Cophylogenetic Reconciliation with ILP

Nicolas Wieseke, Tom Hartmann, Matthias Bernt, and Martin Middendorf

Parallel Computing and Complex Systems Group, Department of Computer Science, Leipzig University, Augustusplatz 10, 04109 Leipzig, Germany

October 28, 2014

Abstract

Whenever species share the same environment they influence each other, which can have an influence on their evolution. Host species and their parasite species are a well known example. A particular challenge is to reconstruct the coevolutionary relations between hosts and parasites from the phylogenetic trees of both groups of species and the observed interactions between the extant species. A common approach is to consider a set of possible cophylogenetic event types with associated costs and to search for a reconciliation, i.e., a mapping from the parasite tree to the host tree, with a minimum sum of the costs of all involved events. In this paper we present an integer linear programming (ILP) approach, called CoRe-ILP, for finding an optimal time consistent cophylogenetic host-parasite reconciliation under the cophylogenetic event model with the events cospeciation, duplication, sorting, host switch, and failure to diverge. Instead of assuming event costs, a simplified model is used, maximizing primarily for cospeciations and secondarily minimizing host switching events. Duplications, sortings, and failure to diverge events are not explicitly scored. Different from existing event based reconciliation methods, CoRe-ILP can use (approximate) phylogenetic branch lengths for filtering possible ancestral host-parasite interactions. Experimentally, it is shown that CoRe-ILP can successfully use branch length information and performs well for biological and simulated data sets. The results of CoRe-ILP are compared with the results of the reconciliation tools Jane 4, Treemap 3b, NOTUNG 2.8 Beta, and Ranger-DTL. Algorithm CoRe-ILP is implemented using IBM ILOG CPLEX\textsuperscript{TM} Optimizer 12.6 and is freely available from \url{http://pacosy.informatik.uni-leipzig.de/core-ilp}.

Keywords: Cophylogeny, tree reconciliation, host-parasite coevolution, integer linear programming, branch length, cospeciation, host switch.

course of evolution of a species is subject to influences from its environment which consists not only of the physical conditions, but also of the coexisting species. The influence between species varies with the type and strength of their mutual dependency. Coevolution describes the common evolution of a set of species. A well known example are pocket gophers and their chewing lice.
parasites. The impact of the mutual influence on the evolution of these species has been evaluated by measuring the congruence between their phylogenetic trees [1]. A particularly interesting problem in coevolution — called cophylogeny reconciliation problem — is the reconstruction of the common history of two groups of species, when the phylogenetic tree of each group and also the interactions between the extant species of both groups are known. It should be noted that cophylogeny reconstruction is not limited to sets of species. It can also be applied to other entities that mutually influence each other during evolution, e.g., genes, species, geographical properties (see, e.g., [2]), or sets of interacting proteins [3].

A common approach for cophylogenetic reconstruction is to apply the parsimony principle. Given a set of possible cophylogenetic events (see Figure 1) one searches for a reconciliation, i.e., a mapping from one phylogenetic tree to the other, such that the minimum number of involved cophylogenetic events is used [4]. Alternatively, each event type can be assigned an associated cost and then the minimum sum of the costs of all involved events is considered. Cospeciation, duplication, lineage sorting (loss), and host switch (horizontal transfer) are the cophylogenetic events that are typically used for such an event based reconciliation of coevolution. They have been used, e.g., for the reconciliation of hosts and their parasites [5] or for gene trees and related species trees [4]. The (decision version of the) cophylogenetic reconciliation problem is NP-complete under the event model considering cospeciations, duplications, losses, and host switches. See [6] for details.

There are several algorithms available for event based reconciliation. The pioneering approach of [7] computes a reconciliation by using the events colonization, exclusion, and cospeciation where a host switch is the combination of a colonization and an exclusion. Most of the other approaches that are reviewed in the following make use of the four events that have been mentioned above.

Algorithm TreeMap is based on the Jungles data structure that represents the set of optimal reconciliations [5, 2]. TreeMap is an exact algorithm for the cophylogenetic reconciliation problem. Another exact algorithm that uses a generalized event model for reconciliations which captures a multitude of cophylogenetic events, including the four common events, has been described recently in [8]. The corresponding tool CoRe-SYM can perform a cophylogenetic analyses for different types of data sets. Beside host-parasite interactions and gene tree/species tree reconciliation, biogeographic data sets or species living in symbiosis can be analyzed. The reconciliation of the latter type of data requires symmetric events that are not available in most of the other reconciliation methods since they are mostly tailored for host-parasite or gene tree/species tree systems. CoRe-SYM integrates an efficient dynamic programming algorithm which might yield reconciliations that imply contradicting timing relations. Therefore, CoRe-SYM also integrates a branch-and-bound algorithm that computes optimal time consistent solutions. Mowgli [9] can compute exact solutions in polynomial time. However, it assumes that divergence times are given and thus considers only a restricted version of the cophylogenetic reconciliation problem.

Other algorithms for the cophylogenetic reconciliation problem have been proposed that either compute optimal solutions under the relaxation that the solutions can be time inconsistent or are heuristics. Tarzan [10], CoRe-PA [11], Ranger-DTL [12], and NOTUNG 2.8 Beta [13] are of the former type. All four have fast reconciliation algorithms (the latter three are based on dynamic pro-
gramming) but do not guaranty time consistent solutions. Jane [14] is a fast heuristic that computes time consistent solutions. Jane uses a genetic algorithm, for determining a node timing, i.e., an order of the nodes of the phylogenetic trees, in combination with a dynamic programming algorithm, that computes a reconciliation for the trees under the determined node order. The algorithms DLCoalRecon [21] and DLCpar [22] are tailored for the analysis of gene/species tree problems and create reconciliations based on the DLCoal model, which includes duplication, loss, and coalescence. Coalescence was previously used for species tree inference from multiple gene trees [23, 24].

For many biological applications additional time information is available. Using this information promises to derive biologically more realistic reconciliations. Therefore, some of the reconciliation tools can integrate such timing information. A time zone model has been used in Tarzan [10] where every node in each of the phylogenetic trees can be assigned to a time zone (in case of the parasite tree a node can also be assigned to an interval of time zones) and only reconciliations are considered where the associated species occur in the same time zone. CoRe-PA [11] and Jane [14] adopted the time zone model of Tarzan [10]. In Jane the model was extended such that an interval of time zones can also be assigned to each node of the host tree. Jane, Mowgli, Ranger-DTL, and NOTUNG 2.8 Beta can use divergence time information in the form of an ordering of the nodes in the phylogenetic trees.

It should be noted that CoRe-PA [11], Jane [14], and NOTUNG 2.8 Beta [13] can handle non-binary phylogenetic trees. Moreover, a method for the automatic estimation of suitable weights for the cophylogenetic events has been addressed in CoRe-PA [11]. It uses the idea that the frequency of the reconstructed events and their costs should be reciprocal.

In [15] the first integer linear programming (ILP) formulation is presented for the reconciliation problem. It can be applied to binary phylogenetic trees and considers the events cospeciation, duplication, loss, and host switch. As stated by the authors of [15] their approach is intended as a proof of concept that is feasible for small data sets only. Trees with 8 leaves can be reconciled within a few minutes, but it takes already several hours for trees with 16 leaves. The strong increase in runtime is due to the cubic number of ILP variables that are used.

In this paper we present an ILP approach and the corresponding tool CoRe-ILP that are based on similar ideas of the ILP formulation of [15]. However, only a quadratic number of variables is used. It is shown that the new approach is a practical method that is suitable for phylogenetic trees of size up to 50 leaves each. CoRe-ILP finds an optimal and time consistent cophylogenetic host-parasite reconciliation under the cophylogenetic event model with cospeciation, duplication, sorting, host switch, and failure to diverge events. Note, that although the presented ILP approach only considers cospeciations and host switches explicitly, it is not restricted to these two types of events. It produces reconciliations containing also duplications, sortings, and failure to diverge events. However, the optimality criterion does only account for cospeciations and host switches.

An advantage of CoRe-ILP is that phylogenetic branch length information can be used for filtering possible ancestral host-parasite interactions that approximately satisfy the branch lengths. CoRe-ILP does not require exact information. Instead, the user can specify the degree of certainty about the reliability
of the branch length information. Branch length information has, so far, been neglected in event based cophylogenetic reconstruction. However, for many biological applications (approximate) branch lengths are available. The timing model of CoRe-ILP can be used, e.g., whenever the molecular clock hypothesis [16] is approximately applicable for the given phylogenetic trees. Furthermore, CoRe-ILP can handle multi-host parasites and non-binary trees.

So far, CoRe-ILP does not consider event costs. Instead, it searches for a reconciliation with a maximum number of cospeciations and, as a second optimization criterion, a minimum number of host switches. Furthermore, duplication, sorting, and failure to diverge events can occur in the reconciliations, but they are not scored in the objective function. It should be noted that the presented ILP formulation can easily be extended to account for event costs, as well as for scoring the number of duplications, sortings, and failure to diverge events. However, as shown in Section 1.2, each scored type of events requires at least a quadratic number of ILP variables and constraints and therefore, increases the runtime.

CoRe-ILP is evaluated experimentally for biological data, as well as for simulated data sets. The resulting reconciliations are compared with those of several other reconciliation tools. In particular, we compare the results with those of Jane 4 [14], Treemap 3b [5, 2], NOTUNG 2.8 Beta [13], and Ranger-DTL [12]. It should be mentioned that a few other reconciliation tools have not been used for comparison for the following reasons: (i) Mougli [9] because it requires fully dated cladograms as input, (ii) CoRe-Sym [8] and the ILP approach presented in [15] because of the exceptionally long runtimes (> 1h) for trees of practical size, and (iii) the reconciliation tools DLCpar [21] and DLCpar [22] because they utilize a disparate event model based on duplication, loss, and coalescence.

