

Strings and Evolutionary Trees

The dominant view of the evolution of species is that all organisms are derived from some common ancestor and that a new species arises by a splitting of one population into two (or more) populations that do not cross-breed, rather than by a mixing of two populations into one. Therefore, the history of species, the high level history of life, is ideally organised and displayed as a *rooted, directed tree*. The extant species (and some of the extinct species) are represented at the leaves of the tree, each internal node represents a point when the history of two sets of species diverged (or represents a common ancestor of those species), the length and direction of each edge represents the passage of time or the evolutionary events that occur in that time, and so the path from the root to each leaf represents the evolutionary history of the species represented there. To quote Darwin [15]

... the great Tree of Life fills with its dead and broken branches the crust of the earth, and covers the surface with its ever-branching and beautiful ramifications.

This view of the history of life as a tree must frequently be modified when considering the evolution of viruses, or even bacteria or individual genes, but remains the dominant way that high-level evolution is viewed in current biology. Hundreds of papers are published yearly that depict deduced evolutionary trees. Several biological journals are primarily focused on such trees and the methods to obtain them.

Increasingly, the methods to create evolutionary trees are encoded into computer programs and the data used by these programs are based, directly or indirectly, on molecular sequence data. In this chapter we discuss some of the mathematical and algorithmic issues involved in evolutionary tree construction, and the interrelationship of tree construction methods to algorithms that operate on strings and sequences.

Biological controversies

In the world of biological systematics (or classification) and evolutionary biology, there are three major competing theories of what classification trees should mean, and how they should be constructed. The theories are *evolutionary taxonomy*, *phenetics* or *numerical taxonomy*, and *cladistics*. Debate between proponents of these three theories is intense, and often bewildering to people outside of evolutionary biology. While the technical side of tree building may appear to be a matter of pure graph theory and combinatorial optimization, the fundamental issues that determine the validity of these methods are sometimes discussed in terms more suited for religion. An insightful and (fairly) neutral account of these tree battles can be found in [50]; a mathematical treatment of some of these issues can be found in [28, 29, 31]. And of course, the larger debate on how to reconstruct evolutionary history goes beyond tree building, with the argument that some important aspects of molecular evolution are not tree-like at all (for example see [56]).

Tree building algorithms

Despite the debates on the biological foundations of trees and tree building, most tree building *algorithms* can be classified into two broad categories: *distance-based* methods and *maximum parsimony* methods¹.

In distance-based approaches, the input to the problem is evolutionary distance data (such as edit distance from sequences, or melting temperature from DNA hybridizations, or the strength of antibody cross reactions, etc.), and the goal is to reconstruct a weighted tree whose pairwise distances “agree” with the given evolutionary distances. When the distance data is *ultrametric*, the problem has an elegant solution that will be detailed in Section 1. When the data is not ultrametric but is *additive*, then the problem has an efficient solution, described in Section 4, based on the algorithm for ultrametric distances. But the real problems arise when the data is not additive. Then, one has to find a tree whose distances “best approximate” the given data. Different definitions of “approximate” have been suggested in the computational biology literature, and there are a variety of (usually heuristic) algorithms that build trees based on these different definitions. But on whole, the problem is still open as there is no approach that both leads to an efficient algorithm and that follows a completely accepted definition of a “good approximation”.

Maximum-parsimony methods take a different approach and do not reduce biological data to distances. Instead, parsimony methods are *character-based* methods that work directly on *character data*, very often using aligned sequence data. The goal is to build a tree, with the input taxa at the leaves and inferred taxa at the internal nodes, that *minimizes* the total amount or cost of the “mutations” implied by that evolutionary history. (*Minimizing mutations* is often described in the literature as *maximizing parsimony*.) When the input taxa are each represented by a molecular sequence, the maximum parsimony problem generalizes the phylogenetic alignment problem discussed in Chapter ?? . The parsimony problem seeks a leaf-labeled tree whose phylogenetic alignment has the minimum cost over all possible trees.

The seminal paper in algorithmic approaches to building phylogenetic trees is by Fitch and Margoliash [32]. The reader interested in the philosophy and mechanics of real tree building, as practiced in current molecular and evolutionary biology, should look at [50] for philosophy, at [55, 39] for mechanics, at [33] for a combination, and at [18] for insightful comments. Applications of these methods to specific phylogenetic and taxonomic studies are ubiquitous and are often reported in the popular press (for a recent example see [10]). For a detailed popular press articles (concerning the ancestry of the Giant Panda and of the Red Wolf), see [48, 58]. For illustrations from

¹Felsenstein [30] writes “Two computational methods have dominated the reconstruction of molecular phylogenies: parsimony and distance. The parsimony method finds the evolutionary tree that requires the fewest changes to nucleotides to explain evolution of the observed sequences. Distance methods compute a table of pairwise numbers of differences between sequences and try to fit this to expected pairwise distances computed from the tree”.

representative research-level papers, see [36], [47] and [19].

String and tree algorithms

Even from this brief description of distance and parsimony methods it should be clear that string algorithms of the type considered in this book play an important role in the ultimate success or failure of methods for evolutionary inference. This is immediate in the case of maximum parsimony methods that operate directly on raw sequences or on characters derived from sequences (possibly multiply aligned). It is less immediate in the case of a distance method, since those methods operate on numbers. However, often the *input* to a distance based method is the *output* from some particular string algorithm that finds pairwise distances or computes multiple alignments². So the model of distance or multiple alignment that is embedded in the algorithm (global, local, gap type, weighted, unweighted, sum-of-pairs, consensus, variable parameter settings, etc.), and the quality of the algorithmic solution, can have a significant impact on the reliability of the final evolutionary tree.

In the next sections we discuss several *idealized* versions of distance-based and maximum-parsimony tree building problems. The discussion focuses on the “elegant and the proven” rather than the “realistic and the practical”, and the results may seem too abstracted from reality to be of *direct* practical use. Moreover, we concentrate on combinatorial aspects of tree building, even though statistical issues have had a very large practical and theoretical role in the field [28, 29, 31]. Still, the problems and results we discuss serve to introduce the field, and cleanly expose the underlying ideal-world models that motivate and drive most practical tree building methods.

1 Ultrametric trees and ultrametric distances

1.1 Introduction

We begin by discussing *ultrametric trees and distances*, constructs that can be used in evolutionary reconstruction when the data “perfectly fits” certain strong assumptions. Even when the data is not perfect, ultrametric trees arise implicitly in many numerical-based tree reconstruction methods. Ultrametric trees, or approximations of them, can be used to deduce *both* the branching patterns of evolutionary history and some measure of the time that has passed along each branch.

Definition Let D be a symmetric n by n matrix of real numbers. An *ultrametric tree* for D is a rooted tree T with the following properties:

1. T contains n leaves, each labeled by a unique row of D .

²In some reconstruction studies, the pairwise distances used are the distances of induced pairwise alignments taken from a *multiple* alignment of all the sequences.

2. Each internal node of T is labeled by one entry from D , and has at least two children.
3. Along any path from the root to a leaf, the numbers labeling internal nodes *strictly decrease*.
4. For any two leaves i, j of T , $D(i, j)$ is the label of the least common ancestor of i and j in T .

