Treewidth-based algorithms for the small parsimony problem on networks

Celine Scornavacca¹ and Mathias Weller²*

Abstract

Background: Phylogenetic reconstruction is one of the paramount challenges of contemporary bioinformatics. A subtask of existing tree reconstruction algorithms is modeled by the Small Parsimony problem: given a tree $T$ and an assignment of character-states to its leaves, assign states to the internal nodes of $T$ such as to minimize the parsimony score, that is, the number of edges of $T$ connecting nodes with different states. While this problem is polynomial-time solvable on trees, the matter is more complicated if $T$ contains reticulate events such as hybridizations or recombinations, i.e. when $T$ is a network. Indeed, three different versions of the parsimony score on networks have been proposed and each of them is NP-hard to decide. Existing parameterized algorithms focus on combining the number $c$ of possible character-states with the number of reticulate events (per biconnected component).

Results: We consider the parameter treewidth $t$ of the underlying undirected graph of the input network, presenting dynamic programming algorithms for (slight generalizations of) all three versions of the parsimony problem on size-$n$ networks running in times $c^t n^{O(1)}$, $(3c)^t n^{O(1)}$, and $6c^t n^{O(1)}$, respectively. Our algorithms use a formulation of the tree-width that may facilitate formalizing treewidth-based dynamic programming algorithms on phylogenetic networks for other problems.

Conclusions: Our algorithms allow the computation of the three popular parsimony scores, modeling the evolutionary development of a (multistate) character on a given phylogenetic network of low treewidth. Our results subsume and improve previously known algorithm for all three variants. While our results rely on being given a “good” tree-decomposition of the input, encouraging theoretical results as well as practical implementations producing them are publicly available. We present a reformulation of tree decompositions in terms of “agreeing trees” on the same set of nodes. As this formulation may come more natural to researchers and engineers developing algorithms for phylogenetic networks, we hope to render exploiting the input network’s treewidth as parameter more accessible to this audience.

Keywords: Phylogenetics, Parsimony, Phylogenetic networks, Parameterized complexity, Dynamic programming, Treewidth

Introduction

Molecular phylogenetic reconstruction consists in inferring a well-founded evolutionary scenario of a set of species from molecular data [1]. An evolutionary scenario, also called a phylogeny, is usually represented by a directed tree with a unique source called root. In a phylogeny, the tips of the tree are associated to extant species for which we have data, and each internal node represents an extinct species giving rise to new species—a speciation. Therefore, each internal node represents the hypothetical ancestor of all species below it, and the root models the lowest common ancestor of all the species at the tips.
Parsimony on trees

In this paper, molecular data consists of a set of molecular sequences (e.g. DNA or protein sequences) of the same length (one sequence per species). This kind of data can be seen as a matrix $M$ of $n$ sequences, each having $m$ characters (exhibiting one of $c$ possible states) where $M_{ij}$ corresponds to the state of the $j$th character exhibited by the $i$th species. There are several methods to reconstruct well-founded phylogenies from matrices of characters [1]. They are all based on the idea of retrieving similarities among species by comparing the states taken by these species at the different characters of $M$. Here, we will focus on parsimony methods. The main hypothesis of these methods is that character changes are not frequent. Thus, the phylogenies that best explain the data are those requiring the fewest evolutionary changes, i.e. the ones having the optimal parsimony score, formally defined in “Parsimony”. The problem of finding the optimal parsimony score for a given phylogeny $T$ with respect to an $n \times m$ matrix on a finite set of $c$ character states is called the Small Parsimony problem and can be solved in $O(n \cdot m \cdot c)$ time [2] since each column in the matrix can be analyzed independently in linear time. When $T$ is unknown, the problem of finding the phylogeny minimizing the parsimony score is called the Big Parsimony problem. This latter is known to be NP-hard and numerous heuristic techniques for it are known [1].

Parsimony on networks

When the evolution of the species of interest include, in addition to speciations, reticulate events such as hybridizations or recombinations, a single species may inherit from multiple direct ancestors. In this case, the phylogenies are no longer represented by rooted trees but by rooted DAGs [3] called networks. When scoring a given network, three very different definitions of the parsimony score have been proposed: the hardwired [4], the softwired [5, 6], and the parental parsimony score [7]. Roughly, the hardwired score takes into account all edges of the given network (characters are inherited from all parents), the softwired score takes only the edges of any “switching” (each character is inherited from one parent), and the parental score allows embedding lineages into the network (each allele of a character is inherited from one parent). See “Parsimony” for details and Fig. 1 for an example. While these definitions coincide for trees, they give rise to three different small parsimony problems for networks.

When tracing mutually dependent characters (e.g. different genomic locations in a same non-recombinant region) on networks, we also have to make sure that dependent characters are inherited from the same parent (some columns of the matrix have to use the same “switching” or “embedding”). To avoid dealing with this problem, the small parsimony problems on networks have been studied predominantly under the assumption of independent genomic locations. This boils down to having $m = 1$ since each column of the matrix can be analyzed independently (as is the case for the small parsimony problem on trees). Another popular restriction is to consider binary networks, in which the root has out-degree 2, tips have indegree 1, and internal nodes have either indegree 1 and outdegree 2 (speciations) or indegree 2 and outdegree 1 (reticulations).

The hardwired small parsimony problem has been proven NP-hard and APX-hard whenever the number of states of a character can take, denoted $c$, is strictly greater than 2, and polynomial-time solvable for binary characters [8]. A polynomial-time 1.35-approximation for all $c$ and a $\frac{12}{7}$-approximation for $c = 3$ have been proposed [8]. Additionally, the problem has been shown fixed-parameter tractable (FPT) in the parsimony score [8, 2$^p$ \cdot O(min(q^2, \sqrt{z}) \cdot q$)$ time], and in $c + r$ [9, $O(n \cdot c^{r+2})$ time], where $n, q, z$ are the number of leaves, vertices and edges in the phylogenetic network and $p$ and $r$ are the hardwired parsimony score and the number of reticulate events in the network.

The softwired small parsimony problem is also NP-hard and APX-hard [8, 10] for binary characters, and not FPT in the parsimony score (it is NP-hard to decide if the softwired parsimony score is 1). Also, it has been shown that, for any constant $\epsilon > 0$, no $n^{1-\epsilon}$ approximation can be computed in polynomial time, unless P = NP. On the positive side, the problem is FPT in $c + r$ [6, 8, $O(2^c \cdot n \cdot c$) time] and $c + \ell$ [8, 11, $O(2^c \cdot c^2 \cdot q \cdot z$) time], where $\ell$ is the maximum number of reticulations over all biconnected components of the network (also called the level of the network).

Unsurprisingly, the parental small parsimony problem has also been proven NP-hard, even for very restricted classes of networks, but it is FPT both with respect
to $c + r$ and with respect to $c + \ell$ [12], $O((2^c)^{r+2} \cdot q)$ and $O((2^c)^{r+3} \cdot q)$ time].

In this paper, we consider the case of independent characters, showing that the three variants of the small parsimony problem on networks are fixed-parameter tractable with respect to $c + t$ (running in time $O(T + c^{t+1} \cdot z)$, $O(T + c^t \cdot T^t \cdot c \cdot q + z)$), and $O(T + 6^t c \cdot 4^t \log (c) \cdot z)$], provided that a width-$t$ tree-decomposition of the input network $N$ can be computed in $T$ time (this is the case for $t$ equaling the treewidth of $N$ and $T \in 2^{O(k^2)}$ [13]). Our proofs are constructive in the sense that a dynamic programming algorithm is provided for each version of the problem. The main strength of our algorithms lies in their parameterization, since the treewidth can be arbitrarily small, even for growing values of $t$. An implication of parameterizing by the treewidth is that our algorithms run in polynomial time even on classes of networks on which previously known algorithms require exponential time while our algorithms run in polynomial time on all classes of networks that were previously known to allow for polynomial-time algorithms. Hence, our algorithms can potentially be orders of magnitude faster than the state-of-the-art solutions. Moreover, our formulations are not limited to binary networks and they can take into account polymorphism as well as external information controlling the states that ancestral species may take.

**Tree-width for phylogenetic networks**

The treewidth of a graph can roughly be described as a measure of “tree-likeness” and it ranks among the smallest of such parameters [14] (in particular, the treewidth can be seen to be smaller than the level $\ell$ on any network). Together with the fact that it facilitates the design of dynamic programming algorithms, this explains the enormous popularity the treewidth received in the parameterized complexity community [15, 16]. Starting with the groundbreaking work of Bryant and Lagergren [17] (using the celebrated result of Courcelle [18]), treewidth also gained traction with researchers studying algorithms for phylogenetics-related problems (surveyed in [19]). While this yielded some algorithms parameterized by the treewidth of the display graph of multiple trees (the result of “gluing” all trees at their leaves), we are not aware of any algorithms parameterized by the treewidth of the input network. In an attempt to facilitate the use of this parameter in future work, we dedicate Sect. “An alternative formulation of treewidth” to presenting a “phylogenetics-friendly” formulation by representing tree-decompositions of the input network as a rooted tree $'$ on the same vertex set as the network. In particular, this formulation generalizes our previously considered parameter “scanwidth” [20], which can be seen as a variant of treewidth that takes directness into account. While we expected scanwidth-based dynamic programming formulations to be easier and more straightforward than their treewidth-counterparts, this comes at the cost of the scanwidth being potentially arbitrarily larger than the treewidth. Intuitively speaking, we expect scanwidth dynamic programming to be easier since phylogenetic networks exhibit a “natural flow of information”: most often, we know everything about the leaves, but the more we approach the root, the more information has to be inferred from the lower parts. In contrast to the scanwidth-layout, tree-decompositions disregard edge directions and, thereby, this “natural flow”. Thus, while using the scanwidth allows for more naïve and intuitive dynamic programming formulations, using the treewidth requires more care and ingenuity.

Since we will suppose that a (not necessarily optimal) tree-decomposition of the input network is given in the input, let us discuss the current state-of-the-art for computing good decompositions. Optimal decompositions are indeed very hard to compute, with even the best-known parameterized algorithm being considered impractical (see survey [15]). This gloomy cloud has, however, two silver linings: First, if we do not insist on optimality, then we can use a recently published algorithm to compute 2-approximated tree-decompositions in $2^{O(k^2)} \cdot H^{O(1)}$ time [21]. We will state our results in a way that allows plugging-in any algorithm that computes or approximates tree decompositions. Second, with development driven by recent instances of the PACE challenge [22], more practical exact algorithms to compute tree decompositions are now available as well [23]. Herein, the running times of Tamaki’s implementation [23] are hard to predict and show erratic behavior even for fixed graph size. As expected, however, examples for high running times occur only for instances with high treewidth, that is, for “highly tangled” networks (see Fig. 2 for two select examples). This hints towards some hidden properties of the input networks that govern the complexity of treewidth computations As we expect “natural networks” to be only moderately tangled, we think that existing algorithms, exact and approximative, are currently well enough developed to deal with real world phylogenetic networks in reasonable timeframes. Indeed, we would welcome efforts similar to those made for the treewidth to also be made for the previously discussed scanwidth, which is also hard to compute [20].

For ease of presentation, the three main proofs (correctness of the dynamic programming formulations) are
given as high-level sketches and their more detailed and formal versions can be found in the appendix.

**Preliminaries**

**Mappings**

For any $x$ and $y$, we define $\delta(x, y)$ to be 0 if $x = y$ and 1, otherwise, and we abbreviate $1 - \delta(x, y) =: \overline{\delta}(x, y)$. We further abbreviate $\delta(\phi(x), \phi(y))$ as $\delta_{\phi}(x, y)$ for any function $\phi$. We may denote a pair $(x, y)$ as $x \rightarrow y$ if it is referring to an assignment of $y$ to $x$ by some function and as $xy$ if it refers to an arc in a network. We sometimes use the name of a function $\phi : X \rightarrow Y$ to refer to its set of pairs $\{x \rightarrow y \mid \phi(x) = y\}$ and we let $\phi|_Z := \{(x \rightarrow y) \in \phi \mid x \in Z\}$ denote the restriction of $\phi$ to $Z$. We say $\phi(x) = \bot$ to indicate that $\phi$ is not defined for $x$. We denote the result of forcing $\phi(x) = y$ (whether or not $x$ is mapped by $\phi$) as

$$\phi[x \rightarrow y] := \begin{cases} \phi \cup \{x \rightarrow y\} & \text{if } \phi(x) = \bot \\ (\phi \setminus \{x \rightarrow \phi(x)\}) \cup \{x \rightarrow y\} & \text{otherwise} \end{cases}$$

Finally, for sets $Z$, $X$, and $Y \subseteq X$ and functions $\phi$ and $\psi$, we write $\psi \leq \phi$ (and say that $\psi$ is a *subfunction* of $\phi$) if (a) $\phi : X \rightarrow Z$ and $\psi : Y \rightarrow Z$ and $\psi(x) \leq \phi(x)$ for all $x \in Y$, or (b) $\phi : X \rightarrow 2^Z$ and $\psi : Y \rightarrow Z$ and $\psi(x) \in \phi(x)$ for all $x \in Y$, or (c) $\phi : X \rightarrow 2^Z$ and $\psi : Y \rightarrow 2^Z$ and $\psi(x) \subseteq \phi(x)$ for all $x \in Y$.

**Graphs and phylogenetic networks**

In this work, we consider directed acyclic graphs (DAGs) $N$ that may have a unique source $\rho_N$ called root. If the sinks (aka leaves) of $N$ are labeled, we call $N$ a phylogenetic network. We refer to the nodes and directed edges (arcs) of $N$ by $V(N)$ and $A(N)$, respectively. The underlying undirected graph of $N$ is the undirected graph on node-set $V(N)$ that contains an edge $(u, v)$ if and only if $N$ contains the arc $(u, v)$. As we do not deal with mixed graphs, we use the term $uv$ to refer to the arc from $u$ to $v$ or the undirected edge between $u$ and $v$, depending on the context. We refer to the edge-set of an undirected graph $G$ as $E(G)$.

We denote the set of nodes of a DAG $N$ with in-degree at least two by $R(N)$ and we call such nodes reticulations. If $R(N) = \emptyset$, then $N$ is called a tree. The result of, for each $v \in R(N)$ removing all but one of its incoming arcs is called a *switching* of $N$ and $S(N)$ denotes the set of all switchings of $N$ (observe that all switchings are spanning trees). For each $v \in V(N)$, we denote the successors (or “children”) of $v$ in $N$ by $\text{Succ}_N(v)$ and its predecessors (or “parents”) by $\text{Pred}_N(v)$. If $N$ contains a directed $u$-$w$-path, then we say that $w$ is a *descendant* of $u$ and $u$ is an *ancestor* of $w$ (denoted as $w \preceq_N u$ and $w \preceq_N u$ if $u \neq w$). A set $Z \subseteq V(N)$ such that $u \not\in_N w$ and $w \not\in_N u$ for all $u, w \in Z$ is called an anti-chain in $N$. The induced subgraph $N[Z]$ of a set $Z \subseteq V(N)$ is the result of
removing all nodes \( x \in V(N) \setminus Z \) from \( N \) (together with their incident arcs) and, for any \( v \in V(N) \), the network \( N_v := N[[w \mid w \leq_N v]] \) is called the subnetwork \textit{rooted at} \( v \).

**An alternative formulation of treewidth**

In this section, we give an alternative definition of the \textit{treewidth}, which allows to tackle the small parsimony problem for networks in a simpler and more intuitive way. Note that this alternative definition is known in the FPT community (Dendris et al. [24]) call it the “support” of a vertex with respect to an ordering while, when referring to Arnborg [25]) and Mescoff et al. [26], call it “tree vertex separation”). However, since in these works its connection to treewidth is mostly touched in passing, we felt the need to prove it explicitly here.

Since tree decompositions are agnostic to edge directions, all results in this section are stated for undirected graphs \( G \) instead of networks \( N \). Keeping in mind that the framework is to be applied to phylogenetic networks, all results are stated for undirected graphs. The reader may simply ignore the edge directions in the framework is to be applied to phylogenetic networks, all results in this section are stated for undirected graphs. However, since in these works its connection to treewidth is mostly touched in passing, we felt the need to prove it explicitly here.

For a linear ordering \( \sigma \) of the nodes of an undirected graph \( G \) and any \( x \in V(G) \), we write \( y \preceq_\sigma x \) for all nodes \( y \) preceeding \( x \) in \( \sigma \) (including \( x \) itself) and let \( \sigma [1..x] \) denote the restriction of \( \sigma \) to these nodes. We write \( x \rightsquigarrow y \) if \( x \) and \( y \) are connected in \( G[\sigma [1..x]] \) (see Fig. 3 for an example). Note that \( G[\sigma] \) is a partial order on \( V(G) \). We consider nodes outside \( \sigma [1..v] \) that have an edge to the parts of \( \sigma [1..v] \) that are connected to \( v \) in \( G[\sigma [1..v]] \). We denote these nodes by \( ZW^G_v \) and their number by \( zw^G_v \).

**Definition 1** Let \( \sigma \) be a linear order of the nodes of an undirected graph \( G \) and let \( v \in V(G) \). Then,

\[
ZW^G_v := \{ u >_\sigma v \mid \exists w \in \sigma [1..v] : \{ uv, w \} \in E(G) \land v \overset{G[\sigma]}{\preceq} w \}
\]

and \( zw^G_v := |ZW^G_v| \).

We abbreviate \( zw(\sigma) := \max, zw^G \) and \( zw(G) := \min, zw(\sigma) \) and we refer to the transitive reduction of the directed graph \( (V(G), \{ uv \in V(G)^2 \mid u \overset{G[\sigma]}{\preceq} v \}) \) as the \textit{canonical tree} \( I^{\sigma} \) of \( \sigma \) for \( G \) (we will see below that \( I^{\sigma} \) is a rooted tree; see Fig. 3).

In the following, we say that a rooted tree \( \Gamma \) on \( V(G) \) agrees with an undirected graph \( G \) if, for all \( uv \in E(G) \) either \( u \preceq_\Gamma v \) or \( v \preceq_\Gamma u \). We also extend the definition of \( G[\sigma] \) to such trees by writing \( u \overset{\Gamma}{\preceq} v \) if \( u \) and \( v \) are connected in \( G[\Gamma_u] \). In analogy to Definition 1, \( G[\Gamma] \) gives rise to a set \( YW^\Gamma_v \) containing the nodes “above” \( v \) in \( \Gamma \) that have an edge in \( G \) to a node “below” \( v \) in \( \Gamma \).

