Markov chain Monte Carlo for the Bayesian analysis of evolutionary trees from aligned molecular sequences

Michael A. Newton ¹  Bob Mau
Department of Statistics  Department of Genetics
University of Wisconsin–Madison  University of Wisconsin–Madison
1210 West Dayton St.  445 Henry Mall
Madison WI 53706  Madison WI 53792
Phone: 608 263 0357  Phone: 608 262 2534
Email: newton@stat.wisc.edu  Email: robertm@genetics.wisc.edu
http://www.stat.wisc.edu/  http://www.stat.wisc.edu/

Bret Larget
Department Mathematics and
Computer Science
Duquesne University
440 College Hall
Pittsburgh PA 15282
Phone: 412-396-6469
Email: largert@mathcs.duq.edu
http://www.mathcs.duq.edu/

For the Proceedings of the AMS-IMS-SIAM Joint Summer Research Conference on

¹To whom correspondence should be addressed.
Markov chain Monte Carlo for the Bayesian analysis of evolutionary trees from aligned molecular sequences

Abstract

We show how to quantify the uncertainty in a phylogenetic tree inferred from molecular sequence information. Given a stochastic model of evolution, the Bayesian solution is simply to form a posterior probability distribution over the space of phylogenies. All inferences are derived from this posterior, including tree reconstructions, credible sets of good trees, and conclusions about monophyletic groups, for example. The challenging part is to approximate the posterior, and we do this by constructing a Markov chain having the posterior as its invariant distribution, following the approach of Mau, Newton, and Larget (1996). Our Markov chain Monte Carlo algorithm is based on small but global changes in the phylogeny, and exhibits good mixing properties empirically. We illustrate the methodology on DNA encoding mitochondrial cytochrome oxidase 1 gathered by Hafner et al. (1994) for a set of parasites and their hosts.
1 Introduction

Stochastic models have long been considered useful for describing variation in the molecular sequences of extant populations (e.g., Jukes and Cantor, 1969; Felsenstein, 1973; Kimura, 1980). Parameters in such models include the phylogeny, which encodes the pattern of evolutionary relationships among populations, and substitution rates, which describe how molecules change over time within populations. It seems quite natural to infer these parameters using the induced likelihood function in some way, but such inference has been difficult in practice because computations can be prohibitively expensive. Owing to the Markovian nature of the standard models, evaluation of the likelihood function follows straightforward recursive equations, and so evaluation is not the difficult part. The difficulty arises with optimization, since the likelihood resides over a complicated parameter space, and seems to admit no simple representation (Felsenstein, 1981, 1983; Goldman, 1990; Yang, Goldman, Friday, 1995). Nevertheless, computer code is available for approximate maximum likelihood calculation (Olson, et al., 1994; Felsenstein, 1995; Swofford, 1996).

Beyond estimation, practitioners have demanded some way to assess uncertainty in aspects of the estimated phylogeny, just as error bars accompany simpler kinds of point estimates. A standard and appealingly simple calculation is to apply Efron’s bootstrap (Felsenstein, 1985), and although this method may accurately approximate sampling distributions, its role for statistical inference about the phylogeny has been a matter of some debate (e.g., Felsenstein and Kishino, 1993; Newton, 1996; Efron, Halloran, and Holmes, 1996; Chernoff, 1997). Of course, bootstrapping a complicated estimator serves to compound the computational problem. Thus, bootstrapping a full-blown maximum likelihood estimator is practically impossible with today’s implementations. A common practice is to bootstrap a much simpler estimator.

An alternative, model-based, assessment of uncertainty was postulated some time ago by J.F.C. Kingman, in the discussion of Joe Felsenstein’s 1983 paper on statistical issues in evolutionary biology:

In view of the difficulties of the maximum likelihood approach, it seems worth asking what a Bayesian analysis would look like. The author has shown us how to write down the likelihood function, and this has only to be multiplied by a suitable prior. ... The result is a set of posterior probabilities for collections of
possible phylogenies, not just a single estimate, and it may well be that there
are tractable approximations of the probabilities of some compound events.
Has this approach been explored?