1 Methods

1.1 Preliminaries

Let $\mathcal{X}$ be a set of taxa. A phylogenetic tree (on $\mathcal{X}$) is a rooted tree $T_{\mathcal{X}} = (V_{\mathcal{X}}, E_{\mathcal{X}})$ with leaf set $\mathcal{X} \subseteq V_{\mathcal{X}}$ such that each inner vertex $v \in V_{\mathcal{X}}^0 := V_{\mathcal{X}} \setminus \mathcal{X}$ has an outdegree of at least two. The outdegree of a node $v$ is denoted by $\text{deg}(v)$. For $u, v \in V_{\mathcal{X}}$, $u$ is an ancestor of $v$ and $v$ a descendant of $u$, if and only if $u$ lies on the unique path from the root $\rho_{T_{\mathcal{X}}}$ to $v$. The set of ancestors and descendants of a node $v$ is denoted by $\text{anc}(v)$ and $\text{desc}(u)$, respectively. Note that $v \in \text{anc}(v)$ and $u \in \text{desc}(u)$. For each $(u, v) \in E_{\mathcal{X}}$ we call $v$ a (direct) child of $u$ and $u$ the parent of $v$. The set of children of node $u$ is denoted by $\text{child}(u)$. If it is clear from the context we denote the parent of $v$ as $v'$ and the children of $u$ as $u_i$, $i \in [1, \ldots, n]$, where $n$ is the number of children of $u$, i.e., $n = \text{deg}(u)$. The (direct) children of a node are called siblings. The set of all siblings of a node $v$ is denoted by $\text{sibl}(v)$. The subtree $T_{\mathcal{X}}(v)$ for $v \in V_{\mathcal{X}}$ is defined as the induced subgraph of $T_{\mathcal{X}}$ on $\text{desc}(v)$.

We assume that each edge $(u, v) \in E_{\mathcal{X}}$, also denoted by $e_v$, has a branch length $bl(e_v) \in \mathbb{R}^+$. With $T_{\mathcal{X}}$ displaying the evolution of a set of species $\mathcal{X}$ and the inner nodes $V_{\mathcal{X}}^0$ referring to speciation events, $bl(e_v)$ can be interpreted as the (approximate) time span of the existence of species $v$. A time is assigned to each node of $\mathcal{X}$ via a timing function $t : V_{\mathcal{X}} \rightarrow \mathbb{R}^+_0$ such that for all $v \in \mathcal{X}$,
Figure 1: Coevolutionary events that are consistent with the ILP formulation; $h, h^*$ denote host species; $p$ denotes parasite species; $h_i$ and $p_j$ are direct descendants of $h$ and $p$, respectively; (a) to (e) events for nodes with outdegree two; (f) multiple co-occurring cospeciation events for associated non-binary nodes (the shown example is counted as two cospeciations, see Section 1.2 for details).
from different subtrees of parasites and a set $\mathcal{H}$ of hosts, respectively. Then, $T_p$ and $T_h$ are called parasite tree and host tree, respectively. Define a leaf to leaf mapping $\phi(p, h) : \mathcal{P} \times \mathcal{H} \rightarrow \{0,1\}$ representing the observed host-parasite interactions, with $\phi(p, h) = 1$ if $h$ is a host of parasite $p$ and, otherwise, $\phi(p, h) = 0$. The triple $(T_p, T_h, \phi)$ is called tanglegram [5].

For a given tanglegram $(T_p, T_h, \phi)$ a cophylogenetic reconciliation is defined (similar as in [8]) as a set of associations $\mathcal{R} \subset V_p \times V_h$ between nodes of the parasite tree and nodes of the host tree. For a given parasite $p$, an association $(p, h) \in \mathcal{R}$ indicates that $h$ is (one of) the most recent host species that is associated with $p$. If $(p, h) \in \mathcal{R}$ then we call $p$ and $h$ to be associated. Note, that there exist multiple $h^*$ with $(p, h^*) \in \mathcal{R}$.

A cophylogenetic reconciliation $\mathcal{R}$ is called valid if the following holds: (i) $\mathcal{R}$ agrees with the leaf to leaf mapping $\phi$, i.e., $\forall p \in \mathcal{P}, h \in \mathcal{H}: \phi(p, h) = 1 \iff (p, h) \in \mathcal{R}$, (ii) the vertex timing agrees with the associations, i.e., $(p, h) \in \mathcal{R} \Rightarrow t_h > t_p \geq t_h$, (iii) $\mathcal{R}$ is complete, i.e., for all $p \in V_p$ it holds that $\exists h \in V_h$ such that $(p, h) \in \mathcal{R}$, (iv) $\mathcal{R}$ is backtraceable (if $p$ is associated with $h$, then at least one child of $p$ is associated with either $h$ or a node from the subtree rooted at $h$), i.e., for all $(p, h) \in \mathcal{R}$ it holds that $\text{child}(p) = \emptyset$ or $\exists p_i \in \text{child}(p), h \in \text{desc}(h)$ such that $(p_i, h) \in \mathcal{R}$. Note, that condition (ii) implies, that if there are multiple $h^*$ with $(p, h^*) \in \mathcal{R}$, these $h^*$ can not be ancestors or descendants of each other, but they have to belong to disjoint subtrees of $T_h$.

In the following, we assume $p \in V_p$ and $h \in V_h$. A parasite $p$ is exclusively subtree-associated (among its siblings) with a node of a subtree $T_h(h)$, if it holds that (i) $\exists h \in \text{desc}(h)$ such that $(p, h) \in \mathcal{R}$, (ii) for all $p^* \in \text{sibl}(p)$ it holds that $\nexists h \in \text{desc}(h)$ such that $(p^*, h) \in \mathcal{R}$, and (iii) for all $h^* \in \text{sibl}(h)$ it holds that $\nexists h^* \in \text{desc}(h^*)$ such that $(p, h^*) \in \mathcal{R}$.

In this paper we use a (mostly standard) cophylogenetic event model as shown in Figure 1. Only multiple cospeciation events are not commonly used. They are introduced here as a simultaneous speciation of a host $h$ and its associated parasite $p$, such that different children of $p$ are associated with nodes from different subtrees of $h$. Formally, a cospeciation between $p$ and $h$ occurs, if the following holds: (i) the speciations occur simultaneously, i.e., $t_p = t_h$, (ii) $(p, h) \in \mathcal{R}$, (iii) no child $p_i$ of $p$ is associated with $h$, (iv) $p$ and $h$ are both inner nodes, and (v) at least two different children $p_i$ and $p_k$ of $p$ are exclusively subtree-associated with nodes from different subtrees $T_h(h_j)$ and $T_h(h_l)$ of $h$.

Precisely, the number of counted cospeciations for two nodes $p$ and $h$ where a cospeciations occurs, is equal to the number of exclusively subtree-associated children (as defined in condition (v) of the definition of cospeciation) minus one. This is motivated by the fact that the number of exclusive subtree-associations minus one is the minimum number of cospeciations which can always be ensure in a reconciliation of a fully resolved (binary) version of the parasite tree with the non-binary host tree. Hence, there always exists a binary parasite tree and a time consistent reconciliation with at least that many cospeciations at this
A host switch is an association of \( p \) and \( h \), where the parent \( p' \) of \( p \) has no association with any ancestor of \( h \), i.e., \((p, h) \in R\) and \( \exists \hat{h} \in \text{anc}(h) \) such that \((p', \hat{h}) \in R\).

The cophylogenetic (host-parasite tree) reconciliation problem is: Given a tanglegram \((T_P, T_H, \phi)\), find a reconciliation \( R \) with a maximum number of cospeciations (primary criterion), a minimum number of host switches (secondary criterion), and a minimal number of associations (tertiary criterion).

The tertiary criterion is used to ensure that in the case of solutions with an equal number of cospeciations and host switches, the reconciliation with fewer multi-host events is chosen. Note, that we did not restrict a reconciliation to associate a parasite with only one host. Hence, multi-host parasites are covered by this definition and the integer linear programming approach that is introduced in the following.

### 1.2 ILP Formulation

In this section we present an integer linear programming (ILP) formulation for solving the cophylogenetic host-parasite tree reconciliation problem. Table 1 summarizes the definitions of ILP variables and constants, and provides a key to the notation used in this section. The proofs of correctness and completeness for each inequation are given in the Supplemental Material.

