CHAPTER 4

# Maximum likelihood methods



## 4.1 Introduction

In this chapter, we will discuss likelihood calculation for multiple sequences on a phylogenetic tree. As indicated at the end of last chapter, this is a natural extension to the parsimony method, when we want to incorporate differences in branch lengths and in substitution rates between nucleotides. Likelihood calculation on a tree is also a natural extension to estimation of the distance between two sequences, discussed in Chapter 1. Indeed Chapter 1 has covered the general principles of Markov chain theory and maximum likelihood (ML) estimation needed in this chapter.

It may be beneficial to distinguish two applications of ML in phylogenetic analysis. The first is estimation of parameters in the evolutionary model and testing of hypotheses concerning the evolutionary process when the tree topology is known or fixed. The likelihood method, with its nice statistical properties, provides a powerful and flexible framework for such analysis (e.g. Stuart et al. 1999). The second is inference of the tree topology. The log likelihood for each tree is maximized by optimizing branch lengths and other substitution parameters, and the optimized log likelihood is used as a tree score for comparing different trees. This second application of ML corresponds to comparison of many statistical models. It involves complexities, which will be discussed in Chapter 5.

## 4.2 Likelihood calculation on tree

### 4.2.1 *Data, model, tree, and likelihood*

The likelihood is defined as the probability of observing the data when the parameters are given, although it is considered to be a function of the parameters. The data consist of $s$ aligned homologous sequences, each $n$ nucleotides long, and can be represented as an $s \times n$ matrix $X = \{x_{jh}\}$, where $x_{jh}$ is the $h$th nucleotide in the $j$th sequence. Let $\mathbf{x}_h$ denote the $h$th column in the data matrix. To define the likelihood, we have to specify the model by which the data are generated. Here we use the K80 nucleotide substitution model (Kimura 1980). We assume that different sites evolve independently of each other and evolution in one lineage is independent of other lineages. We use the tree of five species of Figure 4.1 as an example to illustrate the likelihood calculation. The observed data at a particular site, TCACC, are shown. The ancestral nodes are numbered 0, 6, 7, and 8, with 0 being the root. The length of the branch leading to node $i$ is denoted $t_i$, defined as the expected number of nucleotide substitutions per site. The parameters in the model include the branch lengths and the transition/transversion rate ratio $\kappa$, collectively denoted $\theta = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, \kappa\}$.

*Molecular Evolution: A Statistical Approach*. Ziheng Yang. © Ziheng Yang 2014.
Published 2014 by Oxford University Press.

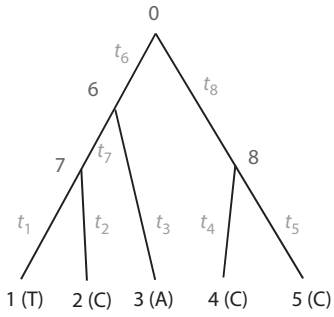


**Fig. 4.1** A tree of five species used to demonstrate calculation of the likelihood function. The nucleotides observed at the tips at a site are shown. Branch lengths $t_1$–$t_8$ are measured by the expected number of nucleotide substitutions per site.

Because of the assumption of independent evolution among sites, the probability of the whole dataset (the alignment) is the product of the probabilities of data at individual sites:

$$L(\theta) = f(X|\theta) = \prod_{h=1}^{n} f(\mathbf{x}_h \,|\theta)\,. \tag{4.1}$$

Equivalently the log likelihood is a sum over sites in the sequence

$$\ell = \log\{L(\theta)\} = \sum_{h=1}^{n} \log\{f(\mathbf{x}_h \,|\theta)\}. \tag{4.2}$$

Here we consider calculation of $\ell$ when parameters $\theta$ are given. We focus on one site, with the data $\mathbf{x}_h$ = TCACC, say. We use $x_i$ to represent the state at ancestral node $i$, and suppress the subscript $h$. Since the data at the site can result from any combination of ancestral nucleotides $x_0x_6x_7x_8$, calculation of $f(\mathbf{x}_h)$ has to sum over all possible nucleotide combinations for the extinct ancestors (nodes 0, 6, 7, and 8)

$$f(\mathbf{x}_h|\theta) = \sum_{x_0}\sum_{x_6}\sum_{x_7}\sum_{x_8}\left[\pi_{x_0} p_{x_0x_6}(t_6) p_{x_6x_7}(t_7) p_{x_7T}(t_1) p_{x_7C}(t_2) p_{x_6A}(t_3) p_{x_0x_8}(t_8) p_{x_8C}(t_4) p_{x_8C}(t_5)\right]. \tag{4.3}$$

Here the summation over each of $x_0, x_6, x_7, x_8$ is over the four nucleotides T, C, A, G. The quantity in the square brackets is the probability of data TCACC for the tips and $x_0x_6x_7x_8$ for the ancestral nodes. This is equal to the probability that the root (node 0) has $x_0$, which is given by $\pi_{x_0} = {}^1/_4$ under K80, multiplied by eight transition probabilities along the eight branches of the tree. We discussed calculation of the transition probabilities in Chapter 1; for example, those under K80 are given in equation (1.10).

Note that given $\theta$, we are able to calculate $f(\mathbf{x}_h|\theta)$ and the log likelihood $\ell$. The ML method then estimates $\theta$ by maximizing $\ell$, often using numerical optimization algorithms (to be discussed in §4.5).

## 4.2.2 *The pruning algorithm*

### 4.2.2.1 *Horner's rule and the pruning algorithm*

Summing over all combinations of ancestral states is expensive because there are $4^{s-1}$ possible combinations for $s-1$ interior nodes. The situation is even worse for amino acid or codon sequences as there will then be $20^{s-1}$ or $61^{s-1}$ possible combinations. An important technique that is useful in calculating such sums is to identify common factors and calculate them only once. This is known as the *nesting rule* or *Horner's rule*, published by

the Irish mathematician William Horner in 1830. The rule was also published in 1820 by a London watchmaker, Theophilus Holdred, and the same principle had been used in 1303 by the Chinese mathematician Zhu Shijie (朱世杰). By this rule, an $n$th-order polynomial can be calculated with only $n$ multiplications and $n$ additions. For example, a naïve calculation of $1 + 2x + 3x^2 + 4x^3$, as $1 + 2\cdot x + 3\cdot x\cdot x + 4\cdot x\cdot x\cdot x$, requires six multiplications and three additions. However, by writing it as $1 + x\cdot(2 + x\cdot(3 + 4\cdot x))$, only three multiplications and three additions are needed. As another example, $\sum_{i=1}^{10}\sum_{j=1}^{10}(x_i y_{ij}) = \sum_{i=1}^{10}\left[x_i\left(\sum_{j=1}^{10} y_{ij}\right)\right]$, but the left-hand side involves 100 multiplications and 99 additions while the right-hand side involves only ten multiplications and 99 additions.

If we apply the nesting rule and move the summation signs in equation (4.3) to the right as far as possible, we get

$$f(\mathbf{x}_h|\theta) = \sum_{x_0}\pi_{x_0}\left\{\sum_{x_6} p_{x_0x_6}(t_6)\left[\left(\sum_{x_7} p_{x_6x_7}(t_7)p_{x_7T}(t_1)p_{x_7C}(t_2)\right)p_{x_6A}(t_3)\right]\right\} \times\left[\sum_{x_8} p_{x_0x_8}(t_8)p_{x_8C}(t_4)p_{x_8C}(t_5)\right]. \tag{4.4}$$

Thus we sum over $x_7$ before $x_6$, and sum over $x_6$ and $x_8$ before $x_0$. In other words, we sum over ancestral states at a node only after we have done so for all its descendant nodes.

The pattern of parentheses and the occurrences of the tip states in equation (4.4), in the form [(T, C), A], [C, C], match the tree of Figure 4.1. This is no coincidence. Indeed calculation of $f(\mathbf{x}_h|\theta)$ by equation (4.4) constitutes the *pruning algorithm* of Felsenstein (1973b, 1981). This is a variant of the dynamic programming algorithm discussed in §3.4.3. Its essence is to successively calculate probabilities of data at the site on many subtrees. Define $L_i(x_i)$ to be the conditional probability of observing data at the tips that are descendants of node $i$, given that the nucleotide at node $i$ is $x_i$. For example, tips 1, 2, 3 are descendants of node 6, so $L_6(\mathrm{T})$ is the probability of observing $x_1x_2x_3 = \mathrm{TCA}$, given that node 6 has the state $x_6 = \mathrm{T}$. With $x_i = \mathrm{T}, \mathrm{C}, \mathrm{A}, \mathrm{G}$, we calculate a vector of conditional probabilities for each node $i$. In the literature, the conditional probability $L_i(x_i)$ is often referred to as the 'partial likelihood' or 'conditional likelihood'; these are misnomers since likelihood refers to the probability of the whole dataset and not probability of data at a single site or part of a single site.

If node $i$ is a tip, its descendant tips include tip $i$ itself only, so that $L_i(x_i) = 1$ if $x_i$ is the observed nucleotide, or 0 otherwise. If node $i$ is an interior node with daughter nodes $j$ and $k$, we have

$$L_i(x_i) = \left[\sum_{x_j} p_{x_ix_j}(t_j)L_j(x_j)\right]\times\left[\sum_{x_k} p_{x_ix_k}(t_k)L_k(x_k)\right]. \tag{4.5}$$

This is a product of two terms, corresponding to the two daughter nodes $j$ and $k$. Note that tips that are descendants of node $i$ must be descendants of either $j$ or $k$. Thus the probability $L_i(x_i)$ of observing all descendant tips of node $i$ (given the state $x_i$ at node $i$) is equal to the probability of observing data at the descendant tips of node $j$ (given $x_i$) times the probability of observing data at the descendant tips of node $k$ (given $x_i$). These are the two terms in the two pairs of brackets in equation (4.5), respectively. For example, node $i = 6$ has daughter nodes $j = 7$ and $k = 3$, and descendant tip nodes 1, 2, 3. The probability of observing $x_1x_2x_3$ given $x_6$ is the probability of observing $x_1x_2$ given $x_6$, times the probability of observing $x_3$ given $x_6$. Given the state $x_i$ at node $i$, the two parts of the tree down node $i$ are independent. If node $i$ has more than two daughter nodes, $L_i(x_i)$ will

be a product of as many terms. Now consider the first term, the term in the first pair of brackets, which is the probability of observing data at descendant tips of node $j$ (given the state $x_i$ at node $i$). This is the probability $p_{x_i x_j}(t_j)$ that $x_i$ will become $x_j$ over branch length $t_j$ times the probability $L_j(x_j)$ of observing the tips of node $j$ given the state $x_j$ at node $j$, summed over all possible states $x_j$.

We calculate the conditional probability vector $L_i(x_i)$ for node $i$ only after the vectors $L_j(x_j)$ and $L_k(x_k)$ for its daughter nodes $j$ and $k$ have been calculated. Thus we calculate the probabilities of data $x_1x_2$ down node 7, then the probabilities of data $x_1x_2x_3$ down node 6, then the probabilities of data $x_4x_5$ down node 8, and finally the probabilities of the whole data $x_1x_2x_3x_4x_5$ down node 0. The calculation proceeds from the tips of the tree towards the root, visiting each node only after all its descendant nodes have been visited. In computer science, this way of visiting all nodes on the tree is known as the *post-order tree traversal* (as opposed to *pre-order tree traversal*, in which ancestors are visited before descendants). After visiting all nodes on the tree and calculating the probability vector for the root $L_0(x_0)$, the probability of data at the site is given as

$$f(\mathbf{x}_h|\theta) = \sum\nolimits_{x_0} \pi_{x_0} L_0(x_0). \tag{4.6}$$

Note that $\pi_{x_0}$ is the (prior) probability that the nucleotide at the root is $x_0$, given by the equilibrium frequency of the nucleotide $x_0$ under the model.

*Example 4.1*. We use the tree of Figure 4.1 to provide a numerical example of the calculation using the pruning algorithm at one site (Figure 4.2). For definiteness, we fix internal

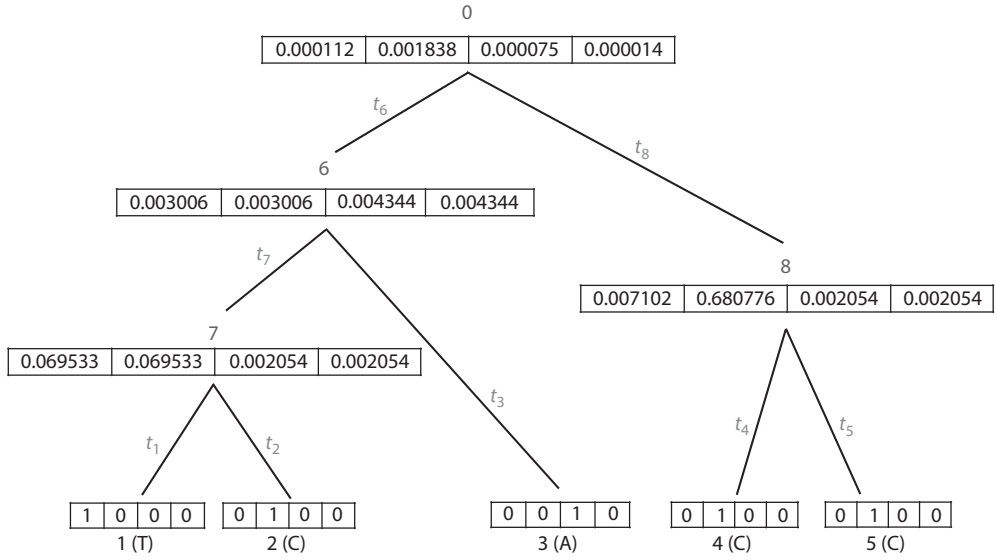


**Fig. 4.2** Illustration of the pruning algorithm for likelihood calculation when the branch lengths and other parameters are fixed. The tree of Figure 4.1 is reproduced, showing the vector of conditional probabilities at each node. The four elements in the vector at each node are the probabilities of observing data at the descendant tips, given that the node has T, C, A, or G, respectively. For example, 0.069533 for node 7 is the probability of observing data $x_1x_2$ = TC at tips 1 and 2, given that node 7 has T. The K80 model is assumed, with $\kappa = 2$. The branch lengths are fixed at 0.1 for the internal branches and 0.2 for the external branches. The transition probability matrices are shown in the text.

branch lengths at $t_6 = t_7 = t_8 = 0.1$ and external branch lengths at $t_1 = t_2 = t_3 = t_4 = t_5 = 0.2$. We also set $\kappa = 2$. The two transition probability matrices are as follows, in which the $ij$th element is $p_{ij}(t)$, with the nucleotides ordered T, C, A, and G (see equation (1.10) for K80):

$$P(0.1) = \begin{bmatrix} 0.906563 & 0.045855 & 0.023791 & 0.023791 \\ 0.045855 & 0.906563 & 0.023791 & 0.023791 \\ 0.023791 & 0.023791 & 0.906563 & 0.045855 \\ 0.023791 & 0.023791 & 0.045855 & 0.906563 \end{bmatrix},$$

$$P(0.2) = \begin{bmatrix} 0.825092 & 0.084274 & 0.045317 & 0.045317 \\ 0.084274 & 0.825092 & 0.045317 & 0.045317 \\ 0.045317 & 0.045317 & 0.825092 & 0.084274 \\ 0.045317 & 0.045317 & 0.084274 & 0.825092 \end{bmatrix}.$$

Consider node 7, which has daughter nodes 1 and 2. Using equation (4.5), we obtain the first entry in the probability vector as $L_7(\mathrm{T}) = p_{TT}(0.2) \times p_{TC}(0.2) = 0.825092 \times 0.084274 = 0.069533$. This is the probability of observing T and C at tips 1 and 2, given that node 7 has T. The other entries, $L_7(\mathrm{C}), L_7(\mathrm{A})$, and $L_7(\mathrm{G})$, can be calculated similarly, as can the vector for node 8. Next the vector at node 6 can be calculated, by using the conditional probability vectors at daughter nodes 7 and 3. Finally, we calculate the vector for node 0, the root. The first entry, $L_0(\mathrm{T}) = 0.000112$, is the probability of observing the descendant tips (1, 2, 3, 4, 5) of node 0, given that node 0 has $x_0 = \mathrm{T}$. Equation (4.5) gives this as the product of two terms. The first term, $\sum_{x_6} p_{x_0 x_6}(t_6) L_6(x_6)$, sums over $x_6$ and is the probability of observing data TCA at the tips 1, 2, 3, given that node 0 has T. This is $0.906563 \times 0.003006 + 0.045855 \times 0.003006 + 0.023791 \times 0.004344 + 0.023791 \times 0.004344 = 0.003070$. The second term, $\sum_{x_8} p_{x_0 x_8}(t_8) L_8(x_8)$, is the probability of observing data CC at tips 4 and 5, given that node 0 has T. This is $0.906563 \times 0.007102 + 0.045855 \times 0.680776 + 0.023791 \times 0.002054 + 0.023791 \times 0.002054 = 0.037753$. The product of the two terms gives $L_0(\mathrm{T}) = 0.00011237$. Other entries in the vector for node 0 can be similarly calculated. Finally application of equation (4.6) gives the probability of data at the site as $f(\mathbf{x}_h|\theta) = 0.000509843$, with $\log\{f(\mathbf{x}_h|\theta)\} = -7.581408$. □

#### 4.2.2.2 *Savings on computation*

The pruning algorithm is a major time saver. As in the dynamic programming algorithm discussed in §3.4.3, in the pruning algorithm the amount of computation required by one calculation of the likelihood increases linearly with the number of nodes or the number of species, even though the number of combinations of ancestral states increases exponentially.

Some other obvious savings may be mentioned here as well. First, the same transition probability matrix is used for all sites or site patterns in the sequence and may be calculated only once for each branch. Second, if two sites have the same data, the probabilities of observing them will be the same and need be calculated only once. Collapsing sites into *site patterns* thus leads to a saving in computation, especially if the sequences are highly similar so that many sites have identical patterns. Under JC69, some sites with different data, such as TCAG and TGCA, also have the same probability of occurrence and can be collapsed further (Saitou and Nei 1986). The same applies to K80, although the saving is not as much as under JC69. It is also possible to collapse *partial site patterns* corresponding to subtrees (e.g. Kosakovsky Pond and Muse 2004). For example, consider the tree of Figure 4.1 and two sites with data TCACC and TCACT. The conditional probability vectors

for interior nodes 6 and 7 are the same (because the data for species 1, 2, and 3 are the same at the two sites) and can be calculated only once. However, such collapsing of partial site patterns depends on the tree topology and involves an overhead for bookkeeping. Reports vary as to its effectiveness.

#### 4.2.2.3 *Hadamard conjugation*

It is fitting to mention here an alternative method, called *Hadamard conjugation*, for calculating the site pattern probabilities and thus the likelihood. The Hadamard matrix is a square matrix consisting of –1 and 1 only. With –1 and 1 representing grey and dark grey, respectively, the matrix is useful for designing pavements. Indeed it was invented by the English mathematician James Sylvester (1814–1897) under the name 'anallagmatic pavement' and later studied by the French mathematician Jacques Hadamard (1865–1963). It was introduced to molecular phylogenetics by Hendy and Penny (1989), who used it to transform branch lengths on an unrooted tree to the site pattern probabilities, and vice versa. The transformation or conjugation works for binary characters or under Kimura's (1981) 3ST model of nucleotide substitution, which assumes three substitution types: one rate for transitions and two rates for transversions. It is computationally feasible for small trees with $< 20$ species, and is sometimes useful in theoretical analysis of phylogenetic methods (Felsenstein 2004; Hendy 2005).

### 4.2.3 *Time reversibility, the root of the tree, and the molecular clock*

As discussed in Chapter 1, most substitution models used in molecular phylogenetics describe time-reversible Markov chains. For such chains, the transition probabilities satisfy $\pi_i p_{ij}(t) = \pi_j p_{ji}(t)$ for any $i$, $j$, and $t$. Reversibility means that the chain will look identical probabilistically whether we view the chain with time running forward or backward. An important consequence of reversibility is that the root can be moved arbitrarily along the tree without affecting the likelihood. This is called the *pulley principle* by Felsenstein (1981). For example, substituting $\pi_{x_6} p_{x_6 x_0}(t_6)$ for $\pi_{x_0} p_{x_0 x_6}(t_6)$ in equation (4.3), and noting $\sum_{x_0} p_{x_6 x_0}(t_6) p_{x_0 x_8}(t_8) = p_{x_6 x_8}(t_6 + t_8)$, we have, by the Chapman–Kolmogorov theorem (equation (1.5)),

$$f(\mathbf{x}_h|\theta) = \sum_{x_6}\sum_{x_7}\sum_{x_8}\left[\pi_{x_6} p_{x_6 x_7}(t_7) p_{x_6 x_8}(t_6 + t_8) p_{x_7 T}(t_1) p_{x_7 C}(t_2) p_{x_6 A}(t_3) p_{x_8 C}(t_4) p_{x_8 C}(t_5)\right]. \quad (4.7)$$

This is the probability of the data if the root is at node 6, and the two branches 0–6 and 0–8 are merged into one branch 6–8, of length $t_6 + t_8$. The resulting tree is shown in Figure 4.3, with the root at node 6.

