

Phylogeny reconstruction: overview

This chapter provides an overview of phylogeny reconstruction methods. We introduce some basic concepts used to describe trees and discuss general features of tree-reconstruction methods. We will describe distance and parsimony methods as well, while likelihood and Bayesian methods are discussed in Chapters 4 and 5.

3.1 Tree concepts

3.1.1 Terminology

3.1.1.1 Trees, nodes (vertexes), and branches (edges)

A phylogeny or phylogenetic tree is a representation of the genealogical relationships among species, among genes, among populations, or even among individuals. Mathematicians define a graph as a set of *vertexes* and a set of *edges* connecting them, and a tree as a connected graph without loops (see, e.g., p. 1 in Tucker 1995). Biologists instead use *nodes* for vertexes and *branches* for edges. Here we consider species trees, but the description also applies to trees of genes or individuals. The *tips*, *leaves*, or *external nodes* represent present-day species, while the *internal nodes* usually represent extinct ancestors for which no sequence data are available. The ancestor of all sequences is the *root* of the tree. Trees are drawn equivalently with the root on the top, at the bottom, or on the side.

3.1.1.2 Root of the tree and rooting the tree

A tree with the root specified is called a *rooted tree* (Fig. 3.1a), while a tree in which the root is unknown or unspecified is called an *unrooted tree* (Fig. 3.1b). If the evolutionary rate is constant over time, an assumption known as the *molecular clock*, distance-matrix and maximum likelihood methods can identify the root and produce rooted trees. Such use of the clock assumption to determine the root of the tree is known as *molecular-clock rooting*. However, the clock assumption is most often violated, except for closely related species. Without the clock, most tree-reconstruction methods are unable to identify the root of the tree and produce unrooted trees. Then a commonly used strategy is the so-called *outgroup rooting*. Distantly related species, called the *outgroups*, are included in tree reconstruction, while in the reconstructed unrooted tree for all species the root is placed on the branch leading to the outgroups, so that the subtree for the *ingroups* is rooted. In the example of Fig. 3.1, the orangutan is used as the outgroup to root the tree for the ingroup species: human, chimpanzee, and

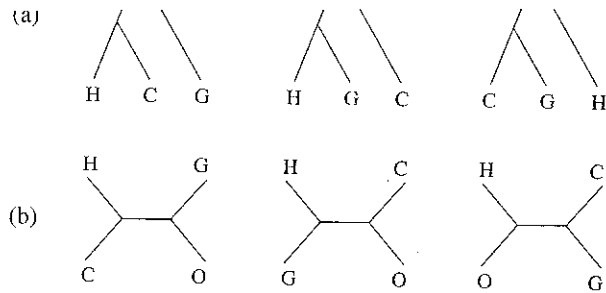


Fig. 3.1 Outgroup rooting. To infer the relationships among human (H), chimpanzee (C), gorilla (G), represented in the three rooted trees in (a), we use orangutan (O) as the outgroup. Tree-reconstruction methods allow us to estimate an unrooted tree, that is, one of the trees in (b). As the root is along the branch leading to the outgroup, these three unrooted trees for all species correspond to the three rooted trees for the ingroup species H, C, and G.

gorilla. In general, outgroups closely related to the ingroup species are better than distantly related outgroups. In the universal tree of life, no outgroup species exist. Then a strategy is to root the tree using ancient gene duplications that occurred prior to the divergences of all existing life forms (Gogarten *et al.* 1989; Iwabe *et al.* 1989). The subunits of ATPase arose through a gene duplication before the divergence of eubacteria, eukaryotes, and archaeobacteria. Protein sequences from both paralogues were used to construct a composite unrooted tree, and the root was placed on the branch separating the two duplicates. Elongation factors Tu and G constitute another ancient duplication, used in rooting the universal tree of life.

3.1.1.3 Tree topology, branch lengths, and the parenthesis notation

The branching pattern of a tree is called the *topology* of the tree. The length of a branch may represent the amount of sequence divergence or the time period covered by the branch. A tree showing only the tree topology without information about branch lengths is sometimes called a *cladogram* (Fig. 3.2a), while a tree showing both the topology and branch lengths is called a *phylogram* (Fig. 3.2b). For use in computer programs, trees are often represented using the parenthesis notation, also known as the Newick format. This uses a pair of parentheses to group sister taxa into one clade, with a semicolon marking the end of the tree. Branch lengths, if any, are prefixed by colons. For example, the trees in Fig. 3.2 are represented as

- a and b: (((A, B), C), D), E);
- b: (((A: 0.1, B: 0.2): 0.12, C: 0.3): 0.123, D: 0.4): 0.1234, E: 0.5);
- c: (((A, B), C), D, E);
- c: (((A: 0.1, B: 0.2): 0.12, C: 0.3): 0.123, D: 0.4, E: 0.6234);

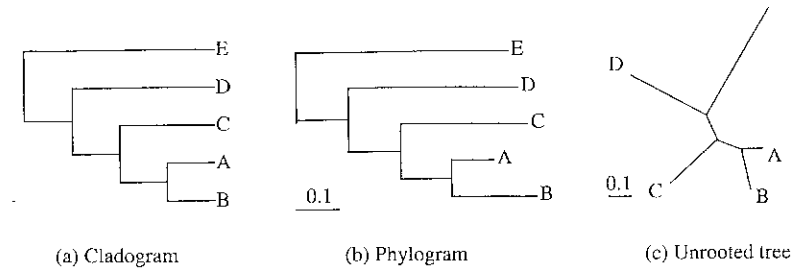


Fig. 3.2 The same tree shown in different styles. (a) The cladogram shows the tree topology without branch lengths or with branch lengths ignored. (b) In a phylogram, branches are drawn in proportion to their lengths. (c) In an unrooted tree, the location of the root is unknown or ignored.

Branch lengths here are measured by the expected number of nucleotide substitutions per site, like the sequence distances discussed in Chapter 1. This format is natural for representing rooted trees. Unrooted trees are represented as rooted and the representation is not unique since the root can be placed anywhere on the tree. For example, the tree in Figs. 3.2(c) can also be represented as '(A, B, (C, (D, E)))';

3.1.1.4 Bifurcating and multifurcating trees

The number of branches connected to a node is called the *degree* of the node. Leaves have a degree of 1. If the root node has a degree greater than 2 or a nonroot node has a degree greater than 3, the node represents a *polytomy* or *multifurcation*. A tree with no polytomies is called a *binary tree*, *bifurcating tree*, or *fully resolved tree*. The most extreme unresolved tree is the *star* or *big-bang tree*, in which the root is the only internal node (see Fig. 3.3 for examples). A polytomy representing truly simultaneous species divergences is sometimes called a *hard polytomy*. It would seem extremely unlikely for one species to diverge into several at exactly the same time, and it may be argued that hard polytomies do not exist. Most often the polytomy represents lack of information in the data to resolve the relationship within a clade (a group of species). Such polytomies are called *soft polytomies*.

3.1.1.5 The number of trees

We can work out the total number of unrooted trees by the following *stepwise addition algorithm* (Cavalli-Sforza and Edwards 1967) (Fig. 3.4). We start with the single tree for the first three species. This tree has three branches to which the fourth species can be added. Thus there are three possible trees for the first four species. Each four-species tree has five branches, to which the fifth species can be added, resulting in five different five-species trees for each four-species tree. In general, a tree of the first $n - 1$ species has $(2n - 5)$ branches, to which the n th species can be added, so that

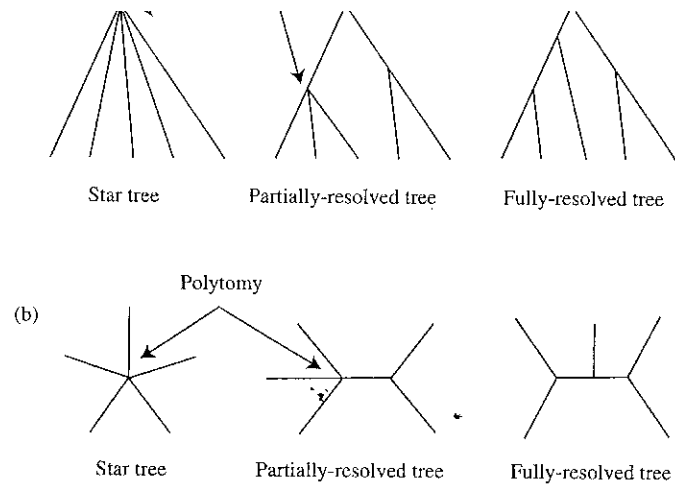


Fig. 3.3 Unresolved and resolved phylogenetic trees: (a) rooted trees, (b) unrooted trees.

