SUPPLEMENTAL METHODS
for

A fast and scalable method for inferring phylogenetic networks from trees by aligning lineage taxon strings

by Zhang et al.

A. Correctness of Algorithm A - - - - - - - - - - - - Page 2

B. Reduction for a reducible tree set - - - - - - - - - - - - Page 11

C. Computing the branch weights of the inferred tree-child network - - - - - - - - - - - Page 12

D. A phylogenetic network for hominin relationships - - - - - - - - - - - Page 14

A. Correctness of Algorithm A

A1. Concepts and notation

Directed graphs

A directed graph G consists of a set V of nodes and a set E of directed edges that are ordered pairs of distinct nodes. Let $e = (u, v) \in E$. We call e an outgoing edge of u and an incoming edge of v. For a node $v \in V$, its *outdegree* and *indegree* are defined as the number of outgoing and incoming edges of v, respectively.

For a graph, *subdividing* an edge (u, v) involves replacing it with a directed path from u to v that passes one or more new nodes. Conversely, an *edge contraction* at a node v of indegree one and outdegree one is to remove v and replace the path $u \to v \to w$ with an edge (u, w), where (u, v) and (v, w) are the unique incoming and outgoing edge of v, respectively.

Phylogenetic networks

A *phylogenetic network* on a set X of taxa is a rooted, directed acyclic graph in which (i) all the edges are oriented away from the root, which is of indegree 0 and outdegree 1; (ii) the nodes of indegree 1 and outdegree 0, called leaves, are uniquely labeled with the taxa; and (iii) all the non-root and non-leaf nodes are either tree nodes that are of indegree 1 and outdegree 2 or reticulate nodes that are of indegree more than 1 and outdegree 1. Reticulate nodes represent evolutionary reticulation events. A phylogenetic network is said to be *binary* if the indegree of every reticulate node is exactly 2 (Figure A1).

Let N be a phylogenetic network. We use $\mathcal{V}(N)$ and $\mathcal{E}(N)$ to denote the node and edge set of N, respectively. We also use $\mathcal{R}(N)$ to denote the set of reticulate nodes, and use $\mathcal{T}(N)$ to denote the set of all non-reticulate nodes, including the root, tree nodes and leaves. Let $u, v \in \mathcal{V}(N)$. The node v is a *child* of u if (u, v) is an edge; v is a *descendant* of u if there is a directed path from u to v. If v is a descendant of u, v is said to be *below* u.

A phylogenetic network N is a *tree-child network* if every non-leaf node has a child that is not reticulate. Equivalently, N is a tree-child network if and only if for every non-leaf node, there is a path from that node to some leaf that passes only tree nodes. Figure A1 presents a binary tree-child network (left) and two non-tree-child networks.

Consider a tree-child network N with k reticulate nodes. Let the root be r_0 and let the reticulate nodes be $r_1, r_2, \cdots, r_k$. After the removal of the incoming edges of every r_i, N becomes the union of $k + 1$ subtrees, which are rooted at $r_0, r_1, \cdots, r_k$, respectively, and have network leaves as their leaves (see Figure A1). These subtrees are called the *tree-node components* of N. Tree-node decomposition is a useful technique in the study of phylogenetic networks.

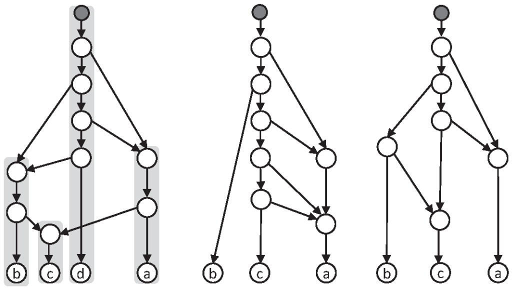

Figure A1: A binary tree-child network (left) in which there are four tree-node components (shaded grey) and two non-tree-child networks (middle) and (right). In the middle network, the child of the top reticulate node is also reticulate. In the right network, the children of a tree node in the middle are both reticulate.

Phylogenetic trees

A phylogenetic tree on X is a phylogenetic network with no reticulate nodes. In fact, a tree is a tree-child network. Let T be a phylogenetic tree on X and $u \in V(T)$. The *node cluster* of u, denoted as $C(u)$, is the subset of taxa that are represented by the leaves below u. Clearly, $C(u) \cap C(v) \in \{C(u), C(v), \emptyset\}$ for any two nodes u and v. The node u and its descendants induce a unique subtree on $C(u)$. We use T_u or $T(C(u))$ to denote the subtree.

Let S be a set of binary phylogenetic trees on X. A *common cluster* of S is a subset of X that is a node cluster in every tree of S. Obviously, each single taxon is common cluster of S, and so is X. Any other common clusters of S are called *non-trivial common clusters. S* is a *reducible* tree set if there is a non-trivial common cluster for S, and it is *irreducible* otherwise. A non-trivial common cluster C of S is *maximal* if any subset C' such that $C \subset C' \subset X$ is not a common cluster of S. Clearly, for any two maximal common cluster C_1 and C_2 of S, $C_1 \cap C_2 = \emptyset$; and any non-trivial common cluster X' of S must be contained in a unique maximal cluster of S if X' is not maximal.