Equation (4.7) also highlights the fact that the model is over-parametrized in Figure 4.1, with one branch length too many. The likelihood is the same for any combinations of $t_6$

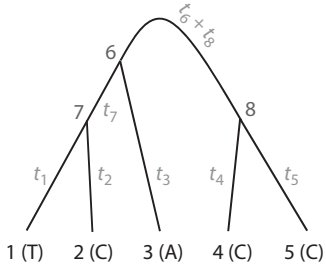


**Fig. 4.3** The ensuing unrooted tree when the root is moved from node 0 to node 6 in the tree of Figure 4.1.

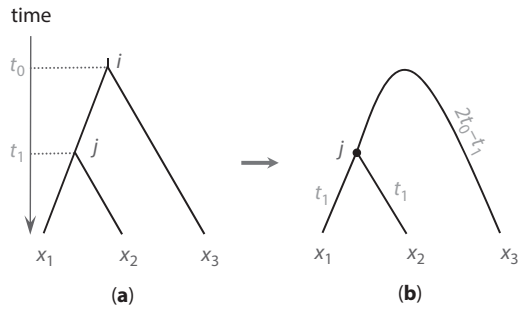


**Fig. 4.4** (**a**) On a rooted tree for three species and under the clock, the model involves two parameters $t_0$ and $t_1$, measured by the expected number of substitutions per site from the ancestral node to the present time. The likelihood calculation has to sum over ancestral states $i$ and $j$ at the two ancestral nodes. (**b**) If the model is reversible, the same calculation can be achieved by moving the root to the ancestor of species 1 and 2, and summing over state $j$ at the new root; the tree then becomes a star tree with three branches of lengths $t_1$, $t_1$, and $2t_0 - t_1$.

and $t_8$ as long as $t_6 + t_8$ is the same. The data do not contain information to estimate $t_6$ and $t_8$ separately and only their sum is estimable. Thus with reversibility, only unrooted trees can be identified if the molecular clock (rate constancy over time) is relaxed and every branch has its own rate.

If we assume the molecular clock, however, the root of the tree can indeed be identified. With a single rate throughout the tree, every tip is equidistant from the root, and the natural parameters are the ages of the ancestral nodes, measured by the expected number of substitutions per site. A binary tree of $s$ species has $s - 1$ internal nodes, and thus $s - 1$ branch length parameters under the clock model. An example for three species is shown in Figure 4.4a. The branch length is then given as the difference of the ages of the two nodes at the ends of the branch. Given the branch lengths, likelihood calculation or the pruning algorithm proceed as before.

Even under the clock, the pulley principle may be used to simplify the likelihood calculation in theoretical studies of small trees. For example, likelihood calculation on the tree of Figure 4.4a involves summing over the ancestral states $i$ and $j$ at the two ancestral nodes. However, it is simpler to move the root to the common ancestor of species 1 and 2, so that one has to sum over ancestral states at only one node (Figure 4.4b). The probability of data $x_1x_2x_3$ at a site becomes

$$\begin{aligned} f(x_1x_2x_3|\theta) &= \sum_i \sum_j \pi_i p_{ij}(t_0 - t_1) p_{jx_1}(t_1) p_{jx_2}(t_1) p_{ix_3}(t_0) \\ &= \sum_j \pi_j p_{jx_1}(t_1) p_{jx_2}(t_1) p_{jx_3}(2t_0 - t_1), \end{aligned} \tag{4.8}$$

where $\theta = \{t_0, t_1\}$ are the parameters under the model. Such arbitrary moving of the root is very similar to the case of two sequences discussed in equation (1.70) and Figure 1.10.

### 4.2.4 *A numerical example: phylogeny of apes*

We use the sequences from the 12 proteins encoded by the heavy strand of the mitochondrial genome from seven ape species. The data are a subset of the mammalian sequences analysed by Cao et al. (1998). The 12 proteins are concatenated into one long sequence and analysed as one dataset as they appear to have similar substitution patterns. The other protein in the genome, ND6, is not included as it is encoded by the opposite strand of the DNA with quite different base compositions. The species and the GenBank accession numbers for the sequences are human (*Homo sapiens,* D38112), common chimpanzee (*Pan troglodytes,* D38113), bonobo chimpanzee (*Pan paniscus,* D38116), gorilla (*Gorilla gorilla,* D38114), Bornean orangutan (*Pongo pygmaeus pygmaeus,* D38115), Sumatran orangutan

(*Pongo pygmaeus abelii,* X97707), and gibbon (*Hylobates lar,* X99256). Alignment gaps are removed, with 3, 331 amino acids in the sequence.

There are 945 binary unrooted trees for seven species, so we evaluate them exhaustively. We assume the empirical MTMAM model for mammalian mitochondrial proteins (Yang et al. 1998). The ML tree is shown in Figure 4.5, which has the log likelihood score $\ell = -14, 558.59$. The worst binary tree has the score –15, 769.00, while the star tree has the score –15, 777.60. Figure 4.6 shows that the same tree (the one of Figure 4.5) has the highest log likelihood, the shortest tree length by parsimony, and also the shortest likelihood tree length (the sum of maximum likelihood estimates (MLEs) of branch lengths). Thus ML, maximum parsimony, and minimum evolution all selected the same best tree for this dataset. (Note that minimum evolution normally estimates branch lengths by applying

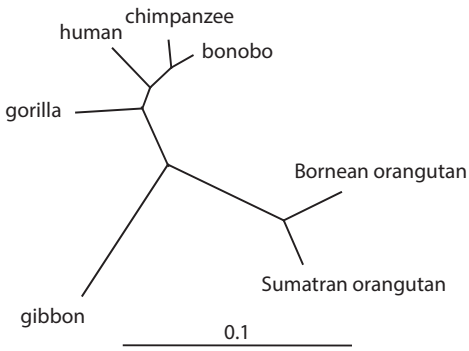


**Fig. 4.5** The ML tree for seven ape species estimated from the 12 mitochondrial proteins. Branches are drawn in proportion to their lengths, measured by the number of amino acid substitutions per site. The MTMAM model is assumed.

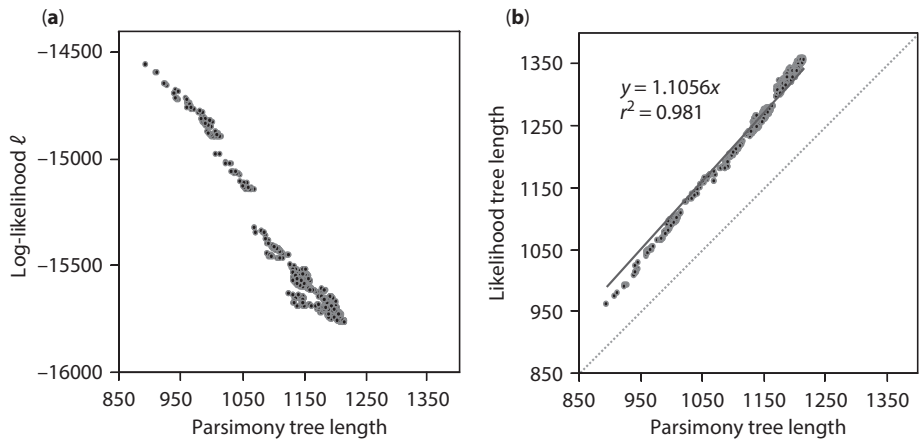


**Fig. 4.6** Different criteria for tree selection calculated for all 945 binary unrooted trees for the mitochondrial protein data. (**a**) The log likelihood score $\ell$ is plotted against the parsimony tree length. (**b**) The likelihood tree length is plotted against the parsimony tree length. The likelihood tree length is measured by the estimated number of amino acid substitutions on the tree, calculated as the sum of estimated branch lengths multiplied by the number of sites. The parsimony tree length is the minimum number of changes and is thus an under-count. The underestimation is slightly more serious for the poor trees (with large tree lengths) but is nearly proportional. All three criteria (parsimony tree length, likelihood tree length, and log likelihood score) select the tree of Figure 4.5 as the best estimate.

least squares to a distance matrix while here I used ML to estimate branch lengths.) Similarly, use of the Poisson model in the likelihood analysis, assuming the same rate between any two amino acids, gives the same tree as the ML tree, with $\ell = -16,566.60$.

### 4.2.5 *Amino acid, codon, and RNA models*

The discussions up to now assume models of nucleotide substitution applied to noncoding DNA sequences. The same theory, including the pruning algorithm, can be applied in a straightforward manner to analyse protein sequences under models of amino acid substitution (Bishop and Friday 1985, 1987; Kishino et al. 1990) or protein-coding DNA sequences under models of codon substitution (Goldman and Yang 1994; Muse and Gaut 1994). Discussions of such models are given in Chapter 2. A difference is that the substitution rate and transition probability matrices are of sizes $20 \times 20$ for amino acids or $61 \times 61$ for codons (as there are 61 sense codons in the universal genetic code), instead of $4 \times 4$ for nucleotides. Furthermore, the summation over ancestral states is now over all ancestral amino acids or codons. As a result, likelihood computation under amino acid and codon models is much more expensive than under nucleotide models.

For phylogenetic analysis of ribosomal RNA (rRNA) genes, Markov models with 16 states for dinucleotides have been developed to describe substitutions in the helical (stem) regions where complementary nucleotides tend to change together (Schoeniger and von Haeseler 1994; Rzhetsky 1995; Tillier and Collins 1998; see also Siepel and Haussler 2004). Assumptions are made to reduce the number of parameters in the $16 \times 16$ rate matrix. Some studies confirm the benefit of taking RNA secondary structures into account in the model, with improved accuracy and robustness of phylogenetic reconstruction (Telford et al. 2005; Keller et al. 2010). Nevertheless, Letsch and Kjer (2011) pointed out that RNA covariation models often fail to recover reasonable trees because the highly divergent single-stranded loop regions contribute much of the information for the analysis while the covarying sites in the stem regions are effectively down-weighted. They advise caution in the uncritical application of RNA substitution models and suggest that loop regions should be assessed for substitutional saturation and alignment difficulties, and should be removed if they appear to contain too much noise.

### *4.2.6 *Missing data, sequence errors, and alignment gaps*

#### 4.2.6.1 *General theory*

The likelihood function provides a natural framework for accommodating incompletely determined nucleotides (ambiguities) and nucleotide changes caused by sequencing errors or DNA degradation in ancient DNA. Let $X$ be the observed data, which may involve ambiguities or sequence errors, and $Y$ be the unknown true alignment with fully determined nucleotides, which is the full exact data that we wish to have observed. The likelihood is the probability of $X$, and not of $Y$. The two are related by $f(X|Y, \psi)$, the probability of observing $X$ given the full data $Y$, which is specified by the mechanism by which missing data or sequence errors arise, and may involve unknown parameters $\psi$. The likelihood is given by the law of total probability

$$L(\theta, \psi) = f(X|\theta, \psi) = \sum_Y f(Y|\theta) f(X|Y, \psi), \tag{4.9}$$

where the summation (integration) is over all possible full data $Y$ and where $f(Y|\theta)$ is the probability of the full data $Y$, as calculated in equations (4.1)–(4.4).

* indicates a more difficult or technical section.

We assume that the ambiguities or sequencing errors occur independently at different sites in the sequence. This assumption may be plausible for ambiguities or missing data, but appears quite unrealistic for sequence errors, which tend to be related to local sequence features (e.g. Nakamura et al. 2011). Under this independence assumption, $f(X|Y, \psi)$ is a product of probabilities at different sites and so is $f(Y|\theta)$. Thus

$$L(\theta, \psi) = f(X|\theta, \psi) = \prod_{h=1}^{n} f(\mathbf{x}_h|\theta, \psi) = \prod_{h=1}^{n} \left[ \sum_{\mathbf{y}_h} f(\mathbf{y}_h|\theta) f(\mathbf{x}_h|\mathbf{y}_h, \psi) \right], \tag{4.10}$$

where $n$ is the number of sites (alignment columns), and $\mathbf{y}_h$ and $\mathbf{x}_h$ are the full and observed data at site $h$, respectively,

We describe a general procedure to deal with both missing data and sequence errors, and then discuss the particulars of each. The Nomenclature Committee of the International Union of Biochemistry (NC-IUB) recognizes 15 incompletely specified bases: T (= U), C, A, G, Y (T or C), R (A or G), M (C or A), K (T or G), S (C or G), W (T or A), H (not G), B (not A), V (not U), D (not C), and N (any base). Suppose that the 'rate' of missing data or sequence errors is sequence-specific, and ambiguities and sequence errors occur at random in the sequence, independently of other species. We define a $4 \times 15$ matrix $E^{(i)} = \left\{\varepsilon_{yx}^{(i)}\right\}$, where $\varepsilon_{yx}^{(i)}$ is the probability that nucleotide or ambiguity symbol $x \in$ {T, C, A, G, Y, R, M, K, S, W, H, B, V, D, N} is observed in the sequence at tip $i$ given that the true nucleotide is $y \in$ {T, C, A, G}. Note that $\sum_x \varepsilon_{yx}^{(i)} = 1$ for every $y$ and every tip $i$. Parameters involved in the $E^{(i)}$ matrices for the tips are collected into the vector $\psi$.

Felsenstein's pruning algorithm allows the summation in equation (4.10) over the true nucleotides ($\mathbf{y}_h$) at site $h$ to be achieved automatically, by setting

$$L_i(y) = \varepsilon_{yx_i}^{(i)}, \tag{4.11}$$

i.e. by setting the conditional probability vector at tip $i$ to $\left\{\varepsilon_{Tx_i}^{(i)}, \varepsilon_{Cx_i}^{(i)}, \varepsilon_{Ax_i}^{(i)}, \varepsilon_{Gx_i}^{(i)}\right\}$. Recall $L_i(y)$ is the probability of observing data ($x_i$) at tip $i$ at the site given that the true nucleotide is $y$. The general model involves too many parameters and does not appear to be identifiable. We can constrain the model and reduce the number of parameters to be estimated from the data.

#### 4.2.6.2 *Ambiguities and missing data*

Now we consider the case of ambiguities and missing data, assuming no sequence errors. In other words, if an observed nucleotide is one of T, C, A, or G, it is assumed to be the true nucleotide. As an example, suppose in the alignment of the three sequences for Figure 4.4, sequences 1 and 3 involve ambiguities at different rates (prevalence). Consider a site with the observed data to be $\mathbf{x}_h$ = YTR. The probability of observing such a site is a sum over all nucleotide configurations ($\mathbf{y}_h$) that are compatible with $\mathbf{x}_h$. From equation (4.10), we have

$$\begin{aligned} f(\text{YTR}|\theta, \psi) = {} & f(\text{TTA}|\theta) \times \varepsilon_{TY}^{(1)} \varepsilon_{AR}^{(3)} + f(\text{TTG}|\theta) \times \varepsilon_{TY}^{(1)} \varepsilon_{GR}^{(3)} \\ & + f(\text{CTA}|\theta) \times \varepsilon_{CY}^{(1)} \varepsilon_{AR}^{(3)} + f(\text{CTG}|\theta) \times \varepsilon_{CY}^{(1)} \varepsilon_{GR}^{(3)}. \end{aligned} \tag{4.12}$$

Note that $f(\mathbf{x}_h|\mathbf{y}_h, \psi) = \varepsilon_{TY}^{(1)} \varepsilon_{AR}^{(3)}$ for $\mathbf{y}_h$ = TTA, or $= \varepsilon_{TY}^{(1)} \varepsilon_{GR}^{(3)}$ for $\mathbf{y}_h$ = TTG, and so on. It appears plausible to assume that $\varepsilon_{TY}^{(1)} = \varepsilon_{CY}^{(1)}$ and $\varepsilon_{AR}^{(3)} = \varepsilon_{GR}^{(3)}$; i.e. the chance for T to be read as Y is the same as that for C to be read as Y, etc. Note that the relevant probability here is the

probability of seeing Y when the true base is T (or C), rather than the probability that the observed Y is in fact T (or C). The former informs us about the sequencing technology and error-generating mechanism, while the latter may mainly reflect the base compositions in the sequence. At any rate, under the assumption that the probability $f(\mathbf{x}_h|\mathbf{y}_h, \psi)$ does not depend on the true state $\mathbf{y}_h$, equation (4.12) can be written as

$$f(\mathbf{x}_h|\theta, \psi) = f(\text{YTR}|\theta) = c[f(\text{TTA}|\theta) + f(\text{TTG}|\theta) + f(\text{CTA}|\theta) + f(\text{CTG}|\theta)], \tag{4.13}$$

where $c = \varepsilon_{TY}^{(1)}\varepsilon_{AR}^{(3)}$ is a constant if we are not interested in parameters $\psi$; i.e. $c$ is independent of the tree topology and parameters $\theta$ such as the branch lengths. We can ignore $c$ and the likelihood for $\theta$ is still correctly defined.

The sum in equation (4.13) can be achieved by setting $L_i(y)$ for tip $i$ to 1 for any nucleotide $y$ that is compatible with the observed state ($x_i$) and to 0 otherwise; in other words, the conditional probability vectors are (1, 1, 0, 0) for tip 1 and (0, 0, 1, 1) for tip 3. For a site with exact data such as $\mathbf{x}_h$ = TCA, we have $\mathbf{y}_h$ = TCA, and $f(\mathbf{x}_h|\theta, \psi) = f(\mathbf{y}_h|\theta) \times \varepsilon_{TT}^{(1)}\varepsilon_{AA}^{(3)}$. Again if we are not interested in parameters $\psi$, we can ignore the constant $\varepsilon_{TT}^{(1)}\varepsilon_{AA}^{(3)}$. This idea of dealing with ambiguities by setting the conditional probabilities at the tips to 0s and 1s is due to Felsenstein (2004, pp. 255–6), and is the version used in ML phylogenetics programs. This has the nice property that the likelihood stays exactly the same if an empty column is added to the alignment which has an 'N' or '?' in every sequence. The discussion above, however, suggests that the strategy works only if the probabilities of seeing the observed ambiguous nucleotide are the same for the different compatible true nucleotides, i.e. if $\varepsilon_{TY} = \varepsilon_{CY}$, $\varepsilon_{TN} = \varepsilon_{CN} = \varepsilon_{AN} = \varepsilon_{GN}$, and so on. If T were harder to read by the sequencing machine and tended to become an ambiguity Y more often than C, this treatment would not be correct. In general, missing data are easier to accommodate in the model if the probability of missing data does not depend on the true state (Little and Rubin 1987, pp. 88–92).

#### 4.2.6.3 *Sequence errors*

The sequence errors we consider here are those in a single sequence, and not those in multiple reads of a single sequence or of a mixture of alleles. If data from multiple genomic regions are available, it may be reasonable to assume that the sequence errors are genome-specific, so that the same error model is applied to sequences from multiple loci of the same genome. Suppose sequence/genome $i$ has errors but no ambiguities, we can define a transition matrix $E^{(i)}$, of size $4 \times 4$ (instead of $4 \times 15$),

$$E = \{\varepsilon_{yx}\} = \begin{bmatrix} \varepsilon_{TT} & \varepsilon_{TC} & \varepsilon_{TA} & \varepsilon_{TG} \\ \varepsilon_{CT} & \varepsilon_{CC} & \varepsilon_{CA} & \varepsilon_{CG} \\ \varepsilon_{AT} & \varepsilon_{AC} & \varepsilon_{AA} & \varepsilon_{AG} \\ \varepsilon_{GT} & \varepsilon_{GC} & \varepsilon_{GA} & \varepsilon_{GG} \end{bmatrix}, \tag{4.14}$$

where $\varepsilon_{yx}$ is the probability of observing base $x$ if the true base is $y$, with the nucleotides ordered T, C, A, and G. Here we suppress the superscript $i$ for genome $i$. Each row sums to 1, so there are 12 free parameters in the general matrix. The error rate may depend on the true nucleotide, and evolution and sequencing errors may have different patterns, such as different transition/transversion rate ratios. To reduce the number of parameters, we can use the HKY85 model to describe sequence errors, with five parameters. The simplest model will be JC69, with the diagonal to be $\varepsilon_{TT} = \varepsilon_{CC} = \varepsilon_{AA} = \varepsilon_{GG} = 1 - 3\varepsilon$ and the off-diagonals to be $\varepsilon$. The conditional probability vector at the tip of the tree is then set to $\{1 - 3\varepsilon, \varepsilon, \varepsilon, \varepsilon\}$ for an observed T, $\{\varepsilon, 1 - 3\varepsilon, \varepsilon, \varepsilon\}$ for C, and so on.

While the error model can be arbitrarily complex in a computer simulation, for inference, we have to consider the information content in the data and avoid non-identifiability. It is prudent to require at least one genome to be free of sequence errors. In comparison of closely related species, the molecular clock may be assumed as well. Sequence errors have the effect of adding an extra amount of evolution to the tip branches of the tree. Thus the apparent violation of the clock due to the sequence errors and the different patterns between evolution and sequence errors may provide information for estimating the error rate parameters (Burgess and Yang 2008).

#### 4.2.6.4 *Alignment gaps*

Alignment gaps pose greater difficulties to likelihood calculation than ambiguities, sequence errors or DNA degradation. The general theory of equation (4.9) still holds, with $X$ being the observed alignment, and $Y$ the unobserved true alignment. However, both $f(Y|\theta)$ and $f(X|Y, \psi)$ are hard to calculate. Here $f(X|Y, \psi)$ is the probability of the 'observed' alignment generated by the alignment program given the true alignment, while $f(Y|\theta)$ is the probability of the true alignment given the phylogeny, branch lengths, insertion and deletion rates, etc.

In theory, it is advantageous to develop models of insertions and deletions as well as substitutions to align sequences in a probabilistic framework (Bishop and Thompson 1986; Thorne et al. 1991, 1992). Such an approach will generate estimates of the insertion and deletion rates, and also provide a probabilistic measurement of alignment accuracy. However, the early methods are based on simplistic assumptions about insertions and deletions and involve intensive computation even for two sequences. While improvements are being made both to the biological realism of the model and to the computational efficiency (see, e.g. Hein et al. 2000, 2003; Lunter et al. 2005), this modelling approach has not reached the stage of producing a useable method or software program. As a result, almost all multiple sequence alignments are generated using heuristic methods. There is also a keen interest in inferring alignment and phylogeny simultaneously, for example, using the Bayesian framework (Fleissner et al. 2005; Holmes 2005; Redelings and Suchard 2005).

Here we discuss a few *ad hoc* procedures for treating gaps in a given alignment, that is, calculation of $f(Y|\theta)$, with alignment errors ignored. There are three options. The first is to treat an alignment gap as the fifth nucleotide or the 21st amino acid, different from all other character states. This is used in some parsimony algorithms, but is uncommon in likelihood implementations (but see McGuire et al. 2001). One problem with this approach is that it treats a stretch of five gaps as five independent evolutionary events, even though it may well represent one event (an insertion or deletion of five nucleotides). Two worse options are commonly used: (i) to delete all sites at which there are alignment gaps in any sequence and (ii) to treat alignment gaps as undetermined nucleotides (ambiguities). The information loss caused by deleting sites with alignment gaps can be substantial if the sequences are divergent and the alignment includes many columns with gaps. The approach of treating alignment gaps as undetermined nucleotides is problematic as well, since gaps mean that the nucleotides do not exist and not that they exist but are unknown. It is not so clear what effects those two approaches have on phylogenetic tree reconstruction. It seems reasonable to delete a site if it contains alignment gaps in most species and to keep the site if it contains alignment gaps in very few species. In analysis of highly divergent species, it is common practice to remove regions of the protein for which the alignment is unreliable.

