Reconstruction of Reticulate Networks from Gene Trees

Daniel H. Huson[1], Tobias Klöpper[1], Pete J. Lockhart[2], and Mike A. Steel[3]

[1] Center for Bioinformatics (ZBIT),
Tübingen University, Sand 14, 72076 Tübingen, Germany
[2] Institute of Molecular BioSciences,
Massey University, Palmerston North, New Zealand
[3] Biomathematics Research Centre,
University of Canterbury, Christchurch, New Zealand

Abstract. One of the simplest evolutionary models has molecular sequences evolving from a common ancestor down a bifurcating phylogenetic tree, experiencing point-mutations along the way. However, empirical analyses of different genes indicate that the evolution of genomes is often more complex than can be represented by such a model. Thus, the following problem is of significant interest in molecular evolution: Given a set of molecular sequences, compute a reticulate network that explains the data using a minimal number of reticulations. This paper makes four contributions toward solving this problem. First, it shows that there exists a one-to-one correspondence between the tangles in a reticulate network, the connected components of the associated incompatibility graph and the netted components of the associated splits graph. Second, it provides an algorithm that computes a most parsimonious reticulate network in polynomial time, if the reticulations contained in any tangle have a certain overlapping property, and if the number of reticulations contained in any given tangle is bounded by a constant. Third, an algorithm for drawing reticulate networks is described and a robust and flexible implementation of the algorithms is provided. Fourth, the paper presents a statistical test for distinguishing between reticulations due to hybridization, and ones due to other events such as lineage sorting or tree-estimation error.

1 Introduction

One of the most powerful approaches for developing an understanding of the evolution of genomes is phylogenetic analysis of genes at independent loci. However, optimal phylogenetic reconstructions for such genes are not always concordant [1]. One possible reason for this is that at the level of organisms, hybridization between diverging evolutionary lineages is a fundamental process important in the evolution of organisms [2]. Unfortunately, at the level of individual gene analyses, the interpretation of hybrid genomes is complicated by phylogenetic error, gene conversion (events of non-reciprocal recombination) and lineage sorting. Despite

S. Miyano et al. (Eds.): RECOMB 2005, LNBI 3500, pp. 233–249, 2005.

this complexity, the importance of the issue has motivated the following problem: Given a set of phylogenetic data that have evolved under a reticulate model of evolution, what is the most efficient way to reconstruct the underlying reticulate network. There has been much interest in this topic, see e.g. [3, 4, 5, 6, 7, 8]. Computationally, the goal is to construct a reticulate network using a minimum number of reticulations that accounts for the given data. Alternatively, one may attempt to give lower or upper bounds for the number of reticulations required to explain the data [9, 10, 11, 7].

In this paper we describe a general frame-work for studying reticulate networks, which is based on the theory of splits and splits graphs [12, 13]. In this setting, a *reticulate network* N is a generalization of a phylogenetic tree in which we additionally allow certain *reticulation* nodes and edges. The set of *splits associated with* N is defined as $\Sigma = \bigcup_{T \in \mathcal{T}(N)} \Sigma(T)$, where $\mathcal{T}(N)$ is the set of all trees that are induced by N and $\Sigma(T)$ is the split encoding of T. We present four new results.

Our first main result is a Decomposition Theorem that implies the existence of a one-to-one correspondence between the tangles of a reticulate network N, the non-trivial connected components of the incompatibility graph $IG(\Sigma(N))$ and the netted components of the splits graph $SG(\Sigma(N))$. This is related to similar results that have been previously described in the context of recombination of binary sequences under the infinite sites model [14, 11], for which we give a new formulation, interpretation and proof.

We say that a reticulate network N has the *overlapping* property, if every set of tangled reticulations in N has the property that all reticulation cycles intersect "nicely" along a common "backbone". Our second main result is an algorithm that computes the most parsimonious reticulate network this type for a given input set in polynomial time, if we limit the number of reticulations contained in any given tangle to k.

Our third main result is an algorithm for drawing reticulate networks. We have developed a robust and flexible implementation of our approach, which is freely available as a plug-in for the program SplitsTree [15]. It takes as input either a set of trees, partial trees or splits and produces as output a (rooted or unrooted) phylogenetic network indicating both the splits contained in the input and also the possible ways to resolve each netted component of the splits graph into a collection of reticulations.

Our fourth main result is a new statistical test for distinguishing between reticulations due to hybridization, on the one hand, and ones due to other events such as lineage sorting or tree estimation error, on the other.

To illustrate our algorithms, we apply them to two different gene trees for New Zealand alpine *Ranunculus* (buttercups) species based on the nuclear ITS gene and the chloroplast J_{SA} region [16]. In an Appendix we provide a second application, namely to haplotype data for the alcohol dehydrogenase locus of *Drosophila melanogaster* [11, 17].

We would like to thank Dan Gusfield for a number of very useful discussions.

2 Phylogenetic Trees and Reticulate Networks

Let X denote a set of taxa. A *phylogenetic tree for* X, or X*-tree*, consists of a tree $T = (V, E)$ in which every node v is either a *leaf* of degree 1 or an *internal* node of degree ≥ 3, together with a node labeling $\nu : X \to V$ such that every leaf of T obtains a label [18]. Additionally, we may designate one of the taxa $o \in X$ to be an *outgroup* and then consider the tree to be "rooted" at the midpoint ρ of the pendant edge leading to $\nu(o)$, in the usual sense. We choose to define the root in this indirect way because our approach is based on the concept of splits, i.e. bipartitionings of the taxon set X, for which it is awkward to specify a root node explicitly.

Let X denote a set of taxa. A *reticulate network* $N = (V, E, \nu)$ consists of a graph (V, E) with node set V and edge set E and a labeling of the nodes by taxa $\nu : X \to V$. The node set $V = V_R \cup V_T$ is partitioned into a set of *reticulation nodes* V_R and *tree nodes* V_T, and the edge set $E = E_R \cup E_T$ is partitioned into a set of *reticulation edges* E_R and *tree edges* E_T. The labeling ν only assigns labels to nodes in V_T and every leaf of N obtains a label. Additionally, we require the following five properties:

(R1) All nodes have degree $\neq 2$.
(R2) Every reticulation node $v \in V_R$ is incident to precisely two reticulation edges, denoted by $p(v)$ and $q(v)$, respectively.
(R3) Every reticulation edge $e \in E_R$ is incident to exactly one reticulation node.
(R4) Every subgraph of N obtainable by deleting precisely one reticulation edge $p(v)$ or $q(v)$ for every reticulation node $v \in V_R$, is an X-tree. We will use $\mathcal{T}(N)$ to denote the set of all such trees *induced* by N.
(R5) We will always assume that an outgroup $o \in X$ has been specified and will require for all trees $T \in \mathcal{T}(N)$ that every reticulation node $v \in V_R$ is separated from $\nu(o)$ by either $p(v)$ or $q(v)$.