Tree display and network inference problems

Let T be a binary phylogenetic tree on X and let N be a tree-child network with k reticulate nodes on X. T is *displayed* by N if T can be obtained from N by applying edge contraction from N after the removal of all but one incoming edge for each reticulation node (Figure 1). For any set of binary phylogenetic trees over X, there is always a tree-child network that displays all the trees [19]. However, such a solution network may not be binary.

Let P by a phylogenetic network. Its *reticulate number* is defined as the number of reticulate nodes. Its *HN*, denoted as $H(P)$, is defined as the sum over all the reticulate nodes of the difference between the indegree and the outdegree of that reticulate node. If P is binary, $H(P)$ is equal to the reticulate number. Here, we studied the following *minimum tree-child network inference* problem:

Input: A set of phylogenetic trees on X.
Output: A parsimonious tree-child network P on X (with the smallest $H(P)$) that displays all input trees.

The SCS problem

Let s and t be two sequences in an alphabet. The sequence s is said to be a supersequence of t if t can be obtained from s by the deletion of one or more letters. The *SCS problem* is, given a set of sequences, to find the shortest sequence that is a supersequence of every given sequence.

The SCS problem can be solved in a quadratic time for two sequences. However, it is NP-hard in general.

Total ordering

Let X be a set of taxa. A (total) ordering R on X is a binary relation on X such that (i) R is anti-symmetric, i.e. if $x_1 R x_2$, then $x_2 \not R\, x_1$. (ii) R is transitive, i.e., if $x_1 R x_2$ and $x_2 R x_3$, then $x_1 R x_3$. (iii) For any x_1, x_2, $x_1 R x_2$ or $x_2 R x_1$. For convention, we write $x <_R y$ if x is related y under R or even $x < y$ if R is clear.

Any non-empty subset X' of X whose elements are ordered according to R has a unique minimum (resp. maximum) element. We use $\min_R X'$ (resp. $\max_R X'$) to denote the minimum (resp. maximum) element of X'.

Let $X = \{x_1, x_2, \cdots, x_n\}$. We use $\pi = \pi_1 \pi_2 \cdots \pi_n$ on $\{1, 2, .., n\}$ to denote the following ordering:

$$x_{\pi_1} < x_{\pi_2} < \cdots < x_{\pi_n}.$$

A2. Proof of Propositions

We use the following algorithm to derive another representation of a phylogenetic tree on $|X|$ given an ordering on X.

```
LABELING

Input A tree $T$ on $X$ and an ordering $\pi$ of $X$

1. Label the degree-1 root of $T$ by $\min_\pi(X)$.
2. Label each internal node $u$ with two children $v$ and $w$ with
     $\max_\pi\{\min_\pi(C(v)), \min_\pi(C(w))\}$, where $C(v)$ consists of all taxa below $v$ in $T$.
```

For each taxon τ, a unique internal node w is labeled with τ by applying the LABELING algorithm. The node w is an ancestor of the leaf τ. Let $Z_{w\tau}$ be the directed path from w to the leaf τ in the tree. The sequence of the labels of the nodes appearing between w and the leaf in the path $Z_{w\tau}$ is called *the lineage taxon string* (LTS) of τ.

Proposition 1. Let π be an ordering of X, $|X| > 1$. For a phylogenetic tree T on X, the LTS $s_\pi(t)$ of each taxon t obtained w.r..t π by applying the LABELING algorithm in T has the following properties:

(i) $s_\pi(\pi_1)$ is always not empty;

(ii) $s_\pi(\pi_n)$ is always empty;

(iii) for each $1 < i \leq n$, π_i appears in the LTS of π_j for a unique j such that $j < i$;

(iv) the smallest taxon π_1 does not appear in any LTS.

Proof. Let the degree-1 root of T be ρ. Let the ancestors of Leaf π_1 be:

$$\rho = u_0, u_1, u_2, \cdots, u_k$$

and $u_{k+1} = \pi_1$, where u_i is the parent of u_{i+1} for $0 \leq i \leq k$. Recall that each non-leaf, non-root node has two children. We let u'_{i+1} be another child of u_i for $0 \leq i \leq k$.

(i) Since $|X| > 1$, $k \geq 1$. Clearly, $\min_\pi C(u_i) = \pi_1$ for each $i \leq k$. Since π_1 is the smallest taxon, in Step 2 of the LABELING algorithm, u_i is labeled with $\max_\pi\{\min_\pi(u_{i+1}), \min_\pi(u'_{i+1})\} = \min_\pi(u'_{i+1})$ for $i = 1, 2, \cdots, k$. Therefore, that $k \geq 1$ implies that $s_\pi(\pi_1)$ contains at least one taxon.

(ii) Let the parent and sibling of Leaf π_n be v and v'. In Step 2 of the LABELING algorithm, v is labeled with $\max_\pi\{\min_\pi(v'), \pi_n\} = \pi_n$. Since there is no node between v and Leaf π_n, $s_\pi(\pi_n)$ is empty.