## 4.3 Likelihood calculation under more complex models

The discussion in the section above assumes that all sites in the sequence evolve at the same rate according to the same rate matrix. This may be a very unrealistic assumption for real sequences. Much progress has been made in the last two decades in extending models used in likelihood analysis. A few important extensions are discussed in this section.

### 4.3.1 *Mixture models for variable rates among sites*

In real sequences, the substitution rates are often variable across sites. Ignoring rate variation among sites can have a major impact on phylogenetic analysis (e.g. Tateno et al. 1994; Huelsenbeck 1995a; Yang 1996c; Sullivan and Swofford 2001). To accommodate variable rates in a likelihood model, one should not in general use a rate parameter for every site, as otherwise there will be too many parameters to estimate and the likelihood method may misbehave. A sensible approach is to use a statistical distribution to model the rate variation. Both discrete and continuous rate distributions have been used.

#### 4.3.1.1 *Discrete-rate model*

In this model, sites are assumed to fall into $K$ discrete classes with different rates (Table 4.1). The rate at any site in the sequence takes a value $r_k$ with probability $p_k$, with $k = 1, 2, \ldots, K$. The $r$s and $p$s are parameters, to be estimated by ML from the data. We place two constraints to avoid the use of too many parameters. First the probabilities sum to one: $\sum p_k = 1$. Second, the average rate is fixed at $\sum p_k r_k = 1$, so that the branch length is measured as the expected number of nucleotide substitutions per site averaged over the site or rate classes. The model with $K$ site classes thus involves $2(K-1)$ free parameters. The rates $r$s are thus relative multiplication factors. Every site is in effect evolving along the same tree topology with proportionally elongated or shrunken branches. In other words, the substitution rate matrix at a site with rate $r$ is $rQ$, with $Q$ shared across all sites.

As we do not know which site class each site belongs to, the probability of observing data at any site is a weighted average over the site classes

$$f(\mathbf{x}_h|\theta) = \sum_{k=1}^{K} p_k \times f(\mathbf{x}_h|r = r_k; \theta). \tag{4.15}$$

The likelihood is again calculated by multiplying the probabilities across sites. The conditional probability, $f(\mathbf{x}_h|r; \theta)$, of observing data $\mathbf{x}_h$ given the rate $r$, is just the probability under the one-rate model, with all branch lengths multiplied by $r$. It can be calculated using the pruning algorithm for each site class. A variable-rate model with $K$ site classes thus takes $K$ times as much computation as the one-rate model.

**Table 4.1** The discrete-rate model

| Site class | 1 | 2 | 3 | ... | $K$ |
|---|---|---|---|---|---|
| Probability | $p_1$ | $p_2$ | $p_3$ | ... | $p_K$ |
| Rate | $r_1$ | $r_2$ | $r_3$ | ... | $r_K$ |

As an example, consider the tree of Figure 4.4b. We have

$$f(x_1x_2x_3|\theta) = \sum_{k=1}^{K} p_k \times f(x_1x_2x_3|r = r_k; \theta),$$
$$f(x_1x_2x_3|r; \theta) = \textstyle\sum_j \pi_j p_{jx_1}(t_1 r) p_{jx_2}(t_1 r) p_{jx_3}((2t_0 - t_1)r) \tag{4.16}$$

(compare with equation (4.8)).

Discrete-rate models are known as *finite-mixture models*, since the sites are a mixture from $K$ classes. The general model of Table 4.1 is implemented by Yang (1995a). As is typical in such finite-mixture models, one can fit only a few rate classes to practical datasets, so $K$ should not exceed 3 or 4. Also the estimates of the rate and proportion parameters from a given dataset tend to change dramatically when $K$ changes, making it hard to interpret those parameters. A special case of the general discrete-rate model is the *invariable-site* model, which assumes two site classes: the class of *invariable sites* with rate $r_0 = 0$ and another class with a constant rate $r_1$(Hasegawa et al. 1985). As the average rate is $p_0 r_0 + (1 - p_0) r_1 = (1 - p_0) r_1 = 1$, we have $r_1 = 1/(1 - p_0)$, where the proportion of invariable sites $p_0$ is the only parameter in the model. Note that a variable site in the alignment cannot have rate $r_0 = 0$. Thus the probability of data at a site, i.e. equation (4.15), becomes

$$f(\mathbf{x}_h|\theta) = \begin{cases} p_0 + p_1 \times f(\mathbf{x}_h|r = r_1; \theta), & \text{if the site is constant,} \\ p_1 \times f(\mathbf{x}_h|r = r_1; \theta), & \text{if the site is variable.} \end{cases} \tag{4.17}$$

*Example 4.2 Discrete-rate model for mitochondrial 12S rRNA*. We fit the general discrete-rate model (Table 4.1) to a dataset of mitochondrial small subunit (12S) rRNA genes from 30 Old World and New World Monkeys. There are 978 sites in the alignment. The tree topology is shown in Figure 4.7. We assume the HKY85 model and examine the effect of the number of rate classes. The log likelihood ($\ell$) is plotted against $K$ in Figure 4.8, while parameter estimates are listed in Table 4.2. Each additional rate class adds two free parameters to the model. The log likelihood improves hugely (by 685.35 units) by the addition of the second rate class (i.e. when $K$ increases from 1 to 2), by 49.35 for the third class, and only by 0.58 for the fourth class. For this dataset, it is not possible to fit more than four classes; models with $K \geq 5$ simply collapse to the model with $K = 4$. Note that use of the constant-rate model leads to underestimation of branch lengths, tree length, and the transition/transversion rate ratio (Wakeley 1994; Yang et al. 1994, 1995c). □

#### 4.3.1.2 *Gamma-rate model*

A second approach is to use a continuous distribution to approximate variable rates among sites. The most commonly used distribution is the gamma (see Figure 1.6 in Chapter 1). In §1.3 we discussed the use of the same gamma model in calculating pairwise distances. The gamma density is

$$g(r; \alpha, \beta) = \frac{\beta^\alpha r^{\alpha-1} e^{-\beta r}}{\Gamma(\alpha)}, \tag{4.18}$$

with mean $\alpha/\beta$ and variance $\alpha/\beta^2$. We let $\beta = \alpha$ so that the mean is 1. The shape parameter $\alpha$ is inversely related to the extent of rate variation among sites. As in the discrete-rate model, we do not know the rate at the site, and have to average over the rate distribution

$$f(\mathbf{x}_h|\theta) = \int_0^\infty g(r) f(\mathbf{x}_h|r; \theta)\, dr. \tag{4.19}$$

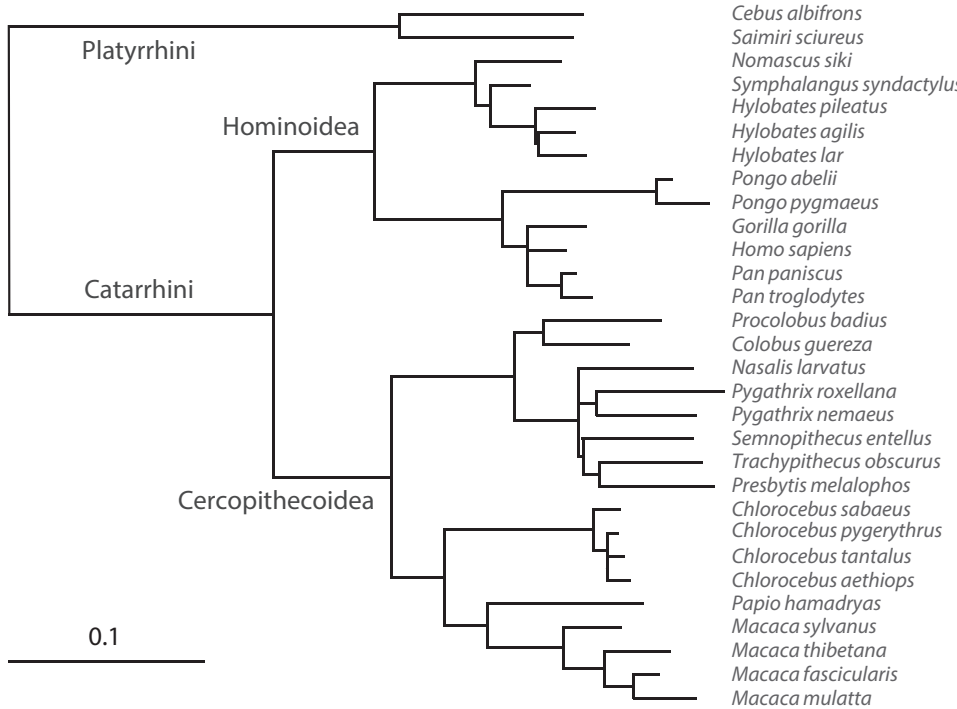


**Fig. 4.7** The tree for the mitochondrial 12S rRNA genes from 30 primate species (catarrhini and platyrrhini), with branch lengths estimated under the HKY + $\Gamma_5$ model. The unrooted tree is used to fit different models of rate variation among sites in Table 4.2 and Figure 4.8, but here the tree is rooted for clarity.

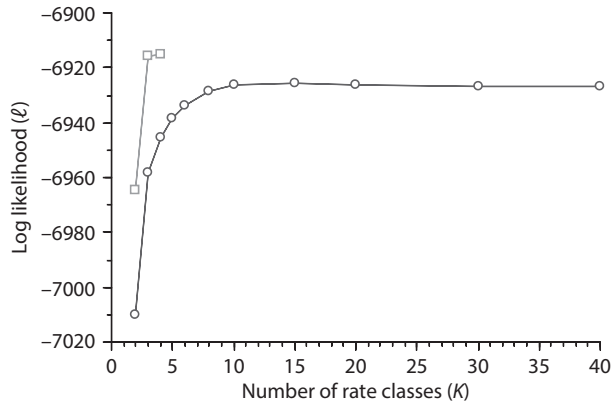


**Fig. 4.8** The log likelihood value ($\ell$) as a function of the number of rate classes ($K$) in the discrete-rate model (□) and in the discrete gamma model (◯). In the discrete-rate model $\ell$ stays the same when $K \geq 4$, and in both models, $\ell = -7650.15$ when $K = 1$. The mitochondrial 12S rRNA genes from 30 primates are analysed (see Figure 4.7 and Table 4.2).

**Table 4.2** Log likelihood values and MLEs of parameters under variable-rate models with different rate classes

| Model | $p$ | $\ell$ | $\hat{T}$ | $\hat{\kappa}$ | $\hat{r}_k$ and $\hat{p}_k$ (or $\hat{a}$) |
|---|---|---|---|---|---|
| Constant-rate model | | | | | |
| $K = 1$ | 61 | –7650.15 | 1.445 | 9.88 | |
| Discrete-rate model | | | | | |
| $K = 2$ | 63 | –6964.80 | 1.920 | 12.97 | $\hat{r}_k$ : 0.096 2.954 |
| | | | | | $\hat{p}_k$ : 0.684 0.316 |
| $K = 3$ | 65 | –6915.45 | 2.159 | 14.30 | $\hat{r}_k$ : 0.036 1.466 5.860 |
| | | | | | $\hat{p}_k$ : 0.608 0.301 0.092 |
| $K = 4$ | 67 | –6914.87 | 2.168 | 14.35 | $\hat{r}_k$ : 0.000 0.341 1.667 6.062 |
| | | | | | $\hat{p}_k$ : 0.486 0.171 0.259 0.084 |
| $K = 5$ | 69 | as for $K = 4$ | | | |
| Discrete gamma model | | | | | $\hat{a}$ |
| $K = 2$ | 62 | –7010.24 | 2.418 | 12.55 | 0.257 |
| $K = 3$ | 62 | –6958.11 | 2.137 | 13.11 | 0.247 |
| $K = 4$ | 62 | –6945.29 | 2.080 | 13.50 | 0.254 |
| ***K* = 5** | **62** | **–6938.65** | **2.143** | **13.87** | **0.259** |
| $K = 6$ | 62 | –6933.72 | 2.220 | 14.21 | 0.256 |
| $K = 8$ | 62 | –6928.26 | 2.250 | 14.53 | 0.249 |
| $K = 10$ | 62 | –6926.36 | 2.221 | 14.63 | 0.247 |
| $K = 15$ | 62 | –6925.79 | 2.187 | 14.73 | 0.247 |
| $K = 20$ | 62 | –6926.13 | 2.194 | 14.79 | 0.247 |
| $K = 30$ | 62 | –6926.56 | 2.226 | 14.85 | 0.246 |
| $K = 40$ | 62 | –6926.68 | 2.251 | 14.88 | 0.245 |
| $K = 50$ | 62 | –6926.70 | 2.266 | 14.89 | 0.244 |
| $K = 100$ | 62 | –6926.59 | 2.288 | 14.89 | 0.242 |

*Note*: The HKY85 model is assumed, with the observed base frequencies used as estimates. $p$ is the number of parameters in the model and $T$ is the tree length (sum of 57 branch lengths). Rates among sites are assumed to be constant, or modelled using the general discrete-rate model or the discrete gamma model, with $K$ rate classes. The data are mitochondrial 12S rRNA gene sequences from 30 primates. The phylogeny is shown in Figure 4.7, which uses the branch length estimates under HKY85 + $\Gamma_5$ (highlighted here in bold).

Here the collection of parameters $\theta$ includes $\alpha$ as well as branch lengths and other parameters in the substitution model (such as $\kappa$). Note that the sum in equation (4.15) for the discrete model is now replaced by an integral for the continuous model.

An algorithm for calculating the likelihood function of equation (4.19) was described by Yang (1993), Gu et al. (1995), and Kelly and Rice (1996). Note that the conditional probability $f(\mathbf{x}_h|r)$ is a sum over all combinations of ancestral states (equation (4.3)). Each term in the sum is a product of transition probabilities across the branches. Each transition probability has the form $p_{ij}(tr) = \sum_k c_{ijk} e^{\lambda_k tr}$, where $t$ is the branch length, $\lambda_k$ is the eigenvalue, and $c_{ijk}$ is a function of the eigenvectors (see equation (1.43)). Thus, after expanding all the products, $f(\mathbf{x}_h|r)$ is a sum of many terms of the form $ae^{br}$, and then the integral over $r$ can be obtained analytically (equation (1.36)). However, this algorithm is very slow because of the huge number of terms in the sum, and is only practical for small trees with fewer than ten sequences.

#### 4.3.1.3 *Discrete gamma model*

One may use the discrete-rate model as an approximation to the continuous gamma, leading to the *discrete gamma model*. Yang (1994a; see also Waddell et al. 1997) tested this strategy, using $K$ equal probability site classes, with the mean or median for each class used to represent all rates in that class (Figure 4.9). Thus $p_k = 1/K$ while $r_k$ is calculated as a function of the gamma shape parameter $\alpha$. If the mean rate is used, we have

$$r_k = K \int_a^b r \times g(r; \alpha, \beta)\, \mathrm{d}r, \tag{4.20}$$

where $a$ and $b$ are the boundary points for the $k$th bin. The probability of data at a site is then given by equation (4.15). The model involves one single parameter $\alpha$, just like the continuous gamma. The discrete gamma may be considered a crude way of calculating the integral of equation (4.19). Yang's (1994a) test on small datasets suggested that as few as $K = 4$ site classes provided good approximation. In large datasets with hundreds of sequences, more categories may be beneficial (Mayrose et al. 2005). Because of the discretization, the discrete gamma means less rate variation among sites than the continuous gamma for the same parameter $\alpha$. Thus, the discrete gamma almost always produces a smaller estimate of $\alpha$ than the continuous gamma when the two models are fitted to the same dataset (which has a fixed amount of rate variation).

Felsenstein (2001a) and Mayrose et al. (2005) discussed the use of numerical integration (quadrature) algorithms to calculate the integral of equation (4.19). We will discuss methods of numerical integration later, in §6.4. Note that the integral is the area under the integrand curve. The simplest of algorithms for calculating the integral cut the $x$-axis

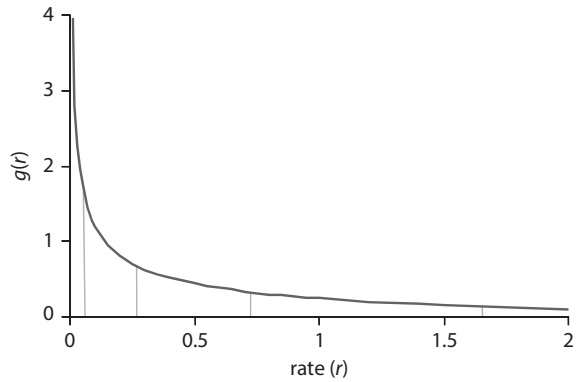


**Fig. 4.9** The discrete gamma model of variable rates across sites uses $K$ equal-probability categories to approximate the continuous gamma distribution, with the mean rate in each category used to represent all rates in that category. Shown here is the use of $K = 5$ categories to approximate the gamma density $g(r)$ with $\alpha = 0.5$ (and mean 1). The four vertical lines are at $r = 0.06418,\ 0.27500,\ 0.70833,$ and 1.64237. These are the 20%, 40%, 60%, and 80% percentiles of the distribution and cut the density into five categories, each of proportion 1/5. The mean rates in the five categories are 0.02121, 0.15549, 0.46708, 1.10712, and 3.24910.

into equal-sized segments and then approximate the integral by the sum of the areas of the rectangles on the segments. More sophisticated algorithms calculate the values of the integrand for a fixed set of values of $r$ and then approximate the integrand by using simpler functions such as the polynomial. A concern is that the integrand in equation (4.19) may peak at zero for constant sites and at large values for highly variable sites, so that an algorithm using fixed points may not approximate the integral well for all sites. Adaptive algorithms are more reliable as they sample more densely in regions where the integrand is large. However, they are too expensive, as different points are used for different sites, meaning that the transition probability matrix has to be calculated for every site along every branch for every rate class. Given that both the continuous and discrete gamma models are empirical without mechanistic biological justifications, there should be no strong preference for either of them over the other. Thus there is not a great need for very accurate calculation of the integral in the continuous gamma model as the discrete model may work equally well.

*Example 4.3 Discrete gamma model for mitochondrial 12S rRNA*. The discrete gamma model is fitted to the primate mitochondrial 12S rRNA genes analysed in Example 4.2. The HKY85 model is assumed in combination with the discrete gamma model of variable rates for sites, with different rate categories ($K$) used (Table 4.2). Note that $K$ is not a parameter, and the models with different $K \geq 2$ are non-nested models with the same number of parameters. One does not optimize $K$ but instead uses a reasonable fixed value (say, 5 or 4) and then the estimates of parameter $\alpha$ will be comparable across datasets. In terms of model adequacy (the log likelihood), the discrete-rate model with $K = 3$ provides a good fit to this dataset, better than the discrete gamma model. However, the former uses four parameters in the rate distribution ($r_k$ and $f_k$) while the latter uses only one ($\alpha$). Furthermore, the estimates of $r_k$ and $f_k$ in the discrete-rate model are unstable. Overall, given the high similarity of estimates of parameters such as $\kappa$ and branch lengths across those variable-rate models, the discrete gamma appears to be a better model than the general discrete-rate model if our objective is to reconstruct the phylogeny or estimate branch lengths, etc. □

*Example 4.4 The phylogeny of seven ape species*. We use the MTMAM + $\Gamma_5$ model to analyse the dataset of mitochondrial protein sequences (see §4.2.4), with a discrete gamma model used to accommodate variable rates among sites, with $K = 5$ site classes used. The shape parameter is fixed at $\alpha = 0.4$, which is the estimate from mammalian species (Yang et al. 1998). The ML tree is the same as under the MTMAM model (Figure 4.5). The log likelihood value for the tree is $\ell = -14,413.38$, much higher than that under the one-rate model ($\ell_0 = -14,558.59$). If $\alpha$ is estimated from the data, the same tree is found to be the ML tree, with $\hat{\alpha} = 0.333$ and $\ell = -14,411.90$. According to the likelihood ratio test (LRT), we should compare $2\Delta\ell = 2(\ell_1 - \ell_0) = 293.38$ with $\chi^2_1$ (the $\chi^2$ distribution with one degree of freedom) or with the 1:1 mixture of 0 and $\chi^2_1$ (Whelan and Goldman 2000). There is no doubt that the discrete gamma model fits the data much better than the one-rate model. □

#### 4.3.1.4 *Other rate distributions*

In addition to the gamma model, several other statistical distributions have been implemented and tested. Waddell et al. (1997) considered the log-normal model. Kelly and Rice (1996) took a nonparametric approach, assuming a general continuous distribution without specifying any particular distributional form. With such a general model, not much inference is possible.

Gu et al. (1995) added a proportion of invariable sites to the gamma distribution, so that a proportion $p_0$ of sites are invariable while the other sites (with proportion $p_1 = 1 - p_0$)

have rates drawn from the gamma. The model is known as 'I + Γ'. The mean is fixed at 1: $E(r) = (1 - p_0)\,\alpha/\beta = 1$ so that $\beta = (1 - p_0)\,\alpha$. The variance is $V(r) = \dfrac{1 + \alpha p_0}{\alpha(1 - p_0)}$. This model is somewhat pathological as the gamma distribution with $\alpha \leq 1$ already allows for sites with very low rates; as a result, adding a proportion of invariable sites creates a strong correlation between $p_0$ and $\alpha$, making it hard to estimate those parameters reliably (Yang 1993; Sullivan et al. 1999; Mayrose et al. 2005). Another drawback of the model is that the estimate of $p_0$ is very sensitive to the number and divergences of the sequences included in the data. The proportion $p_0$ is never larger than the observed proportion of constant sites; with the addition of more and divergent sequences, the proportion of constant sites drops, and the estimate of $p_0$ tends to go down as well. The I + Γ model is typically the most complex among models implemented in phylogenetic programs and is often the recommended model by automatic model selection procedures such as MODELTEST (Posada and Crandall 1998; Posada 2008), due to the large size of phylogenetic datasets. With the drawbacks mentioned here, one should avoid the I + Γ models and use the gamma models instead.