**Definition 2** (see Fig. 3) Let \( G \) be an undirected graph and let \( \Gamma \) agree with \( G \). For each \( v \in V(G) \), we define

\[
YW^\Gamma_v := \{ u \overset{\Gamma}{\preceq} v \mid \exists w \in \Gamma : \{ u, w \} \in E(G) \} \quad \text{and} \quad yw^\Gamma_v := |YW^\Gamma_v|.
\]

Then, we abbreviate \( yw(\Gamma) := \max, yw^\Gamma \) and \( yw(G) : = \min, yw(\Gamma) \).

Note that the path \( P \) resulting from traversing \( \sigma \) from right to left is a rooted tree agreeing with \( G \). However, \( yw(P) \) is expected to be large for this choice. Indeed, we...
can show that the most “refined” trees $\Gamma$ have the smallest $yw(\Gamma)$.

**Lemma 1** Let $\Gamma$ and $\Gamma'$ be rooted trees agreeing with an undirected graph $G$ and let $\leq_{\Gamma}$ be a subset of $\leq_{\Gamma'}$ that is, $x \leq_{\Gamma'} y \Rightarrow x \leq_{\Gamma} y$ for all $x, y \in V(G)$. Then, $yw(\Gamma') \leq yw(\Gamma)$.

**Proof** Let $x \in V(G)$ and let $y \in YW^x$; that is, $y \geq_{\Gamma'} x$ and there is some $z \leq_{\Gamma'} x$ with $yz \in E(G)$. Since $\leq_{\Gamma}$ is a superset of $\leq_{\Gamma'}$, we have $y \geq_{\Gamma} x \geq_{\Gamma}$, implying $y \in YW^x$.

The following lemma proves a number of interesting properties relating $\sigma$ and $\Gamma^\sigma$ such as $\Gamma^\sigma$ being a rooted tree whose descendant relation is a refinement of $\leq_{\sigma}$, culminating in the equality of $ZW^\sigma_x$ and $YW_x^\sigma$ for all $x$.

**Lemma 2** Let $\sigma$ be a linear order of the nodes of a connected undirected graph $G$ and let $\Gamma^\sigma$ be its canonical tree. Then,

(a) for each $u$ and $v$ with $v \leq_{\Gamma^\sigma} u$, we have $v \leq_{\sigma} u$,

(b) for each $u, v \in V(G)$, we have $v \leq_{\Gamma^\sigma} u$ if and only if $u \leq_{\sigma} v$,

(c) $\Gamma^\sigma$ is connected,

(d) $\Gamma^\sigma$ is rooted at the last vertex $r$ of $\sigma$,

(e) $\Gamma^\sigma$ is a tree,

(f) for all $uv \in E(G)$ with $v \leq_{\sigma} u$, we have $v \leq_{\Gamma^\sigma} u$,

(g) $\Gamma^\sigma$ agrees with $G$, and

(h) $YW_x^\sigma = ZW_x^\sigma$ for all $x \in V(G)$.

(i) For each arc $xy \in A(\Gamma^\sigma)$, $\Gamma^\sigma_y$ contains a neighbor of $x$ in $G$.

(j) Each $x \in V(G)$ has at most as many children in $\Gamma^\sigma$ as it has neighbors in $G$.

**Proof** (a), (b): We show for all vertices $w$ on a $u$-$v$-path $p$ in $\Gamma^\sigma$ that $w \leq_{\sigma} u$ and $u \geq_{\sigma} w$. The base case $w = u$ holds trivially. For the induction step, let $q$ precede $w$ in $p$. Since $\Gamma^\sigma$ contains the arc $qw$, Definition 1 implies $q \leq_{\sigma} w$ and, since $q \leq_{\sigma} u$ by induction hypothesis, $w \leq_{\sigma} q \leq_{\sigma} u$ and $u \geq_{\sigma} w$. For the reverse direction of (b), note that, by Definition 1, $uv$ is an arc of the DAG of which $\Gamma^\sigma$ is the transitive reduction.

(c), (d): Since $G$ is connected, each $x \in V(G)$ has an $r$-$x$-path in $G = G[\sigma(1..r)]$, implying $r \leq_{\sigma} x$. Thus, (b) implies that $\Gamma^\sigma$ is connected and rooted at $r$.

(e): To prove that $\Gamma^\sigma$ is a tree, assume there is a vertex $x \in V(G)$ with two distinct parents $y$ and $z$ in $\Gamma^\sigma$. Without loss of generality, let $y \leq_{\sigma} z$. By (b), $y \leq_{\Gamma^\sigma} x$ and $z \geq_{\Gamma^\sigma} x$, implying that $\sigma(1..y]$ contains a $y$-$x$-path $p_y$ in $G$ and $\sigma(1..z]$ contains a $z$-$x$-path $p_z$ in $G$. Since $\sigma(1..y] \subseteq \sigma(1..z]$ the concatenation of $p_z$ with (the reverse) of $p_y$ is a path in $G$ whose nodes are in $\sigma(1..z]$. Thus, $z \geq_{\sigma} y$, implying $y \leq_{\Gamma^\sigma} z$ and, since $xz \in A(\Gamma^\sigma)$, this contradicts $\Gamma^\sigma$ being a transitive reduction.

(f): Note that $u \leq_{\sigma} v$, implying $v \leq_{\Gamma^\sigma} u$ by (b).

(g): For each $uv \in E(G)$, either $u \leq_{\sigma} v$, implying $u \leq_{\Gamma^\sigma} v$, or $v \leq_{\sigma} u$, implying $v \leq_{\Gamma^\sigma} u$ (both by (f)).

(h) “$\leq_{\Gamma^\sigma}$: Let $x \in V(G)$ and let $y \in YW^x$. By Definition 2, $y \geq_{\Gamma^\sigma} x$ (implying $y \geq_{\sigma} x$ by (a)) and there is some $z \geq_{\Gamma^\sigma} x$ (implying $z \leq_{\sigma} x$ by (a)) with $yz \in E(G)$. Then, by (b), $x \leq_{\Gamma^\sigma} z$. But then, $y \in ZW^x$ by Definition 1.

(i) “$\geq_{\sigma}$: Let $x \in V(G)$ and let $y \in ZW^x$. By Definition 1, $x \geq_{\sigma} y$.

(j) is immediate from (i) combined with (e).
Proof. Let \( u \in V(G) \) such that \( X = \{ v <_\Gamma u \mid u \not\in v \} \neq \emptyset \). We will modify \( \Gamma' \) with \( y_w(\Gamma') \leq y_w(\Gamma) \) such that \( \Gamma' \) agrees with \( G \) and the relation \( \leq_\Gamma \) is a strict subset of \( \leq_\Gamma \). To this end, note that \( u \) has a parent \( w \) in \( \Gamma' \) as, otherwise, \( G[\Gamma_u] = G \), implying \( X = \emptyset \). Then, \( \Gamma' \) results from \( \Gamma \) by (see Fig. 4)

1. replacing \( \Gamma \) by \( \Gamma' \uparrow (\Gamma_u \setminus X) \) and
2. danging \( \Gamma_u \uparrow X \) from \( w \).

First, we show that \( \Gamma' \) agrees with \( G \). To this end, let \( xy \in E(G) \) and let \( x \) and \( y \) be unrelated in \( \Gamma' \). If neither \( x \) nor \( y \) are in \( \Gamma_u \), then, by construction of \( \Gamma' \), they are also unrelated in \( \Gamma \), contradicting that \( \Gamma \) agrees with \( G \). So, without loss of generality, suppose \( x \leq_\Gamma u \). Since \( xy \in E(G) \) and \( \Gamma \) is a tree agreeing with \( \Gamma \), we thus know that \( u \) and \( y \) are not unrelated in \( \Gamma \). If \( u \leq_\Gamma y \), then \( w \leq_\Gamma y \) and, thus, \( x \leq_\Gamma y \). Clearly, if \( x, y \in X \) or \( x, y \not\in X \), then \( x \) and \( y \) are also unrelated in \( \Gamma \), contradicting its agreement with \( G \). Thus, without loss of generality, suppose \( x \in X \) and \( y \not\in X \), that is, \( u \not\leq_\Gamma x \) and \( u \not\leq_\Gamma y \), contradicting \( xy \in E(G) \).

Second, we show that \( \leq_{\Gamma'} \) is a strict subset of \( \leq_\Gamma \). To this end, let \( xy \in A(\Gamma') \) and assume towards a contradiction that \( y \not\in X \). Clearly, if \( x \not\leq_{\Gamma'} w \), then \( xy \in A(\Gamma) \) contradicting \( y \not\in X \). Further, if \( x = w \), then either \( y \in X \) or \( y \) is a child of \( w \) in \( \Gamma' \), of which only imply \( x \leq_\Gamma x \). Thus, \( x \leq_{\Gamma'} w \). Since \( xy \cap X = \{ x \} \) or \( xy \cap X = \{ y \} \) contradicts \( xy \in A(\Gamma') \), we have \( x, y \in X \) or \( x, y \not\in X \). But then, \( y \setminus X \) by Observation 1. Thus, \( \leq_{\Gamma'} \) is a subset of \( \leq_\Gamma \) and it is strict since we have \( v \leq_{\Gamma'} u \) and \( v \not\leq_{\Gamma'} u \) for all \( v \in X \neq \emptyset \).

Third, \( y_w(\Gamma') \leq y_w(\Gamma) \) follows by Lemma 1. □

Lemma 4. Let \( \Gamma \) be a rooted tree agreeing with a graph \( G \) and let \( p \) be a non-empty path in \( G \). Then, \( p \) contains a unique maximum \( u \) with respect to \( \Gamma \), that is, \( v \leq_{\Gamma} u \) for all vertices \( v \) of \( p \).

Proof. Let \( x \) on \( p \) be maximal with respect to \( \Gamma \) (that is, for all \( z \) on \( p \), we have \( x \not\leq_{\Gamma} z \)) and assume towards a contradiction that there is another vertex \( y \not\in x \) on \( p \) that is maximal w.r.t. \( \Gamma \). Without loss of generality, let \( x \) precede \( y \) in \( p \) and let \( p_{xy} \) denote the unique \( x-y \)-subpath of \( p \). Since \( y \not\leq_{\Gamma} x \), there is an edge \( st \in E(G) \) on \( p_{xy} \) with \( s \leq_{\Gamma} x \) and \( t \not\leq_{\Gamma} x \). Hence, \( t \not\leq_{\Gamma} s \). Further, \( s \not\leq_{\Gamma} t \) since, otherwise, the unique \( t-s \)-path in \( \Gamma \) contains \( x \), contradicting its maximality. But then \( \Gamma \) does not agree with \( G \). □

Lemma 5. Let \( G \) be a graph. Then, \( zw(G) = yw(G) \).

Proof. “\( \geq \):” Let \( \sigma \) be an ordering of \( V(G) \) such that \( zw(\sigma) = zw(G) \). By Lemma 2(h), we have \( zw(\sigma) = yw(\Gamma(\sigma)) \) for the canonical extension tree \( \Gamma(\sigma) \) of \( \sigma \). Thus, \( zw(G) = zw(\sigma) \geq yw(G) \). “\( \leq \):” Let \( \Gamma \) be some rooted tree agreeing with \( G \) such that \( yw(\Gamma) = yw(G) \). By Lemma 3, we may assume

\[
\forall_{u,v \in V(G)} u \leq_{\Gamma} v \Rightarrow u \not\leq_{\Gamma'} v.
\]

(1)

Let \( \sigma \) be any ordering of \( V(G) \) obtained by repeatedly picking and removing any leaf of \( \Gamma \). □

Claim 1. For each \( u, v \in V(G) \), we have \( u \leq_{\Gamma'} v \) if and only if \( v \not\leq_{\Gamma'} u \).
Proof First, note that all nodes below $v$ in $\Gamma$ are chosen before $v$, so $\Gamma_v \subseteq \sigma[1..v]$.

\[\Rightarrow\]: Let $u \leq \Gamma v$, that is, $u \in \Gamma_v$, implying $u \leq_{\sigma} v$. By (1), $v$ is connected to $u$ in $G[\Gamma_v]$ and, as $\Gamma_v \subseteq \sigma[1..v]$, also in $G[\sigma[1..v]]$.

\[\Leftarrow\]: Let $p$ be a $v$-$u$-path in $G[\sigma[1..v]]$. By Lemma 4, $p$ has a unique maximum $w$ in $\Gamma$. Hence, $v \leq \Gamma w$ and, by \(\Rightarrow\), we have $v \leq_{\sigma} w$. Since $p$ lives entirely in $G[\sigma[1..v]]$, that is, $V(p) \subseteq \sigma[1..v]$, we also have $w \leq_{\sigma} v$. Thus, $v = w$ and, since $u \in V(p)$, we have $u \leq \Gamma w = v$ by maximality of $w$. \qed

To prove the lemma, we show $\YW_{\times}^w \supseteq \ZW_{\times}^w$ for each $x \in V(G)$. Let $y \in \ZW_{\times}^w$, that is, $y >_{\sigma} x$ and there is some $z \in \sigma[1..t]$ with $yz \in E(G)$ and $x \leq_{\sigma} z$. By Claim 1, $z \leq \Gamma x$. Further, as $yz \in E(G)$ and $\Gamma$ agrees with $G$, $y$ and $z$ are not unrelated in $\Gamma$ and, since $z \leq \Gamma x$, neither are $x$ and $y$. Since $y <_{\Gamma} x$ implies $y <_{\sigma} x$ by Claim 1, contradicting $y >_{\sigma} x$, we conclude $x <_{\Gamma} y$. Together with $z \leq \Gamma x$ and $yz \in E(G)$, this implies $y \in \YW_{\times}^w$.

Having shown that the notion of $\zw(G)$ and $\yw(G)$ are equivalent, we can now turn our attention to the treewidth. In particular, we introduce (nice) tree-decompositions and use their properties to show that the treewidth of any undirected graph $G$ equals $\yw(G)$.

Definition 3 (see Fig. 5) Let $G$ be an undirected graph and let $T$ be a rooted tree whose vertices are associated to subsets of $V(G)$ by a function $B : V(T) \rightarrow 2^{V(G)}$ such that

(a) for each $uv \in E(G)$, there is some $x \in V(T)$ with $u, v \in B(x)$

(b) for each $v \in V(G)$, the nodes $x \in V(T)$ with $v \in B(x)$ are weakly connected in $T$.

We call $(T, B)$ a tree decomposition of $G$ and its width is
tw($T, B$) := max$_x \{ B(x) \}$ such that $tw_x(T, B) := \{|B(x)| - 1\}$.
We call $tw(G) := \min_{T, B}$ $tw(T, B)$ the treewidth of $G$.

We call $(T, B)$ nice if $T$ is binary and all $x \in V(T)$ fall into one of the following categories

- **“leaf”:** $x$ is a leaf of $T$ and $B(x) = \emptyset$,
- **“root”:** $x$ is the root of $T$ and $B(x) = \emptyset$,
- **“introduce $v$”:** $x$ has a single child $y$ in $T$ and $B(y) = B(x) - v$,
- **“forget $v$”:** $x$ has a single child $y$ in $T$ and $B(x) = B(y) - v$,
- **“join”:** $x$ has two children $y$ and $z$ and $B(x) = B(y) = B(z)$.

As stated at the beginning of the section, recall that, while tree decompositions are defined for undirected graphs, we may talk about tree decompositions of DAGs, meaning tree decompositions of their underlying undirected graphs. Note that all graphs $G$ have a nice tree decomposition with $|V(T)| \in O(tw(G) \cdot |G|)$ and width $\tw(G)$ [27]. Further, since all bags of $(T, B)$ containing a vertex $v$ of $G$ are connected, we can observe the following.

Observation 2 Let $(T, B)$ be a nice tree decomposition for an undirected graph $G$ and let $v \in V(G)$. Then, $T$ contains a single “forget $v$”-node $x$ and $y <_{\Gamma} x$ for all $y$ with $v \in B(y)$.

Proposition 1 Let $G$ be an undirected graph. Then, $\yw(G) = \tw(G)$. Further, given a tree decomposition $(T, B)$ for $G$, we can compute a tree $\Gamma$ agreeing with $G$ such that $\yw(\Gamma) = \tw(T, B)$ in linear time.

Proof “$\leq$”: Let $(T, B)$ be a nice tree decomposition for $G$ of width $\tw(G)$ and let $F \subseteq V(T)$ denote the set of all “forget”-nodes in $T$ (noting that $F$ contains the root of $T$). We define $\Gamma$ as the transitive reduction of $(F, \geq_T \land (F \times F))$. Note that $u \leq_{\Gamma} v \iff u \leq_T v$ for all $u, v \in F$ and, by Observation 2, $V(\Gamma) = F = V(G)$.

First, we show that $\Gamma$ agrees with $G$. To this end, let $uv \in E(G)$ and let $f_uyf_z \in F$ denote the unique “forget $u$” and “forget $v$”-nodes in $T$, which are distinct since $T$ is nice. By Definition 3(a), there is a node $q \in V(T)$ with $u, v \in B(q)$ and, by Observation 2, $uv \in E(G)$ and $f_uyf_z \in F$ denote the unique “forget $u$” and “forget $v$”-nodes in $T$, which are distinct since $T$ is nice. By Definition 3(a), there is a node $q \in V(T)$ with $u, v \in B(q)$ and, by Observation 2,
Then, by Definition 3(b), $(\mathcal{W}_u) \cap \mathcal{W}_v(\mathcal{W}_x) \subseteq (\mathcal{W}_u) \cup \mathcal{W}_v(\mathcal{W}_x)$. Finally, note that $\mathcal{W}_u^T = \mathcal{W}_u(\Gamma_u) \cup \mathcal{W}_v(\Gamma_v)$ and $\bigcup_{x \in V(N)} \mathcal{W}_x^T \subseteq \mathcal{W}_x^T \cup \{v\}$.

