data reduction heuristic applies when the input profile $P = (t_1, \ldots, t_k)$ contains a subset of taxa $S$ that can be treated as a single supertaxon. Roughly stated, we are looking for a set $S$ such that every tree in $P$ respects the split implied by $S$. We now define this concept more precisely.

Let $\text{Spl}_0(T)$ denote the set of all full splits displayed by $T$. That is, $\text{Spl}_0(T)$ includes the non-trivial and the trivial splits displayed by $T$; in particular, $\mathcal{L}(T) \cap \emptyset = \text{Spl}_0(T)$. We say that $S \subseteq \mathcal{L}(P)$ with $1 < |S| < |\mathcal{L}(P)| - 1$ is a reducible set if, for each $j \in K$, there is a split $A|B \in \text{Spl}_0(t_j)$ such that $A \cap S = A$ and $B \cap S = \emptyset$. Ideally, a reducible set should correspond to a widely-acknowledged biological classification unit. For example, some of the trees in a collection of phylogenies may contain subtrees corresponding to different (possibly empty) subsets of the pri-mates. While these subsets may not be identical, and the subtrees may disagree somewhat in their topologies, the input phylogenies are likely to separate primates from non-primates. In settings like this, it makes intuitive sense to restrict our attention to supertrees where reducible sets appear as clusters.

Given a reducible set $S$ for $P$, we can define two smaller subproblems.

- The **reduced profile associated with a reducible set** $S$ is the profile $P^{\text{Red}} = \langle t_1^{\text{Red}}, \ldots, t_k^{\text{Red}} \rangle$ where, for each $j \in K$, $t_j^{\text{Red}}$ is the tree obtained from $t_j$ by contracting the minimal subtree of $t_j$ containing $S \cap \mathcal{L}(t_j)$ to a single leaf node $\beta_S$. If $S \cap \mathcal{L}(t_j) = \emptyset$, then $t_j^{\text{Red}} = t_j$. We refer to $\beta_S$ as the supertaxon associated with $S$.

- The **satellite profile associated with $S$** is the profile $P^{\text{Sat}} = \langle t_1^{\text{Sat}}, \ldots, t_k^{\text{Sat}} \rangle$ where $t_j^{\text{Sat}}$ is obtained from $t_j$ by contracting the minimal subtree of $t_j$ containing $(\mathcal{L}(P) \setminus S) \cap \mathcal{L}(t_j)$ to a single leaf node $\rho_S$. Note that some of the trees in the satellite profile associated with $S$ may contain only $\rho_S$. The compressed satellite profile associated with $S$ is the satellite profile associated with $S$ with all of the latter trees removed.

An $S$-restricted representative selection for $P$ is a selection $R = (T_1, \ldots, T_i) \in \langle P \rangle$ such that $S(\mathcal{L}(P) \setminus S) \in \text{Spl}(T_i)$ for all $i \in K$. An optimal $S$-restricted candidate representative selection is an $S$-restricted representative selection $R$ with minimum score, and Maj$(R)$ an optimal $S$-restricted candidate supertree. The $S$-restricted majority-rule (+) supertree is the strict consensus of all the optimal $S$-restricted candidate supertrees.

It should be noted that, given an arbitrary reducible set $S$, it is not true in general that an optimal $S$-restricted candidate supertree will be an optimal candidate supertree, nor that an $S$-restricted majority-rule (+) supertree will also be a majority-rule (+) supertree. This is illustrated in Figure 1, which shows a profile $P = (t_1,$

### Table 1 Variables in the ILP formulation

| $F$ | $\Gamma$ | $B$ | $\delta^0$ | $E$ | $C$ | $D$ | $\delta^1$ | $w$ | $z$ |
|-----|---------|-----|-----------|-----|-----|-----|-----------|-----|-----|
| $mn$ | $2m(m-1)$ | $2m(m-1)$ | $m(m-1)/2$ | $2m(m-1)$ | $2m(m-1)$ | $m$ | $mk$ | $m$ | $mk$ |

The number of variables of each kind is expressed in terms of $n$, $m$, and $k$, the total number of taxa, the total number of splits in the input trees, and the number of input trees, respectively.

http://www.almob.org/content/5/1/2
where both \{b, e\} and \{b, d\} are reducible sets, but where neither optimal candidate tree contains the cluster \{b, d\}, although they both contain \{b, e\}.