(iii) and (iv) We prove the statement by mathematical induction. If $|X| = 2$, clearly, the root ρ_T is labeled with π_1 and the other internal node is labeled with π_2. In this case, $s_\pi(1)$ contains only π_2 and $s_\pi(2)$ is empty. Thus, the fact is true.

For $|X| > 2$, from the proof of Part (i), we have that u_i is labeled with the minimum taxon appearing in $C(u'_{i+1})$ for $i = 1, 2, \cdots, k$. Moreover, the internal nodes in each subtree T'_i rooted at u'_i are labeled with the taxa of $C(u'_i) \setminus \{ \min_\pi C(u'_i) \}$ according to the algorithm. Since each T'_i is a proper subtree of T_i, by induction, the fact holds. □

Remark. The LTSs of the taxa obtained according to an ordering on X determine a unique phylogenetic tree T. This can be generalized to an algorithm to reconstruct a tree-child network using the LTSs of taxa.

Tree-Child Network Construction

1. (**Vertical edges**) For each β_i, define a path P_i with $|\beta_i| + 2$ nodes:
   $h_i, v_{i1}, v_{i2}, \cdots, v_{i|\beta_i|}, \ell_{\pi_i}$, where β_n is the empty sequence.
2. (**Left–right edges**) Arrange the n paths from left to right as $P_1, P_2, \cdots, P_n$. If the m-th letter of β_i is π_j, we add an edge (v_{im}, h_j) for each m and each i.
3. Contract each h_i $(i > 1)$ if it is of indegree 1 and outdegree 1.

Proposition 2. Let T_i $(1 \leq i \leq k)$ be k trees on X such that $|X| = n$ and π be an ordering on X. Let $\alpha_{ij} = \beta_{T_i,\pi}(\pi_j)$, the LTS of π_j with respect to π in T_i for each j, $1 \leq j \leq n-1$. If β_j is a common supersequence of $\alpha_{1j}, \alpha_{2j}, \cdots, \alpha_{kj}$ for each j, the Tree-Child Network Construction algorithm outputs a tree-child network that displays the k trees.

Proof. Let N be the directed network constructed by applying the algorithm to $\beta_1, \beta_2, \cdots, \beta_k$. First, N is acyclic due to the two facts: (i) the edges of each path P_i are oriented downwards, and (ii) the so-called left–right edges (u, v) are oriented from a node u in a path defined for π_i to a node v in a path defined for π_j such that $i < j$.

Second, N is tree-child. This is because all the nodes of each P_i are tree nodes except h_i for each $i > 1$ (see Figure 3 in main text). The node h_1 is the network root. For $i > 1$, h_i may or may not be a reticulation node. Therefore, every non-leaf node has a child that is not reticulate.

Lastly, we prove that T_i is displayed by N as follows. By assumption, β_j is a supersequence of $\{\alpha_{ij} \mid i = 1, 2, \cdots, k\}$ for each $j = 1, 2, \cdots, n-1$. Following the notation used in the Tree-Child Network Construction algorithm, we let:

$$\beta_j = \beta_{j1}\beta_{j2}\cdots\beta_{jt_j}, \ \ t_j \geq 1,$$

where t_j is the length of β_j. Since α_{ij} is a subsequence of β_j, there is an increasing subsequence $1 \leq m_1 < m_2 < \cdots < m_{\ell_j} \leq t_j$ such that

$$\alpha_{ij} = \beta_{im_1}\beta_{im_2}\cdots\beta_{im_{\ell_j}}$$

and $\ell_j = |\alpha_{ij}| \geq 1$.

According to Step 1 of the algorithm, in N, each taxon β_{jx} of β_j corresponds one-to-one a node v_{jx} in the path P_j; and there is a (left-right) edge from v_{jx} to the first node $h_{y(x)}$ of the path $P_{y(x)}$ that ends with the taxon $\pi_{y(x)} = \beta_{jx}$, where $y(x) \geq j$.

Conversely, after removing the edge $(v_{jx}, h_{y(x)})$ for each $x \neq m_1, m_2, \cdots, m_{\ell_j}$, we obtain a subtree T'_i of N. This is because each taxon π_t appears exactly once in $\alpha_{i1}, \alpha_{i2}, \cdots, \alpha_{i(n-1)}$ and thus the node h_t is of indegree 1 in the resulting subgraph, where $t = 2, 3, \cdots, n$. It is not hard to see that after contracting degree-2 nodes of T'_i, the resulting subtree T''_i has the same LTS as T_i for each π_j. Thus T''_i is equal to T_i. □

Definition 1. Let P be a phylogenetic network on X, where $|X| > 1$ and π be an ordering on X. P is said to be *compatible* with π if for each reticulate edge (s, r) of P, the minimum taxon below s in the tree-node component C_s is less than the minimum taxon in the tree-node component C_r.

Remark. For a tree-child network P, we can construct a compatible ordering π as follows. We first compute a topological sorting on the vertices of P. Assume the reticulate nodes and the network root ρ appear in the sorted list as: $r_0 = \rho, r_1, r_2, \cdots, r_k$. We construct a desired ordering by listing the taxa in the tree-node component C_{r_i} before the taxa in the tree-node component $C_{r_{i+1}}$ for every $i \leq k-1$.