Until very recently, this Bayesian approach had not been explored. Sinsheimer,
Lake and Little (1996) developed exact Bayesian calculations for the four-species
problem. Several groups have been pursuing Markov chain Monte Carlo (MCMC)
approximations. Mau and Newton (1997) described an MCMC method for models
satisfying a molecular clock, and presented calculations for binary, restriction-sites
data. Mau, Newton, and Larget (1996) have extended these calculations to problems
with more taxa and nucleotide sequence data. Yang and Rannala (1997), and Li,
Pearl and Doss (1996) have developed different Markov chain Monte Carlo strategies
for the same general problem. In fact, the MCMC method of Kuhner, Yamato, and
Felsenstein (1995) can be modified to produce approximate Bayesian phylogenetic
inferences, even though their model considers within population sampling of
sequences. The purpose of the present article is to review the Mau, Newton, Larget
approach and to illustrate the calculations in an example.

Of course Bayesian analysis provides more than assessments of uncertainty about
phylogenies, the focus of this work. The array of inference problems presented in
Huelsenbeck and Rannala (1997), for example, all may be approached from a Bayesian
perspective. We anticipate that future research will clarify the role of Bayesian
analysis for evolutionary biology, but first some essential computational problems
must be addressed.

2 Phylogeny

A phylogeny or evolutionary tree, $\tau$, admits various representations. For the
present discussion, it will be convenient to treat $\tau$ as a pair $(t, \pi)$ where
$t = (t_1, \ldots, t_{s-1}) \in \mathbb{R}^{s-1}$ is a vector of positive speciation times, $s$ is the number of
species under consideration, and $\pi$ is a permutation of $\{1, 2, \ldots, s\}$. The path of
evolution corresponding to $\tau$ can be envisioned by processing $t$ and $\pi$ in a manner as
illustrated in Figure 1. Species labels $\{1, 2, \ldots, s\}$ are arranged along the horizontal
in the order determined by $\pi$, forming $s - 1$ gaps between species. From the center
of the \( i \)th gap, we drop a vertical line of length \( t_i \), and call the lower endpoint a node. We draw a tree by moving downwards from the species labels, connecting each encountered node to the closest parentless nodes above to the left and right. Continuing down the page, we reach another node, and connect labels again. If ever one or another of the species labels defining the gap has already been connected, the drawing rule is to connect this node to the earliest node linking said species label. Eventually, all nodes are connected, and a tree results. The node corresponding to the largest \( t_i \) is called the root.

Several remarks are in order regarding this construction. The horizontal axis serves only to organize the information, and has no intrinsic scale. On the other hand, the vertical axis records time into the past, or an amount of evolution. As drawn, our trees are rooted and have contemporaneous tips, and will be considered parameters of models which satisfy the molecular clock hypothesis. In work to extend our methods to models where evolutionary rates vary among branches of the tree, a somewhat different tree representation is more appropriate than the one just described. It is noteworthy, however, that the essential elements of the Monte Carlo algorithm to be described carry over readily to this more general case. Note that the drawing algorithm has not been defined when two times are equal, and so we omit this case (i.e., assume \( t_i \neq t_j \) for all \( i, j \)). This implies that the trees are binary.

The phylogeny \( \tau = (t, \pi) \) records the path of evolution from a single ancestral population to the present array of \( s \) populations under study. Any point on the tree drawn according to the rules above thus represents a population at some time in the past. Evidently, different \((t, \pi)\) pairs determine the same path of evolution, noting again that the horizontal axis in Figure 1 has no scale. For example, rearranging species 3 and 9 does not change the path of evolution. A given unordered set of times \( \{t_1, t_2, \ldots, t_{s-1}\} \) induces the same path of evolution in \( 2^{s-1} \) different ways, obtained by rotating the graph by 180 degrees above any of the \( s - 1 \) internal nodes. Strictly speaking, therefore, the phylogeny \( \tau \) is an equivalence class containing \( 2^{s-1} \) different versions \((t, \pi)\). The third panel in Figure 1 shows a second version of the preceding tree.