<table>
<thead>
<tr>
<th>Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_v )</td>
<td>scale factor for branch length of edge ( e_v ).</td>
</tr>
<tr>
<td>( t_v )</td>
<td>speciation time of ( v \in (V_P \cup V_H) ).</td>
</tr>
<tr>
<td>( t_{ph} )</td>
<td>( t_{ph} = 1 ) iff ( p ) and ( h ) have the same timing, i.e., ( t_p = t_h ).</td>
</tr>
<tr>
<td>( a_{ph} )</td>
<td>( a_{ph} = 1 ) iff ( p ) and ( h ) are associated, i.e., ((p, h) \in R).</td>
</tr>
<tr>
<td>( d_{ph} )</td>
<td>( d_{ph} = 1 ) iff ( p ) is associated with a node of subtree ( T_X(h) ).</td>
</tr>
<tr>
<td>( e_{ph} )</td>
<td>( e_{ph} = 1 ) iff ( p ) is exclusively subtree-associated with a node of subtree ( T_X(h) ).</td>
</tr>
<tr>
<td>( c_{ph} )</td>
<td>Number of cospeciations occurring between ( p ) and ( h ).</td>
</tr>
<tr>
<td>( s_{ph} )</td>
<td>( s_{ph} = 1 ) iff ( p ) performed a host switch to associate with ( h ).</td>
</tr>
</tbody>
</table>

First, we define the numeric variables \( t_v \) for all \( v \in (V_P \cup V_H) \) to represent the point in time at which the speciation of \( v \) occurred. The speciation time is bounded by the following constraint.

\[
0 \leq t_v \leq N_1
\]

with \( N_1 = N_U(\max\{\text{height}(T_P), \text{height}(T_H)\}) \) being the maximum possible length of a path in the subtree rooted at \( v \), and scaled by \( N_U \), the upper bound for the branch length scale factor. For all extant species \( v \in (P \cup H) \) (i.e., leaves in the trees) the speciation time is constrained to be zero.

\[
t_v = 0
\]
For each edge \( e_v \in (E_P \cup E_H) \) a numeric variable \( f_v \) is defined, indicating how much a given branch length has to be scaled, in order to be in accordance with the speciation times. The following constraint bounds this scale factor.

\[
N_L \leq f_v \leq N_U, \tag{3}
\]

where \( N_L \) and \( N_U \) are parameters, that are chosen with respect to the reliability of the given branch lengths from \( T_P \) and \( T_H \). For an edge \( e_v = (v', v) \in (E_P \cup E_H) \) the speciation time \( t_{v'} \) is constrained by \( t_v \), the branch length \( bl(v) \), and the positive scale factor \( f_v \).

\[
t_{v'} - t_v - f_v bl(v) = 0 \tag{4}
\]

Note that this construction induces \( t_{v'} > t_v \).

In the following we define several binary variables. All these variables can only have values 0 or 1, which is ensured by a constraint \( 0 \leq \text{var} \leq 1 \).

For any two \( p \in V_P, h \in V_H \) a binary variable \( a_{ph} \) is defined, that denotes, whether or not a parasite \( p \) is associated with a host \( h \) at the time of its speciation \( t_p \). If \( p \) is associated with \( h \) then \( a_{ph} = 1 \) and otherwise \( a_{ph} = 0 \). To obtain valid reconciliations, each variable \( a_{ph} \) has to be in accordance with the given leaf to leaf mapping \( \phi \). Thus, we define for all \( p \in P \) and \( h \in H \):

\[
a_{ph} = \phi(p, h). \tag{5}
\]

To obtain a complete reconciliation, it is required that every \( p \in V_P \) is associated with at least one \( h \in V_H \). This constraint is ensured for all \( p \in V_P \) by the inequations:

\[
\sum_{h \in V_H} a_{ph} \geq 1. \tag{6}
\]

Note, that by changing Constraint (6) to \( \sum_{h \in V_H} a_{ph} = 1 \), the reconciliations can be restricted to consider single-host parasites only.

A reconciliation has to comply with the vertex timing, i.e., there has to be an overlap between the time spans of the existence of two associated species \( p, h \) and the speciation of \( p \) has to occur before the speciation of \( h \) (i.e., \( t_{h'} > t_p \geq t_h \)). For all \( p \in V_P \) and \( h \in V_H \) Constraints (7) and (8) ensure this.

\[
-N_2 (1 - a_{ph}) + t_p - t_{h'} \leq -\epsilon \tag{7}
\]

\[
-N_2 (1 - a_{ph}) + t_h - t_p \leq 0 \tag{8}
\]

with \( \epsilon \in (0, 1) \) and \( N_2 > N_U \ast \max\{\text{height}(T_P), \text{height}(T_H)\} \).

Before introducing the constraints that ensure backtraceability, we define binary variables \( d_{ph} \) for any two \( p \in V_P, h \in V_H \). The variables indicate, whether or not \( p \) is associated with a descendant of \( h \), i.e., \( d_{ph} = 1 \iff a_{ph} = 1 \lor \sum_{h_i \in \text{child}(h)} d_{ph_i} \geq 1 \). This is achieved by the following two constraints.

\[
d_{ph} \leq a_{ph} + \sum_{h_i \in \text{child}(h)} d_{ph_i} \tag{9}
\]

\[
N_3 d_{ph} \geq a_{ph} + \sum_{h_i \in \text{child}(h)} d_{ph_i}, \tag{10}
\]
with $N_3 > \text{deg}(h)$. The following constraint ensures backtraceability, i.e., $a_{ph} = 1 \Rightarrow \sum_{p_k \in \text{child}(p)} d_{p_k h} \geq 1$.

$$a_{ph} \leq \sum_{p_k \in \text{child}(p)} d_{p_k h} \quad (11)$$

For counting the number of cospeciations, binary variables $e_{ph}$ and integer variables $c_{ph}$ are defined for any two $p \in V_P$, $h \in V_H$. Variables $e_{ph}$ indicate an exclusive subtree-association of $p$ with a node from subtree $T_H(h)$ and $c_{ph}$ counts the number of cospeciations occurring between $p$ and $h$.

A parasite node $p$ is exclusively subtree-associated with a node from the subtree rooted at $h$, denoted by $e_{ph} = 1$, if $p$ is associated with $h$ or a descendant of $h$, all siblings of $p$ are not associated with a node from the subtree rooted at $h$, and in turn, $p$ is not associated with nodes from the subtrees rooted at siblings of $h$. Formally, $e_{ph} = 1 \iff d_{ph} = 1 \land \sum_{p_k \in \text{sibl}(p)} d_{p_k h} = 0 \land \sum_{h_l \in \text{sibl}(h)} d_{ph_l} = 0$.

This equivalence can be expressed with the following constraints.

$$1 \leq N_4 e_{ph} + (1 - d_{ph}) + \sum_{p_k \in \text{sibl}(p)} d_{p_k h} + \sum_{h_l \in \text{sibl}(h)} d_{ph_l} \leq N_4, \quad (12)$$

where $N_4 > \text{deg}(p) + \text{deg}(h) + 1$.

The number of cospeciations between $p$ and $h$ is counted by $c_{ph}$. Thereby, $c_{ph}$ is zero, if $p$ and $h$ does not cospeciate. If $p$ and $h$ cospeciate, then $c_{ph}$ gives the number of cospeciation events occurring between $p$ and $h$, i.e., the number of exclusive child subtree-associations minus 1. Note, that $c_{ph}$ must be positive. Therefore, we restrict the variables $c_{ph}$ by

$$c_{ph} \geq 0. \quad (13)$$

If there occurs a cospeciation between $p$ and $h$, then $p$ and $h$ have to be associated, i.e., $c_{ph} \geq 1 \Rightarrow a_{ph} = 1$, and if there exists an association $a_{ph} = 1$ between $h$ and a child $p_k$ of $p$, then $c_{ph} = 0$. This can be expressed as

$$N_5 a_{ph} - c_{ph} \geq 0 \quad (14)$$
$$N_6 (1 - a_{ph}) - c_{ph} \geq 0 \quad (15)$$

for every child $p_k \in \text{child}(p)$, where $N_5, N_6 > \min\{\text{deg}(p), \text{deg}(h)\}$. Additionally, we have to ensure that $c_{ph} = 0$ if either $p$ or $h$ is a leaf. For $p \in \mathcal{P}, h \in \mathcal{H}$ we add the constraints

$$c_{ph} = 0. \quad (16)$$

If $p$ and $h$ are associated and there is no association between a child $p_k$ of $p$ and $h$, then the number of cospeciations between $p$ and $h$ is equal to the number of exclusive child subtree-associations minus one (but never less than zero). Note, that this is a lower bound for the minimum number of cospeciations that can be achieved, when considering a resolved (binary) tree. With the help of additional indicator variables it is possible to restrict $c_{ph}$ directly with one ILP constraint per combination of host and parasite. However, to keep the number of variables...
small, the following three constraints are used instead.

\[
N_7 (1 - a_{ph}) + c_{ph} - \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_k h_l} + 1 + N_7 \sum_{p_k \in \text{child}(p)} a_{p_k h} \geq 0 \tag{17}
\]

\[
N_7 (1 - a_{ph}) + \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_k h_l} - c_{ph} + N_7 \sum_{p_k \in \text{child}(p)} a_{p_k h} \geq 0 \tag{18}
\]

\[
N_8 (1 - a_{ph}) + \left( \epsilon - \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_k h_l} \right) + \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_k h_l} - c_{ph} + N_8 \sum_{p_k \in \text{child}(p)} a_{p_k h} \geq 0 \tag{19}
\]

with \(N_7 > \text{deg}(p) \ast \text{deg}(h), N_8 > \text{deg}(p) \ast \text{deg}(h) + 1\) and \(\epsilon \in (0, 1)\).

Constraint (17) ensures that if \(a_{ph} = 1\) and all \(a_{p_k h} = 0\), \(p_k \in \text{child}(p)\), then \(c_{ph} \geq \left( \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_k h_l} \right) - 1\). By Constraint (18) it is guaranteed, that \(c_{ph} \leq \left( \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_k h_l} \right)\). Finally, by Constraint (19) it holds that that if (i) \(p\) and \(h\) are associated, (ii) there is no association between a child \(p_k\) of \(p\) and \(h\), and (iii) the number of exclusive child subtree-associations is greater or equal to two, then \(c_{ph} = \left( \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_k h_l} \right) - 1\). Otherwise, no cospeciation occurs between \(p\) and \(h\) and hence, \(c_{ph} = 0\).