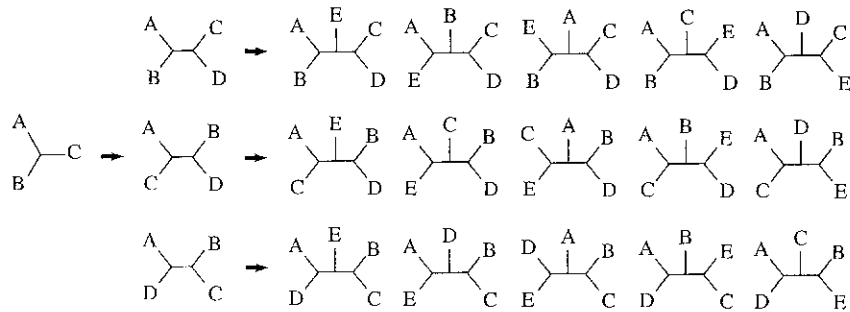


Fig. 3.4 Enumeration of all trees for five taxa A, B, C, D, and E using the stepwise addition algorithm.

each of the $(n - 1)$ -species tree generates $(2n - 5)$ new n -species trees. Thus the total number of unrooted bifurcating trees for n species is

$$T_n = T_{n-1} \times (2n - 5) = 3 \times 5 \times 7 \times \dots \times (2n - 5). \quad (3.1)$$

To work out the number of rooted trees for n species, note that each unrooted tree has $(2n - 3)$ branches, and the root can be placed on any of those branches, generating $(2n - 3)$ rooted trees from each unrooted tree. Thus the number of rooted trees for n species is simply $T_n \times (2n - 3) = T_{n+1}$. As we can see from Table 3.1, the number of trees increases explosively with the number of species.

rooted (T_{n+1}) trees for n species

n	T_n	T_{n+1}
3	1	3
4	3	15
5	15	105
6	105	945
7	945	10 395
8	10 395	135 135
9	135 135	2 027 025
10	2 027 025	34 459 425
20	$\sim 2.22 \times 10^{20}$	$\sim 8.20 \times 10^{21}$
50	$\sim 2.84 \times 10^{74}$	$\sim 2.75 \times 10^{76}$

3.1.2 Topological distance between trees

Sometimes we would like to measure how different two trees are. For example, we may be interested in the differences among trees estimated from different genes, or the differences between the true tree and the estimated tree in a computer simulation conducted to evaluate a tree-reconstruction method.

A commonly used measure of topological distance between two trees is the *partition distance* defined by Robinson and Foulds (1981) (see also Penny and Hendy 1985). We define this distance for unrooted trees here, but the same definition applies to rooted trees as well, by imagining an outgroup species attached to the root. Note that each branch on the tree defines a *bipartition* or *split* of the species; if we chop the branch, the species will fall into two mutually exclusive sets. For example, branch b in tree T_1 of Fig. 3.5 partitions the eight species into two sets: (1, 2, 3) and (4, 5, 6, 7, 8). This partition is also present on tree T_2 . Partitions defined by terminal branches are in all possible trees and are thus not informative for comparison between trees. Thus we focus on internal branches only. Partitions defined by branches $b, c, d,$ and e of tree T_1 are the same as partitions defined by branches $b', c', d',$ and e' of tree T_2 , respectively. The partition defined by branch a of tree T_1 is not in tree T_2 , nor is the partition defined by branch a' of tree T_2 in tree T_1 . The partition distance is defined as the total number of bipartitions in one tree that are not in the other. Thus T_1 and T_2 have a partition distance of 2. As a binary tree of n species has $(n - 3)$ internal branches, the partition distance between any two binary trees of n species ranges from 0 (if the two trees are identical) to $2(n - 3)$ (if the two trees do not share any bipartition).

The partition distance can be equivalently defined as the number of contractions and expansions needed to transform one tree into the other. Removing an internal branch by reducing its length to zero is a contraction while creating an internal branch is an expansion. Trees T_1 and T_2 of Fig. 3.5 are separated by a contraction (from T_1 to T_0) and an expansion (from T_0 to T_2), so that their partition distance is 2.

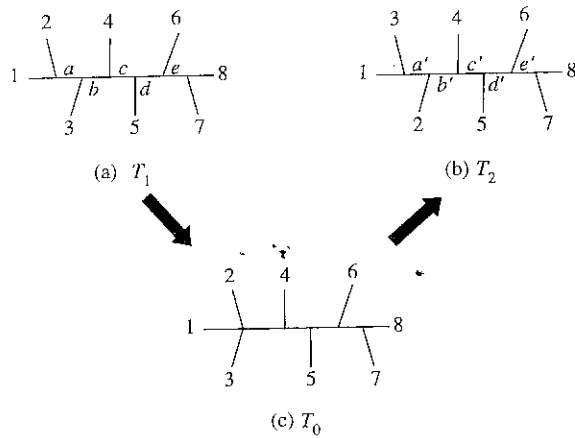


Fig. 3.5 The partition distance between two trees T_1 and T_2 is the total number of bipartitions that are in one tree but not in the other. It is also the number of contractions and expansions needed to change one tree into another. A contraction converts T_1 into T_0 and an expansion converts T_0 into T_2 , so the distance between T_0 and T_1 is 1 while the distance between T_1 and T_2 is 2.

The partition distance has limitations. First, the distance does not recognize certain similarities between trees. The three trees in Fig. 3.6 are identical concerning the relationships among species 2–7 but do not share any bipartitions, so that the partition distance between any two of them is the maximum possible. Indeed, the probability that a random pair of unrooted trees achieve the maximum distance is 70–80% for $n = 5$ –10, and is even greater for larger n . Figure 3.7 shows the distribution of partition distances for the case of $n = 10$. Second, the partition distance ignores branch lengths in the tree. Intuitively, two trees that are in conflict around short internal branches are less different than two trees that are in conflict around long internal branches. There are no good rules to follow concerning incorporation of branch lengths in defining a distance between two trees; one such measure is suggested by Kuhner and Felsenstein (1994). Third, the partition distance may be misleading if either of the two trees has multifurcations. Suppose we conduct a computer simulation to compare two tree-reconstruction methods, using a binary tree to simulate data sets. We use the partition distance to measure performance: $P = 1 - D/D_{\max}$, where $D_{\max} = 2(n - 3)$ is the maximum distance and D is the distance between the true tree and the estimated tree. When both the true tree and the estimated tree are binary, P is the proportion of bipartitions in the true tree that are recovered in the estimated tree. If there is no information in the data, the first method returns the star tree as the estimate while the second method returns an arbitrarily resolved binary tree. Now for the first method, $D = (n - 3) = D_{\max}/2$, so that $P = 50\%$. The second method has a performance of $P = 1/3$ when $n = 3$ or nearly 0 for large n , since a random tree is very likely not to share any bipartitions with the true tree. However, the two methods clearly have the

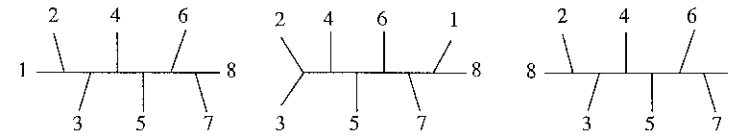


Fig. 3.6 Three trees that do not share any bipartitions and thus achieve the maximum partition distance.

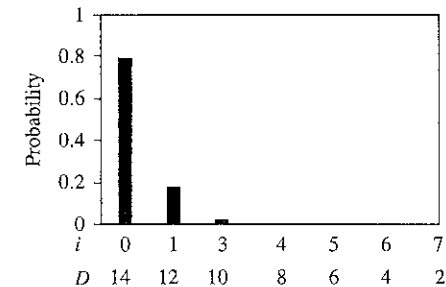


Fig. 3.7 The probability that two random trees from all possible unrooted trees of 10 species share i bipartitions or have partition distance D . Note that $D = 2 \times (10 - 3 - i)$.

same performance, and the measure based on the partition distance is unreasonable for the first method.

3.1.3 Consensus trees

While the partition distance measures how different two trees are, a consensus tree summarizes common features among a collection of trees. Many different consensus trees have been defined; see Bryant (2003) for a comprehensive review. Here we introduce two of them.

The *strict consensus tree* shows only those groups (nodes or clades) that are shared among all trees in the set, with polytomies representing nodes not supported by all trees in the set. Consider the three trees in Fig. 3.8(a). The strict consensus tree is shown in Fig. 3.8(b). The clade (A, B) is in the first and third trees but not in the second, while the clade (A, B, C) is in all three trees. Thus the strict consensus tree shows the clade (A, B, C) as a trichotomy, as is the clade (F, G, H). The strict consensus tree is a conservative way of summarizing the trees and may not be very useful as it often produces the star tree.

The *majority-rule consensus tree* shows nodes or clades that are supported by at least half of the trees in the set. It is also common practice to show the percentage of trees that support every node on the consensus tree (Fig. 3.8c). For example, the clade (A, B) is in two out of the three trees and is thus shown in the majority-rule consensus tree as resolved, with the percentage of support (2/3) shown next to

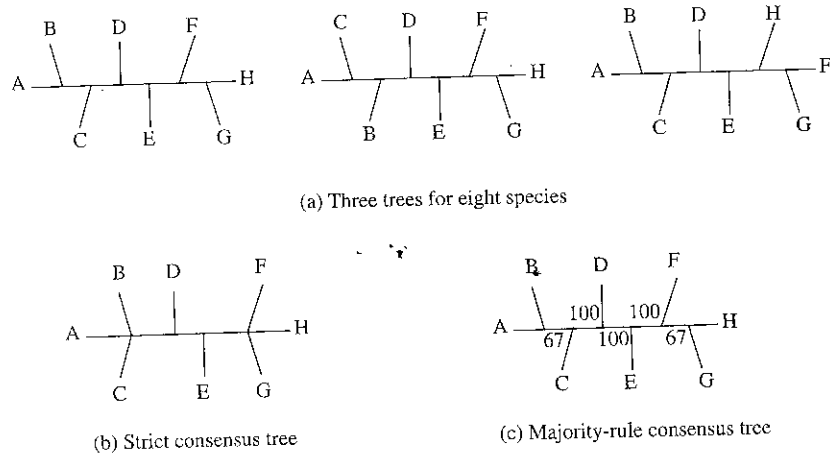


Fig. 3.8 Three trees for eight species (a) and their strict consensus tree (b) and majority-rule consensus tree (c).

the node. It is known that all clades that occur in more than half of the trees in the set can be shown on the same consensus tree without generating any conflict.

