Contents

1 Trees First **3**
1.1 Rooted Perfect-Phylogeny 3
1.1.1 Alternative Definitions 5
1.1.2 The Perfect-Phylogeny Problem and Solution 7
1.2 Alternate, Known Roots 14
1.3 Root-Unknown Pefect-Phylogeny 16
1.3.1 Uniqueness 19
1.4 Splits Equivalence 21
1.5 Advanced Perfect-Phylogeny 25
1.5.1 The Dress-Steel solution to the 3-state Perfect Phylogeny Problem 27
1.5.2 Generalizations of the Four-Gamete and Splits-Equivalence theorems 32
*

Chapter 1

Trees First

Our main interest is in genealogical and phylogenetic *networks*, which by definition are *not* trees. However, many of the network models derive from tree models, and many of the tools that address networks rely critically on tools for trees. Therefore we must first understand some models of tree-like evolution and some combinatorial and algorithmic results about evolutionary trees. The main tree-based model of evolution that we use is called the (rooted, binary-character) *Perfect-Phylogeny* model.

1.1 The (rooted, binary-character) Perfect-Phylogeny Problem

Definition Let M be an n by m matrix representing n taxa in terms of m characters or traits that describe the taxa. Each character takes on one of two possible *states*, 0 or 1, and a cell (f, c) of M has a value of one if and only if the state of character c is 1 for taxon f. Thus the characters are *binary-characters* and M is called a *binary matrix*.

When a taxon f has state 1 for a binary character c, we also say that "f possesses (or contains or has) character c".

Definition Given an n by m binary-character matrix M for n taxa, a *perfect-phylogeny for M* is a *rooted* (directed) tree T with exactly n leaves that obeys the following properties:

1. Each of the n taxa labels exactly one leaf of T.
2. Each of the m characters labels *exactly one* edge of T.
3. For any taxon f, the characters that label the edges along the unique path from the root to leaf labeled f specify all of the characters that taxon f possesses (i.e., whose state is one).

	c_1	c_2	c_3	c_4	c_5
r_1	1	1	0	0	0
r_2	0	0	1	0	0
r_3	1	1	0	0	1
r_4	0	0	1	1	0
r_5	0	1	0	0	0

Table 1.1: Matrix M has a perfect-phylogeny T shown in Figure 1.1.

When needed, we will also assume if a leaf of T is labeled by a taxon f, then it is labeled by the characters that f possesses, or equivalently, is labeled by the binary sequence defined by row f of M.

A perfect-phylogeny exists for some M, but not for all M. The input M shown in Table 1.1 does have a perfect-phylogeny, shown in Figure 1.1.

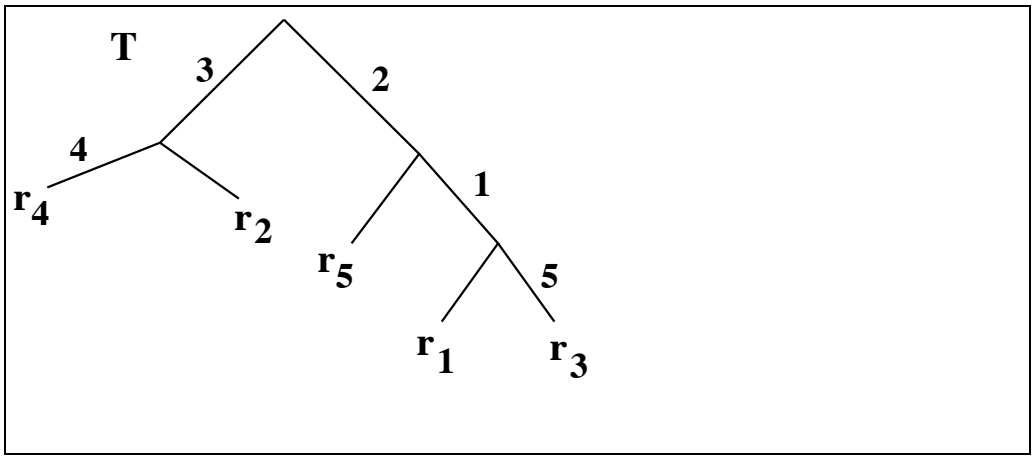


Figure 1.1: Perfect-Phylogeny T for matrix M shown in Table 1.1.

In the definition given for a perfect-phylogeny T, it is required that T be rooted and directed, and that the characters be binary. This is the default case, the technically simplest case and most common case that will be discussed in this book. However, we will sometimes relax these assumptions; when we do, we will always explicitly state the alternative assumptions being used. So, when we use "perfect-phylogeny" with no additional modifiers, we are referring to a rooted, binary-character perfect-phylogeny.

The interpretation of a perfect-phylogeny T for M is that it gives an estimate of the rooted evolutionary history of the taxa (in terms of branching pattern, but not time), based on the following biological and technical assumptions:

1. The taxa in M are generally taxa whose states have been observed and

are known.

2. There exists a taxon r (possibly unknown) which is *ancestral* to all the taxa in M. The character-state sequence for taxon r is called the *ancestral sequence*. This sequence is denoted S_r. For technical and expositional convenience, we assume first that the state of r for each character is zero, so S_r is the all-zero sequence. We will see later how to relax that assumption.

3. In the evolutionary history of the taxa, each of the characters mutates from the zero state to the one state *exactly* once, and never from the one state to the zero state. Hence every character c labels exactly one edge e in a perfect-phylogeny T for M, indicating the unique point in the evolutionary history of the taxa when character c mutates. It follows that any taxon that labels a leaf below e (or in more graph-theoretic terminology, in the "subtree" below e) must possess character c.

The key biological and combinatorial feature of the perfect-phylogeny model is that each character mutates *exactly once* in the evolutionary history of the taxa. This is principally motivated by the *infinite sites* model from population genetics and widely collected SNP data (discussed in Section **??**). The infinite sites assumption is not always valid but is appropriate in the biological settings that are the major focus of this book. The assumption of only one mutation per character is also motivated by the biological basis of *complex characters* as discussed in Section **??**. In molecular sequences, a change at a single site (i.e., of a single character) is called a *mutation*.

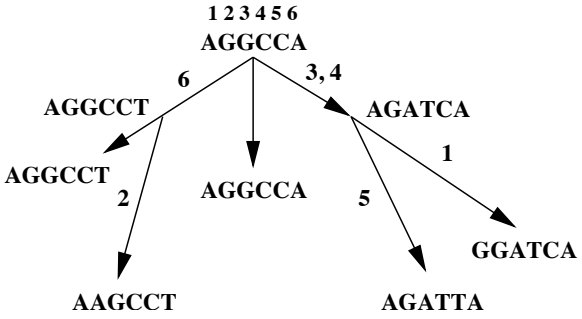
Figure 1.2 shows the phylogeny of six SNP sites published in the Journal of Human Genetics [24]. The SNP data fits the infinite sites model and the phylogeny is a perfect-phylogeny.

1.1.1 Alternative definitions a perfect-phylogeny

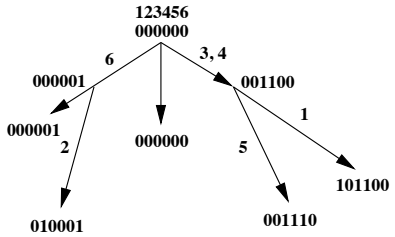
There are two alternative, but equivalent, definitions of a perfect-phylogeny that are often technically helpful, and that generalize nicely to *non-binary* and to *undirected* or *unrooted* perfect-phylogeny problems, which we will discuss later. First we need the following

Definition A node in a rooted tree T is called an *internal node* if it is neither a leaf node nor the root of T.

A Second Definition of a Perfect-Phylogeny A perfect-phylogeny for M is a rooted tree T with n leaves, where each leaf is labeled by a distinct taxon of M, and where the root of T and each internal node of T, is labeled by an m-length binary sequence specifying a state for each of the m characters, such that:



(a) Perfect-phylogeny of five sequences at six SNP sites.



(b) The sequences recoded into binary.

Figure 1.2: The perfect-phylogeny developed in [24] for five sequences at six SNP sites in the Human DTNBP1 gene (Dysbindin) on Chromosome six. It has been suggested that variations in these sequences are associated with schizophrenia, although the evidence is said to be contradictory. There are two DNA variants at each SNP site (for example A and G at site 1, and C, T at site 5), and so the SNPs can be recoded into binary, using 0 for the ancestral state and 1 for the derived state. The six sites labelled 1 through 6 here are actually SNPs 2,3,5,6,8,11 in [24].

For every character c and each state i (either 0 or 1) of c, the nodes labeled with state i for character c form a *connected subtree* of T, denoted $T_c(i)$.

This property is called the *convexity requirement*. Clearly, for any character c and states $i \neq j$, the subtrees $T_c(i)$ and $T_c(j)$ of perfect-phylogeny T are node disjoint. For example, the perfect-phylogeny from Figure 1.1, with its node labels, is shown in Figure 1.3.

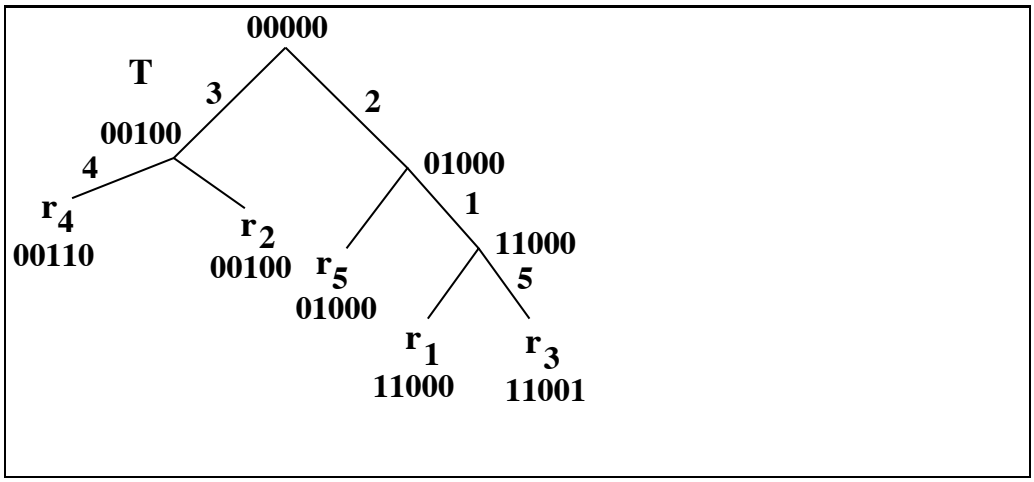


Figure 1.3: Perfect-Phylogeny T with nodes labeled by the derived sequences.