Thus, an ultrametric tree for D (if there is one) is a compact representation of the matrix D . For an example, see Figure 1.

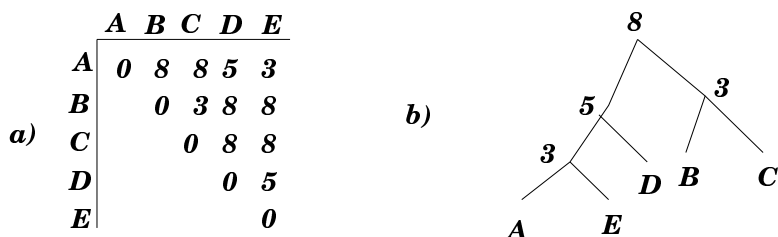


Figure 1: Figure a) shows a symmetric matrix D . Figure b) shows an ultrametric tree for matrix D .

Definition A *min-ultrametric tree* for D is a rooted tree T with all the properties of an ultrametric tree, except that property 3 is changed to the following: In a min-ultrametric tree, along any path from the root to a leaf, the labels of the internal nodes must *strictly increase*.

There is no accepted term for what we have defined as “min-ultrametric”, and this deficiency is sometimes a source of confusion. Note that not every matrix D necessarily has an ultrametric or a min-ultrametric tree representing it. An ultrametric tree, or a min-ultrametric tree, has at most $n - 1$ internal nodes, so if D has more than $n - 1$ distinct values, then there cannot be an ultrametric or a min-ultrametric tree for D .

The main mathematical and algorithmic problems are to characterize the conditions under which a matrix D can be represented by an ultrametric or a min-ultrametric tree, and to develop an efficient algorithm for building these trees when possible. We will examine both of these problems in detail after relating ultrametric trees to evolutionary trees.

1.2 Evolutionary trees as ultrametric trees

If the true evolutionary history of n taxa forms a rooted directed tree T , with the extant taxa at the leaves of the tree, then each internal node v of T represents a *divergence event* or a historical branching. A divergence event is a point in time when

the evolutionary histories of (at least) two taxa, say i and j , diverge. For simplicity, we will say that v is the point when “ i and j diverge”. However, this shorthand does not suggest that i is an ancestor of j , or j is an ancestor of i , or both are descendants of some other (possibly extinct) taxa. It simply says that before point v , i and j shared a common evolutionary history.

Suppose that in addition to knowing the topology of the tree (or *branching order* in the lexicon of *cladistics*), one knows the time (absolute or relative) that each divergence event occurred. Written at the nodes of the tree, those times must strictly *increase* along any path from the root. Moreover, if node v is the least common ancestor of leaves i and j in T , then the label on node v is the time when i and j diverged. Thus T is a *min-ultrametric* tree for the n by n matrix D where, for each pair of leaves i and j , $D(i, j)$ is the time that i and j diverged. So a true evolutionary history (a directed tree plus times of divergence) forms a min-ultrametric tree for pairwise divergence-time data.

Equivalently, if we know the true evolutionary history, we can label each node v in T by the time that has passed *since* the divergence event represented by v . With that labeling, the numbers strictly *decrease* along any path from the root. Thus if $D(i, j)$ now represents the time *since* i and j diverged, then this second labeling of T makes T an ultrametric tree for D . As we will see below, real biological data more usually approximates *time since* divergence rather than *time of* divergence, so it is more natural to concentrate on ultrametric trees.

Now the real problem is that we don’t know true evolutionary history, either trees or divergence times. Rather, we want to infer a plausible history from data reflecting *time since* divergence. Knowing that a true evolutionary history should form an ultrametric tree, the goal is to construct ultrametric trees from the time-since-divergence data. But that raises an immediate question: Even if there is an ultrametric tree T for a matrix D of divergence data, how confident can we be that T actually captures the true evolutionary history we seek, or even captures some of it? We will address that question, as well as the original algorithmic question of how to build ultrametric trees, in the next section. After that, we will return to the issue of how to find biological data use in building ultrametric trees.

1.3 How to test for an ultrametric tree

In this section we state and prove the major theorem concerning ultrametric trees and distances, and develop an efficient algorithm to build ultrametric trees when possible.

Definition A symmetric matrix D of real numbers defines an *ultrametric distance* if and only if for *every three* indices i, j and k , there is a tie for the *maximum* of $D(i, j), D(i, k)$ and $D(j, k)$. That is, the maximum of those three numbers is *not unique*. Similarly, D defines a *min-ultrametric distance* if and only if for each triple i, j and k , there is a tie for the *minimum* of $D(i, j), D(i, k)$ and $D(j, k)$.

When D defines an ultrametric distance, we will often say that D is an ultrametric *matrix*. Similarly, we may refer to D as a min- ultrametric matrix.

It is easy to see that if D has an ultrametric (or min-ultrametric) tree, then D is an ultrametric (or min-ultrametric) matrix. See Figure 2 for more explanation. The converse statement is less obvious. We next show that it is also true.

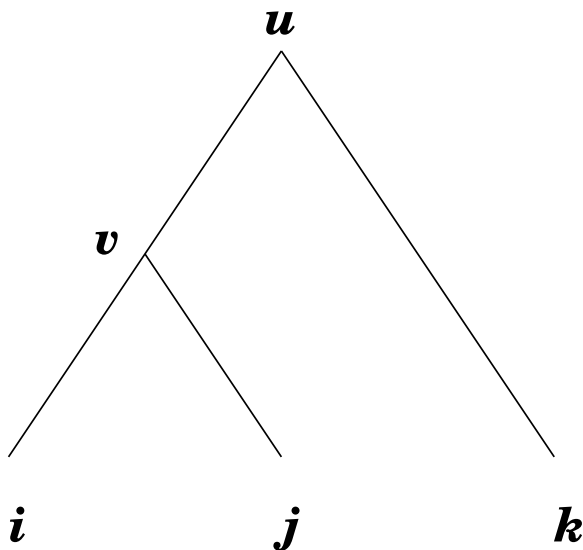


Figure 2: Suppose that D has an ultrametric tree. This figure shows the generic subtree containing leaves i, j, k . The subtree will generally contain other nodes that are not shown. Node v is the least common ancestor of i and j , and node u is the least common ancestor of the three leaves i, j and k . Since this is an ultrametric tree for D , the number written at u must be strictly larger than the number at v . By definition, the number at v is $D(i, j)$ and the number at u is $D(i, k) = D(j, k)$. Hence these three numbers satisfy the condition that the maximum of $D(i, j), D(i, k)$ and $D(j, k)$ is not unique. Indices i, j, k are arbitrary, so the picture is general and shows that if D has an ultrametric tree then D is an ultrametric matrix.

Theorem 1.1 *A symmetric matrix D has an ultrametric tree (or a min-ultrametric tree) if and only if D is an ultrametric (or min-ultrametric) matrix.*

Additional consequences

The proof of Theorem 1.1 immediately establishes several important facts.