Like the partition distance, the majority-rule consensus tree, as a summary of trees in the set, has limitations. Suppose that there are only three distinct trees in the set, which are the trees of Fig. 3.6, each occurring in proportions around 33%. Then the majority-rule consensus tree will be the star tree. In such cases, it appears more informative to report the first few whole trees with the highest support values.

3.1.4 Gene trees and species trees

The phylogeny representing the relationships among a group of species is called the *species tree*. The phylogeny for a set of gene sequences from the species is called the *gene tree*. A number of factors may cause the gene tree to differ from the species tree.

First, estimation errors may cause the estimated gene tree to be different from the species tree even if the (unknown) true gene tree agrees with the species tree. The estimation errors may be either random, due to the limited amount of sequence data, or systematic, due to deficiencies of the tree-reconstruction method. Second, during the early stages of evolution near the root of the universal tree of life, there appears to have been substantial lateral (horizontal) gene transfer (LGT). As a result, different genes or proteins may have different gene trees, in conflict with the species tree. The LGT appears to be so extensive that some researchers question the concept of a universal tree of life (see, e.g., Doolittle 1998). Third, gene duplications, especially if followed by gene losses, can cause the gene tree to be different from the species tree if paralogous copies of the gene are used for phylogeny reconstruction (Fig. 3.9a). Fourth, when the species are closely related, *ancestral polymorphism* or *lineage sorting* can cause

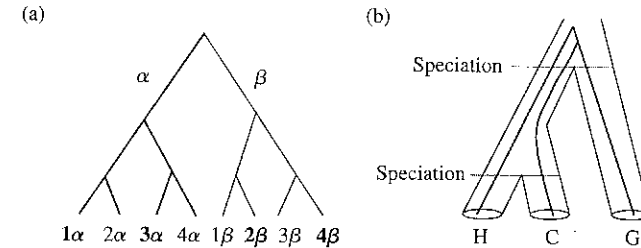


Fig. 3.9 Conflict between species tree and gene tree can be due to gene duplication (a) or ancestral polymorphism (b). In (a), a gene duplicated in the past, creating paralogous copies α and β , followed by divergences of species 1, 2, 3, and 4. If we use gene sequences 1 α , 3 α , 2 β , 4 β for phylogeny reconstruction, the true gene tree is ((1 α , 3 α), (2 β , 4 β)), different from the species tree ((1, 2), (3, 4)). In (b), the species tree is ((human, chimpanzee), gorilla). However, due to ancestral polymorphism or lineage sorting, the true gene tree is ((human, (chimpanzee, gorilla))).

gene trees to be different from the species tree. An example is shown in Fig. 3.9(b). Here the species tree for human, chimpanzee, and gorilla is ((H, C), G). However, because of sequence variations (polymorphisms) in the extinct ancestral species, the true gene tree is (H, (C, G)). The probability that the gene tree differs from the species tree is greater if the speciation events are closer in time (that is, if the species tree is almost a star tree) and if the long-term population size of the H–C common ancestor is greater. Such information concerning the conflicts between the species tree and the gene trees can be used to estimate the effective population sizes of extinct common ancestors by using sequences from extant species at multiple neutral loci (Takahata 1986; Takahata *et al.* 1995; Yang 2002; Rannala and Yang 2003).

3.1.5 Classification of tree-reconstruction methods

Here we consider some overall features of phylogeny reconstruction methods. First, some methods are *distance based*. In those methods, distances are calculated from pairwise comparison of sequences, forming a distance matrix, which is used in subsequent analysis. A clustering algorithm is often used to convert the distance matrix into a phylogenetic tree (Everitt *et al.* 2001). The most popular methods in this category include UPGMA (unweighted pair-group method using arithmetic averages) (Sokal and Sneath 1963) and neighbour joining (Saitou and Nei 1987). Other methods are *character based*, and attempt to fit the characters (nucleotides or amino acids) observed in all species at every site to a tree. Maximum parsimony (Fitch 1971b; Hartigan 1973), maximum likelihood (ML) (Felsenstein 1981), and Bayesian methods (Rannala and Yang 1996; Mau and Newton 1997; Li *et al.* 2000) are all character based. Distance methods are often computationally faster than character-based methods, and can be easily applied to analyse different kinds of data as long as pairwise distances can be calculated.

Table 3.2 Optimality criteria used for phylogeny reconstruction

Method	Criterion (tree score)
Maximum parsimony	Minimum number of changes, minimized over ancestral states
Maximum likelihood	Log likelihood score, optimized over branch lengths and model parameters
Minimum evolution	Tree length (sum of branch lengths, often estimated by least squares)
Bayesian	Posterior probability, calculated by integrating over branch lengths and substitution parameters

Tree-reconstruction methods can also be classified as being either *algorithmic* (cluster methods) or *optimality based* (search methods). The former include UPGMA and neighbour joining, which use cluster algorithms to arrive at a single tree from the data as the best estimate of the true tree. Optimality-based methods use an optimality criterion (objective function) to measure a tree's fit to data, and the tree with the optimal score is the estimate of the true tree (Table 3.2). In the maximum parsimony method, the tree score is the minimum number of character changes required for the tree, and the *maximum parsimony tree* or *most parsimonious tree* is the tree with the smallest tree score. The ML method uses the log likelihood value of the tree to measure the fit of the tree to the data, and the *maximum likelihood tree* is the tree with the highest log likelihood value. In the Bayesian method, the posterior probability of a tree is the probability that the tree is true given the data. The tree with the maximum posterior probability is the estimate of the true tree, known as the *MAP tree*. In theory, methods based on optimality criteria have to solve two problems: calculation of the criterion for a given tree and search in the space of all trees to identify the tree with the best score. The first problem is often straightforward but the second is virtually impossible when the number of sequences is greater than 20 or 50 because of the huge number of possible trees. As a result, heuristic algorithms are used for tree search. Optimality-based search methods are usually much slower than algorithmic cluster methods.

Some tree-reconstruction methods are model based. Distance methods use nucleotide or amino acid substitution models to calculate pairwise distances. Likelihood and Bayesian methods use substitution models to calculate the likelihood function. These methods are clearly model based. Parsimony does not make explicit assumptions about the evolutionary process. Opinions differ as to whether the method makes any implicit assumptions, and, if so, what they are. We will return to this issue in Chapter 6.

3.2 Exhaustive and heuristic tree search

3.2.1 Exhaustive tree search

For parsimony and likelihood methods of tree reconstruction, which evaluate trees according to an optimality criterion, one should in theory calculate the score for every

possible tree and then identify the tree having the best score. Such a strategy is known as *exhaustive search* and is guaranteed to find the best tree. As mentioned above, the stepwise addition algorithm provides a way of enumerating all possible trees for a fixed number of species (Fig. 3.4).

Exhaustive search is, however, computationally unfeasible except for small data sets with, say, fewer than 10 taxa. For the parsimony method, a branch-and-bound algorithm has been developed to speed up the exhaustive search (Hendy and Penny 1982). Even so, the computation is feasible for small data sets only. For the likelihood method, such an algorithm is not available. Thus most computer programs use heuristic algorithms to search in the tree space, and do not guarantee to find the optimal tree.

3.2.2 Heuristic tree search

Heuristic search algorithms may be grouped into two categories. The first includes hierarchical clustering algorithms. These may be subdivided into *agglomerative* methods, which proceed by successive fusions of the n species into groups, and *divisive* methods, which separate the n species successively into finer groups (Everitt *et al.* 2001). Whether every step involves a fusion or fission, the algorithm involves choosing one out of many alternatives, and the optimality criterion is used to make that choice. The second category of heuristic tree-search algorithms includes *tree-rearrangement* or *branch-swapping* algorithms. They propose new trees through local perturbations to the current tree, and the optimality criterion is used to decide whether or not to move to a new tree. The procedure is repeated until no improvement can be made in the tree score. We describe two cluster algorithms in this subsection and a few branch-swapping algorithms in the next.

Stepwise addition or *sequential addition* is an agglomerative algorithm. It adds sequences one by one, until all sequences are in the tree. When each new sequence is added, all the possible locations are evaluated and the best is chosen using the optimality criterion. Figure 3.10 illustrates the algorithm for the case of five sequences, using parsimony score as the optimality criterion. Note that this algorithm of heuristic tree search is different from the stepwise addition algorithm for enumerating all possible trees explained in Fig. 3.4. In the heuristic search, the locally best subtree is selected at each step, and trees that can be generated from the suboptimal subtrees are ignored. In our example, the 10 five-species trees on the second and third rows of Fig. 3.4 are never visited in the heuristic search. Thus the algorithm is not guaranteed to find the globally optimal tree. It is less clear whether one should add the most similar sequences or the most divergent sequences first. A common practice is to run the algorithm multiple times, adding sequences in a random order.