\section*{Parsimony}

\textbf{Notation}\hspace{1em}Large parts of this work are in context of a rooted tree $\Gamma$ on the node set $V(N)$ of a given phylogenetic net $N$ (see Fig. 6). Specifically for the tree $\Gamma$, we permit ourselves to abbreviate $V(\Gamma_x)$ to $\Gamma_x$ to increase readability. In such context, we additionally define the following sets for any nodes $y, z \in V(N)$: $\text{Pred}_y^z(z) := \text{Pred}_N(z) \cap \Gamma_y$ and $\text{Succ}_y^z(z) := \text{Succ}_N(z) \cap \Gamma_y$ denote the respective predecessors of $z$ in $N$ that are or are not in $\Gamma_y$. Likewise, $\text{Succ}_y^z(z) := \text{Succ}_N(z) \cap \Gamma_y$ and $\text{Succ}_y^z(z) := \text{Succ}_N(z) \cap \Gamma_y$ denote the respective successors of $z$ in $N$ that are or are not in $\Gamma_y$. Note that the arrow in the notation indicates the direction of the arc between $z$ and the members of the set when drawing $\Gamma$ top-down. If $z = y$, we drop $y$ and simply write $\text{Pred}_y^z(z)$, $\text{Succ}_y^z(z)$, and $\text{Succ}_y^z(z)$.

\begin{figure}
\centering
\includegraphics[width=\textwidth]{fig6}
\caption{A tree $\Gamma$ is depicted in gray and some arcs of $N$ are depicted in black. Recall that $t$ is the number of children of $x$ and $Z_t := \bigcup_{x \in V(N)} \Gamma_x$. Note that $x \in \text{Succ}_y^z(z) \setminus \text{Succ}_y^z(x)$ since $z$ is an ancestor of a node of $\Gamma_y$ in $N$. Note that $x$ is a reticulation of $N$ with parents $y$ (drawn) and $z$ (not drawn) with $y \prec t < z \prec x < r \prec t$. Thus, $z \in \text{Pred}_y^z(x)$ but $y \in \text{Pred}_y^z(z) \subseteq \text{Pred}_y^z(x)$. Finally, note that $\mathcal{W}_u^T = \mathcal{W}_u(\Gamma_u) \cup \mathcal{W}_v(\Gamma_v)$ and $\bigcup_{x \in V(N)} \mathcal{W}_x^T \subseteq \mathcal{W}_x^T \cup \{v\}$}.
\end{figure}

This follows by definition of $B(u)$. Otherwise, $x \prec \Gamma u$ since $x \in V(p)$. But then, $w \leq \Gamma v \leq \Gamma x \prec \Gamma u$ and $uw \in E(G)$, implying $u \in \mathcal{W}_x^T \subseteq B(x)$.

\section*{Parsimony}

\textbf{Notation}\hspace{1em}Large parts of this work are in context of a rooted tree $\Gamma$ on the node set $V(N)$ of a given phylogenetic net $N$ (see Fig. 6). Specifically for the tree $\Gamma$, we permit ourselves to abbreviate $V(\Gamma_x)$ to $\Gamma_x$ to increase readability. In such context, we additionally define the following sets for any nodes $y, z \in V(N)$: $\text{Pred}_y^z(z) := \text{Pred}_N(z) \cap \Gamma_y$ and $\text{Succ}_y^z(z) := \text{Succ}_N(z) \cap \Gamma_y$ denote the respective predecessors of $z$ in $N$ that are or are not in $\Gamma_y$. Likewise, $\text{Succ}_y^z(z) := \text{Succ}_N(z) \cap \Gamma_y$ and $\text{Succ}_y^z(z) := \text{Succ}_N(z) \cap \Gamma_y$ denote the respective successors of $z$ in $N$ that are or are not in $\Gamma_y$. Note that the arrow in the notation indicates the direction of the arc between $z$ and the members of the set when drawing $\Gamma$ top-down. If $z = y$, we drop $y$ and simply write $\text{Pred}_y^z(z)$, $\text{Succ}_y^z(z)$, and $\text{Succ}_y^z(z)$. We also abbreviate $\text{Pred}_N(z) \cap R(G) := \text{Pred}_N^R(z)$ and $\text{Succ}_N(z) \cap R(G) := \text{Succ}_N^R(z)$ as well as $\text{Pred}_N(z) \setminus R(G) := \text{Pred}_N^L(z)$ and $\text{Succ}_N(z) \setminus R(G) := \text{Succ}_N^L(z)$. All these functions generalize to sets $Z \subseteq V(N)$ (for example, $\text{Pred}_N(Z) := \bigcup_{z \in Z} \text{Pred}_N(z) \setminus Z$). Further, for any $X \subseteq V(N)$, we define the sets of arcs of $N$

\begin{enumerate}
\item[(a)] from a node $u \in X$ to any ancestor of $u$ in $\Gamma$ as $A_\Gamma^X(N) := \{uw \in A(N) \mid u \in X \land u < \Gamma w\}$ and
\item[(b)] to a node $u \in X$ from any ancestor of $u$ in $\Gamma$ as $A_\Gamma^X(N) := \{uw \in A(N) \mid w \in X \land w < \Gamma u\}$.
\end{enumerate}
For brevity, we abbreviate \( A_X(N) := \bigcup \{ A_X^1(N), A_X^2(N) \} \)
\( A_Y(N) := \bigcup \{ A_Y^1(N), A_Y^2(N) \} \), and \( A_v(N) := A_{v}^{1}(N) \).

Introduction to Parsimony. Given states of a character, observed in extant species, as well as a species phylogeny, the small parsimony problem asks to infer states of the same character for all ancestral species such as to minimize the "parsimony score" of this assignment. This problem comes in three flavors called "hardwired", "softwired", and "parental" parsimony. Throughout this section, let \( C \) be a fixed finite set (a "character"). For convenient use of the \( \subseteq \) relation, let \( C \) be an anti-chain (that is, for each \( x, y \in C \), we have \( x \leq y \) only if \( x = y \)). Formally, for a phylogeny \( N \) and a function \( \phi: V(N) \to 2^C \), we define the hardwired and softwired parsimony score as

\[
\begin{align*}
\text{par}_N^H(\phi) &:= \min_{\psi: V(N) \to C} \psi \leq \phi \sum_{uv \in A(N)} \delta(\psi(u), \psi(v)) \\
\text{par}_N^S(\phi) &:= \min_{\psi: V(N) \to C} \min_{T \in S(N)} \sum_{v \in A(T)} \delta(\psi(u), \psi(v)).
\end{align*}
\]

The "parental parsimony" is defined using "parental trees" but, in this work, we use the equivalent formulation using lineage functions [12].

**Definition 4** A lineage function for a phylogeny \( N \) is any function \( \phi: V(N) \to 2^C \). The cost of \( \phi \) is

\[
\text{cost}_N(\phi) := \sum_{v \in V(N)} \text{cost}_v(\phi)
\]

For each of the presented variants, we give a dynamic programming formulation using a given tree \( \Gamma \) that agrees with the undirected graph \( G \) underlying the input network and corresponds to Lemma 3, that is, each non-leaf \( x \) of \( \Gamma \) has a child \( v \) with \( x \in YW^l_v \). The running time of the resulting algorithm will depend on the width \( yw(\Gamma) \) of \( \Gamma \) (recalling that \( yw(\Gamma) \) coincides with the treewidth of \( G \) for optimal \( \Gamma \)).

As stated in the introduction, in this paper we focus on the case of analyzing a specific position in the genome. Since the function \( \phi \) can associate several states to a same leaf, our definition permits to describe polymorphism in a population. While in our current formulation the algorithms "choose" an optimal state to associate to each leaf, the parental parsimony can be easily modified to explain all states of each leaf at the end of the run. This allows keeping the information on polymorphism in all steps of the algorithm (see "Parental parsimony"). Note also that \( \phi \) can associate information to internal nodes, thus permitting the user to impose restrictions on the states associated to ancestral species.

In the presentation of the dynamic programming, a table entry \( Q^z_i \) means that \( x \) and \( y \) are considered fix for this table and \( z \) is a variable index. Further, tables \( Q^z_{i1} \) and \( Q^z_{i2} \) are independent of one another, allowing an implementation to forget \( Q^z_{i1} \) if it is no longer needed, even if \( Q^z_{i2} \) still is. In the following, for an anti-chain \( Y \) in \( \Gamma \) and a class \( \mathcal{G} \) of subnetworks of \( N \), a \( Y \)-substitution system of \( \mathcal{G} \) is a series of subnetworks \( (N^y)_{y \in Y} \) of \( N \) such that, for all \( N' \in \mathcal{G} \), the digraph \( (V(N), (A(N') \setminus \bigcup_{y \in Y} A_y(N')) \cup \bigcup_{y \in Y} A_y(N')) \) is also in \( \mathcal{G} \). Roughly, we can "swap out" the arcs in \( A_y(N') \) for \( A_y(N^y) \) for each \( y \in Y \) without loosing membership in \( \mathcal{G} \). Note that the \( N^y \) are not necessarily distinct, so a trivial \( Y \)-substitution system for \( \{N\} \) would be \( (N^y)_{y \in Y} \). The formulations are based on the following lemma about independent sub-solutions, showing that an optimal solution, \( (S, \psi) \), for a sub-network (of \( \mathcal{G} \)) below an anti-chain \( Z \) in \( \Gamma \) is also optimal on any sub-network "below" an anti-chain \( Y \) in \( \Gamma \) that is itself "below" \( Z \) (among all solutions with \( \psi \)'s behavior on \( \bigcup_{y \in Y} YW^l_v \)).

**Lemma 6** (see Fig. 7) Let \( Y, Z \subseteq V(N) \) be anti-chains in \( \Gamma \) such that \( Y \subseteq \bigcup_{y \in Z} \Gamma_y \). Let \( \mathcal{G} \) be a class of subnetworks of \( N \) and let \( S \subseteq \mathcal{G} \) and \( \psi: V(N) \to C \) such that (a) \( \sum_{y \in Y} \sum_{uw \in A_y(S)} \delta(\psi(u), \psi(w)) \) is minimum among all such \( S \) and \( \psi \). Let \( (S^y)_{y \in Y} \) be a \( Y \)-substitution system for \( \mathcal{G} \) and let \( \psi_y: V(N) \to C \) for each \( y \in Y \) such that (b) \( \psi_y \) and \( \psi \) coincide on \( YW^l_y \). Then

\[
\sum_{y \in Y} \sum_{uw \in A_y(S^y)} \delta(\psi_y(u), \psi_y(w)) \geq \sum_{y \in Y} \sum_{uw \in A_y(S)} \delta(\psi(u), \psi(w)).
\]

**Proof** Towards a contradiction, assume that the lemma is false. We construct \( \psi^*: V(N) \to C \) with

\[
\psi^*(u) = \begin{cases} 
\psi_y(u) & \text{if } u \in \Gamma_y \text{ for any } y \in Y \\
\psi(u) & \text{otherwise}
\end{cases}
\]

Note that \( \psi^* \) and \( \psi \) coincide with \( \psi_y \) on \( YW^l_y \) for all \( y \in Y \). Thus, \( \delta(\psi^*(u), \psi(w)) = \delta(\psi_y(u), \psi(w)) \) if \( uw \in A_y(S^y) \) for any \( y \in Y \) and \( \delta(\psi^*(u), \psi(w)) = \delta(\psi(u), \psi(w)) \), otherwise. Further, we construct a digraph \( S^y := (V(N), (A(S) \setminus \bigcup_{y \in Y} A_y(S^y)) \cup \bigcup_{y \in Y} A_y(S^y)) \) which is in \( \mathcal{G} \) since \( (S^y)_{y \in Y} \) is a \( Y \)-substitution system for \( \mathcal{G} \). Since all \( S^y \) are subnetworks of \( N \), we know that \( \Gamma \) agrees with \( S^y \). Furthermore,
since \( Y \subseteq \bigcup_{z \in Z} \Gamma_z \), we know that each \( y \in Y \) has a \( z \in Z \) with \( y \preceq \Gamma z \). Thus,}

\[
\sum_{z \in Z} \sum_{u \in A_z(S^*)} \delta \psi^*(u, w) = \sum_{z \in Z} \sum_{v \in \Gamma_z} \sum_{u \in A(v)(S^*)} \delta \psi^*(u, w) = \sum_{z \in Z} \sum_{v \in \Gamma_z} \sum_{u \in A(v)(S^*)} \delta \psi^*(u, w) + \sum_{z \in Z} \sum_{v \in \Gamma_z} \sum_{y \in Y} \sum_{u \in A(v)(S^*)} \delta \psi^*(u, w) + \sum_{z \in Z} \sum_{v \in \Gamma_z} \sum_{y \in Y} \sum_{u \in A(v)(S^*)} \delta \psi^*(u, w)
\]

\[
= \sum_{z \in Z} \sum_{v \in \Gamma_z} \sum_{u \in A(v)(S^*)} \delta \psi^*(u, w) + \sum_{z \in Z} \sum_{v \in \Gamma_z} \sum_{y \in Y} \sum_{u \in A(v)(S^*)} \delta \psi^*(u, w) + \sum_{z \in Z} \sum_{v \in \Gamma_z} \sum_{y \in Y} \sum_{u \in A(v)(S^*)} \delta \psi^*(u, w)
\]

contradicting optimality of \( S \) and \( \psi \) (that is, Lemma 6(a)) since \( S^* \in G \).

**Hardwired parsimony**

To compute the hardwired parsimony score at a node \( v \) of \( N \), we require knowledge of the character assigned to \( v \) and its neighbors. For all \( u \in YW^I \), we thus “guess” the character \( \psi(u) \) assigned to \( u \) by an optimal assignment. In our dynamic programming, we scan \( \Gamma \) bottom-up, computing a table entry \( T^{\psi^*}[x, \psi] \) for each \( x \in V(\Gamma) = V(N) \) and each \( \psi : YW^I \rightarrow C \), containing the parsimony cost incurred by all arcs in \( A_x(N) \), assum-
Lemma 7 Let \( x \in V(N) \) and let \( \psi_x : \text{YW}_x^\uparrow \rightarrow C \) with \( \psi_x \subseteq \phi \). Let \( \psi : \phi : V(N) \rightarrow C \) with \( \psi \subseteq \phi \) such that \( \psi \) minimizes \( \sum_{uw \in A_x(N)} \delta(\psi, u, w) \). Then,

\[
T^{\text{HW}}[x, \psi_x] := \min_{c_x \in \psi(x)} \left( \sum_{1 \leq i \leq t} T^{\text{HW}}[v_i, \psi_x[x \rightarrow c_x] | \text{YW}_x^\uparrow_i] + \sum_{z \in \text{Pred}^{-1}_x(s) \cup \text{Succ}^+_x(s)} \delta(c_x, \psi_x(z)) \right) \tag{3}
\]

Proof Sketch. For “\( \geq \)”, we construct a mapping \( \psi' \) from mappings \( \psi \) that are optimal on \( A_x(N) \) among all mappings with \( \psi_i(x) := c_x \). This is possible since all such \( \psi_i \) coincide with \( \psi' \) and \( \psi_x \) on \( \text{YW}_x^\uparrow \). By induction hypothesis, the cost of \( \psi' \) on \( A_x(N) \) is

\[
\sum_{1 \leq i \leq t} T^{\text{HW}}[v_i, \psi' | \text{YW}_x^\uparrow_i] + \sum_{uw \in A_x(N)} \delta(\psi', u, w). \]

Then, “\( \geq \)” follows from optimality of \( \psi_x \) on \( A_x(N) \).

For “\( \leq \)”, it suffices to show that the cost of \( \psi \) on \( A_x(N) \) is equal to the result of setting \( c_x := \psi(x) \) in the right hand side of (3) (which is a valid choice for the minimum since \( \psi(x) \in \phi(x) \)). First, the cost of \( \psi \) on \( A_x(N) \) is

\[
T^{\text{HW}}[v_i, \psi' | \text{YW}_x^\uparrow_i] \]

by independence of sub-solutions and the induction hypothesis. Second, the cost of \( \psi \) on \( A_x[N] \) is

\[
\sum_{z \in \text{Pred}^{-1}_x(s)} \delta(c_x, \psi_x(z)) \]

and the cost of \( \psi \) on \( A_x[N] \) is

\[
\sum_{z \in \text{Succ}^+_x(s)} \delta(c_x, \psi_x(z)) \]

since \( \psi \) and \( \psi_x \) coincide on \( \text{YW}_x^\uparrow \).

□

In order to solve the hardwired parsimony problem given \( N, \phi \) and \( \Gamma \), all we have to do is compute \( T^{\text{HW}}[x, \psi_x] \) for each \( x \) bottom-up in \( \Gamma \) and each of the (at most) \(|C| \cdot |\text{YW}_x^\uparrow| \) many choices of \( \psi_x : \text{YW}_x^\uparrow \rightarrow C \) with \( \psi_x \subseteq \phi \). Then, by Lemma 7, the hardwired parsimony score of \( N \) with respect to \( \phi \) can be read from \( T^{\text{HW}}[\rho_{\Gamma}, \phi] \). To compute \( T^{\text{HW}} \), the sum over the children of \( x \) for all \( x \in V(N) \) in (3) can be computed in amortized \( O(|A(N)|) \) time and, with a bit of bookkeeping, it is possible to maintain the value of the second sum in (3) in \( O(|A(N)|) \) amortized time per choice of \( \psi \). Then the following holds:

Theorem 1 Given a network \( N \), some \( \phi : V(N) \rightarrow 2^C \) and a tree \( \Gamma \) agreeing with \( N \), the hardwired parsimony score of \( (N, \phi) \) can be computed in \( O(|C|^{|\text{YW}(\Gamma)|+1} \cdot |A(N)|) \) time.

Proposition 1 lets us turn tree decompositions of \( N \) into trees \( \Gamma \) agreeing with \( N \), allowing us to replace \( \text{yw}(\Gamma) \) by \( \text{tw}(N) \), incurring an additional running time of \(|N| \cdot 2^{O(\text{tw}(N)^3)} \) [13].

Corollary 1 Let \( (N, \phi) \) be an instance of HARDWIRED PARSIMONY. Let \( t \geq \text{tw}(N) \) and let \( \Gamma \) be the time in which a width-\( t \) tree decomposition of \( N \) can be computed. Then, the hardwired parsimony score of \( (N, \phi) \) can be computed in \( O(T + |C|^{t+1} \cdot |A(N)|) \) time.