As in the case of trees, we will usually consider N to be rooted at the center of the pendant edge leading to the node $\nu(o)$ labeled by the outgroup o. In Figure 1 we show an example of a reticulate network N with three reticulations. In Figure 2(a–b) we show two different trees induced by N. We explicitly allow the graph to contain *unresolved* nodes of degree > 3, labeled internal nodes and nodes with multiple labels.

It follows from these definitions that each reticulation node (or *reticulation*, for short) $v \in V_R$ is contained in one or more cycles of the form $C = (v, p(v), w_1, e_1, \ldots, e_{k-1}, w_k, q(v), v)$, $C = (v, p(v), w_1, e_1, \ldots, e_{k-1}, w_k, v)$ or $C = (v, q(v), w_1, e_1, \ldots, e_{k-1}, v)$, with $w_i \in V$ and $e_i \in E \setminus \{p(v), q(v)\}$ for all i. Any such cycle C is called a *reticulation cycle* and we define its *backbone* as $B(C) = (w_1, e_1, \ldots, e_{k-1}, w_k)$. Note that a reticulation v possesses at most one reticulation cycle C whose backbone B contains only tree edges and in this case we call C a *tree cycle*.

We say that two different reticulations $v \in V_R$ and $v' \in V_R$ are *dependent*, if they are contained in reticulation cycles C and C', respectively, such that C and C' share at least one edge. Otherwise, they are called *independent*. (Previous

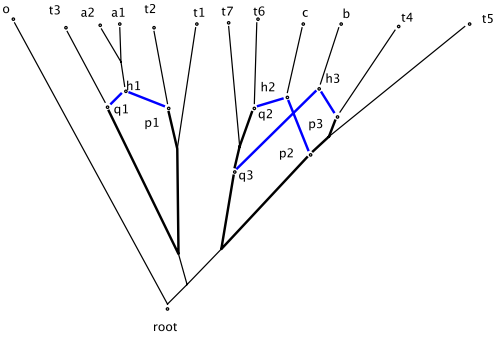

Fig. 1. A reticulate network N displaying three reticulations at nodes h_1, h_2 and h_3, each involving edges p_i and q_i, with $i = 1, 2, 3$, respectively. The reticulation at node h_1 is independent of the reticulations at nodes h_2 and h_3, whereas the latter two are tangled. The backbone edges of each reticulation are highlighted by heavier lines

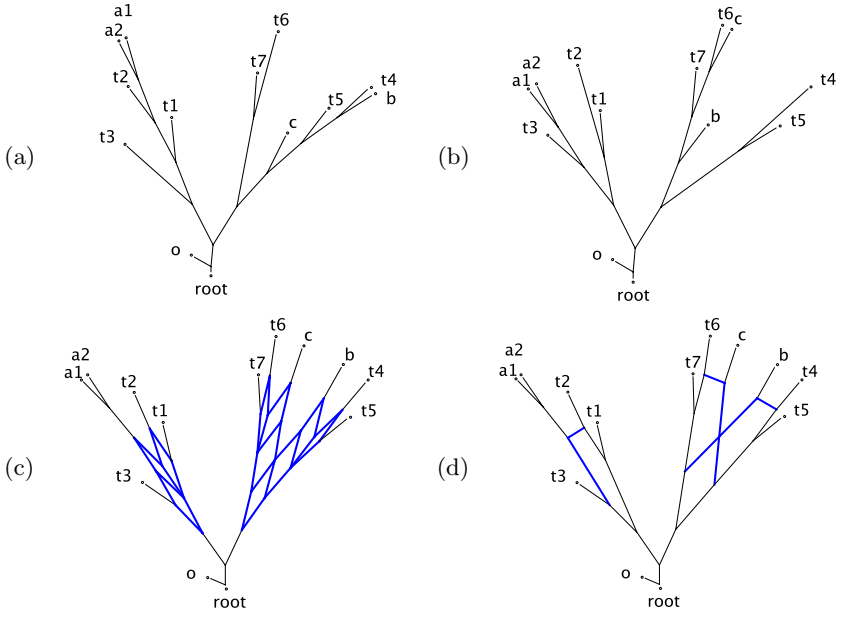

Fig. 2. Exactly eight different "gene trees" are induced by the reticulate network N in Figure 1, and we depict two such trees, T_1 in (a) and T_2 in (b). In (c) we show a splits graph SG representing the union of the splits $\Sigma(T_1) \cup \Sigma(T_2)$ of the two trees T_1 and T_2. This graph has two netted components, highlighted by heavy lines, and these correspond precisely to the two sets of tangled reticulations contained in N. In (d) we show the reticulate network N' reconstructed from SG using the algorithm described in the text, with reticulation edges highlighted by heavy lines

definitions of independence were more restrictive by requiring node-disjointness [5, 8].) Additionally, we call v and v' *tangled*, if there exists a chain of reticulation cycles $C = C_1, C_2, \ldots, C_k = C'$ such that C_i and C_{i+1} are dependent for each $i = 1, \ldots, k-1$. A reticulation that is independent of all other reticulations is also called a *gall* [6] and a reticulate network N that contains only galls is called a *galled tree*.

A reticulate network N gives rise to a *reticulate model* of evolution as follows: Starting at the root, molecular sequences evolve along tree edges in the usual fashion, experiencing point-mutations along the way [19]. However, the sequence that arises at a reticulation node v is obtained as a mixture of sequences along the two reticulation edges $p(v)$ and $q(v)$. The three main biological mechanisms that may operate here are *hybridization*, *horizontal gene transfer (HGT)* and *recombination*. In hybridization or HGT, we think of a sequence as consisting of an unordered set of genes and the net result of an hybridization or HGT event is a mixture of complementary genes or sites. In the case of recombination within a population, we consider individual sites ordered along the sequences and a resulting recombinant sequence is usually obtained by *cross-over* events, in which a prefix and a suffix of two ancestor sequences are combined together.