Note that in this second definition of a perfect-phylogeny there is no mention of mutations, or labels on edges, or the requirement that each character mutates only once. However, it is easy to see that the two definitions of a perfect-phylogeny are equivalent. We leave the proof to the reader.

There is a third related, and equivalent, view of a perfect-phylogeny that is also often used.

A Third Definition of a Perfect-Phylogeny Given M, let T be a rooted tree with n leaves, where each leaf is labeled by a distinct taxon of M, where the non-leaf nodes of T are *unlabeled*. For a state i of character c, let $T_c(i)$ now denote the *smallest* subtree of T connecting all the leaves of T that are labeled with state i for character c. Then T is called a perfect-phylogeny if and only if for each c, the subtrees $T_c(0)$ and $T_c(1)$ are node disjoint.

We again leave it to the reader to convince themselves that this definition is equivalent to the two prior definitions.

1.1.2 The Perfect-Phylogeny Problem and Solution

The Perfect-Phylogeny Problem: Given an n by m, binary matrix M, determine whether there is a perfect-phylogeny for M, and if so, build one.

original column	2	1	3	5	4
	c_1	c_2	c_3	c_4	c_5
r_1	1	1	0	0	0
r_2	0	0	1	0	0
r_3	1	1	0	1	0
r_4	0	0	1	0	1
r_5	1	0	0	0	0

Table 1.2: Matrix $\overline{M}$ resulting from sorting the columns of the matrix M shown in Table 1.1. The first row of numbers above $\overline{M}$ indicates the original column of each character in M. The second row of numbers gives the new name for each character.

We will solve the perfect-phylogeny problem with a simple $O(nm)$-time algorithm where each comparison operation and each reference to M takes one time unit.

For the algorithm and its proof of correctness, it will be helpful to first sort the columns of M by the *number* of ones they contain, largest first, breaking ties arbitrarily. Let $\overline{M}$ denote the sorted matrix M. For an example, see Table 1.1.2. Certainly, M has a perfect-phylogeny if and only if $\overline{M}$ does, and the perfect-phylogeny for $\overline{M}$ differs from the perfect-phylogeny for M only by a change in edge labels corresponding to the sorting of the columns of M. For example, see Figure 1.4.

From this point on, the name of each character will be the same as the column that it occupies in $\overline{M}$, rather than in M. For example the character at the left of $\overline{M}$ will be called character one. Therefore, for two characters c and d, with $c < d$, character c must be to the left of character d in $\overline{M}$. Similarly, if character c is to the left of character d in $\overline{M}$, then it must be that $c \leq d$.

Theorem 1.1.1 The Perfect-Phylogeny Theorem *Matrix $\overline{M}$ (or M) has a perfect-phylogeny (with all-zero ancestral sequence) if and only if no pair of columns c, d contains the three binary pairs 0,1; 1,0; and 1,1.*

Proof First, suppose that T is a perfect-phylogeny for $\overline{M}$ and consider two characters c and d. Let e_c be the edge of T on which character c changes from state zero to state one, and let e_d be the similar edge for character d. Note that all of the taxa that possess character c (or d) are found at the leaves of T below edge e_c (or edge e_d), and one of four cases must hold: Either 1) $e_c = e_d$, or 2) e_c is on the path from the root of T to e_d, or 3) e_d is on the path from the root of T to e_c, or 4) the paths to e_c and e_d diverge before reaching either of those edges.

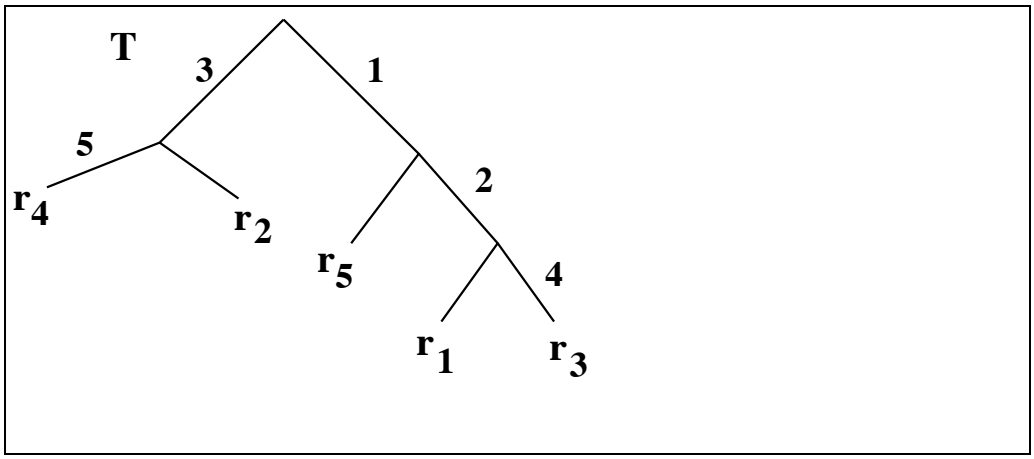


Figure 1.4: Perfect-Phylogeny for the sorted matrix $\overline{M}$ shown in Table 1.1.2.

In case 1), there cannot be a taxon with (ordered) states 0,1 or 1,0 for character pair c, d. In case 2) there cannot be a taxon with (ordered) states 0,1 for c, d; similarly in case 3) there cannot be a taxon with (ordered) states 1,0 for c, d. In case 4) there cannot be a taxon with states 1,1 for c, d. This proves the "only if" direction of the theorem.

We now consider the other direction. We can assume that no pair of columns, c, d, are identical to each other, for if they were identical we could remove one, column c say, and then if T is a perfect-phylogeny for the remaining characters we could add character c on the edge of T that is labeled by character d. The resulting tree would be a perfect-phylogeny for all characters. Hence, it suffices to prove this direction of the theorem assuming that every column is distinct.

By definition, in *any* perfect-phylogeny T for $\overline{M}$ (assuming one exists), and for any taxon f, the characters that appear on the path from the root of T to the leaf labeled f, are exactly the characters that have state one for taxon f. Moreover, those characters must appear in exactly the same order that they appear (left to right) in row f of $\overline{M}$. To see this, suppose taxon f possesses both characters c and d, and that $c < d$. As above, characters c and d are both on the path from the root of T to leaf f. Since the columns of $\overline{M}$ are sorted by the number of ones they contain, there are more taxa with state one for character c then there are with state one for character d, and so the edge e_c labeled by character c occurs above the edge labeled by character d. This is because all the taxa that label the leaves below e_c must possess character c. Hence, in a perfect-phylogeny T (if one exists) the characters on the path from the root of T to a leaf f must be in exactly the same order that they are in $\overline{M}$. So, assuming there is a perfect-phylogeny T for $\overline{M}$, for any taxon f, the set of characters and the order that those characters appear on the path from the root of T to leaf f is precisely and *uniquely* determined. It then follows that there

is a perfect-phylogeny for $\overline{M}$ if and only if those n separate and forced paths can be assembled into a single tree, i.e., where each character labels exactly one edge.

We will show constructively how to assemble those n forced paths into a perfect-phylogeny for $\overline{M}$, under the stated premise of the theorem, that no pair of columns in $\overline{M}$ contains all the binary pairs $0,1$; $1,0$ and $1,1$. To that end, we first develop a property that $\overline{M}$ has when no pair of columns contains all those three binary pairs.

The Shared Prefix Property For two taxa f and g, let d be the largest (rightmost in $\overline{M}$) character that taxa f and g both possess (i.e., where both have state one). Then, assuming no pair of columns contain all three binary pairs $0,1$; $1,0$; $1,1$, rows f and g in $\overline{M}$ must be identical from column one (at the left end of $\overline{M}$) to column d. For example, in Table 1.1.2, character 3 is the largest character that taxa B and D both possess; as required, the rows for B and D are identical (containing 0,0,1) from columns 1 to 3.

To establish the Shared Prefix Property, suppose taxon f possesses a character $c < d$ in $\overline{M}$, so that columns c, d contain the binary pair $1,1$. Since the columns of $\overline{M}$ are distinct and are sorted by the number of ones they contain, columns c, d must also contain the ordered binary pair $1,0$. Therefore (by the premise of the theorem) columns c, d cannot contain the ordered pair $0,1$, and hence taxon g must also possess character c. Now the choice of taxon f in this argument was arbitrary, so the conclusion holds for taxon g also, and hence if either f or g possess a character $c < d$, then both f and g possess character c. It follows that rows f and g are identical in $\overline{M}$ from column one to column d. This establishes the *shared prefix property* for $\overline{M}$.

Constructing a Perfect-Phylogeny The shared prefix property allows a simple algorithm to construct a perfect-phylogeny. The algorithm builds up the perfect-phylogeny T for $\overline{M}$ by processing the rows of $\overline{M}$ in order. It first creates a root node for T and adds to it a single path from the root to a leaf labeled by taxon 1. If taxon 1 possesses t characters, that path will contain t edges successively labeled by one character possessed by taxon 1, in the order that those characters appear in row 1 of $\overline{M}$, followed by a single unlabeled edge leading to leaf 1. Note that this single path is a perfect-phylogeny for the first taxon of $\overline{M}$.

Let T_f denote the intermediate tree that contains all the paths for the taxa from 1 to f. We assume inductively that T_f is a perfect-phylogeny for the first f taxa of $\overline{M}$. Then tree T_{f+1} is constructed from T_f as follows: Starting at the root of T_f, examine the characters that taxon $f+1$ possesses (from left to right in $\overline{M}$) and in parallel, walk from the root of T_f down the (unique) path in the tree, as long as the successive characters on the path match the successive characters that taxon $f+1$ possesses. For example, see Figure 1.5. The path is unique because no character appears more than once anywhere in the perfect-phylogeny

T_f, and in particular, no character appears more than once on the edges leading out of any node. The walk ends at a node, denoted v_{f+1}, where no label on any edge out of v_{f+1} matches the next character that taxon $f+1$ possesses, or where all the characters that taxon $f+1$ possesses have been matched. Let c denote the last matched character on the walk. Then create a new path out of v_{f+1} containing all the characters to the right of c that taxon $f+1$ possesses (in the order they appear in $\overline{M}$), followed by an unlabeled edge to a leaf labeled $f+1$. The result is a tree T_{f+1}.