Softwired parsimony

In contrast to the hardwired parsimony score, where the computation of the cost of the incident edges of a node \( x \) only required knowledge of the characters assigned to neighbors of \( x \), computing the softwired score additionally requires knowledge of which parent of \( x \) remains a parent in the sought switching. A table entry \( T^{\text{SW}}[x, \ldots] \) contains the smallest combined cost of all arcs in \( A_x(S) \) for a switching \( S \) of \( N \) minimizing this cost. To be able to compute an entry for \( x \in V(N) \), we not only need to “guess” \( \psi_x \) but, additionally, some representation of the switching \( S \). In particular, in \( S \), no child of \( x \) may have another parent than \( x \). However, since children of \( x \) in \( N \) may be above \( x \) in \( \Gamma \), we have to “guess” which children of \( x \) in \( N \) are still children of \( x \) in \( S \). Such a guess manifests itself as an additional index \( R^x \) of the dynamic programming table (note that we clearly only have to store this information for children of \( x \) that are reticulations). Indeed, this information has to be stored for all nodes considered below \( x \) who still have children in \( \text{YW}_x^\uparrow \). Thus, we index our DP-table also by a subset \( R^x \subseteq \text{YW}_x^\uparrow \cap \text{R}(N) \) containing a reticulation \( r \in \text{R}(N) \) if and only if \( \Gamma \) contains a parent \( v \) of \( r \) and \( v \) is an arc of an optimal switching \( S \) for \( N[\Gamma_x \cup \text{YW}_x^\uparrow] \).

Definition 6 Let \( \Gamma \) be a tree that agrees with \( N \), let \( x \in V(N) \), let \( \psi_x : \text{YW}_x^\uparrow \rightarrow C \) with \( \psi_x \subseteq \phi \), and let \( R^x \subseteq \text{Succ}^-_{\Gamma_x}(x) \). Let \( v_1, v_2, \ldots, v_l \) denote the children of \( x \) in \( \Gamma \) (\( t = 0 \) if \( x \) is a leaf in \( \Gamma \)). Then, set
\[ T^{SW}[x, \psi_x, R^z] := \min_{c_x \in \phi(x)} \min_{R^y \subseteq R^z \cap \text{Succ}_N^{R^z}(x)} \sum_{r \in R^y \cap \text{Succ}_N^{R^z}(x)} \delta(c_x, \psi_x(r)) \]

\[
\quad + \min \left\{ \begin{array}{ll}
Q^{\psi_x}_{x,c_x}[t, R^y \setminus R^*] + \min_{y \in \text{Pred}_N^{R^z}(x)} \delta(c_x, \psi_x(y)) & \text{if } \text{Pred}_N^{R^z}(x) \neq \emptyset \\
\bigcup \{ (x) \cap R(N) \} & \text{if } \text{Pred}_N^{R^z}(x) = \emptyset
\end{array} \right. \]

(4)

where

\[ Q^{\psi_x}_{x,c_x}[i, R^*] := \begin{cases} 
\min_{R^y \subseteq R^z \cap \text{Succ}_N^{R^z}(x)} Q^{\psi_x}_{x,c_x}[i-1, R^y \setminus R^*] + T^{SW}[v_i, \psi_x, R^*] & \text{if } i \neq 0 \\
0 & \text{if } i = 0 \text{ and } R^* = \emptyset \\
\infty & \text{otherwise}
\end{cases} \]

(5)

Lemma 8 Let \( \Gamma \) be a tree that agrees with \( N \), let \( x \in V(N) \), let \( \psi_x : \text{YW}_{W^i}^r \rightarrow C \) with \( \psi_x \subseteq \phi \), and let \( R^x \subseteq \text{Succ}_N^{R^z}(x) \). If \( \text{Succ}_N^{R^z}(x) = \emptyset \), then \( T^{SW}[x, \psi_x, R^x] = \infty \). Otherwise, let \( S \in \text{Succ}_N^{R^z}(N) \) and \( \psi : V(N) \rightarrow C \) such that \( \psi_x \subseteq \psi \subseteq \phi \) and \( \sum_{R \in S(x)} \delta(\psi(x), w) \) is minimum among all such \( S \) and \( \psi \).

Then,

\[ T^{SW}[x, \psi_x, R^z] = \sum_{R \in \text{Succ}_N^{R^z}(x)} \delta(\psi(x), w). \]

(6)

Proof Sketch. Let us abbreviate \( Z_i := \bigcup_{j \geq i} V(\Gamma_j) \). We first show that the table \( Q \) does what we expect it to do.

Claim 2 \( Q^{\psi_x}_{x,c_x}[i, R^*] = \sum_{R^y \subseteq R^z \cap \text{Succ}_N^{R^z}(x)} \sum_{R \in S(x)} \min_{R \in S(x)} \delta(\psi(x), w) \) for optimal \( S \in \text{Succ}_N^{R^z}(x) \) and \( \psi_x \) coincides with \( \psi_x[x \rightarrow c_x] \) on \( \bigcup_{j < i} \text{YW}_{W^i}^r \).

Proof Sketch. For \( \leq \), let \( R^* \subseteq R^z \cap \text{Succ}_N^{R^z}(x) \) such that equality holds in (5). We consider a switching \( S' \in \text{Succ}_N^{R^z}(x) \) constructed from switchings \( S_{i-1} \subseteq \text{Succ}_N^{R^z}(x) \) and \( S^* \in \text{Succ}_N^{R^z}(x) \) as well as a mapping \( \psi^* \) coinciding with \( \psi_x[x \rightarrow c_x] \) on \( \bigcup_{j < i} \text{YW}_{W^i}^r \). By induction hypotheses, these costs are \( Q^{\psi_x}_{x,c_x}[i-1, R^y \setminus R^*] \) and \( T^{SW}[v_i, \psi_x, R^*] \) for \( i \neq 0 \). Then, \( \leq \) follows from the fact that \( R^* \) is only one of the possible choices for the minimum in (5).

For \( \geq \), let \( c_x \in \phi(x) \). We consider a switching \( S' \in \text{Succ}_N^{R^z}(x) \) constructed from switchings \( S_{i-1} \subseteq \text{Succ}_N^{R^z}(x) \) and \( S^* \subseteq \text{Succ}_N^{R^z}(x) \) such that \( \text{Pred}_N^{R^z}(x) \neq \emptyset \). Then, \( \geq \) follows from the fact that our choices of

In the following, for any anti-chain \( X \) in \( \Gamma \) and all \( Z \subseteq \bigcup_{x \in X} \text{YW}_{W^i}^r \), let \( S^X \rightarrow Z(N) \) denote the set of all switchings \( S \) of \( N \) with \( \text{Succ}_N^{R^z}(X) = Z \).

Proof. By optimality of \( S \) and \( \phi \).

For \( \geq \), we let \( R^* := \text{Succ}_N^{R^z}(\Gamma_r) \) and use independence of sub-solutions and the induction hypotheses to show that the cost of \( \phi_i \) on \( A_{Z_{i-1}}(S_i) \) is \( Q^{\psi_x}_{x,c_x}[i-1, R^y \setminus R^*] \) and the cost of \( \phi_i \) on \( A_{Z_i}(S_i) \) is \( T^{SW}[v_i, \psi_x, R^*] \). Then, \( \geq \) follows from the fact that \( R^* \) is only one of the possible choices for the minimum in (5).

□
$c_x$ and $R^*$ are one of the possible choices for the minimum in (4).

In order to solve the softwired parsimony problem given $N, \phi$ and $\Gamma$, all we have to do is compute $T^{SW}[x, \psi, R^*]$ for each $x$ bottom-up in $\Gamma$, each of the (at most) $|C||YW|^r_x$ many choices of $\psi : YW^r_x \to C$ with $\psi_x \subseteq \phi$, and each $R^* \subseteq Succ^x_N(x) \subseteq YW^r_x \cap R(N)$. To this end, $Q_x^{\psi^x}[i, R^* \setminus R^*]$ and $Q_x^{\psi^x}[i, (R^* \setminus R^* ) \cup \{x\}]$ have to be computed for each child $v_i$ of $x$ in $\Gamma$ and each $R^* \subseteq R^* \cap Succ^x_N(x)$. Then, by Lemma 8, the softwired parsimony score of $N$ with respect to $\phi$ can be read from $T^{SW}[\rho_{\Gamma}, \emptyset, \emptyset]$. In the following, let $\psi_x$ be fix. Then, for $x \in N$, we can compute $Q_x^{\psi^x}[i, R^*]$ for all choices of $x, i$ and $R^*$ in $O(2|YW|^r_x | Succ^x_N(v_i)| + \sum_{x \in \Gamma} |Succ^x_{\Gamma}(x)|)$ time. Further, the values of $\min_{y \in Pred_{\Gamma}^N(x)} \delta(c_x, \phi_x(y))$ can be pre-computed for all $x \in \Gamma$ in $O([A(N)])$ time. Then, to compute $T^{SW}[x, \psi_x, R^*]$ for all $x \in N$, we have to check $|V(N)|$ choices for $x$, as well as $|\phi(x)| \leq |C|$ choices for $c_x$ and $3|Succ^x_N(x)|$ choices for $R^*$ and $R^* \subseteq R^*$ combined. Altogether, the table $T^{SW}$ can be computed in $O(|C||YW|^r_x | (3|YW|^r_x | |C| | |V(N)| + |A(N)|))$ time. The computation of $Q_x^{\psi^x}$ in $O(2|YW|^r_x + |A(N)|)$ time is absorbed by this. For practical purposes, note that estimating $|Succ^x_N(x)| \leq |YW|^r_x$ is quite crude and equality will almost never be attained. Then, the following result holds:

**Theorem 2** Given a network $N, \phi : V(N) \to 2^C$ and a tree $\Gamma$ agreeing with $N$, the softwired parsimony score of $(N, \phi)$ can be computed in $O(|C||YW|^r_{\Gamma} | (3|YW|^r_{\Gamma} | |C| | |V(N)| + |A(N)|))$ time.

Again, we can replace $yw(\Gamma)$ by $tw(N)$ using Proposition 1.

**Corollary 2** Let $(N, \phi)$ be an instance of SOFTWIRED PARSIMONY. Let $t \geq tw(N)$ and let $T$ be the time in which a width-$t$ tree decomposition of $N$ can be computed. Then, the softwired parsimony score $(N, \phi)$ can be computed in $O(T + |C|^t | (3^t | |C| | |V(N)| + |A(N)|))$ time.

**Parental parsimony**

For ease of presentation, we introduce some additional notation. First, for any $a$ and $b$, we abbreviate $\max\{a - b, 0\} := a - b$. Let $\psi$ and $\psi'$ be functions. If $\psi$ maps all items to $\emptyset$ or to $0$, then we say that $\psi$ is a zero-function and we write $\psi = \emptyset$. We use $\psi - \psi'$ to denote the function defined on the domain of $\psi$ for which $(\psi - \psi')(x) = \psi(x) - \psi'(x)$ if $\psi'(x) = \emptyset$ and $(\psi - \psi')(x) = \psi(x) - \psi'(x)$, otherwise. This definition extends to functions mapping to sets in a natural way.

Each finite-cost lineage function $f$ corresponds to a phylogenetic tree “embedded” in $N$ whose branches are called lineages (see Fig. 1(right)). For each $x \in V(N)$, $f(x)$ represents the set of such lineages passing through $x$. Each such lineage may “choose” a parent among the parents of $x$ in $N$. This models the biological circumstance that a character trait may be inherited from any parent. We compute (the cost of) an optimal lineage function on $N$ using a tree $\Gamma$ that agrees with $N$. To compute $cost_f(x)$, we require knowledge of $\sum_{y \in Pred_{\Phi}(x)} f(y)$ as well as $\bigcup_{y \in Pred_{\Phi}(x)} f(y)$ (see Definition 4). We partition the predecessors of $x$ over which the formula iterates into those above $x$ in $\Gamma$ and those below (since $\Gamma$ agrees with $N$, all predecessors of $x$ in $N$ are comparable to $y$ in $\Gamma$). For all $y \in YW^r_y$, we thus store

1. the set $\lambda(x) := f(y)$ of lineages in $y$,
2. the subset $\psi(x)$ of lineages of $y$ that also occur in parents (in $N$) of $y$ that are below $x$ in $\Gamma$, that is, in $Pred^\Gamma_{\Phi}(y)$ (such lineages are inherited by $y$ at no cost),
3. the total number $\eta(y)$ of lineages of $y$ that can be inherited from parents (in $N$) of $y$ that are below $x$ in $\Gamma$, that is, from $Pred^\Gamma_{\Phi}(y)$ (cost 0 or 1).

Then, in order to compute an entry $T^{SW}[x, \lambda_x, \psi_x, \eta_x]$, we “guess” the set $U \subseteq \phi(x)$ of lineages passing through $x$ in an optimal solution, as well as the set $D \subseteq U$ of lineages inherited from nodes in $Pred^\Gamma_{\Phi}(x)$. This allows us to infer $\eta(x) = |\lambda(x)| - \sum_{r \in Pred^\Gamma_{\Phi}(x)} \lambda(r)$ and $\psi(x) := D$. Then, by Definition 4, the cost incurred by $f$ on $x$ can be computed from $\sum_{y \in Pred_{\Phi}(x)} f(y) = \eta(x) + \sum_{y \in Pred_{\Phi}(x)} \lambda(r)$ and $\bigcup_{y \in Pred_{\Phi}(x)} f(y) = \psi(x) \cup \bigcup_{y \in Pred_{\Phi}(x)} \lambda(r)$.

We will compute table entries for $x$ using the already computed table entries for the children $v_i$ of $x$ in $\Gamma$. In these lookups, we have $x \in YW^r_v$ so, to be consistent with the semantics, we have to make sure that $\lambda(x) = U$, $\psi(x) = D$, and that all lineages of $x$ that are not inherited from $Pred^\Gamma_{\Phi}(x)$ can be inherited from $Pred^\Gamma_{\Phi}(x)$, that is, $\eta(x) = |\lambda(x)| - \sum_{r \in Pred^\Gamma_{\Phi}(x)} \lambda(r)$.

Further, each child $y$ of $x$ in $N$ may inherit a lineage from $x$ and, if $y$ is above $x$ in $\Gamma$, this has to be registered by removing the lineages of $U$ from $\psi(y)$ and subtracting $|U|$ from $\eta(y)$. Finally, the lineages represented by $\psi$ and $\eta$ are distributed among the children of $x$ in $\Gamma$ using the table $Q$. In the following, in order to avoid treating the case that $x = \rho_N$ separately, we define $\rho(x) := 1 - \delta(x, \rho_N)$, that is, $\rho(x) = 1$ if and only if $x = \rho_N$. 
Definition 7 Let $\Gamma$ be a tree that agrees with $N$, let $x \in V(N)$, let $\lambda_x : YW^\Gamma_x \rightarrow 2^C$ with $\lambda_x \leq \phi$ and let $\psi_x \leq \lambda_x$. Let $\{v_1, v_2, \ldots, v_t\} = \text{Succ}_x(t) = \emptyset$ if $x$ is a leaf in $\Gamma$. Then, set $T^{PT}[x, \lambda_x, \psi_x, \eta_x]$ to

$$\min_{D \subseteq U \subseteq \phi(x)} Q^i_{x} = \min_{\psi} \left[ \begin{array}{l} Q_{x}^i = \left| I - 1, \psi - \psi', \eta - \eta' \right| + T^{PT}[v_i, \lambda, YW^\Gamma_{v_i}, \psi', \eta'] \text{ if } i > 0 \\ -\rho(x) \text{ if } i = 0 \text{ and } \psi = \emptyset \text{ and } \eta = \emptyset \\ \infty \text{ otherwise} \end{array} \right]$$

where $Q^i_{x}[i, \psi, \eta]$ equals

$$\min_{\psi' \leq \psi \leq \lambda_x} Q^i_{x}[i - 1, \psi - \psi', \eta - \eta'] + T^{PT}[v_i, \lambda, YW^\Gamma_{v_i}, \psi', \eta'] \text{ if } i > 0$$

$$\min_{\eta' \leq \eta \leq \phi} \left| I - 1, \psi - \psi', \eta - \eta' \right| + T^{PT}[v_i, \lambda, YW^\Gamma_{v_i}, \psi', \eta'] \text{ if } i > 0$$

$$\min_{\eta' \leq \eta \leq \phi} \left| I - 1, \psi - \psi', \eta - \eta' \right| + T^{PT}[v_i, \lambda, YW^\Gamma_{v_i}, \psi', \eta'] \text{ if } i > 0$$

Lemma 9 Let $x \in V(N)$, let $i \in \mathbb{N}$, let $\lambda : YW^\Gamma_x \rightarrow 2^C$, let $\eta, \eta' : YW^\Gamma_x \rightarrow \mathbb{N}$, and let $\psi, \psi' : YW^\Gamma_x \rightarrow 2^C$ such that $\psi' \leq \psi \leq \lambda$ and $\overrightarrow{x} = \emptyset \text{ if } x = \rho(x) \leq \eta' \leq \eta$. Then,

$$T^{PT}[x, \lambda, \psi, \eta] \leq T^{PT}[x, \lambda, \psi, \eta'] \text{ and } Q^i_{x}[i, \psi, \eta] \leq Q^i_{x}[i, \psi, \eta']$$

Proof Sketch. The lemma can be proved by induction on the height of $x$ in $\Gamma$ and the value of $i$. If $x$ is a leaf, then $Q^i_{x}[0, \psi, \eta]$ is finite only if $\psi = \emptyset$ and $\eta = \emptyset$ [x = \rho(x)]$, implying the second inequality. For monotony of $T^{PT}$, fix the sets $D \subseteq U \subseteq \phi(x)$ for which the minimum in the formula of $T^{PT}[x, \lambda, \psi, \eta]$ is attained. Then, by monotony of $Q^i_{x}$, replacing $\psi$ by $\psi'$ and $\eta$ by $\eta'$ in this formula does not increase its value and this value is at most $T^{PT}[x, \lambda, \psi', \eta']$ since it is obtained for one of several possible choices for $D$ and $U$. If $x$ is not a leaf in $\Gamma$ then monotony of $Q^i_{x}[i, \ldots, \ldots]$ is implied by monotony of $Q^i_{x}[i - 1, \ldots, \ldots]$ and monotony of $T^{PT}$ for the children $v$ of $x$. Finally, monotony of $T^{PT}$ follows from monotony of $Q^i_{x}$ as in the induction base. \)

Lemma 10 Let $\Gamma$ be a tree agreeing with $N$, let $x \in V(N)$, let $\lambda_x, \psi_x : YW^\Gamma_x \rightarrow 2^C$ and $\eta_x : YW^\Gamma_x \rightarrow \mathbb{N}$. Let $f$ minimize $\text{cost}(f)$ among all lineage functions in $\mathcal{LF}_{N, \phi}$ such that, for all $w \in YW^\Gamma_x$, $\lambda_x(w) = f(w)$, $\psi_x(w) = f(w) \cap \bigcup_{u \in \text{Pred}^\Gamma_x(w)} f(u)$, and $\eta_x(w) \leq \sum_{u \in \text{Pred}^\Gamma_x(w)} f(u)$. If there are no such $f$,

$$T^{PT}[x, \lambda, \psi, \eta] = \sum_{x \subseteq x} \text{cost}_f(z)$$

Proof Sketch. Let us abbreviate $Z_i := \bigcup_{x \leq i} V(\Gamma_i)$. We first show that the table $Q$ does what we expect it to do.