On the other hand, a reducible set may represent useful biological knowledge that should be incorporated into a supertree analysis. There are also computational benefits. With the right choice of \(S\) (one where \(|S|\) is far from the extreme values of 2 and \(|\mathcal{L}(P)|\cdot2\)), the reduced and satellite profiles can be considerably smaller than the original profile, and the corresponding integer programs will have fewer unknown variables. As the following theorem indicates, an optimal \(S\)-restricted candidate supertree can be found by solving the associated subproblems separately and combining their answers.

**Theorem 5.** Let \(P\) be a profile and \(S\) be a reducible set in \(P\). Let \(T^{\text{Red}}\) and \(T^{\text{Sat}}\) be optimal candidate trees for the reduced profile associated with \(S\) and the compressed satellite profile associated with \(S\). Let \(T\) be the tree obtained by identifying the node \(b_S\) in \(T^{\text{Red}}\) and node \(p_S\) in \(T^{\text{Sat}}\) and then suppressing the resulting degree-two vertex. Then, \(T\) is an optimal \(S\)-restricted candidate supertree for \(P\). Further, if \(R\) is the optimal \(S\)-restricted representative selection corresponding to \(T\) and \(R^{\text{Red}}\) and \(R^{\text{Sat}}\) are the optimal representative selections corresponding to \(T^{\text{Red}}\) and \(T^{\text{Sat}}\), respectively, then \(s(R) = s(R^{\text{Red}}) + s(R^{\text{Sat}})\).

The straightforward proof of this result is omitted. A direct consequence is that the \(S\)-restricted majority-rule (+) supertree can be obtained by piecing together the majority-rule (+) supertrees for the reduced and satellite profiles. Observe that if multiple pairwise disjoint reducible sets are known, then each of the corresponding compressed satellite profiles can be solved independently, and the original profile can be reduced by replacing each reducible set to a distinct supertaxon. In fact, the idea can be used recursively, so that a satellite profile can itself be decomposed to a reduced profile and (sub) satellites. As we shall see later, this can result in dramatic problem size reductions.

**Results and discussion**

Here we report on computational tests with the exact ILP method and the data reduction heuristic. All our experiments were conducted on real data sets, rather than simulated data. We did this because we were interested in seeing if the groupings of taxa generated by majority-rule (+) supertrees would coincide with those commonly accepted by biologists. Another goal of our experiments was to compare the performance of the ILP formulation without data reduction, which we refer to as the basic method, against that of ILP plus data reduction. All trees considered in our tests were rooted.

To conduct our tests of the basic method, we wrote a program to generate the ILPs from the input profiles.

For our tests of the data reduction heuristic, we used different methods to find reducible sets in a profile; these are outlined later. Given the reducible sets, the corresponding reduced and satellite profiles, as well as the associated ILPs, were generated automatically. All ILPs were then solved using CPLEX (CPLEX is a trademark of ILOG, Inc.) on an Intel Core 2 64 bit quad-core processor (2.83 GHz) with 8 GB of main memory and a 12 MB L2 cache per processor.

**Experiments with the basic ILP formulation**

We tested the basic ILP formulation on five published data sets. The *Drosophila A* data set is the example studied in [14], which was extracted from a larger *Drosophila* data set considered by Cotton and Page [32]. *Primates* is the smaller of the data sets from [5]. *Drosophila B* is a larger subset of the data studied in [32] than that considered in [14]. *Chordata A and B* are two extracts from a data set used in a widely-cited study by Delsuc et al. [33]. Chordata A consists of the first 6 trees with at least 35 taxa (out of 38). Chordata B consists of the first 12 trees with at least 37 taxa (out of 38).

The results are summarized in Table 2. Here \(n, m, k\), and \(\ell\) are the number of taxa, total number of splits, and number of trees, respectively. \(\mathcal{L}\) denotes the number of question marks in \(M(P)\), the matrix representation of the input; \(N\) is the size of the CPLEX-generated reduced ILP. Table 2 shows the time to solve the ILP and produce an optimal candidate supertree \(T^*\) and the time to verify all the splits of \(T^*\) to produce \(\text{Maj}^*(P)\).