The representation of a phylogeny as \( 2^{s-1} \) points \((t, \pi)\) is particularly conducive to Markov chain Monte Carlo (MCMC), as we discuss in Section 4. A key feature is
that the tree is part of a continuum, and the branching pattern of the tree is induced by the permutation \( \pi \) and the relative ordering of the times \( t_1, \ldots, t_{s-1} \). Indeed, the branching pattern inherent in \( \tau \) may be of interest, but we do not represent that pattern directly, choosing instead to work with more elementary objects which combine to produce the pattern. This somewhat indirect approach leads to very simple MCMC steps (Section 4) and may be associated with the efficiency of the algorithm.

Different summaries of \( \tau \) may be of interest to the biologist. The labeled history describes the branching pattern of \( \tau \) obtained by ignoring the magnitude of the times \( t_1, t_2, \ldots, t_{s-1} \), but respecting their ordering. Incidentally, counting labeled histories is quite simple given our construction, as there are \( s! \) ways to arrange the \( s \) species labels, \( (s-1)! \) orderings of the times, but we have overcounted by \( 2^{s-1} \), leaving \( s!(s-1)!/2^{s-1} \) distinct labeled histories. The tree topology corresponding to \( \tau \) is another property of its branching pattern, where we record only the sequence of connections, but ignore details of their time ordering. The tree topology may be characterized by nested parentheses, such as

\[
\text{top}(\tau) = (((1, (3, 9)), ((2, (7, 8)), ((4, 6), 10))), 5))
\]

for the phylogeny shown in Figure 1. Here we have taken the convention that when two groups of organisms are merged, we place on the left that group containing the smallest species label.

### 3 Modeling Substitutions

The probability of data given a tree is derived from a model of DNA evolution along the branches of \( \tau \), and many such models have been studied. We follow convention here and take the same general assumptions as those characterizing many standard models. That is, we consider the DNA sequences to be aligned, and we treat evolution of different sites as stochastically independent. Further, we allow base substitutions to occur at times of branch-specific Poisson processes, with independent evolution among branches. These restrictions still leave us some flexibility in the modeling of substitutions (Yang, Goldman, and Friday, 1994). It is noteworthy that the MCMC algorithm discussed in Section 4 is not linked to the particular model of evolution.
As long as likelihood evaluation is a feasible calculation, we can readily implement a posterior simulation. This is in contrast to Gibbs sampler algorithms, for example, whose very structure is determined by the likelihood function under consideration.

In Section 5 we report calculations for the model of Hasegawa, Kishino, and Yano (1985), which, being richer in parameters, subsumes the earlier models of Jukes and Cantor (1969), Kimura (1980), and Felsenstein (1981). The generator matrix for a process governed by HKY85 contains the following infinitesimal rates of change:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>G</th>
<th>C</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-(κπ_g + π_c + π_t)</td>
<td>κπ_g</td>
<td>π_c</td>
<td>π_t</td>
</tr>
<tr>
<td>G</td>
<td>κπ_a</td>
<td>-(κπ_a + π_c + π_t)</td>
<td>π_c</td>
<td>π_t</td>
</tr>
<tr>
<td>C</td>
<td>π_a</td>
<td>π_g</td>
<td>-(κπ_t + π_a + π_g)</td>
<td>κπ_t</td>
</tr>
<tr>
<td>T</td>
<td>π_a</td>
<td>π_g</td>
<td>κπ_t</td>
<td>-(κπ_t + π_a + π_g)</td>
</tr>
</tbody>
</table>

The π’s indicate long-run probabilities of each base along one very long branch and κ allows different substitution rates for transitions and transversions. The derivation of transition probabilities from one base value to another is a standard calculation, and involves branch lengths θt where θ is an overall mutation rate and t is time.

By the independent-sites assumption, likelihood evaluation proceeds by combining the probability of each observed pattern of s bases in the aligned sequences. Furthermore, the Poisson process assumption implies that evolution is Markovian, and thus that the probability of a given pattern can be calculated recursively in O(s) steps. This pruning algorithm is a critical component of our procedure, and so we review it briefly (see also, Felsenstein, 1983). Let u_i denote the unknown base at the site of interest in the ancestral sequence associated with internal node i of the tree τ. Note that s + 1 ≤ i ≤ 2s − 1 and u_i takes one of four nucleotide values. Each internal node partitions the descendant species into two distinct groups, whose observed DNA data we label A(i) and B(i). By the assumed independence of substitutions among branches, the conditional probability of all data descending from i, given u_i, is

\[(2) \quad p\{A(i), B(i) | u_i\} = p\{A(i)|u_i\} \times p\{B(i)|u_i\}.\]