Star decomposition is a divisive cluster algorithm. It starts from the star tree of all species, and proceeds to resolve the polytomies by joining two taxa at each step. From the initial star tree of n species, there are $n(n-1)/2$ possible pairs, and the pair that results in the greatest improvement in the tree score is grouped together. The root of the tree then becomes a polytomy with $(n-1)$ taxa. Every step of the algorithm

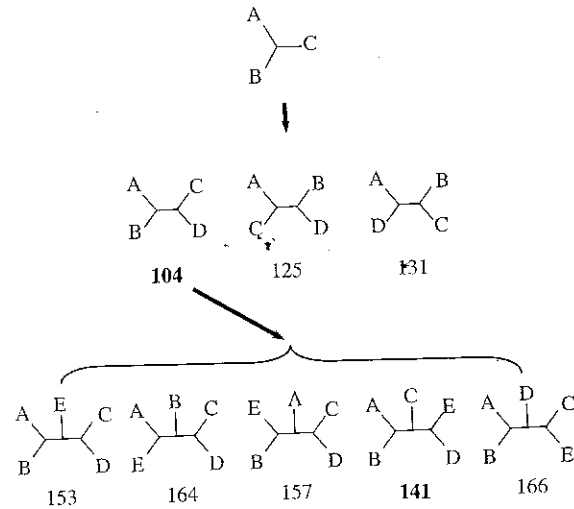


Fig. 3.10 Stepwise addition algorithm under the maximum parsimony criterion. The tree score is the minimum number of changes required by the tree.

reduces the number of taxa connected to the root by one. The procedure is repeated until the tree is fully resolved. Figure 3.11 shows an example of five sequences, using the log likelihood score for tree selection.

For n species, the stepwise-addition algorithm evaluates three trees of four species, five trees of five species, seven trees of six species, with a total of $3 + 5 + 7 + \dots + (2n - 5) = (n - 1)(n - 3)$ trees in total. In contrast, the star-decomposition algorithm evaluates $n(n - 1)/2 + (n - 1)(n - 2)/2 + \dots + 3 = n(n^2 - 1)/6 - 7$ trees in total, all of which are for n species. Thus for $n > 4$, the star-decomposition algorithm evaluates many more and bigger trees than the stepwise-addition algorithm and is expected to be much slower. The scores for trees constructed during different stages of the stepwise-addition algorithm are not directly comparable as the trees are of different sizes. Trees evaluated in the star-decomposition algorithm are all of the same size and their tree scores are comparable.

Both the stepwise-addition and star-decomposition algorithms produce resolved trees of all n species. If we stop at the end of either algorithm, we have an algorithmic cluster method for tree reconstruction based on the optimality criterion. However, in most programs, trees generated from these algorithms are treated as starting trees and subjected to local rearrangements. Below are a few such algorithms.

3.2.3 Branch swapping

Branch swapping or tree rearrangements are heuristic algorithms of hill climbing in the tree space. An initial tree is used to start the process. This can be a random tree, or a tree produced by stepwise-addition or star-decomposition algorithms, or by

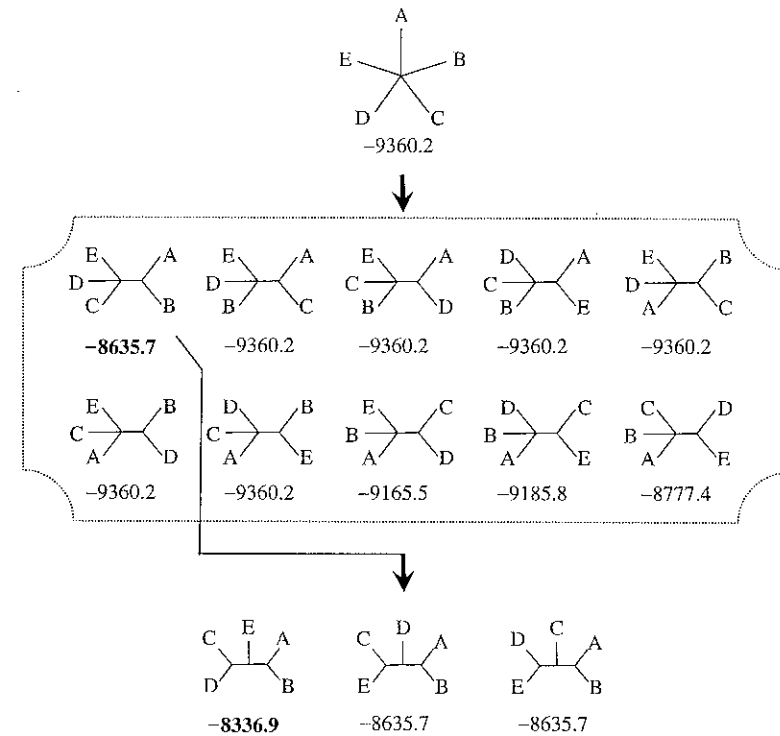


Fig. 3.11 Star-decomposition algorithm under the likelihood criterion. The tree score is the log likelihood value calculated by optimizing branch lengths on the tree.

other faster algorithms such as neighbour joining (Saitou and Nei 1987). The branch-swapping algorithm generates a collection of neighbour trees around the current tree. The optimality criterion is then used to decide which neighbour to move to. The branch-swapping algorithm affects our chance of finding the best tree and the amount of computation it takes to do so. If the algorithm generates too many neighbours, each step will require the evaluation of too many candidate trees. If the algorithm generates too few neighbours, we do not have to evaluate many trees at each step, but there may be many local peaks in the tree space (see below) and the search can easily get stuck at a local peak.

In *nearest-neighbour interchange (NNI)* each internal branch defines a relationship among four subtrees, say, $a, b, c,$ and d (Fig. 3.12). Suppose the current tree is $((a, b), c, d)$ and the two alternative trees are $((a, c), b, d)$ and $((a, d), b, c)$. The NNI algorithm allows us to move from the current tree to the two alternative trees by swapping a subtree on one side of the branch with a subtree on the other side. An unrooted tree for n species has $n - 3$ internal branches. The NNI algorithm thus generates $2(n - 3)$ immediate neighbours. The neighbourhood relationships among the 15 trees for five species are illustrated below in Fig. 3.14.

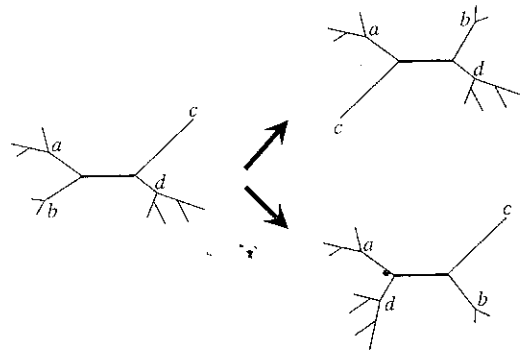


Fig. 3.12 The nearest-neighbour interchange (NNI) algorithm. Each internal branch in the tree connects four subtrees or nearest neighbours (*a, b, c, d*). Interchanging a subtree on one side of the branch with another on the other side constitutes an NNI. Two such rearrangements are possible for each internal branch.

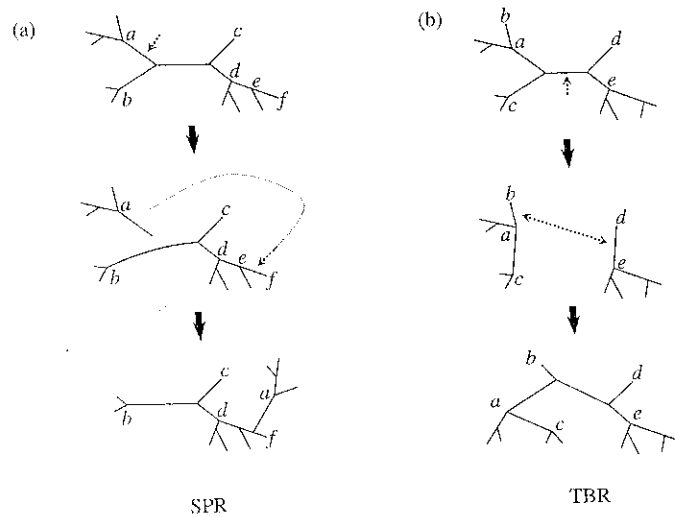


Fig. 3.13 (a) Branch swapping by subtree pruning and regrafting (SPR). A subtree (for example, the one represented by node *a*), is pruned, and then reattached to a different location on the tree. (b) Branch swapping by tree bisection and reconnection (TBR). The tree is broken into two subtrees by cutting an internal branch. Two branches, one from each subtree, are then chosen and rejoined to form a new tree.