Claim 3 Let $\lambda, \psi : YW^\Gamma_x \cup \{x\} \rightarrow 2^C$ and $\eta : YW^\Gamma_x \cup \{x\} \rightarrow \mathbb{N}$ such that $\psi \leq \lambda \leq \phi$. Let $f_i \in \mathcal{LF}_{N, \phi}$ have minimum cost on $\bigcup_{x \leq i} \Gamma_i$ among all lineage functions satisfying (a)−(c) for $i - 1$. Let $f^* \in \mathcal{LF}_{N, \phi}$ minimize $\sum_{x \leq i} \text{cost}_{f^*}(u)$ among all lineage functions satisfying (a)−(c) for $i - 1$. Let $f^* \in \mathcal{LF}_{N, \phi}$ minimize $\sum_{x \leq i} \text{cost}_{f^*}(u)$ among all lineage functions that, for all $w \in YW^\Gamma_x$, satisfy $\lambda(w) = f^*(w)$, $\psi(w) = f^*(w) \cap \bigcup_{u \in \text{Pred}^\Gamma_x(w)} f^*(u)$, and $\eta(w) = \sum_{u \in \text{Pred}^\Gamma_x(w)} f^*(u)$. By induction hypotheses, the cost of $f_i - 1$ on $Z_i$ is $Q^i_{x}[i - 1, \psi - \psi', \eta - \eta']$.
and the cost of \( f^* \) on \( \Gamma_i \) is \( T^{PT}[\nu_i, \lambda | YW_{\psi}^i, \psi', \eta'] \). From \( f_{i-1} \) and \( f^* \), we construct a lineage function \( f' \in \mathcal{LC}_{N,\phi} \) whose cost on \( Z_{i-1} \) is \( \sum_{\tau \subseteq \subset} \text{cost}_{\tau \subseteq \nu_i} + \sum_{\tau \subseteq \nu_i} \text{cost}_{\tau \subseteq \nu_i} \). Then, \( \geq \) follows by optimality of \( f_i \) on \( Z_i \).

For \( \leq \), let \( \psi' \) and \( \eta' \) be such that, for all \( w \in YW_{\psi}^i \), we have \( \psi'(w) = f_i(w) \cap \bigcup_{u \in \text{Pred}_{N}^{i}(w)} f_i(w) \cup \psi(w) \) and \( \eta'(w) = \sum_{u \in \text{Pred}_{N}^{i}(w)} \psi_i(w) \). By independence of sub-solutions, \( f_i \) is optimal on \( Z_{i-1} \) and on \( \Gamma_i \) so, by induction hypotheses, the cost of \( f_i \) on \( Z_{i-1} \) is \( Q_{\phi}^a[i, \psi, \eta] \) computed for each \( x, \psi, \lambda, \psi, \eta \) by making at most \( 2|C||YW_{\phi}^x| \cdot |C||YW_{\phi}^x| \) queries to \( Q_{\phi}^a[i, \psi, \eta] \) and \( T^{PT} \). As there are \( O(|A(N)|) \) valid combinations of \( x \) and \( i \), the table \( Q \) can be computed in \( O(|A(N)| \cdot 3^{|C||YW_{\phi}^x|} \cdot |C||YW_{\phi}^x| \cdot 2^{|C||YW_{\phi}^x|} \cdot 4^{|YW_{\phi}^x|} \cdot 4 \) time. Further, computing each \( T^{PT}[x, \psi, \lambda, \psi, \eta] \) requires testing \( 3^{(\phi(x))} \leq 3^{|C|} \) choices for \( D \subseteq U \subseteq \phi(x) \) and computing \( |U \cap \bigcup_{u \in \text{Pred}_{N}^{i}(x)} \lambda_u(u)\big| \) in \( O(1) \) time (we precompute \( \bigcup_{u \in \text{Pred}_{N}^{i}(x)} \lambda_u(u) \) for each fix \( x \) and \( \lambda_u \)). Thus, the table \( Q \) can be computed in \( O(3^{|C||YW_{\phi}^x|} \cdot |C||YW_{\phi}^x|) \cdot \big( (|C||YW_{\phi}^x|)^{\phi(x)} + 3^{|C|} + |A(N)|) \big) \) time, which is dominated by the construction of \( Q \).

Theorem 3 Given a network \( N, \phi : V(N) \to 2^C \) and a tree \( \Gamma \) agreeing with \( N \), the parental parsimony score of \((N, \phi)\) can be computed in \( O(6^{|YW_{\phi}^x|} \cdot |C| \cdot 4^{4 \log |C|} \cdot |A(N)|) \) time.

Again, we can replace \( w(x) \) by \( tw(x) \) using Proposition 1.

Corollary 3 Let \((N, \phi)\) be an instance of PARENTAL PARSIMONY. Let \( t \geq tw(N) \) and let \( T \) be the time in which a width-\( t \) tree decomposition of \( N \) can be computed. Then, the parental parsimony score of \((N, \phi)\) can be computed in \( O(T + 6^{|C|} \cdot 4^{4 \log |C|} \cdot |A(N)|) \) time.

Note that the parental parsimony setting supports assigning multiple states to a character to a single species, thereby modeling species carrying multiple alleles of a single gene. By forcing \( D \subseteq U \subseteq \phi(x) \) instead of \( D \subseteq U \subseteq \phi(x) \) if \( x \) is a leaf, we can trivially modify our dynamic programming to explain multiple character states in extant species.

Corollaries 1, 2 and 3 give the running times of our algorithms as depending on the treewidth of \( N \). The state-of-the-art solutions for HARDWIRED PARSIMONY, SOFTWARED PARSIMONY and PARENTAL PARSIMONY have the following respective running times: \( O(3^{|C|} \cdot |V(N)| \cdot |A(N)|) \) \[9\], \( O(2^{|C|} \cdot |V(N)| \cdot |A(N)|) \) \[8\] and \( O(2^{|C|} \cdot |V(N)|) \) \[12\]. Since the scanwidth of \( N \) is
potentially much smaller than its level $\ell$ [28], and the
treewidth of $N$ is smaller than its scanwidth [20], we have
$\text{tw}(N) - 1 \leq \ell \leq r$. Thus, we expect that there will be
several cases where our algorithms will be faster than the
current best-known ones.

**Discussion**

In this paper, we focused on the small version of the
parsimony problem for networks given a specific position
in the genome. When markers can be assumed to
be independent, as it is the case when a certain distance
is preserved between genomic locations included in the
matrix, each position can be analyzed separately, and the
parsimony score of a network w.r.t. the matrix is simply
the sum of the parsimony scores of the network for each
genomic location. Thus, the algorithms presented here
can be easily expanded to several independent genomic
locations. Moreover, our formulations are defined for
networks that are not necessarily binary, can account for
polymorphism and can impose restrictions on ancestral
states. As discussed above, our algorithms can be orders
of magnitude faster than the state-of-the-art solutions.

A comparison of the reticulation number, the level, the
scanwidth and the treewidth for practically relevant
classes of networks would thus be an interesting project
for future work.

Our results are slightly overshadowed by the fact that
optimal tree decompositions are very hard to compute.
However, practical exact and approximative algorithms
are available today and we expect them to perform well,
as phylogenetic networks can be expected to only be
moderately tangled.

$$T^{\nabla\mathrm{HW}}[x, \psi_x] = \min_{c_x \in \phi(x)} \sum_{z \in \text{Pred}_V(x) \cup \text{Succ}_V(x)} \delta(c_x, \psi_x(z))$$

$$= \min_{c_x \in \phi(x)} \left( \sum_{x \in A_\ell(N)} \delta(c_x, \psi_x(z)) + \sum_{x \in A_\ell(N)} \delta(c_x, \psi_x(z)) \right)$$

$$= \min_{c_x \in \phi(x)} \sum_{uw \in A_e(N)} \delta(\psi_x[x \to c_x], (u, w))$$

(9)

Furthermore, closer inspection of our dynamic pro-
gramming formulations (most prominently Definition 6)
unveils that their computation is faster when the
maximum number of reticulations in each bag is small.
Thus, it would be interesting to be able to compute tree
decompositions in which this quantity is low, to the point
where one could improve running time of the algorithm
by sacrificing optimality of the decomposition in favor
of reducing this “reticulation density”. Research in this
direction is, to the best of our knowledge, limited to a

paper by Bichoore and Bodlaender [29], considering tree
decompositions minimizing a weight function over the
bags.

The ability to fast-score phylogenetic networks under
the parsimony framework could be a big help in designing
likelihood-based heuristics or bayesian methods to infer
networks from independent markers [28, 30] by
providing fast heuristics to compute the initial networks
with which to start the likelihood or bayesian search, or
to design fast local-search techniques.

In the future, we would like to tackle the small parsi-
mony problem for several dependent genomic locations
e.g. a gene). Little is known for this problem, except that
it stays NP-hard even for binary characters on level-1
networks [31] and that it is fixed-parameter tractable in
the number of reticulations of the network [6]. Another
important direction would be to study the big parsimony
problem, which is currently wide open, even lacking a
consensus of the definition of optimality [6, 32–34].

**Appendix**

**Lemma 7** Let $x \in V(N)$ and let $\psi_x : YW^x \to C$ with
$\psi_x \leq \phi$. Let $\psi : V(N) \to C$ with $\psi \leq \phi$ such that $\psi$
minimizes $\sum_{uw \in A_e(N)} \delta(\psi(u, w))$

$$T^{\nabla\mathrm{HW}}[x, \psi_x] = \sum_{uw \in A_e(N)} \delta(\psi(u, w))$$

**Proof** The proof is by induction on the height of $x$ in $\Gamma$.
For the induction base, suppose that $x$ is a leaf in $\Gamma$
and note that $A_e(N) = A_{[x]}(N)$ in this case. Then, (3) simpli-
fies to

$$\psi(x) \in \phi(x),$$

we know that $\psi(x)$ participates in the
minimum in (9), implying the “$\leq$"-direction. For the “$\geq$"-
direction, assume that $T^{\nabla\mathrm{HW}}[x, \psi_x] < \sum_{uw \in A_e(N)} \delta(\psi(u, w))$. By
(9), there is some $c_x \neq \psi(x)$ with $c_x \in \phi(x)$ and
$\sum_{uw \in A_e(N)} \delta(\psi_x[x \to c_x], (u, w))$ $< \sum_{uw \in A_e(N)} \delta(\psi(u, w))$. Since
$c_x \in \phi(x)$, we still have $\psi_x \leq \psi_x[x \to c_x] \leq \phi$, contradict-
ing optimality of $\psi$ on $A_e(N)$. For the induction step, sup-
pose that $t > 0$ and consider both directions separately.
“≤”: Let \( i \leq t \), and let \( \psi_i := \psi \mid_{YW^i} \). Then, by Lemma 6 (with \( Z = \{ x \} \), \( Y = \{ v_i \} \), \( G = \{ N \} \) and \( S') \in Y = (N) \in Y \)), optimality of \( \psi \) on \( A_\delta(N) \) implies optimality of \( \psi_i \) on \( A_\delta(N) \). Thus, we can use the induction hypothesis on \( T^{\mathcal{N}_i}[v_i, \psi_i] \). Since \( \psi \) participates in the minimum of (3),

\[
T^{\mathcal{N}_i}[v_i, \psi_i] \leq \sum_{1 \leq i \leq t} T^{\mathcal{N}_i}[v_i, \psi_i] + \sum_{u \in \text{Pred}_A(N)} \delta(\psi(z), \psi(z)) \]

Since, by assumption, \( T^{\mathcal{N}_i}[x, \psi_x] \) is strictly less than the cost of \( \psi \) on \( A_\delta(N) \), we conclude that the cost of \( \psi' \) on \( A_\delta(N) \) is strictly less than that of \( \psi \), contradicting optimality of \( \psi \). \( \square \)

**Lemma 8** Let \( \Gamma \) be a tree that agrees with \( N \), let \( x \in V(N) \), let \( \psi_x : YW^x \to C \) with \( \psi \leq \phi \), and let \( R^x \subseteq \text{Succ}_N^x \Gamma_\delta(N) \).

If \( S^{\mathcal{N}_i \to R^x}(N) = \emptyset \), then \( T^{\mathcal{N}_i}[x, \psi_x, R^x] = \infty \). Otherwise, let \( S \in S^{\mathcal{N}_i \to R^x}(N) \) and \( \psi : V(N) \to C \) such that (a) \( \psi \leq \phi \) and (b) \( \sum_{u \in A_\delta(S)} \delta(\psi(u, w)) \) is minimum among all such \( S \) and \( \psi \). Then,

\[
T^{\mathcal{N}_i}[x, \psi_x, R^x] \leq \sum_{u \in A_\delta(S)} \delta(\psi(u, w)).
\]

**Proof** Note that arcs that are incoming to tree nodes cannot be switched off and, thus, \( \text{Succ}_N^x \Gamma_\delta(z) = \text{Succ}_N^x \Gamma_\delta(z) \) for all \( z \in V(N) \) and all switchings \( S' \in S(N) \). The proof is by induction on the height of \( x \) in \( \Gamma \).

**Case 1:** \( x \) is a leaf in \( \Gamma \), that is, \( t = 0 \). First, note that \( R^x \subseteq \text{Succ}_N^x \Gamma_\delta(x) \) and no \( r \in R^x \subseteq R(N) \) can have all their parents in \( \Gamma_\delta(x) = \{ x \} \), thus implying \( S^{\mathcal{N}_i \to R^x}(N) \neq \emptyset \). Next, let \( y \) be the predecessor of \( x \) in \( \Gamma \) and note that \( y \in \text{Pred}_A(N) \). Further, \( y \) minimizes \( \delta(\psi(y, x)) \) among all \( y \in \text{Pred}_A(N) \) as, otherwise, we can construct...
a new switching $S' \in S^{Y, R^*}(N)$ by replacing $xy$ by some $y'x$ with $y' \in \text{Pred}_N(x)$, thereby contradicting (b). Clearly, $\text{Pred}_N^+(x) = \emptyset$ and $Q^{Y}_{A^*}$ \([0, R^* \setminus R^*] \neq \infty \) only if $R^* = R^*$. Thus,

$$T_{SW}^{[x, \psi], R^*} = \min_{c_x \in \phi(x)} \left( \sum_{r \in R^* \setminus \text{Succ}_{N}^{+}(x)} \delta(c_x, \psi(x)) + \min_{y \in \text{Pred}_N^+(x)} \delta(c_x, \psi(y)) \right)$$

and there is some $c_x \in \phi(x)$ such that equality holds if

$$\psi(x) = c_x.$$  

Let $\psi^* := \psi[x \rightarrow c_x]$ be the result of changing the assignment of $x$ to $c_x$ in $\psi$ and note that $\psi_x \leq \psi^*$. Clearly, we still have $S \in S^{Y, R^*}(N)$. Thus,

$$T_{SW}^{[x, \psi], R^*} = \sum_{r \in R^* \setminus \text{Succ}_{N}^{+}(x)} \delta(c_x, \psi(x)) + \min_{y \in \text{Pred}_N^+(x)} \delta(c_x, \psi(y))$$

Case 2: $x$ has children $v_1, v_2, \ldots, v_t$ in $\Gamma$. Recall that we suppose that $x \in \bigcup_{y \in Y} Y_{v_y}$ by Lemma 3. For all $S^* \in S(N)$ and all anti-chains $Y$ in $\Gamma$, abbreviate $S^* \cap \bigcup_{y \in Y} \text{Succ}_{S^*}^{+}(\Gamma_y)(N) =: S^{Y, S^*}(N)$, that is, roughly, the set of switchings of $N$ with the same “behavior” as $S^*$ on $Y$. The proof of Case 2 relies on the independence of partial solutions established by Lemma 6 with $\mathcal{G} = S^{Y, S^*}(N)$. To apply Lemma 6, we show that any set of switchings $S^*$ such that $\{\text{Succ}_{S^*}^{+}(\Gamma_y) \mid y \in Y\}$ is a partition of $\bigcup_{y \in Y} \text{Succ}_{S^*}^{+}(\Gamma_y)$ is a $Y$-substitution system for $S^{Y, S^*}(N)$.

Claim 4 Let $S^* \in S(N)$ and let $Y$ be an anti-chain in $\Gamma$. For each $y \in Y$, let $S^*_y \in S(N)$ such that $\{\text{Succ}_{S^*_y}^{+}(\Gamma_y) \mid y \in Y\}$ is a partition of $\bigcup_{y \in Y} \text{Succ}_{S^*_y}^{+}(\Gamma_y)$. Let

$$S' := \left( V(N), \left(A(S^*) \setminus \bigcup_{y \in Y} A_y(S^*)\right) \cup \bigcup_{y \in Y} A_y(S^*) \right).$$

Then, $S' \in S^{Y, S^*}(N)$.

Proof Since $\{\text{Succ}_{S^*_y}^{+}(\Gamma_y) \mid y \in Y\}$ is a partition of $\bigcup_{y \in Y} \text{Succ}_{S^*_y}^{+}(\Gamma_y)$, it is sufficient to show that $S' \in S(N)$. Towards a contradiction, assume there is a node $w \in V(N) - \rho_N$ that does not have exactly one parent in $S'$ and let $u'$ be the parent of $w$ in $S^*$. Clearly, for each $y \in Y$, we have $w \notin \Gamma_y$ as, otherwise, $\text{Pred}_y(w) = \text{Pred}_y(w)$. Further, $w \in \bigcup_{y \in Y} Y_{v_y}$ as, otherwise, $\text{Pred}_y(w) = \text{Pred}_y(w)$.