Theorem 1.2 *If D is an ultrametric matrix, then the ultrametric tree for D is unique.*

The fact that the ultrametric tree for an ultrametric matrix is unique answers the question posed earlier in Section 1.2: How informative is an ultrametric tree derived

from D ? That is, we are trying to deduce an unknown evolutionary tree T that is thought to have given rise to the observed data in D . How does the tree derived from D relate to the unknown T ? The answer is that *if* the data in D is truly proportional to the node numbers labeling the unknown evolutionary tree T , then the ultrametric tree derived from D *must be* T . That is a very strong, and desirable, property.

1.4 How are ultrametric data obtained?

Theorem 1.2 implies that ultrametric time-since-divergence data is very powerful in reconstructing evolutionary history, both the tree topology and relative divergence times. But what hope is there that ultrametric data can be obtained in practice?

The Molecular Clock Theory

The *molecular clock theory*, proposed in the early 1960's by Emile Zuckerkandl and Linus Pauling [59, 45], states that for any given protein, *accepted mutations* in the amino acid sequence for the protein (and in the underlying DNA) occur at a constant rate. The term *accepted* here means those mutations that still allow the protein to function properly (i.e., that are not lethal to the protein). The implication is that the number of accepted mutations occurring in any time interval is *proportional* to the length of that interval. Hence once the clock has been calibrated, the length of an unknown interval can be measured by the number of accepted mutations that occur in that interval. Of course, this theory is affected somewhat by granularity, and is only asserted to hold over “long-enough” time intervals. In addition, the rate of *accepted* mutations is different for different proteins, i.e., they have different clocks. For example, *hemoglobin* mutates faster than *cytochrome c*, but both are relatively stable (and are very similar in all mammals) compared to other proteins such as *fibrinopeptides* [4]. In fact, different parts of a protein may evolve at different rates [18], so care must be taken when invoking the clock. Underlying the molecular clock theory is the assumption that mutations (*accepted or not*) in all DNA occur at some constant rate. Differences in rates of *accepted* mutations result from differences in how constrained particular proteins are (by natural selection at the organism level or by physical chemistry at the molecular level).

A great deal of ink (and blood) has been spilt over the molecular clock, and the theory is intimately tied to another major controversy in evolutionary biology – the *neutral theory of molecular evolution*, developed by Motoo Kimura [44]. That theory concerns the relative importance of natural selection verses random genetic drift in evolution. For two very readable discussions on the neutral theory, one by Kimura and one by Stephen Jay Gould, see [43] and [34].

It should be clear how the molecular clock simplifies the task of collecting ultrametric data. Let A and B be two taxa that both make and utilize the same protein (for example hemoglobin). Suppose one can determine that k accepted mutations

have occurred in the DNA or amino acid sequences for the hemoglobins of A and B since the time that A and B diverged. (By the molecular clock theory, it follows that roughly $k/2$ separate accepted mutations occurred in each of the two evolutionary histories since their point of divergence.) *If* the number of accepted mutations (or some number proportional to it) can be deduced in this manner for each pair of n taxa being studied, then those $\binom{n}{2}$ numbers are related proportionally to the respective times since divergence of those pairs of taxa. Those $\binom{n}{2}$ numbers then perfectly satisfy the requirements for an ultrametric tree, and by Theorem 1.2, that tree recaptures the *true* evolutionary history of the taxa. So the molecular clock theory reduces the problem of finding time-since-divergence data to the problem of finding the number of mutations since divergence. That reduction is very appealing but how is mutation data obtained?

1.4.1 Laboratory-based methods

The first methods used to estimate the number of accepted mutations between two taxa generally used physical/chemical means. One approach takes the DNA of the two taxa, denatures it (by heating) so that the double strands separate, mixes the single strands from the two DNA sources together to allow them to hybridize, and then checks the temperature at which the hybrid strands separate. The idea is that the more similar the two DNA sources are, the stronger the hybridization and the higher the temperature must be in order to separate the two strands. The separating temperature of two taxa A and B is then assumed (after some adjustments) to be proportional to the total number of accepted mutations that have occurred in the time since A and B diverged.

The *highly* simplified description of an experimental procedure is essentially what was done in the classic, massive study of bird evolution [53, 52] by Sibley and Ahlquist. Moreover, the Sibley and Ahlquist data had a strong “self-verifying” property – the data was nearly ultrametric³. In turn, the ultrametric nature of their data gave a strong boost to the molecular clock theory since ultrametric data is hard to explain without accepting a molecular clock and also believing that the obtained data correctly reflects true evolutionary history⁴.

³In the language of evolutionary biology, the test to see if the data is ultrametric is called the *relative rate test* suggested by V. Sarich and A. Wilson. Ahlquist and Sibley write in [52]: “We have found that the DNA clock seems to tick at the same average rate in all lineages of birds. The evidence comes from a procedure known as the relative rate test ... The relative rate test compares any three species ...”.

⁴We should note however, that the chemistry behind the work of Sibley and Ahlquist has been severely questioned [51], along with some of their record keeping, and it is no longer clear how reliable their results are. A more current view of bird evolution appears in [27].

1.4.2 Sequence-based methods

More current methods for estimating the numbers of accepted mutations are based directly on DNA or amino-acid *sequences*. For two taxa A and B , one estimates the number of accepted mutations that have occurred since their divergence by examining differences in the DNA or amino acid sequences coding for proteins common to both A and B . Usually (see [18]) this estimate is related to *edit distance*. (Big surprise? – Of course, it had to be edit distance or some related concept to make this topic appropriate for a book on string algorithms.)

In actuality, edit distance has to be adjusted because a given nucleotide position might mutated several times in the course of evolutionary history. Also, certain DNA mutations (particularly in the third nucleotide of a codon) do not change the amino acid specified by the DNA. But after adjusting for these factors (see [18]), the edit distance between existing DNA or protein sequences for A and B is used to estimate the total number of accepted mutations that have occurred since the divergence of A and B . Those pairwise edit distance numbers form a matrix D , and if D is ultrametric then its tree is used as a hypothesis for the true evolutionary history of the taxa.

To recap, under the molecular clock theory, edit distance can be used to estimate the number of accepted mutations that have occurred since the time any two taxa diverged. That number should then be proportional to the *actual time* since their divergence. Hence, given a set of n taxa and the $\binom{n}{2}$ edit distances between the pairs of taxa, an ultrametric tree (if one exists) for the data should give a possible evolutionary history explaining the data. Moreover, when an ultrametric tree exists for D , it is the *only* ultrametric tree that exists for D . The conclusion is that *if* pairwise edit distances can be used to obtain numbers that are proportional to the true pairwise time-since-divergence data (as the molecular clock theory suggests), then the unique ultrametric tree for that data should be *the true* evolutionary tree with the correct topology and proportionally-correct edge lengths. And, if the absolute time of divergence is known for any pair (say from fossil data) then the molecular clock can be “set” and proportionally-correct times can be converted to correct *absolute* times. This is the ideal case.

1.4.3 Final comments

Most often, real data is not ultrametric and even ultrametric data does not necessarily reflect true times since divergence. Still, being ultrametric is a property of data that is not likely to occur by chance, so when data is ultrametric, or nearly so, it is taken as strong evidence for the molecular clock theory and as strong evidence that the data does capture true evolutionary history.