Two other commonly used algorithms are *subtree pruning and regrafting* (SPR) and *tree bisection and reconnection* (TBR). In the former, a subtree is pruned and then reattached to a different location on the tree (Fig. 3.13a). In the latter, the tree is cut into two parts by chopping an internal branch and then two branches, one from

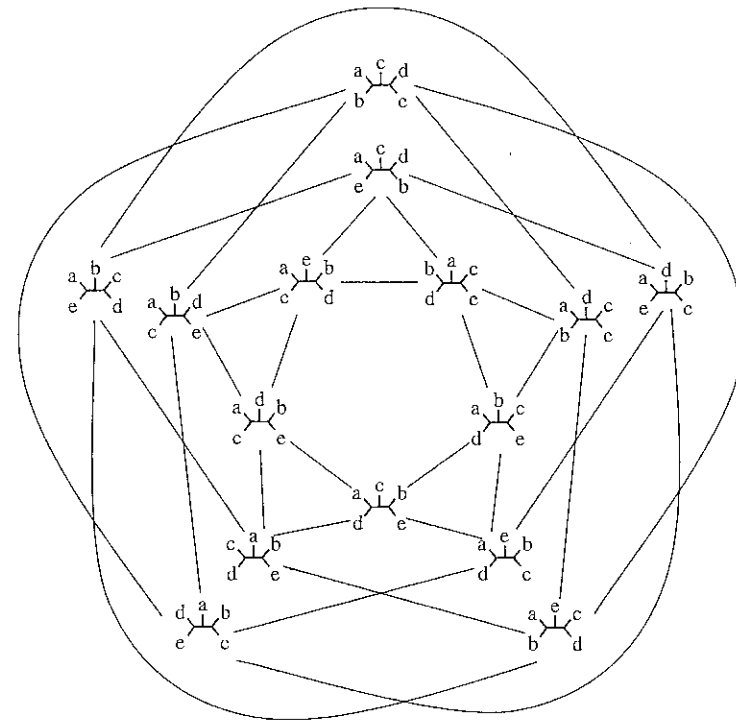


Fig. 3.14 The 15 trees for five species, with neighbourhood relationships defined by the NNI algorithm. Trees that are neighbours under NNI are connected. Note that this visually appealing representation has the drawback that trees close by may not be neighbours. Drawn following Felsenstein (2004).

each subtree, are chosen and rejoined to form a new tree (Fig. 3.13b). TBR generates more neighbours than SPR, which in turn generates more neighbours than NNI.

3.2.4 Local peaks in the tree space

Maddison (1991) and Charleston (1995) discussed local peaks or tree islands in the tree space. Figure 3.15 shows an example for five species and 15 trees. The neighbourhood relationship is defined using the NNI algorithm, with each tree having four neighbours (see Fig. 3.14). The parsimony tree lengths for the two trees on the top of the graph, T_1 and T_2 , are 656 and 651, with T_2 being the most parsimonious tree. The eight trees that are neighbours of T_1 and T_2 have tree lengths ranging from 727 to 749, while the five trees that are two steps away from T_1 and T_2 have tree lengths ranging from 824 to 829. Trees T_1 and T_2 are separated from each other by trees of much poorer scores and are thus local peaks. They are local peaks for the SPR and TBR algorithms as well. Also T_1 and T_2 are local peaks when the data are analysed using the likelihood criterion. Indeed for this data set, the rank order of the 15 trees is

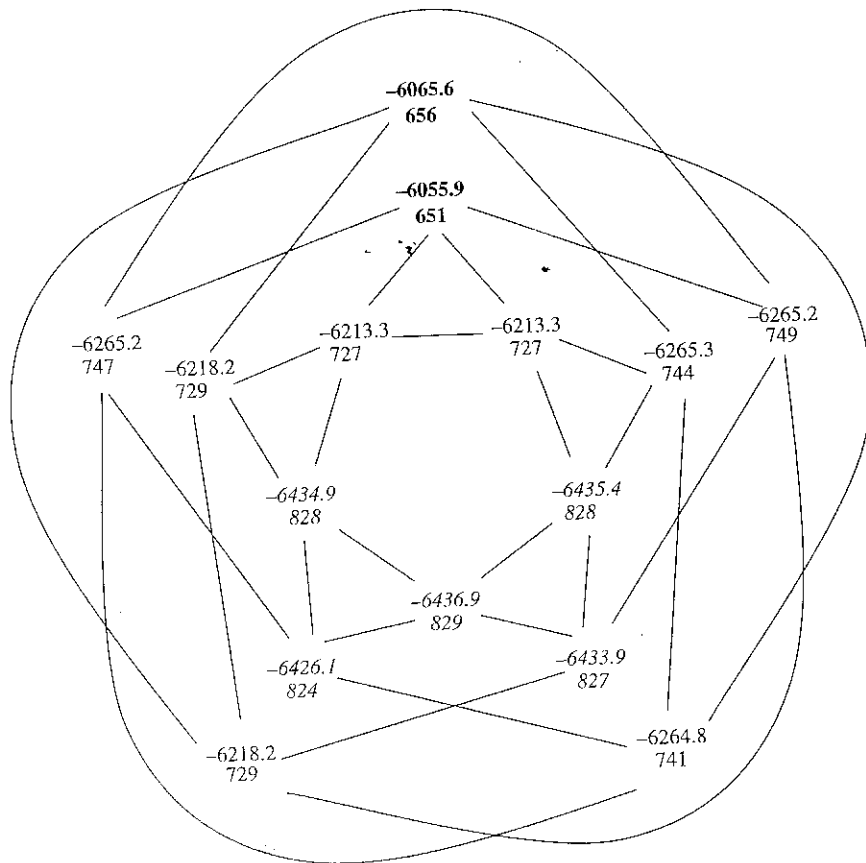


Fig. 3.15 *Local peaks in the tree space.* The log likelihood values (above) and parsimony scores (below) for the 15 trees of Fig. 3.14, shown in the same locations as the trees. The data set was simulated following the construction of Mossel and Vigoda (2005). It consists of 2000 nucleotide sites simulated under the JC69 model (Jukes and Cantor 1969) on the top two trees in Fig. 3.14: T_1 : ((a, b), c, (d, e)) and T_2 : ((a, e), c, (d, b)), with 1000 sites from each tree. Each external branch had length 0.01 and each internal branch had length 0.1. Trees T_1 and T_2 are two local optima under both parsimony and likelihood criteria.

almost identical under the likelihood and parsimony criteria. Similarly the data set poses serious computational problems for Bayesian algorithms (Huelsenbeck *et al.* 2001; Ronquist and Huelsenbeck 2003), as discussed by Mossel and Vigoda (2005).

One can design a branch-swapping algorithm under which trees T_1 and T_2 are neighbours. However, such an algorithm will define a different neighbourhood relationship among trees, and may have different local peaks or may have local peaks for different data sets. The problem should be more serious for larger trees with more species as the tree space is much larger. Similarly, in larger data sets with more sites,

the peaks tend to be higher and the valleys deeper, making it very difficult to traverse between peaks (Salter 2001).

3.2.5 Stochastic tree search

A hill-climbing algorithm that always goes uphill is called a *greedy algorithm*. Greedy algorithms may easily get stuck at a local peak. Some search algorithms attempt to overcome the problem of local peaks by allowing downhill moves. They can work under either parsimony or likelihood criteria. The first such algorithm is *simulated annealing* (Metropolis *et al.* 1953; Kirkpatrick *et al.* 1983), inspired by annealing in metallurgy, a technique involving heating and controlled cooling of a metal to reduce defects. The heat causes the atoms to move at random, exploring various configurations, while the slow cooling allows them to find configurations with low internal energy. In a simulated-annealing algorithm of optimization, the objective function is modified to have a flattened surface during the early (heated) stage of the search, making it easy for the algorithm to move between peaks. At this stage downhill moves are accepted almost as often as uphill moves. The ‘temperature’ is gradually reduced as the simulation proceeds, according to some ‘annealing schedule’. At the final stage of the algorithm, only uphill moves are accepted, as in a greedy algorithm. Simulated-annealing algorithms are highly specific to the problem, and their implementation is more art than science. The efficiency of the algorithm is affected by the neighbourhood function (branch-swapping algorithms) and the annealing schedule. Implementations in phylogenetics include Goloboff (1999) and Barker (2004) for parsimony, and Salter and Pearl (2001) for likelihood. Fleissner *et al.* (2005) used simulated annealing for simultaneous sequence alignment and phylogeny reconstruction.

A second stochastic tree-search algorithm is the *genetic algorithm*. A ‘population’ of trees is kept in every generation; these trees ‘breed’ to produce trees of the next generation. The algorithm uses operations that are similar to mutation and recombination to generate new trees from the current ones. ‘Survival’ of each tree into the next generation depends on its ‘fitness’, which is the optimality criterion. Lewis (1998), Katoh *et al.* (2001), and Lemmon and Milinkovitch (2002), among others, implemented genetic algorithms to search for the ML tree.

A third stochastic tree-search algorithm is the Bayesian Markov chain Monte Carlo algorithm. If all trees have the same prior probability, the tree with the highest posterior probability will be the maximum (integrated) likelihood tree. The Bayesian algorithm has a huge advantage over simulated-annealing or genetic algorithms: it is a statistical approach and produces not only a point estimate (the tree with the highest likelihood) but also a measure of uncertainties in the point estimate through posterior probabilities estimated during the search. Chapter 5 discusses Bayesian phylogenetics in detail.

3.3 Distance methods

Distance methods involve two steps: calculation of genetic distances between pairs of species and reconstruction of a phylogenetic tree from the distance matrix. The

simplest distance method is perhaps UPGMA (Sokal and Sneath 1963). This method is based on the molecular clock assumption and generates rooted trees. It is applicable to population data and seldom used to analyze species data, as the clock is often violated when the sequences are divergent. Below we describe two other methods that do not require the clock assumption: the least-squares (LS) and neighbour-joining (NJ) methods.