We claim that T_{f+1} is a perfect-phylogeny for the first $f+1$ taxa of $\overline{M}$. Clearly, each path to a leaf $h \leq f+1$ in T_{f+1} contains exactly the characters that taxon h possesses. Also, since T_f is a perfect-phylogeny, no character on the path to node v_{f+1} is anywhere else in T_f. So to prove that T_{f+1} is a perfect-phylogeny for the first $f+1$ taxa only requires proving that none of the characters on the new path out of v_{f+1} are in T_f. Let d be the rightmost character (in $\overline{M}$) that taxon $f+1$ possesses, such that d labels some edge in T_f. Let e_d denote that edge in T_f. Any taxon h labeling a leaf below e_d in T_f possesses character d, and by the shared prefix property, rows h and $f+1$ of $\overline{M}$ are identical from column 1 to column d. Hence, the walk to v_{f+1} is a walk towards the leaf labeled by taxon h. Moreover, by the choice of character d, and the fact that all characters that taxon h possess are in T_f, taxa h and $f+1$ do not possess any common characters to the right of d. Hence, the characters on the walk from the root of T_f to node v_{f+1} exactly match all the characters that taxon $f+1$ possesses, from the left end of $\overline{M}$ to character d. Therefore, by the choice of d, none of the characters on the new path out of v_{f+1} are in T_f, and hence T_{f+1} is a perfect-phylogeny for the first $f+1$ taxa of $\overline{M}$.

When all taxa have been processed, the resulting tree T is perfect-phylogeny for $\overline{M}$ and for M. This proves the *if* direction, and finishes the proof of Theorem 1.1.1. ■

Note that in this construction, each internal node is labeled. If the resulting perfect-phylogeny contains a node with degree two, other than the root node, we can merge the two incident edges, possibly creating an edge labeled by more than a single character.

The proof of Theorem 1.1.1 shows that *any* perfect-phylogeny for $\overline{M}$ (assuming no duplicate columns) *must* be the superposition of n forced paths. This establishes the following

Corollary 1.1.1 *If there is a perfect-phylogeny for M, and every column in M is distinct, there is only one, unique, perfect-phylogeny for M.*

Corollary 1.1.1 applies when all the columns of M are distinct, but it is easy to remove this assumption. As discussed in the proof of Theorem 1.1.1, we can always remove duplicate characters and still construct a perfect-phylogeny

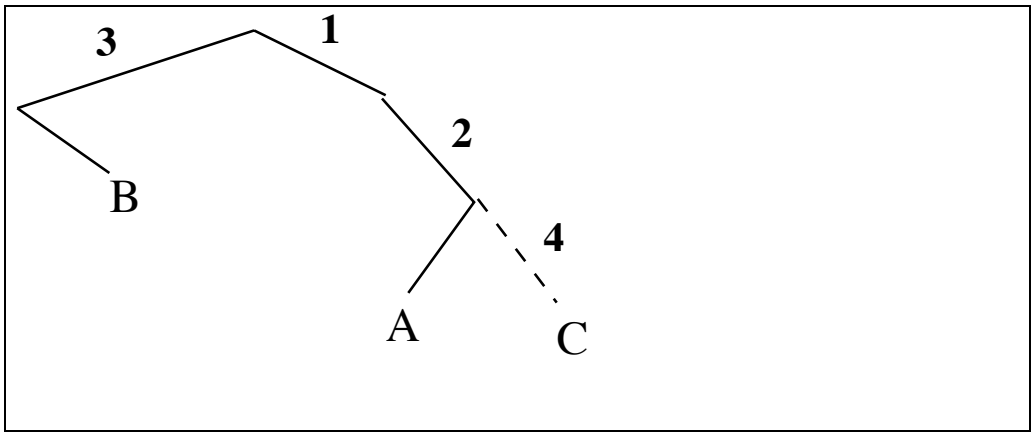


Figure 1.5: The extension from T_B to T_C in the creation of the perfect-phylogeny T shown in Figure 1.4. Here, $f = B$, and the walk from the root follows edges labeled with characters 1 and 2. Then, a new edge to a leaf labeled C is added, and that edge is labeled with character 4.

(if one exists) for all of the data. It then follows from Corollary 1.1.1 that a perfect-phylogeny is always unique, if the reinserted duplicate characters label their common edge as a *set* of characters. It is also possible to build a perfect-phylogeny where duplicate characters label different edges. If so, then the edges labeled by those duplicate characters must be in a consecutive path and the intermediate nodes (nodes not at the two ends of the path) on that path must have degree two. In that case, the perfect-phylogeny is unique except for the order of the characters on that path. We leave the details to the reader. We summarize these facts as:

Theorem 1.1.2 *Suppose M contains some columns which are duplicated and there is a perfect-phylogeny for M. If we require that all internal nodes of any perfect-phylogeny have degree greater than two, then all the perfect-phylogenies for M have the same leaf-labeled topology (i.e., the leaf-labeled trees are identical after removal of all edge labels). Similarly, if we require that every edge be labeled with at most one character, then all the perfect phylogenies for M have the same leaf-labeled topology.*

The proof of Theorem 1.1.1 not only establishes the theorem, it gives a constructive method to build a perfect-phylogeny for M, when one exists.

Theorem 1.1.3 *If there is a perfect-phylogeny T for M, it can be constructed in $O(nm)$ operations. And if each row f of M is presented as the set of characters that taxon f possesses, then T can be constructed in time proportional to the number of ones in M.*

Proof First, to create $\overline{M}$ from M we count the number of ones in each column of M, in $O(nm)$ operations. Those counts range from 0 to n, so we can sort the numbers and the associated columns in $O(m)$ operations by standard bucket sort or counting sort, in decreasing order. The number of operations needed to create T_{f+1} from T_f is $O(m)$, since no character appears twice in T_f, implying that T_f has at most m labeled edges. Thus the total number of operations needed to build the perfect-phylogeny for M is $O(nm)$.

For the second part of the theorem, let t denote the number of ones in M; when the characters possessed by each taxon are presented as a set, t is just the sum of the sizes of these n sets. By scanning the sets, we can create a linked-list $L(c)$ for each character c containing the taxa that possess c. These linked-lists can be built, and the size of each list can be determined, in $O(t)$ operations. The size of the list for character c indicates the number of taxa which possess character c; as above, those numbers (and their associated characters) can be sorted in $O(m)$ operations. Let C be the ordered list of characters, based on this sort. Next we want to reorder the set of characters that each taxon f possesses, to agree with the order of the characters in C. Let $S(f)$ denote the desired ordered set of characters that f possesses. These ordered sets are obtained by processing the characters in C in order; when a character c is processed, we put c at the end of the growing ordered set $S(f)$, for every taxon f in $L(c)$. The n ordered sets are thus built in $O(t)$ total operations, and essentially describe the matrix $\overline{M}$. Next we use the $S(f)$ sets to build up the perfect-phylogeny T as described in the proof of Theorem 1.1.1. The only added detail needed is that when building up T, we create a pointer indexed by c, to the (unique) location of each character c in the tree, at the time that character c is on a new path entered into the tree. Then when inserting the path for any taxon $f+1$, we process the set $S(f+1)$ in order (simulating the left to right scan of row $f+1$ in $\overline{M}$), and when a character c is the next character in $S(f)$, we use the pointer indexed by c to determine if c appears in T_f, and if so, where it appears. In this way, the number of operations needed to build T_{f+1} from T_f is proportional to the size of $S(f+1)$, and the perfect-phylogeny T can be built in $O(t)$ operations in total. ■

The first proven $O(nm)$-time perfect-phylogeny algorithm was given in [12], with a method that is different from the one given here. An alternate version appears in [13]. An earlier perfect-phylogeny algorithm, developed in [23] and discussed in [28], can also be shown to run in $O(nm)$ time, but no time bound was established in [23]. The $O(t)$ method for building a perfect-phylogeny from sets was first established in [2]. Note that a straightforward implementation of Theorem 1.1.1 would give an $\Omega(nm^2)$ time algorithm to determine if M has a perfect-phylogeny, and would not construct one.

An Alternate statement of the Perfect-Phylogeny Theorem

There is an alternate statement of the Perfect-Phylogeny Theorem that is often used. For emphasis, we remind the reader that a perfect-phylogeny is a *rooted* tree and the characters are binary.

Definition: For any column c of $\overline{M}$, let O_c be the set of taxa that possess character c.

Theorem 1.1.4 *There is a perfect-phylogeny for M if and only if for every two characters c and d, either $O_c \cap O_d = \emptyset$, or one set is contained in the other.*

We leave the justification of this to the reader. A direct proof of Theorem 1.1.4 appears in [13].

Other proofs of Theorems 1.1.1 and 1.1.4 and of Corollary 1.1.1 appear in a number of places, for example [5, 7, 6], and in somewhat different language in [28]).

1.2 The case of a known, non-zero, ancestral sequence

We will now examine some ways that the basic assumptions in the definition of a perfect-phylogeny can be relaxed, and how the modified Perfect-Phylogeny Problem can be solved in those cases.

In the perfect-phylogeny model the root of the perfect-phylogeny is labeled with the all-zero sequence, corresponding to the assumption that the ancestral taxa does not possess any of the characters in M. This is a convenient technical assumption, but it is not necessary. Suppose that the binary characters in M are such that the known ancestral taxon r *does* possess some of the characters in M. Therefore, the binary sequence labeling the root should not be the all-zero sequence, but rather a binary sequence where values of one indicate the characters that r possesses. What is the modified model for a perfect-phylogeny in this case, and how can we solve the problem of determining if there is such a perfect-phylogeny?

Definition: The Root-Known Perfect-Phylogeny Given an n by m binary-character matrix M for n taxa, and a given binary sequence S_r (whose 1's indicate the characters that the ancestral taxon r possesses) an *S_r-Perfect-Phylogeny for M* is a *rooted* (directed) tree T with exactly n leaves, with the following properties:

1. The root of T is labeled by the ancestral sequence S_r.
2. Each of the n taxa labels exactly one leaf of T.
3. Each of the m characters labels *exactly one* edge of T.

4. For any taxon f, let C_f be the set of characters labeling edges on the path in T from the root to the leaf labeled f. Then, the binary sequence for f (i.e., the row in M for f) and the binary sequence S_r differ at *exactly* the characters in C_f.