Let π be an ordering on X and P be a tree-child network on X that is compatible with π. The compatibility property implies that the smallest taxon is in the tree-node component C_ρ that is rooted at the network root ρ. We use the following generalized Labelling algorithm to label all the tree nodes of P, which is identical to Labelling when P is a phylogenetic tree.

Generalized Labelling

S1: For every reticulate node r, label all parents of r with the smallest taxon in the tree-node component C_r. Similarly, the network root ρ is labeled with the smallest taxon in C_ρ.

S2: For each tree node z that is not a parent of any reticulate node, label x with $\max_\pi(\min_\pi(C(x)), \min_\pi(C(y))$, where x and y are the two children of z, and $C(x)$ and $C(y)$ are the set of taxa below x and y in the tree-node component where they belong to.

Proposition 3. Let $T_1, T_2, \cdots, T_k$ be k trees on X and P be a tree-child network on X with the smallest $H(P)$, compared with those displaying all T_i. For any ordering Π of X such that P is compatible with it, if we label the tree nodes of P using the Generalized Labelling algorithm, the LTS $\beta_{P,\Pi}(x)$ obtained for each taxon x is a shortest common supersequence of $\{\beta_{T_i,\Pi}(x) \mid i = 1, 2, \cdots, k\}$. Moreover, applying the Tree-child Construction algorithm to the obtained supersequences $\beta_{P,\Pi}(x)$ produces the same network as P.

The proof of Proposition 3 is divided into several lemmas.

Lemma 1. Let π be an ordering on X and let $T_1, T_2, \cdots, T_k$ be k phylogenetic trees on X. For each $x \in X$ and each T_i, we use $\beta_x(T_i, \pi)$ to denote the LTS of x obtained w.r.t. π using the Labelling algorithm in T_i. Assume β_x is a common supersequence of $\{\beta_x(T_1, \pi), \beta_x(T_2, \pi), \cdots, \beta_x(T_k, \pi)\}$

for each $x \in X$. For the tree-child network P constructed from $\{\beta_x \mid x \in X\}$ by using the Tree-Child Network Construction algorithm, $H(P) = \sum_{x \in X} |\beta_x| - |X| + 1$.

Proof. Since only the first node h_i of each path can be a reticulate node and that each node in the middle of each path is a parent of some h_i, $H(P) = \sum_{i=2}^{|X|}(d_{in}(h_i) - 1) = \sum_{x \in X} |\beta_x| - |X| + 1$, where $d_{in}(h_i)$ is the indegree of h_i. □

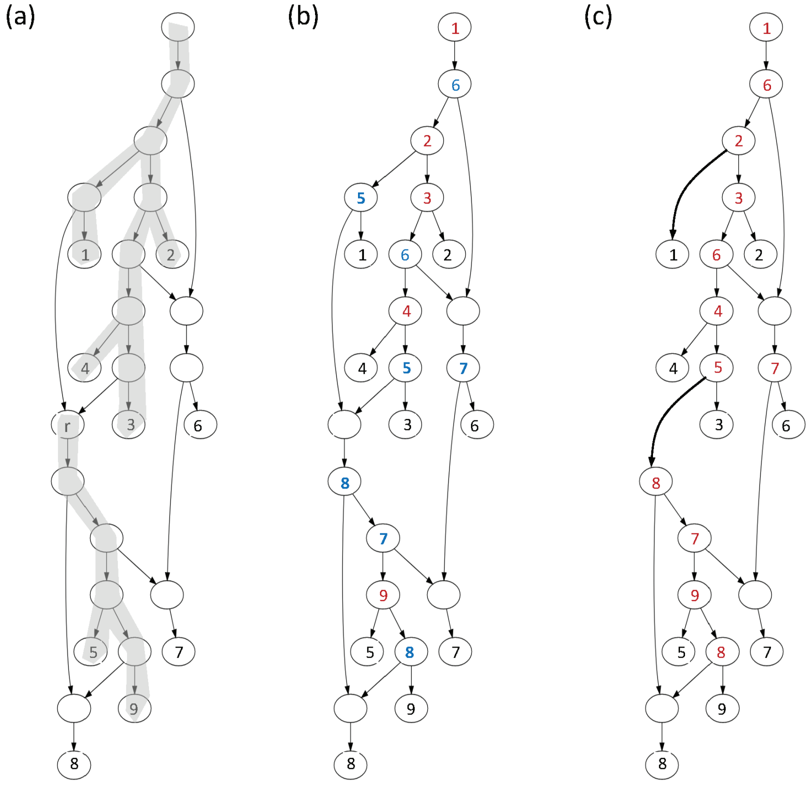

Figure A2: Illustration of the Generalized Labelling algorithm and the proof of Lemma 3. (a) A tree-child network on the taxa from 1 to 9, which has two tree-node components each containing at least two taxa. (b) Labelling all the tree nodes in a tree-child network using the increasing order of taxa: $i < i + 1, i = 1, 2, ..., 8$, which is compatible. The labels of the parents of a reticulation node are in blue; while the labels of other tree-nodes are in red. (c) the resulting network after the removal of the left incoming edge of the reticulation node r, in which the tree-nodes are labeled identically if the same ordering is used.