**Experiments with the data reduction heuristic**

As a preliminary test, we compared the results obtained via the reduction heuristic with the exact solutions, obtained using the basic ILP method, for two of the data sets listed in Table 2. For simplicity, only clusters from the input trees were used as reducible sets. (Note that unions of input clusters could have also been used as reducible sets.) We wrote a program that chooses clusters greedily. At every step, it selects the largest non-trivial cluster present in some input tree that does not overlap with any of the previously chosen clusters.

For the Primates data set, the optimal objective value (i.e., the score of an optimal candidate supertree) for the original profile is 9. We found six pairwise disjoint reducible sets, and built the corresponding reduced and satellite profiles. The optimal objective values of the reduced profile, first, second and third satellite profiles are 0, 4, 3, and 2, respectively. The other satellite profiles have an optimal objective value of 0. Thus, the total score of the reduced and satellite profiles matches the optimal score for the original profile, and the supertree obtained using the heuristic is also optimal. The reduction method also gives a correct optimal candidate supertree for *Drosophila B*. Here the original profile has
Figure 1 Optimal candidate trees do not always display reducible sets. Top row: An input profile $P = (t_1, t_2, t_3)$ where \{b, d\} and \{b, e\} are reducible sets. Second and third rows: Two optimal representative selections. Fourth row: Optimal candidate trees corresponding to the representative selections in rows two and three. Neither candidate tree contains the cluster \{b, d\}, although they both contain \{b, e\}.
an objective value of 8. We found nine pairwise disjoint reducible sets, and built the corresponding reduced and satellite profiles. The reduced profile has an optimal objective value of 8 and all satellite profiles have an optimal objective value of 0.

It should be pointed out that the reducible sets used for Primates and Drosophila B do not necessarily correspond to clusters in the majority-rule (+) supertree, although they are displayed by some optimal candidate trees. Thus, one will not obtain a majority-rule (+) supertree by simply composing the solutions to the reduced problems and the satellites. This indicates the importance of choosing relatively few large and well-supported reducible sets. Biological knowledge can serve as a good guide. For example using the clade Haplorhini as a reducible set for Primates data set, solving the corresponding reduced and satellite profiles and combining the respective majority-rule (+) supertrees one gets exactly the same supertree as through the basic (and exact) method. Similarly, using the subgenus Sophophora as a reducible set for Drosophila B, we, obtained precisely the majority-rule (+) supertree for the data set.

Next, we considered some data sets that are well beyond the reach of our basic ILP method. The Drosophila C data set is the full 6-tree Drosophila data set of Cotton and Page [32] from which the Drosophila A and B data sets were extracted. The Seabirds data set consists of the 7 trees in the seabirds study by Kennedy and Page [17]; which encompasses 122 taxa (note that one of these taxa is an outgroup, so we do not count it in our study). We also examined the full Chordata set of Delsuc et al. [33], which has 38 taxa and 146 trees.

### Chordata

We looked for reducible sets in the full Chordata data set by considering increasingly larger subprofiles, starting with one input tree and then including one more input tree at every step. For each subprofile, we conducted an exhaustive search for reducible sets. The number of reducible sets increased at first, then fluctuated, and finally declined. After the 20th tree, there were no reducible sets. Thus, the data reduction heuristic proved to be ineffective for this data set.

### Drosophila C

We identified seven reducible sets for Drosophila C. Six of these were found by the greedy approach; the seventh corresponded to the subgenus Sophophora (the latter was selected manually, after some of the subproblems identified by our program proved impossible to solve). Four of the associated satellites were trivially solvable, since each contained only two taxa. We then solved ILPs for the reduced and the nontrivial satellites. The running time statistics are summarized in Table 3, which shows the same kind of data shown in Table 2, except that this time it reports these statistics for the original, reduced and satellite problems. Notably, even though the original ILP was too large to be solved, the reduced profile was solved in less than 10 minutes and the satellite profiles were solved almost instantly.