Another way to visualize this definition is that each leaf f is labeled with a binary sequence (the row in M for f) which is derived from the ancestral sequence S_r by walking from the root to leaf f, changing the state of character c when an edge labeled by c is encountered.

Note that a perfect-phylogeny is just a root-known perfect-phylogeny in the case that the known ancestral sequence S_r is the all-zero sequence. Thus, the concept of a root-known perfect-phylogeny generalizes a perfect-phylogeny, but through a small change in the definition of a perfect-phylogeny given earlier.

> **The Root-Known Perfect-Phylogeny Problem:** Given M and m-length binary sequence S_r, determine if there is an S_r-perfect-phylogeny for M.

We efficiently solve this problem by *reducing* it back to the original (all-zero root) perfect-phylogeny problem.

Theorem 1.2.1 *Let M' be the binary matrix obtained from M by interchanging all the 0 and 1 entries in each column c of M where $S_r(c) = 1$. Then there is an S_r-perfect-phylogeny T for M if and only if there is a perfect phylogeny T' (with all-zero ancestral sequence) for M'. Moreover, T and T' have the same leaf-labeled topology.*

As an example, see Table 1.2. Note that no changes are made in any column c of M where $S_r(c) = 0$. Hence the same transformation (changing any 1's to 0's) applied to S_r, creates the all-zero sequence.

Theorem 1.2.1 gives an efficient, constructive way to determine whether there is a perfect-phylogeny with a given ancestral sequence S_r. The proof of Theorem 1.2.1 is simple and is left to the reader as an exercise.

The reduction of the root-known perfect-phylogeny problem to the original perfect-phylogeny problem, combined with the Perfect-Phylogeny Theorem (Theorem 1.1.1), yield the following:

The Root-Known Perfect-Phylogeny Theorem

Theorem 1.2.2 *Binary matrix M has an S_r-perfect-phylogeny if and only if no pair of columns c, d in M contains the three binary pairs that differ from the (ordered) binary pair in positions c, d of S_r. Moreover, if all the columns of M are distinct, and there is an S_r-perfect-phylogeny for M, there is one, unique S_r-perfect-phylogeny for M.*

	c_1	c_2	c_3	c_4	c_5
S_r	1	1	0	0	1
r_1	1	1	0	0	0
r_2	0	0	1	0	0
r_3	1	1	0	0	1
r_4	0	0	1	1	0
r_5	0	1	0	0	0

Table 1.3: When the matrix above is transformed based on S_r, the result is the matrix M shown in Table 1.1. Matrix M has a perfect-phylogeny (with the all-zero ancestral sequence), so there is an S_r-perfect-phylogeny for the above table, and it has the same leaf-labeled topology as the perfect-phylogeny for M.

Proof By case analysis, we see that a pair of columns c, d in M contain the three distinct binary pairs (in rows f, g, h say) that differ from the ordered binary pair in positions c, d of S_r, if and only if the same rows f, g, h in M' contain the distinct ordered binary pairs that differ from the transformed S_r at c, d. But the transformed S_r is the all-zero sequence, so columns c, d in M' contain all the binary pairs $0,1;1,0;1,1$, if and only if columns c, d in M contain all the binary pairs that differ from the binary pair in positions c, d of S_r. Now by Theorem 1.1.1 there is a perfect-phylogeny for M' if and only if no pair of columns in M' contain all three binary pairs $0,1;1,0;1,1$. The first part of the theorem then follows by application of Theorem 1.2.1.

To show uniqueness, let T_1 be an S_r-perfect-phylogeny for M, and let T_1' be the perfect-phylogeny for M'. By Theorem 1.2.1, T_1 and T_1' have the same leaf-labeled topology. Now if there exists a different S_r-perfect-phylogeny T_2 for M, then there exists a perfect-phylogeny T_2' for M that differs from T_1', which violates the statement of uniqueness in Theorem 1.1.1. ■

Theorem 1.2.2 can be extended to handle the case of duplicate characters, in the same way that Theorem 1.1.2 extends Corollary 1.1.1. We leave this as an exercise for the reader.

1.3 The Root-Unknown Perfect-Phylogeny Problem

We now consider the most relaxed perfect-phylogeny model, when *no* ancestral sequence is specified as part of the problem instance?

Definition Given a binary matrix M, but no specified ancestral sequence, the *root-unknown Perfect-Phylogeny Problem* is to determine if there exists some binary sequence S_r such that there is an S_r-perfect-phylogeny. Moreover, an ancestral sequence S_r should be explicitly identified if one exists.

Definition In an undirected, unrooted tree with a node of degree two, the *contraction* of a node v of degree two removes v and merges the two edges incident with v into a single edge. The new edge is labeled by the union of the characters that labeled the two merged edges. See Figure 1.3, parts a) and b).
Definition An *undirected perfect-phylogeny* for M is a tree that is obtained from a rooted perfect-phylogeny T for M, after removing all the directions on the edges of T, and successively contracting nodes of degree two. See Figure 1.3 parts c) and d).

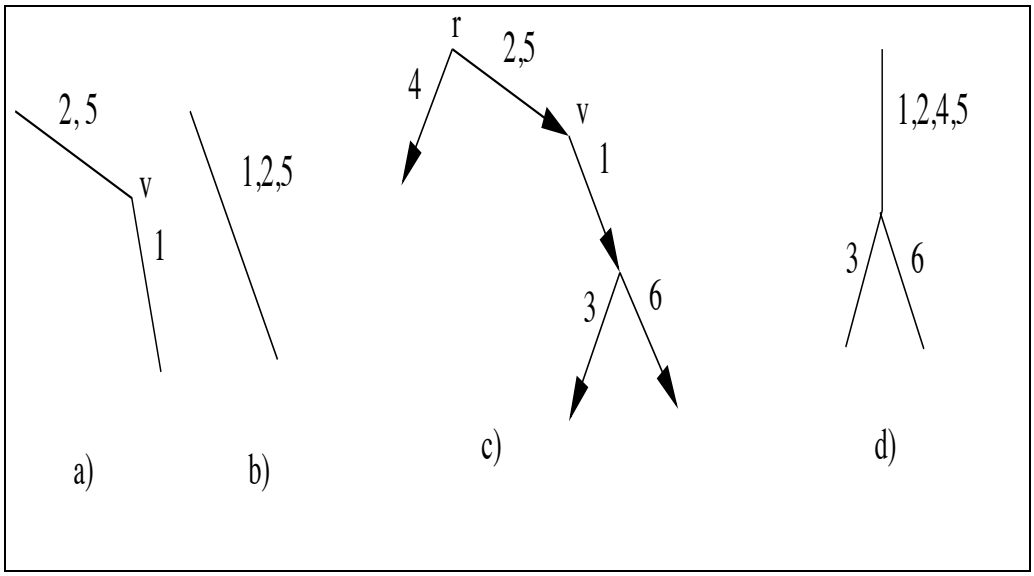


Figure 1.6: a) Two edges incident with node v of degree two. b) The result of contracting node v. c) A rooted directed tree with two nodes, r and v of degree two. d) The undirected tree obtained by removing the directions on all edges, and successively contracting the nodes r and v of degree two.

The contraction of nodes of degree two is in order to address the case of duplicated characters, and the fact that the root of an $S_r - perfect - phylogeny$ might have degree two even if there are no duplicate characters. Note that in the definition of an undirected perfect-phylogeny T, all internal nodes of T have degree greater than two.

The root-unknown perfect-phylogeny problem can equivalently be stated as the problem of determining if there is an undirected perfect-phylogeny for M, and the root-unknown perfect-phylogeny problem is also called the *undirected* or *unrooted* perfect-phylogeny problem. We prefer the term *root-unknown perfect-phylogeny problem* because we have defined a perfect phylogeny as a rooted, directed tree. However, when no ancestral sequence is known (and this is often the case), it may be more consistent with the known information if a perfect-phylogeny T is converted to an undirected perfect-phylogeny.

To solve the root-unknown perfect-phylogeny problem, we again use reduction, in this case to the root-known perfect-phylogeny problem. To see the reduction, suppose there is a perfect-phylogeny T for M with some ancestral sequence S_r. Each taxon f in M defines the sequence labeling a leaf of T, and hence there is a directed path in T from the root to leaf f. Now imagine grabbing the leaf f and raising it above the root of T, making T hang down from f. We could then consider f as the root of the resulting directed tree T_f, where all of the sequences in M are derived from f. In particular, the sequence for any taxon g is obtained from the sequence for f by changing the state of the characters on the path from f to g. Tree T_f is almost a f-perfect-phylogeny for M, but it does not strictly obey the definition of a perfect phylogeny because f does not label a leaf. To rectify that, we simply add an unlabeled edge from the root of T_f to a new leaf labeled f. This establishes the following:

Theorem 1.3.1 *Given a binary matrix M and no specified ancestral sequence, if there is an undirected perfect-phylogeny for M, then there is an f-perfect-phylogeny for M, where f is any of the taxa in M.*

Note that in the transformation from T to T', no edge labels changed, and the leaf-labeled topology also did not change except for the addition of the edge to the new leaf labeled f.

Theorem 1.3.1 leads to an efficient algorithm for the root-unknown perfect-phylogeny problem: declare the sequence of some (any) taxon f in M to be the ancestral sequence S_r, and then solve the root-known perfect-phylogeny problem. If a (rooted, directed) S_r-perfect-phylogeny T is obtained, but an undirected tree is more biologically valid, remove the directions on all edges of T.

The Root-Unknown Perfect-Phylogeny Theorem

Theorem 1.3.1 not only leads to an efficient algorithm, it leads to the classical necessary and sufficient condition for the existence of a *root-unknown* perfect-phylogeny for binary matrix M:

Theorem 1.3.2 The Four-Gametes Theorem *When no ancestral sequence is known, matrix M has an undirected perfect-phylogeny if and only if no pair of columns contains all four binary pairs 0,0; 0,1; 1,0, and 1,1.*

Proof If there is an undirected perfect-phylogeny for M then by Theorem 1.3.1 there is an f-perfect-phylogeny for any taxon f in M. Therefore by Theorem 1.2.2 no pair of columns c, d in M can have all three binary pairs that differ from the binary pair in columns c, d for sequence f. Then, since f is also in M, no pair of columns in M can have all four binary pairs.