These probabilities are important because the likelihood contribution from a site with
the given pattern is

\[ \sum_{u_{\text{root}}} p\{A(\text{root}), B(\text{root}) \mid u_{\text{root}}\} p(u_{\text{root}}). \]

That is, it is a mixture of transition probabilities against the distribution of the unknown base at the root node. By taking these initial base probabilities equal to the stationary base probabilities \( \pi_a, \pi_c, \pi_t, \) or \( \pi_g, \) the Markov process becomes reversible. To implement the pruning algorithm, one observes that the Markov property, probabilities in (2) may be obtained recursively, moving from the leaves of the tree to the root. In our labeling system, the recursion moves successively through internal nodes \( i = s + 1 \) to \( i = 2s - 1. \)

We note that the \((t, \pi)\) representation of \( \tau \) is not the one most conducive to the pruning calculation which relies directly on relationship information in \( \tau. \) Our software uses a second representation in which every internal node is associated with its descendant nodes.

In summary, likelihood evaluation is a straightforward calculation when we fix the data, the tree, and parameters governing the substitution model.

4 The Posterior and MCMC

In contrast to other forms of statistical inference, Bayesian inference centers on the extent to which opinion about an unknown is affected by data. Furthermore, probability is the sole medium for transmitting uncertainty and opinion (e.g., Bernardo and Smith, 1994). To implement an analysis, a Bayesian evolutionary biologist must therefore begin with a probability distribution over the set of possible phylogenies. This might be derived from a model of speciation, or from the analysis of existing data. To our knowledge, little work has been done on the assessment of prior probabilities for trees, but this certainly represents an important problem if Bayesian analysis is to be helpful in evolutionary biology. In the present work, we illustrate calculations with a particularly simple flat prior, and note that the algorithm proceeds easily with any user-supplied prior distribution. The flat prior we assume is relative to the \((t, \pi)\) representation of the phylogeny, in suggestive notation:

\[ p(\tau) = p(t) p(\pi) \]
\[\begin{align*}
&= \left( \prod_{i=1}^{s-1} p(t_i) \right) \frac{1}{s!} \\
&\propto \left( \prod_{i=1}^{s-1} 1[0 \leq t_i \leq t_{\text{max}}] \right) \\
&\propto 1 [0 \leq t_i \leq t_{\text{max}}, \text{ for all } i].
\end{align*}\]

Here \(t_{\text{max}}\) bounds the time to the root node. One consequence of this prior is that, like the Kingman coalescent (Kingman, 1982), we induce a uniform probability distribution over labeled histories, and thus a non-uniform distribution over topologies, (e.g., Lapointe and Legendre, 1991). This prior favors balanced tree topologies over extremely unbalanced topologies. Checking the sensitivity of our calculations to the choice of prior will be critical in applications, but we do not pursue such sensitivity analysis here.

In fact, parameters of the substitution model also are unknown quantities, and a proper analysis requires a prior distribution for these parameters. The present development focuses on the phylogeny, and thus considers the substitution parameters as fixed. For example, the stationary base probabilities \(\pi_a, \pi_c, \pi_t, \pi_g\) are estimated by the relative frequency of the different bases in the observed sequences. For binary restriction-sites data, Mau and Newton (1997) developed Bayesian calculations when there is uncertainty both in the phylogeny and the substitution parameters.

In light of the data, and taking as reasonable the stochastic model of evolution, inference about the phylogeny \(\tau\) must be based on the posterior distribution, having density

\[(4) \quad p(\tau|\text{data}) \propto p(\text{data}|\tau) \times p(\tau).\]

Monte Carlo appears to be the only effective method for summarizing this distribution, even though the pruning algorithm enables evaluation of the posterior up to a constant. Inference about monophyletic groups, most probable topologies, and the uncertainty in certain branch points, for example, all are based on expectations with respect to this posterior. Within the class Monte Carlo algorithms, Markov chain methods present the most promising integration methods, and we review here the proposal of Mau, Newton, and Larget (1996).