### Seabirds

To handle the Seabirds data set, we identified three reducible sets, which yielded a reduced profile and three satellite profiles, numbered 1, 2, and 3. Satellite profile 3 was too big to be solved by the basic ILP method, so it was further reduced by identifying three reducible sets within it, which resulted in three (sub-) satellite profiles, numbered 3.1, 3.2, and 3.3. The various reducible sets correspond to biologically meaningful classification units, as we explain next. In what follows, we refer to

**Table 2 Summary of experimental results with the basic ILP method**

| Data set   | \(n\) | \(m\) | \(k\) | \(U\) | \%\(U\) | \(N\) | Sol. (sec) | Verif. (sec) |
|------------|-------|-------|-------|------|--------|------|----------|-------------|
| Drosophila A | 9     | 17    | 5     | 60   | 39.2   | 9.8 e5| 0.83     | 1.6         |
| Primates   | 33    | 48    | 3     | 590  | 37.3   | 7.8 e7| 15.83    | 2.86        |
| Drosophila B | 40    | 55    | 4     | 1133 | 51.5   | 1.25 e9| 362      | 19          |
| Chordata A | 38    | 290   | 6     | 330  | 3      | 1.40 e8| 120      | 258         |
| Chordata B | 38    | 411   | 12    | 306  | 2      | 1.05 e8| 986      | 1784        |

Size and solution times for the ILP formulations of the various data sets. Here \(U\) denotes the number of question marks in \(M(P)\), the matrix representation of the input; \(N\) is the size of the CPLEX-generated reduced ILP; \(n\), \(m\), and \(k\) are as in Table 1. Shown are the time to solve the ILP and produce an optimal candidate supertree \(T^*\) and the time to verify all the splits of \(T^*\) to produce \(\text{Maj}^\dagger(P)\).

**Table 3 Results of Drosophila C analysis using data reduction**

| Data set   | \(n\) | \(m\) | \(U\) | \%\(U\) | \(N\) | Sol. (sec) | Verif. (sec) |
|------------|-------|-------|------|--------|------|----------|-------------|
| Original   | 46    | 70    | 1998 | 62.1   | 9.4 e9| N/A      | N/A         |
| Reduced    | 17    | 33    | 264  | 47.06  | 1.7e7 | 543.16   | 50.4        |
| Satellite 1 | 17    | 17    | 146  | 50.52  | 2.3e6 | 0.23     | 0.28        |
| Satellite 2 | 6     | 4     | 0    | 0      | 0    | 0.00     | 0.00        |
| Satellite 3 | 3     | 3     | 0    | 0      | 0    | 0.00     | 0.02        |

Size and solution times for all six trees in trees in Cotton and Page’s Drosophila data set [32] without and with data reduction. The notation is the same as in Tables 1 and 2. Shown are the time to solve the ILP and produce an optimal candidate supertree, and the values of the objective function for the reduced and satellite profiles. The latter two values are not available (N/A) for the original profile, since the model was too big to be solved. Not listed in the table are four other trivially solvable two-taxon satellite profiles.
the 7 input trees of Kennedy and Page’s seabirds data set by the same letters A-G that those authors used in [17].

Satellite 1 comprises the family Spheniscidae (Penguins, 10 taxa), which agrees with widely-accepted classifications for seabirds [34]. Members of this family appear in input trees E, F, and G of [17], and clearly form clusters of their own. Satellites 2 and 3 correspond to Diomedeinae (Albatrosses, 22 taxa), and Procellariinae (gadfly petrels, shearwaters, fulmars and diving petrels, 73 taxa). This agrees with the Sibley-Ahlquist classification [35] (represented by tree G). The resulting reduced profile has 19 taxa (16 original taxa and three supertaxa).

Satellite 3 (Procellariinae) has three subsatellites. Satellite profile 3.1 comprises the genus Pterodroma (30 taxa). Satellite 3.2 is for genus Pelecanoides (four taxa). Satellite 3.2 is a combination of Puffinus and Calonectris (10 taxa), which is supported by [36] (tree E). With these three sub-satellites, the reduced Procellariinae profile has 23 taxa (20 original taxa and three supertaxa).

Table 4 summarizes the results on the Seabirds data set. The majority-rule (+) supertree is shown in Figure 2, along with the MRP strict consensus tree of [32]. While the original problem was too big for CPLEX to solve on our machine, the reduced model was solved in 6.5 seconds. Most subproblems were solved and verified in a negligible amount of time. A notable exception was the reduced version of satellite 3, which required almost a minute to solve and nearly one hour and 45 minutes to verify.