First, suppose $w$ has no parent in $S'$. Then, $u^*w \in \bigcup_{y \in Y} A_y(S^*)$ that is, $u^* \in \Gamma_y$ for some $y \in Y$, but $w \notin A_y(S^*)$. But since $S^* \in S(N)$, we know that $w$ has a parent in $S$ (which is not $u^*$ since $w \notin A_y(S^*)$), implying that $w$ is a reticulation in $N$. Thus, $w \in \text{Succ}_{S^*_y}^{+}(\Gamma_y) \subseteq \bigcup_{y \in Y} \text{Succ}_{S^*_y}^{+}(\Gamma_y)$ so there is some $y' \in Y$ with $w \in \text{Succ}_{S^*_y}^{+}(\Gamma_y)$ (note that $y' \neq y$ is possible). But then, $S_y$ contains an arc $uw \in A_y(S_y)$ which is in $S'$ by construction, thus contradicting $w$ having no parents in $S'$.

Second, suppose that $w$ has at least two distinct parents $u$ and $u^*$ in $S'$ and note that, again, $w$ is a reticulation in $N$. Since $S^*$ is a switching, at least one of them, say $u$, is such that $uw \in \bigcup_{y \in Y} A_y(S^*)$. However, since the $\text{Succ}_{S^*_y}^{+}(\Gamma_y)$ are disjoint and each $S^*$ is a switching, we cannot have $u^*w \in \bigcup_{y \in Y} A_y(S^*)$. Thus, $u^*w \in A(S^*) \setminus \bigcup_{y \in Y} A_y(S^*)$. However, since $\bigcup_{y \in Y} \text{Succ}_{S^*_y}^{+}(\Gamma_y) = \bigcup_{y \in Y} \text{Succ}_{S^*_y}^{+}(\Gamma_y)$, we
know that \(uw \in A_{\gamma}(\Gamma_y)\) for some \(y \in Y\). But then, \(w\) has two parents in \(S^*\) contradicting \(S^* \in S(N)\).

In the following, we prove the semantics of the table \(Q_{\psi_x}^{\phi}\).

For all \(i \leq t\), abbreviate \(\bigcup_{1 \leq i \leq t} \Gamma_w = Z_t\).

**Claim 5** Let \(1 \leq i \leq t\), let \(c_x \in \phi(x)\), and let \(R' \subseteq R(N)\). If \(S^i_{Z_t} \rightarrow R'(N) = \emptyset\), then \(Q_{\psi_x}^{\phi} [i, R'] = \infty\). Otherwise, let \(S_i \in S^i_{Z_t} \rightarrow R'(N)\) and \(\psi_i : V(N) \rightarrow C\) such that (a) \(\psi_i \leq \phi\), (b) \(\psi_i\) coincides with \(\psi_x[x \rightarrow c_x]\) on \(\bigcup_{j \leq i} YW_{\psi_i}^j\) and (c) \(\sum_{j \leq i} \sum_{uw \in A_{\psi_i}(S_i)} \delta_{\psi_i}(u, w)\) is minimum among all such \(S_i\) and \(\psi_i\) and

\[
Q_{\psi_x}^{\phi} [i, R'] = \sum_{j \leq i} \sum_{uw \in A_{\psi_i}(S_i)} \delta_{\psi_i}(u, w)
\]

**Proof** The proof is by induction on \(i\), noting that \(\psi_x[x \rightarrow c_x \big| YW_{\psi_i}^i = \psi_i \big| YW_{\psi_i}^1\) by Claim 5(b).

**Case** \(i = 1\): By (5), \(Q_{\psi_x}^{\phi}[0, R' \setminus R^*] \neq \infty\) only if \(R^* = R' \subseteq \text{Succ}_{\psi_1}(Z_1)\) and \(T^{{S\psi_1}^1}[\psi_1 \big| YW_{\psi_1}^1], R^*] = \infty\). However, if \(S^i_{Z_t} \rightarrow R'(N) = \emptyset\) then, by induction hypothesis (of the lemma), \(T^{{S\psi_1}^1}[\psi_1 \big| YW_{\psi_1}^1], R^*] = \infty\) and so \(Q_{\psi_x}^{\phi}[0, R' \setminus R^*] = \infty\). Furthermore, \(S_1, \psi_1\), and \(R'\) satisfy the conditions of the lemma for \(v_1\), so we can employ the induction hypothesis of the lemma. Thus,

\[
Q_{\psi_x}^{\phi}[1, R'] = 0 + T^{{S\psi_1}^1}[\psi_1 \big| YW_{\psi_1}^1], R^*]
\]

by IH lemma

\[
\sum_{uw \in A_{\psi_1}(S_1)} \delta_{\psi_1}(u, w)
\]

**Case** \(i > 1\): First, by (5), \(Q_{\psi_x}^{\phi}[i, R' \setminus R^*] \neq \infty\) only if \(Q_{\psi_x}^{\phi}[i - 1, R' \setminus R^*] \neq \infty\) and \(T^{{S\psi_i}^i}[\psi_i \big| YW_{\psi_i}^i], R^*] \neq \infty\). By induction hypotheses (of the claim and the lemma), there are switchings \(S_{i-1}\) and \(S'\) of \(N\) with \(\text{Succ}_{\psi_i}^{R^i}(Z_{i-1}) = R' \setminus R^*\) and \(\text{Succ}_{\psi_i}^{R^i}(\Gamma_{\psi_i}) = R^*\). Now, we construct a digraph \(S_i := (V(N), (A(S_{i-1} \setminus A_{\psi_i}(S_{i-1})) \cup A_{\psi_i}(S'))\) and show that \(S_i \in S^i_{Z_t} \rightarrow R'(N)\). Since \(\text{Succ}_{\psi_i}^{R^i}(Z_i) = \text{Succ}_{\psi_i}^{R^i}(Z_{i-1}) \cup \text{Succ}_{\psi_i}^{R^i}(\Gamma_{\psi_i})\), \(\text{Succ}_{\psi_i}^{R^i}(Z_i) = R' \setminus R^* = R\), it is sufficient to show that \(S_i\) can be turned into a switching of \(N\) without changing \(\text{Succ}_{\psi_i}^{R^i}(Z_i)\). To this end, suppose that there is a node \(w \neq \emptyset_{N}\) that does not have exactly one parent in \(S_i\). Since \(S_{i-1}\) and \(S'\) are switchings, \(w\) has parents \(u_{i-1}\) and \(u'\) in \(S_{i-1}\) and \(S'\), respectively. If \(w\) has no parent in \(S_i\), then \(u_{i-1}w \in A_{\psi_i}(S_{i-1})\) and \(u'w \notin A_{\psi_i}(S')\) and, thus, \(u_{i-1}w \in A_{\psi_i}(S)\) and \(u'w \notin A_{\psi_i}(S')\). Then, we can just add the arc \(u'w\) to \(S_i\) without changing \(\text{Succ}_{\psi_i}^{R^i}(Z_i)\). If \(w\) has at least two parents, then \(u_{i-1}\) and \(u'\) are both parents of \(w\) in \(S_{i-1}\) such that \(u_{i-1}w \in A_{\psi_i}(S)\) and \(u'w \notin A_{\psi_i}(S')\) and, thus, \(u'w \notin A_{\psi_i}(S')\). But then, we can remove \(u_{i-1}w\) from \(S_i\) without changing \(\text{Succ}_{\psi_i}^{R^i}(v_i)\). Repeating this argument, we can turn \(S_i\) into a switching of \(N\) with \(\text{Succ}_{\psi_i}^{R^i}(Z_i) = R'\), implying that \(S^i_{Z_t} \rightarrow R'(N) = \emptyset\). The second part of the claim, we show both inequalities separately.

Finally, let \(S_i \in S^i_{Z_t} \rightarrow R'(N)\) and \(\psi_i : V(N) \rightarrow C\) coincide with \(\psi_x[x \rightarrow c_x]\) on \(\bigcup_{j \leq i} YW_{\psi_i}^j\) and \(\sum_{j \leq i} \sum_{uw \in A_{\psi_i}(S_i)} \delta_{\psi_i}(u, w)\) is minimum among all such \(S_i\) and \(\psi_i\). Further, let \(R^* := \text{Succ}_{\psi_i}^{R^i}(\Gamma_{\psi_i})\). Note that \(\text{Succ}_{\psi_i}^{R^i}(Z_{i-1})\) and \(\text{Succ}_{\psi_i}^{R^i}(\Gamma_{\psi_i}) = R^*\) are disjoint since \(S_i\) is a switching, implying \(\text{Succ}_{\psi_i}^{R^i}(Z_{i-1}) = R' \setminus R^*\) and, thus, \(Q_{\psi_x}^{\phi}[i - 1, R' \setminus R^*]\) and \(T^{{S\psi_i}^i}[\psi_i \big| YW_{\psi_i}^i], R^*\) are finite by induction hypotheses. Then, as \(R^* \subseteq R' \cap \text{Succ}_{\psi_i}^{R^i}(\Gamma_{\psi_i})\), we know that \(R^*\) participates in the minimum of (3). Thus,

\[
Q_{\psi_x}^{\phi}[i, R'] \leq Q_{\psi_x}^{\phi}[i - 1, R' \setminus R^*] + T^{{S\psi_i}^i}[\psi_i \big| YW_{\psi_i}^i], R^*]
\]

by IH claim

\[
\sum_{j \leq i} \sum_{uw \in A_{\psi_i}(S_i)} \delta_{\psi_i}(u, w) + \sum_{uw \in A_{\psi_i}(S_i)} \delta_{\psi_i}(u, w)
\]

= \sum_{j \leq i} \sum_{uw \in A_{\psi_i}(S_i)} \delta_{\psi_i}(u, w)
“≥” Clearly: This direction is trivial if \( Q_{\omega, e}^\psi[i, R'] \) is infinite, so suppose it is finite. By (5), there is some \( R' \subseteq R' \cap \text{Succ}_{\omega}^\psi(\Gamma_v) \) with \( Q_{\omega, e}^\psi[i, R'] = Q_{\omega, e}^\psi[i - 1, R' \cap R'] + T^{S \psi}[\omega, \psi, R'] \). First, since
\( T^{S \psi}[\omega, \psi, R'] \neq \infty \), the induction hypothesis (of the lemma) guarantees that there is some \( S' \in S^{\psi}\cap R' \) (N) and \( \psi^* : V(N) \rightarrow C \) such that (a) \( \psi^* \mid Y \psi^* \leq \psi^* \leq \psi \), (b) \( (S', \psi^*) \) is optimal on \( A_{\omega}(S') \), and (c) \( T^{S \psi}[\omega, \psi, R'] = \sum_{u \in A_{\omega}(S')} \delta \psi(u, w) \). Second, since \( Q_{\omega, e}^\psi[i - 1, R' \cap R'] \neq \infty \), the induction hypothesis (of the claim) guarantees that there are \( S_{i-1} \in S^{\psi}\cap R' \cap R' \) (N) and \( \psi_{i-1} : V(N) \rightarrow C \) such that (a) \( \psi_{i-1} \leq \psi' \), (b) \( \psi_{i-1} \) coincides with \( \psi_{x}[x \rightarrow c_{x}] \) on \( \bigcup_{i < j} \sum_{u \in A_{\omega}(S_{i-1})} \delta \psi_{i-1}(u, w) \). Finally, we construct a new solution \( S' \) by replacing \( S_{i-1} \) by \( S' \) on \( \Gamma_v \) and by \( S_{i-1} \) on \( Z_{i-1} \) and we use Claim 5(c) to show that the cost of \( S_{i-1} \) is at most that of \( S' \). More formally, let
\[
S' := \left( V(N), \bigcup_{j<i} A_{\omega}(S_{j}) \right) \bigcup \bigcup_{j<i} A_{\omega}(S_{j-1}) \bigcup A_{\omega}(S')
\]
(11)

Since \( \{v_{1}, v_{2}, \ldots, v_{i}\} \) is an anti-chain in \( \Gamma' \) and \( \{\text{Succ}_{\omega}^{\psi}(Z_{i-1}), \text{Succ}_{\omega}^{\psi}(\Gamma_{v}^{i})\} = \{R', R' \cap R' \} \) is a partition of \( \text{Succ}_{\omega}^{\psi}(Z_{i}) = R' \), Claim 4 implies \( S' \in S^{\psi} \cap R' \). Further, let \( \psi' : V(N) \rightarrow C \) such that, for all \( a \in A(S') \),
\[
\psi'(a) := \psi_{i-1}(a) \text{ if } a \in A_{Z_{i-1}}, \quad \psi'(a) := \psi_{i-1}(a) \text{ if } a \in A_{\omega}(S'), \text{ and } \psi'(a) := \psi(a), \text{ otherwise.}
\]
Note that \( \psi' \leq \psi \). Further, \( \psi_i \) and \( \psi_{i-1} \) coincide on \( Y \psi^* \cap R' \) and, thus, \( \psi' \) and \( \psi_{i-1} \) coincide on all nodes touched by \( A_{Z_{i-1}}(S') = A_{Z_{i-1}}(S_{i-1}) \). Further, \( \psi_{i} \) and \( \psi^* \) coincide on \( Y \psi^* \) and, thus, \( \psi' \) and \( \psi^* \) coincide on all nodes touched by \( A_{\omega}(S') = A_{\omega}(S_\omega) \). Thus,

\[
Q_{\omega, e}^\psi[i, R'] = Q_{\omega, e}^\psi[i - 1, R' \cap R'] + T^{S \psi}[\omega, \psi, R'] \geq \sum_{u \in A_{\omega}(S')} \delta \psi(u, w) + \sum_{u \in A_{\omega}(S')} \delta \psi'(u, w)
\]
(10)

Having established the semantics of \( Q_{\omega, e}^\psi \), we can finish proving Case 2 of Lemma 8. First, consider the case that \( S^{\psi} \cap R' = \emptyset \) and assume that \( T^{S \psi}[\omega, \psi, R'] \neq \infty \). By Eq. (4) and Claim 5, there is some \( c_x \) and \( R' \subseteq R' \cap \text{Succ}_{\omega}^{\psi}(\Gamma) \) such that \( S^{\psi} \cap R' \cap R' \cap R' \neq \emptyset \). Let \( S' \) be a switching in one of these sets and note that \( \text{Succ}_{\omega}^{\psi}(\Gamma_x) = R' \cap R' \). If there is some \( y \in R' \cap \text{Succ}_{\omega}^{\psi}(\Gamma_x) \), then \( y \in R' \) and \( S' \) contains an arc \( xy \) for some \( x \notin \Gamma_x \), implying that we can swap \( y \) for \( x \) in \( S' \) without affecting \( \text{Succ}_{\omega}^{\psi}(Z_{i}) \) or \( S' \) being a switching. Thus, we can assume without loss of generality that \( \text{Succ}_{\omega}^{\psi}(\Gamma_x) = R' \). But then, \( S' \in S^{\psi} \cap R' \) contradicting \( S^{\psi} \cap R' = \emptyset \). In the following, we thus assume that \( S^{\psi} \cap R' = \emptyset \) and we show both directions of the lemma separately.

\[
\sum_{u \in A_{\omega}(S')} \delta \psi(u, w) + \sum_{u \in A_{\omega}(S')} \delta \psi'(u, w) \geq \sum_{x \in \omega \cup \psi, y \in \omega \cup \psi} \delta \psi(x, y) + \sum_{y \in \omega \cup \psi} \delta \psi'(y)
\]
(12)
Then, since $c_x$ and $R^s$ are valid choices for the minima in (4), we have

$$T^{SW}[x, \psi_x, R^x] \leq \sum_{r \in R^s \cup \text{Succ}_N^T(x)} \delta(c_x, \psi_x(r)) + \sum_{u \in \text{A}_z} \delta(\psi(u, w))$$

From $S'$ we construct a switching $S^* \in S^{\Gamma_x \rightarrow R^x}(N)$ by 1. swapping each arc $z \in A(S')$ with $r \in R^x$ for $x_r$ (which exists in $N$ since $R^s \subseteq \text{Succ}_N^T(x)$), 2. swapping each arc $x_r \in A(S')$ with $r \notin R^x$ for an arc $x_z$ with $z \notin \Gamma_x$ (which exists in $N$ since $S^{\Gamma_x \rightarrow R^x}(N) \neq \emptyset$), and 3. swapping the arc $x_z \in A_{\text{in}}^T(S')$ with an arc $x \in \text{Pred}_N^T(x \times \{x\}$ minimizing $\psi(x, z)$. Since this operation does not change the in-degree of any node, $S^*$ is still a switching of $N$ and we have $\text{Succ}_{N^*}^T(x) = R^x$ and $A_{\text{in}}^T(S^*) = A_{\text{in}}^T(S)$ by construction. Thus, $\text{Succ}_{N^*}^T(Z_t) = R^x \setminus R^s$ and $\text{Succ}_{N^*}^T(G_x) = R^x$. Altogether,

$$T^{SW}[x, \psi_x, R^x] = \sum_{r \in R^s \cup \text{Succ}_N^T(x)} \delta(c_x, \psi_x(r)) + \sum_{u \in \text{A}_z} \delta(\psi(u, w))$$

Lemma 8(b) 

Case 2.1: $\text{Pred}_N^T(x) \neq \emptyset$ and there are $c_x \in \phi(x)$ and $R^s \subseteq R^x \cap \text{Succ}_N^T(x)$ such that

$$T^{SW}[x, \psi_x, R^x] = \sum_{r \in R^s \cup \text{Succ}_N^T(x)} \delta(c_x, \psi_x(r)) + Q_{x,c_x}^T[R^s \setminus R^x] + \min_{y \in \text{Pred}_N^T(x)} \delta(c_x, \psi_x(y))$$

By Claim 5, there is some $S' \in S^{Z \rightarrow R} \setminus R^x(N)$ and some $\psi' : V(N) \rightarrow C$ such that (a) $\psi' \subseteq \phi$, (b) $\psi'$ coincides with $\psi_x[x \rightarrow c_x]$ on $\bigcup_{i \leq t} \text{Y}W^{T'}_i$ (recall that $x \in \bigcup_{i \leq t} \text{Y}W^{T'}_i$) (c) $\sum_{u \in A_z(S)} \delta(\psi(u, w))$ is minimum among all such $S'$ and $\psi'$ and

$$Q_{x,c_x}^T[R^s \setminus R^x] = \sum_{u \in A_z(S')} \delta(\psi(u, w))$$

Case 2.2: $\text{Pred}_N^T(x) \neq \emptyset$ and there are $c_x \in \phi(x)$ and $R^s \subseteq R^x \cap \text{Succ}_N^T(x)$ such that

$$T^{SW}[x, \psi_x, R^x] = \sum_{r \in R^s \cup \text{Succ}_N^T(x)} \delta(c_x, \psi_x(r))$$