If a cospeciation occurred between \(p\) and \(h\), then the speciation times \(t_p\) and \(t_h\) have to be equal. Therefore, we define binary variables \(t_{ph}\) for any two \(p \in V_p\) and \(h \in V_h\). For these variables it holds that \(c_{ph} \geq 1 \Leftrightarrow t_{ph} = 1\), which is ensured by Constraints (20) and (21), and \(t_{ph} = 1 \Rightarrow t_p = t_h\) is implemented by Constraints (22) and (23).

\[
N_9 t_{ph} \geq c_{ph} \tag{20}
\]

\[
c_{ph} \geq t_{ph} \tag{21}
\]

\[
-N_{10} (1 - t_{ph}) + t_p - t_h \leq 0 \tag{22}
\]

\[
-N_{10} (1 - t_{ph}) + t_h - t_p \leq 0, \tag{23}
\]

with \(N_9 > \text{deg}(p) \ast \text{deg}(h)\) and \(N_{10} > N_U \ast \max\{\text{height}(T_p), \text{height}(T_h)\}\). Note that Inequality (23) is defined for the sake of completeness only, since it is implicitly given by the combination of Constraints (7,15,21).

To indicate the occurrence of a host switch between \(p\) and \(h\), we define a binary variable \(s_{ph}\) for any two \(p \in V_p, h \in V_h\). Variable \(s_{ph} = 1\) if \(p\) is associated with \(h\), and \(p\) switched to \(h\) (or an ancestor of \(h\)) before speciating. Precisely, if \(a_{ph} = 1 \land \hat{h} \in \text{anc}(h)\) such that \(a_{p' h} = 1\), then \(s_{ph} = 1\). Furthermore, if \(p\) and \(h\) are not associated and thus \(a_{ph} = 0\), then no host switch occurs and hence \(s_{ph} = 0\). Finally, if there exists an ancestor \(\hat{h} \in \text{anc}(h)\) which is associated with \(p'\) (the parent of \(p\)), then no host switch occurred and thus \(s_{ph} = 0\). Note, that in this approach a host switch is counted whenever a parasite \(p\) colonizes a new
We search for cophylogenetic reconciliations which are optimal with respect to the following three ordered criteria: (i) maximum number of cospeciations, (ii) minimum number of host switches, and (iii) minimum number of associations. This is expressed by the following objective function.

\[
\min \left( -N_{11}^2 \sum_{p \in V_P} \sum_{h \in V_H} c_{ph} + N_{11} \sum_{p \in V_P} \sum_{h \in V_H} s_{ph} + \sum_{p \in V_P} \sum_{h \in V_H} a_{ph} \right),
\]

(27)

where \(N_{11}\) is chosen such that \(N_{11} > |V_P| * |V_H| * maxdeg(V_P \cup V_H)\). Note that the minimum number of associations is selected as third criterion, to prefer reconciliations with fewer multi-host parasites, if they have the same number of cospeciations and host switches.

In summary, with \(n = |V_P|\) and \(m = |V_H|\) being the number of nodes of \(T_P\) and \(T_H\), respectively, the ILP formulation uses \(O(nm)\) variables and \(O(nm^2)\) constraints.

As mentioned above, it would be easily possible to use event costs \(C_{cosp}\) and \(C_{switch}\) instead of simply maximizing the number of cospeciations and minimizing the number of host switches. Therefore, the objective function has to be changed to

\[
\min \left( C_{cosp} \sum_{p \in V_P} \sum_{h \in V_H} c_{ph} + C_{switch} \sum_{p \in V_P} \sum_{h \in V_H} s_{ph} \right).
\]

(28)

For reasonable choices of \(C_{cosp}\) and \(C_{switch}\) the reconstructions for Objective (27) (\(obj1\)) and (28) (\(obj2\)) are similar – bounded by a specific degree – with respect to the number of cospeciations and host switches. As duplications and sortings are not scored (i.e., scored zero) the cospeciation cost \(C_{cosp}\) have to be negative. Otherwise, in a cost minimal reconciliation, a cospeciation is replaced by a duplication, followed by some sorting events, whenever possible. Furthermore, \(C_{switch}\) has to be positive, to avoid dispensable switching events. Note that this is consistent with the assumption that cospeciation is the predominant event in host-parasite cophylogenetic systems and host switch events are comparatively rare [5]. Thereby, cospeciations should have the lowest cost value among all events. Clearly, the two optimization criteria (\(obj1\) and \(obj2\)) can yield distinct reconciliations, but for \(C_{cosp} < 0\) and \(C_{switch} > 0\), they will be very similar with respect to the number of cospeciations and host switches. If they differ, then there are fewer cospeciations, as well as fewer host switches in the \(obj2\)-minimal reconciliations, compared to the \(obj1\)-minimal reconciliations. Furthermore, an \(obj2\)-minimal reconciliation that differs by \(x\) cospeciations (compared to an \(obj1\)-minimal reconciliation), differs at least by \(x \times |C_{cosp}/C_{switch}|\) switching events. Otherwise, the reconciliation would be dominated by the \(obj1\)-minimal reconciliation (under the \(obj2\) scoring scheme).
The ratio $|C_{\text{cosp}}/C_{\text{switch}}|$ indicates how many host switching events have to compensate for one cospeciation event. Since frequencies (and thereby the costs) of the different cophylogenetic events are generally unknown, the simplified cost function that is advocated here, appears to be advantageous, given the provable and empirically verified similarity between solutions obtained for the two objectives.

Although we did not implement duplications, sortings, and failure to diverge, these events can be treated along the lines of our framework. In case of duplications, binary variables $d_{ph}$ have to be defined, with $d_{ph} = 1$ indicating that $p$ and $h$ are associated, and there is either a child $p_i$ of $p$ being associated with the same $h$, or there are at least two children $p_i$ and $p_j$ of $p$, being associated descendants of the same child $h_k$ of $h$. Lineage sorting can be considered using binary variables $l_{ph}$, with $l_{ph} = 1$ indicating that the parent $p'$ of $p$ is associated with an ancestor $\hat{h} \in \text{anc}(h)$ (including $h$) and $p$ is associated with a descendant $\bar{h} \in \text{desc}(h) \setminus \{h\}$. For failure to diverge events binary variables $f_{ph} = 1$ indicate, that the parent $p'$ of $p$ is associated with an ancestor $\hat{h} \in \text{anc}(h)$ (including $h$), and $p$ is associated with at least two nodes $\bar{h} \in \text{desc}(h_i), \bar{h} \in \text{desc}(h_j), i \neq j$, being descendants of different children $h_i$ and $h_j$ of $h$.

These three types of events are not scored within the presented ILP formulation. Nevertheless, duplications, sortings, and failure to diverge events also occur in the reconciliations, and the number of occurrences can be extracted from the associations in a post processing step, using the aforementioned ideas.

1.3 Biological Data, Simulated Data, and Experiments

As a biological example we use pocket gophers ($\text{Geomys}$) and their associated chewing lice ($\text{Geomydoecus}$) that were analyzed first in [1]. The corresponding data set is often considered as a reference for cophylogeny analyses, e.g., in [17], which is also the source of the (non-binary) branch length annotated phylogenetic trees, that are used in this study. The reconciliations have been computed with branch length factors $f_i \in [0, 2.5]$ and objective function (27).

Simulated tanglegrams for the experimental part were generated according to [18]. The simulation starts with a parasite tree and a host tree, each consisting of a single node, assuming an interaction between the two species. Repeatedly a node, which is selected according to the age branching model [19] from one (uniform randomly chosen) of the two trees, is speciated. This is done until the total number of leaves, i.e., the sum of host and parasite leaves, exceeds 50. If the node is a parasite node, a switching event is performed with probability $p_{sw}$ and a duplication otherwise. In case a host node was selected, each associated parasite performs a cospeciation with probability $p_{co}$ and a sorting otherwise. Tanglegrams were generated with $p_{co}, p_{sw} \in \{0.25, 0.5, 0.75\}$. Each node gets the time assigned, at which it performed a speciation, cospeciation, duplication, or host switching event. Starting from 0, the time is increased by one, whenever a new node gets selected. The time of all leaves is set to the maximum time value that occurs at an inner nodes in one of the trees plus one. For generating a data set with reliable and less reliable branch lengths, branch lengths are computed by multiplying the node time differences with a random number chosen from $[0.5, 2]$ and $[0.25, 4]$, respectively. For the reconciliation we use the corresponding interval as branch length factor, i.e., $f_i \in [0.5, 2]$ or $f_i \in [0.25, 4]$. For each
combination of parameters $p_{co}$ and $p_{sw}$. 100 repetitions are performed, resulting in 900 simulations in total.

The computed reconciliations are evaluated based on the precision of the reconstructed host-parasite associations of all inner vertices. Therefore, let $R_{Sim}$ and $R_{Reco}$ be the associations of all inner vertices obtained from the simulation and reconciliation, respectively. The precision is then defined as $|R_{Reco} \cap R_{Sim}| / |R_{Reco}|$. Note that in the absence of multi-host events each parasite node is associated with exactly one host. Therefore, $|R_{Reco}| = |R_{Sim}|$ which results in equal values for precision and recall. This obviously holds for tools that do not support failure to diverge events. However, for the simulated data sets, none of the evaluated tools showed reconciliations with failure to diverge events, as the simulated data did not include parasite leaves, being associated with more than one host leaf.