In response to data that is not ultrametric, one might consider the problem of perturbing the data by “the smallest” amount possible so that the resulting data is ultrametric. If the perturbations are not too large, then the data may still support the molecular clock theory. Several variants of this perturbation problem have been

studied. Consider the problem when the perturbed data is required to be ultrametric and the perturbations can only *decrease* the initial values. Then there is a solution to the problem where each value is (simultaneously) decreased by as little as possible so that the end result is ultrametric. This will be explored in the exercises. When the changes can be both positive as well as negative, then there is an efficient algorithm that creates ultrametric data and minimizes the maximum change to any single initial value. If only increases are allowed, then the problem is NP-hard. The last two results are due to Farach, Kannan and Warnow [26].

2 Additive-distance trees

2.1 Introduction

Ultrametric data is the holy grail of phylogenetic reconstruction – when time-since-divergence data is ultrametric, the belief is that the true evolutionary history can be reconstructed. But this is mostly an idealized abstraction, and real data is rarely ultrametric. What can be done in this case? A weaker requirement on evolutionary-distance data is that it be *additive*.

Definition Let D be a symmetric n by n matrix where the numbers on the diagonal are all zero and the off-diagonal numbers are all strictly positive. Let T be an edge weighted tree with at least n nodes, where n distinct nodes of T are labeled with the rows of D . Tree T is called an *additive tree* for matrix D if, for every pair of *labeled* nodes (i, j) , the path from node i to node j has total weight (or distance) exactly $D(i, j)$.

That is, T encodes the matrix D in that every entry $D(i, j)$ equals the distance in T from node i to node j . For example see Figure 3.

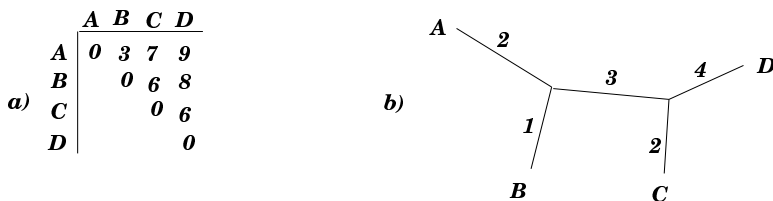


Figure 3: Figure a) shows a symmetric matrix D . Figure b) shows an additive tree for matrix D . Note that T contains some unlabeled nodes.

Additive tree problem Given a symmetric matrix D with diagonal entries equal to zero and other entries strictly positive, find an additive tree for D , or determine that none exists.

When the data in D reflects some evolutionary distance between pairs of taxa, such as weighted edit distance or other estimates of time since divergence, an additive

tree for D gives one possible reconstruction of evolutionary history. The tree shows a branching pattern and gives edge distances (reflecting time) consistent with the data in D . But since the tree is undirected, it doesn't indicate ancestral relations or the direction of evolution. Those have to be deduced by separate means.

It is easy to establish that if D is ultrametric and $D(i, i) = 0$ for each i , then D is also additive. In fact, such an ultrametric matrix D can be characterized as follows: D is ultrametric if there is an additive tree T for D and a node v in T such that all leaves in T have the same distance to v . However, it is not true that when D is additive it must be ultrametric. Thus the condition that distance data be additive is a weaker requirement for evolutionary validity than the requirement that it be ultrametric. Still, the additive tree problem, like the ultrametric problem, is a very idealized problem since data is rarely additive, either due to small errors in the data or large problems in the evolutionary model. One large problem is that evolution (of plants perhaps, and bacteria certainly) is not always divergent, i.e., tree-like. Genetic material can merge through *horizontal transfer* (see [56] for example), causing evolutionary histories to merge rather than diverge.

When data is not perfectly additive but the tree model is still valid, then one looks for a tree whose pairwise distances deviate from the distances in D as “little as possible”. This is a large area of research with different suggested definitions of deviation, and many suggested (heuristic) algorithms. For a practical treatment of this problem see [55]. For a theoretical method with a *provable* analysis, see [26, 1]. For NP- completeness results see [16].

2.2 Algorithms for the additive tree problem

There are several $O(n^2)$ -time algorithms in the literature that solve the problem of reconstructing, if possible, an additive tree from an n by n matrix. Some examples are [14, 6, 57, 38]. Before these, the first polynomial time solution appears to be in [11, 13], where the basic “four-point condition” was proved (see Exercise ?? in this chapter). That condition implies a direct $O(n^4)$ -time algorithm for the problem. Interest in additive tree problems has arisen in many diverse fields, and many results have been rediscovered several times. A history and bibliography of much of this work appears in [7] which states: “The works of various authors were remarkably little known one to another. Thus, for example, the four-point condition was published at least five times!” In fact, we now know of ten papers that give polynomial time solutions to the additive tree problem.

Many of the published $O(n^2)$ -time algorithms for the additive tree problem are similar and involve some amount of linear equation solving that obscures the true *combinatorial* nature of the additive tree problem. So we will not present any of those algorithms here. Instead, we will defer the algorithmic question until Section 4, where several evolutionary tree problems will be reduced to ultrametric tree problems. We will show there that the additive tree problem can be reduced in $O(n^2)$ time to

an ultrametric tree problem, so that the given (combinatorial) algorithm for building an ultrametric tree can be adapted to building an additive tree. In this way, we solve the additive tree problem using only the *sorted order* of certain numbers derived from matrix D , without equation solving.

3 Parsimony: character-based evolutionary reconstruction

3.1 Introduction

In this section we introduce a different, *character-based* approach to reconstructing evolutionary history. In this approach, the input is a set of *attributes* called *characters* that the objects may possess⁵. If the input characters are chosen well, then the distribution of the attributes among the objects may be used to deduce partial evolutionary history in the form of an evolutionary tree. This tree provides the branching order of the history, but does not by itself establish the times of divergence events. Alternatively, character attributes can be used to form a *taxonomy* (a systematic classification) of the objects without any suggested historical importance. After discussing character-based reconstruction problems, we will relate this approach to ultrametric trees.

The algorithmic study of character-based reconstruction has been very active in recent years, and there are many results that could be discussed here. However, the intent is only to *introduce* character-based reconstruction. We therefore will only discuss a few very idealized and simplified versions of character-based (or maximum-parsimony) problems. We will discuss in detail only *binary-character* problems, where object either do or do not have any particular character, although generalizations will be mentioned. The major focus will be on the (binary) *perfect phylogeny* problem which is a special case of the general maximum-parsimony problem.

Definition Let M be an n by m , 0-1 (binary) matrix representing n objects in terms of m characters or traits that describe the objects. Each character takes on one of two possible *states*, 0 or 1, and cell (p, i) of M has a value of one if and only if object p has character i .

Definition Given an n by m binary-character matrix M for n objects, a *phylogenetic tree for M* is a rooted tree T with exactly n leaves that obeys the following properties:

1. Each of the n objects labels exactly one leaf of T .
2. Each of the m characters labels *exactly one* edge of T .

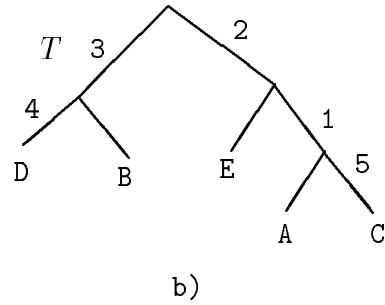
⁵Note that the word "character" in this context does not refer to a member of an alphabet, but to an attribute or a trait of an object.