3.3.1 Least-squares method

The least-squares (LS) method takes the pairwise distance matrix as given data and estimates branch lengths on a tree by matching those distances as closely as possible, that is, by minimizing the sum of squared differences between the given and predicted distances. The predicted distance is calculated as the sum of branch lengths along the path connecting the two species. The minimum sum of squared differences then measures the fit of the tree to data (the distances) and is used as the tree score. This method was developed by Cavalli-Sforza and Edwards (1967), who called it the *additive-tree method*.

Let the distance between species i and j be d_{ij} . Let the sum of branch lengths along the path from species i to j on the tree be \hat{d}_{ij} . The LS method minimizes the sum, over all distinct pairs i and j , of the squared differences, $(d_{ij} - \hat{d}_{ij})^2$, so that the tree fits the distances as closely as possible. For example, the pairwise distances calculated under the K80 model (Kimura 1980) for the mitochondrial data of Brown *et al.* (1982) are shown in Table 3.3. These are taken as observed data. Now consider the tree ((human, chimpanzee), gorilla, orangutan) and its five branch lengths t_0, t_1, t_2, t_3, t_4 (Fig. 3.16). The predicted distance in the tree between human and chimpanzee is $t_1 + t_2$, and the predicted distance between human and gorilla is $t_1 + t_0 + t_3$, and so on. The sum of squared differences is then

$$\begin{aligned}
 S &= \sum_{i < j} (d_{ij} - \hat{d}_{ij})^2 \\
 &= (d_{12} - \hat{d}_{12})^2 + (d_{13} - \hat{d}_{13})^2 + (d_{14} - \hat{d}_{14})^2 + (d_{23} - \hat{d}_{23})^2 \\
 &\quad + (d_{24} - \hat{d}_{24})^2 + (d_{34} - \hat{d}_{34})^2.
 \end{aligned}
 \tag{3.2}$$

Table 3.3 Pairwise distances for the mitochondrial DNA sequences

1. Human				
2. Chimpanzee	0.0965			
3. Gorilla	0.1140	0.1180		
4. Orangutan	0.1849	0.2009	0.1947	
	1. Human	2. Chimpanzee	3. Gorilla	4. Orangutan

As the distances (d_{ij}) are calculated already, S is a function of the five unknown branch lengths t_0, t_1, t_2, t_3 , and t_4 . The values of branch lengths that minimize S are the LS estimates: $\hat{t}_0 = 0.008840, \hat{t}_1 = 0.043266, \hat{t}_2 = 0.053280, \hat{t}_3 = 0.058908, \hat{t}_4 = 0.135795$, with the corresponding tree score $S = 0.00003547$. Similar calculations can be done for the other two trees. Indeed the other two binary trees both converge to the star tree, with the internal branch length estimated to be 0; see Table 3.4. The tree ((human, chimpanzee), gorilla, orangutan) has the smallest S and is called the LS tree. It is the LS estimate of the true phylogeny.

Estimation of branch lengths on a fixed tree by the least-squares criterion uses the same principle as calculating the line of best fit $y = a + bx$ in a scatter plot. If there are no constraints on the branch lengths, the solution is analytical and can be obtained by solving linear equations (Cavalli-Sforza and Edwards 1967). Efficient algorithms that require less computation and less space have also been developed by Rzhetsky and Nei (1993), Bryant and Waddell (1998), and Gascuel (2000). Those algorithms may produce negative branch lengths, which are not meaningful biologically. If the branch lengths are constrained to be nonnegative, the problem becomes one of constrained optimization, which is much more expensive. However, the unconstrained LS criterion may be justified by ignoring the interpretation of branch lengths. The method will select the true tree as the LS tree if and only if the score for the true tree is smaller than the scores for all other trees. If this condition is satisfied in infinite data, the method will be guaranteed to converge to the true tree when more and more data are available. If the condition is satisfied in most finite data sets, the method will recover the true tree with high efficiencies. Thus the unconstrained method can be a well-behaved method of tree reconstruction without a sensible definition of branch lengths. While some simulation studies suggest that constraining branch lengths to be nonnegative

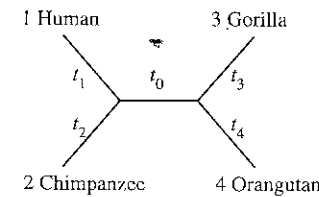


Fig. 3.16 A tree to demonstrate the least-squares criterion for estimating branch lengths.

Table 3.4 Least-squares branch lengths under K80 (Kimura 1980)

Tree	t_0 for internal branch	t_1 for H	t_2 for C	t_3 for G	t_4 for O	S_j
$\tau_1: ((H, C), G, O)$	0.008840	0.043266	0.053280	0.058908	0.135795	0.000035
$\tau_2: ((H, G), C, O)$	0.000000	0.046212	0.056227	0.061854	0.138742	0.000140
$\tau_3: ((H, G), C, O)$	as above					
$\tau_0: (H, G, C, O)$	as above					

leads to improved performance in tree reconstruction (e.g. Kuhner and Felsenstein 1994), most computer programs implement the LS method without the constraint. It is noted that when the estimated branch lengths are negative, they are most often close to zero.

The least-squares method described above uses equal weights for the different pairwise distances and is known as the ordinary least squares (OLS). It is a special case of the following generalized or weighted least squares (GLS) with weights $w_{ij} = 1$:

$$S = \sum_{i < j} w_{ij} (d_{ij} - \hat{d}_{ij})^2 \quad (3.3)$$

Fitch and Margoliash (1967) suggested the use of $w_{ij} = 1/d_{ij}^2$, while Bulmer (1990) used the variance: $w_{ij} = 1/\text{var}(d_{ij})$. However, such weighted LS methods were found not to work well in computer simulations, especially when the distances are large, presumably because the estimated variances are unreliable.

3.3.2 Neighbour-joining method

A criterion used for tree comparison, especially in distance methods, is the amount of evolution measured by the sum of branch lengths in the tree (Kidd and Sgaramella-Zonta 1971; Rzhetsky and Nei 1993). The tree with the smallest sum of branch lengths is known as the *minimum evolution* tree; see Desper and Gascuel (2005) for an excellent review of this class of methods.

Neighbour joining is a cluster algorithm based on the minimum-evolution criterion proposed by Saitou and Nei (1987). Because it is computationally fast and also produces reasonable trees, it is widely used. It is a divisive cluster algorithm (i.e. a star-decomposition algorithm), with the tree length (the sum of branch lengths along the tree) used as the criterion for tree selection at each step. It starts with a star tree and then joins two nodes, choosing the pair to achieve the greatest reduction in tree length. A new node is then created to replace the two nodes joined (Fig. 3.17), reducing the dimension of the distance matrix by one. The procedure is repeated until the tree is fully resolved. The branch lengths on the tree as well as the tree length are updated

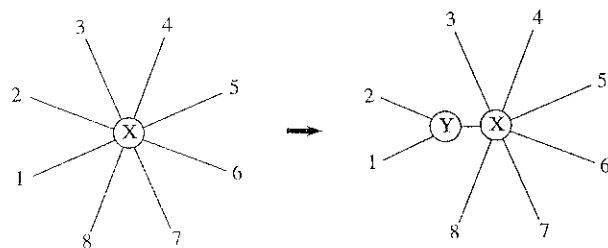


Fig. 3.17 The neighbour-joining method of tree reconstruction is a divisive cluster algorithm, dividing taxa successively into finer groups.

during every step of the algorithm. See Saitou and Nei (1987), Studier and Keppler (1988), and Gascuel (1994) for the update formulae.

A concern with the NJ method, and indeed with any distance-matrix method, is that large distances are poorly estimated, with large sampling errors. For very divergent sequences, the distance formulae may be even inapplicable. Some effort has been made to deal with the problem of large variances in large distance estimates. Gascuel (1997) modified the formula for updating branch lengths in the NJ algorithm to incorporate approximate variances and covariances of distance estimates. This method, called BIONJ, is close to the generalized least-squares method, and was found to outperform NJ, especially when substitution rates are high and variable among lineages. Another modification is the weighted neighbour-joining or WEIGHBOR method of Bruno *et al.* (2000). This used an approximate likelihood criterion for joining nodes to accommodate the fact that long distances are poorly estimated. Computer simulations suggest that WEIGHBOR produces trees similar to ML, and is more robust to the problem of long-branch attraction (see Subsection 3.4.4 below) than NJ (Bruno *et al.* 2000). Another idea, due to Ranwez and Gascuel (2002), is to improve distance estimates. When calculating the distance between a pair of sequences, those authors used a third sequence to break the long distance into two even parts and used ML to estimate three branch lengths on the tree of the three sequences; the original pairwise distance is then calculated as the sum of the two branch lengths. Simulations suggest that the improved distance, when combined with the NJ, BIONJ, and WEIGHBOR algorithms, led to improved topological accuracy.