An MCMC algorithm realizes a Markov chain \(\tau^1, \tau^2, \ldots, \tau^B\) that has (4) as its stationary distribution (e.g., Tierney, 1994). Empirical averages in the chain converge
as $B$ grows to posterior expectations by the law of large numbers for Markov chains. Furthermore, Monte Carlo error of the empirical averages may be estimated from the chain itself using time-series methods (e.g., Geyer, 1992). We construct our Markov chain using the Metropolis-Hastings approach. That is, we move from $\tau^i = \tau$ to the next state $\tau^{i+1}$ by first proposing a candidate phylogeny $\tau^*$ generated according to a proposal distribution that has transition density $q(\tau, \tau^*)$. Next we compute the Metropolis-Hastings ratio

$$r = \frac{p(\tau^* | \text{data}) q(\tau^*, \tau)}{p(\tau | \text{data}) q(\tau, \tau^*)}.$$  

If $r \geq 1$, then $\tau^{i+1} = \tau^*$. Otherwise, we move to $\tau^*$ with probability $r$ and stay put with probability $1 - r$. The power of this approach resides both in its simplicity and in its great flexibility, because the choice of $q$, which affects the Monte Carlo efficiency of the algorithm, is almost arbitrary.

In most implementations of the Metropolis-Hastings algorithm, a collection of proposal distributions determine the complete algorithm (e.g., Besag et al., 1995). We have found that a single proposal distribution works for the phylogeny problems considered so far. This proposal distribution is global in that $\tau^*$ can differ from $\tau$ in all respects, and so, in a sense, we have attempted to design efficiency into the algorithm. Inefficient algorithms are ones which traverse the parameter space slowly and thus exhibit significant positive correlations on one-dimensional summaries. Local, single-site updating proposals change parts of the parameter at a time, and are at risk for low efficiency. One risk of a global proposal distribution, on the other hand, is that we may reject candidates too frequently, and thus produce an inefficient algorithm. We avoid this in two ways: by making our global changes small in magnitude, and by basing changes on distance within the tree, so that proposed trees are close in posterior density to the current tree.

More specifically, our proposal distribution works like this. We obtain at random from the equivalence class defining the current tree $\tau$ one of its $2^{s-1}$ versions, thus identifying a pair $(t, \pi)$. Fixing the leaf label permutation $\pi$, we generate a new vector $t^*$ of times by

$$t^*_i = t_i \oplus \epsilon_i, \quad \text{for } i = 1, 2, \ldots, s - 1$$

where $\epsilon_i$ are independent and identically distributed $\text{Uniform}(-\delta, \delta)$ random variables.
for some tuning parameter $\delta > 0$, and $\oplus$ indicates addition reflected into the interval $(0, t_{\text{max}})$. For example, $\oplus$ returns $|t_i + \epsilon|$ if $t_i + \epsilon < 0$. Thus the proposal is to perturb the speciation times of a version of the current tree.

When the tuning parameter $\delta$ is small, the candidate tree is close to the current tree in terms of pairwise distance between species, and so we expect the likelihood of the candidate tree to be close to that of the current tree. Similarity in likelihood is derived by the similar distance structure, and not by a direct appeal to the model, making the proposal method independent of the model form. Interestingly, the candidate tree can be quite different from the current tree in terms of branching structure. In Figure 2, a version of $\tau$ from Figure 1 has had its species times perturbed, leading to a tree $\tau^*$ with a different topology:

$$\text{top}(\tau^*) = ((1, (3, 9)), ((2, (4, 6), (7, 8))), (5, 10)))$$

Compare with (1). Mau, Newton, and Larget (1996) established irreducibility and symmetry of this proposal distribution, symmetry implying that the Metropolis-Hastings ratio (5) reduces to a ratio of posterior densities, and thus, under a flat prior, to a ratio of likelihoods.

