9.5 Simulating molecular evolution

9.5.1 Simulating sequences on a fixed tree

Here we consider generation of a nucleotide sequence alignment when the tree topology and branch lengths are given. The basic model assumes the same substitution process at all sites and along all branches, but we will also consider more complex models in which the evolutionary process may vary across sites or branches. We consider nucleotide models; amino acid or codon sequences can be generated using the same principles. Several approaches can be used and produce equivalent results.

9.5.1.1 Method 1. Sampling sites from the multinomial distribution of site patterns

If the substitution model assumes independent evolution at different sites in the sequence and all sites evolve according to the same model, data at different sites will have independent and identical distributions; they are said to be *i.i.d*. The sequence data set will then follow a multinomial distribution, with every site to be a sample point, and every site pattern to be a category (cell) of the multinomial. For a tree of s species, there are 4^s, 20^s, or 64^s possible site patterns for nucleotide, amino acid, or codon sequences, respectively. Calculation of the probability for every site pattern is explained in Section 4.2, which describes the calculation of the likelihood function under the model. A sequence alignment can thus be generated by sampling from this multinomial distribution. The result will be the numbers of sites having the site patterns, many of which may be zero if the sequence is short. If a pattern, say TTTC (for four species), is observed 50 times, one simply writes out 50 sites with the same data TTTC, either with or without randomizing the sites. Most phylogeny programs, especially those for likelihood and Bayesian calculations, collapse sites into patterns to save computation, since the probabilities of observing sites with the same data are identical. As this simulation method generates counts of site patterns, those counts should in theory be directly usable by phylogeny programs.

The multinomial sampling approach is not feasible for large trees because the number of categories becomes too large. However, it is very efficient for small trees with only four or five species, especially when combined with an efficient algorithm for sampling from the multinomial, such as the alias method.

9.5.1.2 Method 2. Evolving sequences along the tree

This approach 'evolves' sequences along the given tree, and is the algorithm used in programs such as Seq-Gen (Rambaut and Grassly 1997) and EVOLVER (Yang 1997a). First, we generate a sequence for the root of the tree, by sampling nucleotides according to their equilibrium distribution under the model: π_T, π_C, π_A, π_G. Every nucleotide is sampled independently from the discrete distribution (π_T, π_C, π_A, π_G). If the base frequencies are all equal, one can use the algorithm for discrete uniform distributions, which is more efficient. The sequence for the root is then allowed to

evolve to produce sequences at the daughter nodes of the root. The procedure is repeated for every branch on the tree, generating the sequence at a node only after the sequence at its mother node has been generated. Sequences at the tips of the tree constitute the data, while sequences for ancestral nodes are discarded.

To simulate the evolution of a sequence along a branch of length t, calculate the transition probability matrix $P(t) = \{p_{ij}(t)\}$ (see Sections 1.2 and 1.5), and then simulate nucleotide substitutions at every site independently. For example, if a site is occupied by C in the source sequence, the nucleotide in the target sequence will be a random draw from the discrete distribution of T, C, A, G, with probabilities $p_{CT}(t)$, $p_{CC}(t)$, $p_{CA}(t)$, $p_{CG}(t)$, respectively. This process is repeated to generate all sites in the target sequence. As the transition probabilities apply to all sites in the sequence for the branch, one has to calculate them only once for all sites for that branch.

9.5.1.3 Method 3. simulating the waiting times of a Markov chain

This is a variation to method 2. One generates a sequence for the root, and then simulates the evolution of any site along any branch as follows. Suppose the branch length is t, and the rate matrix of the Markov chain is $Q = \{q_{ij}\}$. Let $q_i = -q_{ii} = -\sum_{j \neq i} q_{ij}$ be the substitution rate of nucleotide i. Suppose the site is currently occupied by nucleotide i. Then the waiting time until the next substitution event has an exponential distribution with mean $1/q_i$. We draw a random waiting time s from the exponential distribution. If $s > t$, no substitution occurs before the end of the branch so that the target sequence has nucleotide i at the site as well. Otherwise a substitution occurs and we decide which nucleotide the site changes into. Given that the site with nucleotide i has experienced a change, the probability that it changes into nucleotide j is q_{ij}/q_i, and we can sample j from this discrete distribution. The remaining time for the branch becomes $t - s$. We then draw a random waiting time from the exponential distribution with mean $1/q_j$. The process is repeated until the time for the branch is exhausted.

This simulation procedure is based on the following characterization of a continuous-time Markov chain (Fig. 9.2). The waiting time until the next transition (change) is exponential with mean $1/q_i$. If we ignore the waiting times between transitions, the sequence of states visited by the process constitutes a discrete-time Markov

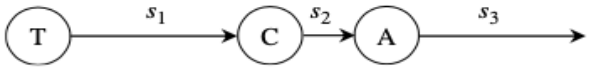

Fig. 9.2 Characterization of the Markov process as exponential waiting times and a jump chain. The waiting times until the next substitution s_1, s_2, and s_3 are independent random variables from the exponential distributions with means $1/q_T$, $1/q_C$, and $1/q_A$, respectively, where q_T, q_C, and q_A are the substitution rates of nucleotides T, C, and A, respectively.

chain, which is called the *jump chain*. The transition matrix of the jump chain is given as

$$M = \begin{bmatrix} 0 & \frac{q_{TC}}{q_T} & \frac{q_{TA}}{q_T} & \frac{q_{TG}}{q_T} \\ \frac{q_{CT}}{q_C} & 0 & \frac{q_{CA}}{q_C} & \frac{q_{CG}}{q_C} \\ \frac{q_{AT}}{q_A} & \frac{q_{AC}}{q_A} & 0 & \frac{q_{AG}}{q_A} \\ \frac{q_{GT}}{q_G} & \frac{q_{GC}}{q_G} & \frac{q_{GA}}{q_G} & 0 \end{bmatrix}. \quad (9.7)$$

Note that every row sums to 1.