Reconciliations are computed with an implementation of the ILP approach and compared with the results of Jane 4 [14], TreeMap 3b [2], NOTUNG 2.8 Beta [13], and Ranger-DTL [12]. For all tools the same simulated data sets were reconciled using the respective default parameters.

Note that TreeMap 3b does not use a single cost model but tries to find all optimal solutions for any possible cost model. From this solutions we select a random solution, i.e., one of the most parsimonious reconciliations constructed without any timing information and independently from a certain cost model.

To evaluate the runtime of data sets with larger trees, 180 data sets of size 100 and 150 are generated and reconciled using the ILP approach with reliable (CI1) and less reliable (CI2) branch lengths (ten data sets for each combination of parameters $p_{co}, p_{sw} \in \{0.25, 0.5, 0.75\}$ and size 100 and 150). Note that the size of these data sets represents the sum of leaves from the host and parasite tree, resulting in trees with approximately 50, respectively 75, leaves each. All computations were performed on 8 Dual-Core AMD Opteron™ Processors with 2.6 GHz.

2 Results and Discussion

For the data set of pocket gophers and their chewing lice CoRe-ILP reconstructed eleven cospeciations, two host switches, three duplications, and 13 sorting events. Figure 2 shows one out of four optimal reconciliations (the remaining solutions are shown in Figures S1 to S3 in the Supplemental Material). When computing the reconciliation without constraining the branch length factors, optimal reconciliations contain eleven cospeciations, no host switches, five duplications, and 20 sorting events. However, four parasites are associated with the host root, and there are parasites with up to five successive sorting events, which appears to be unrealistic. In comparison to [20] CoRe-ILP obtained one cospeciation and one host switch more, as well as seven sortings and two duplications fewer. Note that these differences result, to some extent, from differences in the phylogenetic trees used in [20]. In accordance with our solution the most recent ancestor of Geomydidae lice interacted with the most recent ancestor of Cratogeomys, Orthogeomys, Pappogeomys, Zygogeomys, and Geomys pocket gophers and there was a cospeciation at the origin of Geomyidae and Trichodectidae.

The results for the simulated data sets show, with the exception of the results
Figure 2: An optimal reconciliation for the pocket gophers (black tree) and their associated chewing lice (grey tree) obtained by CoRe-ILP, using $f_v \in [0.25, 4.0]$. Pentagon: multiple cospeciation, stars: host switches, small black circles: sorting events, squares: duplication events, and big grey circles: cospeciations.
Figure 3: Precision (top) and runtimes (bottom) for all combinations of $p_{co}, p_{sw} \in \{0.25, 0.50, 0.75\}$ (see also Supplemental Material) computed with CoRe-ILP with reliable (CI1) and less reliable (CI2) branch lengths, Jane 4 (J4), Treemap 3b (T3b), and the two algorithms that do not guaranty time consistent solutions NOTUNG 2.8 Beta (N2.8), and Ranger-DTL (RDTL). Each analysis was repeated 100 times.
from TreeMap, that all methods yield reconciliations with similar precision (see Figures 3 and S4 to S6 in the Supplemental Material). TreeMap returns worse results presumably because we selected a random reconciliation out of its highly variable result set. All tools return results of higher precision for data sets that are generated with a higher probability of cospeciations and a lower probability of switches. From the tools that guarantee time consistent solutions CoRe-ILP obtained the highest precision values.

Despite being an exact algorithm, CoRe-ILP is clearly faster than the heuristic algorithm Jane when using reliable branch length factors. NOTUNG and Ranger-DTL, i.e., the two programs that do not use timing information, are the fastest algorithms and produce results of high precision. However, both algorithms do not ensure time consistent solutions. In fact, in many cases time inconsistent solutions have been produced. From the 900 data sets 13% of the solutions of Ranger-DTL and 16% of the solutions of NOTUNG are not time consistent. In general, the amount of time inconsistent solutions is larger for data sets, generated with a smaller cospeciation probability and a larger host switching probability. The 100 data sets generated with the lowest cospeciation probability \((p_{co} = 0.25)\) and the highest host switching probability \((p_{sw} = 0.75)\) resulted in 29% and 37% of time inconsistent solutions for Ranger-DTL and NOTUNG, respectively.

As expected, using less reliable branch length factors has a negative effect on the precision of the reconstructions from CoRe-ILP (see Figures 3 and S4 to S6 in the Supplemental Material). Furthermore, the less reliable branch length factors result in an increased runtime of CoRe-ILP, which is explained by the ILP being less constrained (see Figures 3 and S7 to S9 in the Supplemental Material).

For the 900 data sets of size 50 CoRe-ILP has a mean runtime of three and six seconds for the reliable (CI1) and less reliable (CI2) data sets, respectively (see Figures 3 and S7 to S9 in the Supplemental Material). The mean runtime for the 90 data sets of size 100 are 24 and 190 seconds, respectively. There is a single outlier among the less reliable data sets with a runtime of more than one hour (10,435 seconds). The computations for the data sets of size 150 result in a mean runtime of 139 (CI1) and 1385 seconds (CI2). While all reliable data sets could be solved in less than 45 minutes, 40% of the less reliable data sets have a runtime of more than one hour, including three with more than one day.

3 Conclusion

This paper introduced an ILP formulation for the cophylogenetic tree reconciliation problem that computes reconciliations with a maximum number of cospeciations that has a minimum number of host switching events. The use of this simple model allows for an efficient ILP formulation with a quadratic number of variables. Furthermore, it does not require the specification of cost parameters. The corresponding program, called CoRe-ILP, has a runtime that is similar or faster than other state of the are programs that ensure time consistent solutions. Hence, CoRe-ILP is feasible for medium size trees (e.g., 50 leaves each).

Different from other event based reconciliation tools, CoRe-ILP can use branch length information to obtain more meaningful results. It supports data
sets with multi-host parasites and non-binary trees. **CoRe-ILP** produces optimal and time consistent solutions. The only required parameters are the lower and upper bounds for the branch length factors. However, these factors can be estimated with respect to the applicability of the molecular clock hypothesis for a certain data set. For the tested problem instances, the empirical results show that from the tools that guarantee time consistent solutions, **CoRe-ILP** and **Jane** return results with the highest precision, with **CoRe-ILP** being slightly faster. It was shown that the use of available branch length information increases the precision of the reconstructed coevolutionary histories, and hence, overcomes the shortcomings of the limited optimality criterion.

By introducing new variables it is possible to account for events such as sorting or duplication, or any of the generalized events discussed in [8]. Thus, future versions will be able to handle, e.g., symbiotic systems with symmetric events.

**Acknowledgments**

This work was funded by the German Research Foundation (DFG) (Proj. No. MI439/14-1).

**References**


Cophylogenetic Reconciliation with ILP: Supplemental Material

Nicolas Wieseke, Tom Hartmann, Matthias Bernt, and Martin Middendorf

Parallel Computing and Complex Systems Group, Department of Computer Science, Leipzig University, Augustusplatz 10, 04109 Leipzig, Germany

S1 Supplemental Proofs

Constraint:

\[ \sum_{h \in V_H} a_{ph} \geq 1 \]  

(6)

Proof. By considering one \( p \in V_P \) the sum over all hosts is at least one. Hence there exist at least one \( a_{ph}^* = 1 \). Thus \( h^* \in V_H \) is associated with \( p \). This is satisfied for every \( p \in V_P \). \( \square \)

Constraints:

\[ -N_2 (1 - a_{ph}) + t_h - t_p \leq 0 \]  

(7)

\[ -N_2 (1 - a_{ph}) + t_p - t_{h'} \leq -\epsilon, \]  

(8)

where \( \epsilon \in (0, 1) \) and \( N_2 > N_U \ast \{\text{max}\{\text{height}(T_P), \text{height}(T_H)\}\} \).

Proof. Let us assume \( \epsilon \in (0, 1) \), \( h' \in V_H \) is a parent of \( h \in V_H \) and \( p \in V_P \). If \( a_{ph} = 1 \) then from inequation (7) we obtain: \( t_h - t_p \leq 0 \) which is equivalent to \( t_h \leq t_p \). Inequation (8) yields \( t_p - t_{h'} \leq -\epsilon \) and it follows that \( t_p < t_{h'} \). These two inequations ensure the existence of a time overlap. Regarding the case \( a_{ph} = 0 \) it follows that \( -N_2 + t_h - t_p \leq 0 \) and \( -N_2 - t_p + t_{h'} \leq -\epsilon \) which is satisfied for \( N_2 \geq \text{max}\{t_h - t_p, t_p - t_{h'} + \epsilon\} \). This has to hold for all \( p \) and \( h \) with \( a_{ph} = 1 \). Thus it is easy to derive that \( N_2 \geq N_U \ast \{\text{height}(T_P), \text{height}(T_H)\} \geq \text{max}\{t_h - t_p, t_p - t_{h'} + \epsilon\} \) achieves this condition. Consequently it concludes the proposition. \( \square \)

Constraints:

\[ d_{ph} \leq a_{ph} + \sum_{h_i \in \text{child}(h)} d_{phi} \]  

(9)

\[ N_3 d_{ph} \geq a_{ph} + \sum_{h_i \in \text{child}(h)} d_{phi} \]  

(10)
where $N_3 > \text{deg}(h)$.

Proof. For reiteration, the variables $a_{ph}, d_{ph}$ are binary. We have to prove the following three implications:

(i) $d_{ph} = 1 \Rightarrow a_{ph} + \sum_{h_l \in \text{child}(h)} d_{ph_l} \geq 1$ (9), (10)

(ii) $a_{ph} + \sum_{h_l \in \text{child}(h)} d_{ph_l} \geq 1 \Rightarrow d_{ph} = 1$

(iii) $d_{ph} = 0 \Rightarrow a_{ph} = d_{ph_l} = 0 \forall h_l \in \text{child}(h)$.