3.4 Maximum parsimony

3.4.1 Brief history

When using allele frequencies (mainly for blood group alleles) to reconstruct the relationships among human populations, Edwards and Cavalli-Sforza (1963) (see also Cavalli-Sforza and Edwards 1967) suggested that a plausible estimate of the evolutionary tree is the one that invokes the minimum total amount of evolution. They called this method the *minimum-evolution method*. In modern terminology, the method discussed by Edwards and Cavalli-Sforza, when applied to discrete data, is identified with parsimony, while minimum evolution nowadays refers to methods minimizing the sum of branch lengths after correcting for multiple hits, as discussed in last section. For discrete morphological characters, Camin and Sokal (1965) suggested use of the minimum number of changes as a criterion for tree selection. For molecular data, minimizing changes on the tree to infer ancestral proteins appears most natural and was practised by many pioneers in the field, for example by Pauling and Zuckerkandl (1963) and Zuckerkandl (1964) as a way of 'restoring' ancestral proteins for 'palaeogenetic' studies of their chemical properties, and by Eck and Dayhoff (1966) to construct empirical matrices of amino acid substitution rates. Fitch (1971b) was the first to present a systematic algorithm to enumerate all and only the most-parsimonious reconstructions. Fitch's algorithm works on binary trees only. Hartigan

(1973) considered multifurcating trees as well and provided a mathematical proof for the algorithm. Since then, much effort has been made to develop fast algorithms for parsimony analysis of large data sets; see, e.g., Ronquist (1998), Nixon (1999), and Goloboff (1999).

3.4.2 Counting the minimum number of changes given the tree

The minimum number of character changes at a site is often called the *character length* or *site length*. The sum of character lengths over all sites in the sequence is the minimum number of required changes for the entire sequence and is called the *tree length*, *tree score*, or *parsimony score*. The tree with the smallest tree score is the estimate of the true tree, called the *maximum parsimony tree* or the *most parsimonious tree*. It is common, especially when the sequences are very similar, for multiple trees to be equally best; that is, they have the same minimum score and are all shortest trees.

Suppose the data for four species at a particular site are AAGG, and consider the minimum number of changes required by the two trees of Fig. 3.18. We calculate this number by assigning character states to the extinct ancestral nodes. For the first tree, this is achieved by assigning A and G to the two nodes, and one change ($A \leftrightarrow G$ on the internal branch) is required. For the second tree, we can assign either AA (shown) or GG (not shown) to the two internal nodes; in either case, a minimum of two changes is required. Note that the set of character states (nucleotides) at a site assigned to ancestral nodes is called an *ancestral reconstruction*. The total number of reconstructions at each site is thus $4^{(n-2)}$ for nucleotides or $20^{(n-2)}$ for amino acids as a binary tree of n species has $n-2$ interior nodes. The reconstruction that achieves the minimum number of changes is called the *most parsimonious reconstruction*. Thus, for the first tree, there is one single most parsimonious reconstruction, while for the second tree, two reconstructions are equally most parsimonious. The algorithm of Fitch (1971b) and Hartigan (1973) calculates the minimum number of changes and enumerates all the most parsimonious reconstructions at a site. We will not describe this algorithm here. Instead we describe in the next subsection a more general algorithm due to Sankoff (1975), which is very similar to the likelihood algorithm to be discussed in Chapter 4.

Some sites do not contribute to the discrimination of trees and are thus non-informative. For example, a constant site, at which the different species have the

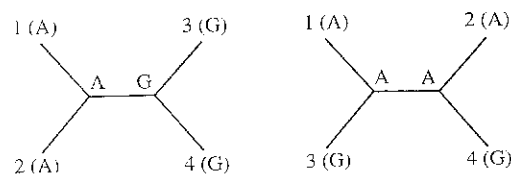


Fig. 3.18 Data AAGG at one site for four species mapped onto two alternative trees ((1, 2), 3, 4) and ((1, 3), 2, 4). The tree on the left requires a minimum of one change while the tree on the right requires two changes to explain the data.

same nucleotide, requires no change for any tree. Similarly a *singleton* site, at which two characters are observed but one is observed only once (e.g. TTTC or AAGA), requires one change for every tree and is thus not informative. Perhaps more strikingly, a site with data AAATAACAAG (for 10 species) is not informative either, as a minimum of three changes are required by any tree, which is also achieved by every tree by assigning A to all ancestral nodes. For a site to be a *parsimony-informative site*, at least two characters have to be observed, each at least twice. Note that the concepts of informative and noninformative sites apply to parsimony only. In distance or likelihood methods, all sites including the constant sites affect the calculation and should be included.

We often refer to the observed character states in all species at a site as a *site configuration* or *site pattern*. The above discussion means that for four species only three *site patterns* are informative: $xyxy$, $xyyx$, and $xyxx$, where x and y are any two distinct states. It is obvious that those three site patterns ‘support’ the three trees T_1 : ((1, 2), 3, 4); T_2 : ((1, 3), 2, 4); and T_3 : ((1, 4), 2, 3), respectively. Let the number of sites with those site patterns be n_1 , n_2 , and n_3 , respectively. Then T_1 , T_2 , or T_3 is the most parsimonious tree if n_1 , n_2 , or n_3 is the greatest among the three.

3.4.3 Weighted parsimony and transversion parsimony

The algorithm of Fitch (1971b) and Hartigan (1973) assumes that every change has the same cost. In weighted parsimony, different weights are assigned to different types of character changes. Rare changes are penalized more heavily than frequent changes. For example, transitions are known to occur at a higher rate than transversions and can be assigned a lower cost (weight). Weighted parsimony uses a *step matrix* or *cost matrix* to specify the cost of every type of change. An extreme case is *transversion parsimony*, which gives a penalty of 1 for a transversion but no penalty for a transition. Below, we describe Sankoff’s (1975) dynamic-programming algorithm, which calculates the minimum cost at a site and enumerates the reconstructions that achieve this minimum given any arbitrary cost matrix.

We first illustrate the basic idea of dynamic-programming algorithms using a fictitious example of a camel caravan travelling on the Silk Route. We start from the source city X , Chang-an in central China, to go to the destination Y , Baghdad in Iraq (Fig. 3.19). The route goes through four countries A , B , C , and D , and has to pass

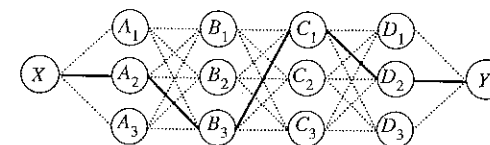


Fig. 3.19 Caravan-travelling example used for illustrating the dynamic-programming algorithm. It is required to determine the shortest route from X to Y , through four countries A , B , C , and D . Stops between neighbouring countries are connected, with their distances known.

one of three caravan stops in every country: A_1, A_2 or A_3 in country A ; B_1, B_2 or B_3 in country B ; and so on. We know the distance between any two stops in two neighbouring countries, such as XA_2 and A_1B_2 . We seek to determine the shortest distance and the shortest route from X to Y . An obvious strategy is to evaluate all possible routes, but this can be expensive as the number of routes (3^4 in the example) grows exponentially with the number of countries. A dynamic-programming algorithm answers many smaller questions, with the new questions building on answers to the old ones. First we ask for the shortest distances (from X) to stops A_1, A_2 , and A_3 in country A . These are just the given distances. Next we ask for the shortest distances to stops in country B , and then the shortest distances to stops in country C , and so on. Note that the questions at every stage are easy given the answers to the previous questions. For example, consider the shortest distance to C_1 , when the shortest distances to B_1, B_2 , and B_3 are already determined. This is just the smallest among the distances of the three routes going through B_1, B_2 , or B_3 , with the distance through B_j ($j = 1, 2, 3$) being the shortest distance (from X) to B_j plus the distance between B_j and C_1 . After the shortest distances to D_1, D_2 , and D_3 are determined, it is easy to determine the shortest distance to Y itself. It is important to note that adding another country to the problem will add another stage in the algorithm, so that the amount of computation grows linearly with the increase in number of countries.

We now describe Sankoff's algorithm. We seek to determine the minimum cost for a site on a given tree as well as the ancestral reconstruction that achieves the minimum cost. We use the tree of Fig. 3.20 as an example. The observed nucleotides at the site at the six tips are CCAGAA. Let $c(x, y)$ denote the cost of change from state x to state y , so $c(x, y) = 1$ for a transitional difference and $c(x, y) = 1.5$ for a transversion (Fig. 3.20).