Conversely, for any taxon f, if no pair of columns in M contain all the four binary pairs, then no pair of columns of M contain all three binary pairs that

differ from the binary pair that sequence f has in those columns. Therefore, by Theorem 1.2.2, M has an f-perfect-phylogeny T, and so M has the undirected perfect-phylogeny obtained from T. ■

The condition in Theorem 1.3.2 is called the *four-gametes* condition, or four-gametes test in the population genetics literature [15], and is called the *compatibility condition* in the phylogenetics literature [28, 8].

Back to a rooted problem

Through the reductions used in the proofs, we have seen a very close relationship between unrooted, undirected versions of problems and results and root-known versions. As another reflection, we can now give a different, but equivalent, way to discuss Theorem 1.2.2, as follows:

Definition Given an n by m binary matrix M and a fixed binary sequence S of length m (which need not be in M), let $M+S$ be the binary matrix created by adding S to M.

Corollary 1.3.1 *Binary matrix M, and an ancestral sequence S_r (which need not be in M), M has an S_r-perfect-phylogeny if and only if no pair of sites in $M + S_r$ contains all four gametes.*

1.3.1 Uniqueness

The Four-Gamete Theorem establishes when there is an undirected perfect-phylogeny for a binary matrix, but does not address the question of uniqueness. We address that now.

Theorem 1.3.3 *Suppose M has an undirected perfect-phylogeny. If all columns of M are distinct, then there is one, unique, undirected perfect-phylogeny for M. If the columns of M are not all distinct, all of the undirected perfect phylogenies for M have the same leaf-labeled topology.*

Proof Suppose all the columns of M are distinct and that there are two different undirected perfect phylogenies, T and T', for M. Let f be a taxon in M; hence there is a leaf labeled f in both T and T'. By Theorem 1.3.1 there are two f-perfect-phylogenies T_f and T'_f. These trees must be different because T and T' are different, and the transformations of T and T' to T_f and T'_f respectively add a new edge from the root to the new leaf f, and preserve all the prior edges and edge labels. But by Theorem 1.2.2, when all columns of M are distinct, there can be only one unique f-perfect-phylogeny for M. ■

A note on the importance of uniqueness Theorems 1.2.2 and 1.3.3, and Corollary 1.3.1 all establish that when there is a perfect-phylogeny (under different models), there is only one, *unique* perfect-phylogeny. Uniqueness has probably been a critical factor in making the perfect-phylogeny model (under different names in different areas of biology) of interest in biology. Uniqueness implies that *if or when* the underlying biological reality fits the mathematical assumptions of the perfect-phylogeny model, the obtained tree will be *the* biologically correct and relevant tree. It is not just one tree among many that display some aspects of the data – it is the tree that explains the true evolutionary history of the sequences. In contrast, even if a model perfectly captures the underlying biological reality, when a large set of trees fit that model, one does not know how much biologically correct information can be extracted from any *one* of those trees, or which of the trees is the most biologically informative.

So, the uniqueness of the perfect-phylogeny is a very significant feature. Of course, the caveat is that this feature is of greatest utility *when* the biological reality fits, or nearly fits, the mathematical assumptions of the model. We will see in Chapter **??** that uniqueness, or essential-uniqueness, is also one of the primary attractions of a generalization of perfect-phylogeny to networks called galled-trees.

In addition to uniqueness, one of the arguments in favor of the perfect-phylogeny model, is that a set of random binary sequences is very unlikely to be derivable on a by perfect-phylogeny. Therefore, when one has a set of binary sequences that can be derived on a perfect-phylogeny (or can after some small modification of the data), there is a strong belief that the history of the sequences did conform to the perfect-phylogeny model.

Alternate roots

There is an alternative reduction that can solve the undirected perfect-phylogeny problem, which uses an interesting fact detailed in Theorem 1.3.4 below.

Definition For any character c, if more than half of the taxa have state i for c, then i is called the *majority state* of c. A character c does not have a majority state if and only if the number of taxa that possess character c exactly equals the number of taxa that do not possess c. A sequence S_m is said to be a *majority sequence* if S_m has the majority state for every character c that has a majority state. If character c does not have a majority state, then the value of S_m at position c is permitted to be either 0 or 1.

Theorem 1.3.4 *Given a binary matrix M and no specified ancestral sequence, if there is an undirected perfect-phylogeny for M, then there is an S_m-perfect-phylogeny for M, where S_m is a majority sequence for M.*

There are applications where the use of Theorem 1.3.4 is preferred to the use of Theorem 1.3.1. We leave the proof of Theorem 1.3.4 as a simple exercise for the reader.

Finally, we note the undirected analog to Theorem 1.1.4. Recall that O_c and O_d are the sets of taxa that posses characters c and d respectively. Let $\overline{O_c}$ and $\overline{O_d}$ be the sets of taxa that don't posses characters c and d respectively.

Theorem 1.3.5 *There is an undirected perfect-phylogeny for M if and only if for every two characters c and d, one of the sets $O_c \cap O_d$, $\overline{O_c} \cap O_d$, $O_c \cap \overline{O_d}$, $\overline{O_c} \cap \overline{O_d}$ is empty.*

1.4 The Splits-Equivalence Theorem

In this book we have chosen to first expose the combinatorial structure of evolutionary trees through the viewpoint of the perfect-phylogeny model and variants of it. However, a different approach is also common [28], using the viewpoint of *splits* and the *Splits-Equivalence Theorem*. In this section we will develop that theorem and show that it is essentially the same as Theorem 1.3.3 developed for the root-unknown perfect-phylogeny problem. Thus, although the two viewpoints may at first seem different, they are really addressing the same combinatorial phenomena.

Let T be an undirected tree whose leaves have distinct labels. The removal of any edge e from T creates exactly two connected subtrees, defining a bipartition the leaves of T.

Definition We define the *split for e* as the bipartition of the leaves (equivalently, the leaf labels) defined by the two undirected subtrees resulting from the removal of edge e from T. Given a tree T with m edges, we define the *splits of T* as the set of m splits, one split for each edge in T. Note that if T has a node of degree two, then there will be two adjacent edges which define exactly the same split. This will not happen if each non-leaf node has degree at least three.

For example, there are eight splits in the tree shown in Figure 1.4 (on page 9). Those splits are:

$$\{r_4\},\{r_1,r_2,r_3,r_5\};\{r_2\},\{r_1,r_3,r_4,r_5\};\{r_2,r_4\},\{r_1,r_3,r_5\};\{r_2,r_4\},\{r_1,r_3,r_5\};\{r_5\},$$

$$\{r_1,r_2,r_3,r_4\};\{r_1,r_3\},\{r_2,r_4,r_5\};\{r_1\},\{r_2,r_3,r_4,r_5\};\{r_3\},\{r_1,r_2,r_4,r_5\}.$$

Note that the two edges labeled with sites 1 and 3 define the same split.

The splits of a tree are very informative, as shown next.

Theorem 1.4.1 The Splits-Equivalence Theorem *For any undirected tree T with distinct leaf labels, the splits of T uniquely define T.*

Stated differently, for any (distinctly) leaf-labeled, undirected tree T, there is no other undirected tree with the same set of splits as T. Therefore, we can uniquely reconstruct T if we know the splits of T. This is one of the most fundamental and useful facts about the combinatorial structure of trees. We will prove the Splits-Equivalence Theorem by relating it to the root-unknown perfect-phylogeny problem and Theorem 1.3.3.

Proof of Theorem 1.4.1 We assume that the splits of T are distinct, and leave the case when they are not distinct to the reader. We represent the splits of T in a binary matrix SP where each leaf of T is represented by a row of SP and each split of T (equivalently, each edge in T) is represented by a column of SP. The zeros in the column for split e identify the leaves on one side of the split, and the ones in the column identify the leaves on the other side of the split. Note that we could interchange all the zeros and ones in any column and still define exactly the same bipartition; the zeros and ones only serve to specify the bipartition and have no other meaning.

Now consider matrix SP as an input matrix to the root-unknown perfect-phylogeny problem. We claim that there is an undirected perfect-phylogeny for SP with the same leaf-labeled topology as T. To show this in detail, we must exhibit an S_r-perfect-phylogeny for SP which becomes T when all edge directions are removed, and every node of degree two is contracted. To create the desired S_r-perfect-phylogeny, label each edge e in T by the column in SP associated with e (i.e. the column that describes the split for e in T), and label each leaf in T by the row in SP associated with that leaf. Next, choose any leaf f in T to be the root and set the ancestral sequence S_r to the sequence for f in SP. Finally, add a new edge from the root to a new leaf labeled f and direct all the edges away from the root. The result is an S_r-perfect-phylogeny for SP that establishes that T is an undirected perfect-phylogeny for SP.

It follows from Theorem 1.3.3, since all the splits in T are distinct, that T is the *unique* undirected perfect-phylogeny for matrix SP. Further, if we interchange the zeros and ones in any column of SP, creating matrix SP', tree T will also be the unique undirected perfect-phylogeny for SP'. The needed S'_r-perfect-phylogeny for SP' is obtained from the S_r-perfect-phylogeny for SP by interchanging the zeros and ones in S_r at every position where the values in SP were interchanged to create SP'. Hence, no matter how the splits of T are encoded in SP, tree T is the unique undirected perfect-phylogeny for SP.

Now suppose there is another tree T' which is different from T but has exactly the same splits as T, encoded in a matrix SP'. By the argument in the prior paragraphs, T' is an undirected perfect-phylogeny for SP'. But no matter how the splits of T' are encoded, matrix SP' also describes the splits of T, so T is the *unique* undirected perfect-phylogeny for SP', contradicting the assumption that T and T' are different. ■

The proof of Splits-Equivalence Theorem assumes that all splits are distinct,

but it can be easily extended to the case when there are duplicate splits. In that case, all undirected perfect-phylogenies for the matrix SP (encoding the splits of T) will have the same leaf-labeled perfect phylogenies, and this can be used to prove that the splits of T uniquely define T, even if some of the splits are not distinct. We leave the details to the reader. We note that T can have a duplicate split only if it has an internal node of degree two, so the Splits-Equivalence theorem as stated applies to the common case that all internal nodes have degree greater than two.

The existence problem

The Splits-Equivalence Theorem says that the splits of an existing undirected tree T uniquely define T. But often we are given a set of splits, and need to determine *if* they come from an undirected tree T. A little reflection shows that this existence problem has already been solved.