### Discussion

Our results using the basic ILP formulation compare well with the published ones. For Drosophila A we obtained exactly the same tree reported in [14]. For Primates, the output is exactly the same as [5], which was produced by PhySIC method. The coincidence with PhySIC is noteworthy, since this supertree is less controversial than the MRP, Mincut, and PhySICpC supertrees reported in [5]. The reason for the coincidence may lie in the fact that, while heuristic, PhySIC requires that all topological information contained in the supertree be present in an input tree or collectively implied by the input trees, which bears some similarity with properties (CW1)-(CW4) of majority (+) supertrees.

For Drosophila B, Cotton and Page [32] show four supertrees: strict consensus of gene tree parsimony (GTP), Adams consensus of GTP, strict consensus of MRP, Adams consensus of MRP. Among the 10 clusters found by our ILP, two are in all four of these supertrees, three are found in the Adams consensus of GTP and Adams consensus of MRP, one is in the strict and Adams consensus of GTP, and one is found in the strict and Adams consensus of MRP. Thus, with only four input trees we were able to generate a tree that is quite similar to the published results. For Chordata A, the 12 splits found matched published results [33] exactly. For Chordata B, the 14 splits found matched [33].

We have not mapped out the precise boundary within which it is feasible to use the basic ILP method. However, it appears that it may not extend much beyond the dimensions of the problems listed in Table 2. For example, Drosophila B contains four out of 6 of the trees studied in [32]. Adding a fifth tree to the data set yields a problem that could not be solved by the basic ILP method. A major factor here is that the size of our ILP grows as the square of the total number of splits in all trees, and the solution time is exponential in the worst case. Incorporating a new tree to Drosophila B could easily add enough splits to the problem to put it well beyond the reach of our technique. We should add that model size does not appear to be the sole factor that makes instances hard — sparsity also seems to play a role.

### Drosophila C

The majority-rule (+) supertree for Drosophila C constructed by our method (available upon request) has 15

| Data set       | n   | m   | U     | %U | N     | Sol. (sec) | Verif. (sec) |
|----------------|-----|-----|-------|----|-------|------------|--------------|
| Original       | 121 | 188 | 12620 | 55.4 | 2.63e12 | N/A        | N/A          |
| Reduced        | 19  | 24  | 188   | 41.2 | 7.1e6  | 6.51       | 1562         |
| Sat. 1         | 10  | 8   | 42    | 52.5 | 1.1e5  | 0.05       | 0.07         |
| Sat. 2         | 22  | 29  | 129   | 20.2 | 1.2e6  | 0.09       | 1.06         |
| Satellite 3 (reduced) | 23 | 39  | 370   | 41.3 | 5.1e7  | 52.3       | 6110         |
| Subsatellite 3.1 | 30 | 42  | 196   | 15.6 | 6.8e5  | 0.06       | 0.04         |
| Subsatellite 3.2 | 4  | 2   | 0     | 0   | 0     | 0.00       | 0.00         |
| Subsatellite 3.3 | 19 | 20  | 113   | 29.7 | 9.5e5  | 0.17       | 0.06         |

Size and solution times for all seven trees in Kennedy and Page’s seabirds data set [17] without and with data reduction. The notation is the same as in Table 3. The solution and verification times for the original profile are not available (N/A), since the model was too big to be solved.
Figure 2 Comparing the MRP strict consensus with the majority-rule (+) supertree. Left: The strict consensus of the most parsimonious trees obtained by Kennedy and Page for their seabirds data set [17]. Right: The majority-rule (+) supertree obtained by using the data reduction heuristic; the reducible sets used to construct it are indicated by heavy lines. Clusters that appear in one tree but not the other are numbered; the differences are explained in the text.
nontrivial clusters, while the MRP strict consensus tree of Cotton and Page [32] has 11. Of these only three appear in both trees. This rather surprising result motivated us to try to assess how well the input trees are represented by the supertree. To this end, we relied on the notions of support and conflict, along the lines proposed by Wilkinson et al. [37].