Our definitions capture the essence of the graphs used to describe hybridization and HGT scenarios [8], and the underlying graph employed to describe recombination scenarios [3, 5, 6]. However, the latter possess additional structure, namely a labeling of the tree edges by mutation sites and a labeling of the reticulation nodes by cross-over positions, and thus our approach requires further development before it will be able to fully address the reconstruction of recombination scenarios.

Throughout this paper we will use the term *gene* to mean a segment of sequence that is atomic with respect to the mechanism of reticulate evolution in operation. Hence, the evolutionary history of any given gene will be a single phylogenetic tree $T \in \mathcal{T}(N)$ [20].

3 Splits, Incompatibility and Splits Graphs

Suppose we are given a set of taxa X. A *split* (or, more precisely, X*-split*) is a bipartitioning of X into two non-empty sets A and B, denoted by $S = \frac{A}{B} (= \frac{B}{A})$.

For a given X-tree T, deletion of any single edge e will produce a graph with exactly two connected components and this defines a split $\sigma_T(e) = \frac{A}{B}$, given by the two sets of taxa labeling the two components [12]. The set of all splits obtainable in this way is called the *splits encoding* $\Sigma(T)$ of T. For a given set H of X-trees, we define $\Sigma(H) = \bigcup_{T \in H} \Sigma(T)$.

Two X-splits $S = \frac{A}{B}$ and $S' = \frac{A'}{B'}$ are called *compatible*, if one of the four possible intersections $A \cap A', A \cap B', B \cap A'$ and $B \cap B'$ is empty. A set of X-splits Σ is called *compatible*, if all pairs of splits in Σ are compatible. The *incompatibility graph* $IG(\Sigma) = (V, E)$ has node set $V = \Sigma$ and edge set $E \subseteq \binom{V}{2}$, in which any two nodes S and S' are connected, if and only if they are incompatible.

It is a well-known result that a set of X-splits Σ is pairwise compatible, if and only if there exists a unique X-tree T with $\Sigma = \Sigma(T)$. In this case we say that T *represents* Σ. Moreover, an arbitrary set of splits Σ, not necessarily compatible, can also be represented by a graph. Such a *splits graph* $SG(\Sigma)$ consists of a connected graph (V, E) together with a node labeling $\nu : X \to V$ and an edge coloring $\sigma : \Sigma \to E$, whose essential property is that deleting all edges colored by a given split $S = \frac{A}{B} \in \Sigma$ will produce precisely two connected components, labeled by A and B, respectively, see [13] for details. (This splits graph is not uniquely defined, however we will refer to it as *the* splits graph $SG(\Sigma)$ representing Σ, as the differences are inconsequential.) For example, the splits graph depicted in Figure 2 (c) represents the union of the split encodings of the two trees shown in Figure 2 (a–b).

Suppose we are given a set of splits Σ. A *netted component* Z of $SG(\Sigma)$ is a maximum set of nodes such that any two nodes $v, w \in Z$ are connected by two different node-disjoint paths in $SG(\Sigma)$ (in graph-theoretic terminology, a 2-connected component). The splits graph depicted in Figure 2 (c) has two netted components, each highlighted by heavy lines. Any node v that is contained in some netted component Z and is labeled, or is incident to an edge that is not contained in Z, is called a *gate node*. For example, in Figure 2(c), the left-hand netted component has precisely five gate nodes and the right-hand one has seven.

It is a simple observation that any two splits $S, S' \in \Sigma$ are incompatible, if any only if the edges representing S and S' are contained in the same netted component [12]. More precisely:

Lemma 1. *Suppose we are given a set of X-splits Σ. There exists a one-to-one correspondence between the netted components of the splits graph $SG(\Sigma)$ and the non-trivial connected components of the incompatibility graph $IG(\Sigma)$.*

4 Parsimonious Reconstruction Problem

An X-tree T' is called a *refinement* of an X-tree T, if $\Sigma(T) \subseteq \Sigma(T')$, that is, if $T' = T$ or if T can be obtained by contracting some edges of T'. Given a (usually unknown) reticulate network N. We say that a set of trees H was *sampled* from N, or that N *supports* H, if each tree $T \in H$ possesses a refinement $T' \in \mathcal{T}(N)$. Similarly, we say that a set of splits Σ' was *sampled* from N, or that N *supports* Σ', if $\Sigma' \subseteq \bigcup_{T \in \mathcal{T}(N)} \Sigma(T)$.

In molecular evolution we are interested in the following problem: Given a collection of gene trees that have evolved from a common ancestor under a reticulate model of evolution, reconstruct the underlying reticulate network. We address this as follows:

Problem 1. Parsimonious Reticulate Network from Gene Trees Problem: *Given a set of X-trees H. Construct a reticulate network N that supports H and contains a minimal number of reticulations.*

This is a purely combinatorial problem. It is easy to see that a solution always exists [21]. In practice a key issue is how to obtain a collection of sufficiently accurate gene trees that contain all edges necessary to be able to detect the reticulations in the underlying network. A second key issue is that, even if we can construct such a network N, it is not certain that a reticulation node v was indeed caused by a hybridization event, as we discuss in Section 8.

Problem 1 in its most general form is known to be computationally intractable [5]. The following simpler version of the problem is tractable and different solutions have been proposed [20, 8, 6, 7]:

Problem 2. Parsimonious Independent Reticulate Network from Gene Trees Problem: *Given a set of X-trees H. Construct a reticulate network N containing only independent reticulations that supports H and contains a minimal number of reticulations, if one exists.*

This problem is a special case of the following more general problem. Let N be a reticulate network. We say that two reticulation nodes v and v' *overlap*, if they possess tree cycles C and C', respectively, that intersect "nicely", that is, whose backbones B and B' overlap either in prefixes or suffixes of each other, or for which one backbone is contained in the other.

Problem 3. Parsimonious Overlapping Reticulate Network from Gene Trees Problem: *Given a set of X-trees H. Construct a reticulate network N containing only independent or overlapping reticulations that supports H and contains a minimal number of reticulations, if one exists.*

One of our main results is that this problem is computationally tractable, if we limit the maximum number of reticulations contained in any given tangle, and we present an algorithm to solve it in Section 6.

5 The Decomposition Theorem

Suppose that $N = (V, E, \nu)$ is a reticulate network. We define $\Sigma(N) := \bigcup_{T \in \mathcal{T}(N)} \Sigma(T)$, and for any edge $e \in E$, we take $\Sigma(e) := \{\sigma_T(e) \mid e \text{ is edge of } T \in \mathcal{T}(N)\} \subseteq \Sigma(N)$ to be the set of all splits generated by e in trees induced by N.