- For any object p , the characters that label the edges along the unique path from the root to leaf p specify all of the characters of p whose state is one.

In the example in Figure 4, the first matrix M has a phylogenetic tree T , but the second matrix M' does not.

	1	2	3	4	5
A	1	1	0	0	0
B	0	0	1	0	0
M C	1	1	0	0	1
D	0	0	1	1	0
E	0	1	0	0	0

a)



	1	2	3	4	5
A	1	1	0	0	0
B	0	0	1	0	1
M' C	1	1	0	0	1
D	0	0	1	1	0
E	0	1	0	0	1

c)

Figure 4: Matrix M shown in a) has a phylogenetic tree T shown in b). However, matrix M' shown in c) has no phylogenetic tree.

The interpretation of a phylogenetic tree for M is that it gives an estimate of the evolutionary history of the objects (in terms of branching pattern, but not time), based on the following biological assumptions:

- The root of the tree represents an ancestral object which has none of the present m characters. That is, the state of each character is zero in the ancestral object.
- Each of the characters changes from the zero state to the one state *exactly* once, and never from the one state to the zero state.

The key feature of a phylogenetic tree (without which there would be no interesting problem) is that each character labels *exactly one edge* of the tree. This corresponds to

the second biological assumption above, and represents the point in the evolutionary history of the objects when that character changes from its zero state to its one state. Hence any objects below that edge definitely have that character.

Finding a set of characters that obey the assumptions is often a difficult (and controversial) task in evolutionary research. So it is worth discussing the issue of where character data actually comes from.

3.2 Where do character data come from?

In the case of biological objects (generally species), the characters used have traditionally been *morphological* features or traits of the objects. These morphological characters may be gross features such as “possessing a backbone” or may be very fine features only understood by specialists studying those organisms. But characters can also be based on DNA or protein sequences possessed by the different species. For example, whether or not the amino-acid sequence for a given protein contains a specific substring makes a perfectly good binary character.

Characters based on morphology can be problematic because, under selection pressures, similar morphological features could arise independently, or a feature could be gained and lost and then regained. For example, it is thought that wings have evolved independently several times. Behavioral traits are equally problematic. For example, the character “walks on knuckles” is a trait possessed by both chimpanzees and gorillas but not (generally) by humans⁶. Therefore, if this feature is used to build a phylogenetic tree, chimpanzees and gorillas should be found in a subtree that does not contain humans. But, contrary to that reasoning, current thinking has humans and chimpanzees branching together *away* from gorillas.

Non-functional substrings in DNA generally make better characters for character-based evolutionary reconstruction than do morphological or behavioral traits. Substrings of DNA outside of protein coding regions (genes) and regulatory regions mutate semi-randomly, so that a long randomly mutated substring in the DNA of one species is very unlikely to be found in the DNA of (evolutionarily) distant species. But if the mutation rate is slow enough for the span of history being studied, the random substring (or vestiges of it) may be recognized in more closely related species. Thus character-based evolutionary reconstruction from certain stretches of DNA should be increasingly important as more genomic DNA sequences are obtained.

One increasingly important type of character used in evolutionary studies is the specific nucleotide in a fixed position of a DNA sequence. Generally, analogous DNA sequences from a number of species are first multiply aligned. Each column of that alignment then specifies a single character which can take on one of four states A, T, C or G . The state of character i , for any given species p , is the nucleotide in position p, i of the multiple alignment. In this case the characters are not binary, but DNA sequences are sometimes used as binary characters by grouping A with G (the purines)

⁶Thanks to Gary Churchill for this example.

and grouping C with T (the pyrimidines). Similarly, multiply aligned amino-acid sequences are sometimes used as characters having twenty states (or fewer, if characters are grouped).

Another type of *binary* character of potential importance is based on whether or not the expression of a particular protein is *regulated* by another particular protein. (For introductory discussions on regulation and three common structures in regulatory proteins – zinc fingers, helix-turn-helix motifs, and leucine zippers – see [49, 8].) It is increasingly understood that regulation and control of protein expression (when and in what cells a gene for a particular protein is expressed) is as important (or maybe more important) in species differentiation than differences in the protein sequences. For example, mice and humans make essentially the same complement of proteins, and for many particular proteins, there is little difference in the amino acid sequences between the human and the mouse analogs of the protein⁷. What makes a mouse different from a human may be determined more by differences in protein regulation (which is reflected in DNA sequences outside of protein coding regions) than by differences in the corresponding protein sequences. Recent work [37] supports this view, as reflected in a review of that work [46]: “genes can gain new developmental functions not just through alterations in their protein sequences but also by acquiring new regulatory sequences that alter where and when they are expressed.” Evolutionary history therefore may be better understood by examining changes in protein regulation (which proteins act to enhance or repress the expression of which other proteins) than by examining changes in amino acid sequence. Moreover, the character “protein A acts to enhance (or repress) the expression of protein B” is a *binary* character, and hence such binary characters may be of increasing utility as more regulatory data is obtained.

An excellent example of an evolutionary tree based on purely on binary characters appears in [40]. There, an early history of bird radiation is deduced from characters such as “feathers” or “loss of postorbital bone”.

Finally we mention an implicit binary character used [5] to tackle the controversial relationship of fungi, plants and animals. The question is, are fungi more closely related to plants or animals? In the approach of [5], a multiple alignment of certain protein sequences from fungi, plants and animals was first computed. This multiple alignment contained gaps, and many of the aligned strings shared the same gap(s). The width, regularity and frequency of the shared gaps was such that each gap could be treated as a binary character, although that language is not used in [5]. Relative to the multiple alignment, each string either contained or didn’t contain a specific gap. These “binary characters” were then used (implicitly) in [5] to build an evolutionary tree which placed fungi closer to animals than to plants.

Similarly, one can identify discrete gaps in the alignment of the RNA sequences represented in Figure ?? on page ?? (and in the more complete figure in [17]). These

⁷I’ve heard one biologist refer to mice as “furry, little people”.

gaps can be used as binary characters to construct an evolutionary tree of the viruses containing those sequences. Given these examples of binary characters, we now turn to the algorithmic question of how to build evolutionary trees from binary characters.

3.3 Perfect Phylogeny

Perfect Phylogeny Problem: Given the n by m , 0-1 matrix M , determine whether there is a phylogenetic tree for M , and if so, build one.

3.4 Generalized perfect phylogeny

We have discussed the perfect phylogeny problem in detail when each character can take on two states. *The generalized phylogeny problem* allows a character to take on more than two states. In that case, a perfect phylogeny for M is a directed tree T where each object labels exactly one leaf of T as before, but now edges are labeled with *character-state transitions*. That is, the label applied to an edge is an *ordered triple* (c, x, y) indicating that character c changes from state x to state y along that edge.