Let \( t \) be an input tree for a profile \( P \), \( T \) be a supertree for \( P \), and \( S \) be a non-trivial cluster in \( T \) (i.e., \( S \) does not contain the root of \( T \) and \( S \in \mathcal{L}(P)\backslash S \in \mathcal{S}(P)) \). Let \( S' = S \cap \mathcal{L}(t) \). We say that tree \( t \) supports \( S \) if \( S' \) is a non-trivial cluster in \( t \). Tree \( t \) is in conflict with \( S \) if \( S' \) is incompatible with \( t \); i.e., there is no tree \( t' \) with \( \mathcal{L}(t') = \mathcal{L}(t) \) such that \( \mathcal{S}(t) \cup \{S'\} \backslash \mathcal{S}(t') \subseteq \mathcal{S}(t) \). If \( t \) neither supports nor is in conflict with \( S \), we say that \( t \) is irrelevant to \( S \).

Theorem 1 hints that each cluster \( S \) in the majority-rule (+) supertree should have more input trees supporting it than contradicting it, even when most trees are irrelevant to \( S \). This indeed holds for the Drosophila C majority-rule (+) supertree: Every one of its non-trivial clusters is supported by at least one input tree and does not conflict with any input tree. In contrast, of the five clusters in the MRP strict consensus supertree for which support outweighs conflict, only three have no conflict with any input tree. Of the remaining clusters, three have the same amount of conflict as support, and for three others the amount of support is outweighed by the amount of conflict. In fact, among the latter, there is a cluster that is in conflict with five out of six of the input trees; the remaining tree is irrelevant to that cluster. We refrain from claiming the superiority of one supertree over the other, since the biological relevance of both trees needs to be studied in more detail.

**Seabirds**

Figure 2 compares the majority-rule (+) supertree for the seabirds data set, constructed using the data reduction heuristic, with the MRP strict consensus supertree that Kennedy and Page presented for the same data set [17]. The latter is the strict consensus of 10,000 equally parsimonious trees obtained using MRP. There are 66 nontrivial clusters in the majority-rule (+) supertree, compared with 75 nontrivial clusters in the MRP strict consensus tree (ignoring the outgroup). Among these clusters, 63 are present in both trees (95% of 66 and 84% of 75). The reducible sets used to construct the majority-rule (+) supertree are indicated by heavy lines. Note that these sets are also clusters in the MRP supertree.

Three clusters, numbered 1-3 in Figure 2, are in the majority-rule (+) supertree but not in the MRP tree; 12 clusters, numbered 4-15 in Figure 2, appear in the MRP tree but not in the majority-rule (+) tree. For each of the seven input trees (labeled A-G in [17]) and each of these 15 clusters, Table 5 indicates whether the tree supports, is in conflict with, or is irrelevant to the cluster. As Theorem 1 would lead us to expect, each of clusters 1-3 (from the majority-rule (+) tree) has more input trees supporting it than in conflict with it. Of the 12 clusters (4-15) that are present only in the MRP strict consensus tree, seven have as many trees in support as in conflict. The others have more support than conflict.

In general, it appears that MRP may have a bias toward preserving clusters that are present in trees that contain many members of the families represented in those clusters. This is noticeable for *Pterodroma*, where the disagreement between trees D and E is resolved in favor of the former five times to one, in clusters 7, 8, 9, 10, and 12 versus cluster 11. This may be related to the “size bias” that previous researchers have observed in MRP [38]: Here, even though E is the larger tree (90 taxa versus 30), D has more taxa in the *Pterodroma* genus (30 versus 16). Majority-rule (+) trees seem not to have such a bias, because the expansion process used to construct representative selections tends to put all input trees, regardless of their size, on equal footing. These are, of course, only preliminary observations; this issue clearly deserves further analysis.

**Conclusions**

Our results indicate that the majority-rule (+) method produces biologically reasonable phylogenies (i.e., phylogenies with no unsupported groups), and that the

**Table 5 Support and conflict for the Seabirds data set**

| Cluster | A | B | C | D | E | F | G |
|---------|---|---|---|---|---|---|---|
| 1       | i | s | i | i | i | i | i |
| 2       | s | s | s | i | s | s | s |
| 3       | s | s | s | i | c | s | s |
| 4       | s | s | s | i | s | c | c |
| 5       | i | s | s | i | s | i | c |
| 6       | i | s | s | i | s | c | c |
| 7       | i | i | i | s | c | i | i |
| 8       | i | i | i | s | c | i | i |
| 9       | i | i | i | s | c | i | i |
| 10      | i | i | i | s | c | i | i |
| 11      | i | i | i | c | s | i | c |
| 12      | i | i | i | s | c | i | i |
| 13      | i | i | i | s | i | c |
| 14      | i | i | s | i | s | i | c |
| 15      | i | i | s | i | s | c | c |