There is a close relationship between reticulate networks and splits graphs. More precisely, there exists a one-to-one relationship between the tangles of a reticulate network N, the connected components of the incompatibility graph $IG(\Sigma(N))$ and the netted components of the splits graph $SG(\Sigma(N))$. This is implied by the following result:

Theorem 1. *[Decomposition Theorem] Suppose N is a reticulate network. Two tree edges $e, f \in E_T$ are contained in a cycle in N, if and only if there exist two splits $S \in \Sigma(e)$ and $S' \in \Sigma(f)$ that are contained in the same connected component of the incompatibility graph $IG(\Sigma(N))$.*

Similar results, using different definitions, interpretations and proofs, are reported in [11, 14][1].

To formulate our proof, we first introduce some additional definitions and results. Suppose that N is a reticulate network and C is a path or cycle in N. We say that C *fully contains* a reticulation $v \in V_R$, if C contains both $p(v)$ and $q(v)$ (consecutively, of course), and we use $s(C)$ to denote the number of reticulations fully contained in C.

Lemma 2. *For every cycle C in a reticulate network N we have $s(C) \geq 1$.*

Proof: Direct all edges away from the root of N. The edges of C cannot all be oriented in the same direction and thus there must exist two consecutive edges e_1 and e_2 in C that are oriented toward their common vertex v. By definition of N, we must have $v \in V_R$ and $\{p(v), q(v)\} = \{e_1, e_2\}$, and thus $s(C) \geq 1$. □

Lemma 3. *Consider a reticulate network N. If $e, f \in E_T$ are two different tree edges contained in a common cycle C with $s(C) = 1$, then there exist two splits $S \in \Sigma(e)$ and $S' \in \Sigma(f)$ that are incompatible.*

Proof: Consider two edges $e, f \in E_T$ and assume they are both part of a cycle C that contains precisely one reticulation $v \in V_R$ for which both $p(v)$ and $q(v)$ are edges in C. Then there exist two trees $T_p \in \mathcal{T}(N)$ and $T_q \in \mathcal{T}(N)$ that contain all edges of C except for $q(v)$ and $p(v)$, respectively, and differ only by these two edges. Assume that $C = (e_0 = p(v), w_1 = v, e_1 = q(v), w_2, \ldots, w_\alpha, e_\alpha = e, w_{\alpha+1}, \ldots, w_\beta, e_\beta = f, w_{\beta+1}, \ldots, w_k)$ for appropriate α, β. For any node u in C, let V_u denote the set of all nodes that can be reached in T_p or T_q from a

[1] In [14], the input to the problem is a set M of binary sequences of length n that are assumed to have been generated under the infinite sites model. If we discard all constant sites, then the set of remaining columns of M is equivalent (up to a choice of ancestral states) to a set of splits Σ and the definition of an *incompatibility graph* in [14] is equivalent to our definition. A *phylogenetic network*, as defined in [14] (and perhaps better termed a *recombination network*), is based on a directed acyclic graph with a specified root, and certain coalescence and recombination nodes. Such a network N is considered to explain an input set M, if there exists a labeling of the leaves by M and a labeling of the internal nodes by additional sequences of length n, together with a labeling of the edges by columns of M (that is, splits) and a labeling of the recombination nodes by certain recombination events, such that each split occurs precisely once and the implied mutations and recombinations give rise to the specified labeling of the nodes by sequences. Because there is some choice in the placement of individual splits within such a network, the Decomposition Theorem in [14] holds only in one direction, namely *if two splits are contained in the same connected component of the incompatibility graph, then there exists a phylogenetic network such that the corresponding edges are contained in a cycle*, but not vice-versa. In our definition of a reticulate network, we do not explicitly label edges by splits. Rather, each edge implicitly corresponds to a set of splits that is defined via the set of trees that can be sampled from the network. This lack of choice explains why our version of the Decomposition Theoren holds in *both* directions.

node u without using any edge in C. Note that all four sets V_v, V_{w_α}, V_{w_β} and $V_{w_{\beta+1}}$ must be disjoint, as both T_p and T_q are cycle-free. Let X_u denote the set of taxa that occur as labels in V_u. The split $S = \sigma_{T_p}(e)$ induced by e in T_p separates $X_v \cup X_{w_\alpha}$ from $X_{w_\beta} \cup X_{w_{\beta+1}}$. Similarly, the split $S' = \sigma_{T_q}(f)$ induced by f in T_q separates $X_{w_\alpha} \cup X_{w_\beta}$ from $X_{w_\beta+1} \cup X_v$. Thus, S and S' are incompatible. □

Now we prove Theorem 1:

Proof: "⇒": Suppose we are given a reticulate network N and two different tree edges $e, f \in E_T$ that are contained in a cycle C. Orient all edges away from the root and use g^- and g^+ to indicate the implied start and end of an edge $g \in E$. We have two cases, which we both prove by induction:

Case 1: The edges e and f have opposite orientations in C. By Lemma 2 we have $s(C) > 0$. If $s(C) = 1$, then Lemma 3 implies the result. So assume that $s(C) = n > 1$ and $C = (e^-, e, e^+, \ldots, p(v_1), v_1, q(v_1), w, \ldots, f^+, f, f^-, \ldots)$, where v_1 is the first encountered reticulation that is fully contained in C. By properties (R4–R5) of N, for any node u there exists a path P_u from the root ρ to u with $s(P_u) = 0$. Let $-P_u$ denote the reversal of P_u. Construct a cycle C' by concatenating P_{e^+}, the section of C that links e^+ to w, and $-P_{w_k}$. It has $s(C') = 1$. Construct a second cycle C'' by concatenating P_{w_k}, the section of C that links w to f^+, and $-P_{f^+}$. It has $s(C'') < n$. Let f' be the edge contained in P_w that is adjacent to w. This must be a tree edge, because $w_k \in V_T$. By Lemma 3, there exist two incompatible splits $S \in \Sigma(e)$ and $S' \in \Sigma(f')$. By induction, there exists a chain of pairwise incompatible splits from S' to some split $S'' \in \Sigma(f)$. Hence, the claim follows.