As in the binary case, we specify the starting state for each character at the root node, and require that the path from the root to a leaf labeled p must describe the character states of object p : the ending states specified by the changes on the path to p must correctly specify the character states of object p . The critical constraint now is, for any state y of any character c , there can be at most one edge where the state of character c changes to y . That is, there can be at most one edge labeled with a triple that begins with c and that ends with y .

The definition of phylogenetic tree in the case of binary characters is easily seen as a special case of this more general definition. In the binary case, a character label c on an edge is just an abbreviation for the triple $(c, 0, 1)$. We should note that the description of a generalized perfect phylogeny given here is somewhat non-standard but equivalent to the more commonly used definitions found, for example, in [41]. The definition here is used to make the continuity between the binary and the general case more explicit.

The generalized perfect phylogeny problem Given a character matrix M where each character may take on up to r states, determine if there is a perfect phylogeny for M , and construct one if so.

It is beyond the scope of this book to discuss the generalized problem in any detail. It was first proposed in a paper by [12] in 1974 and was open for almost twenty years. Sampath Kannan and Tandy Warnow [41] first established that the problem has a polynomial time solution in terms of n and m , if r is *fixed* at three or four. Independently, Dress and Steele [20] gave a polynomial time solution for r fixed at three. However, if r is variable (specified in the input to the problem) then the generalized perfect phylogeny problem is NP-complete [9, 54]. Richa Agarwalla

and David Fernandez- Baca [2] subsequently established that if r is *any fixed* number, then the generalized perfect phylogeny problem can be solved in polynomial time in terms of n, m . Of course, r appears in the exponent of the worst-case time bound. More recently, Kannan and Warnow [42] simplified that solution and considered a way to represent all the solutions to any given problem instance. That approach may be very valuable for computing statistical estimates of the “significance” of any given tree or of any edge in the tree.

4 The centrality of the ultrametric problem

Although the four tree problems we have considered in detail – ultrametric, additive, (binary) perfect phylogeny, and tree compatibility – are quite different in appearance, they are actually strongly related. In fact, all of these problems are solvable as ultrametric tree problems. We have already seen that the tree compatibility problem reduces to the (binary) perfect phylogeny problem. So the main work in this section is to show how to reduce the additive tree problem and the perfect phylogeny problem to ultrametric tree problems. In the case of the additive tree problem, this reduction will also give an efficient algorithm for solving the additive tree problem.

4.1 The additive tree problem viewed as an ultrametric problem

We will show how to reduce the additive tree problem to an ultrametric problem in $O(n^2)$ time, by creating a matrix D' that is ultrametric if and only if matrix D is additive. To introduce the idea of the reduction, assume that D is additive and that an additive tree T for D is *known*. Also, assume without loss of generality that each of the n taxa in D labels a *leaf* of T .⁸ We will label the nodes of T with particular numbers to create an ultrametric tree, and this tree will expose the idea of the desired reduction.

Let v be the row of D which contains the largest entry in D , and let m_v be that maximum entry. Hence over all nodes in T , node v has the maximum distance to any leaf in T . Now root T at node v creating a directed tree. We want to “stretch” leaf edges (edges which end at a leaf) so that v is equidistant to each leaf in the resulting tree. To do that for each leaf i , simply add $m_v - D(v, i)$ to the distance on the edge in T into leaf i . The result is a rooted, edge-weighted tree T' where the distance from v to any leaf is exactly m_v , and where each internal node is equidistant to any leaf in its subtree. (See Figure 5 a), b)). Note that only distances on leaf edges were changed. Now label each node of T' with the (unique) distance from it to any of the leaves in its subtree. Those labels are non-increasing, and can therefore be used to define

⁸This requires that we relax the *strictness* condition in the definition of an ultrametric, allowing two adjacent nodes on a path from the root to have equal node labels.

an ultrametric matrix D' , where $D'(i, j)$ is the label at the least common ancestor of leaves i and j in T' (see Figure 5 c).

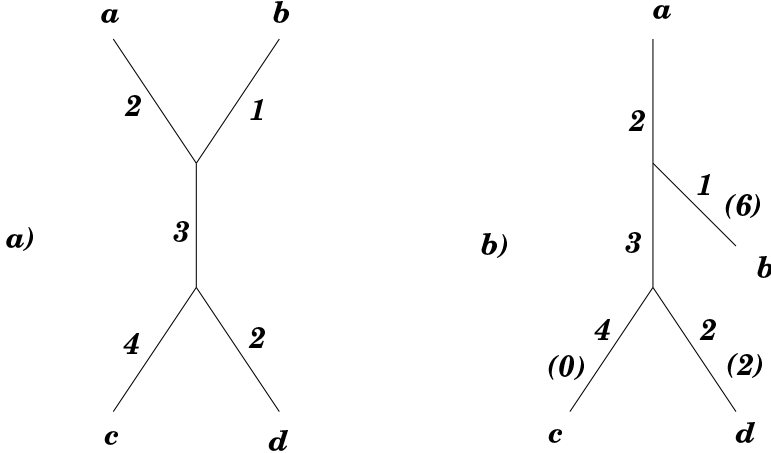


Figure 5: Figure a) shows an edge-weighted tree with four labeled leaves. Node v is node a , and m_v is nine. Figure b) shows the tree rooted at node a ; the numbers in parentheses are the values $m_a - D(a, i)$ for each leaf i which are added to the weight of each leaf edge. Then each leaf is at distance nine from the root. Also, each internal node is equidistant from each leaf in its subtree. Figure c) shows the ultrametric tree derived from those distances.

How the reduction works on the matrix

In the above exposition, matrix D' depends on T' , which depends on T which comes from D , and this defines the reduction from additive matrix D to ultrametric matrix D' . But we want a reduction from D to D' directly without having to build T or T' . To see the idea of a direct reduction, consider two leaves i and j of T and let w be their least common ancestor. We can deduce their (equal) distance to w in T' , without explicitly knowing T' .

Let x be the distance in T from the root v to node w , and let y be the distance from node w to leaf i . The distance from w to i (or j) in T' is exactly $y + m_v - D(v, i)$. But what is y ? Distance y is exactly $D(v, i) - x$, and $2x$ is easily seen, from Figure 6, to be $D(v, i) + D(v, j) - D(i, j)$. Similar reasoning can be used to obtain the distance from w to j . Hence,

Lemma 4.1 *Without knowing T or T' explicitly, we can deduce that $D'(i, j) = m_v + (D(i, j) - D(v, i) - D(v, j))/2$.*

Given Lemma 4.1, we have

Theorem 4.1 *If D is an additive matrix, then D' is an ultrametric matrix, where $D'(i, j) = m_v + (D(i, j) - D(v, i) - D(v, j))/2$.*