Case (i): Let $d_{ph} = 1$. Then

$$a_{ph} + \sum_{h_l \in \text{child}(h)} d_{ph_l} \geq 1$$

follows from (9) and

$$N_3 \geq a_{ph} + \sum_{h_l \in \text{child}(h)} d_{ph_l} \geq 1$$

from (10). This is is not an additional restriction for

$$N_3 \geq \text{deg}(h) + 1 \geq a_{ph} + \sum_{h_l \in \text{child}(h)} d_{ph_l} \leq \text{deg}(h)$$

and thus

$$N_3 > \text{deg}(h).$$

Case (ii): Now assume that

$$a_{ph} + \sum_{h_l \in \text{child}(h)} d_{ph_l} \geq 1$$

and hence there exists either child of $h$, which implies $\text{deg}(h) > 0$, or $h$ is a leaf and $a_{ph} \geq 1$. With this premise (9) yields $d_{ph} \leq 1$. Note that $d_{ph}$ can still be 1 or 0. Inequation (10) bounds $d_{ph}$ because it concludes

$$N_3 d_{ph} \geq a_{ph} + \sum_{h_l \in \text{child}(h)} d_{ph_l} \geq 1.$$ 

By choosing $N_3 > \text{deg}(h) \geq 0$ it follows

$$d_{ph} \geq \frac{1}{N_3} \in (0,1)$$

and thus

$$d_{ph} > 0 \Rightarrow d_{ph} \in \{0,1\}$$

and

$$d_{ph} = 1.$$
Case (iii): Let \( d_{ph} = 0 \). Then (9) yields
\[
0 \leq a_{ph} + \sum_{h_l \in \text{child}(h)} d_{ph_l}.
\]
From (10) follows
\[
0 \geq a_{ph} + \sum_{h_l \in \text{child}(h)} d_{ph_l},
\]
and thus
\[
0 = a_{ph} + \sum_{h_l \in \text{child}(h)} d_{ph_l}.
\]
Note, that \( a_{ph} \geq 0 \) and \( d_{ph_l} \geq 0 \). Hence \( a_{ph} = d_{ph_l} = 0 \) \( \forall h_l \in \text{child}(h) \) is the only possibility to ensure this equation.

Constraint:
\[
a_{ph} \leq \sum_{p_k \in \text{child}(p)} d_{p_k h}
\] (11)

Proof. In the first case we assume \( a_{ph} = 1 \). It follows that
\[
1 \leq \sum_{p_k \in \text{child}(p)} d_{p_k h}.
\]
Thus there exists at least one \( d_{p_k h} \) with \( d_{p_k h} = 1 \). Consider the second case \( a_{ph} = 0 \). It implies
\[
0 \leq \sum_{p_k \in \text{child}(p)} d_{p_k h},
\]
This is not an additional restriction because variable \( d_{p_k h} \) is binary.

Constraint:
\[
1 \leq N_4 e_{ph} + (1 - d_{ph}) + \sum_{p_k \in \text{sibl}(p)} d_{p_k h} + \sum_{h_l \in \text{sibl}(h)} d_{ph_l} \leq N_4,
\] (12)
where \( N_4 > \deg(p) + \deg(h) + 1 \).

Proof. We have to prove that (12) holds for the following three implications.

(i) \( e_{ph} = 1 \Rightarrow d_{ph} = 1 \), \( \sum_{p_k \in \text{sibl}(p)} d_{p_k h} + \sum_{h_l \in \text{sibl}(h)} d_{ph_l} = 0 \)

(ii) \( d_{ph} = 1 \), \( \sum_{p_k \in \text{sibl}(p)} d_{p_k h} + \sum_{h_l \in \text{sibl}(h)} d_{ph_l} = 0 \Rightarrow e_{ph} = 1 \)

(iii) \( \sum_{p_k \in \text{sibl}(p)} d_{p_k h} + \sum_{h_l \in \text{sibl}(h)} d_{ph_l} + (1 - d_{ph}) \geq 1 \Rightarrow e_{ph} = 0 \).
Case (i): Let us assume $e_{ph} = 1$. Then (12) yields
\[ 1 \leq N_4 + (1 - d_{ph}) + \sum_{p_k \in \text{sibl}(p)} d_{p_k h} + \sum_{h_l \in \text{sibl}(h)} d_{ph_l} \leq N_4. \]
The left inequation holds for every $N_4 > 1$. The right inequation holds if $1 - d_{ph} = 0 \iff d_{ph} = 1$, \[ \sum_{p_k \in \text{sibl}(p)} d_{p_k h} = 0 \text{ and } \sum_{h_l \in \text{sibl}(h)} d_{ph_l} = 0. \] Consider that the descendant association variable $d_{xy} \geq 0 \forall x \in V_P, y \in V_H$ is binary. Thus it follows that $d_{p_k h} = d_{ph_l} = 0 \forall p_k \in \text{sibl}(p) \forall h_l \in \text{sibl}(h)$.

Case (ii): Let $d_{ph} = 1$. Inequation (12) implies
\[ 1 \leq N_4 e_{ph} + \sum_{p_k \in \text{sibl}(p)} d_{p_k h} + \sum_{h_l \in \text{sibl}(h)} d_{ph_l} \leq N_4. \]
With the second premise
\[ \sum_{p_k \in \text{sibl}(p)} d_{p_k h} + \sum_{h_l \in \text{sibl}(h)} d_{ph_l} = 0 \]
this inequation yields
\[ 1 \leq N_4 e_{ph} \leq N_4 \iff 0 < \frac{1}{N_4} \leq e_{ph} < 1 \]
for a $N_4 > 1$. This is equivalent to $e_{ph} = 1$ because $e_{ph}$ is binary.

Case (iii): Let
\[ \sum_{p_k \in \text{sibl}(p)} d_{p_k h} + \sum_{h_l \in \text{sibl}(h)} d_{ph_l} + (1 - d_{ph}) \geq 1. \]
Then it follows
\[ 1 \leq N_4 e_{ph} + (1 - d_{ph}) + \sum_{p_k \in \text{sibl}(p)} d_{p_k h} + \sum_{h_l \in \text{sibl}(h)} d_{ph_l} \leq N_4. \]
The left inequation is satisfied in any case. The right inequation is satisfied if $e_{ph} = 0$ and $N_4$ is chosen large enough, such that
\[ N_4 \geq (1 - d_{ph}) + \sum_{p_k \in \text{sibl}(p)} d_{p_k h} + \sum_{h_l \in \text{sibl}(h)} d_{ph_l}, \]
which holds for $N_4 > \deg(p) + \deg(h) + 1 > 1$. Thus the left inequation is satisfied in every case.

Constraint:
\[ N_5 a_{ph} - e_{ph} \geq 0, \quad (14) \]
where $N_5 > \min \{\deg(p), \deg(h)\}$. 

4
Proof. Assume \( p \in V_p \) is not associated with a host \( h \in V_h \). Then \( a_{ph} = 0 \). We derive \( -c_{ph} \geq 0 \) from the inequation. This is equivalent to \( c_{ph} \leq 0 \). We know that \( c_{ph} \geq 0 \) because the number of cospeciations cannot be negative. Thus, \( c_{ph} = 0 \). On the other hand we need to ensure that if \( a_{ph} = 1 \) then (14) is satisfied. If \( a_{ph} = 1 \) then it follows that \( N_5 \geq c_{ph} \). This holds for each \( N_5 > \min \{ \deg(p), \deg(h) \} \).

Constraint:

\[
N_6 (1 - a_{ph}) - c_{ph} \geq 0
\]  

and \( N_6 > \min \{ \deg(p), \deg(h) \} \).

Proof. Let \( a_{ph} = 1 \). Then we obtain \( -c_{ph} \geq 0 \) which is equal to \( c_{ph} \leq 0 \). Thus, \( c_{ph} = 0 \) because the number of cospeciations \( c_{ph} \) cannot be negative. Otherwise, let us assume \( a_{ph} = 0 \). Then (15) yields to \( N_6 \geq c_{ph} \) which is ensured for \( N_6 > \min \{ \deg(p), \deg(h) \} \).

Constraints:

\[
N_7 (1 - a_{ph}) + c_{ph} - \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_k h_l} + 1 + N_7 \sum_{p_k \in \text{child}(p)} a_{p_k h} \geq 0
\]  

\[
N_7 (1 - a_{ph}) + \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_k h_l} - c_{ph} + N_7 \sum_{p_k \in \text{child}(p)} a_{p_k h} \geq 0
\]

where we define \( N_7 \) such that \( N_7 > \deg(p) \times \deg(h) \).

Proof. We have to prove that if \( a_{ph} = 1 \) and \( \sum_{p_k \in \text{child}(p)} a_{p_k h} = 0 \) then

\[
\sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_k h_l} - 1 \leq c_{ph} \leq \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_k h_l}.
\]

and otherwise if \( a_{ph} = 0 \) then there should be no additive bound for \( c_{ph} \).