Mayrose et al. (2005) suggested a gamma mixture model, which assumes that the rate for any site is a random variable from a mixture of two gamma distributions with different parameters. This appears more stable than the I + Γ model. For estimating branch lengths and phylogenies, the different distributions tend to produce similar results, so that the simple gamma should be adequate. For estimating rates at sites from large datasets with many sequences, the mixture of two gammas may be preferable.

Note that under both continuous- and discrete-rate models, data at different sites are independent and identically distributed (i.i.d.). While the model allows different sites to evolve at different rates, it does not specify *a priori* which sites should have which rates. Instead the rate for any site is a random draw from a common distribution. As a result, data at different sites have the same distribution.

#### 4.3.1.5 *Empirical Bayesian estimation of substitution rates at sites*

Large datasets with many sequences may provide opportunities for multiple changes at individual sites, making it possible to estimate the relative substitution rate at each site. In both discrete- and continuous-rate models, the rates for sites are random variables and are integrated out in the likelihood function. Thus to estimate the rate, we use the conditional (posterior) distribution of the rate given the data at the site

$$f(r|\mathbf{x}_h; \theta) = \frac{f(r|\theta) f(\mathbf{x}_h|r; \theta)}{f(\mathbf{x}_h|\theta)}. \tag{4.21}$$

Parameters $\theta$ may be replaced by their estimates, such as the MLEs. This is known as the empirical Bayes (EB) approach. Under the continuous-rate model, one can use the posterior mean as the estimate of the rate at the site (Yang and Wang 1995). Under the discrete-rate model, the rate $r$ in equation (4.21) takes one of $K$ possible values, with $f(r|\theta) = p_k$. One may calculate the posterior mean or use the rate with the highest posterior probability as the best estimate (Yang 1994a).

The EB approach ignores sampling errors in the parameter estimates, which may be a source of concern in small datasets. One can assign a prior on the parameters and use a full Bayesian approach to deal with uncertainties in the parameters. Mateiu and Rannala (2006) developed an interesting Markov chain Monte Carlo (MCMC) algorithm for estimating rates at sites, using a uniformized Markov chain (Kao, 1997 pp. 273–277) to implement the continuous gamma model. This strategy allows large datasets

with hundreds of sequences to be analysed under the continuous model. The authors' simulation shows that the discrete gamma model provides good estimates of branch lengths, but tends to underestimate high rates unless a very large number of site classes is used (with $K \geq 40$, say).

Another likelihood approach to estimating rates at sites was implemented by Nielsen (1997), who treated the rate at every site as a parameter, estimated by ML. This method suffers from the use of too many parameters, so that the estimates are often zero or infinity.

#### 4.3.1.6 *Correlated rates at adjacent sites*

Yang (1995a) and Felsenstein and Churchill (1996) implemented models that allow rates to be correlated across adjacent sites. In the *auto-discrete gamma* model (Yang 1995a), the rates at two adjacent sites have a bivariate gamma distribution (i.e. the marginal distributions of both rates are gamma), discretized to make the computation feasible. This is an extension to the discrete gamma model and includes a parameter $\rho$, which measures the strength of autocorrelation. The model is implemented through a hidden Markov chain, which describes the transition from one rate class to another along the sequence. Felsenstein and Churchill (1996) described a similar hidden Markov chain model, in which a segment of nucleotide sites is assumed to have the same rate, with the length of the segment reflecting the strength of the correlation. The hidden Markov model involves about the same amount of computation as the independent discrete-rate model. Tests on real data suggest that substitution rates are indeed highly correlated. Nevertheless, estimation of branch lengths or other parameters in the model did not seem to be affected by ignoring the correlation of rates at adjacent sites, although the variances of the MLEs were underestimated when the correlation is ignored.

#### 4.3.1.7 *Covarion models*

In both the discrete- and continuous-rate models, the rate for a site is applied to all branches in the tree, so that a fast-evolving site is fast-evolving throughout the phylogeny. This assumption is relaxed in the *covarion* (for COncomitantly VARIable codON) models, which are based on Fitch's (1971a) idea of coevolving codons in a protein-coding gene, with substitutions in one codon affecting substitutions in other codons. Such models allow a site to switch from one site class to another. As a result, a site may be fast-evolving along some lineages while slowly evolving along others. Tuffley and Steel (1998) discussed an extension to the invariable-site model, in which a site switches between two states: the 'on' state (+) , in which the nucleotide has a constant rate of evolution and the 'off' state (–), in which the nucleotide is invariable. A likelihood implementation of this model is provided by Huelsenbeck (2002). Similarly Galtier (2001) implemented an extension to the discrete gamma model, in which a site switches from one site class to another over evolutionary time, with $K$ site classes from the discrete gamma.

Here we use the simpler model of Tuffley and Steel (1998) and Huelsenbeck (2002) to describe the implementation of such models. A simple approach is to construct a Markov chain with an expanded state space. Instead of the four nucleotides, we consider eight states: $T_+$, $C_+$, $A_+$, $G_+$, $T_-$, $C_-$, $A_-$, and $G_-$, with '+' and '–' representing the 'on' and 'off' states, respectively. Nucleotides of the '+' state can change between themselves, according to a substitution model such as JC69 or HKY85. A nucleotide of the '–' state can change to the same nucleotide of the '+' state only.

In the likelihood calculation, one has to sum over all the eight states for each ancestral node to calculate the probability of data at each site. Furthermore, a nucleotide observed at a tip node can be in either '+' or '–' states, so that the probability of data at a site

will be a sum over all compatible patterns of the expanded states. Suppose the observed data at a site for three species are TCA. Then Pr(TCA) will be a sum over eight site patterns: $T_+C_+A_+$, $T_+C_+A_-$, $T_+C_-A_+$, . . . , and $T_-C_-A_-$. This summation can be achieved efficiently using likelihood treatment of missing data in the pruning algorithm, as discussed in §4.2.6; we let the conditional probability $L_i(x_i)$ for the tip node $i$ be 1 for both states $x_i = T_+$ and $x_i = T_-$ if the observed nucleotide for tip $i$ is T, say. The model of Galtier (2001) can be implemented in the same way, although the Markov chain has $4K$ instead of 8 states.

An interesting use of the same idea is made by Guindon et al. (2004), who implemented a codon-based switching model, which allows a codon to switch between site classes with different $\omega$ ratios. The model thus allows the selective pressure on the protein indicated by the $\omega$ ratio to vary both among sites and among lineages, and is perhaps close to Fitch's (1971a) original idea of coevolving codons. Chapter 11 provides further discussions of codon-based models. It may be noted that all the covarion models discussed above assume that data at different sites are i.i.d.

### 4.3.2 *Mixture models for pattern heterogeneity among sites*

Rate variation among sites is a common feature of genetic sequence evolution. While in nonfunctional and noncoding DNA that is fast evolving without any selective constraint, the mutation rate may be nearly constant over sites, most datasets used in phylogenetic analysis show considerable mutation/substitution rate variation. The dramatic improvement in the fit of the model to data, indicated by the huge increase in the log likelihood, upon adding the single gamma shape parameter seen in Examples 4.3 and 4.4 is typical of many datasets. Furthermore, incorrectly assuming a constant rate for sites when rates vary can have a significant impact on different aspects of phylogenetic analysis such as tree topology reconstruction and branch length estimation (Yang 1996c).

While perhaps not as important as the rate, other aspects of the evolutionary process may vary among sites as well. In theory, the same approaches as discussed above can be taken to accommodate any such among-site heterogeneity in the model. For example, Huelsenbeck and Nielsen (1999) used a gamma distribution to describe variable transition/transversion rate ratios among sites, and another gamma distribution to describe variable rates, both discretized. For protein sequences, the different domains of the protein may have strong preferences for different amino acids (Bruno 1996; Halpern and Bruno 1998). One could use a set of amino acid frequency parameters for each site, but the model would involve too many parameters to implement in an ML method. Treating the amino acid frequencies as random variables (like the gamma-distributed rates for sites) would mean high-dimensional integrals in the likelihood function.

Besides amino acid frequencies, the whole rate matrix $Q$ representing the amino acid substitution pattern may be allowed to vary among sites (Koshi and Goldstein 1996b; Thorne et al. 1996; Goldman et al. 1998; Koshi et al. 1999). Given the plethora of empirical amino acid substitution matrices, a computationally feasible strategy is to have a mixture model of several site classes, with different empirical matrices used for different classes. As the empirical matrices do not involve new parameters, the computational cost is tolerable. Such phylogenetic mixture models were implemented by Le et al. (2008) in the PhyML program (Guindon and Gascuel 2003), combined with gamma-distributed rates among sites. The authors' systematic test using large alignments in a database found that the mixture models of substitution pattern heterogeneity provide improved fit to data.

In general, it is easier to implement parameter-rich mixture models in the Bayesian framework (e.g. Lartillot and Philippe 2004) than in ML, so we will discuss those models in Chapter 8.

### 4.3.3 *Partition models for combined analysis of multiple datasets*

Two approaches may be taken to accommodate the heterogeneity of the substitution rate and pattern among sites. If we know *a priori* which sites are likely to be evolving fast and which sites evolving slowly, it is natural to use *partition* models, which assign different parameters to sites in different partitions. If we lack such information, we can assume a random statistical distribution, i.e. a *mixture model* to accommodate the among-site heterogeneity, as in last subsection. It is also possible to combine the two, with the partition model accommodating the large-scale variations among partitions and mixture model accommodating the remaining variation within each partition. In statistical jargon, partition models are *fixed-effect* models and mixture models are *random-effect* models. Models that include both fixed and random effects are called *mixed-effects models*.

Such partition and partition-mixture models have been implemented by Yang, Lauder, and Lin (1995b; see also Yang 1995a, 1996b; Pupko et al. 2002b) and applied to analysis of hominoid mitochondrial DNA sequences. Different rates, transition/transversion rate ratios and base frequencies are assigned to sites at the three codon positions while the gamma model is used to accommodate the remaining rate variation within each codon position. LRTs are used to compare models of different complexity. The codon position-based partition models are also used by Shapiro et al. (2006; see also Ren et al. 2005) to analyse hundreds of alignments of viral and yeast genes. While the nucleotide-based partition models may not fit the protein-coding gene sequences as well as the codon models, they involve much less computation and appear to be effective in phylogenetic analysis of protein-coding genes.

Note that the site partitioning should be done *a priori* (i.e. not based on analysis of the same data) but can be based on any criteria. The rationale is that sites within the same partition have similar evolutionary characteristics, describable using the same parameters, while sites from different partitions have different evolutionary dynamics, requiring different parameters to accommodate the heterogeneity. The different partitions may correspond to different genes, which evolve at different rates, have different base frequencies, and so on. They may also correspond to different codon positions in a protein-coding gene. Also besides the rate, parameters reflecting other features of the evolutionary process, such as the transition/transversion rate ratio or base compositions, and even the tree topology, can be allowed to differ among partitions.

Consider the rate as an example. Suppose there are $K$ partitions (e.g. genes or codon positions), with rates $r_1, r_2, \ldots, r_K$. To avoid the use of too many parameters, we may fix $r_1 = 1$, so that the branch length is measured by the expected number of substitutions for the first partition. Equivalently, we may fix the average rate to 1, so that the branch length is measured by the number of substitutions per site, averaged over all partitions. Let $I(h)$ label the partition that site $h$ belongs to; i.e. $I(h) = 3$ if site $h$ is from the third codon position. The log likelihood is then

$$\ell(\theta, r_1, r_2, \ldots, r_K; X) = \sum\nolimits_h \log\{f(\mathbf{x}_h|r_{I(h)}; \theta)\}. \tag{4.22}$$

Here the rates for site partitions are parameters in the model, while $\theta$ includes branch lengths and other substitution parameters. The probability of data at a site is calculated by using the correct rate parameter $r_{I(h)}$ for the site, in contrast to the random-rate (mixture)

model, in which the probability for a site is an average over the site classes (equations (4.15) and (4.19)). Likelihood calculation under the fixed-rates (partition) model therefore takes about the same amount of computation as under the one-rate model. The partition model dealing with among-site rate heterogeneity is sometimes known as the *site-specific rate model*, even though the rates are partition-specific but not site-specific.

The most parameter-rich form of the partition model assumes that all parameters in the substitution model are different among partitions. ML under this model is equivalent to separate analysis of data of different partitions, summing up the log likelihood values. This was used by Yang (1996b) when the tree topology is fixed, and by Hasegawa et al. (1997), who evaluated different tree topologies, referring to the method as the *total evidence* approach.

In the partition model, one knows which partition or site class each site is from, and the rates (and other parameters) for site partitions are parameters, estimated by ML. In the mixture models, one does not know which rate each site has; instead one treats the rate as a random draw from a statistical distribution and estimates parameters of that distribution (such as $\alpha$ for the gamma model) as a measure of the variability in the rates. In the mixture model, data at different sites are i.i.d. In the partition model, data at different sites within the same partition are i.i.d., but sites from different partitions have different distributions. This distinction should be taken into account in statistical tests based on resampling sites, such as the bootstrap (Felsenstein 1985a).

The partition models are useful for analysing multiple heterogeneous loci, to assemble information concerning common features of the evolutionary process among genes while accommodating their heterogeneity. It allows the estimation of gene-specific rates and parameters, and hypothesis testing concerning similarities and differences among genes. Both likelihood (Yang 1996b; Pupko et al. 2002b; Leigh et al. 2008) and Bayesian (Suchard et al. 2003; Nylander et al. 2004; Pagel and Meade 2004) approaches can be taken. A similar use of this strategy is in the estimation of species divergence times under local-clock models, in which the divergence times are shared across loci while the multiple loci may have different evolutionary characteristics (Kishino et al. 2001; Yang and Yoder 2003).

In the literature there has been a debate concerning *combined analysis* versus *separate analysis* (see, e.g. Huelsenbeck et al. 1996). In the former, sequences from multiple loci are concatenated and then treated as one 'super-gene', with possible differences in the evolutionary dynamics among the genes ignored. The latter analyses different genes separately. The separate analysis can reveal differences among genes but does not provide a natural way of assembling information from multiple heterogeneous datasets. It may also over-fit the data. Thus neither approach is ideal. The appropriate approach should be a combined analysis that accommodates the heterogeneity among partitions. Another related debate is between *supermatrix* and *supertree* approaches, especially in the context of analysing genomic datasets in which some genes may be missing in some species (see, e.g. Bininda-Emonds 2004). The supermatrix approach concatenates sequences, patching up missing sequences with question marks, and is equivalent to the combined analysis mentioned above. It fails to accommodate possible differences among loci. The supertree approach reconstructs phylogenies using data from different genes, and then assembles the subtrees into a supertree for all species. It is typically difficult to accommodate the uncertainties in the individual subtrees so that heuristic approaches are used to deal with conflicts among the subtrees (Wilkinson et al. 2005). The supertree approaches may be useful in assembling information concerning phylogenies estimated from different sources (such as molecules and morphology). For analysis

of sequence data, the likelihood-based approach to combined analysis, which accommodates the heterogeneity among multiple datasets, should have an advantage (Ren et al. 2009).

### 4.3.4 *Nonhomogeneous and nonstationary models*

In the analysis of divergent sequences, one often observes considerable variation in nucleotide or amino acid compositions among sequences. The assumption of a homogeneous and stationary Markov chain model is clearly violated. One can test whether base compositions are homogeneous by using a contingency table of nucleotide or amino acid counts in the sequences to construct a $X^2$ statistic (e.g. Tavaré 1986; Ababneh et al. 2006). However, such a formal test is hardly necessary because typical molecular datasets are large and such a test can reject the null hypothesis with ease. Empirical studies (e.g. Lockhart et al. 1994) suggest that unequal base compositions can mislead tree reconstruction methods, causing them to group sequences according to the base compositions rather than genetic relatedness.

Dealing with the drift of base compositions over time in a likelihood model is difficult. Yang and Roberts (1995) implemented a few nonhomogeneous models, in which every branch in the tree is assigned a separate set of base frequency parameters ($\pi_T, \pi_C, \pi_A, \pi_G$) in the HKY85 model. Thus the sequences drift to different base compositions during the evolutionary process, after they diverged from the root sequence. This model involves many parameters and is useable for small trees with a few species only. A modification to the model allows the user to specify how many sets of frequency parameters should be assumed and which set each branch should be assigned to. Previously, Barry and Hartigan (1987b) described a model with even more parameters, in which a general transition probability matrix $P$ with 12 parameters is estimated for every branch. The model does not appear to have ever been used in data analysis. Galtier and Gouy (1998) implemented a simpler version of the Yang and Roberts model. Instead of the HKY85 model, they used the model of Tamura (1992), which assumes that G and C have the same frequency and A and T have the same frequency, so that only one frequency parameter for the GC content is needed in the substitution rate matrix. Galtier and Gouy (1999; Boussau and Gouy 2006) estimated different GC content parameters for branches on the tree. Because of the reduced number of parameters, this model was successfully used to analyse relatively large datasets. However, in some datasets, the base compositional drift may not be described properly by just GC content change.

The problem of too many parameters may be avoided by constructing a prior on them, using a stochastic process to describe the drift of base compositions over branches or over time. The likelihood calculation then has to integrate over the trajectories of base frequencies. This integral is daunting, but may be calculated using Bayesian MCMC algorithms (Foster 2004; Blanquart and Lartillot 2006). We discuss such models in Chapter 8.

## 4.4 Reconstruction of ancestral states

### 4.4.1 *Overview*

Evolutionary biologists have had a long tradition of reconstructing traits in extinct ancestral species and using them to test interesting hypotheses. The MacClade program (Maddison and Maddison 2000) provides a convenient tool for ancestral reconstruction using different variants of the parsimony method. Maddison and Maddison (2000) also

provided an excellent review of the many uses (and misuses) of ancestral reconstruction. The *comparative method* (e.g. Felsenstein 1985b; Harvey and Pagel 1991; Schluter 2000) uses reconstructed ancestral states to uncover associated changes between two characters. Although association does not necessarily mean a cause–effect relationship, establishing an evolutionary association is the first step in inferring the adaptive significance of a trait. For example, butterfly larvae may be palatable ($P_+$) or unpalatable ($P_-$) and they may be solitary ($S_+$) or gregarious ($S_-$) . If we can establish a significant association between character states $P_+$ and $S_+$, and if in particular, $P_+$ always appears before $S_+$ on the phylogeny based on ancestral state reconstructions, a plausible explanation is that palatability drives the evolution of solitary behaviour (Harvey and Pagel 1991). In such analysis, ancestral reconstruction is often the first step.

For molecular data, ancestral reconstruction has been used to estimate the relative rates of substitution between nucleotides or amino acids (e.g. Dayhoff et al. 1978; Gojobori et al. 1982), to count synonymous and nonsynonymous substitutions on the tree to infer adaptive protein evolution (e.g. Messier and Stewart 1997; Suzuki and Gojobori 1999), to infer changes in nucleotide or amino acid compositions (Duret et al. 2002), to detect coevolving nucleotides or amino acids (e.g. Shindyalov et al. 1994; Tuff and Darlu 2000; Dutheil et al. 2005), and to conduct many other analyses. Many of these procedures have been superseded by more rigorous likelihood analyses. A major application of ancestral sequence reconstruction is in the so-called chemical paleogenetic restoration studies envisaged by Pauling and Zuckerkandl (1963; see also Zuckerkandl 1964). Those studies use parsimony or likelihood to infer ancestral proteins and then synthesize them using site-directed mutagenesis and examine their chemical and physiological properties in the laboratory (e.g. Malcolm et al. 1990; Stackhouse et al. 1990; Libertini and Di Donato 1994; Jermann et al. 1995; Thornton et al. 2003; Ugalde et al. 2004; Gaucher et al. 2008). Hypotheses concerning the sequences, functions, and structures of ancient proteins are formulated and tested in this way. A number of reviews have been published on the topic (e.g. Golding and Dean 1998; Chang and Donoghue 2000; Benner 2002; Thornton 2004; Dean and Thornton 2007), as well as an edited book (Liberles 2009).

We discussed a dynamic programming algorithm for ancestral reconstruction under weighted parsimony in §3.4.3, which includes as special cases the parsimony method of Fitch (1971b) and Hartigan (1973). While discussing ancestral reconstruction by parsimony, Fitch (1971b) and Maddison and Maddison (1982) emphasized the advantage of a probabilistic approach to ancestral reconstruction and the importance of quantifying the uncertainty in the reconstruction. Nevertheless, early studies (and some recent ones) failed to calculate the right probabilities. When a Markov chain model is used to describe the evolution of characters, the ancestral states are random variables in the model: they do not appear in the likelihood function, which averages over all possible ancestral states (see §4.2), and cannot be estimated from the likelihood function. To infer ancestral states, one should calculate the conditional (posterior) probabilities of ancestral states given the data. This is the EB approach, proposed by Yang et al. (1995a) and Koshi and Goldstein (1996a). It is empirical, as parameter estimates (such as MLEs) are used in the calculation of posterior probabilities of ancestors. Many statisticians consider EB to be a likelihood approach (as opposed to Bayesian or full Bayesian approach). To avoid confusion, I will refer to the approach as EB (instead of likelihood).

Compared with parsimony reconstruction, the EB approach takes into account different branch lengths and different substitution rates between nucleotides or amino acids. It also provides posterior probabilities as a measure of the accuracy of the reconstruction. The EB approach has the drawback of not accommodating sampling errors in parameter

estimates, and may be problematic in small datasets, which lack information to estimate parameters reliably. Huelsenbeck and Ballback (2001) implemented a full (hierarchical) Bayesian approach to ancestral state reconstruction, which assigns priors on parameters and averages over their uncertainties through MCMC algorithms (see Chapters 7 and 8). Another approach, proposed by Nielsen (2002; Huelsenbeck et al. 2003; Bollback 2006; Minin and Suchard 2008) and known as *stochastic mapping*, samples substitutions on branches of the tree. Used correctly, the samples of substitutions on the tree should lead to the same inference as reconstructed states at interior nodes, but the approach may have a computational advantage at low sequence divergence. For divergent sequences, there may be many substitutions at a site and the approach of sampling substitutions on the tree is computationally expensive.