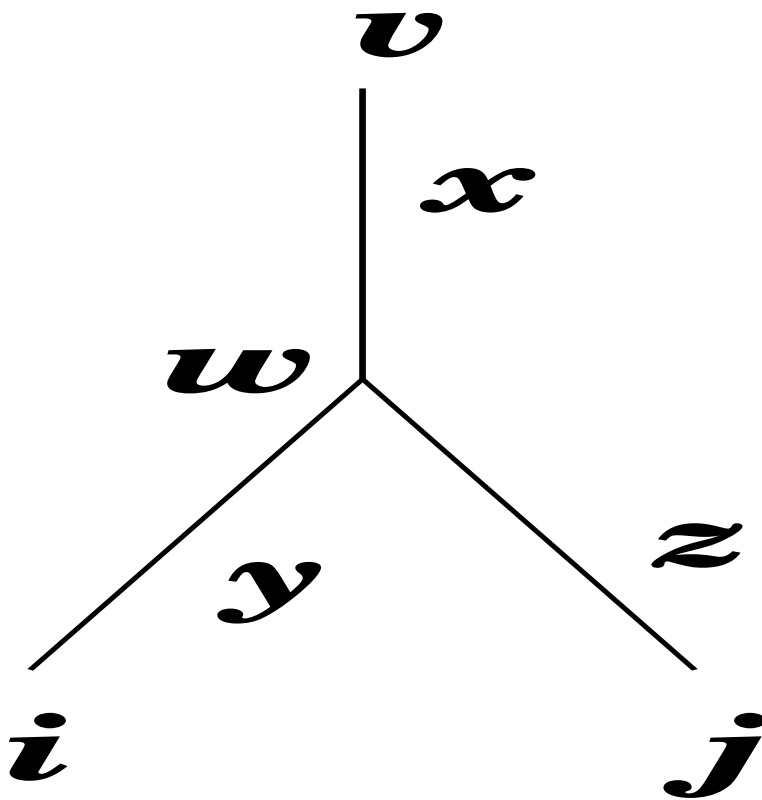


Figure 6: A schematic of distances in tree T . The distance from the root v to w is x , and the distance from w to leaf i (or j) is y (or z). Therefore, $D(v, i) = x + y$, $D(v, j) = x + z$, and $D(i, j) = y + z$. It follows that $2x = (x + y) + (x + z) - (y + z) = D(v, i) + D(v, j) - D(i, j)$.

Proof The proof requires assembling the pieces established above. $D'(i, j) = y + m_v - D(v, i)$, and $y = D(v, i) - x$, and $x = D(v, i) + D(v, j) - D(i, j)$. Substituting and cancelling equal terms establishes the theorem. \square

Hence, if we are given a matrix D and want to establish whether D is additive, we can create matrix D' and test if D' is ultrametric. If not, then D is not additive. But what of the converse?

Theorem 4.2 *If matrix D' is ultrametric, then matrix D is additive.*

Proof Let T'' be the ultrametric tree for matrix D' . (We don't use T' here since it was defined from T , which is unknown). First, assign weights to edges of T'' so that the path from any leaf i to an ancestor node w has a distance equal to the number labeling node w . (The label on each leaf is zero). To do this, just assign to each edge (p, q) the absolute difference between the number written at node p and the number written at q . The resulting path-distance between a pair of leaves (i, j) is exactly twice the number written at the least common ancestor of i and j . But, because T'' is an ultrametric tree for matrix D' , that distance is exactly $2 \times D'(i, j) = 2 \times m_v + D(i, j) - D(v, i) - D(v, j)$. Therefore, if for each leaf i , we now "shrink" the leaf edge into i by $m_v - D(v, i)$, then the resulting path between leaves i and j will have distance exactly $D(i, j)$. That creates an additive tree for D from an ultrametric tree for D' . \square

In summary, matrix D is additive if and only if D' is ultrametric. Moreover, if D is additive, then the additive tree T for D can be created as follows:

Additive tree algorithm Create matrix D' from D and construct the ultrametric tree T'' from D' . Next, assign a distance to each edge equal to the absolute difference between the node labels of its end points. Then, for each leaf i , subtract $m_v - D(v, i)$ from the distance on the edge into leaf i . The resulting tree is an additive tree for matrix D .

For an example, see Figure 7.

All the steps of the algorithm are easily implemented in $O(n^2)$ time, hence

Theorem 4.3 *An additive tree for an additive matrix can be constructed in $O(n^2)$ time.*

At heart, this algorithm relies on the algorithm for building an ultrametric tree (Section 1.3), and the heart of that algorithm involves only sorting and partitioning of numbers. Hence, when fully implemented along these lines, an additive tree can be constructed for D by a combinatorial algorithm (rather than a numerical one) that only sorts and partitions numbers held in matrix D' .

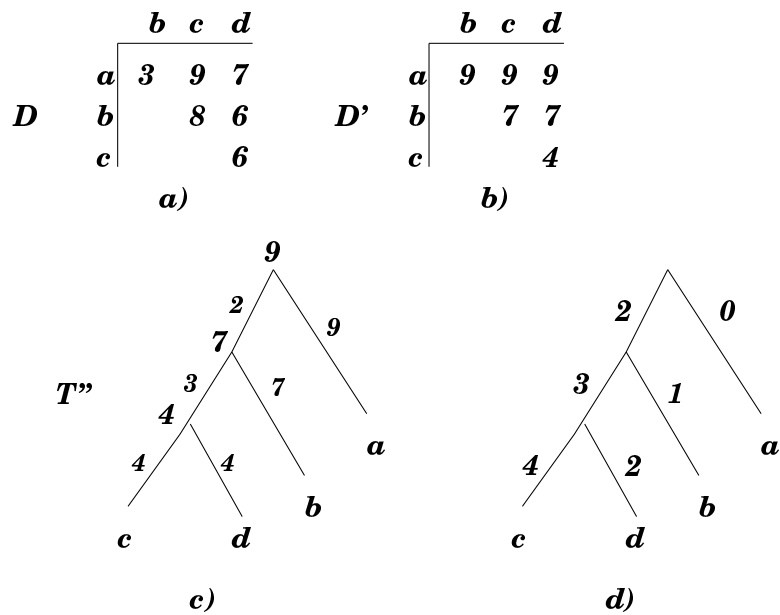


Figure 7: Figure a) shows the distance matrix D obtained from the tree in Figure 5 a). The largest entry has value 9 and is in row a . Figure b) shows the derived ultrametric matrix D' , and Figure c) shows the ultrametric tree T'' along with the derived edge weights. Figure d) shows the resulting tree after $m_a - D(a, i)$ is subtracted from leaf edges. The original tree is recovered after contracting the zero-weight edge to leaf a .