We showed above that when the splits come from a tree T and are encoded in a binary matrix SP, tree T is the unique undirected perfect-phylogeny for SP, so to determine if a set of splits come from a tree T, we simply consider the splits in matrix SP and apply the Four-Gametes Theorem. However, the splits literature uses somewhat different terminology, as follows.

Definition A pair of columns in a binary matrix is called *incompatible* if they contain all four binary pairs 0,0; 0,1; 1,0; and 1,1. Otherwise the pair is called *compatible*. A split can be represented as a column in a binary matrix SP, as above. We say that two splits are compatible if and only if their associated columns in SP are compatible.

Using this terminology, the Four-Gamete Theorem becomes:

Theorem 1.4.2 *Let SP be a binary matrix defining a set of splits of a set Z. Then there exists an undirected tree T whose leaves are labeled by Z, and whose splits, defined by the edges of T, contain the splits of SP, if and only if every pair of columns in SP is compatible.*

Definition An edge $e = (u, v)$ in a tree is called an *internal edge* if neither u nor v is a leaf node.

We can strengthen Theorem 1.4.2, requiring that every internal edge of T define a *distinct* split of SP, by successively identifying and contracting any edge that defines a split defined by another remaining edge of T. We call such a tree a "reduced tree". Note that in a reduced tree, the split defined by an edge that touches a leaf need not be a split in SP. However, every tree whose leaves are labeled by Z will contain this set of $|Z|$ splits defined by an edge touching a leaf. Then by Theorem 1.4.1, it follows that

Theorem 1.4.3 *If there is an undirected tree T whose splits contain the splits of SP, then there is a unique reduced tree whose splits contain the splits of SP.*

Although compatibility and incompatibility have been defined as properties of *pairs* of columns (sites, characters), we will sometimes need to focus on individual characters, and will somewhat abuse the definitions as follows.

Definition An individual character c in M is called *compatible* if c is not incompatible with any character in M. If a character c is incompatible with some other character, then we will simply say that c is *not compatible*.

A rooted version of the Splits-Equivalence Theorem

Suppose T is a *rooted* tree with n leaves and m edges, where each leaf has a distinct label. A split for an edge e again creates a bipartition of the leaves of T, but now the two sides of the split can be distinguished by noting which side contains the root of T. These *rooted* splits can be represented by an n by m matrix SP where each row represents a leaf and each column represents an edge of T; a cell $SP(i, e)$ has the value of 0 if leaf i is in the subtree of $T - e$ containing the root of T, and has value 1 otherwise. Note that the set of leaves with value 1 in the column for edge e, precisely specifies all of the leaves below e in T. In some literature, this set of leaves is called a *cluster*.

Observe that $SP(i, e) = 1$ if and only if edge e is on the path from the root of T to leaf i. That is, each row in SP precisely specifies the edges in the path from the root of T to leaf i. Hence we can interpret SP as input M to the perfect-phylogeny problem, and interpret T as a perfect-phylogeny for M. Moreover, any other tree T' that has exactly the same splits as given in SP is also a perfect-phylogeny for M. By the Perfect-Phylogeny Theorem, when a binary matrix M can be represented by a (rooted) perfect-phylogeny (with all-zero ancestral sequence), the perfect-phylogeny for M is unique. This implies the following

Theorem 1.4.4 *The set of rooted splits (or clusters) of a rooted, directed, tree T uniquely determine T, including its root and the direction of each edge.*

Incompatibility is defined for undirected problems. There is also a notion of incompatibility that is used for rooted or directed problems.

Definition Given an n by m binary matrix M and a binary sequence s of length m, then two sites c and d in M are said to *conflict relative to s* if c and d are incompatible in $M + s$.

With this definition, we can restate Theorem 1.2.2 as follows:

Theorem 1.4.5 *Binary matrix M has an S_r-perfect-phylogeny if and only if no pair of sites conflict relative to S_r.*

Sometimes we can assume that S_r is part of M, in which case we will only need results concerning incompatibility, rather than results explicitly about conflict.

1.5 Advanced Material: extensions of perfect-phylogeny to non-binary data

So far in this book, we have assumed that the data is binary. This assumption is biologically valid in most contexts of current interest, for example in modeling SNP data, or modeling the presence or absence of complex traits. In all but this section of the book, we will continue to to assume that the data is binary; this has been the universal assumption in all studies of networks with recombination.

However, despite the current and anticipated centrality of binary data, other important *multi-state* (non-binary) polymorphism data are now being systematically collected in populations, and the frequencies of these polymorphisms are much greater than has been assumed in the past [29, 16, 27, 32, 33, 22, 17]. Some of these polymorphisms have functional consequence and may be under selective pressure [25, 26, 30]. Non-binary data consists of sequences, one from each sampled individual in a population, where the value at a single site is not restricted to 0 or 1, but can be a larger *integer* (the allele is "multi-allelic" rather than "diallelic"). The meaning of an integer at a site varies by the type of polymorphism. In some cases it is an actual count and has an ordinal meaning, and in other cases it only identifies the state of the polymorphism (as in the binary case). The need to handle such data has lead to a generalization of the (binary character) perfect-phylogeny model to the *Multi-State Perfect-Phylogeny* model.

In this section we introduce a little bit about the multi-state perfect-phylogeny problem. In particular, we will discuss the *three-state* perfect-phylogeny problem in detail, as it is closely related to the binary case.

Introduction to k-state Perfect-Phylogeny

In the **k-state Perfect-Phylogeny Problem**, the input is an n by m matrix M whose values are *integers* from the set $Z(k) = \{1, 2, ..., k\}$. Each row of M again represents a single taxon; each column of M represents a single character; and each value in cell (f, c) is the state of character c that taxon f possesses.

A *directed k-state Perfect-Phylogeny* for M is a generalization of a perfect-phylogeny (for binary data). In the binary case, each character changes (mutates) exactly once from the state it has in the ancestral sequence. The natural generalization to k states is to allow a character to mutate $k-1$ times in a perfect-phylogeny, but to insist that for any character c and any state i of character c, there is at most one mutation in the tree that changes the state of character c to i.

Although true evolutionary history is always directed (in time), the true ancestral sequences and the true root may not be known. Consequently, the literature on multi-state perfect-phylogeny has usually addressed the undirected version, and we will do that here. We now state a more formal definition.

Definition For any character c and any sate i of c, the set of taxa in M that possess state i for character c is denoted by $X_c(i)$; the set of taxa which do not possess state i is denoted by $\overline{X_c(i)}$. Note that $X_c(i), \overline{X_c(i)}$ defines a split of the taxa.

Definition Given M as above, a *taxa-labeled* tree T for M is an undirected tree with n leaves, where each leaf is labeled by a distinct taxon in M, and each internal node of T is labeled by a vector from $Z(k)^m$ (which need not be in M), specifying the state of each of the m characters. We define $T_c(i)$ as the subgraph of T induced by the nodes in T that are labeled with state i for character c.

Definition A taxa-labeled tree T for M is a Perfect-Phylogeny for M if and only if, for every character c and every state i of c, the subgraph $T_c(i)$ is a *connected subtree* of T. An example is shown in Figure 1.5.

This definition of a *multi-state* perfect-phylogeny is the natural generalization of the Second Definition given for a (rooted, binary) perfect-phylogeny on page 5.

The requirement in the definition of a perfect-phylogeny that each subgraph $T_c(i)$ be a subtree is called the *convexity* requirement. For another way to view convexity, arbitrarily designate a node in T as the root and direct all the edges in T away from the root; consider this directed tree as giving a history of character mutations. The convexity requirement is then equivalent to saying that for any character/state pair (c, i), there is at most one edge in T where the state of character c mutates to i. Note that for any character c and states $i \neq j$, the convexity requirement implies that subtrees $T_c(i)$ and $T_c(j)$ of a perfect-phylogeny T must be node disjoint.

Definition The k-state Perfect Phylogeny Problem is to determine, for input M, if there is a k-state perfect-phylogeny for M, and to construct one if there is one.

If none of the parameters k, n or m is fixed (so k can grow with n), then the k-state perfect-phylogeny problem is NP-complete [3, 31]. In contrast, if k is any fixed integer, independent of n, then the problem can be solved in time that is polynomial in n and m. In fact, for $k = 2$, we showed in Theorem 1.1.3 that the problem can be solved in linear time. A polynomial-time solution for $k = 3$ was shown in [4]; a polynomial-time solution for $k = 3$ or 4 was shown in [18]; and a polynomial bound for any *fixed* k was shown in [1]. The later result was improved in [19] to a time bound of $O(2^{2k}nm^2)$. An excellent survey of most of these results appears in [9].

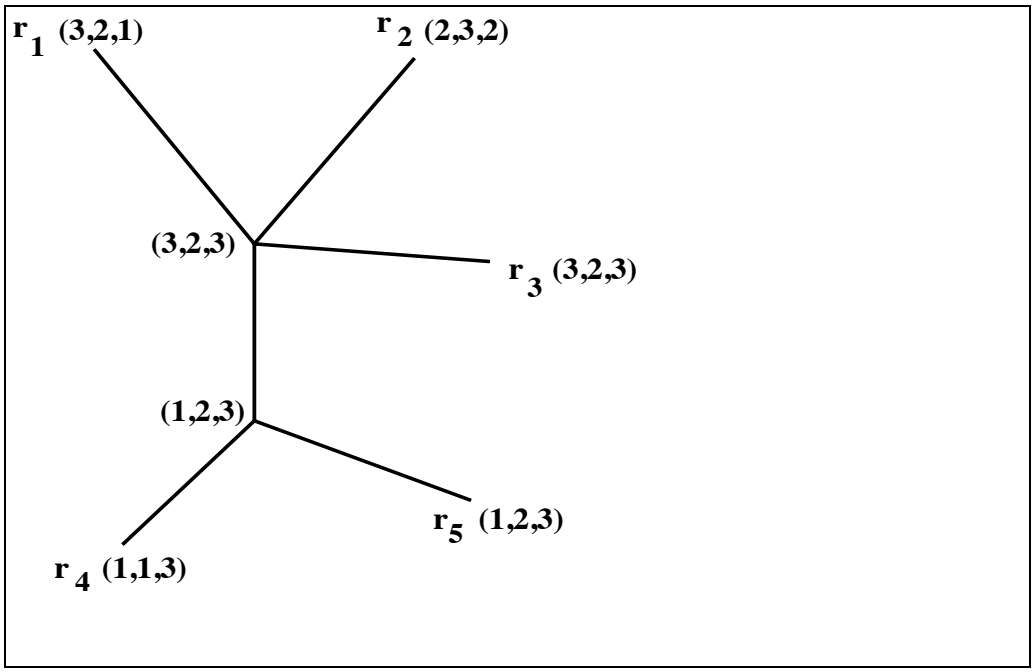


Figure 1.7: A three-state perfect-phylogeny with $n = 5, m = 3$. The input M is given in Table 1.5. The subtree $T_3(3)$ contains the leaves labeled r_3, r_4, r_5 and the two interior nodes.