In this section, I will describe the EB approach to ancestral sequence reconstruction, and discuss the modifications in the hierarchical Bayesian approach. I will also discuss the reconstruction of ancestral states of a discrete morphological character, which can be achieved using the same theory, but the inference involves greater uncertainties due to lack of information to estimate model parameters.

### 4.4.2 *Empirical and hierarchical Bayesian reconstruction*

A distinction can be made between the *marginal* and *joint reconstructions*. The former assigns a character state to a single node, while the latter assigns a set of character states to all ancestral nodes. For instance, given the observed nucleotides at the tips $x_1x_2x_3x_4x_5$ = TCACC in Figure 4.2, $x_6$ = T is a marginal reconstruction for node 6, while $x_0x_6x_7x_8$ = TTTT is a joint reconstruction. Marginal reconstruction is more suitable when one wants the sequence at a particular node, as in the molecular restoration studies. Joint reconstruction is more suitable when one counts changes at each site.

Here we use the example of Figure 4.2 to illustrate the EB approach to ancestral reconstruction. We pretend that the branch lengths and the transition/transversion rate ratio $\kappa$, used in calculating the conditional probabilities (i.e. the $L_i(x_i)$'s shown on the tree), are the true values. In real data analysis, the parameters should be replaced by the MLEs from the data, and furthermore, an unrooted tree should be used when the molecular clock is not assumed.

#### 4.4.2.1 *Marginal reconstruction*

We calculate the posterior probabilities of character states at one ancestral node. Consider node 0, the root. The posterior probability that node 0 has the nucleotide $x_0$, given the data at the site $\mathbf{x}_h$, is

$$f(x_0|\mathbf{x}_h;\theta) = \frac{f(x_0|\theta)f(\mathbf{x}_h|x_0;\theta)}{f(\mathbf{x}_h|\theta)} = \frac{\pi_{x_0}L_0(x_0)}{\sum_{x_0}\pi_{x_0}L_0(x_0)}. \quad (4.23)$$

Note that $\pi_{x_0}L_0(x_0)$ is the joint probability of the state $x_0$ at the root and the states $\mathbf{x}_h$ at the tips. This is calculated by summing over all other ancestral states except $x_0$ (see §4.2.2, for the definition of $L_0(x_0)$, and equation (4.5)). Figure 4.2 shows $L_0(x_0)$, while the prior probability $f(x_0|\theta) = \pi_{x_0} = 1/4$ for any nucleotide $x_0$ under the K80 model. The probability of data at the site is $f(\mathbf{x}_h|\theta) = 0.000509843$. Thus the posterior probabilities at node 0 are 0.055 (= $0.25 \times 0.00011237/0.000509843$), 0.901, 0.037, and 0.007, for T, C, A, and G, respectively. C is the most probable nucleotide at the root, with posterior probability 0.901.

Posterior probabilities at any other interior node can be calculated by moving the root to that node and redoing the calculation using the same algorithm. These are 0.093 (T), 0.829 (C), 0.070 (A), 0.007 (G) for node 6; 0.153 (T), 0.817 (C), 0.026 (A), 0.004 (G) for node 7; and 0.010 (T), 0.985 (C), 0.004 (A), 0.001 (G) for node 8. From these marginal reconstructions, one might guess that the best joint reconstruction is $x_0x_6x_7x_8$ = CCCC, with posterior probability $0.901 \times 0.829 \times 0.817 \times 0.985 = 0.601$. However, this calculation is incorrect, since the states at different nodes are not independent. For example, given that node 0 has $x_0$ = C, the probability that nodes 6, 7, and 8 will have C as well will be much higher than when the state at node 0 is unknown.

The above description assumes the same rate for all sites, but applies also to the fixed-rate (partition) models. In a random-rate model, equation (4.23) still gives the correct posterior probability but both $f(\mathbf{x}_h|\theta)$ and $f(\mathbf{x}_h|x_0;\theta)$ are sums over the rate categories and can similarly be calculated using the pruning algorithm.

*4.4.2.2 Joint reconstruction*

With this approach, we calculate the posterior probability for a set of character states assigned to all interior nodes at a site. Let $\mathbf{y}_h = (x_0, x_6, x_7, x_8)$ be such an assignment or reconstruction.

$$\begin{aligned} f(\mathbf{y}_h|\mathbf{x}_h;\theta) &= \frac{f(\mathbf{x}_h,\mathbf{y}_h|\theta)}{f(\mathbf{x}_h|\theta)} \\ &= \frac{\pi_{x_0} p_{x_0x_6}(t_6) p_{x_6x_7}(t_7) p_{x_7T}(t_1) p_{x_7C}(t_2) p_{x_6A}(t_3) p_{x_0x_8}(t_8) p_{x_8C}(t_4) p_{x_8C}(t_5)}{f(\mathbf{x}_h|\theta)}. \end{aligned} \tag{4.24}$$

The numerator $f(\mathbf{x}_h,\mathbf{y}_h|\theta)$ is the joint probability of the tip states $\mathbf{x}_h$ and the ancestral states $\mathbf{y}_h$, given parameters $\theta$. This is the term in the square brackets in equation (4.3). The probability of data at a site, $f(\mathbf{x}_h|\theta)$, is a sum over all possible reconstructions $\mathbf{y}_h$, while the percentage of contribution from any particular reconstruction $\mathbf{y}_h$ is the posterior probability for that reconstruction.

The difficulty with a naïve use of this formula, as in Yang et al. (1995a) and Koshi and Goldstein (1996a), is the great number of ancestral reconstructions (all combinations of $x_0, x_6, x_7, x_8$). Note that only the numerator is used to compare the different reconstructions, as $f(\mathbf{x}_h|\theta)$ is fixed. Instead of maximizing the product of the transition probabilities to find the best reconstruction, we can maximize the sum of the logarithms of the transition probabilities. The dynamic programming algorithm for determining the best reconstructions for weighted parsimony, described in §3.4.3, can thus be used after the following minor modifications. First, each branch now has its own cost matrix while a single cost matrix was used for all branches for parsimony. Second, the score for each reconstruction involves an additional term $\log(\pi_{x_0})$. The resulting algorithm is equivalent to that described by Pupko et al. (2000). It works under the one-rate and fixed-rate models (the partition models) but not under the random-rate models (the mixture models). For the latter, Pupko et al. (2002a) implemented a branch-and-bound algorithm.

Application of the dynamic programming algorithm to the problem of Figure 4.2 leads to $x_0x_6x_7x_8$ = CCCC as the best reconstruction, with posterior probability 0.784. This agrees with the marginal reconstruction, according to which C is the most probable nucleotide for every node, but the probability is higher than the incorrect value 0.601, mentioned above. The next few best reconstructions can be obtained by a slight extension of the dynamic programming algorithm, as TTTC (0.040), CCTC (0.040), CTTC (0.040), AAAC (0.011), and CAAC (0.011). The above calculations are for illustration only. In real

data analysis, we should use the MLEs of branch lengths and other parameters, and also use an unrooted tree since we are not assuming the clock.

The marginal and joint reconstructions use slightly different criteria. They normally produce consistent results, with the most probable joint reconstruction for a site consisting of character states that are also the best in the marginal reconstructions. Conflicting results may arise when the competing reconstructions have similar probabilities, in which case neither reconstruction is very reliable. The significance of the distinction lies mostly in that one should not multiply the probabilities for the marginal reconstructions to calculate the probability for the joint reconstruction.

#### 4.4.2.3 *Comparison with parsimony*

If we assume the JC69 model with symmetrical substitution rates and also equal branch lengths, the EB and parsimony approaches will give exactly the same rankings of the joint reconstructions. Under JC69, the off-diagonal elements of the transition probability matrix are all equal and smaller than the diagonals: $P_{ij}(t) < P_{ii}(t)$, so that a reconstruction requiring fewer changes will have a higher posterior probability than one requiring more changes (see equation (4.24)). When branch lengths are allowed to differ as they are in a likelihood analysis, and the substitution rates are unequal as under more complex substitution models than JC69, parsimony, and EB may produce different results. Even in such cases, the two approaches are expected to produce very similar results, in that the most parsimonious reconstructions most often have the highest posterior probabilities. In both approaches, the main factor influencing the accuracy of ancestral reconstruction is the sequence divergence level. The reconstruction is less reliable at more variable sites or for more divergent sequences (Yang et al. 1995a; Zhang and Nei 1997). The main advantage of EB over parsimony is that EB provides posterior probabilities as a measure of accuracy.

#### 4.4.2.4 *Hierarchical Bayesian approach*

In real data analysis, parameters $\theta$ in equations (4.23) and (4.24) are replaced by their estimates, say, the MLEs. In large datasets, this may not be a problem as the parameters are reliably estimated. In small datasets, the parameter estimates may involve large sampling errors, so that the EB approach may suffer from inaccurate parameter estimates. For example, if a branch length is estimated to be zero, no change will be possible along that branch during ancestral reconstruction, even though the zero estimate may not be reliable. In such a case, it is advantageous to use a hierarchical (full) Bayesian approach, which assigns a prior on parameters $\theta$ and integrates them out. Such an approach is implemented by Huelsenbeck and Bollback (2001; see also Pagel et al. 2004).

Uncertainty in the phylogeny is a more complex issue. If the purpose of ancestral reconstruction is for use in further analysis, as in comparative methods (Felsenstein 1985b; Harvey and Purvis 1991), one can average over uncertainties in the phylogenies, in substitution parameters, as well as in the ancestral states by sampling from the posterior of these quantities in an MCMC algorithm (Huelsenbeck and Bollback 2001). The most important uncertainty in this regard is probably that of the ancestral states; in other words, use of the most likely ancestral states while ignoring the suboptimal reconstructions may produce different results from averaging over different ancestral reconstructions (see §4.4.4 about biases in ancestral reconstruction). If the purpose is to reconstruct the sequence at a particular ancestral node, the best approach may be to use a fixed phylogeny that is as reliable as possible (e.g. the ML tree). Use of a multifurcating consensus tree is not advisable as the consensus tree is necessarily incorrect.

Averaging over binary trees in a Bayesian algorithm may not have an advantage over using a fixed tree. Hanson-Smith et al. (2010) used simulation to examine whether accommodating uncertainties in the phylogeny improves the accuracy of ancestral sequence reconstruction. They found that use of the ML tree produces robust and accurate reconstructions and that the Bayesian approach of incorporating phylogenetic uncertainties is not necessary or beneficial. When there is phylogenetic uncertainty, the plausible trees produce identical ancestral reconstructions; conversely, when different phylogenies produce different ancestral states, there is little or no ambiguity about the true phylogeny.

### *4.4.3 *Discrete morphological characters*

It may be fitting to discuss here the similar problem of reconstructing morphological characters. Parsimony used to be the predominant method for such analysis. Schluter (1995), Mooers and Schluter (1999), and Pagel (1999) emphasized the importance of quantifying the uncertainty in ancestral reconstruction and championed the likelihood reconstruction. This effort has met with two difficulties. First, their formulation of the likelihood method was not correct. Second, a single morphological character has little information for estimating model parameters, such as branch lengths on the tree and relative substitution rates between characters, and this lack of information makes the analysis highly unreliable. Below I will discuss the first problem but can only lament on the second.

The problem considered by Schluter and Pagel is reconstruction of a binary morphological character evolving under a Markov model with rates $q_{01}$ and $q_{10}$ (see Problem 1.3 in Chapter 1). To reduce the number of parameters to be estimated, all branches on the tree are assumed to have the same length. The correct solution to this problem is the EB approach (equation (4.24)) (Yang et al. 1995a; Koshi and Goldstein 1996a). As discussed above, the likelihood (EB) method of ancestral reconstruction does not directly use the likelihood function. Instead Schluter (1995), Mooers and Schluter (1999), and Pagel (1999) used $f(\mathbf{x}_h, \mathbf{y}_h|\theta)$ of equation (4.24) as the 'likelihood function' for comparing ancestral reconstructions $\mathbf{y}_h$. This is equivalent to the EB approach except that Mooers and Schluter neglected the $\pi_{x_0}$ term, and Pagel used $1/2$ for $\pi_{x_0}$; the prior probabilities at the root should be given by the substitution model as $\pi_0 = q_{10}/(q_{01} + q_{10})$ and $\pi_1 = q_{01}/(q_{01} + q_{10})$. Note that $f(\mathbf{x}_h, \mathbf{y}_h|\theta)$ is the joint probability of $\mathbf{x}_h$ and $\mathbf{y}_h$ and is not the likelihood of $\mathbf{y}_h$. The 'log likelihood ratio' for comparing two states at a node discussed by those authors is the ratio of the posterior probabilities for the two states and cannot be interpreted in the sense of Edwards (1992); thus a 'log likelihood ratio' of 2 means posterior probabilities 0.88 $\left(= e^2/(e^2 + 1)\right)$ and 0.12 for the two states. Pagel (1999) described the EB calculation as the 'global' approach and preferred an alternative 'local approach', in which both substitution parameters $\theta$ and ancestral states $\mathbf{y}_h$ are estimated from the joint probability $f(\mathbf{x}_h, \mathbf{y}_h|\theta)$. This local approach is invalid, as $\theta$ should be estimated from the likelihood $f(\mathbf{x}_h|\theta)$, which averages over all possible ancestral states $\mathbf{y}_h$.

If one insisted on estimating ancestral states from the likelihood function, it would be possible to do so only for the marginal reconstruction at the root. Note that the likelihood function is the probability of the observed data, i.e. the states at the tips. The likelihood function would be $L_0(x_0)$, the probability of data at the site given the state $x_0$ at the root. The placement of the root would then affect the likelihood and the rooted tree should be used. For molecular data, this approach suffers from the problem that the number of parameters grows without bound when the sample size increases. At any rate, it is not

* indicates a more difficult or technical section.

possible to use the likelihood function for the joint reconstruction. For the example of Figure 4.2 for nucleotides

$$f(\mathbf{x}_h|\mathbf{y}_h;\theta) = p_{x_7T}(t_1)p_{x_7C}(t_2)p_{x_6A}(t_3)p_{x_8C}(t_4)p_{x_8C}(t_5). \quad (4.25)$$

This is a function of $x_6, x_7$, and $x_8$ (for nodes that are connected to the tips), and not of $x_0$ (for nodes that are not directly connected to the tips): under the Markov model, given the states at the immediate ancestors of the tips, the states at the tips are independent of states at older ancestors. Thus the likelihood function is independent of states at the interior nodes not directly connected to the tips and it is impossible to infer ancestral states from the likelihood function.

Even with the EB approach, the substitution rates $q_{01}$ and $q_{10}$ cannot be estimated reliably from only one character. One might hope to use an LRT to compare the one-rate $(q_{01} = q_{10})$ and two-rate $(q_{01} \neq q_{10})$ models. However, this test is problematic as the asymptotic $\chi^2$ distribution may not be reliable due to the small sample size (which is one) and more importantly, the failure to reject the one-rate model may reflect a lack of power of the test rather than a genuine symmetry of the rates. Analyses by a number of authors (e.g. Mooers and Schluter 1999) suggest that ancestral reconstruction is sensitive to the assumption of rate symmetry. Another worrying assumption is that of equal branch lengths. This means that the expected amount of evolution is the same for every branch on the tree, and is more unreasonable than the clock assumption (rate constancy over time). An alternative may be to use estimates of branch lengths obtained from molecular data, but this is open to the criticism that the molecular branch lengths may not reflect the amount of evolution in the morphological character.

An approach to dealing with uncertainties in the parameters is hierarchical Bayesian, which averages over uncertainties in substitution rates and branch lengths through a prior. Schultz and Churchill (1999) implemented such an algorithm, and found that the posterior probabilities for ancestral states are very sensitive to priors on relative substitution rates, even in seemingly ideal situations where the parsimony reconstruction is unambiguous. Those studies highlight the misleading overconfidence of parsimony as well as the extreme difficulty of reconstructing ancestral states for a single character. The Bayesian approach does offer the consolation that if different inferences are drawn from the same data under the same model, they must be due to differences in the prior.

In general, classical statistical approaches such as ML are not expected to work well in datasets consisting of one sample point, or in problems involving as many parameters as the observed data. When ML is applied to analyse one or two morphological characters (Pagel 1994; Lewis 2001), the asymptotic theory for MLEs and LRTs may not apply.

### 4.4.4 *Systematic biases in ancestral reconstruction*

The ancestral character states reconstructed by parsimony or likelihood (EB) are our best guesses under the respective criteria. However, if they are used for further statistical analyses or tests, one should bear in mind that they are inferred pseudo-data rather than real observed data. They involve random errors due to uncertainties in the reconstruction. Worse still, use of only the best reconstructions while ignoring the suboptimal ones can cause systematic biases. A number of authors (Collins et al. 1994; Perna and Kocher 1995; Eyre-Walker 1998) have discussed such biases with parsimony reconstruction. They also exist with likelihood (EB) reconstruction. The problem lies not in the use of parsimony versus likelihood or Bayesian for ancestral reconstruction, but in the use of the optimal reconstructions while ignoring the suboptimal ones. For example,

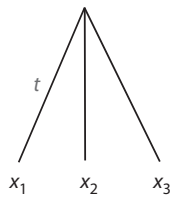


**Fig. 4.10** A tree of three species for demonstrating the bias in ancestral reconstruction. The three branch lengths are equal, at $t = 0.2$ substitutions per site.

using simulation, Williams et al. (2006) found that use of the optimal reconstructions by parsimony and likelihood leads to overestimated thermostability of ancestral proteins, while a Bayesian method that sometimes chooses less probable residues from the posterior probability distribution does not.

As an example, consider the star tree of three sequences of Figure 4.10. Suppose that the substitution process has been stationary and followed the model of Felsenstein (1981), with parameters $\pi_T = 0.2263$, $\pi_C = 0.3282$, $\pi_A = 0.3393$, and $\pi_G = 0.1062$ (these frequencies are from the human mitochondrial D-loop hypervariable region I). Suppose that each branch length is 0.2 nucleotide substitutions per site, so that the transition probability matrix is

$$P(0.2) = \begin{bmatrix} 0.811138 & 0.080114 & 0.082824 & 0.025924 \\ 0.055240 & 0.836012 & 0.082824 & 0.025924 \\ 0.055240 & 0.080114 & 0.838722 & 0.025924 \\ 0.055240 & 0.080114 & 0.082824 & 0.781821 \end{bmatrix}. \tag{4.26}$$

We use the correct model and branch lengths to calculate posterior probabilities for ancestral states at the root, and examine the frequencies of A and G in the reconstructed ancestral sequence. This mimics studies which use ancestral reconstruction to detect possible drift in base compositions. At sites with data AAG, AGA, and GAA, the posterior probabilities for the states at the root are 0.006 (T), 0.009 (C), 0.903 (A), and 0.083 (G), so that A is much more likely than G (see Problem 4.4). However, if we use A and ignore G at every such site, we will over-count A and under-count G. Similarly, at sites with data GGA, GAG, and AGG, the posterior probabilities are 0.002 (T), 0.003 (C), 0.034 (A), and 0.960 (G). The best reconstruction is G, so that we over-count G and under-count A at such sites. The greater frequency of A than of G in the data means that there are more sites with data AAG, AGA, and GAA (with probability 0.02057) than sites with data GGA, GAG, and AGG (with probability 0.01680). The net bias is then an over-count of A and under-count of G in the ancestor. Indeed, the base compositions in the reconstructed sequence for the root are 0.212 (T), 0.324 (C), 0.369 (A), and 0.095 (G), more extreme than the frequencies in the observed sequences. Thus ancestral reconstruction indicates an apparent gain of the rare nucleotide G over the time of evolution from the root to the present. As the process is in fact stationary, this apparent drift in base compositions is an artefact of the EB (and parsimony) reconstruction, caused by ignoring the suboptimal reconstructions at every site.

Jordan et al. (2005) used parsimony to reconstruct ancestral protein sequences and observed a systematic gain of rare amino acids (and loss of common ones) over evolutionary time. The trend is the same as discussed above, and appears to be an artefact of ancestral reconstruction (Goldstein and Pollock 2006). Perna and Kocher (1995) studied the use of parsimony to infer ancestral states and then to count changes along branches to estimate the substitution rate matrix. The bias involved in such an analysis appears even greater than in counts of nucleotides discussed

above. Clearly, the problem is more serious for more divergent sequences since ancestral reconstruction is poorer, but the bias can be considerable even in datasets of closely related sequences, such as the human mitochondrial D-loop sequences (Perna and Kocher 1995) or population data (Hernandez et al. 2007).

Despite those caveats, the temptation to infer ancestors and use them to perform all sorts of statistical tests appears too great to resist. Ancestral reconstruction is thus used frequently, with many interesting and spurious discoveries being made all the time.

Instead of ancestral reconstruction, one should try to rely on a likelihood-based approach, which sums over all possible ancestral states, weighting them appropriately according to their probabilities of occurrence (see §4.2). For example, the relative rates of substitutions between nucleotides or amino acids can be estimated by using a likelihood model (Yang 1994b; Adachi and Hasegawa 1996a; Yang et al. 1998; Whelan and Goldman 2001; Le and Gascuel 2008). Possible drifts in nucleotide compositions may be tested by implementing models that allow different base frequency parameters for different branches (Yang and Roberts 1995; Galtier and Gouy 1998). Chapter 11 discusses a few more examples in which both ancestral reconstruction and full likelihood-based approaches are used to analyse protein-coding genes to detect positive selection.