Case 2: This is dealt with in the same way, but using slightly different paths.
"⇐": If e and f are two tree edges not contained in a cycle, then there exists a cut-edge h (or at least a cut-vertex, which can be refined to provide a cut-edge h) that separates e and f. The edge h induces the same split $S = \frac{A}{B}$ in every tree $T \in \mathcal{T}(N)$. Thus, every split $S' \in \Sigma(N)$ subdivides either A or B, but not both sets. This implies the claim. □

6 Algorithms

Suppose we are given a set of gene trees H sampled from a reticulate network N and would like to solve Problem 3. By Theorem 1 and Lemma 1, we can assume that N consists of precisely one tangle. We can reduce the problem size further by assuming that X is *Σ-separated*, that is, that for every pair of distinct taxa $x, y \in X$ there exists a split $S = \frac{A}{B} \in \Sigma(N)$ with $|A \cap \{x, y\}| = 1$.

Lemma 4. *To solve the posed computational problems, it suffices to consider the* reduced case *of a collection of X-trees H such that X is Σ-separated and the incompatibility graph $IG(\Sigma)$ consists of precisely one connected component, or, equivalently, the splits graph $SG(\Sigma)$ consists of exactly one netted component.*

In the following we restrict our attention to the reduced case by virtue of Lemma 4. Given a taxon set X, we define a *reticulation scenario* (R, B, I) to consist of a set of *reticulate taxa* $R \subset X$ and an ordered list of *backbone taxa* $B = (b_1, b_2, \ldots, b_k)$, so that X equals the disjoint union $R \cup \{b_1, \ldots, b_k\}$, together with a mapping I from R to distinct intervals of backbone taxa in $(b_2, \ldots, b_{k-1})$. We require that the outgroup taxon $o \in X$ is contained in B, if specified. Moreover, we set

$$\Sigma(R,B,I) = \left\{ \tfrac{R^-(i) \cup \{b_1,\ldots,b_i\} \cup R_1(i)}{R_2(i) \cup \{b_{i+1},\ldots,b_k\} \cup R^+(i)} \mid i = 1,\ldots,k-1 \right\} \cup \left\{ \tfrac{R^-(i) \cup \{b_1,\ldots,b_i\} \cup R_2(i)}{R_1(i) \cup \{b_{i+1},\ldots,b_k\} \cup R^+(i)} \mid i = 2,\ldots,k \right\},$$

with

$$R_1(i) \cup R_2(i) \text{ is any partitioning of } R(i) := \{r \in R \mid b_i \in I(r)\} \neq \emptyset,$$
$$R^-(i) := \{r \in R \mid I(r) \subseteq \{b_1, \ldots, b_{i-1}\}\},$$
$$\text{and } R^+(i) := \{r \in R \mid I(r) \subseteq \{b_{i+1}, \ldots, b_k\}\}.$$

We interpret $B = (b_1, \ldots, b_k)$ as the joint "super-backbone" along which all overlapping reticulations are arranged. Every taxon $r \in R$ corresponds to a reticulation node and the associated interval $I(r) \subseteq B$ defines precisely which part of B will form the backbone of the tree cycle associated with r.

Given a reticulation scenario (R, B, I), we can construct a corresponding reticulate network $N(R, B, I)$ in polynomial time, as follows:

Algorithm 1. *Assign a reticulate node $v(r)$ to each reticulate taxon $r \in R$ and a tree node $v(b)$ to each backbone taxon $b \in B$. Additionally, for each consecutive pair of taxa b_i, b_{i+1} in B, define a tree node $v(b_i, b_{i+1})$, and then connect $v(b_i)$ to $v(b_i, b_{i+1})$, and $v(b_i, b_{i+1})$ to $v(b_{i+1})$, using tree edges ($i = 1, \ldots, k-1$). Finally, connect each reticulate node $v(r)$ to the two nodes $v(b_i, b_{i+1})$ and $v(b_j, b_{j+1})$ using reticulate edges, where i, j are chosen such that $I(r) = \{b_{i+1}, b_{i+2}, \ldots, b_{j-1}\}$ holds.*

By construction we have:

Lemma 5. *A reticulation scenario (R, B, I) corresponds to a solution $N(R, B, I)$ of Problem 3, if and only if H can be sampled from $N(R, B, I)$ and $|R|$ is minimal.*

The following useful observation follows directly from the definition of $\Sigma(R, B, I)$:

Lemma 6. *Let Σ denote the set of splits in the reduced case. If (R, B, I) corresponds to a solution of Problem 3, then* every *split $S \in \Sigma$ must separate b_1 and b_k, that is, we must have $|\{b_1, b_k\} \cap A| = 1$ for all $S = \frac{A}{A'} \in \Sigma$.*

The following polynomial-time algorithm takes as input a set of trees H in the reduced case and produces as output a minimal reticulation scenario, if it exists:

Algorithm 2. *Consider all possible subsets R of X of size $\leq k$ in order of increasing cardinality, and set $B = X \setminus R$. There are $O(|X|^k)$ such subsets. Define $\Sigma|_B := \left\{ \frac{A \cap B}{A' \cap B} \mid \frac{A}{A'} \in \Sigma \right\} \setminus \left\{ \frac{\emptyset}{B}, \frac{B}{\emptyset} \right\}$. To obtain an ordering of B, first determine b_1 using Lemma 6. Let $\Sigma|_B(b_1) = \{A \in S \in \Sigma|_B : b_1 \in A\}$ denote the set of split parts in $\Sigma|_B$ containing b_1. Then, these must be ordered by inclusion $\{b_1\}, \{b_1, b_2\}, \{b_1, b_2, b_3\}, \ldots$ and we thus obtain an ordering $b_1, b_2, \ldots$ of B. To compute the set $I(r) \subseteq B$ for a given reticulate taxon $r \in R$, determine the set of all $b \in B$ with the property that $\frac{A \cup \{b,r\}}{A'} \in \Sigma|_{B \cup \{r\}} \Leftrightarrow \frac{A \cup \{b\}}{A' \cup \{r\}} \in \Sigma|_{B \cup \{r\}}$. Finally, check whether $\Sigma(H) \subseteq \Sigma(R, B, I)$ holds.*

From this algorithm one can derive the following consequence:

Theorem 2. *If the number of tangled reticulations is limited to k, then Problem 3 is solvable in polynomial time. In particular, Problem 2 is solvable in polynomial time.*

A visualization of a reticulate network N can be obtained using the following algorithm:

Algorithm 3. *Given a set of splits Σ, compute the splits graph $IG(\Sigma)$ using [13]. For each component Z of the incompatibility graph $IG(\Sigma)$, use Algorithm 2 to compute a reticulation scenario for Z. Then, use Algorithm 1 to compute the topology of the reticulate network $N(Z)$ associated with Z. Let Z' denote the netted component in $SG(\Sigma)$ associated with Z. Replace Z' by $N(Z)$ by first removing all non-gate nodes and any edge contained in Z and then mapping the appropriate nodes of $N(Z)$ onto the gate nodes of Z'.*

Although the computation of $SG(\Sigma)$ can take exponential time for an arbitrary split set Σ, for our graph-layout purposes it suffices to compute the splits graph for a subset of splits chosen in such a way that the associated splits graph contains no cliques of size greater than 4, say, which can be done in polynomial time (unpublished result).