Abbreviate $R' := (N \setminus \{x\}) \cup R^x \setminus R^s$. By Claim 5, there is some $S' \in S^{Z \rightarrow R'}$ (N) and some $\psi' : V(N) \rightarrow C$ such that (a) $\psi' \subseteq \phi$, (b) $\psi'$ coincides with $\psi_x[x \rightarrow c_x]$ on $\bigcup_{i \leq t} \text{Y}W^{T'}_i$, (c) $\sum_{u \in A_z(S')} \delta(\psi(u, w))$ is minimum among all such $S'$ and $\psi'$ and

$$Q_{x,c_x}^T[R^s \setminus R^x] = \sum_{u \in A_z(S')} \delta(\psi(u, w))$$

We construct a switching $S^* \in S^{\Gamma_x \rightarrow R^x}(N)$ by 1. swapping each arc $z \in A(S')$ with $r \in R^x$ for $x_r$ (which
exists in \( N \) since \( R^* \subseteq \text{Succ}^{\Gamma_1}_N(x) \) and 2. swapping each arc \( xr \in A(S') \) with \( r \notin R^* \) for an arc \( zr \) with \( z \notin \Gamma_x \) (which exists in \( N \) since \( S^{\Gamma_x \rightarrow R^*}(N) \neq \emptyset \). Since this operation does not change the in-degree of any node, \( S^* \) is still a switching of \( N \) and we have \( \text{Succ}^{\Gamma_1}_N(x) = R^* \) and \( A_{\Gamma_x}(S') = A_{\Gamma_x}(S^{\Gamma_x}) \) by construction. Thus, \( \text{Succ}^{\Gamma_1}_N(x) = R^* \) and \( \text{Succ}^{\Gamma_1}_N(x) = A_{\Gamma_x}(S^{\Gamma_x}) = R^x \). Further, note that if \( x \) is a tree node, then \( \text{Pred}_N(x) \neq \emptyset \) implies \( A^{\Gamma_1}_{[\iota]}(S') \subseteq A_{[\iota]}(N) \neq \emptyset \) and, otherwise, \( x \in R^* \) implying \( A_{[\iota]}(S') = \emptyset \). Altogether,

\[
T^{SW}_{[x, \psi_x, R^x]}(\eta) = \sum_{r \in R^* \cup \text{Succ}^{\Gamma_1}_N(x)} \delta \left( c_r, \psi_x(r) \right) + \sum_{u \in A_{\Gamma_x}(S')} \delta \left( \psi(x), u \right) \]

\[
= \sum_{r \in R^* \cup \text{Succ}^{\Gamma_1}_N(x)} \delta \left( c_r, \psi_x(r) \right) + \sum_{u \in A_{\Gamma_x}(S')} \delta \left( \psi(x), u \right) \]

\[
\psi = \psi' : YW^\Gamma_x \rightarrow 2^C, \quad \psi' = YW^\Gamma_x \rightarrow 2^C \quad \text{such that} \quad \psi' \leq \psi \leq \lambda \quad \text{and} \quad 0 \times \rho(x) \subseteq \eta' \subseteq \eta. \quad \text{Then},
\]

\[
T^{PT}_{[x, \lambda, \phi, \eta]} \leq T^{PT}_{[x, \lambda, \psi', \eta']} \quad \text{and} \quad Q^i_{[x, \lambda, \psi', \eta']} \leq Q^i_{[x, \lambda, \psi, \eta]}
\]

**Proof** Note that the inequality on \( Q^i \) trivially holds if \( Q^i_{[x, \lambda, \psi, \eta]} = \infty \) and, similarly for \( T^{PT} \). The proof is based on the observation that the transformations done to \( \psi \) and \( \eta \) in Equations (7) and (8) are monotone.

**Claim 6** Let \( U, D \in \mathbb{N} \). The following functions (acting on functions) are monotone

\[
f_{U,D}(\psi) := \psi \left[ x \rightarrow D \bigwedge_{y \in \text{Succ}^{\Gamma_1}_N(x)} \psi(y) \setminus U \right] = \eta \left[ x \rightarrow |U| - \sum_{u \in \text{Pred}^{\Gamma_1}_N(x)} |\lambda_x(u)| \right]
\]

\[
g_{U,D}(\eta) := \eta \left[ x \rightarrow |U| - \sum_{u \in \text{Pred}^{\Gamma_1}_N(x)} |\lambda_x(u)| \right] \quad \forall y \in \text{Succ}^{\Gamma_1}_N(x) \rightarrow \eta(y) - |U|
\]

The proof for \( g_{U,D} \) is completely analogous. \( \square \)

With Claim 6, we can show that monotonicity of \( Q^i \) implies monotonicity of \( T^{PT} \).

**Claim 7** Let \( v_1, v_2, \ldots, v_t \) be the children of \( x \) in \( \Gamma \) and suppose that \( Q^i \) is monotone. Then, \( T^{PT} \) is monotone.

**Proof** If \( T^{PT}_{[x, \lambda, \phi, \eta]} \neq \infty \), there are \( D \subseteq U \subseteq \phi(x) \) such that the minimum in Equation (7) in Definition 7 is attained, that is,

\[
T^{PT}_{[x, \lambda, \phi, \eta]} = Q^i_{[x, \lambda, \phi, \eta]} + c_{U,D} = Q^i_{[x, \lambda, \phi, \eta]} + c_{U,D}
\]

for some constants \( c_{U,D} \) and \( c_{U,D} \) that are independent of \( \phi \) and \( \eta \). Since, by assumption, \( Q^i \) is monotone for all \( \lambda \) and both \( f_{U,D} \) and \( g_{U,D} \) are monotone by Claim 6, we conclude
\[ T^{PT}[x, \lambda, \psi, \eta] \geq Q^2[i, f_{U,D}(\psi), g_{U,D}(\eta)] + c_{U,D} \]
\[ \geq Q^2[i, f_{U,D}(\psi'), g_{U,D}(\eta')] + c_{U,D} \geq T^{PT}[x, \lambda, \psi', \eta'] \]

Note the last “\( \geq \)” since we only know that this particular value participates in the minimum that forms \( T^{PT}[x, \lambda, \psi', \eta'] \), while this minimum may be attained at an even smaller value.

By Claim 7, in order to prove Lemma 9, it is sufficient to show that \( Q^2 \) is monotone. This proof is by induction on the height of \( x \) in \( \Gamma \) and the value of the first argument \( i \) of \( Q^2 \).

For the induction base, suppose that \( x \) is a leaf in \( \Gamma \) and note that \( x \) has \( t = 0 \) children. If \( Q^2[0, \psi, \eta] \neq \infty \), we have \( \psi = 0 \) and \( \eta = 0 \). Then, \( \psi = \psi' = \eta = \eta' \), implying \( Q^2[0, \psi', \eta'] = Q^2[0, \psi, \eta] \).

For the induction step, let \( x \) have \( t \) children \( v_1, v_2, \ldots, v_t \) and let \( 0 < i < t \). First, let \( \psi^* \leq \psi \) and \( \eta^* \leq \eta \) be such that the minimum in Equation (8) in Definition 7 is attained, that is, \( Q^2[i, \psi, \eta] = Q^2[i, \psi, \eta]_{i - 1} \psi^*, \eta - \eta^* \psi^* + T^{PT}i, \lambda, \psi^*, \eta^* \). Further, let \( \psi^* \) and \( \psi^* \) be defined as \( \psi^*(y) := \psi(y) \cap \psi^*(y) \) and \( \eta^*(y) := \min\{\eta^*(y), \eta^*(y)\} \).

Clearly, \( \psi^* \leq \psi^* \) and \( \psi^* \leq \psi^* \).

Thus for all \( y \),
\[ (\psi^* - \psi^*)(y) = (\psi^* - \psi^*)(y) \]
\[ (\eta^* - \eta^*)(y) = (\eta^* - \eta^*)(y) \]
\[ \leq (\eta^* - \eta^*)(y) \]

so \( \psi^* \leq \psi^* \) and \( \eta^* \leq \eta^* \). Since \( \psi^* \) and \( \eta^* \) participate in the minimum in the definition of \( Q^2[i, \psi, \eta] \),
\[ Q^2[i, \psi, \eta] = Q^2[i, \psi, \eta]_{i - 1} \psi^*, \eta - \eta^* \psi^* + T^{PT}i, \lambda, \psi^*, \eta^* \]
\[ \geq Q^2[i, \psi, \eta]_{i - 1} \psi^*, \eta^* \]

there are no such \( f \), then \( T^{PT}[x, \lambda, \psi, \eta] = \infty \). Otherwise,
\[ T^{PT}[x, \lambda, \psi, \eta] = \sum_{z \leq x} \text{cost}_f(z) \]

**Proof** Note that, if the cost of \( f \) is finite, then \( |f(v)| \leq \sum_{v \in \text{Pred}(v)} |f(u)| \) for all \( v \neq \rho_N \) and \( |f(\rho_N)| = 1 \) by Definition 4. Again, the proof is by induction on the height of \( x \) in \( \Gamma \).

**Case 1:** \( x \) is a leaf in \( \Gamma \), that is, \( t = 0 \) and \( \text{Pred}^\uparrow_N(v) \subseteq \{x\} \) for all \( v \). Then, by Definition 7, \( T^{PT}[x, \lambda, \psi, \eta] \) is finite if and only if
\[ \psi^*_x \]
\[ \psi^*_x \]
and
\[ \eta^*_x \]
\[ \eta^*_x \]
if and only if (considering the assignments of the above modifications of \( \psi_x \) and \( \eta_x \) individually) (a) \( D = \emptyset \), (b) \( |U| - \sum_{v \in \text{Pred}^\uparrow_N(x)} |\lambda_x(v)| = \rho(x) \) (c) for each \( y \in \text{Succ}_N(x) \),
\[ U \supseteq \psi_x(y) \cap f(x) \text{ and } |U| \geq \eta_x(y) = |f(x)| \]

In this case, the table entry is assigned the cost
\[ |U| - \sum_{v \in \text{Pred}^\uparrow_N(x)} |\lambda_x(v)| - \rho(x) = |U| - \sum_{v \in \text{Pred}(x)} f(r) - \rho(x) \].

**Lemma 10** Let \( \Gamma \) be a tree agreeing with \( N \), let \( x \in V(N) \), let \( \psi_x, \lambda_x : \text{YWS}^\uparrow \to \text{N}^\uparrow \) and \( \eta_x : \text{YWS}^\uparrow \to \text{N}^\uparrow \). Let \( f \) minimize \( \text{cost}_f \) among all code in \( \mathcal{L} \) such that, for all \( w \in \text{YWS}^\uparrow \), \( \lambda_x(w) = f(w) \), \( \psi_x(w) = f(w) \) and \( \eta_x(w) = f(w) \) and \( \eta_x(w) \leq \sum_{u \in \text{Pred}_N(w)} f(u) \). If

\( x = \rho_N \), this simplifies to \( |U| - 1 \) and, since \( |f(\rho_N)| = 1 \), the cost is minimized by \( U = f(\rho_N) \) and the table entry equals \( 0 = \text{cost}_f(\rho_N) \). Thus, in the following, let \( x \neq \rho_N \).

**“*”:** Since (20) is satisfied for \( U = f(x) \), the minimum over all \( U \) is at most the cost when choosing \( U = f(x) \), which is \( |f(x) - \sum_{v \in \text{Pred}(x)} f(r)| = \text{cost}_f(x) \).
“≥”; Towards a contradiction, assume that there is a \( U \) satisfying (20) such that \( T^{PT}[x, \lambda_x, \psi_x, \eta_x] = |U \setminus \bigcup_{v \in \text{Pred}(x)} f(r)| < |f(x) \setminus \bigcup_{v \in \text{Pred}(x)} f(r)| = \text{cost}_f(x) \).

We show that \( f' := f[x \rightarrow U] \) has less overall cost than \( f \), contradicting its optimality. Since changing \( f(x) \) to \( U \) only influences the cost of \( x \) and its children in \( N \), it suffices to consider them. To this end, let \( y \) be any child of \( x \) in \( N \).

First, by (b), \( |f'(x)| = |U| \leq \sum_{v \in \text{Pred}(x)} |f'(r)| \) so \( \text{cost}_{f'}(x) = |U \setminus \bigcup_{v \in \text{Pred}(x)} f(r)| < \text{cost}_f(x) \) by assumption.

Case 1: By Definition 7, \( Q^x_i[0, \psi_1 - \eta', \eta_1 - \eta'] \) is finite if and only if \( \psi'_1 = \psi_1, \eta'_1 = \eta_1 \) and \( T^{PT}[v_1, \lambda |YW^{T}_{v_1}, \psi', \eta'] \) is finite, that is, by induction hypothesis of the lemma, there is a lineage function \( f' \) that is eligible for \( Y_1, \lambda, \psi_1 = \psi' \) and \( \eta_1 = \eta' \). Thus, the first part of the claim follows. Since \( \psi_1 \) and \( \eta_1 \) are the only valid choices for the minima in (8) that result in finite values, we conclude

\[
Q^x_i[1, \psi_1, \eta_1] = -\rho(x) + T^{PT}[v_1, \lambda |YW^{T}_{v_1}, \psi_1, \eta_1]|^{\text{Hi lemma}} = \sum_{z \in Z_i} \text{cost}_{f_i}(z) - \rho(x)
\]

Case 2: \( x \) has children \( v_1, v_2, \ldots, v_t \) with \( t \geq 1 \) in \( \Gamma \). In the following, we abbreviate \( Y_i := \bigcup_{y \in Y} YW^{T}_{v_i} \) and \( Z_i := \bigcup_{y \in Y} \Gamma_y \). Further, we call a lineage function \( f' \) eligible with respect to an anti-chain \( Y \) in \( \Gamma \) and functions \( \lambda', \psi', \eta' \) if, for all \( w \in \bigcup_{y \in Y} YW^{T}_{v_i} \), we have \( \lambda(w) = f'(w), \psi'(w) \leq f'(w) \cap \bigcup_{y \in Y} \bigcup_{u \in \text{Pred}(w)} f'(u) \) and \( \eta'(w) \leq \sum_{y \in Y} \sum_{v \in \text{Pred}(w)} f'(u) + \rho(w) \) and the cost of \( f' \) is finite on \( \bigcup_{y \in Y} \Gamma_y \). We first show how the table \( Q^x_i \) is used to distribute lineages along the \( v_i \).

Claim 8 Let \( 1 \leq i \leq t \), \( \eta_i : Y_i \rightarrow \mathbb{N} \) and let \( \lambda, \psi : Y_i \rightarrow 2^C \) with \( \psi_i \subseteq \lambda \). Let \( f_i \) minimize \( \sum_{z \in Z_i} \text{cost}_{f_i}(z) - \rho(x) \) among all lineage functions that are eligible with respect to \( Y_i, \lambda, \psi_i, \) and \( \eta_i \). If no such \( f_i \) exists, then \( Q^x_i[i, \psi_i, \eta_i] = \infty \). Otherwise,

\[
Q^x_i[i, \psi_i, \eta_i] = \sum_{z \in Z_i} \text{cost}_{f_i}(z) - \rho(x).
\]

Proof The proof of the claim is by induction on \( i \).
Finally, the cost of $f^*$ on $Z_i$ equals the cost of $f_{i-1}$ on $Z_{i-1}$ plus the cost of $f'$ on $\Gamma_i$ and is, therefore, finite. Thus, $f^*$ is eligible for $Y_i$, $\hat{\lambda}$, $\psi$, and $\eta_i$, implying the contraposition of the first part of the lemma. For the cost equality, we consider both directions separately.