If a likelihood analysis under the model is too complex and one has to resort to ancestral reconstruction, a heuristic approach to reducing the bias may be to use the suboptimal as well as the optimal reconstructions in the analysis. One may use a simpler existing likelihood model to calculate posterior probabilities for ancestral states and use them as weights to accommodate both optimal and suboptimal reconstructions. In the example above, the posterior probabilities for the root state at a site with data AAG are 0.006 (T), 0.009 (C), 0.903 (A), and 0.083 (G). Instead of using A for the site and ignoring all other states, one can use both A and G, with weights 0.903 and 0.083 (rescaled so that they sum to one). If we use all four states at every site in this way, we will recover the correct base compositions for the root sequence with no bias at all, since the posterior probabilities are calculated under the correct model. If the likelihood model assumed in ancestral reconstruction is too simplistic and incorrect, the posterior probabilities (weights) will be incorrect as well. Even so this approach may be less biased than ignoring suboptimal reconstructions entirely. Akashi et al. (2007) applied this approach to count changes between preferred (frequently used) and unpreferred (rarely used) codons in protein-coding genes in the *Drosophila melanogaster* species subgroup and found that it helped to reduce the bias in ancestral reconstruction. Similarly Dutheil et al. (2005) used reconstructed ancestral states to detect coevolving nucleotide positions, indicated by an excess of substitutions at two sites that occur along the same branches. They calculated posterior probabilities for ancestral states under an independent-site model and used them as weights to count substitutions along branches at the two sites under test.

## *4.5 Numerical algorithms for maximum likelihood estimation

The likelihood method estimates parameters $\theta$ by maximizing the log likelihood $\ell$. In theory, one may derive the estimates by setting to zero the first derivatives of $\ell$ with respect to $\theta$ and solving the resulting system of equations, called the *likelihood equations*:

$$\frac{\partial \ell}{\partial \theta} = 0. \tag{4.27}$$

* indicates a more difficult or technical section.

This approach leads to analytical solutions to pairwise distance estimation under the JC69 and K80 models, as discussed in §1.4. For three species, the analytical solution is possible only in the simplest case of binary characters evolving under the molecular clock (Yang 2000a). The problem becomes intractable as soon as we consider the case of nucleotides with four states (Problem 4.2) or the case of four species. The latter case, of four species under the clock, was studied by Chor and Snir (2004), who derived analytical estimates of branch lengths for the 'fork' tree ((a, b), (c, d)) but not for the 'comb' tree (((a, b), c), d).

In general, numerical iterative algorithms have to be used to maximize the log likelihood. Developing a reliable and efficient optimization algorithm for practical problems is a complicated task. This section gives only a flavour of such algorithms. Interested readers should consult a textbook on nonlinear programming or numerical optimization, such as Gill et al. (1981) and Fletcher (1987).

Maximization of a function (called the *objective function*) is equivalent to minimization of its negative. Below, we will follow the convention and describe our problem of likelihood maximization as a problem of minimization. The objective function is thus the negative log likelihood: $f(\theta) = -\ell(\theta)$. Note that algorithms that reach the minimum with fewer function evaluations are more efficient, since $f(\theta)$ is expensive to calculate.

### *4.5.1 *Univariate optimization*

If the problem is one-dimensional, the algorithm is called *line search* since the search is along a line. Suppose we determine that the minimum is in the interval $[a, b]$. This is called the *interval of uncertainty*. Most line search algorithms reduce this interval successively, until its width is smaller than a pre-specified small value. Assuming that the function is unimodal in the interval (i.e. there is only one valley between $a$ and $b$), we can reduce the interval of uncertainty by comparing the function values at two interior points $\theta_1$and $\theta_2$ (Figure 4.11). Different schemes exist concerning choices of the points. Here we describe the golden section search.

#### *4.5.1.1 Golden section search*

Suppose the interval of uncertainty is [0, 1]; this can be rescaled to become $[a, b]$. We place two interior points at $\gamma$ and $(1-\gamma)$, where the golden ratio $\gamma \approx 0.6180$ satisfies $\gamma/(1-\gamma) = 1/\gamma$. The new interval becomes $(1-\gamma, 1)$ if $f(\gamma) < f(1-\gamma)$ or $(0, \gamma)$ otherwise (Figure 4.12). No matter how the interval is reduced, one of the two points will be in

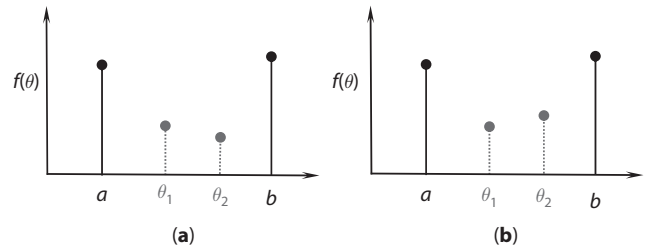


**Fig. 4.11** Reduction of the interval of uncertainty $(a, b)$, which contains the minimum. The objective function $f$ is evaluated at two interior points $\theta_1$ and $\theta_2$. (**a**) If $f(\theta_1) \geq f(\theta_2)$, the minimum must lie in the interval $(\theta_1, b)$. (**b**) Otherwise if $f(\theta_1) < f(\theta_2)$, the minimum must lie in the interval $(a, \theta_2)$.

* indicates a more difficult or technical section.

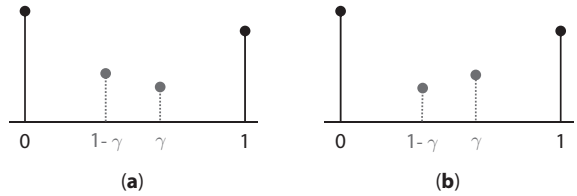


**Fig. 4.12** The golden section search. Suppose the minimum is inside the interval (0, 1). Two points are placed at $\gamma$ and $(1 - \gamma)$ of the interval, where $\gamma = 0.6180$. (**a**) If $f(\gamma) < f(1 - \gamma)$, the new interval will be $(1 - \gamma, 1)$. (**b**) If $f(\gamma) \geq f(1 - \gamma)$, the new interval will be $(0, \gamma)$. No matter how the interval is reduced, one of the two points will be in the correct position inside the new interval.

the correct position inside the new interval for the next iteration. With the golden section search, the interval of uncertainty is reduced by $\gamma$ at each step. The algorithm is said to have a linear convergence rate.

*4.5.1.2 Newton's method and polynomial interpolation*

For smooth functions, more efficient algorithms can be implemented by approximating $f$ using simple functions whose minimum can be obtained analytically. For example, if we approximate $f$ by a parabola (quadratic), of the form

$$\tilde{f} = a\theta^2 + b\theta + c, \tag{4.28}$$

with $a > 0$, then $\tilde{f}$ has a minimum at $\theta^* = -b/(2a)$. If the function value and its first and second derivatives at the current point $\theta_k$ are known, we can use the first three terms of the Taylor expansion to approximate $f(\theta)$:

$$\tilde{f}(\theta) = f(\theta_k) + f'(\theta_k)(\theta - \theta_k) + \frac{1}{2}f''(\theta_k)(\theta - \theta_k)^2. \tag{4.29}$$

This is a quadratic function in $\theta$, in the form of equation (4.28), with $a = f''(\theta_k)/2$ and $b = f'(\theta_k) - f''(\theta_k)\theta_k$. If $f''(\theta_k) > 0$, the quadratic (4.29) achieves its minimum at

$$\theta_{k+1} = -\frac{b}{2a} = \theta_k - \frac{f'(\theta_k)}{f''(\theta_k)}. \tag{4.30}$$

As $f$ may not be a quadratic, $\theta_{k+1}$ may not be the minimum of $f$. We thus use $\theta_{k+1}$ as the new current point to repeat the algorithm. This is Newton's method, also known as the Newton–Raphson method.

Newton's method is highly efficient. Its rate of convergence is quadratic, meaning that, roughly speaking, the number of correct figures in $\theta_k$ doubles at each step (e.g. Gill et al. 1981, p. 57). A problem, however, is that it requires the first and second derivatives, which may be expensive, troublesome, or even impossible to compute. Without the derivatives, a quadratic approximation can be constructed by using the function values at three points. Similarly a cubic polynomial can be constructed by using the function values and first derivatives (but not the second derivatives) at two points. It is generally not worthwhile fitting high-order polynomials. Another serious problem with Newton's method is that its fast convergence rate is only local, and if the iteration is not close to the minimum, the algorithm may diverge hopelessly. The iteration may also encounter numerical

difficulties if $f''(\theta_k)$ is zero or too small. Thus it is important to obtain good starting values for Newton's method, and certain safeguards are necessary to implement the algorithm.

A good strategy is to combine a guaranteed reliable method (such as golden section) with a rapidly convergent method (such as Newton's quadratic interpolation), to yield an algorithm that will converge rapidly if $f$ is well-behaved, but is not much less efficient than the guaranteed method in the worst case. Suppose a point $\tilde{\theta}$ is obtained by quadratic interpolation. We can check to make sure that $\tilde{\theta}$ lies in the interval of uncertainty $(a, b)$ before evaluating $f(\tilde{\theta})$. If $\tilde{\theta}$ is too close to the current point or too close to either end of the interval, one may revert to the golden section search. Another idea of safeguarding Newton's method is to redefine

$$\theta_{k+1} = \theta_k - \alpha f'(\theta_k)/f''(\theta_k), \tag{4.31}$$

where the step length $\alpha$, which equals 1 initially, is repeatedly halved until the algorithm is non-increasing, that is, until $f(\theta_{k+1}) < f(\theta_k)$.

### *4.5.2 *Multivariate optimization*

Most models used in likelihood analysis in molecular phylogenetics include multiple parameters, and the optimization problem is multidimensional. A naïve approach is to optimize one parameter at a time with all other parameters fixed. However, this is inefficient when the parameters are correlated. Figure 4.13 shows an example in which the two parameters are (positively) correlated. A search algorithm that updates one parameter at a time makes impressive improvements to the objective function initially, but becomes slower and slower when it is close to the minimum. As every search direction is at a 90° angle to the previous search direction, the algorithm zigzags in tiny baby steps. Standard optimization algorithms update all variables simultaneously.

#### 4.5.2.1 *Steepest-descent search*

Many optimization algorithms use the first derivatives, $g = \mathrm{d}f(\theta)$, called the *gradient*. The simplest among them is the *steepest-descent* algorithm (or *steepest-ascent* for maximization). It finds the steepest-descent direction, locates the minimum along that direction,

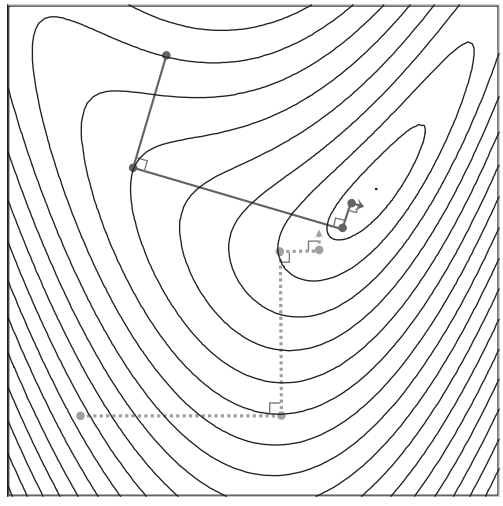

**Fig. 4.13** The log likelihood contour when two parameters are positively correlated. A search algorithm changing one parameter at a time (the dotted lines and arrow) is very inefficient. Similarly the steepest-ascent search (the solid lines and arrow) is inefficient because its search direction is always perpendicular to the previous search direction.

* indicates a more difficult or technical section.

and repeats the procedure until convergence. The gradient $g$ is the direction that the function increases the quickest locally and is thus *the steepest-ascent direction*, while $-g$ is the *steepest-descent direction*. Note that the minimum along a search direction occurs when the search direction becomes a tangent line to a contour curve. At that point, the new gradient is perpendicular to the tangent line or the previous search direction (Figure 4.13). The steepest-descent algorithm suffers from the same problem as the naïve algorithm of changing one variable at a time: every search direction forms a 90° angle with the previous search direction. The algorithm descends very quickly initially but becomes slower and slower when it is close to the minimum.

#### 4.5.2.2 *Newton's method*

The multivariate version of the Newton algorithm relies on quadratic approximation to the objective function. Let $G = \mathrm{d}^2 f(\theta)$ be the *Hessian matrix* of second partial derivatives. With $p$ variables, both $\theta$ and $g$ are $p \times 1$ vectors while $G$ is a $p \times p$ matrix. A second-order Taylor expansion of $f$ around the current point $\theta_k$ gives

$$f(\theta) \approx f(\theta_k) + g_k^{\mathrm{T}}(\theta - \theta_k) + \frac{1}{2}(\theta - \theta_k)^{\mathrm{T}} G_k (\theta - \theta_k), \tag{4.32}$$

where the superscript $^{\mathrm{T}}$ means transpose. By minimizing the right-hand side of equation (4.32), i.e. by setting its gradient to 0, one obtains the next iterate as

$$\theta_{k+1} = \theta_k - G_k^{-1} g_k. \tag{4.33}$$

Note that this is the same as equation (4.30) for the univariate case. Similarly, the multivariate version shares the rapid convergence rate as well as the major drawbacks of the univariate algorithm; i.e. the method requires calculation of the first and second derivatives and may diverge when the iterate is not close enough to the minimum. A common strategy is to take $s_k = G_k^{-1} g_k$ as a search direction, called the *Newton direction*, and to perform a line search to determine how far to go along that direction.

$$\theta_{k+1} = \theta_k + \alpha s_k = \theta_k - \alpha G_k^{-1} g_k. \tag{4.34}$$

Here $\alpha$ is called the *step length*. It is often too expensive to optimize $\alpha$ in this way. A simpler version is to try $\alpha = 1, 1/2, 1/4, \ldots,$ until $f(\theta_{k+1}) \leq f(\theta_k)$. This is sometimes known as the *safeguided Newton algorithm*. When $G_k$ is not *positive definite*, one can reset it to the identity matrix: $G_k = I$.

When the objective function is the negative log likelihood, $f = -\ell$, the Hessian matrix $G = -\mathrm{d}^2 \ell(\theta)$ is also called the *observed information matrix*. In some simple statistical problems, the expected information, $-E\{\mathrm{d}^2 \ell(\theta)\}$, may be easier to calculate, and can be used in Newton's algorithm. The method is then known as *scoring*. Both Newton's method and scoring have the benefit that the approximate variance–covariance matrix of the MLEs is readily available at the end of the iteration. We used such approximate variances in §1.4.1.

#### 4.5.2.3 *Quasi-Newton methods*

Quasi-Newton methods include a class of methods that require first derivatives but not second derivatives. While Newton's method calculates the second derivatives $G$ at every current point, quasi-Newton methods build up information about $G$ or its inverse from the calculated values of the objective function $f$ and the first derivatives $g$ during the iteration. If the first derivatives are not available, they may be calculated using the difference

approximation. Without the need for second or even first derivatives, quasi-Newton algorithms greatly increased the range of problems that can be solved. The basic algorithm can be sketched as follows:

a. Supply an initial guess $\theta_0$.
b. For $k = 0, 1, 2, \ldots$, until convergence:
   1. Test $\theta_k$ for convergence;
   2. Calculate a search direction $s_k = -B_k g_k$;
   3. Perform a line search along $s_k$ to determine the step length $\alpha_k : \theta_{k+1} = \theta_k + \alpha_k s_k$;
   4. Update $B_k$ to give $B_{k+1}$.

Here $B_k$ is a symmetric positive definite matrix, which can be interpreted as an approximation to $G_k^{-1}$, the inverse of the Hessian matrix. Note the similarity of this algorithm to Newton's method. In step b3, the scalar $\alpha_k > 0$ is chosen to minimize $f(\theta_{k+1})$, by using a line search algorithm as discussed above. A number of strategies have been developed to update the matrix $B$. Well-known ones include the Broyden–Fletcher–Goldfard–Shanno (BFGS) and Davidon–Fletcher–Powell (DFP) formulae. See Gill et al. (1981) and Fletcher (1987) for details.

When the first derivatives are impossible or expensive to calculate, an alternative approach is to use a *derivative-free method*. See Brent (1973) for discussions of such methods. According to Gill et al. (1981), quasi-Newton methods, even if the first derivatives are calculated using the difference approximation, are more efficient than derivative-free methods.

#### 4.5.2.4 *Bounds and constraints*

The discussion up to now has assumed that the parameters are unconstrained and can take values over the whole real line. In most practical problems, parameters have bounds. For example, branch lengths should be nonnegative, and the nucleotide frequency parameters $\pi_1, \pi_2, \pi_3$ should satisfy the following constraints: $\pi_1, \pi_2, \pi_3 > 0$ and $\pi_1 + \pi_2 + \pi_3 < 1$. Even unconstrained optimization can be improved by having rough guesses of the parameter values and applying bounds on the parameters.

Algorithms for constrained optimization are much more complex, with simple lower and upper bounds being easier than general linear inequality constraints (Gill et al. 1981). Variable transform is often an effective approach for dealing with linear inequality constraints. For example, to estimate the nucleotide frequencies $(\pi_1, \pi_2, \pi_3, \pi_4)$, we can use new variables $-\infty < x_1, x_2, x_3 < \infty$, with $x_4 = 0, \pi_1 = x_1/s, \pi_2 = x_2/s, \pi_3 = x_3/s$, and $\pi_4 = 1/s$, where $s = e^{x_1} + e^{x_2} + e^{x_3} + 1$. Another transform is $x_1 = \pi_1/\pi_4, x_2 = \pi_2/\pi_4, x_3 = \pi_3/\pi_4$ so that $0 < x_1, x_2, x_3 < \infty$, but this is often less efficient than the exponential transform. As another example, consider estimation of divergence times (node ages) $t_0$, $t_1$, $t_2$, $t_3$ on a five-species tree, with the constraints $t_0 > t_1 > t_2 > t_3$. We can define new variables $x_0 = t_0$ (for the root age), $x_1 = t_1/t_0, x_2 = t_2/t_1, x_3 = t_3/t_2$, so that the new variables have simple bounds: $0 < x_0 < \infty, 0 < x_1, x_2, x_3 < 1$. In general the new variable $x_i$ for any non-root node $i$ is defined as the ratio of the node age $(t_i)$ to the age of its mother node.

## 4.6 ML optimization in phylogenetics

### 4.6.1 *Optimization on a fixed tree*

In a phylogenetic problem, the parameters to be estimated include branch lengths in the tree and parameters in the substitution model. Given the values of parameters, one can

use the pruning algorithm to calculate the log likelihood. In theory, one can then apply any of the general purpose optimization algorithms discussed above to find the MLEs iteratively. However, this will almost certainly produce an inefficient algorithm. Consider the recursive calculation of the conditional probabilities $L_i(x_i)$ in the pruning algorithm. When a branch length changes, $L_i(x_i)$ for only those nodes ancestral to that branch are changed, while those for all other nodes are not affected. Direct application of a general-purpose multivariate optimization algorithm thus leads to many duplicated calculations of the same quantities.

To take advantage of such features of likelihood calculation on a tree, one can optimize one branch length at a time, keeping all other branch lengths and substitution parameters fixed. Suppose one branch connects nodes $a$ and $b$. By moving the root to coincide with node $a$, we can rewrite equation (4.6) as

$$f(\mathbf{x}_h|\theta) = \sum_{x_a} \sum_{x_b} \pi_{x_a} p_{x_a x_b}(t_b) L_a(x_a) L_b(x_b). \tag{4.35}$$

The first and second derivatives of $\ell$ with respect to $t_b$ can then be calculated analytically (Adachi and Hasegawa 1996b; Yang 2000b), so that $t_b$ can be optimized efficiently using Newton's algorithm. One can then estimate the next branch length by moving the root to one of its ends. A change to any substitution parameter, however, typically changes the conditional probabilities for all nodes, so saving is not possible. To estimate substitution parameters, two strategies appear possible. Yang (2000b) tested an algorithm that cycles through two phases. In the first phase, branch lengths are optimized one by one while all substitution parameters are held fixed. Several cycles through the branch lengths are necessary to achieve convergence. In the second phase, the substitution parameters are optimized using a multivariate optimization algorithm such as BFGS, with branch lengths fixed. This algorithm works well when the branch lengths and substitution parameters are not correlated, for example, under the HKY85 or GTR (REV) models, in which the transition/transversion rate ratio $\kappa$ for HKY85 or the rate ratio parameters in GTR are not strongly correlated with the branch lengths. However, when there is strong correlation, the algorithm can be very inefficient. This is the case with the gamma model of variable rates at sites, in which the branch lengths and the gamma shape parameter $\alpha$ often have strong negative correlations. A second strategy (Swofford 2000) is to embed the first phase of the above algorithm into the second phase. One uses a multivariate optimization algorithm (such as BFGS) to estimate the substitution parameters, with the log likelihood for any given values of the substitution parameters calculated by optimizing the branch lengths. It may be necessary to optimize the branch lengths to a high precision, as inaccurate calculations of the log likelihood (for given substitution parameters) may cause problems for the BFGS algorithm, especially if the first derivatives are calculated numerically using difference approximation.

### 4.6.2 *Multiple local peaks on the likelihood surface for a fixed tree*

Numerical optimization algorithms discussed above are all local hill-climbing algorithms. They converge to a local peak, but may not reach the globally highest peak if multiple local peaks exist. Fukami and Tateno (1989) presented a proof that under the F81 model (Felsenstein 1981) and on a tree of any size, the log likelihood curve for one branch length has a single peak when other branch lengths are fixed. Tillier (1994) suggested that the result applies to more general substitution models as well. However, Steel (1994a) pointed out that this result does not guarantee one single peak in the whole parameter

space. Consider a two-parameter problem and imagine a peak in the northwest region and another peak in the southeast region of the parameter space. Then there is always one peak if one looks only in the north–south direction or west–east direction, although in fact two local peaks exist. Steel (1994a) and Chor et al. (2000) further constructed counterexamples to demonstrate the existence of multiple local peaks for branch lengths even on small trees with four species.

Nevertheless, local peaks do not appear to be common in real data analysis unless the assumed model is complex and parameter-rich. A symptom of the problem is that different runs of the same analysis starting from different initial values may lead to different results. Rogers and Swofford (1999) used computer simulation to examine the problem, and reported that local peaks were less common for the ML tree than for other poorer trees. It is hard to imagine that the likelihood surfaces are qualitatively different for the different trees, so one possible reason for this finding may be that more local peaks exist at the boundary of the parameter space (say, with zero branch lengths) for the poor trees than for the ML tree. Existence of multiple local peaks is sometimes misinterpreted as an indication of the unrealistic nature of the assumed substitution model. In fact multiple local peaks are much less common under simplistic and unrealistic models like JC69 than under more realistic parameter-rich models.