7 Implementation and Application

We have implemented the algorithms described in Section 6, and our implementation is freely available as a plug-in for the program SplitsTree [15]. We thus provide a robust and flexible tool for biologists to explore real datasets for evidence of reticulate evolution. Given a set of taxa X, the input can be either a set of X-trees, a set of *partial* X-trees (that is, a set of X'-trees for different subsets of taxa $X' \subseteq X$) or a set of X-splits. In the context of investigating gene trees, it is seldom the case that all trees are full X-trees and so the capability to process partial trees is of particular importance. We achieve this in practice by first applying the Z-closure method [22] to a given set of partial trees to obtain a set of full X-splits.

An important aspect of our implementation is that we provide a visualization of the computed reticulate network. It is based on an algorithm for constructing splits graphs [13] and provides a useful visualization of the complete input

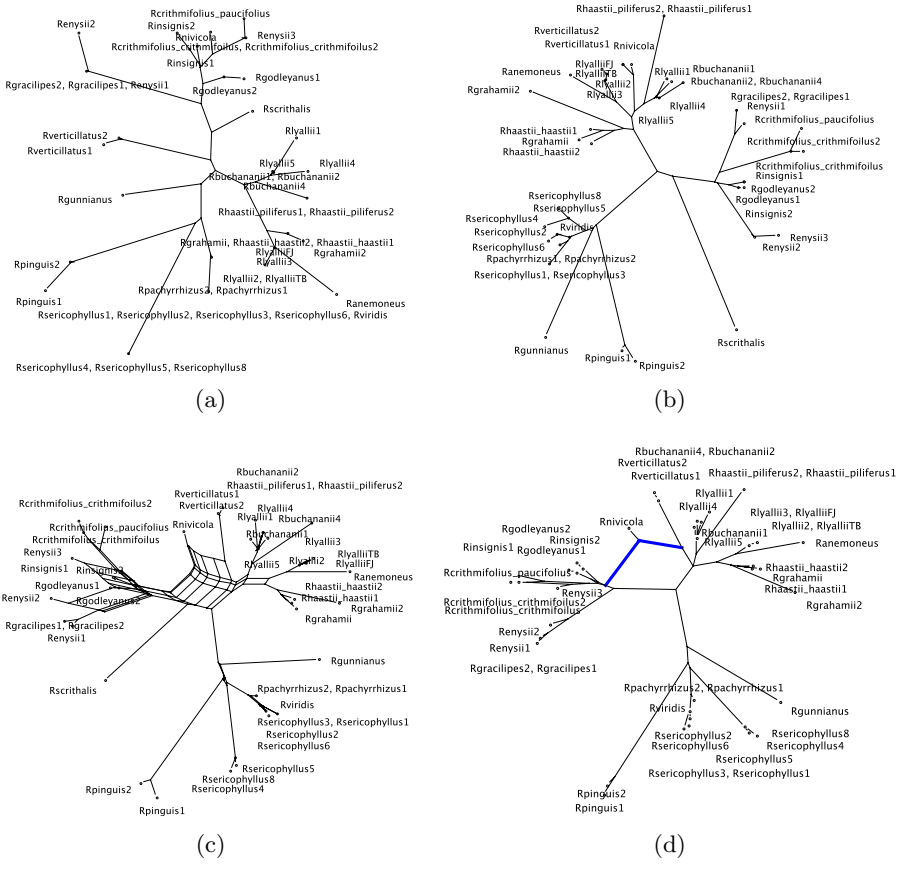

Fig. 3. In (a) and (b) we show two gene trees for 46 buttercups, based on the chloroplast J_{SA} region and nuclear ITS gene [16]. In (c) we show the splits graph of all splits of both trees. Removal of one taxon (*R.scrithalis*) and five interfering splits leads to a configuration that is recognized by our algorithm as a reticulation that gives rise to *R.nivicola*, as shown in (d)

data. Netted regions of the graph that can be explained by a set of overlapping reticulations are drawn as such and the others remain netted, see Figure 2(c–d).

In Figure 3(a–b) we depict two different gene trees for New Zealand alpine *Ranunculus* (buttercup) species based on the nuclear ITS gene and the chloroplast J_{SA} region [16]. The splits graph in Figure 3(c) suggests that *R.nivicola* may be a hybrid between the evolutionary lineages on the left- and right-hand side of the splits graph. However, our algorithm fails to explain the netted region by a collection of overlapping reticulations. This failure is due to additional incompatible splits in both the left- and right-hand side of the graph that extend into the netted component containing *R.nivicola*. Additionally, the placement of *R.scrithalis* is problematic. Interactive deletion of one split on the right-hand side

and four splits on the left-hand side, and removal of *R.scrithalis*, leads to a simplified netted component that had a detectable reticulation that may correspond to a hybridization event, Figure 3(d). For this specific case, studies of morphological variation and chromosome numbers had earlier suggested that *R.nivicola* was an allopolyploid (hybrid) formed between *R.insignis* and *R.verticillatus* [16].

8 A Statistical Test for Reticulate Evolution

The mere incompatibility of gene trees does not necessarily constitute evidence for reticulate evolution, as gene phylogenies may conflict by (at least) three other processes: firstly, the gene trees may not be historically accurate due to (i) model misspecification or inappropriate methodology, or due to (ii) sampling effects (insufficient sites to compensate for site saturation or short interior edges); alternatively the gene phylogenies may be historically correct but differ from the species tree due to (iii) the population-genetic effect known as *lineage sorting* [20, 23, 24]. One scenario that could easily be mistaken for reticulation under process (i) is the following. Suppose the evolutionary history of the taxa is accurately described by a tree T (i.e. no reticulate evolution occurred) and this tree describes the history of gene 1 and gene 2 (so there is no lineage sorting effect for these two genes). Suppose further that two taxa x and y that are widely separated in T have independently acquired a strong compositional bias (such as increased GC richness). Then most tree reconstruction methods will reconstruct a phylogeny for gene 2 that is different to T (grouping taxa x and y as sister taxa) but which together with the (correct) tree for gene 1 would be explained by a single reticulation event - namely that taxon x (or y) was a hybrid. In this case one can test the null hypothesis that the two trees differ simply due to compositional variation, against the alternative of genuine reticulation, by adapting the statistical test described by [25].