5 An Example

We illustrate the MCMC calculations with a subset of the data reported by Hafner et al. (1994) regarding a study of molecular evolution in hosts and their parasites. We consider DNA sequences encoding mitochondrial cytochrome oxidase 1 (CO1) for $s = 13$ species of chewing lice and for $s = 13$ species of their gopher hosts (Table 1). We use the same subset studied by Huelsenbeck and Rannala (1997) (hereafter, HR97) to facilitate a comparison. In summary, the aligned sequences contain 379 sites with 154 unique patterns in the lice data and 128 unique patterns in the gopher data. We estimated the nucleotide base probabilities with their relative frequencies in the raw data:

<table>
<thead>
<tr>
<th>data set</th>
<th>$\hat{\pi}_a$</th>
<th>$\hat{\pi}_g$</th>
<th>$\hat{\pi}_c$</th>
<th>$\hat{\pi}_t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>lice</td>
<td>.244</td>
<td>.247</td>
<td>.138</td>
<td>.371</td>
</tr>
<tr>
<td>gophers</td>
<td>.286</td>
<td>.174</td>
<td>.194</td>
<td>.345</td>
</tr>
</tbody>
</table>
The analysis by HR97 suggested a transition/transversion bias and the possibility of rate heterogeneity among sites (Yang, 1993). We allow the former but not the latter in the present calculation, and take the HKY85 $\hat{\kappa} = 7.17$ for lice and $\hat{\kappa} = 4.63$ for gophers. Time is scaled so that overall mutation rate $\theta = 1$ and the prior maximum speciation time is $t_{\text{max}} = 2$. The HR97 analysis of the molecular clock hypothesis indicated some dependence on the selected model. We restrict our analysis to a molecular clock assumption.

Proper interpretation of MCMC output hinges on useful diagnostic tools (e.g., Cowles and Carlin, 1996). Our strategy is to study simple time-series summaries of a production run after initial experimentation on shorter series used to tune MCMC parameters. A wider range of output analysis diagnostics were considered by Mau, Newton, and Larget (1996).

We analyzed the lice and gopher data separately. For each analysis, our Monte Carlo estimate of the posterior distribution over phylogenies was based on a production run of $B = (2500) \times (400)$ steps of the Metropolis-Hastings algorithm, after a burn-in period of $(100) \times (400)$ steps from a randomly chosen starting tree. Trees were stored every 400 steps. For burn-in, we began with a relatively large uniform window, $\delta = .4$, which was halved at 400-step intervals until the acceptance rate rose to a reasonable level, about 40%. We observed a rapid increase in the tree log-likelihoods until reaching a noisy plateau in both calculations. For the lice, the eventual most probable tree topology was reached by the fifth stored tree and the log-likelihood of the sixth stored tree exceeded its eventual long-run mean. These are good indicators that approximate stationarity was reached well before we terminated the burn-in run. The production run used a $\delta = .004$, leading to an acceptance rate of about 30%. Figure 3 shows some plots from our output analysis. There was very little positive autocorrelation in the log-posterior density (i.e., log-likelihood), an indication of good mixing, at least locally. Another interesting summary of the Markov chain is the indicator of hitting a particular tree topology, such as the best one. The cusum path plot (Yu, 1995) of the best-tree indicator showed substantial variability. When compared to the cusum path plot of a matched sample of independent Bernoulli random variables, the two paths showed essentially the same variability, another indication of rapid mixing (not shown). By the same diagnostics (Figure 4), our
gopher Markov chain was also well-behaved.

Tables 2 and 3 show our Monte Carlo estimates of the posterior distribution over tree topologies separately for the lice and gophers. A striking feature of the lice topologies is that they are very well determined by this CO1 data, and very little uncertainty remains. Note that the most probable topology according to our analysis matches the topology reported in HR97. By stark contrast, the uncertainty in the gopher topology is quite high, with the modal topology carrying just 12% of the probability. The topology reported by HR97 is the fifth most probable according to our analysis. (Incidentally, it is possible that the topology of the phylogeny having maximum likelihood is not the most probable topology, ignoring differences in the models used in these two calculations). Nearly all the topological uncertainty is caused by the placement gopher species 5 and 13. Looking closer at Table 3, it is helpful to identify clades $A = \{1, 2, 3, 4\}$, $B = \{6, 7, 8, 9, 10\}$, and $C = \{11, 12\}$. We estimate that each of these clades is, with posterior probability 1, a monophyletic group. Further, there is relatively little variation in each subtree topology. Ignoring species 5 and 13, the three clades $A, B, C$ join most often as $(A, C), B)$, (2086/2500 times). but $(A, (B, C))$ arises 381/2500 times and $(A, (B, C))$ arises 33/2500 times. Again, the vast majority of the uncertainty comes from the uncertain placement of 5 and 13. The posterior mode is $(((A, C), (5, 13)), B)$.