Case (i): Let \( a_{ph} = 1 \) and \( \sum_{p_k \in \text{child}(p)} a_{p_k h} = 0 \) then (17) implies

\[
c_{ph} - \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_k h_l} + 1 \geq 0
\]

which is equivalent to

\[
c_{ph} \geq \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_k h_l} - 1.
\]

With the same approach (18) yields

\[
\sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_k h_l} - c_{ph} \geq 0,
\]

thus

\[
c_{ph} \leq \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_k h_l}.
\]
Case (ii): Let $a_{ph} = 1$ and $\sum_{p_k \in \text{child}(p)} a_{p_k h} \geq 1$. Then
\[
c_{ph} - \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_k h_l} + 1 + N_7 \sum_{p_k \in \text{child}(p)} a_{p_k h} \geq 0 \geq 1
\]
can be derived by (17). Furthermore (18) yields
\[
\sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_k h_l} - c_{ph} + N_7 \sum_{p_k \in \text{child}(p)} a_{p_k h} \geq 0.
\]
If we choose $N_7 > \deg(p) \ast \deg(h)$ then both inequations can be ensured.

Case (iii): Let $a_{ph} = 0$ and $\sum_{p_k \in \text{child}(p)} a_{p_k h} = 0$. Then it is easy to show that if $N_7 > \deg(p) \ast \deg(h)$ then the inequation holds.

Case (iv): Assume $a_{ph} = 0$ and $\sum_{p_k \in \text{child}(p)} a_{p_k h} \geq 1$. Then, (17), (18) imply
\[
N_7 + c_{ph} - \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_k h_l} + 1 + N_7 \sum_{p_k \in \text{child}(p)} a_{p_k h} \geq 0
\]
\[
\Leftrightarrow 2N_7 \geq \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_k h_l} - 1 - c_{ph} \leq 0
\]
\[
\Leftrightarrow 2N_7 > \deg(p) \deg(h) \geq \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_k h_l} - 1 - c_{ph}
\]
and analogously
\[
2N_7 > \min \{\deg(p), \deg(h)\} \geq c_{ph} - \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_k h_l}.
\]
Both inequations can be ensured by choosing $N_7 > \frac{\deg(p) \ast \deg(h)}{2}$ and the definition $N_7 > \deg(p) \ast \deg(h)$ satisfies this constraint.

\[\Box\]

Constraint:
\[
N_8 (1 - a_{ph}) + \left( \epsilon - \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_k h_l} \right)
\]
\[
N_8 \left( \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_k h_l} - c_{ph} \right) + N_8 \sum_{p_k \in \text{child}(p)} a_{p_k h} \geq 0
\]
where $\epsilon \in (0, 1)$ can be chosen arbitrarily and $N_8 > \deg(p) \ast \deg(h) + 1$. 
Proof. We have to prove that inequation (19) has the following implications.

(i) \( a_{ph} = 1, \sum_{p_k \in \text{child}(p)} a_{pk} h_{1} = 0, \sum_{p_k \in \text{child}(p)} \sum_{h_1 \in \text{child}(h)} e_{pk h_1} = 0 \Rightarrow c_{ph} = 0 \)

(ii) \( a_{ph} = 1, \sum_{p_k \in \text{child}(p)} a_{pk} h_{1} = 0, \sum_{p_k \in \text{child}(p)} \sum_{h_1 \in \text{child}(h)} e_{pk h_1} \geq 1 \)
\[ \Rightarrow c_{ph} = \sum_{p_k \in \text{child}(p)} \sum_{h_1 \in \text{child}(h)} e_{pk h_1} - 1 \]

(iii) \( a_{ph} = 0, \sum_{p_k \in \text{child}(p)} a_{pk} h_{1} = 0, \sum_{p_k \in \text{child}(p)} \sum_{h_1 \in \text{child}(h)} e_{pk h_1} = 0 \Rightarrow c_{ph} = 0 \)

(iv) \( a_{ph} = 0, \sum_{p_k \in \text{child}(p)} a_{pk} h_{1} = 0, \sum_{p_k \in \text{child}(p)} \sum_{h_1 \in \text{child}(h)} e_{pk h_1} \geq 1 \Rightarrow c_{ph} = 0 \)

(v) \[ \sum_{p_k \in \text{child}(p)} a_{pk} h_{1} \geq 1, a_{ph} \in \{0, 1\}, \sum_{p_k \in \text{child}(p)} \sum_{h_1 \in \text{child}(h)} e_{pk h_1} \geq 0 \Rightarrow c_{ph} = 0 \]

For reiteration, we proved:

(a) \( a_{ph} = 1 \) \( \Rightarrow c_{ph} \leq \sum_{p_k \in \text{child}(p)} \sum_{h_1 \in \text{child}(h)} e_{pk h_1} \)

(b) \( a_{ph} = 1 \) \( \Rightarrow c_{ph} \geq \sum_{p_k \in \text{child}(p)} \sum_{h_1 \in \text{child}(h)} e_{pk h_1} - 1 \)

(c) \( a_{ph} = 0 \) \( \Rightarrow c_{ph} = 0 \)

(d) \( \sum_{p_k \in \text{child}(p)} a_{pk} h_{1} \geq 1 \) \( \Rightarrow c_{ph} = 0 \)

Case (i): Let \( a_{ph} = 1 \). From \( \sum_{p_k \in \text{child}(p)} a_{pk} h_{1} = 0 \) and conclusions (a), (b) follows

\[ \sum_{p_k \in \text{child}(p)} \sum_{h_1 \in \text{child}(h)} e_{pk h_1} - 1 \leq c_{ph} \leq \sum_{p_k \in \text{child}(p)} \sum_{h_1 \in \text{child}(h)} e_{pk h_1}. \]

This and the second assumption \( \sum_{p_k \in \text{child}(p)} \sum_{h_1 \in \text{child}(h)} e_{pk h_1} = 0 \) yield \(-1 \leq c_{ph} \leq 0\).

We restricted the number of cospeciations \( c_{ph} \) such that \( c_{ph} \geq 0 \). Thus \( c_{ph} = 0 \). Now we have to ensure that (19) does not contradict this implication. Consider this case. Then (19) implies

\[ \epsilon - N_8 c_{ph} \geq 0 \iff c_{ph} \leq \frac{\epsilon}{N_8}. \]

By choosing \( \epsilon \in (0, 1) \) and \( N_8 > 1 \) which holds for \( N_8 > \deg (p) \ast \deg (h) + 1 \), it follows that

\[ c_{ph} \leq \frac{\epsilon}{N_8} < 1. \]

We also know that \( c_{ph} \in \mathbb{N}_0 \), thus the only possibility is \( c_{ph} = 0 \).
Case (ii): Assume $a_{ph} = 1$, $\sum_{p_k \in \text{child}(p)} a_{p_kh} = 0$ and $\sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_kh_l} \geq 1$.

Then analogously to case (i) it follows that

$$\sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_kh_l} - 1 \leq c_{ph} \leq \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_kh_l}.$$ 

Note that $c_{ph}$ can only be $\sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_kh_l} - 1$ or $\sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_kh_l}$ because $c_{ph} \in \mathbb{N}_0$. Furthermore it is easy to show that

$$\sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_kh_l} - c_{ph} \in \{0, 1\}.$$ 

With this assumption (19) we derive

$$\left( \epsilon - \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_kh_l} \right) + N_8 \left( \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_kh_l} - c_{ph} \right) \geq 0.$$ 

Consider the case $\sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_kh_l} - c_{ph} = 0$. We obtain

$$0 > \epsilon - \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_kh_l} \geq 0$$

which is a contradiction. Thus the only possibility to satisfy this inequation is the case $\sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_kh_l} - c_{ph} = 1$ because we achieve

$$N_8 \geq \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_kh_l} - \epsilon$$

while using this assumption. The inequation just mentioned holds for a $N_8 > \min_{p,h} \{\deg(p), \deg(h)\}$. Thus it holds for $N_8 > \deg(p) \cdot \deg(h) + 1$. Furthermore equation

$$\sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_kh_l} - c_{ph} = 1$$

has to be satisfied. Note that this is equivalent to

$$c_{ph} = \sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_kh_l} - 1.$$ 

Case (iii): Let $a_{ph} = 0$, $\sum_{p_k \in \text{child}(p)} a_{p_kh} = 0$ and $\sum_{p_k \in \text{child}(p)} \sum_{h_l \in \text{child}(h)} e_{p_kh_l} = 0$. Then we obtain

$$1 + \frac{\epsilon}{N_8} = \frac{N_8 + \epsilon}{N_8} \geq c_{ph}.$$
from (19). Note that $c_{ph}$ can either be 0 or 1 and with the proven implication (c) it follows that $c_{ph} = 0$.

Case (iv): Assume $a_{ph} = 0$, \( \sum_{p_k \in \text{child}(p)} a_{ph} = 0 \) and \( \sum_{p_k \in \text{child}(p)} \sum_{h_i \in \text{child}(h)} e_{ph} \geq 1 \).

Then

\[
N_8 + \left( \epsilon - \sum_{p_k \in \text{child}(p)} \sum_{h_i \in \text{child}(h)} e_{ph} \right) + N_8 \left( \sum_{p_k \in \text{child}(p)} \sum_{h_i \in \text{child}(h)} e_{ph} - c_{ph} \right) \geq 0
\]

\[
\Leftrightarrow N_8 + (N_8 - 1) \sum_{p_k \in \text{child}(p)} \sum_{h_i \in \text{child}(h)} e_{ph} + \epsilon \geq N_8 c_{ph}
\]

\[
\Leftrightarrow \frac{N_8 + (N_8 - 1) \sum_{p_k \in \text{child}(p)} \sum_{h_i \in \text{child}(h)} e_{ph} + \epsilon}{N_8} \geq c_{ph}
\]

can be derived from (19). The choosing $N_8 > \deg(p) \ast \deg(h) + 1 > 1$ leads to

\[
\frac{N_8}{N_8} + \frac{(N_8 - 1)}{N_8} \sum_{p_k \in \text{child}(p)} \sum_{h_i \in \text{child}(h)} e_{ph} + \epsilon \geq 1 \geq c_{ph}.
\]

Note that inequation (19) does not contradict the implication (c). Thus $c_{ph} = 0$.