Lineage sorting (process (iii)) can also be distinguished from reticulation, either by a parametric approach based on divergence time estimates [24], or by a non-parametric approach when the number of gene trees is large. This non-parametric procedure can also allow sampling effects (process (ii)) as well as, or in place of, lineage sorting - provided the substitution process follows a molecular clock. To illustrate this approach – for a sequence of gene trees $(g_1, g_2, \ldots, g_k)$– suppose we have just three taxa a, b, c and a hypothesized rooted species tree $(ab)c$. If n_1 of these gene trees support $(ac)b$ and n_2 of them support $(bc)a$, let $m = n_1 + n_2$, and consider the test statistic

$$\Delta := |n_1 - n_2|.$$

We describe how one can use Δ to test the null hypothesis H_o that $(ab)c$ is the species tree and that the underlying process that resulted in the m other trees is independent occurrences of lineage sorting (or sampling effects subject to a molecular clock) against the alternative hypothesis H_1 that there has also been a reticulation event involving the transfer of some genes from one lineages in the past across to another. Let $I := \{i : g_i \text{ supports } (ac)b \text{ or } (bc)a\}$ so we can

write $\Delta = |\sum_{i \in I} \delta_i|$ where $\delta_i = 1$ (resp. -1) if gene g_i supports tree $(ac)b$ (resp. tree $(bc)a$). We will regard the δ_i values as realizations of a sequence of m independent outcomes $D_1, \ldots, D_m$ that take values in $\{1, -1\}$ (i.e. lineage sorting or sampling effects that change the tree structure are assume to be independent from gene to gene). Then under H_o, $\mathbb{E}[D_i] = 0$ since when a lineage sorting event occurs that results in a gene tree at variance with the species tree, then that gene tree is equally likely to be either of two alternative trees (this follows from population-genetic considerations [26, 23]), while for sampling effects subject to a molecular clock $\mathbb{E}[D_i] = 0$ also holds by symmetry. In contrast, under H_1 we would expect $\mathbb{E}[D_i]$ to be systematically negative or positive depending (respectively) on whether $(ac)b$ or $(bc)a$ is the other dominant gene tree involved in the reticulation. Furthermore, $|\Delta - \Delta'| \leq 2$ if Δ and Δ' differ on just value of $i \in I$. Thus, regarding Δ as a function of m independent random variables $(D_1, \ldots, D_m)$ we can apply the Azuma-Hoeffding inequality [27] to deduce that under H_o,

$$\mathbb{P}[\Delta \geq a] \leq 2 \exp(-a_2/8m).$$

This allows us to use Δ as a test statistic to test H_o against H_1. As an example, suppose we find that $n_1 = 60, n_2 = 10$. Then $\Delta(g) = 60 - 10 = 50$, and $m = 70$, so $\mathbb{P}[\Delta \geq 50] \leq 2 \exp(-502/8 \cdot 70) = 0.023$, and consequently we could reject H_o at the 5% significance level.

References

1. Holland, B., Huber, K., Moulton, V., Lockhart, P.J.: Using consensus networks to visualize contradictory evidence for species phylogeny. Molecular Biology and Evolution **21** (2004) 1459–1461
2. Rieseberg, L.H., Raymond, O., Rosenthal, D.M., Lai, Z., Livingstone, K., Nakazato, T., Durphy, J.L., Schwarzbach, A.E., Donovan, L.A., Lexer, C.: Major ecological transitions in annual sunflowers facilitated by hybridization. Science **301** (2003) 1211–1216
3. Hein, J.: Reconstructing evolution of sequences subject to recombination using parsimony. Math. Biosci. (1990) 185–200
4. Hein, J.: A heuristic method to reconstruct the history of sequences subject to recombination. J. Mol. Evol. **36** (1993) 396–405
5. Wang, L., Zhang, K., Zhang, L.: Perfect phylogenetic networks with recombination. Journal of Computational Biology **8** (2001) 69–78
6. Gusfield, D., Eddhu, S., Langley, C.: Efficient reconstruction of phylgenetic networks with constrained recombination. In: Proceedings of the 2003 IEEE CSB Bioinformatics Conference. (2003)
7. D. Gusfield, S.E., Langley, C.: The fine structure of galls in phylogenetic networks. to appear in: INFORMS J. of Computing Special Issue on Computational Biology (2004)
8. Nakhleh, L., Warnow, T., Linder, C.R.: Reconstructing reticulate evolution in species - theory and practice. RECOMB'04 (2004) 337–346
9. Hudson, R.R., Kaplan, N.L.: Statistical properties of the number of recombination events in the history of a sample of DNA sequences. Genetics **111** (1985) 147–164