Lemma 2. Let C be a tree-node component of P and let it contain t taxa $x_1, x_2, \cdots, x_t$ in P. All $t-1$ tree nodes that are not a parent of any reticulate node are uniquely labeled with some $x_j \neq \min_\pi\{x_i \mid 1 \leq i \leq t\}$ (red labels in Figure A2b).

Proof. This can be proved using the same mathematical induction as in Prop. 1.iii. □

Definition 2. Let π be an ordering on X and N be a tree-child network on X that is compatible with π. Assume the tree nodes of N are labeled by using the Generalized Labelling algorithm. The LTS of a taxon x obtained according to π is defined to be the sequence of the labels of the x's ancestors that are a tree node in C_x, if x is the smallest taxon in C; it is the sequence of the labels of the x's ancestors that are a tree-node below the unique tree node labeled with x in C_x otherwise. The LTS of x obtained in this way is denoted by $\beta_{N,\pi}(x)$.

Definition 3. Let P be a tree-child network on X and let (s, r) be a reticulate edge. $P-(s,r)$ is defined to be the tree-child network obtained through the removal of (s,r) and contraction of s (and also r if r is of indegree 2 in N).

Lemma 3. Let π be an ordering on X and P be a tree-child network on X such that $H(P) \geq 1$ and P is compatible with π. For any reticulate node r and each parent s of r, the tree-child network $P-(s,r)$ has the following properties:

1. $P-(s,r)$ is also compatible with π;

2. For each taxon x, $\beta_{P,\pi}(x)$ is a supersequence of $\beta_{P-(s,r),\pi}(x)$.

Proof. These properties are illustrated in Figure A2. Let (s,r) be a reticulate edge. We have that s is a tree node, and r is a reticulate node. Recall that $C_N(z)$ denotes the tree-node component containing z for each node z and for $N = P$, or $P-(s,r)$. We consider the two cases.

Case 1. The r is of indegree 3 or more.

In this case, after (s,r) is removed, s will be contracted and all the other nodes remains the same in $P-(s,r)$. Moreover, $P-(s,r)$ has the same tree-nodes components as P and also has the same labelling as P. For any reticulate edge (s',r'), $C_{P-(s,r)}(s') = C_P(s')$ and $C_{P-(s,r)}(r') = C_P(r')$. As such, the constraint is also satisfied for (s',r') in $P-(s,r)$. Therefore, the first fact holds.

Let x be a taxon. If $\beta_{P,\pi}(x)$ contains the label y of s, say $\beta_{P,\pi}(x) = \beta_1 y \beta_2$, then, $\beta_{P-(s,r),\pi}(x) = \beta_1\beta_2$. If $\beta_{P,\pi}(x)$ does not contain the label of s, $\beta_{P-(s,r),\pi}(x) = \beta_{P,\pi}(x)$. This concludes that $\beta_{P,\pi}(x)$ is a supersequence of $\beta_{P-(s,r),\pi}(x)$. Therefore the second fact is true.

Case 2. The r is of indegree 2.

This case is illustrated in Figure A2b. Let s' be another parent of r. After (s,r) is removed, the r becomes a node of indegree 1 and outdegree 1 and thus is contracted, together with s. All the other nodes remains in $P-(s,r)$. Therefore, s' becomes a tree node in $P-(s,r)$. The tree-node component $C_{P-(s,r)}(s')$ is the merge of $C_P(s')$ and $C_P(r)$. Assume (s'',r') be a reticulate edge of $P-(s,r)$.

If $C_{P-(s,r)}(s'') \neq C_{P-(s,r)}(s')$ and $C_{P-(s,r)}(r') \neq C_{P-(s,r)}(s')$, then, $C_{P-(s,r)}(s'') = C_P(s'')$ and $C_{P-(s,r)}(r') = C_P(r')$. The constraint is satisfied for (s'',r').

If $C_{P-(s,r)}(s'') \neq C_{P-(s,r)}(s')$ and $C_{P-(s,r)}(r') = C_{P-(s,r)}(s')$, the constraint is satisfied for s'',r' because of the fact that $\min_\pi C_{P-(s,r)}(r') = \min_\pi C_P(r')$.

If $C_{P-(s,r)}(s'') = C_{P-(s,r)}(s')$ and $C_{P-(s,r)}(r') \neq C_{P-(s,r)}(s')$, then the minimum taxon below s'' in $C_{P-(s,r)}(s'')$ is equal to that in $C_P(s'')$, the constraint is satisfied for (s'',r').

We have proved the first statement. We prove the second statement as follows. To this end, we use $c_P(r)$ to denote the unique child of r in P.