References

- [1] R. Agarwala, V. Bafna, M. Farach, B. Narangyan, M. Paterson, and M. Thorup. On the approximability of numerical taxonomy: fitting distances with trees. *Proc. 7th ACM-SIAM Symp. on Discrete Algs.*, pages 365–372, 1996.
- [2] 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.
- [3] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J. Watson. *Molecular Biology of the Cell (third edition)*. Garland press, New York, NY., 1994.
- [4] S. Baldauf and J. Palmer. Animals and fungi are each other’s closest relatives: Congruent evidence from multiple proteins. *Proc. of the Nat. Academy of Science*, 90:11558–11562, 1993.
- [5] H. J. Bandelt. Recognition of tree metrics. *SIAM J. on Discrete Math*, 3:3–6, 1990.
- [6] J.P. Barthelemy and A. Guenoche. *Trees and proximity representations*. Wiley, New York, 1991.
- [7] T. Beardsley. Smart genes. *Scientific American*, pages 86–95, August 1991.
- [8] H. Bodlaender, M. Fellows, and T. Warnow. Two strikes against perfect phylogeny. *Proc. of the 19th Inter. colloquium on Automata, Languages and Programming*, pages 273–283, 1992.
- [9] W. J. Broad. Clues to fiery origins of life sought in hothouse microbes. *The New York Times*, May 9, 1995.
- [10] P. Buneman. The recovery of trees from measures of dissimilarity. In D.G. Kendall and P. Tautu, editors, *Mathematics in the archaeological and historical sciences*, pages 387–385. Edinburgh University Press, 1971.
- [11] P. Buneman. A characterization of rigid circuit graphs. *Discrete Math*, 9:205–212, 1974.
- [12] P. Buneman. A note on metric properties of trees. *J. of Combinatorial Theory (B)*, 17:48–50, 1974.
- [13] J. Culberson and P. Rudnicki. A fast algorithm for constructing trees from distance matrices. *Info. Proc. Lets.*, 30:215–220, 1989.
- [14] C. R. Darwin. *The origin of species*. John Murray, London, 1859.

- [15] W. H. Day. Computational complexity of inferring phylogenies from dissimilarity matrices. *Bull. of Math. Biology*, 49:461–467, 1987.
- [16] N.J. Deacon, J. Mills, and et al. Genomic structure of an attenuated quasi species of hiv-1 from a blood transfusion donor and recipients. *Science*, 270:988–991, 1995.
- [17] R. F. Doolittle. *Of Urfs and Orfs: A primer on how to analyze derived amino acid sequences*. University Science Books, Mill Valley, CA., 1986.
- [18] R. F. Doolittle, R.F. Feng, S. Tsang, G. Cho, and E. Little. Determining divergence times of the major kingdoms of living organisms with a protein clock. *Science*, 271:470–477, 1996.
- [19] A. Dress and M. Steel. Convex tree realizations of partitions. *Applied Math Letters*, 5:3–6, 1993.
- [20] M. Farach, S. Kannan, and T. Warnow. A robust model for finding evolutionary trees. *Algorithmica*, 13:511–520, 1995.
- [21] A. Feducia. Explosive evolution in tertiary birds and mammals. *science*, 267:637–638, 1995.
- [22] J. Felsenstein. Numerical methods for inferring evolutionary trees. *The Quarterly Review of Biology*, pages 379–404, 1982.
- [23] J. Felsenstein. Parsimony in systematics: Biological and statistical issues. *Ann. Rev. Ecol. Syst.*, 14:313–333, 1983.
- [24] J. Felsenstein. Perils of molecular introspection. *Nature*, 335:118–118, 1988.
- [25] J. Felsenstein. Phylogenies from molecular sequences: Inference and reliability. *Ann. Review of Genetics*, 22:521–565, 1988.
- [26] W. M. Fitch and E. Margoliash. The construction of phylogenetic trees. *Science*, 155, 1967.
- [27] P.L. Foley, C.J. Humphries, I.L. Kitching, R.W. Scotland, D.J. Siebert, and D.M. Williams. *Cladistics: A practical course in systematics*. Oxford University Press, Oxford, 1992.
- [28] S.J. Gould. Through a lens, darkly. *Natural History*, pages 16–24, September 1989.
- [29] K. Halanych, J. D. Bacheller, A. M. Aguinaldo, S. M. Liva, D. M. Hillis, and J. A. Lake. Evidence from 18s ribosomal DNA that lophophorates are protostome animals. *Science*, 267:1641–1643, 1995.

- [30] M. Hanks, W. Wurst, L. Anson-Cartwright, A. Auerback, and A. Joyner. Rescue of the en-1 mutant phenotype by replacement of en-1 with en-2. *Science*, 269:679–682, 1995.
- [31] J. Hein. An optimal algorithm to reconstruct trees from additive distance data. *Bull. of Math. Biology*, 51:597–603, 1989.
- [32] D. Hillis and C. Moritz (eds). *Molecular Systematics*. Sinauer Associates, Sunderland MA, 1990.
- [33] L. Hou, L. Martin, Z. Zhou, and A. Feduccia. Early adaptive radiation of birds: Evidence from fossils from northeastern china. *Science*, 274:1164–1167, 1996.
- [34] S. Kannan and T. Warnow. Inferring evolutionary history from dna sequences. *SIAM J. on Computing*, 23:713–737, 1994.
- [35] S. Kannan and T. Warnow. A fast algorithm for the computation and enumeration of perfect phylogenies when the number of character states is fixed. *Proc. 6th ACM-SIAM Symp. on Discrete Algs.*, pages 595–603, 1995.
- [36] M. Kimura. The neutral theory of molecular evolution. *Scientific American*, pages 98–126, November 1979.
- [37] M. Kimura. *The neutral theory of molecular evolution*. Cambridge University Press, Cambridge, 1983.
- [38] W. H. Li and D. Graur. *Fundamentals of Molecular Evolution*. Sinauer, Sunderland, MA, 1991.
- [39] J. Marx. Developmental biology: Knocking genes in instead of out. *Science*, 269:636, 1995.
- [40] M. Max, P. McKinnon, K. Seidenman, R. Barret, M. Applebury, J. Takahashi, and R. Margolskee. Pineal opsin: A nonvisual opsin expressed in chick pineal. *Science*, 267:1502–1506, 1995.
- [41] S. O’Brien. The ancestry of the giant panda. *Scientific American*, pages 102–116, November 1987.
- [42] R. Rhodes and A. Klug. Zinc fingers. *Scientific American*, pages 56–65, February 1993.
- [43] M. Ridley. *Evolution and Classification: The reformation of cladism*. Longman, London, 1986.
- [44] C. Schmid and J. Marks. Dna hybridization as a guide to phylogeny: Chemical and physical limits. *J. Mol. Evol.*, 30:237–246, 1990.

- [45] C.G. Sibley and J.E. Ahlquist. Reconstructing bird phylogeny by comparing dna's. *Scientific American*, pages 82–92, February 1986.
- [46] C.G. Sibley and J.E. Ahlquist. *Phylogeny and classification of birds*. Yale University Press, New Haven, Connecticut, 1990.
- [47] M. Steel. The complexity of reconstructing trees from qualitative characters and subtrees. *J. of Classification*, 9:91–116, 1992.
- [48] D. L. Swofford and G. L. Olsen. Phylogeny reconstruction. In D. M. Hillis and C. Moritz, editors, *Molecular systematics*, pages 411–501. Sinauer, 1990.
- [49] M. Syvanen. Horizontal gene transfer: Evidence and possible consequences. *Ann. Review of Genetics*, 28:237–261, 1994.
- [50] M. Waterman, T. Smith, M. Singh, and W. Beyer. Additive evolutionary trees. *J. Theo. Biol.*, 64:199–213, 1977.
- [51] R. Wayne and J. Gittleman. The problematic red wolf. *Scientific American*, pages 36–39, July 1995.
- [52] E. Zuckerkandl and L. Pauling. Molecular disease, evolution and genic heterogeneity. In M. Kash and B. Pullman, editors, *Horizons in Biochemistry*, pages 189–225. Academic Press, 1962.