For each of the 15 clusters that is present in only the majority-rule (+) supertree or only the MRP strict consensus supertree of the Seabirds data set, we indicate whether each of input trees A through G supports (s), is in conflict with (c), or is irrelevant to (i) the cluster. The numbering of the clusters follows Figure 2.
method is practical for medium-scale problems. Unfortunately, while polynomial, the size of our ILP is quadratic in the total number of splits in the input trees. This, together with the fact that solving the ILP takes exponential time in the worst case limits the range of applicability of the basic ILP formulation. It also explains in part why the addition of a single tree to a data set can convert a tractable problem into an intractable one. More extensive tests are needed to assess the limitations of the basic ILP approach accurately. In any event, our computational experience shows that the technique does handle some real, biologically significant, problems nicely. Moreover, our results suggest that the ILP approach, in combination with our data reduction heuristic is a promising way to tackle larger problems.

Acknowledgements
The authors benefited greatly from discussions with James Cotton, William HE Day, RC Powers, and Mark Wilkinson. We thank Frédéric Delucia for providing us the data set from [33]. This work was supported in part by the National Science Foundation under grants DEB-0334832 and DEB-0829674.

Author details
1 Department of Computer Science, Iowa State University, Ames, IA 50011, USA. 2 Department of Applied Mathematics, Illinois Institute of Technology, Chicago, IL 60616, USA.

Authors’ contributions
JD developed the methods, programmed them, conducted the computational experiments, and wrote the first draft of the manuscript. DFB supervised the work of JD, and contributed to the method development, computational experiments, and wrote the first draft of the manuscript. DFB, WM, and DCM contributed to the method development, especially to the supervised data set can convert a tractable problem into an intractable one. More extensive tests are needed to assess the limitations of the basic ILP approach accurately. In any event, our computational experience shows that the technique does handle some real, biologically significant, problems nicely. Moreover, our results suggest that the ILP approach, in combination with our data reduction heuristic is a promising way to tackle larger problems.

Acknowledgements
The authors benefited greatly from discussions with James Cotton, William HE Day, RC Powers, and Mark Wilkinson. We thank Frédéric Delucia for providing us the data set from [33]. This work was supported in part by the National Science Foundation under grants DEB-0334832 and DEB-0829674.

Author details
1 Department of Computer Science, Iowa State University, Ames, IA 50011, USA. 2 Department of Applied Mathematics, Illinois Institute of Technology, Chicago, IL 60616, USA.

Authors’ contributions
JD developed the methods, programmed them, conducted the computational experiments, and wrote the first draft of the manuscript. DFB supervised the work of JD, and contributed to the method development, the experimental design, and to the writing of the manuscript. FRMC contributed to the theoretical foundations of the method, especially to the formulation and proof of Theorem 1; he also contributed to the writing of the manuscript.

Competing interests
The authors declare that they have no competing interests.