6 Concluding Remarks

Much remains to be done before we understand the utility of Bayesian methods for evolutionary biology. Even the most simple questions require sophisticated computation, however, and so we started our investigation by trying to approximate the posterior distribution over phylogeny space within the context of a standard parametric model of evolution. Initial experimentation with these computations is cause for some optimism. The problem with Markov chain Monte Carlo is not so much in developing an algorithm as it is in developing a reasonably efficient algorithm, and we think that our simple technique of perturbing speciation times shows promise. Further research is needed to uncover the relative merits of competing algorithms. In addition, it may be helpful to clarify the relationship of Bayesian and bootstrap
methodology, so that biologists will with more confidence be able to assess uncertainty in evolutionary hypotheses.

REFERENCES


Figure Captions:

Figure 1: Phylogeny: The top panel shows the raw ingredients in the \((t, \pi)\) representation of a phylogeny. Internal nodes appear at the lower end of the vertical line dropped from the horizontal. These speciation times are labeled in increasing order. The middle panel shows the tree formed by processing the first panel, that is by moving down from the leaves towards the root, establishing connections each time an internal node is reached. The bottom panel shows a second version of the same phylogeny.

Figure 2: Proposal: This graph shows how a candidate phylogeny \(\tau^*\) is obtained from one version of the current tree by perturbing speciation times in the \((t, \pi)\) representation. The shaded boxes indicate the range of the uniform perturbations. The dark circles indicate times within the current tree, and crosses indicate times in a particular candidate \(\tau^*\).

Figure 3: Output Analysis, Lice: Panels on the right are autocorrelation functions for two summaries of the phylogeny sequence sampled by MCMC: log-likelihood, and indicator of best topology. For the log-likelihood series, the left panel shows simple time series plots of the output. A cusum plot is given in the left panel for the binary indicator of best topology.

Figure 4: Output Analysis, Gophers: Same diagnostics as Figure 3.
Figure 1: Phylogeny
Figure 2: Proposal
Figure 3: Output Analysis: Lice
Figure 4: Output Analysis: Gophers
Table 1: Gopher/Lice Species Labels

<table>
<thead>
<tr>
<th>Label</th>
<th>Louse Species</th>
<th>Label</th>
<th>Gopher Species</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>G. texanus</td>
<td>1</td>
<td>G. personatus</td>
</tr>
<tr>
<td>2</td>
<td>G. ewingi</td>
<td>2</td>
<td>G. breviceps</td>
</tr>
<tr>
<td>3</td>
<td>G. oklahomensis</td>
<td>3</td>
<td>G. bursarius (a)</td>
</tr>
<tr>
<td>4</td>
<td>G. geomydis</td>
<td>4</td>
<td>G. bursarius (b)</td>
</tr>
<tr>
<td>5</td>
<td>G. nadleri</td>
<td>5</td>
<td>P. bulleri</td>
</tr>
<tr>
<td>6</td>
<td>G. chapini</td>
<td>6</td>
<td>O. hispidus</td>
</tr>
<tr>
<td>7</td>
<td>G. panamensis</td>
<td>7</td>
<td>O. cavator</td>
</tr>
<tr>
<td>8</td>
<td>G. setzeri</td>
<td>8</td>
<td>O. underwoodi</td>
</tr>
<tr>
<td>9</td>
<td>G. cherriei</td>
<td>9</td>
<td>O. cherriei</td>
</tr>
<tr>
<td>10</td>
<td>G. costaricensis</td>
<td>10</td>
<td>O. heterodus</td>
</tr>
<tr>
<td>11</td>
<td>G. expansus</td>
<td>11</td>
<td>C. castanops</td>
</tr>
<tr>
<td>12</td>
<td>G. perotensis</td>
<td>12</td>
<td>C. merriami</td>
</tr>
<tr>
<td>13</td>
<td>G. trichopi</td>
<td>13</td>
<td>Z. trichopus</td>
</tr>
</tbody>
</table>
Table 2: Posterior Distribution over Topologies, Lice: Smallest collection of topologies containing at least 90% of posterior probability.