The polynomial-time algorithm for $k = 3$, developed in the paper by A. Dress and M. Steel [4] is relatively simple in comparison to the other methods, and is related to the solution to the binary case. Hence, we will discuss that method in detail.

1.5.1 The Dress-Steel solution to the 3-state Perfect Phylogeny Problem

For the exposition, create another matrix $\overline{M}$ derived from the input matrix M, with three characters $C_c(1), C_c(2), C_c(3)$ for each character c in M. All the taxa that have state i for c in M are given state 1 for character $C_c(i)$ in $\overline{M}$, and the other taxa are given state 0 for $C_c(i)$. So, the original input matrix M is recoded as a *binary* matrix $\overline{M}$ with three *expanded* binary characters for each character in M. Table 1.5.1 shows the matrix $\overline{M}$ expanded from matrix M in Table 1.5. Note that each expanded character defines a split of the taxa. The main structural result in [4], interpreted in terms of $\overline{M}$ is:

Theorem 1.5.1 *[4] Given matrix M with $k = 3$, there is a 3-state perfect-phylogeny for M, if and only if there is a set of (binary) characters S of $\overline{M}$ which are pairwise compatible, where for each character c in M, S contains at*

	c_1	c_2	c_3
r_1	3	2	1
r_2	2	3	2
r_3	3	2	3
r_4	1	1	3
r_5	1	2	3

Table 1.4: Input matrix M for the 3-state perfect-phylogeny show in Figure 1.5.

	$C_1(1)$	$C_1(2)$	$C_1(3)$	$C_2(1)$	$C_2(2)$	$C_2(3)$	$C_3(1)$	$C_3(2)$	$C_3(3)$
r_1	0	0	1	0	1	0	1	0	0
r_2	0	1	0	0	0	1	0	1	0
r_3	0	0	1	0	1	0	0	0	1
r_4	1	0	0	1	0	0	0	0	1
r_5	1	0	0	0	1	0	0	0	1

Table 1.5: Matrix $\overline{M}$ resulting from expanding the matrix M shown in Table 1.5

.

least two of the characters $C_c(1), C_c(2), C_c(3)$.

Proof Suppose there is a 3-state perfect-phylogeny T for M. For any character c of M, the subtrees $T_c(1), T_c(2)$ and $T_c(3)$ are node disjoint and contain all the nodes of T. Now for each character c, contract, in T, all of the nodes of $T_c(i)$ to a single node. The resulting graph must be a path P_c with three nodes; we label each node v in P_c with the distinct state (1, 2, or 3) of the nodes that contract to v. For example, in the perfect-phylogeny T shown in Figure 1.5, if we contract each of the subtrees $T_3(1), T_3(2), T_3(3)$ to a single node, we get a path P_3 labeled with end nodes 1 and 2 and with interior node labeled 3.

In general, we use i and j to denote the state-labels of the two nodes at the leaves of P_c. Since P_c is a path with two edges, there is an edge e in P_c the node labeled i from the interior node and the node labeled j. Edge e is an uncontracted edge from T, and so edge e separates all the taxa with state i for character c from all the taxa with the other two states, and hence defines a bipartition of the taxa and the split $(X_c(i), \overline{X_c(i)})$. Similarly, there is also an edge in T that defines the split $(X_c(j), \overline{X_c(j)})$. Then, for character c, select characters $C_c(i)$ and $C_c(j)$ to be in S. Repeating this for each character c of M selects a set S of characters of $\overline{M}$ that contains exactly two expanded characters for each character c in M. Further, since each selected split is defined by an edge

in T, and every pair of splits defined by edges in T are compatible (by Theorem 1.4.2), the characters in S are pairwise compatible, and the necessary direction of Theorem 1.5.1 is proved.

Conversely, suppose there is a set of characters S in $\overline{M}$ satisfying the conditions of Theorem 1.5.1. Let $\mathcal{Z}$ denote the set of taxa in M. By construction, each character in $\overline{M}$ defines a split of the taxa $\mathcal{Z}$, and so S defines a set of pairwise-compatible splits of the taxa. For a taxon s in M, the "trivial split" for s is the bipartition $\{s, \mathcal{Z} - s\}$, which is clearly compatible with any other split. We augment the splits defined by S with these n trivial splits, and call the resulting set of splits S'. By the Splits-Equivalent Theorem there is some tree T' with n leaves, each labeled with a distinct taxon in $\mathcal{Z}$, and containing edges that define the splits in S'. We can assume that each edge in T' actually defines one of the splits in S', by contracting any edge that does not define such a split. Also, we can assume that no internal node of T' has degree two, since otherwise two neighboring edges define the same split, in which case one edge can be contracted. We now show how to map the taxa to leaves of T and how to label the interior nodes in T' so that T becomes a perfect-phylogeny for M.

Because of the trivial splits in S', each taxon in $\mathcal{Z}$ labels a leaf of T', satisfying one requirement for a perfect-phylogeny for M. We next need to show how to label the interior nodes of T' so that for every character c and every state i for c, $T'_c(i)$ is a connected subtree of T'. For a character c in M, suppose, without loss of generality, that characters $C_c(1)$ and $C_c(2)$ are in S', and let $e(1)$ and $e(2)$ be the edges in T' that define the splits $(X_c(1), \overline{X_c(1)})$, and $(X_c(2), \overline{X_c(2)})$. Removal of $e(1)$ from T' creates two connected subtrees, one which contains all and only the taxa in $X_c(1)$ labeling its leaves. Label each of the nodes in that subtree with state 1 for character c, defining subtree $T'_c(1)$. Define T'' as the tree T' after the removal of all nodes and edges in $T'_c(1)$. Clearly, T'' contains all the leaves labeled by taxa in $X_c(2)$. T'' also contains edge $e(2)$; otherwise $e(2)$ would be an edge in $T'_c(1)$ and since all interior nodes have degree three or more, there would be a leaf labeled 1 on both sides of $e(2)$, contradicting the assumption that $e(2)$ defines the split $(X_c(2), \overline{X_c(2)})$. So, removal of $e(2)$ from T'' defines two connected subtrees of T', one which contains all and only the taxa in $X_c(2)$; label the nodes of that subtree with state 2 for character c, defining $T'_c(2)$. Removing $T'_c(2)$ from T'' leaves a connected subtree of T' that must contain all and only the leaves labeled by taxa in $X_c(3)$. Label the nodes in that subtree with state 3 for character c, creating $T'_c(3)$. These three subtrees are node disjoint and show that character c obeys the convexity requirement. Since the argument holds for any c, we conclude that T' (with interior nodes labeled as above) is a 3-state perfect-phylogeny for M. ■

A polynomial-time algorithm for the three-state perfect-phylogeny problem

In order to find a perfect-phylogeny for M, Theorem 1.5.1 requires we select a set S of at least two characters from $C_c(1), C_c(2), C_c(3)$, for each character c in M, such that the characters in S are pairwise compatible in $\overline{M}$. This may seem at first to be a computationally difficult task since there are four possible choices for each character c, leading to a time of $\Omega(4^m)$ if all choices are explicitly considered. How can we make the selections efficiently? Below we will explain the polynomial-time solution developed in [4]. However, a more direct approach [14] is to observe that the selection problem can be formulated as the *satisfiability* problem where clauses only contain two literals. This is the classic *2-SAT* problem, which is well known [11] to have a polynomial-time solution[1].

Given $\overline{M}$ we want to create, in polynomial time, a 2-SAT formula $\mathcal{F}$ that is satisfiable if and only if we can select a set of characters S in $\overline{M}$ that obeys the conditions described above. To model the condition that S cannot contain two incompatible characters in $\overline{M}$, suppose $c(i)$ and $c'(i')$ are incompatible. The clause

$$(\neg c(i) \vee \neg c'(i')),$$

where $\neg$ indicates boolean negation, imposes that condition that S cannot contain both characters. Formula $\mathcal{F}$ will contain such a clause for each pair of incompatible characters in $\overline{M}$. To model the condition that S must contain at least two characters from $C_c(1), C_c(2), C_c(3)$, create the following three clauses:

$$(C_c(1) \vee C_c(2))$$

$$(C_c(1) \vee C_c(3))$$

$$(C_c(2) \vee C_c(3))$$

Formula $\mathcal{F}$ will conatin such a set of three clauses for each character c in M. Clearly, $\mathcal{F}$ can be constructed in polynomial time from $\overline{M}$, and every clause in $\mathcal{F}$ has only two literals. We leave it to the reader to fully prove that $\mathcal{F}$ is satisfiable if and only if a proper set of characters S can be selected.

1.5.1.1 The Dress-Steel algorithm

For a warm-up to the general method, consider a character c in M and its three expanded characters $C_c(1)C_c(2), C_c(3)$ in $\overline{M}$. If one of these characters, say $C_c(1)$ for concreteness, is incompatible with two expanded characters from another character c', then character C_c1 must be excluded (not selected for S) because selecting it would make it impossible to select two (and certainly

[1] If you are not familiar with the 2-SAT problem, you can skip ahead to the subsection 1.5.1.1 without loss of understanding.

three) compatible characters from $C_{c'}(1)C_{c'}(2), C_{c'}(3)$. If any character in M has two expanded characters in $\overline{M}$ that must be excluded, then there is no perfect-phylogeny for M.

In general when a character $C_c(j)$ is excluded from S, both of the other characters that are expanded from c must be included in S. Similarly, when a character $C_c(j)$ is selected for S, any character that is incompatible with it must be excluded from S. So when a character $C_c(j)$ is included or excluded, a series of other *forced* inclusions and exclusions may be created.