Recall that after (s,r) was removed, s and r were contracted to obtain $P-(s,r)$. Note that in $P-(s,r)$, s' becomes the parent of $c_P(r)$. Since P is compatible with π, the minimum taxon y below $c_P(r)$ is larger than the minimum taxon below s' in π. This implies that s' is labeled with y, as s' is not a parent of any reticulate node in $P-(s,r)$. Therefore, for any taxon $x \in X$, if $\beta_{P,\pi}(x)$ contains the label y of s, say $\beta_{P,\pi}(x) = \beta_1 y \beta_2$, then, $\beta_{P-(s,r),\pi}(x) = \beta_1\beta_2$. If $\beta_{P,\pi}(x)$ does not contain the label of s, $\beta_{P-(s,r),\pi}(x) = \beta_{P,\pi}(x)$. This concludes that $\beta_{P,\pi}(x)$ is a supersequence of $\beta_{P-(s,r),\pi}(x)$ for each $x \in X$. □

Proof of Proposition 3. Let P be a tree-child network on X with the smallest $H(P)$, compared with those displaying all T_i. For each i, T_i can be obtained from P by deleting all but one incoming edge for each reticulate node. For convention, we assume that all removed reticulate edges are (s_j,r_j), $1 \leq j \leq H(P)$. Let x be a taxon. By Lemma 3, $\beta_{P,\Pi}(x)$ is a supersequence of $\beta_{P-(s_1,r_1),\Pi}(x)$ and $\beta_{P-\sum_{t=1}^{j}(s_t,r_t),\Pi}(x)$ is a supersequence of $\beta_{P-\sum_{t=1}^{j+1}(s_t,r_t),\Pi}(x)$ for each $j=1,..,H(P)-1$. Therefore, for any x, $\beta_{P,\Pi}(x)$ is a supersequence of $\beta_{T_i,\pi}(x)$ for each T_i, as $T_i = P - \sum_{j=1}^{H(P)}(s_j,r_j)$.

Let P contain m reticulate nodes. P has $m+1$ tree-node components. In a tree-node component C, there are $|X(C)|-1$ tree nodes that are not the parents of any reticulation nodes, where $X(C)$ is the set of taxa in C. Hence

$$\begin{aligned}
& \sum_{x\in X} |\beta_{P,\Pi}(x)| \\
= \;& \sum_{C}(|X(C)|-1) + \sum_{r\in\mathcal{R}(P)} d_{in}(r) \\
= \;& |X| - (m+1) + H(P) + m \\
= \;& |X| - 1 + H(P).
\end{aligned}$$

This implies that $H(P) = \sum_{x\in X} |\beta_{P,\Pi}(x)| - |X| + 1$.

Assume $\beta_{P,\Pi}(x)$ is not a shortest supersequence of $\beta_{T_i,\Pi}(x)$ $(i=1,2,\cdots,k)$ for some x. Let β_x

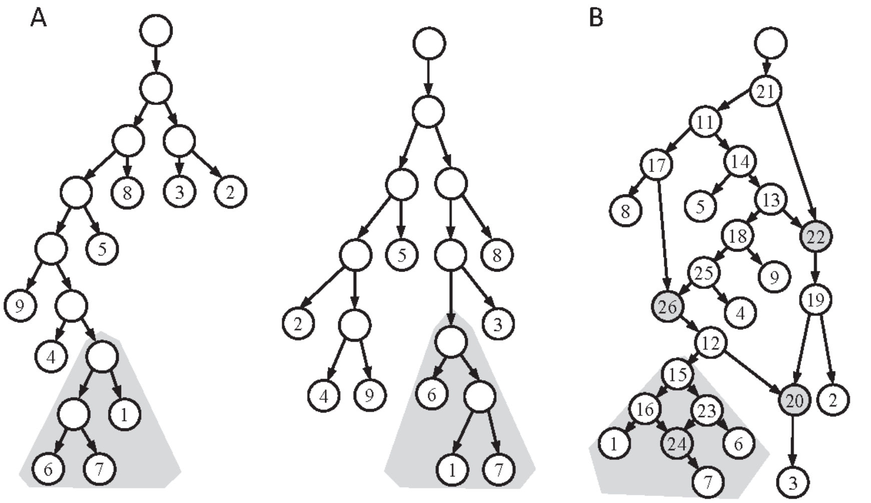

Figure A3: A. Two input trees over taxa 1–9 that contain an identical node cluster: (1, 6, 7). B. A tree-child network that display both input trees, which is a merge of two tree-child networks.

be a shortest supersequence of $\beta_{T_i,\Pi}(x)$ $(i = 1, 2, \cdots, k)$. Then, $|\beta_x| < |\beta_{P,\Pi}(x)|$. By Lemma 1, we can use the Tree-Child Network Construction algorithm to obtain a tree-child network with the HN smaller than $H(P)$, a contradiction.

It is obvious that the we obtain P if the Tree-Child Network Construction algorithm is applied to the LTSs $\beta_{P,\Pi}(x)$ of the taxa x. □

B. Reduction for a reducible tree set

A set of multiple trees is reducible if there is a non-trivial node cluster that appears in every tree and is irreducible otherwise. ON way for improving the scalability is to decompose the input tree set into irreducible sets of trees if the input trees are reducible.