There is no foolproof remedy to the problem of local peaks. A simple strategy is to run the iteration algorithm multiple times, starting from different initial values. If multiple local peaks exist, one should use the estimates corresponding to the highest peak. Stochastic search algorithms that allow downhill moves, such as simulated annealing and genetic algorithms, are useable as well (see §3.2.5).

### 4.6.3 *Search in the tree space*

The above discussion concerns estimation of branch lengths and substitution parameters for a given tree topology. This will be sufficient if our purpose is to estimate parameters in the substitution model or to test hypotheses concerning them, with the tree topology known or fixed. This is a straightforward application of the conventional ML estimation when the likelihood function $L(\theta; X) = f(X|\theta)$ is fully specified.

If our interest is in reconstruction of the phylogeny, we take the optimized log likelihood for the tree as the score for tree selection, and should, at least in theory, solve as many optimization problems as the number of tree topologies. This is far more complex than the conventional ML estimation. We have two levels of optimization: one of optimizing branch lengths (and substitution parameters) on each fixed tree to calculate the tree score and the other of searching for the best tree in the tree space.

An analytical characterization of the tree space is available for the case of rooted trees for three species with binary characters evolving under the molecular clock (Yang 2000a; see also Newton 1996; Steel 2011). This is the simplest problem of tree reconstruction. The transition probability matrix under the model is given in equation (1.79). The three binary rooted trees are $\tau_1 = ((1, 2), 3)$, $\tau_2 = ((2, 3), 1)$, and $\tau_3 = ((3, 1), 2)$, shown in Figure 4.14, where $t_{i0}$ and $t_{i1}$ are the two branch lengths in tree $\tau_i$, for $i = 1, 2, 3$. The star tree $\tau_0$ has only one branch length $t_{01}$. The data are summarized as the counts $\boldsymbol{n} = (n_0, n_1, n_2, n_3)$ or frequencies $(f_0, f_1, f_2, f_3)$ of the four site patterns: *xxx*, *xxy*, *yxx*, and *xyx*, where *x* and *y* are any two distinct characters. Since $f_0 + f_1 + f_2 + f_3 = 1$, the sample space, i.e. the space of all possible datasets, is thus the tetrahedron OABC of Figure 4.15. Let $p_0, p_1, p_2, p_3$ be the probabilities of the site patterns given a tree and its branch lengths. The likelihood functions for the three trees are then

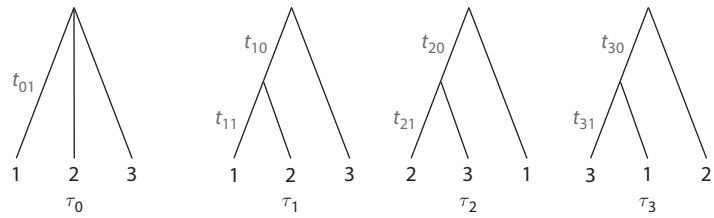


**Fig. 4.14** The three rooted trees for three species: $\tau_1, \tau_2,$ and $\tau_3$. Branch lengths $t_{i0}$ and $t_{i1}$ in each tree $\tau_i$ ($i = 1, 2, 3$) are measured by the expected number of character changes per site. The star tree $\tau_0$ is also shown with its branch length $t_{01}$.

$$\begin{aligned} f(\boldsymbol{n}|\tau_1, t_0, t_1) &= p_0^{n_0} p_1^{n_1} p_2^{n_2+n_3}, \\ f(\boldsymbol{n}|\tau_2, t_0, t_1) &= p_0^{n_0} p_1^{n_2} p_2^{n_3+n_1}, \\ f(\boldsymbol{n}|\tau_3, t_0, t_1) &= p_0^{n_0} p_1^{n_3} p_2^{n_1+n_2}, \end{aligned} \tag{4.36}$$

where the probabilities for the site patterns are

$$\begin{aligned} p_0(t_0, t_1) &= \tfrac{1}{4} + \tfrac{1}{4}e^{-4t_1} + \tfrac{1}{2}e^{-4(t_0+t_1)}, \\ p_1(t_0, t_1) &= \tfrac{1}{4} + \tfrac{1}{4}e^{-4t_1} - \tfrac{1}{2}e^{-4(t_0+t_1)}, \\ p_2(t_0, t_1) &= \tfrac{1}{4} - \tfrac{1}{4}e^{-4t_1} = p_3(t_0, t_1). \end{aligned} \tag{4.37}$$

One may use the site pattern probabilities as parameters and represent $\tau_1$ as $p_0 > p_1 > p_2 = p_3$, $\tau_2$ as $p_0 > p_2 > p_3 = p_1$, and $\tau_3$ as $p_0 > p_3 > p_1 = p_2$, while the star tree is $p_0 > p_1 = p_2 = p_3$. Those parameter spaces are superimposed on the sample space as well. The (probability) space for tree $\tau_1$ with parameters $0 < t_{10}, t_{11} < \infty$ corresponds to the triangle *OPR*; in other words, $p_1$ and $p_2$ in the triangle (with $0 < p_2 = p_3 \le p_1 \le p_0$) is a reparametrization of $t_{10}$ and $t_{11}$ (with $0 < t_{10}, t_{11} < \infty$) (Problem 4.1). Similarly the parameter space of $\tau_2$ is the triangle *OPS* and that for $\tau_3$ is *OPT*. The probability spaces of the four trees thus form a 'paper airplane' (Newton 1996) or 'paper dart' (Steel 2011).

For a given dataset or data point $(f_1, f_2, f_3)$, the MLE of $t_{10}$ and $t_{11}$ for $\tau_1$ corresponds to the point in the triangle *OPR* closest to the data point, with the distance measured by the Kullback–Leibler (K-L) divergence. The K-L divergence from distribution $p$ to distribution $f$ is defined as

$$D_{\mathrm{KL}}(f, p) = \sum_i f_i \log \frac{f_i}{p_i} = \sum_i f_i \log f_i - \sum_i f_i \log p_i. \tag{4.38}$$

Note that $\sum_i f_i \log f_i$ is a constant when the data are observed, while $n \sum_i f_i \log p_i = \sum_i n_i \log p_i$ is the log likelihood, so that minimization of $D_{\mathrm{KL}}$ is equivalent to maximization of the log likelihood. Finally, if the data point is closer (by $D_{\mathrm{KL}}$) to triangle *OPR* than it is to *OPS* or *OPT*, $\tau_1$ will be the ML tree. The full likelihood solution is given by Yang (2000a, Table 4) and summarized in Figure 4.15: the ML tree is $\tau_1$, $\tau_2$, or $\tau_3$, if the data point is in the region *OPFAD*, *OPDBE*, or *OPECF*, respectively.

Thus search in the space of trees is more complex than optimization in the conventional ML estimation. In the latter, one can define the gradient (the direction of steepest ascent) and local curvature (Figure 4.13) and use a quadratic approximation to predict the peak of

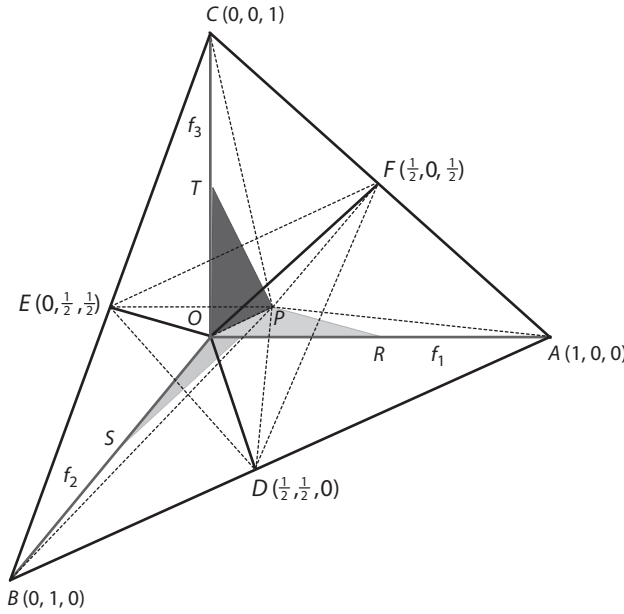


**Fig. 4.15** The sample space (the space of all possible datasets) and the parameter space (the space of all trees) for the phylogeny–inference problem of Figure 4.14. Binary characters evolving under the clock with symmetrical substitution rates are used to reconstruct the rooted trees for three species. Each dataset (i.e. the alignment of three sequences) can be summarized as the counts $(n_0, n_1, n_2, n_3)$ or proportions $(f_0, f_1, f_2, f_3)$ of sites with the four site patterns *xxx*, *xxy*, *yxx*, and *xyx*. Since $f_0 = 1 - (f_1 + f_2 + f_3)$, each dataset is a point in the $f_1 - f_2 - f_3$ space or a point inside the tetrahedron *OABC*. The sample space *OABC* is partitioned into four regions, corresponding to the four trees; tree $\tau_i$ is the ML tree if and only if the data point falls within region $i$ ($i = 0, 1, 2, 3$). The region for $\tau_0$ is the line segment *OP* plus the tetrahedron *PDEF*. In this region, the three binary trees have the same likelihood as the star tree, so $\tau_0$ is taken as the ML tree. The region for $\tau_1$ is a contiguous block *OPFAD*, consisting of three tetrahedrons *OPAD*, *OPAF*, and *PDAF*. The regions for $\tau_2$ and $\tau_3$ are *OPDBE* and *OPECF*, respectively. The parameter (probability) space for each tree is superimposed onto the sample space. For $\tau_0$ this is line segment *OP*, corresponding to $0 \le f_1 = f_2 = f_3 < 1/4$ or $0 \le t_{01} < \infty$. The parameter spaces for $\tau_1, \tau_2$, and $\tau_3$ are triangles *OPR*, *OPS*, or *OPT*, respectively. The coordinates for the points are $O(0, 0, 0)$, $P(\frac{1}{4}, \frac{1}{4}, \frac{1}{4})$, $R(1/2, 0, 0)$, $S(0, 1/2, 0)$, and $T(0, 0, 1/2)$. Phylogeny reconstruction thus corresponds to partitioning the sample space and may also be viewed as projecting the observed data point onto the three parameter planes, with the distance measured by the K-L divergence. Redrawn after Yang (2000a).

the surface, as in the Newton or quasi-Newton algorithms. In tree search, such concepts are not meaningful when one moves from one tree (one triangle) to another (another triangle). With more species, the huge number of trees have intricate relationships among themselves, and the landscape may be complex, with local peaks (see Figure 3.18) and plateaus (Charleston 1995).

In practice, tree search by ML does not attempt to determine the best search direction in the tree space, but instead uses tree-rearrangement algorithms (such as NNI or SPR) to move between trees. Because the neighbouring trees generated during branch swapping share subtrees, tree search algorithms should avoid repeated computations of the same quantities. Much effort has been expended to develop fast algorithms for likelihood tree search, such as the FASTDNAML algorithm of Olsen et al. (1994), the genetic algorithms of Lewis (1998) and Lemmon and Milinkovitch (2002), and the parallel likelihood programs PHYML (Guindon and Gascuel 2003) and RAXML (Stamatakis 2006, Stamatakis et al. 2012). For example, it is well recognized that optimizing branch lengths to a high precision may not be a good use of time if the tree is poor. Thus in the algorithm of Guindon and Gascuel (2003), tree topology and branch lengths are adjusted simultaneously. Candidate trees are generated by local rearrangements of the current tree, for example, by using the NNI or SPR algorithms, and in calculating their likelihood scores, only the branch lengths affected by the local rearrangements are optimized, while branch lengths within subtrees unaffected by the rearrangements are not always optimized.

The likelihood algorithms have advanced greatly in the past decade, so that the ML method is now feasible for the analysis of large datasets with thousands of species/sequences (Guindon and Gascuel 2003; Stamatakis 2006; Zwickl 2006). Algorithms that take advantage of new computer hardware with multicore processors and graphical processing units (GPUs) (Suchard and Rambaut 2009; Zierke and Bakos 2010; Stamatakis et al. 2012) are pushing the boundary even further.

### 4.6.4 *Approximate likelihood method*

The computational burden of the likelihood method prompted the development of approximate methods. One idea is to use other methods to estimate branch lengths on a given tree rather than optimizing branch lengths by ML. For example, Adachi and Hasegawa (1996b) used least-squares estimates of branch lengths calculated from a pairwise distance matrix to calculate approximate likelihood scores. Similarly Rogers and Swofford (1998) used parsimony reconstruction of ancestral states to estimate approximate branch lengths. The approximate branch lengths provide good starting values for a proper likelihood optimization, but the authors suggested that they could also be used to calculate the approximate likelihood values for the tree without further optimization, leading to approximate likelihood methods of tree reconstruction.

Strimmer and von Haeseler (1996), and Schmidt et al. (2002) implemented an approximate likelihood algorithm for tree search called *quartet puzzling*. This uses ML to evaluate the three trees for every possible quartet of species. The full tree for all $s$ species is then constructed by a majority-rule consensus of those quartet trees. This method may not produce the ML tree but is fast. Ranwez and Gascuel (2002) implemented an algorithm that combines features of NJ and ML. It is based on triplets of taxa, and shares the divide-and-conquer strategy of the quartet approach.

With the development of modern likelihood programs such as RAXML, those approximate methods are no longer important. However, they may be useful for generating initial trees, or good candidates or proposals for ML or Bayesian tree search.

## 4.7 Model selection and robustness

### 4.7.1 *Likelihood ratio test applied to rbcL dataset*

We introduced the LRT in §1.4.3. Here we apply it to the data of the plastid *rbcL* genes from 12 plant species. The sequence alignment was kindly provided by Dr Vincent Savolainen. There are 1,428 nucleotide sites in the sequence. The tree topology is shown in Figure 4.16, which will be used to compare models. Table 4.3 shows the log likelihood values and parameter estimates under three sets of nucleotide substitution models. The first set includes JC69, K80, and HKY85, in which the same Markov model is applied to all sites in the sequence. The second set includes the '+$\Gamma_5$' models, which use the discrete gamma to accommodate the variable rates among sites, using five site classes (Yang 1994a). The third set includes the partition ('+C') models (Yang 1995a, 1996b). They assume that different codon positions have different substitution rates $(r_1, r_2, r_3)$, and, for K80 and HKY85, different substitution parameters as well. Parameters such as branch lengths in the tree, $\kappa$ in K80 and HKY85, and the relative rates for codon positions in the '+C' models, are

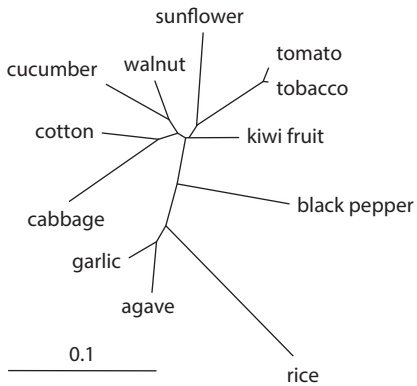


**Fig. 4.16** The ML tree for the plastid *rbcL* genes from 12 plant species, estimated under the HKY85+$\Gamma_5$ model. The branches are drawn in proportion to their estimated lengths.

**Table 4.3** Log likelihood values and MLEs of parameters under different models for the *rbcL* dataset

| Model | $p$ | $\ell$ | MLEs |
|---|---|---|---|
| JC69 | 21 | –6,262.01 | |
| K80 | 22 | –6,113.86 | $\hat{\kappa} = 3.561$ |
| HKY85 | 25 | –6,101.76 | $\hat{\kappa} = 3.620$ |
| JC69 + $\Gamma_5$ | 22 | –5,937.80 | $\hat{\alpha} = 0.182$ |
| K80 + $\Gamma_5$ | 23 | –5,775.40 | $\hat{\kappa} = 4.191, \hat{\alpha} = 0.175$ |
| HKY85 + $\Gamma_5$ | 26 | –5,764.26 | $\hat{\kappa} = 4.296, \hat{\alpha} = 0.175$ |
| JC69 + C | 23 | –5,922.76 | $r_1 : \hat{r}_2 : \hat{r}_3 = 1 : 0.556 : 5.405$ |
| K80 + C | 26 | –5,728.76 | $\hat{\kappa}_1 = 1.584, \hat{\kappa}_2 = 0.706, \hat{\kappa}_3 = 5.651,$ $r_1 : \hat{r}_2 : \hat{r}_3 = 1 : 0.556 : 5.611$ |
| HKY85 + C | 35 | –5,624.70 | $\hat{\kappa}_1 = 1.454, \hat{\kappa}_2 = 0.721, \hat{\kappa}_3 = 6.845$ $r_1 : \hat{r}_2 : \hat{r}_3 = 1 : 0.555 : 5.774$ |

*Note*: $p$ is the number of parameters in the model, including 21 branch lengths in the tree of Figure 4.16. The base frequency parameters under the HKY85 models are estimated using the observed frequencies (see Table 4.4).

**Table 4.4** Observed base compositions at the three codon positions for the 12 *rbcL* dataset (Figure 4.16)

| Position | $\pi_T$ | $\pi_C$ | $\pi_A$ | $\pi_G$ |
|---|---|---|---|---|
| 1 | 0.1829 | 0.1933 | 0.2359 | 0.3878 |
| 2 | 0.2659 | 0.2280 | 0.2998 | 0.2063 |
| 3 | 0.4116 | 0.1567 | 0.2906 | 0.1412 |
| All | 0.2867 | 0.1927 | 0.2754 | 0.2452 |

estimated by ML. The base frequency parameters in the HKY85 models are estimated by using the observed frequencies, averaged over the sequences (Table 4.4).

Here we consider three tests in detail. First, the JC69 and K80 models can be compared using an LRT, to test the null model $H_0$: $\kappa = 1$. Note that K80 will be reduced to JC69 when parameter $\kappa = 1$ is fixed. Thus JC69 is the null model, and includes $p_0 = 21$ branch lengths as parameters. The (optimized) log likelihood is $\ell_0 = -6{,}262.01$. K80 is the alternative model, with one extra parameter $\kappa$. The log likelihood is $\ell_1 = -6{,}113.86$. The test statistic is $2\Delta\ell = 2(\ell_1 - \ell_0) = 296.3$. This is much greater than the critical value $\chi^2_{1,\,1\%} = 6.63$, indicating that JC69 is rejected by a big margin. The transition and transversion rates are very different, as is clear from the MLE $\hat{\kappa} = 3.56$ under K80. Similarly comparison between K80 and HKY85 using an LRT with three degrees of freedom leads to rejection of the simpler K80 model.

A second test compares the null model JC69 against JC69+$\Gamma_5$, to test the hypothesis that different sites in the sequence evolve at the same rate. The one-rate model is a special case of the gamma model when the shape parameter $\alpha = \infty$ is fixed. The test statistic is $2\Delta\ell = 648.42$. In this test, the regularity conditions are not all satisfied, as the value $\infty$ is at the boundary of the parameter space in the alternative model. As a result, the null distribution is not $\chi^2_1$, but is a 1:1 mixture of the point mass at 0 and $\chi^2_1$ (Chernoff 1954; Self and Liang 1987; Whelan and Goldman 2000). In other words, one expects $2\Delta\ell$ to be 0 in half of the datasets and to be $\chi^2_1$ distributed in the other half when many datasets are simulated under the null model. The critical values are 2.71 at 5% and 5.41 at 1%, rather than 3.84 at 5% and 6.63 at 1% according to $\chi^2_1$. This null mixture distribution may be intuitively understood by considering the MLE of the parameter in the alternative model. If the true value is inside the parameter space, its estimate will have a normal distribution around the true value, and will be smaller than the true value half of the time and greater than the true value half of the time. If the true value is at the boundary, half of the time the estimate would be outside the space if there were no constraint; in such cases, the estimate will be forced to the true value and the log likelihood difference will be 0. Note that the use of $\chi^2_1$ makes the test too conservative; if the test is significant under $\chi^2_1$, it will be significant when the mixture distribution is used. For the *rbcL* dataset, the observed test statistic is huge, so that the null model is rejected whichever null distribution is used. The rates are highly variable among sites, as also indicated by the estimate of $\alpha$. Similar tests using the K80 and HKY85 models also suggest significant variation in rates among sites.

A third test compares JC69 and JC69+C. In the alternative model JC69+C, the three codon positions are assigned different relative rates $r_1\,(=1)\,, r_2$, and $r_3$, while the null model JC69 is equivalent to constraining $r_1 = r_2 = r_3$, reducing the number of parameters by 2. The test statistic is $2\Delta\ell = 678.50$, to be compared with $\chi^2_2$. The null model is clearly rejected, suggesting that the rates are very different at the three codon positions. The same conclusion is reached if the K80 or HKY95 model is used in the test.

The two most complex models in Table 4.3, HKY85+$\Gamma_5$ and HKY85+C, are the only models not rejected in such LRTs. (The two models themselves are not nested and a $\chi^2$ approximation to the LRT is not applicable.) This pattern is typical in the analysis of molecular datasets, especially large datasets; we seem to have no difficulty in rejecting an old simpler model whenever we develop a new model by adding a few extra parameters, sometimes even if the biological justification for the new parameters is dubious. The pattern appears to reflect the fact that most molecular datasets are very large and the LRT tends to favour parameter-rich models in large datasets.

### 4.7.2 *Test of goodness of fit and parametric bootstrap*

The LRT we discussed above compares two nested and closely related parametric models and addresses the question whether one model (the general model such as K80) fits the data significantly better than another model (the simpler model such as JC69). Even if K80 fits the data much better than JC69, neither may be adequate. A *test of general adequacy* of a model, also known as the *goodness of fit test*, can be constructed by noting that the data or the site pattern counts follow a multinomial distribution, with each possible site pattern being a category of the multinomial. For a dataset of $s$ sequences, there are $4^s$ possible site patterns, and thus $4^s$ categories, with $4^s - 1$ probability parameters. For example, the 64 site patterns for $s = 3$ sequences are TTT, TTC, TTA, TTG, TCT, TCC, ..., GGG. Under the multinomial model, the MLEs of the probabilities for the categories are the observed frequencies, so that the maximum log likelihood is

$$\ell_{\max} = \sum_{i=1}^{4^s} n_i \log\left\{\frac{n_i}{n}\right\}, \tag{4.39}$$

where $n$ is the total number of sites (columns) in the alignment and $n_i$ is the number of sites with the $i$th site pattern. Many of the site patterns may be missing in the data ($n_i = 0$) and they do not contribute to $\ell_{\max}$ (note that the convention is $0^0 = 1$ and $0 \log 0 = 0$).