Case (v): Let \( \sum_{p_k \in \text{child}(p)} a_{ph} \geq 1 \). Then implication (d) shows $c_{ph} = 0$. Now we have to prove that (19) does not contradict this conclusion. Consider (19). We show

\[
0 \leq N_8 (1 - a_{ph}) \leq N_8,
\]

\[
\epsilon - \min\{\deg(p), \deg(h)\} \leq \left( \epsilon - \sum_{p_k \in \text{child}(p)} \sum_{h_i \in \text{child}(h)} e_{ph} \right) \leq \epsilon,
\]

\[
0 \leq N_8 \left( \sum_{p_k \in \text{child}(p)} \sum_{h_i \in \text{child}(h)} e_{ph} - c_{ph} \right) \leq N_8,
\]

\[
N_8 \leq N_8 \sum_{p_k \in \text{child}(p)} a_{ph} \leq N_8 \deg(p).
\]

It is easy to ensure that the upper limit of (19) does not contradict the former implication. Thus we have to prove only the lower limit which is

\[
\epsilon - \min\{\deg(p), \deg(h)\} + N_8 \geq c_{ph},
\]

by choosing $N_8 \geq 1 + \min\{\deg(p), \deg(h)\} - \epsilon$. Thus by choosing $N_8 > \deg(p) \ast \deg(h) + 1$, we obtain

\[
1 \geq c_{ph}.
\]

In summary, inequation (19) does not contradict with $c_{ph} = 0$ in any case.
Constraints:

\begin{align}
N_9 t_{ph} & \geq c_{ph}, \\
c_{ph} & \geq t_{ph},
\end{align}

where \( N_9 > \text{deg}(p) \times \text{deg}(h) \).

\textit{Proof.} Let us consider the first case \( c_{ph} \geq 1 \). From (20) follows \( N_9 t_{ph} \geq 1 \) which yields

\[ t_{ph} \geq \frac{1}{N_9} > 0 \]

for \( N_9 > \min\{\text{deg}(p), \text{deg}(h)\} \). Due to the fact that \( t_{ph} \) is binary it follows that \( t_{ph} = 1 \). Inequality (21) yields \( 1 \geq t_{ph} \) which is not a constraint for any binary variable. Otherwise \( c_{ph} = 0 \). We infer from constraint (20) the inequality \( N_9 t_{ph} \geq 0 \) which is equivalent to \( t_{ph} \geq 0 \). Note that this is not a bound for the binary variable \( t_{ph} \). Considering (21) it follows \( 0 \geq t_{ph} \). Thus \( t_{ph} = 0 \).

Constraints:

\begin{align}
-N_{10} (1 - t_{ph}) + t_p - t_h & \leq 0 \quad (22) \\
-N_{10} (1 - t_{ph}) + t_h - t_p & \leq 0 \quad (23)
\end{align}

\textit{Proof.} Let us assume \( t_{ph} = 1 \). Then (22) and (23) yield \( t_p \leq t_h \) and \( t_h \leq t_p \). Thus \( t_h = t_p \). Otherwise if \( t_{ph} = 0 \) then inequality (22) conducts to \( -N_{10} + t_p - t_h \leq 0 \). This is equivalent to \( N_{10} \geq t_p - t_h \). (23) shows \( -N_{10} + t_h - t_p \leq 0 \) and thus \( N_{10} \geq t_h - t_p \). By the choice \( N_{10} > \max\{\text{height}(T_p), \text{height}(T_h)\} \geq \max\{t_p - t_h, t_h - t_p\} \) we ensure both inequations. Hence \( t_p \) and \( t_h \) are not restricted in this case.

Constraint:

\[ (1 - a_{ph}) + \sum_{h_i \in \text{anc}(h)} a_{p' h_i} + s_{ph} \geq 1 \quad (24) \]

\textit{Proof.} Let us assume \( a_{ph} = 1 \) and \( \sum_{h_i \in \text{anc}(h)} a_{p' h_i} = 0 \). Then (24) implies

\[ s_{ph} \geq 1 \quad s_{ph} \in \{0, 1\} \quad s_{ph} = 1. \]

Otherwise if \( a_{ph} = 0 \) and \( \sum_{h_i \in \text{anc}(h)} a_{p' h_i} \geq 0 \) we obtain

\[ s_{ph} \geq 0 \geq - \sum_{h_i \in \text{anc}(h)} a_{p' h_i} \geq 0 \]

which is no additional constraint because \( s_{ph} \) is binary. The last case we have to prove is \( a_{ph} = 1 \) and \( \sum_{h_i \in \text{anc}(h)} a_{p' h_i} \geq 1 \). Considering this case (24) yields

\[ 1 - \sum_{h_i \in \text{anc}(h)} a_{p' h_i} \leq 0 \leq s_{ph}, \]

thus \( s_{ph} \) is not an additional bound.
Constraint:

\[ a_{ph} \geq s_{ph} \]  \hspace{1cm} (25)

Proof. We obtain \( 0 \geq s_{ph} \) by inserting the case \( a_{ph} = 0 \). This yields \( s_{ph} = 0 \) because \( s_{ph} \) is binary. The insertion of \( a_{ph} = 1 \) leads to \( 1 \geq s_{ph} \). Thus \( s_{ph} \) can still be zero or one.

Constraint:

\[ \forall \hat{h} \in \text{anc}(h), 1 - a_{p'h} \geq s_{ph}. \] \hspace{1cm} (26)

Proof. Consider the case \( a_{p'h} = 1 \). We obtain \( 0 \geq s_{ph} \) and this is equal to \( s_{ph} = 0 \) because \( s_{ph} \) is binary. Otherwise if \( a_{p'h} = 0 \) then (26) yields \( 1 \geq s_{ph} \). Thus we get no further constraint for the binary variable \( s_{ph} \).
S2  Supplemental Figures

Figure S1: The figure shows one out of four optimal reconciliations for the gopher-lice data set. This reconciliation is obtained by CoRe-ILP using the branch length factor \( f_v \in [0.25, 4.0] \).
Figure S2: The figure shows one out of four optimal reconciliations for the gopher-lice data set. This reconciliation is obtained by CoRe-ILP using the branch length factor $f_v \in [0.25, 4.0]$. 
Figure S3: The figure shows one out of four optimal reconciliations for the gopher-lice data set. This reconciliation is obtained by CoRe-ILP using the branch length factor $f_v \in [0.25, 4.0]$. 
Figure S4: Violin plots showing the distribution for the precision of 100 recon-
ciliations computed with CoRe-ILP with reliable (CI1) and less reliable (CI2)
branch lengths, Jane 4 (J4), TreeMap 3b (T3b), NOTUNG 2.8 Beta (N2.8), and
Ranger-DTL (RDTL). The analysis was repeated 100 times for $pco = 0.25$ and
$psw \in \{0.25, 0.5, 0.75\}$. Black dots indicate the average precision.

Figure S5: Violin plots showing the distribution for the precision of 100 recon-
ciliations computed with CoRe-ILP with reliable (CI1) and less reliable (CI2)
branch lengths, Jane 4 (J4), TreeMap 3b (T3b), NOTUNG 2.8 Beta (N2.8), and
Ranger-DTL (RDTL). The analysis was repeated 100 times for $pco = 0.5$ and
$psw \in \{0.25, 0.5, 0.75\}$. Black dots indicate the average precision.
Figure S6: Violin plots showing the distribution for the precision of 100 reconciliations computed with CoRe-ILP with reliable (CI1) and less reliable (CI2) branch lengths, Jane 4 (J4), TreeMap 3b (T3b), NOTUNG 2.8 Beta (N2.8), and Ranger-DTL (RDTL). The analysis was repeated 100 times for $pco = 0.75$ and $psw \in \{0.25, 0.5, 0.75\}$. Black dots indicate the average precision.

Figure S7: Boxplots plots showing the runtime of 100 reconciliations computed with CoRe-ILP with reliable (CI1) and less reliable (CI2) branch lengths, Jane 4 (J4), TreeMap 3b (T3b), NOTUNG 2.8 Beta (N2.8), and Ranger-DTL (RDTL). The analysis was repeated 100 times for all combinations of parameters $pco = 0.25$ and $psw \in \{0.25, 0.5, 0.75\}$. 
Figure S8: Boxplots plots showing the runtime of 100 reconciliations computed with CoRe-ILP with reliable (CI1) and less reliable (CI2) branch lengths, Jane 4 (J4), TreeMap 3b (T3b), NOTUNG 2.8 Beta (N2.8), and Ranger-DTL (RDTL). The analysis was repeated 100 times for all combinations of parameters \( pco = 0.5 \) and \( psw \in \{0.25, 0.5, 0.75\} \).

Figure S9: Boxplots plots showing the runtime of 100 reconciliations computed with CoRe-ILP with reliable (CI1) and less reliable (CI2) branch lengths, Jane 4 (J4), TreeMap 3b (T3b), NOTUNG 2.8 Beta (N2.8), and Ranger-DTL (RDTL). The analysis was repeated 100 times for all combinations of parameters \( pco = 0.75 \) and \( psw \in \{0.25, 0.5, 0.75\} \).