Let S be a reducible set of k trees on X, which are ordered as: $\langle T_1, T_2, \cdots T_k\rangle$. We assume that $C_1, C_2, \cdots, C_t$ are all the maximal common clusters of S. We introduce t new taxa y_i and let $Y = \{y_1, y_2, \cdots, y_t\}$. By replacing $T_i(C_j)$ with y_j in T_i for each i and j, we obtain a set S' of k trees T'_i on $Y \cup \left[X \setminus \left(\cup_{i=1}^{t} C_i\right)\right]$. In this way, we decompose S into an irreducible tree set $S' = \langle T'_1, T'_2, \cdots, T'_k\rangle$ and t ordered sets of trees $S'_i = \langle T_1(C_i), T_2(C_i), \cdots, T_k(C_i)\rangle$, $1 \le i \le t$. Combining the tree-child networks constructed from S' and all of S'_i gives tree-child networks that display all the trees of S, as shown in Figure A3.

C. Computing the branch weights of the inferred tree-child network

A phylogenetic network is weighted if every branch has a non-negative value, which represents time or other evolutionary measures. A weighted phylogenetic tree T is said to be displayed in a weighted network N if the tree is displayed in the network when the branch weights are ignored. For a display T' of T in N, its *fitness score* $||T-T'||_2$ is defined as $\sqrt{\sum_{e\in E(T)} |w_T(e) - w_{T'}(P(u',v'))|^2}$, where $w_T(e)$ is the weight of $e=(u,v)$ in T and $w_{T'}(P(u',v'))$ is the weight of the unique path between the images u' and v' of u and v under the display mapping, respectively.

Recall that a tree can be displayed multiple times in a network. The *score* of the display of T in N is the smallest fitness score which a display of T in N can have, denoted $d(T,N)$. If $d(T,N)=0$, we say that N perfectly displays T.

If the input trees are weighted, we will first compute tree-child networks that each display all the trees. We then use branch weights of trees and the information on how the trees are displayed in a tree-child network to compute the weights of the network branches.

We model the branch weight assignment problem as an optimization problem with the following assumption on the inferred tree-child network N that displays all the trees:

> For any reticulate edge e, the tree-child network $P-e$ obtained after removal of e fails to display one input tree at least.

By ordering the edges of N on X, we may assume

$$E(N) = \{e_1, e_2, \cdots, e_m\}.$$

Let $S=\{T_1, T_2, \cdots, T_s\}$, where $|S|=s$. We further assume that T'_k is a display of T_k in N. Then, each edge e'_i of T_k is mapped to a path P'_i of T'_k, where $1 \le i \le 2|X|-2$. Since N displays T_i, we derive the following linear equation system from the display of T_k:

$$\sum_{1\le j\le m} a_{ij} w(e_j) = w(e'_i), \;\; i = 1, 2, \cdots, 2|X|-2, \tag{1}$$

where

$$a_{ij} = \begin{cases} 1 & e_j \in E(P'_i); \\ 0 & e_j \notin E(P'_i). \end{cases}$$

Let the coefficient matrix of Eqn. (1) be $A_k = (a_{ij})$, which is a $(2|X|-2) \times m$ matrix, and let:

$$W_k = \begin{pmatrix} w(e'_1) \\ w(e'_2) \\ \vdots \\ w\left(e'_{2|X|-2}\right) \end{pmatrix}.$$

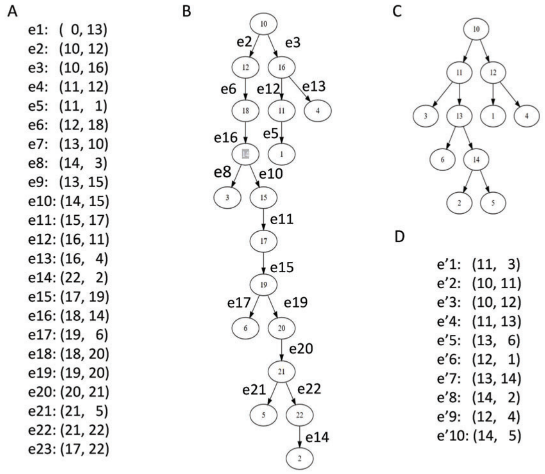

Figure A4: An illustration of how to derive linear equations from a tree display. (A) The list of the edges of a tree-child network. (B) A display of the tree in C. (C) a phylogenetic tree on six taxa (1 to 6). (D) the list of the edges of the tree in C.

Since N displays every tree of S, we then determine the edge weights of N by solving the following linear equation system:

$$\begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_s \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} W_1 \\ W_2 \\ \vdots \\ W_s \end{pmatrix} \tag{2}$$

Note that Eqn. (2) is a linear equation system that contains $2s(|X|-1)$ equations and at most $5|X|-4$ variable, as each T_i contains $2|X|-2$ edges and N contains $3r+2|X|-1$, where r is the number of reticulations, which is at most $|X|-1$.