Substitution models such as JC69, K80, HKY85+$\Gamma_5$, etc., assume identical and independent distribution among sites. They are all special cases of the general multinomial model. One might expect the $\chi^2$ distribution to apply when we compare the JC69 model, say, against this multinomial by an LRT, with $\mathrm{df} = (4^s - 1) - (2s - 3)$. However, this is not so because many of the possible site patterns (categories) have very low counts or are entirely missing. The rule of thumb is that there should be at least five expected counts in each category for the $\chi^2$ to be reliable.

When the $\chi^2$ distribution does not apply, we can use simulation to generate the null distribution. This approach is called a *parametric bootstrap* (e.g. Goldman 1993a; Yang et al. 1994). Here we illustrate it by using the plastid *rbcL* genes from the 12 plant species analysed above to test the goodness of fit of HKY85+$\Gamma_5$. We do not have a model for alignment gaps, so we remove sites with gaps, with 1,312 sites left.

First we use the two models to analyse the original real data. Under the general multinomial model, we have $\ell_{\max} = -4025.03$. We then fit HKY85+$\Gamma_5$ to estimate the 21 branch lengths and the parameters in the HKY85+$\Gamma_5$ model. The latter are $\hat{\kappa} = 4.40165$ and $\hat{\alpha} = 0.196551$. The base frequencies are estimated using the observed frequencies in the data: 0.29071 (T), 0.19372 (C), 0.27274 (A), and 0.24282 (G), although they could be obtained by ML. The log likelihood is $\ell_{\mathrm{HKYG}} = -5242.35$. (The difference from the value in Table 4.3 is due to the removal of sites with alignment gaps.) The test statistic is $\Delta\ell = \ell_{\max} - \ell_{\mathrm{HKYG}} = 1217.32$.

Next we simulate datasets to estimate the null distribution of the test statistic $\Delta\ell$. Use the MLEs of branch lengths for the tree of Figure 4.16 as well as the substitution parameters in

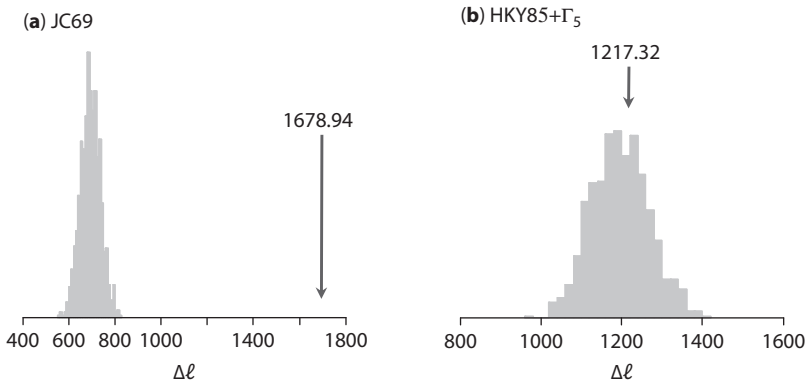


**Fig. 4.17** The null distribution estimated using parametric bootstrap for the test of goodness of fit of (**a**) the JC69 model and (**b**) the HKY85+$\Gamma_5$ model. The plastid *rbcL* genes from 12 plant species (Figure 4.16) are analysed.

the HKY85+$\Gamma_5$ model obtained from the original data to simulate 1,000 replicate datasets. Analyse each replicate dataset in the same way as the original data are analysed. In other words, each replicate dataset $i$ is analysed to calculate $\ell_{\max}$ and $\ell_{\text{HKYG}}$ and their difference $\Delta_i$. Collect the $\Delta_i$ values into a histogram (Figure 4.17b), which is an estimate of the null distribution. The observed statistic, $\Delta\ell = 1217.32$, falls well within the null distribution, so HKY85+$\Gamma_5$ provides an adequate fit to the data according to this test. The $p$-value is the proportion of the simulated $\Delta_i$ values that are greater than the observed $\Delta\ell$ value; this is $p = 38.7\%$.

A similar test of the general adequacy of the JC69 model leads to strong rejection of the model (Figure 4.17a) (Problem 4.5).

The parametric bootstrap is a general method for deriving the null distribution when it is unknown. It can be used to compare two models that are not nested or to compare two nested models when the dataset is small so that the asymptotic $\chi^2$ distribution may not be reliable (Goldman 1993a) (see Problem 2.5). It is nevertheless expensive as each replicate dataset has to be analysed in the same way as the original dataset.

### *4.7.3 *Diagnostic tests to detect model violations*

When the LRT of the goodness of fit of a model rejects the model, one may use diagnostic statistics (quantities that can be calculated using the observed data) to detect which aspects of the model assumptions are violated. Those statistics do not work well for data with alignment gaps or ambiguity sites, which should be removed before applying the tests. For any test statistic, one can use simulation (parametric bootstrap) to generate the distribution expected under the model, against which the observed value of the statistic can be compared. Here we mention a few such statistics (Tavaré 1986; Goldman 1993b; Rzhetsky and Nei 1995). We will consider nucleotide sequences although the test works for amino acid sequences as well.

The first diagnostic statistic is the number of distinct site patterns (Goldman 1993b). Similar statistics include the number of constant sites and the number of highly variable sites, appropriately defined. Early applications of such statistics highlight the importance of accommodating the among-site heterogeneity in the substitution model (Reeves 1992; Goldman 1993b; Yang et al. 1994, 1995c). When substitution rates vary among sites but the model ignores the rate variation, the data tend to show too few distinct

* indicates a more difficult or technical section.

site patterns, too many constant or highly variable patterns and too few intermediately variable patterns, relative to the model's expectations. Note that the expectation under the model, or the null distribution of the test statistic under the model can be generated by parametric bootstrap (simulation). One uses the parameter estimates under the model to generate 1,000 replicate datasets and then to calculate the statistic in each to produce a histogram. One can then decide whether the observed number of distinct site patterns is too small or too large relative to the model expectation (Goldman 1993b). Alternatively, one may calculate the mean and variance of the statistic under the model directly if the dataset is very small. As discussed above, the sequence data can be summarized as the counts from a multinomial distribution with $4^s$ cells for $s$ sequences, corresponding to the $4^s$ site patterns. The statistic is the number of non-empty cells in the multinomial. If $s$ is very small, one can calculate the probabilities for all site patterns using the MLEs of parameters under the model, and then calculate the mean and variance of the statistic expected under the model. However, it is far simpler to generate the null distribution by simulation.

If the substitution process is stationary, one should expect the different sequences to have the same nucleotide (or amino acid) frequencies, apart from chance fluctuations. Thus one can construct an $s \times 4$ sequence $\times$ nucleotide contingency table and test whether the nucleotide frequencies are homogeneous across sequences (e.g. Rzhetsky and Nei 1995). When the substitution process is reversible and stationary, the probability of observing a site with nucleotides $i$ and $j$ in two sequences equals the probability for a site with $j$ and $i$. Such expected symmetry can be tested by using the $4 \times 4$ contingency table of site pattern counts for two sequences (Tavaré 1986). Let $N_{ij}$ be the number of sites with nucleotide $i$ in sequence 1 and $j$ in sequence 2. Then

$$X^2 = \sum_{i<j} \frac{(N_{ij} - N_{ji})^2}{N_{ij} + N_{ji}} \tag{4.40}$$

is compared with the asymptotic $\chi^2$ distribution with six degrees of freedom. Jermiin et al. (2008) have explored a few variants of this matched-pairs test. Such tests are in general expected to be very powerful, and are able to detect small violations that may not have a noticeable impact on phylogenetic analysis.

### 4.7.4 *Akaike information criterion (AIC and AIC$_c$)*

The LRT is applicable for comparing two nested models. Although Cox (1961, 1962; see also Atkinson 1970; Lindsey 1974a, 1974b; Sawyer 1984) discussed the use of LRT to compare non-nested models, the idea has not been used much in practical data analysis (but see Goldman 1993a). In the Cox test one of the two non-nested models is designated as the null hypothesis and the other as the alternative hypothesis. Simulation (parametric bootstrap) is often necessary to derive the null distribution of the LRT statistic. With non-nested models, it is often arbitrary which one should be the null hypothesis. To avoid this arbitrariness, two tests are often conducted instead of one, with each of the two models being used as the null hypothesis. The tests may then fail to reject either model, in which case it is unclear how to make inferences about the quantities of interest, especially if the two models lead to different conclusions. In other datasets (especially large ones) the tests may reject both models, in which case we do not have a useful comparison of the two models apart from knowing that neither fits the data well.

The Akaike information criterion (AIC, Akaike 1974) can be used to compare models that are not necessarily nested. The AIC score is calculated for each model, defined as

$$\text{AIC} = -2\ell + 2p, \tag{4.41}$$

where $\ell = \ell(\hat{\theta})$ is the optimum log likelihood under the model, and $p$ is the number of parameters. Models with small AICs are preferred. According to this criterion, an extra parameter is worthwhile if it improves the log likelihood by more than one unit.

The AIC is perceived not to penalize parameter-rich models enough. A correction is thus introduced, which incorporates the data size in the criterion (Sugiura 1978; Hurvich and Tsai 1989)

$$\text{AIC}_c = -2\ell + \frac{2np}{n-p-1} = \text{AIC} + \frac{2p(p+1)}{n-p-1}. \tag{4.42}$$

### 4.7.5 *Bayesian information criterion*

In large datasets, both LRT and AIC are known to favour complex parameter-rich models and to reject simpler models too often (Schwarz 1978). The Bayesian information criterion (BIC) is based on a Bayesian argument and penalizes parameter-rich models more severely. It is defined as

$$\text{BIC} = -2\ell + p\log(n), \tag{4.43}$$

where $n$ is the sample size (sequence length) (Schwarz 1978). Again models with small BIC scores are preferred.

Qualitatively, LRT, AIC, and BIC are all mathematical formulations of the *parsimony principle* of model building. Extra parameters are deemed necessary only if they bring about significant or considerable improvements to the fit of the model to data, and otherwise simpler models with fewer parameters are preferred. However, in large datasets, these criteria can differ markedly. For example, if the sample size $n > 8$, BIC penalizes parameter-rich models far more severely than does AIC.

Model selection is an active research area in statistics. Posada and Buckley (2004) provided a nice overview of methods and criteria for model selection in molecular phylogenetics. For automatic model selection, Posada and Crandall (1998; Posada 2008) developed MODELTEST. Well-known substitution models are compared hierarchically using the LRT, AIC, or BIC. The program enables the investigator to avoid making thoughtful decisions concerning the model to be used in phylogeny reconstruction. However, mechanical application of MODELTEST has led to widespread use of the most complex models, such as the pathological 'I + Γ' models, in real data analysis.

*Example 4.5. Model selection for the ape mitochondrial protein data.* We use the model selection criteria LRT, AIC, $\text{AIC}_c$, and BIC to compare a few models applied to the dataset analysed in §4.2.4. The ML tree of Figure 4.5 is assumed. The three empirical amino acid substitution models DAYHOFF (Dayhoff et al. 1978), JTT (Jones et al. 1992), and MTMAM (Yang et al. 1998) are fitted to the data, with either one rate for all sites or gamma rates for sites. In the discrete gamma model, five rate categories are used; the estimates of the shape parameter $\alpha$ range from 0.30 to 0.33 among the three models. The results are shown in Table 4.5. The LRT can be used to compare nested models only, so each empirical model (e.g. DAYHOFF) is compared with its gamma counterpart (e.g. DAYHOFF +$\Gamma_5$). The LRT statistics are very large, so there is no doubt that the substitution rates are highly variable among sites, whether $\chi^2_1$ or the 50:50 mixture of 0 and $\chi^2_1$ is used for the test. The AIC, $\text{AIC}_c$, and BIC scores can be used to compare non-nested models, such as the three empirical models. As they involve the same number of parameters, the ranking using AIC or BIC is the same as using the log likelihood. MTMAM fits the data better than the other two

**Table 4.5** Comparison of models for the mitochondrial protein sequences from the apes

| Model | $p$ | $\ell$ | LRT | AIC | $AIC_c$ | BIC |
|---|---|---|---|---|---|---|
| DAYHOFF | 11 | −15,766.72 | | 31,555.44 | 31,555.52 | 31,622.66 |
| JTT | 11 | −15,332.90 | | 30,687.80 | 30,687.88 | 30,755.02 |
| MTMAM | 11 | −14,558.59 | | 29,139.18 | 29,139.26 | 29,206.40 |
| DAYHOFF + $\Gamma_5$ | 12 | −15,618.32 | 296.80 | 31,260.64 | 31,260.73 | 31,333.97 |
| JTT + $\Gamma_5$ | 12 | −15,192.69 | 280.42 | 30,409.38 | 30,409.47 | 30,482.71 |
| MTMAM + $\Gamma_5$ | 12 | −14,411.90 | 293.38 | 28,847.80 | 28,847.89 | 28,921.13 |

*Note*: $p$ is the number of parameters in the model. The sample size is $n = 3,331$ amino acid sites for the AICc and BIC calculations. The LRT column shows the test statistic $2\Delta\ell$ for comparing each empirical model with the corresponding gamma model.

models, which is expected since the data are mitochondrial proteins while DAYHOFF and JTT were derived from nuclear proteins. The best model for the data according to all three criteria is MTMAM +$\Gamma_5$. □

### 4.7.6 *Model adequacy and robustness*

*All models are wrong but some are useful*. (George Box, 1979)

Models are used for different purposes. In some cases, the model is an interesting biological hypothesis we wish to test, so that the model (hypothesis) itself is our focus. For example, the molecular clock (rate constancy over time) is an interesting hypothesis predicted by certain theories of molecular evolution, and it can be examined by using an LRT to compare a clock model and a nonclock model. In other cases, the model, or at least some aspects of the model assumptions, is not our main interest, but has to be dealt with in the analysis. For example, in testing the molecular clock, we need a Markov model of nucleotide substitution (JC69 or HKY85+Γ). We are not interested in the substitution model but we may be concerned about its impact on our test of the molecular clock. In phylogeny reconstruction, our interest is in the tree, but we have to assume an evolutionary model to describe the mechanism by which the data are generated. The model is then a nuisance, but its impact on our analysis cannot be ignored. Some writers distinguish a *hypothesis* from a *model*, and use the term *hypothesis* to refer to the first case (where the model is our focus) and *model* to refer to the second case (where the model is a nuisance). While we do not make such a distinction here, we should bear in mind what the model is used for. Model selection discussed in this section refers to selection of models that are not the focus of our analysis.

It should be stressed that a model's fit to data and its impact on inference are two different things. Often model robustness is even more important than model adequacy. It is neither possible nor necessary for a model to match the biological reality in every detail. The aim of model selection is not to find the 'true model' but to find a model with sufficient parameters to capture the key features of the data (Steel 2005). What features are important will depend on the question being asked, and one has to use knowledge of the subject matter to make the judgement. Structural biologists tend to emphasize the uniqueness of every residue in the protein. Similarly one has every reason to believe that every species is unique. However, by no means should one use one separate parameter for every site and every branch in formulating a statistical model to describe the evolution of the protein sequence. Such a model, saturated with parameters, is not workable.

One should appreciate the power of the i.i.d. models, which assume that the sites in the sequence are independent and identically distributed. A common misconception is that i.i.d. models assume that every site evolves at the same rate and follows the same pattern. It should be noted that almost all models implemented in molecular phylogenetics, such as models of variable rates among sites (Yang 1993, 1994a), models of variable selective pressures among codons (Nielsen and Yang 1998), and the covarion models that allow the rate to vary both among sites and among lineages (Tuffley and Steel 1998; Galtier 2001; Guindon et al. 2004) are i.i.d. models. The i.i.d. assumption is a statistical device which is useful for reducing the number of parameters.

Some features of the process of sequence evolution are both important to the fit of the model to the data and critical to our inference. They should be incorporated in the model. Variable rates among sites appear to be such a factor for phylogeny reconstruction or estimation of branch lengths (Tateno et al. 1994; Huelsenbeck 1995a; Gaut and Lewis 1995; Sullivan et al. 1995; Yang 1996c). Some factors may be important to the model's fit, as judged by the likelihood, but may have little impact on the analysis. For example, adding the transition/transversion rate ratio $\kappa$ to the JC69 model almost always leads to a huge improvement to the log likelihood, but often has minimal effect on estimation of branch lengths. The difference between HKY85 and GTR (REV) is even less important, even though HKY85 is rejected in most datasets when compared against GTR. The most troublesome factors are those that have little impact on the fit of the model but a huge impact on our inference. For example, in estimation of species divergence times under local molecular clock models, different models for lineage rates appear to fit the data almost equally well, judged by their log likelihood values, but they can produce very different time estimates (see Chapter 10). Such factors have to be carefully assessed even if the statistical test does not indicate their importance.

For phylogeny reconstruction, a number of computer simulations have been conducted to examine the robustness of different methods to violations of model assumptions. Such studies have in general found that model-based methods such as ML are quite robust to the underlying substitution model (e.g. Hasegawa et al. 1991; Gaut and Lewis 1995). However, the importance of model assumptions appears to be dominated by the shape of the tree reflected in the relative branch lengths, which determines the overall level of difficulty of tree reconstruction. 'Easy' trees, with long internal branches or with long external branches clustered together, are successfully reconstructed by all methods and models; indeed wrong simplistic models tend to show even better performance than the more complex true model (we will discuss such counterintuitive results later in §5.2.3). 'Hard' trees, with short internal branches and with long external branches spread over different parts of the tree, are difficult to reconstruct by all methods. For such trees, simplistic models may not even be statistically consistent, and use of complex and realistic models is critical.

## 4.8 Problems

4.1 Calculate the probabilities of site patterns *xxx*, *xxy*, *yxx*, and *xyx* as a function of the branch lengths $t_{10}$ and $t_{11}$ in the tree $\tau_1$ of Figure 4.14. Assume the symmetrical substitution model for binary characters (equation (1.79)).

4.2* Try to estimate the single branch length under the JC69 model for the star tree of three sequences under the molecular clock (see Saitou 1988; Yang 1994c, 2000a, for

* indicates a more difficult or technical problem.

discussions of likelihood tree reconstruction under this model). The tree is shown in Figure 4.10, where $t$ is the only parameter to be estimated. Note that there are only three site patterns, with one, two, or three distinct nucleotides, respectively. The data are the observed numbers of sites with such patterns: $n_0$, $n_1$, and $n_2$, with the sum to be $n$. Let the proportions be $f_i = n_i/n$. The log likelihood is $\ell = n\sum_{i=0}^{2} f_i \log(p_i)$, with $p_i$ to be the probability of observing site pattern $i$. Derive $p_i$ by using the transition probabilities under the JC69 model, given in equation (1.4). You can calculate $p_0$ = Pr(TTT), $p_1$ = Pr(TTC), and $p_2$ = Pr(TCA). Then set $d\ell/dt = 0$. Show that the MLE for the transformed parameter $z = e^{-4t/3}$ is a solution to the following quintic equation:

$$36z^5 + 12(6 - 3f_0 - f_1)z^4 + (45 - 54f_0 - 42f_1)z^3 + (33 - 60f_0 - 36f_1)z^2 + (3 - 30f_0 - 2f_1)z + (3 - 12f_0 - 4f_1) \equiv 0. \tag{4.44}$$

4.3 *Long-branch attraction for parsimony.* Calculate the probabilities of sites with data *xxyy*, *xyyx*, and *xyxy* in four species for the unrooted tree of Figure 4.18, using two branch lengths $p$ and $q$ under a symmetrical substitution model for binary characters (equation (1.79)). Here it is more convenient to define the branch length as the proportion of different sites at the two ends of the branch. Show that Pr(*xxyy*) < Pr(*xyxy*) if and only if $q(1-q) < p^2$. With such branch lengths, parsimony for tree reconstruction is inconsistent (Felsenstein 1978a).

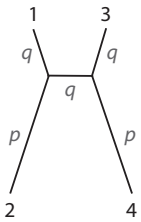


**Fig. 4.18** A tree of four species with two branch lengths $p$ and $q$, defined as the probability that any site is different at the two ends of the branch. For a binary character, this probability is $p = (1 - e^{-2t})/2$, where $t$ is the expected number of character changes per site (see equation (1.79)).

4.4 Bias in ancestral state reconstruction. Calculate the posterior probabilities for T, C, A, and G at the root of the tree of Figure 4.10 when the observed data at the site is $x_1x_2x_3$ = AAG. Assume the F81 substitution model, with base frequency parameters $\pi_T$ = 0.2263, $\pi_C$ = 0.3282, $\pi_A$ = 0.3393, and $\pi_G$ = 0.1062. Suppose that each branch length is 0.2, and the transition probability matrix is given in equation (4.26). Hint: Use equation (4.23).

4.5 Use the plastid *rbcL* genes from 12 plant species to test the goodness of fit of the JC69 model. Follow the example of §4.7.2. Use BASEML in the PAML package to analyse the original data to generate branch lengths and calculate $\Delta\ell = \ell_{\max} - \ell_{JC}$. Use those branch lengths to simulate 1,000 datasets using the program SEQ-GEN or EVOLVER. Then use a likelihood program (such as BASEML) to analyse the 1,000 replicate datasets to calculate $\Delta_i$ to construct a histogram. Your results should be similar to Figure 4.17a.

4.6 *Phylogenetic reconstruction using ML*. Use your own data or find a small dataset of 10–50 species from the literature to infer the ML phylogeny under various substitution models, such as JC69, K80, HKY85, GTR, and the gamma variants JC69+$\Gamma_5$, K80+$\Gamma_5$, HKY85+$\Gamma_5$, and GTR +$\Gamma_5$. You can use PHYML to run tree search under those models.