"≤": Let $\psi': YW^\Gamma_{Y_i} \rightarrow 2^C$ and $\eta': YW^\Gamma_{Y_i} \rightarrow \mathbb{N}$ be defined on $YW^\Gamma_{Y_i}$ as

$$\psi'(w) := \psi_i(w) \cap f_i(w) \cap \bigcup_{u \in \text{Pred}^\Gamma_{Y_i}(w)} f_i(u)$$

and

$$\eta'(w) := \min\{\eta_i(w), \sum_{u \in \text{Pred}^\Gamma_{Y_i}(w)} |f_i(u)| + \rho(w)\}.$$ Clearly, $\psi' \leq \psi_i |_{YW^\Gamma_{Y_i}}$ and $\eta' \leq \eta_i |_{YW^\Gamma_{Y_i}}$. Furthermore, define $\psi_{i-1}$ and $\eta_{i-1}$ by, for all $w \in \bigcup_{y \in Y_{i-1}} YW^\Gamma_y$, setting

$$\psi_{i-1}(w) := \psi_i(w) - \psi'(w) \leq \left(f'(w) \cap \bigcup_{y \in Y_{i-1}} \bigcup_{u \in \text{Pred}^\Gamma_{Y_i}(w)} f'(u)\right) \setminus \left(f'(w) \cap \bigcup_{u \in \text{Pred}^\Gamma_{Y_i}(w)} f'(u)\right)$$

and

$$\eta_{i-1}(w) := \eta_i(w) - \eta'(w) \leq \sum_{y \in Y_{i-1}} \sum_{u \in \text{Pred}^\Gamma_{Y_i}(w)} |f_i(u)| - \sum_{u \in \text{Pred}^\Gamma_{Y_i}(w)} |f_i(u)|$$

$$= \sum_{y \in Y_{i-1}} \sum_{u \in \text{Pred}^\Gamma_{Y_i}(w)} |f_i(u)|.$$
Thus, \( f_i \) is eligible with respect to \( Y_{i-1}, \lambda, \psi_{i-1} \) and \( \eta_{i-1} \), implying

\[
Q^h[i, \psi_i, \eta_i] \leq Q^h[i-1, \psi_{i-1}, \psi_i - \psi', \eta_i - \eta'] + T^P T[\psi_i - \psi', \eta_i - \eta']
\]

by the induction hypothesis of the claim, there is a lineage function \( f_{i-1} \) that is eligible with respect to \( Y_{i-1}, \lambda, \psi_i - \psi', \eta_i - \eta' \) with \( Q^h[i-1, \psi_i - \psi', \eta_i - \eta'] \) held. By (8), there are \( \psi' \leq \psi_i \mid Y_{W_{Y_i}} \) and \( \eta' \leq \eta_i \mid Y_{W_{Y_i}} \) such that

\[
Q^h[i, \psi_i, \eta_i] = Q^h[i-1, \psi_i - \psi', \eta_i - \eta'] + T^P T[\psi_i, \lambda \mid Y_{W_{Y_i}}, \psi', \eta']
\]

Towards a contradiction, assume that this value is strictly smaller than \( \sum x \in Z_i \text{cost}_y(z) - \rho(x) \). By the induction hypothesis of the lemma, there is a lineage function \( f' \) that is eligible with respect to \( \{v_i\}, \lambda, \psi', \) and \( \eta' \) with \( T^P T[v_i, \lambda \mid Y_{W_{Y_i}}, \psi', \eta'] = \sum x \in Z_i \text{cost}_y(z) - \rho(x) \). We construct a lineage function \( f^* \) by setting

\[
f^*(w) := \begin{cases} f_{i-1}(w) & \text{if } w \in Z_i \\ f'(w) & \text{if } w \in \Gamma_i \\ f_i(w) & \text{otherwise} \end{cases}
\]

By eligibility of \( f_{i-1}, f_i \) and \( f' \), we know that \( f_{i-1}, f_i \) and \( f^* \) coincide with \( \lambda \) on \( \bigcup_{y \in Y_{i-1}} Y_{W_{Y_i}} \) and \( f', f_i \) and \( f^* \) coincide with \( \lambda \) on \( Y_{W_{Y_i}} \). To contradict optimality of \( f_i \) it thus suffices to show that \( f^* \) is eligible with respect to \( Y_{i-1}, \lambda, \psi_i, \) and \( \eta_i \). To this end, note that, for all \( w \in Y_{i-1} \), we have

\[
\psi_i(w) = (\psi_i - \psi')(w) \cup \psi'(w) \\
\leq \left( \lambda(w) \cap \bigcup_{y \in Y_{i-1}} \bigcup_{u \in \text{Pred}_N^v}(u) \right) \cup \left( \lambda(w) \cap \bigcup_{u \in \text{Pred}_N^v}(u) \right) \\
= f^*(w) \cap \bigcup_{y \in Y_i} \bigcup_{u \in \text{Pred}_N^v}(u)
\]

as well as

\[
\eta_i(w) = (\eta_i - \eta')(w) + \eta'(w) \leq \sum_{y \in Y_{i-1}} \sum_{u \in \text{Pred}_N^v}(u) f_{i-1}(u) + \sum_{u \in \text{Pred}_N^v}(u) \left[ f'(u) \right] + \rho(w) \\
= \sum_{y \in Y_i} \sum_{u \in \text{Pred}_N^v}(u) f^*(u) + \rho(w)
\]
Having established the equality for \( Q^\_N \), we can now prove the lemma for \( i > 1 \). For the first part, suppose that \( T^{\_P} [x, \lambda_x, \psi_x, \eta_x] \neq \infty \). By (7), there are \( D \subseteq U \subseteq \phi(x) \) such that \( T^{\_P} [x, \lambda_x, \psi_x, \eta_x] = Q^\_N \cup [t, \psi_t, \eta_t] + |U \setminus (D \cup \bigcup_{u \in \text{Pred}_N (x)} \lambda_x (u))| \), where

\[
\psi_t := \psi_x \left[ x \rightarrow D, \forall w \in \text{Succ}_N (x) \ w \rightarrow \psi_x (w) \setminus U \right] \text{ and } \\
\eta_t := \eta_x \left[ x \rightarrow |U| \sum_{u \in \text{Pred}_N (x)} |\lambda_x (u)|, \forall w \in \text{Succ}_N (x) \ w \rightarrow \eta_x (w) \setminus |U| \right].
\]

By Claim 8, there is a lineage function \( f_t \) that is eligible for \( \{ v_t \}, \lambda_t := \lambda_x [x \rightarrow U], \psi_t \), and \( \eta_t \). Without loss of generality, suppose that \( f_t (w) = \lambda_x (w) \) for all \( w \in (YW^\_x \setminus \{ x \}) \setminus \bigcup_{y \in Y_t} YW^\_y \). In particular, \( f_t (x) = \lambda_x (x) = U \) and \( f_t \) has finite cost on \( Z_t \).

\[
|f_t (x)| > \sum_{u \in \text{Pred}_N (x)} |f_t (u)|
\]

\[
= \sum_{y \in Y_t} |f_t (u)| + \sum_{y \in Y_t} \sum_{u \in \text{Pred}_N (x)} |f_t (u)|
\]

\[
\geq \sum_{u \in \text{Pred}_N (x)} |\lambda_x (u)| + n_x (x) \geq |U|
\]

contradicting \( f_t (x) = U \). Further, for each \( w \in YW^\_N \setminus \text{Succ}_N (x) \),

\[
\psi_x (w) = \psi_t(w) \subseteq f_t (w) \cap \bigcup_{y \in Y_t} \bigcup_{u \in \text{Pred}_N (x)} \lambda_x (u) = f_t (w) \cap \bigcup_{u \in \text{Pred}_N (x)} \lambda_x (u)
\]

\[
\eta_x (w) = \eta_t (w) \leq \sum_{y \in Y_t} \sum_{u \in \text{Pred}_N (x)} |f_t (u)| = \sum_{u \in \text{Pred}_N (x)} |f_t (u)|
\]

and, for each \( w \in \text{Succ}_N (x) \),

\[
\psi_x (w) \subseteq \psi_t (w) \cup U \subseteq f_t (w) \cap \bigcup_{y \in Y_t} \bigcup_{u \in \text{Pred}_N (x)} \lambda_x (u) \cup f_t (x) = f_t (u) \cup \bigcup_{u \in \text{Pred}_N (x)} \lambda_x (u)
\]

\[
\eta_x (w) \leq \eta_t (w) + |U| \leq \sum_{y \in Y_t} \sum_{u \in \text{Pred}_N (x)} |f_t (u)| + |f_t (x)| = \sum_{u \in \text{Pred}_N (x)} |f_t (u)|
\]

We show that \( f_t \) is eligible with respect to \( \{ x \}, \lambda_x, \psi_x \) and \( \eta_x \). First, assume that \( \text{cost}_t (x) = \infty \), that is, either \( x = \rho_N \) and \( |f_t (x)| = |U| > 1 \) or \( x \neq \rho_N \) and \( |f_t (x)| > \sum_{u \in \text{Pred}_N (x)} |f_t (u)| \). In the first case, \( n_t (x) = |U| > 1 \), contradicting \( \eta_t (x) \leq \rho (x) \). In the second case, \( n_t (x) = |U| - \sum_{u \in \text{Pred}_N (x)} |\lambda_x (u)| \), implying the first part of the lemma. For the second part, we consider the directions separately.
We pick up the definition of \( f_t \) and show that 
\[
T^{PT}[x; \lambda_x, \psi_x, \eta_x] \geq \sum_{z \in Z_x} \text{cost}_z(z).
\]
Then, \( \succeq \) follows from optimality of \( f \) on \( \Gamma_x \). Indeed,
\[
T^{PT}[x; \lambda_x, \psi_x, \eta_x] = Q_{x \rightarrow D}^t[t, \psi_t, \eta_t] + |U \setminus (D \cup \bigcup_{u \in \text{Pred}_x^d(x)} \lambda_x(u))|,
\]
and, since \( \psi_t(x) = D \subseteq f_t(x) \cap \bigcup_{u \in \text{Pred}_x^d(x)} f_t(u), \)
\[
= \sum_{z \in Z_x} \text{cost}_z(z) + |f_t(x) \setminus \bigcup_{u \in \text{Pred}_x^d(x)} f_t(u)| + \sum_{u \in \text{Pred}_x^d(x)} \text{cost}_u(u),
\]
\[
= \sum_{z \in Z_x} \text{cost}_z(z) + \text{cost}_u(u) \geq \sum_{z \in Z_x} \text{cost}_z(z) - \|D\|.
\]

Let \( U := f(x) \) and let \( D := f(x) \cap \bigcup_{u \in \text{Pred}_x^d(x)} f(u) \subseteq U. \)
Then, \( |U \setminus (D \cup \bigcup_{u \in \text{Pred}_x^d(x)} f(u) \lambda_x(u))| = \text{cost}_f(x) + \rho(x). \) Further, let
\[
\psi_t := \psi_x \bigg[x \rightarrow D, \forall_{u \in \text{Succ}_x^d(x)} w \rightarrow \psi_x(w) \setminus \{x\} \bigg]\]
and
\[
\eta_t := \eta_x \bigg[x \rightarrow |U| \setminus \sum_{u \in \text{Pred}_x^d(x)} \lambda_x(u), \forall_{u \in \text{Succ}_x^d(x)} w \rightarrow \eta_x(w) \setminus \{x\} \bigg].
\]

We show that \( f \) is eligible with respect to \( Y_t, \lambda_x, \psi_t \) and \( \eta_t \). Then, \( Q_{x \rightarrow D}^t[t, \psi_t, \eta_t] = \sum_{z \in Z_x} \text{cost}_z(z) - \rho(x) \) by Claim \( 8 \), implying
\[
T^{PT}[x, \lambda_x, \psi_x, \eta_x] \leq \sum_{z \in Z_x} \text{cost}_z(z) - \rho(x) + \text{cost}_f(x) + \rho(x) = \sum_{z \in Z_x} \text{cost}_z(z)
\]
since \( U \) and \( D \) are valid choices for the minimum in (7).
To see that \( f \) is eligible, note that \( f(w) = \lambda_x[w \rightarrow U] \) for all \( w \in \bigcup_{y \in Y_t} YW^y_y \). Consider \( w \in \bigcup_{y \in Y_t} YW^y_y \cup \{x\} \). Further, for the conditions on \( \psi_t \) and \( \eta_t \), consider three cases for nodes in \( \bigcup_{y \in Y_t} YW^y_y \). First, if \( w = x \), then
\[
\psi_t(x) = D = \bigcup_{u \in \text{Pred}_x^d(x)} f(u) \cup f(x) \bigcup_{y \in Y_t} \bigcup_{u \in \text{Pred}_x^d(y)} f(u)
\]
\[
\eta_t(x) = |U| \setminus \sum_{u \in \text{Pred}_x^d(x)} \lambda_x(u) = |f(x)| \setminus \sum_{u \in \text{Pred}_x^d(x)} |f(u)| \leq \sum_{u \in \text{Pred}_x^d(x)} |f(u)|
\]
Second, if \( w \in \bigcup_{y \in Y_t} YW^y_y \setminus \text{Succ}_x^d(x) \), then
\[
\psi_t(w) = \psi_x(w) \setminus U = f(w) \cap \bigcup_{u \in \text{Pred}_x^d(w)} f(u) \setminus f(x) = f(w) \cap \bigcup_{u \in \text{Pred}_x^d(w) \setminus \{x\}} f(u)
\]
as well as
\[
\eta_t(w) = |\eta_x(w) \setminus U| \leq \sum_{u \in \text{Pred}_x^d(w) \setminus \{x\}} |f(u)| + \rho(x) - |f(x)|
\]
\[
= \sum_{u \in \text{Pred}_x^d(w) \setminus \{x\}} |f(u)| + |f(x)| + \rho(x) - |f(x)| = \sum_{y \in Y_t} \sum_{u \in \text{Pred}_x^d(y)} |f(u)| + \rho(x)
Otherwise, \( w \in \bigcup_{y \in Y} Y \bigcup \left( \text{Succe}^\uparrow_N(x) \cup \{ x \} \right) \) and we have

\[
\psi_t(w) = \psi_x(w) \cap \bigcup_{u \in \text{Pred}_N^\uparrow(w)} f(u) = \bigcup_{y \in Y \cup \text{Pred}_N^\uparrow(y)} (\bigcup_{y \in Y} Y \cup \{ x \}) \cup \{ x \} \cup \bigcup_{y \in Y} Y \cup \{ x \}
\]

\[
\eta_t(w) = \eta_x(w) \subseteq \bigcup_{u \in \text{Pred}_N^\uparrow(w)} f(u) = \bigcup_{y \in Y \cup \text{Pred}_N^\uparrow(y)} (\bigcup_{y \in Y} Y \cup \{ x \}) \cup \{ x \} \cup \bigcup_{y \in Y} Y \cup \{ x \}
\]

Acknowledgements

We thank Christophe Paul for sharing his expertise on treewidth formulations, and an anonymous reviewer for suggesting an interesting variant of decomposition minimizing the maximum number of reticulation nodes per bag for future work.

Author contributions

CS and MW contributed equally to the paper. Both authors read and approved the final manuscript.

Funding

This work was supported by French Agence Nationale de la Recherche through the CoCoAlSeq Project (ANR-19-CE45-0012).

Declarations

Competing interests

The authors declare that they have no competing interests

Author details

1SEM, Université de Montpellier, CNRS, IRD, EPHE, Montpellier, France. 2LIGM, Université Gustave Eiffel, CNRS, Paris, France.

Received: 24 January 2022   Accepted: 17 March 2022

References

1. Felsenstein J. Inferring phylogenies, vol. 2. Sunderland: Sinauer Associates; 2004.
2. Fitch WM. Toward defining the course of evolution: minimum change for a specific tree topology. Syst Biol. 1971;20(4):406–16.
3. Huson DH, Rupp R, Scornavacca C. Phylogenetic networks: concepts, algorithms and applications. Cambridge: Cambridge University Press; 2010.
4. Kannan L, Wheeler WC. Maximum parsimony on phylogenetic networks. Algo Mol Biol. 2012;7(1):9.
5. Hein J. Reconstructing evolution of sequences subject to recombination using parsimony. Math Biosci. 1990;89(2):185–200.
6. Nakhleh L, Jin G, Zhao F, Meller-Crummey J. Reconstructing phylogenetic networks using maximum parsimony. In: 2005 IEEE Computational Systems Bioinformatics Conference (CSB’05), pp. 93–102 (2005). IEEE.
7. Zhu J, Yu Y, Nakhleh L. In the light of deep coalescence: revisiting trees within networks. BMC Bioinf. 2016;17(14):271–82.
8. Fischer M, Iersel LV, Kelk S, Scornavacca C. On computing the maximum parsimony score of a phylogenetic network. Syst Biol. 2015;64(1):55–85.
9. Kannan L, Wheeler WC. Exactly computing the parsimony scores on phylogenetic networks using dynamic programming. J Comput Biol. 2014;21(4):303–19.
10. Jin G, Nakhleh L, Snir S, Tuller T. Parsimony score of phylogenetic networks: hardness results and a linear-time heuristic. IEEE/ACM Trans Comput Biol Bioinf. 2009;6(3):495–505.
11. Jin G, Nakhleh L, Snir S, Tuller T. Maximum likelihood of phylogenetic networks. Bioinformatics. 2006;22(21):2604–11.
12. Van Iersel L, Jones M, Scornavacca C. Improved maximum parsimony models for phylogenetic networks. Syst Biol. 2018;67(3):518–42.
13. Bodlaender H. A linear-time algorithm for finding tree-decompositions of small treewidth. SIAM J Comput. 1996;25(6):1305–17.
14. Authors V. The graph parameter hierarchy. Available at https://gitlab.com/gruenwald/parameter-hierarchy. 2021.
15. Bodlaender H. Discovering treewidth. In: Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM’05), pp. 1–16. Springer, Berlin, Heidelberg. 2005.
16. Bodlaender H. Treewidth: structure and algorithms. In: International Colloquium on Structural Information and Communication Complexity, pp. 11–25. Springer. 2007.
17. Bryant D, Lagergren J. Compatibility of unrooted phylogenetic trees is FPT. Theoret Comput Sci. 2006;351(3):296–302.
18. Courcelle B. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. Inf Comput. 1990;85(1):12–75.
19. Bulteau L, Weller M. Parameterized algorithms in bioinformatics: an overview. Algorithms. 2019;12(2):256.
20. Berry V, Scornavacca C, Weller M. Scanning phylogenetic networks is NP-hard. In: Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM’20), pp. 519–530. Springer, 2020.
21. Korhonen T. Single-exponential time 2-approximation algorithm for treewidth. CoRR abs/2104.07463. 2021.
22. Dell H, Komusiewicz C, Talmon N, Weller M. The PACE 2017 Parameterized Algorithms and Computational Experiments Challenge: The Second Iteration. In: 12th International Symposium on Parameterized and Exact Computation (IPEC 2017), vol. 89. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik; 2018. p. 30–130.
23. Tamaki H. Positive-instance driven dynamic programming for treewidth. J Comb Optim. 2019;37(4):1283–311.
24. Dendris ND, Kioussis LM, Thilikos DM. Fugitive-search games on graphs and related parameters. Theoret Comput Sci. 1997;172(1):233–54.
25. Arnborg S. Efficient algorithms for combinatorial problems on graphs with bounded decomposability—a survey. BIT Numer Math. 1985;25(1):1–23.
26. Mesoiff G, Paul C, Thilikos D. A polynomial time algorithm to compute the connected tree-width of a series-parallel graph. 2021. 004.0547v5.
27. Kloks T. Treewidth: computations and approximations, vol. 842. Berlin: Springer; 1994.
28. Rabier C-E, Berry V, Stolz M, Santos JdA, Wang W, Jean-Christophe G, Pardi F, Scornavacca C. On the inference of complicated phylogenetic networks by Markov chain Monte-Carlo. Submitted.
29. Bachoore E, Bodlaender HL. Weighted treewidth algorithmic techniques and results. In: International Symposium on Algorithms and Computation. pp. 893–903. Springer; 2007.
30. Zhu J, Wen D, Yu Y, Meudt HM, Nakhleh L. Bayesian inference of phylogenetic networks from bi-allelic genetic markers. PLoS Comput Biol. 2018;14(1):1005932.
31. Kelk S, Pardi F, Scornavacca C, van Iersel L. Finding a most parsimonious or likely tree in a network with respect to an alignment. J Math Biol. 2019;78(1–2):527–47.
32. Jin G, Nakhleh L, Snir S, Tuller T. Inferring phylogenetic networks by the maximum parsimony criterion: a case study. Mol Biol Evol. 2006;24(1):324–37.
33. Wheeler WC. Phylogenetic network analysis as a parsimony optimization problem. BMC Bioinformatics. 2015;16(1):1–9.
34. Bryant C, Fischer M, Linz S, Semple C. On the quirks of maximum parsimony and likelihood on phylogenetic networks. J Theor Biol. 2017;417:100–8.

Publisher’s note
Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.