Example 1. The edge list of a tree-child network is given in Figure A4A, where the full network is not given here. Figure A4B presents a particular display of the tree in Figure A4C, whose edges are listed in Figure A4D. In the display of the tree, the edge e'_2 is mapped to the path from the node 10 to the node 14, which consists of three edges $e2, e6, e16$ (Figure A4B). From e'_2 and its image, we obtain the following equation in the linear equation system Eqn. (2):

$$x_2 + x_4 + x_{16} = w(e'_2).$$

In general, N may not perfectly display every T when branch weights are considered. Therefore, let us set:

$$A = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_s \end{pmatrix} \tag{3}$$

$$W = \begin{pmatrix} W_1 \\ W_2 \\ \vdots \\ W_s \end{pmatrix}. \tag{4}$$

Noticing that

$$\sum_{i=1}^{s} ||T_i' - T_i||_2^2 = ||AX - W||_2^2,$$

we determine the branch weights of N by solving the following quadratic optimization problem:

$$\min ||AX - W||_2^2 \tag{5}$$

subject to:

$$x_j \geq 0, \;\; 1 \leq j \leq m. \tag{6}$$

Remark. Let r be a reticulation node that has incoming $e_1, e_2, \cdots, e_d$ and the outgoing e_{d+1}. For each input tree T_i, there is exactly one of edge pairs (e_1, e_{d+1}), (e_2, e_{d+1}), $\cdots$, (e_d, e_{d+1}) appearing in the display of T_i. Thus, solving the above optimization problem can only determine the value of $w(e_i) + w(e_{d+1})$ for $i \leq d$.

C. A phylogenetic network for hominin relationships

We analysed the morphological data in Dembo et al. (Proc Royal Soc B: Biol. Sci., vol. 282, 2015) by sampling 500 phylogenetic trees from a posterior collection of trees estimated from the morphological data. We computed the distance between each pair of trees using the rooted tree metric described in Kendall and Colijn (Mol. Biol. Evol., vol. 33, 2016). Briefly, this metric is the Euclidean distance between two vectors (one for each tree). The vector captures the amount of shared ancestry between each pair of tips, as well as each tip's distance from its parent. We used the tree topology only ($\lambda = 0$ in the tree metric in the 'treespace' function in the 'treespace' package in R (Jombart et al., Mol. Ecol. Resour., vol. 17, 2017)). The amount of shared ancestry is the length of the path (in a phylogeny) between the root and the most recent common ancestor of a pair of tips. Having found pairwise distances between all pairs of trees in our sample of 500, we clustered the trees into five clusters using Ward clustering. We chose two trees uniformly at random from each of the five clusters, as input for the analysis presented here.

Hominins' phylogenetic relationships are not fully established. Due to the nature of the morphological data, the trees were discordant, and no single tree captures a highly-supported pattern of ancestry among the taxa. This motivates using a network to illustrate the complex ancestral relationships among these data. Using ALTS, we reconstructed a network model (Figure A5) for hominin species using the 10 phylogenetic trees.

The resulting network model contains 12 reticulation events. The top tree-node component contains the two outgroup species *G. gorilla* and *P. troglodytes*, as well as the oldest hominin species, *S. tchadensis*. The three earliest members of the genus *Homo* (African *H. erectus*, *H. rudolfensis* and *H. habilis*), together with *Au. africanus*, appear in a tree-node component, whereas four recent members of the genus *Homo* (*H. heidelbergensis*, *H. neanderthalensis*, *H. sapiens* and *H. naledi*) compose another tree-node component. The three members of the genus *Paranthropus*, together with *Au. garhi*, compose a tree-node component. The model also reflects the high uncertainty about the phylogenetic position of *H. floresiensis*, who lived in the island of Flores, Indonesia (Argue et al., J Human Evol., vol. 57, 2009).

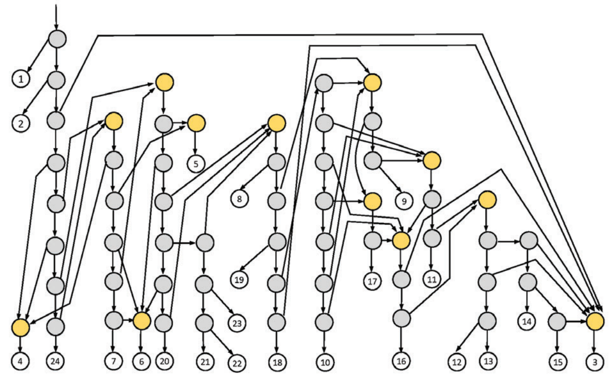

Figure A5: A network model of hominin relationships. 1: *G. gorilla*; 2: *P. troglodytes*; 3: *H. floresiensis*; 4: *Ar. ramidus*; 5: *Au. anamensis*; 6: *Au. afarensis*; 7: *K. platyops*; 8: *Au. africanus*; 9: *Au. sediba*; 10: African *H. erectus*; 11: Asian *H. erectus*; 12: *H. heidelbergensis*; 13: *H. neanderthalensis*; 14: *H. sapiens*; 15: *H. naledi*; 16: *H. antecessor*; 17: Georgian *H. erectus*; 18: *H. rudolfensis*; 19: *H. habilis*; 20: *Au. garhi*; 21: *P. robustus*; 22: *P. boisei*; 23: *P. aethiopicus*; 24: *S. tchadensis*.