The algorithm of simulating exponential waiting times and the jump chain may be applied to the whole sequence instead of one site. The total rate of substitution is the sum of the rates across sites, and the waiting time until a substitution occurs at any site in the whole sequence has an exponential distribution with the mean equal to the reciprocal of the total rate. If a substitution occurs, it is assigned to sites with probabilities proportional to the rates at the sites.

An advantage of this simulation procedure is that it does not require calculation of the transition-probability matrix $P(t)$ over branch length t, as both the waiting times and the transition matrix for the jump chain are fully specified by the instantaneous rates. As a result, this procedure can be adapted to simulate more complex sequence changes such as insertions and deletions. One simply calculates the total rate of all events (including substitutions, insertions, and deletions) for the whole sequence, and simulates the exponential waiting time until the next event. If an event occurs before the end of the branch, one assigns the event to a site and to an event type (a substitution, insertion, or deletion) with probabilities in proportion to the rates of those events at the sites.

9.5.1.4 Simulation under more complex models

The methods discussed above can be modified to simulate under more complex models, for example, to allow different substitution parameters among branches (such as different transition/transversion rate ratio κ, different base frequencies, or different ω ratios). The multinomial sampling approach (method 1) applies as long as the site pattern probabilities are calculated correctly under the model. The approaches of evolving sequences along branches on the tree, either by calculating the transition probabilities (method 2) or by simulating the waiting times and the jump chain (method 3) are also straightforward; one simply use the appropriate model and parameters for the branch when simulating the evolutionary process along that branch.

One may also simulate under models that allow heterogeneity among sites. We will consider as an example variable substitution rates but the approaches apply to other kinds of among-site heterogeneity. There are two kinds of models that incorporate rate variation among sites (see Subsections 4.3.1 and 4.3.2). The first is the so-called

fixed-sites model, under which every site in the sequence belongs to a predetermined site partition. For example, one may simulate five genes evolving on the same tree but at different rates $r_1, r_2, \ldots, r_5$. The transition probability matrix for gene k is $p_{ij}(tr_k)$. Sites in different genes do not have the same distribution, but within each gene, the sites are *i.i.d*. Thus one can use either of the three methods discussed above to simulate the data for each gene separately and then merge them into one data set.

A second kind of heterogeneous-sites model are the so-called *random-sites models*. Examples include the gamma models of variable rates for sites (Yang 1993, 1994a), the codon models of variable ω ratios among sites (Nielsen and Yang 1998; Yang *et al.* 2000), and the covarion-like models (Galtier 2001; Huelsenbeck 2002; Guindon *et al.* 2004). The rates (or some other features of the substitution process) are assumed to be random variables drawn from a common statistical distribution, and we do not know a priori which sites have high rates and which have low rates. Data at different sites are *i.i.d*. The approach of multinomial sampling can be used directly, although the site pattern probabilities have to be calculated under the heterogeneous-sites model. One may also sample the rate for each site and apply the method of simulating evolution along branches. If a continuous distribution is used, one should in theory calculate the transition probability matrix $P(t)$ for every site and every branch. If a few site classes are assumed (as in the discrete-gamma model), one may sample rates for sites first, and then simulate data for the different rate classes separately, perhaps followed by a randomization of the sites. Note that under the random-sites model, the number of sites in any site class varies among simulated replicates, whereas in the fixed-sites model, the number is fixed.

9.5.2 Generating random trees

Several models can be used to generate random trees with branch lengths: the standard coalescent model, the Yule branching model, and the birth–death process model either with or without species sampling. All those models assign equal probabilities to all labelled histories. (A labelled history is a rooted tree with the internal nodes ranked according to their ages.) One may generate a random genealogical or phylogenetic tree by starting with the tips of the tree, and joining nodes at random until there is one lineage left.

The ages of the nodes are independent of the labelled history and can be attached to the tree afterwards. Under the coalescent model, the waiting time until the next coalescent event has an exponential distribution (see equation 5.41). Under the birth–death process model, the node ages on the labelled history are order statistics from a kernel density (see equation 7.24) and can be simulated easily (Yang and Rannala 1997). Trees generated in this way have branch lengths conforming to a molecular clock. One may modify the substitution rates to produce trees in which the clock is violated.

One may also generate random rooted or unrooted trees without assuming any biological model, by sampling at random from all possible trees. Branch lengths may also be sampled from arbitrary distributions such as the exponential or the gamma.