Received: 11 August 2009 Accepted: 4 January 2010 Published: 4 January 2010

References
1. Gordon AD: Consensus supertrees: The synthesis of rooted trees containing overlapping sets of labelled leaves. Journal of Classification 1986, 3(3-4).
2. Bininda-Emonds ORP, Cardillo M, Jones KE, MacPhee RDE, Beck RMD, Greer R, Price SA, Vos RA, Gittleman J, Purvis A: The delayed rise of present-day mammals. Nature 2007, 446:507-512.
3. Bininda-Emonds ORP, D'Eil Cladistic Supertrees: Combining Information to Reveal the Tree of Life, Volume 4 of Series on Computational Biology Berlin: Springer 2004.
4. Wilkinson M, Cotton JA, Lapointe FJ, Pisani D: Properties of supertree methods in the consensus setting. Systematic Biology 2007, 56(3):330-337.
5. Ranwez V, Berry V, Criscuolo A, Fabre PH, Guillemot S, Scornavacca C, Douzery EJP: PhyloSC: A veto supertree method with desirable properties. Systematic Biology 2007, 56(5):798-817.
6. Adams EN: Consensus techniques and the comparison of taxonomic trees. Systematic Zoology 1972, 21(4):390-397.
7. Bryant D: A classification of consensus methods for phylogenetics. Bioconsensus, Volume 61 of Discrete Mathematics and Theoretical Computer Science Providence, RI: American Mathematical Society 2007, 153-58.
8. Day W, McMorris F: Axiomatic Consensus Theory in Group Choice and Biomathematics Philadelphia, PA: SIAM Frontiers in Mathematics 2003.
9. Barthélemy JP, McMorris FR: The median procedure for n-trees. Journal of Classification 1986, 3:229-334.
10. Margush T, McMorris FR: Consensus n-trees. Bulletin of Mathematical Biology 1981, 43:239-244.
11. Amerini, N, Clarke F, St John K: A linear-time majority tree algorithm. Proc 3rd Workshop Algs. in Bioinformatics (WABI'03), Volume 2812 of Lecture Notes in Computer Science Springer-Verlag 2003, 216-226.
12. Pattengale ND, Gottlieb EI, Moret BM: Efficiently computing the Robinson-Foulds metric. Journal of Computational Biology 2007, 14(6):724-735.
13. Robinson DF, Foulds LR: Comparison of phylogenetic trees. Mathematical Biosciences 1981, 53:131-147.
14. Cotton JA, Wilkinson M: Majority-rule supertrees. Systematic Biology 2007, 56:445-452.
15. Goloboff PA, Pol D: Semi-strict supertrees. Cladistics 2005, 21(5):514-525.
16. Dong J, Fernández-Baca D: Properties of majority-rule supertrees. Systematic Biology 2009, 58(3):560-567.
17. Kennedy M, Page RDM: Seabird supertrees: combining partial estimates of procellariiform phylogeny. The Auk 2002, 119(1):88-108.
18. Baum BR: Combining trees as a way of combining data sets for phylogenetic inference, and the desirability of combining gene trees. Taxon 1992, 41:3-10.
19. Ragan MA: Phylogenetic inference based on matrix representation of trees. Molecular Phylogenetics and Evolution 1992, 1:53-58.
20. Swofford D: PAUP*: Phylogenetic analysis using parsimony (*and other methods). Sinauer Assoc., Sunderland, Massachusetts, U.S.A. Version 4.0 beta.
21. Goloboff P: Minority rule supertrees? MRP, compatibility, and minimum flip may display the least frequent groups. Cladistics 2005, 21:282-294.
22. Pisani D, Wilkinson M: MRP, taxonomic congruence and total evidence. Systematic Biology 2002, 51:151-157.
23. Brown DG, Harrower IM: Integer programming approaches to haplotype inference by pure parsimony. IEEE/ACM Trans Comput Biol Bioinformatics 2006, 3(2):141-154.
24. Gunstfeld D: Haplotype inference by pure parsimony. OPIT, Volume 2676 of Lecture Notes in Computer Science SpringerBerlin-Chaatzs RA, Chaves E, Crochemore M 2003, 144-155.
25. Gunstfeld D: The multi-state perfect phylogeny problem with missing and removable data: Solutions via integer-programming and chordal graph theory. RECOMB, Volume 5541 of Lecture Notes in Computer Science SpringerBerlin-Chaatzs RA, Chaves E, Crochemore M 2003, 144-155.
26. Sridhar S, Lann F, Belloch GE, Ravi R, Schwartz R: Mixed integer linear programming for maximum-parsimony phylogeny inference. IEEE/ACM Trans Comput Biol Bioinformatics 2008, 5(3):323-331.
27. Sibley CG, Ahlquist JE: Phylogeny and Classification of Birds: A Study in Phylogeny and Classification of Birds: A Study in Systematic Zoology 1998, 158:965-968.
28. Nunn GB, Stanley SE: Body size effects and rates of cytochrome b evolution in tubenosed seabirds. Molecular Biology and Evolution 1998, 15:1360-1371.
37. Wilkinson M, Pisani D, Cotton JA, Corfe I. Measuring support and finding unsupported relationships in supertrees. *Systematic Biology* 2005, 54(5):823-831.

38. Purvis A. A modification to Baum and Ragan's method for combining phylogenetic trees. *Systematic Biology* 1995, 44:251-255.

doi:10.1186/1748-7188-5-2

Cite this article as: Dong et al. Constructing majority-rule supertrees. *Algorithms for Molecular Biology* 2010 5:2.