10. Myers, S.R., Griffiths, R.C.: Bounds on the minimal number of recombination events in a sample history. Genetics **163** (2003) 375–394
11. Bafna, V., Bansal, V.: The number of recombination events in a sample history: conflict graph and lower bounds. IEEE/ACM Transactions in Computational Biology and Bioinformatics **1** (2004) 78–90
12. Bandelt, H.J., Dress, A.W.M.: A canonical decomposition theory for metrics on a finite set. Advances in Mathematics **92** (1992) 47–105
13. Dress, A.W.M., Huson, D.H.: Constructing splits graphs. IEEE/ACM Transactions in Computational Biology and Bioinformatics **1** (2004) 109–115
14. Gusfield, D., Bansal, V.: A fundamental decomposition theory for phylogenetic networks and incompatible characters. To appear in: Proceedings of RECOMB'2005 (2004)
15. Huson, D.H., Bryant, D.: Estimating phylogenetic trees and networks using SplitsTree 4. Manuscript in preparation, software available from `www-ab.informatik.uni-tuebingen.de/software` (2004)
16. Lockhart, P.J., McLenachan, P.A., Havell, D., Glenny, D., Huson, D.H., Jensen, U.: Phylogeny, dispersal and radiation of New Zealand alpine buttercups: molecular evidence under split decomposition. Ann Missouri Bot Gard **88** (2001) 458–477
17. Kreitman, M.: Nucleotide polymorphism at the alcohol dehydrogenase locus of Drosophila melanogaster. Genetics **11** (1985) 147–164
18. Semple, C., Steel, M.A.: Phylogenetics. Oxford University Press (2003)
19. Jukes, T.H., Cantor, C.R.: Evolution of protein molecules. In Munro, H.N., ed.: Mammalian Protein Metabolism. Academic Press (1969) 21–132
20. Maddison, W.P.: Gene trees in species trees. Syst. Biol. **46** (1997) 523–536
21. Baroni, M., Semple, C., Steel, M.A.: A framework for representing reticulate evolution. Annals of Combinatorics (In press)
22. Huson, D.H., Dezulian, T., Kloepper, T., Steel, M.A.: Phylogenetic super-networks from partial trees. IEEE/ACM Transactions in Computational Biology and Bioinformatics, in press (2004)
23. Rosenberg, N.A.: The probability of topological concordance of gene trees and species trees. Theor. Pop. Biol. **61** (2002) 225–247
24. Sang, T., Zhong, Y.: Testing hybrization hypotheses based on incongruent gene trees. System. Biol. **49** (2000) 422–424
25. Steel, M.A., Lockhart, P., Penny, D.: Confidence in evolutionary trees from biological sequence data. Nature **364** (1993) 440–442
26. Tajima, F.: Evolutionary relationships of DNA sequences in finite populations. Genetics **105** (1983) 437–460
27. Alon, N., Spencer, J.H.: The Probabilistic Method. 2nd edn. John Wiley (2000)

Appendix

The examples shown in this paper are based on gene trees, and Figures 2–3 were generated using our software. As stated in Section 7, our implementation can also process other types of input, for example binary sequences representing haplotype data. We illustrate this using the data presented in [11], which was reportedly taken from the alcohol dehydrogenase locus from 11 chromosomes of *Drosophila melanogaster* [17]. This data consists of a reduced set of 9 haplotypes typed at 16 sites:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
b	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
c	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
d	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
e	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	1
f	0	1	0	0	0	1	0	0	0	1	0	1	0	1	1	1
g	0	1	0	0	0	1	0	0	1	1	1	1	1	1	0	1
h	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	1
i	1	1	1	1	0	1	0	0	1	1	1	1	1	1	0	1

Each column of this matrix defines a split of the taxon set $X = \{a, b, \ldots, i\}$ and let Σ denote the set of all such splits. This data can be entered directly into the SplitsTree program. The resulting splits graph $SG(\Sigma)$ is shown in Figure 4(a) (with all trivial splits added for clarity). This figure indicates that the configuration of splits is quite complex and, as a consequence, our algorithms fail to detect a reticulate network for this data.

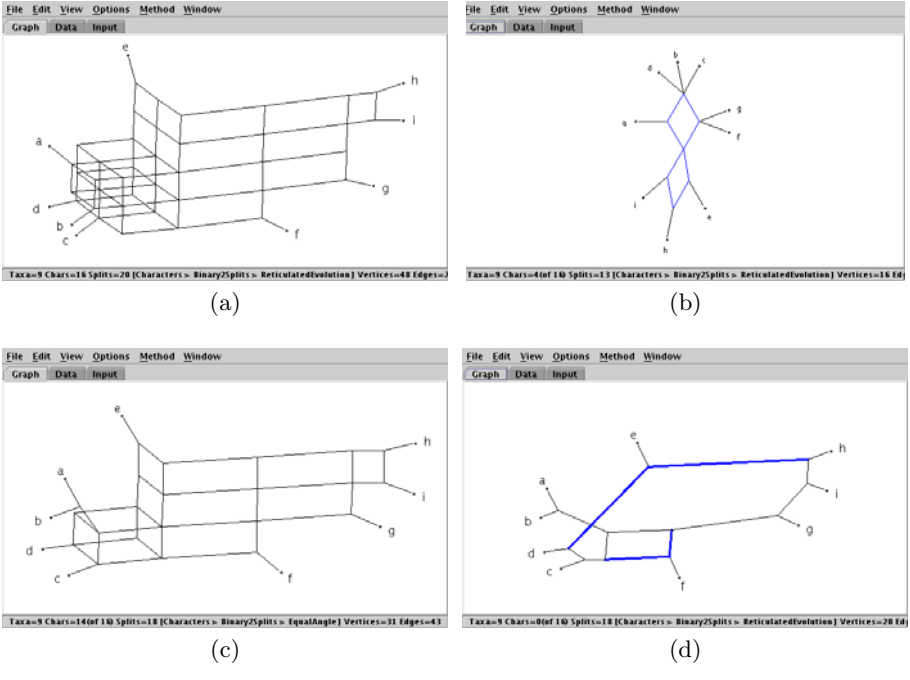

Fig. 4. In (a) we show the splits graph associated with the full sixteen columns of haplotype data taken from [11]. In (b), we show the splits graph for the four columns $\{1, 4, 5, 6\}$. It consists of two cycles with one reticulation per cycle. In (c), we show the splits graph for 14 columns of the data, with columns 2 and 4 removed. In (d), we show the reticulate network computed from this reduced data set, involving two reticulations, with reticulation edges highlighted by heavy lines

Our implementation allows one to easily add or remove sites from the analysis. In Figure 4(b), we show the splits graph for the subset of sites $\{1, 4, 5, 6\}$. It contains two netted components. In this case, the graph topology alone does not determine which nodes are to be interpreted as reticulation nodes. Declaring one of the taxa to be an outgroup will reduce the number of possible choices of reticulate nodes, but even then there will still be more than one choice.

Let Σ' denote the 14 splits that remain after removing sites 2 and 4. We show the resulting splits graph $SG(\Sigma')$ in Figure 4(c). Application of Algorithm 3 to this reduced set of splits Σ' results in the detection of a solution involving precisely two overlapping reticulations, as shown in Figure 4(d).

Inspection of the sequences reveals that the reticulation displayed at sequence e can be interpreted as a recombination, as e can be obtained from sequences d and h via combination of the first seven (or five, not counting sites 2 and 4) positions of h and the last nine positions of d, with only one mutation in either area. However, it is not possible to obtain sequence f from c and g via a single cross-over and a small number of mutations.