Instead of the minimum cost for the whole tree, we calculate the minimum costs for many subtrees. We refer to a branch on the tree by the node it leads to or by the two nodes it connects. For example branch 10 is also branch 8–10 in Fig. 3.20. We say that each node i on the tree defines a subtree, referred to as subtree i , which consists of branch i , node i , and all its descendant nodes. For example, subtree 3 consists of branch i , node i , and all its descendant nodes. For example, subtree 3 consists of the single tip branch 10–3 while subtree 10 consists of branch 8–10 and nodes 10, 3, and 4. Define $S_i(x)$ as the minimum cost incurred on subtree i , given that the mother node of node i has state x . Thus $\{S_i(T), S_i(C), S_i(A), S_i(G)\}$ constitute a cost vector for subtree i at node i . They are like the shortest distances to stops in a particular country in the caravan example. We calculate the cost vectors for all nodes on the tree, starting with the tips and visiting a node only after we have visited all its descendant nodes. For a tip node i , the subtree is just the tip branch and the cost is simply read from the cost matrix. For example, tip 3 has the cost vector $\{1.5, 1.5, 0, 1\}$, meaning that the (minimum) cost of subtree 3 is 1.5, 1.5, 0, or 1, if mother node 10 has T, C, A, or G, respectively (Fig. 3.20). If the nucleotide at the tip is undetermined, the convention is to use the minimum cost among all compatible states (Fitch 1971b). For an interior node i , suppose its two daughter nodes are j and k . Then

$$S_i(x) = \min_y [c(x, y) + S_j(y) + S_k(y)]. \quad (3.4)$$

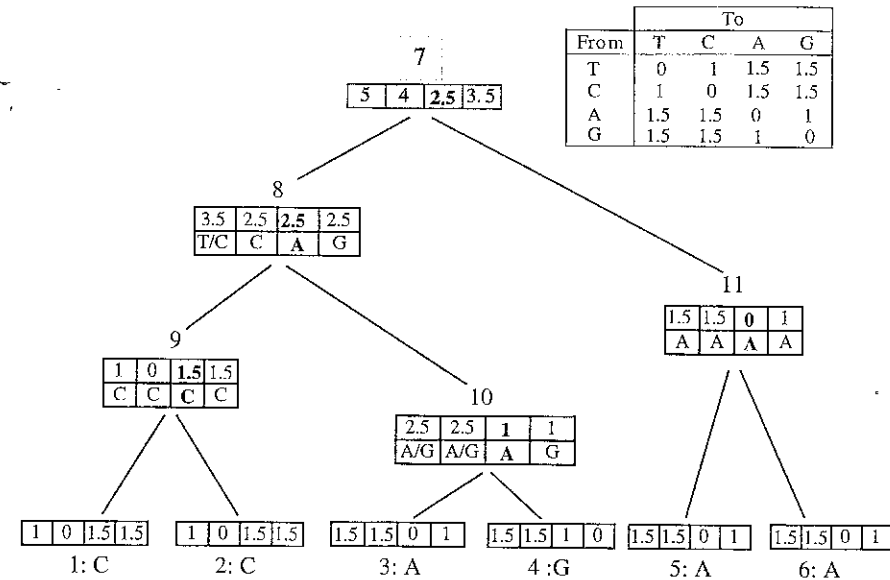


Fig. 3.20 Dynamic-programming algorithm for calculating the minimum cost and enumerating the most parsimonious reconstructions using weighted parsimony. The site has observed data CCAGAA. The cost vector at each node gives the minimum cost of the subtree induced by that node (which includes the node itself, its mother branch, and all its descendants), given that the mother node has nucleotides T, C, A, or G. The nucleotides at the node that achieved the minimum cost are shown below the cost vector. For example, the minimum cost of the subtree induced by node 3 (including the single branch 10–3) is 1.5, 1.5, 0, or 1, if node 10 has T, C, A, or G, respectively. The minimum cost of the subtree induced by node 10 (including branches 8–10 and nodes 10, 3 and 4) is 2.5, 2.5, 1, or 1, if node 8 has T, C, A, or G, respectively; the said minimum is achieved by node 10 having A/G, A/G, A, or G, respectively. The cost vectors are calculated for every node, starting from the tips and proceeding towards the root. At the root (node 7), the cost vector gives the minimum cost of the whole tree as 5, 4, 2.5, or 3.5, if the root has T, C, A, or G, respectively.

Note that subtree i consists of branch i plus subtrees j and k . Thus the minimum cost of subtree i is the cost along branch i , $c(x, y)$, plus the minimum costs of subtrees j and k , minimized over the state y at node i . We use $C_i(x)$ to record the state y that achieved the minimum.

Consider node 10 as an example, for which the cost vector is calculated to be $\{S_{10}(T), S_{10}(C), S_{10}(A), S_{10}(G)\} = \{2.5, 2.5, 1, 1\}$. Here the first entry, $S_{10}(T) = 2.5$, means that the minimum cost of subtree 10, given that mother node 8 has T, is 2.5. To see this, consider the four possible states at node 10: $y = T, C, A$, or G . The (minimum) cost on subtree 10 is $3 = 0 + 1.5 + 1.5, 4, 2.5$, or 2.5 , if node 10 has the state $y = T, C, A$, or G , respectively (and if node 8 has T). Thus the minimum is 2.5, achieved by node 10 having $y = A$ or G ; that is, $S_{10}(T) = 2.5$ and $C_{10}(T) = A$ or G .

(Fig. 3.20). This is the minimization over y in equation (3.4). Similarly, the second entry in the cost vector at node 10, $S_{10}(C) = 2.5$, means that the minimum cost for subtree 10, given that node 8 has C, is 2.5. This minimum is achieved by having $C_{10}(C) = A/G$ at node 10.

Similar calculations can be done for nodes 9 and 11. We now consider node 8, which has daughter nodes 9 and 10. The cost vector is calculated to be $\{3.5, 2.5, 2.5, 2.5\}$, meaning that the minimum cost of subtree 8 is 3.5, 2.5, 2.5, or 2.5, if mother node 7 has T, C, A, or G, respectively. Here we derive the third entry $S_8(A) = 2.5$, with mother node 7 having A. By using the cost vectors for nodes 9 and 10, we calculate the minimum cost on subtree 8 to be $5 = 1.5 + 1 + 2.5, 4, 2.5$, or 4.5, if node 8 has T, C, A, or G, respectively (and if mother node 7 has A). Thus $S_8(A) = 2.5$ is the minimum, achieved by node 8 having $C_8(A) = A$.

The algorithm is applied successively to all nodes in the tree, starting from the tips and moving towards the root. This upper pass calculates $S_i(x)$ and $C_i(x)$ for all nodes i except the root. Suppose the root has daughter nodes j and k and note that the whole tree consists of subtrees j and k . The minimum cost of the whole tree, given that the root has y , is $S_j(y) + S_k(y)$. This cost vector is $\{5, 4, 2.5, 3.5\}$, for $y = T, C, A, G$ at the root (Fig. 3.20). The minimum is 2.5, achieved by having A at the root. In general, if j and k are the daughter nodes of the root, the minimum cost for the whole tree is

$$S = \min_y [S_j(y) + S_k(y)]. \quad (3.5)$$

After calculation of $S_i(x)$ and $C_i(x)$ for all nodes through the upper pass, a down pass reads out the most parsimonious reconstructions. In our example, given A for the root, node 8 achieves the minimum for subtree 8 by having A. Given A at node 8, nodes 9 and 10 should have C and A, respectively. Similarly given A for the root, node 11 should have A. Thus the most parsimonious reconstruction at the site is $y_7y_8y_9y_{10}y_{11} = ACAA$, with the minimum cost 2.5.

3.4.4 Long-branch attraction

Felsenstein (1978a) demonstrated that the parsimony method can be statistically inconsistent under certain combinations of branch lengths on a four-species tree. In statistics, an estimation method is said to be consistent if the estimate converges to the true value of the parameter when the amount of data (sample size) approaches infinity. Otherwise the method is said to be inconsistent. In phylogenetics, we say that a tree-reconstruction method is inconsistent if the estimated tree topology converges to a wrong tree when the amount of data (the number of sites) increases to infinity.

The tree Felsenstein used has the characteristic shape shown in Fig. 3.21(a), with two long branches separated by a short internal branch. The estimated tree by parsimony, however, tends to group the two long branches together (Fig. 3.21b), a phenomenon known as 'long-branch attraction'. Using a simple model of character evolution, Felsenstein calculated the probabilities of observing sites with the three site patterns $xxyy, xyxy, yxyx$, where x and y are any two distinct characters, and

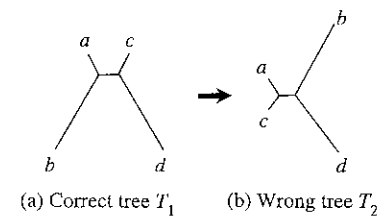


Fig. 3.21 Long-branch attraction. When the correct tree (T_1) has two long branches separated by a short internal branch, parsimony tends to recover a wrong tree (T_2) with the two long branches grouped in one clade.

found that $\Pr(xyxy) > \Pr(xxyy)$ when the two long branches are much longer than the three short branches. Thus with more and more sites in the sequence, it will be more and more certain that more sites will have pattern $xyxy$ than pattern $xxyy$, in which case parsimony will recover the wrong tree T_2 instead of the true tree T_1 (Fig. 3.21). The phenomenon has been demonstrated in many real and simulated data sets (see, e.g., Huelsenbeck 1998) and is due to the failure of parsimony to correct for parallel changes on the two long branches. Likelihood and distance methods using simplistic and unrealistic evolutionary models show the same behaviour.

3.4.5 Assumptions of parsimony

Some concerns may be raised about the parsimony reconstruction of ancestral states. First, the method ignores branch lengths. Some branches on the tree are longer than others, meaning that they have accumulated more evolutionary changes than other branches. It is thus illogical to assume that a change is as likely to occur on a long branch as on a short one, as parsimony does, when character states are assigned to ancestral nodes on the tree. Second, the simple parsimony criterion ignores different rates of changes between nucleotides. Such rate differences are taken into account by weighted parsimony through the use of a step matrix, although determining the appropriate weights may be nontrivial. In theory, how likely a change is to occur on a particular branch should depend on the length of the branch as well the relative rate of the change. Attempts to derive appropriate weights for the observed data lead naturally to the likelihood method, which uses a Markov-chain model to describe the nucleotide substitution process, relying on probability theory to accommodate unequal branch lengths, unequal substitution rates between nucleotides, and any other features of the evolutionary process. This is the topic of the next chapter.