| Rank | Topology $\text{top}(\tau)$                                                                 | $p[\text{top}(\tau)|\text{data}]$ | cumulative |
|------|---------------------------------------------------------------------------------------------|-------------------------------------|------------|
| 1    | (((((1,2),(3,4)),11),(5,13)),((6,((7,8),(9,10))),12))                                        | .855                               | .855       |
| 2    | (((((1,2),(3,4)),11),(5,13)),12),(6,((7,8),(9,10))))                                        | .036                               | .891       |
| 3    | (((((1,2),(3,4)),11),(5,13)),((6,12),(7,8),(9,10)))                                         | .032                               | .923       |
Table 3: Posterior Distribution over Topologies, Gophers: Smallest collection of topologies containing at least 90% of posterior probability.

| Rank | Topology top(τ)                                                                 | p[top(τ)|data] | cumulative |
|------|---------------------------------------------------------------------------------|----------------|------------|
| 1    | (((1,(2,(3,4)),(11,12)),(5,13)),(6,((7,8),(9,10))))                             | 0.121          | 0.121      |
| 2    | (((1,(2,(3,4)),(11,12)),(5,13),(6,((7,8),(9,10))))                             | 0.102          | 0.222      |
| 3    | (((1,(2,(3,4)),(11,12)),5),(6,((7,8),(9,10))),13))                              | 0.091          | 0.314      |
| 4    | (((1,(2,(3,4)),13),(5,(6,((7,8),(9,10))),(11,12)))                              | 0.079          | 0.393      |
| 5    | (((1,(2,(3,4)),(11,12)),(5,(6,((7,8),(9,10)))),(11,12)))                       | 0.078          | 0.471      |
| 6    | (((1,(2,(3,4)),(11,12)),5,13),(6,((7,8),(9,10))))                              | 0.076          | 0.547      |
| 7    | (((1,(2,(3,4)),(11,12)),5),(6,((7,8),(9,10))),13))                              | 0.059          | 0.606      |
| 8    | (((1,(2,(3,4)),(11,12)),13),(5,(6,((7,8),(9,10))))                             | 0.054          | 0.660      |
| 9    | (((1,(2,(3,4)),(11,12)),5),(6,((7,8),(9,10))),13))                              | 0.041          | 0.701      |
| 10   | (((1,(2,(3,4)),(11,12)),(6,((7,8),(9,10)))),(5,13))                            | 0.032          | 0.733      |
| 11   | (((1,(2,(3,4)),(11,12)),(5,(6,((7,8),(9,10)))),(13)))                          | 0.029          | 0.762      |
| 12   | (((1,(2,(3,4)),13),(5,(11,12)),(6,((7,8),(9,10))))                             | 0.024          | 0.786      |
| 13   | (((1,(2,(3,4)),13),(5,(11,12)),(5,(6,((7,8),(9,10))))(11,12)))                | 0.020          | 0.806      |
| 14   | (((1,(2,(3,4)),(5,(11,12))),(6,((7,8),(9,10)))),(13))                          | 0.014          | 0.820      |
| 15   | (((1,(2,(3,4)),13),(5,(11,12))),(6,((7,8),(9,10))))                             | 0.014          | 0.834      |
| 16   | (((1,(2,(3,4)),(11,12)),13),(5),(6,((7,8),(9,10))))                            | 0.012          | 0.846      |
| 17   | (((1,(2,(3,4)),13),(5,(6,((7,8),(9,10))))),(11,12))                            | 0.010          | 0.856      |
| 18   | (((1,(2,(3,4)),(11,12)),5),(6,((7,8),(9,10)))),(13))                           | 0.010          | 0.866      |
| 19   | (((1,(2,(3,4)),(11,12)),(5,13),(6,((7,8),(9,10))))                             | 0.010          | 0.876      |
| 20   | (((1,(2,(3,4)),(11,12)),(5,13),(6,((7,8),(9,10))))                             | 0.010          | 0.886      |
| 21   | (((1,(2,(3,4)),(5,(11,12))),(13),(6,((7,8),(9,10))))                           | 0.008          | 0.894      |
| 22   | (((1,(2,(3,4)),(5,(6,((7,8),(9,10))))),(11,12)),13)                            | 0.006          | 0.900      |