With this warm up, we can present the full algorithm, shown in Figure 1.5.1.1.

```
Dress-Steel Algorithm (M)
    S = ∅

    while (there is a character c in M where fewer than two of the
    expanded characters C_c(1), C_c(2), C_c(3) have been selected)  do

       Tentatively, select one of those unselected characters,
        say C_c(1) for concreteness, and then follow the series of
        forced character inclusions and exclusions until either there
        are no more forced decisions, or until the series finds
        a problem, i.e., two characters expanded from some c are forced
        to be excluded or until the series leads to a reversal of a prior decision.

       if (the forced series ends in the first way) then
          accept all of the decisions made in the series.
       endif

       if (If the series ends by finding a problem) then
          undo all the decisions made in that series, exclude C_c(1) from S
          and follow the new series of forced decisions.
       endif

       if (the second series also ends with a problem) then
          declare there is no perfect-phylogeny for M, and stop early.
       else
          accept all the decisions made in the second series.
       endif

       if (the algorithm reaches this point,
       i.e., it has not declared there is no perfect-phylogeny) then
          declare there is a three-state perfect-phylogeny for M
           and use S to construct one, as detailed in the proof of Theorem 1.5.1.
       endif

    endwhile
```

Theorem 1.5.2 *The Dress-Steel Algorithm correctly determines, in polynomial time, if M has a three-state perfect-phylogeny and constructs one if there is one.*

Proof Consider any forced series of decisions that starts with the tentative selection of character $C_c(j)$, and makes a forced decision about some character $C_{c'}(i)$ for $c' \neq c$. In that series of decision, two of the characters $C_{c'}(1), C_{c'}(2), C_{c'}(3)$ will be selected for S and the other character excluded, so no further decisions about those characters will be made in the algorithm, unless the forced series ends with a problem. So if the algorithm does declare there is no perfect-phylogeny, then at least two of the characters $C_c(1), C_c(2), C_c(3)$ will be in S, for each character c in M.

Next, note that when a series of forced decisions is made, adding some characters to S, any character $C_{c'}(i)$ that was not excluded from S in that series must be compatible with all the characters that were selected for S. If that were not true, then $C_{c'}(i)$ would have been part of the forced series. So inductively, the characters that are selected for S are pairwise compatible. So, the algorithm is correct when it declares there is a perfect-phylogeny.

Finally, note that if the algorithm declares there is no perfect-phylogeny, then some character $C_c(j)$ was first tentatively selected for S and next excluded from S, and in both cases, forced decisions discovered a problem. Therefore, the algorithm is correct when it declares there is no perfect-phylogeny.

For the polynomial time bound, note that the time to implement any forced series of decisions is polynomial in m (in fact, it can be made linear in m), and there can be at most two forced series for each expanded character. So there can be at most $O(m)$ forced series of decisions. ■

1.5.2 Generalizations of the Four-Gamete and Splits-Equivalence theorems

The Four-Gamete Theorem (and equivalently the Splits-Equivalence Theorem) contains two, separable, mathematical facts about the existence of a perfect-phylogeny for a binary matrix M. One fact is that there is a perfect-phylogeny for M if and only if there is a perfect phylogeny for each *pair* of sites in M. A second fact is that there is a perfect-phylogeny for a pair of sites in M if and only if the rows of M do not contain all four binary combinations 0,0; 0,1, 1,0; 1,1 at that pair of sites. One or the other, or both, of these fact might be generalizable to multi-state perfect-phylogeny problems. In fact, for the case of three states, that generalization has been found [20, 21]. The generalization of the first fact is:

Theorem 1.5.3 *Let M be a matrix with up to three states per site. There is a 3-state perfect-phylogeny for M if and only if there is a 3-state perfect-phylogeny for each subset of three sites in M.*

Much earlier [10], Fitch established that this is the "tightest" possible generalization:

Theorem 1.5.4 *There is a matrix M, where every pair of sites in M has a 3-state perfect-phylogeny, but M does not have a 3-state perfect-phylogeny.*

The second fact has also been generalized [20, 21], in the case of three states:

Theorem 1.5.5 *A subset S of three sites in M has a 3-state perfect-phylogeny, if and only if $M(S)$ does not contain one of four specific patterns of data.*

We will not detail those four patterns here. The reader is refereed to [20, 21] for details.

Given these generalizations of the Four-Gametes Theorem to the case of three states, it is natural to conjecture what a generalization to k states would be. That question is still open, however, the following is known:

Theorem 1.5.6 *For any fixed k, there is a matrix M where every subset of $k-1$ sites has a k-state perfect-phylogeny, but M does not have a k-state perfect-phylogeny.*

This was first stated, and examples given for $k = 3, 4, 5$ in [23]. The result was more fully formalized and a full proof was given in [20, 21].

Bibliography

[1] R. Agarwala and D. Fernandez-Baca. A polynomial-time algorithm for the perfect phylogeny problem when the number of character states is fixed. *SIAM J. on Computing*, 23:1216–1224, 1994.

[2] R. Agarwala, D. Fernandez-Baca, and Giora Slutzki. Fast algorithms for inferring evolutionary trees. *J. of Comp. Biology*, 2:397–408, 1995.

[3] H. Bodlaender, M. Fellows, and T. Warnow. Two strikes against perfect phylogeny. *Proc. of the 19'th Inter. colloquium on Automata, Languages and Programming*, pages 273–283, 1992.

[4] A. Dress and M. Steel. Convex tree realizations of partitions. *Applied Math Letters*, 5:3–6, 1993.

[5] G. Estabrook, C. Johnson, and F. McMorris. An idealized concept of the true cladistic character. *Math Bioscience*, 23:263–272, 1975.

[6] G. Estabrook, C. Johnson, and F. McMorris. An algebraic analysis of cladistic characters. *Discrete Math*, 16:141–147, 1976.

[7] G. Estabrook, C. Johnson, and F. McMorris. A mathematical foundation for the analysis of cladistic character compatibility. *Math. Bioscience*, 29, 1976.

[8] J. Felsenstein. *Inferring Phylogenies*. Sinauer, Sunderland, MA., 2004.

[9] D. Fernandez-Baca. The perfect phylogeny problem. In D.Z. Du and X. Cheng, editors, *Steiner Trees in Industries*. Kluwer Academic Publishers, 2000.

[10] W. Fitch. Towards finding the tree of maximum parsimony. In G. F. Estabrook, editor, *Proceedings of the eighth international conference on numerical taxonomy*, pages 189–230. W. H. Freeman, 1975.

[11] M. Garey and D. Johnson. *Computers and intractability*. Freeman, San Francisco, 1979.

[12] D. Gusfield. Efficient algorithms for inferring evolutionary history. *Networks*, 21:19–28, 1991.

[13] D. Gusfield. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology.* Cambridge University Press, Cambridge, UK, 1997.

[14] D. Gusfield and Y. Wu. The three-state perfect phylogeny problem reduces to 2-SAT, 2009.

[15] J. Hein, M. Schierup, and C. Wiuf. *Gene Genealogies, Variation and Evolution: A primer in coalescent theory.* Oxford University Press, UK, 2005.

[16] D. Hinds, L. Stuve, G. Nilsen, E. Halperin, E. Eskin, D. Gallinger, K. Frazer, and D. Cox. Whole-genome patterns of common DNA variation in three human populations. *Science*, 307:1072–1079, 2005.

[17] Z. Jiang, P. Pevzner, and E. Eichler et al. Ancestral reconstruction of segmental duplications reveals punctuated cores of human genome evolution. Published online October 7, 2007.

[18] S. Kannan and T. Warnow. Inferring evolutionary history from DNA sequences. *SIAM J. on Computing*, 23:713–737, 1994.

[19] S. Kannan and T. Warnow. A fast algorithm for the computation and enumeration of perfect phylogenies when the number of character states is fixed. *SIAM J. on Computing*, 26:1749–1763, 1997.

[20] F. Lam, D. Gusfield, and S. Sridhar. Generalizing the four gamete condition and splits equivalence theorem: Perfect phylogeny on three state characters. In S.L. Salzberg and T. Warnow, editors, *Proc. of WABI 2009, Lecture Notes in Computer Science*, volume 5724, pages 206–219, 2009.

[21] F. Lam, D. Gusfield, and S. Sridhar. Generalizing the four gamete condition and splits equivalence theorem: Perfect phylogeny on three state characters. *SIAM J. on Discrete Math*, 25:1144–1175, 2011.

[22] S. Levi, G. Sutton, and J. C. Venter et al. The diploid genome sequence of an individual human. *PLOS Biology*, 5, 2007.

[23] C. A. Meacham. Theoretical and computational considerations of the compatibility of qualitative taxonomic characters. In J. Felsenstein, editor, *Numerical Taxonomy*, pages 304–314. Springer-Verlag Nato ASI series Vol. G1, 1983.

[24] M. Mutsuddi, D. Morriss, S. Waggoner, M. Daly, E. Scolnick, and P. Sklar. Analysis of high-resolution HapMap of DTNBP1 (Dysbindin) suggests no consistency between reported common variant associations and schizophrenia. *American J. of Human Genetics*, 79:903–909, 2006.

[25] J. Novembre, J. K. Pritchard, and G. Coop. Adaptive drool in the gene pool. *Nature Genetics*, 39:1188–1190, 2007.

[26] G. Perry and N. J. Dominy et al. Diet and evoution of human amylase gene copy number variation. *Nature Genetics*, 39:1256–1260, 2007.

[27] R. Redon and et al. Global variation in copy number in the human genome. *Nature*, 444:444–454, 2006.

[28] C. Semple and M. Steel. *Phylogenetics*. Oxford University Press, UK, 2003.

[29] A. J. Sharp, D. P. Locke, S. D.. McGrath, Z. Cheng, J. A. Bailey, R. U. Vallente, L. M. Pertz, R. A. Clark, S. Schwartz, R. Segraves, V. V. Oseroff, D. G. Albertson, D. Pinkel, and E. E. Eichler. Segmental duplications and copy-number variation in the human genome. *Amercian Journal of Human Genetics*, 77:78–88, 2005.

[30] C. J. Shaw and J. R. Lupski. Implications of human genome architecture for rearrangement-based disorders: the genomic basis of diseasee. *Human Molecular Genetics*, 13:R57–R64, 2004.

[31] M. Steel. The complexity of reconstructing trees from qualitative characters and subtrees. *J. of Classification*, 9:91–116, 1992.

[32] B. E. Stranger, A. C. Nica, and E. T. Dermitzakis. Populations genomics of humand gene expression. *Nature Genetics*, 39:1217–1224, 2007.

[33] K. Wong and R. deLeeuw et al. A comprehensive analysis of common copy-number variations in the human genome. *American J. of Human Genetics*, 80:91–104, 2007.