University of Alberta

Inferring, testing and summarizing a posterior distribution of phylogenies

by

Karen Ann Cranston 

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Medical Sciences - Medical Genetics

Edmonton, Alberta
Spring 2007

Your file Votre référence
ISBN: 978-0-494-29661-5
Our file Notre référence
ISBN: 978-0-494-29661-5

ABSTRACT

Phylogenetics is the study of the evolutionary histories, or phylogenies, for groups of species. Inferring phylogenies is a difficult estimation problem, and Bayesian methods are a relatively new approach. Rather than returning a point estimate of the optimal tree, a Bayesian analysis integrates over the distribution of branch lengths and model parameters, producing a posterior distribution of phylogenies. Given the large sample space and inability to calculate the required integrals analytically, Bayesian methods use Markov chain Monte Carlo (MCMC) to sample from the posterior distributions of the parameters. A good MCMC algorithm finds the regions of high probability quickly and explores these regions efficiently. Creating MCMC algorithms is challenging, and the task is further complicated by the difficulty of testing the performance of the methods. We must ensure that the sampled states have reached a stationary distribution and that we have run the method for a sufficient number of iterations for accurate inference from the sampled states.

In this thesis, I develop a new algorithm, BranchSlide, for exploration of the tree space, and then test the algorithm against existing methods. I assess the performance of the algorithm using a variety of convergence diagnostics, including a novel statistic based on the partition probabilities of the tree topology.

Results indicate that the BranchSlide proposal algorithm, given an appropriate tuning parameter, works very well over a wide range of inference problems. Very informative data sets are robust to changes in the proposal method, while harder inference problems are very sensitive to proposal methods. The analyses also indicate that a very flat posterior distribution of tree topologies still contains a large amount of information, leading to the development of a method to extract a stronger topology signal from the posterior distribution. I implement these methods in BayesTrees, a novel software package for Bayesian phylogenetic inference.

ACKNOWLEDGEMENTS

First, I want to sincerely thank my supervisor Dr. Bruce Rannala being a source of many mathematical insights and for introducing me to a world of statistics beyond the t-test. Phylogenetic theory is a relatively uncommon research focus, and throughout my degree, Bruce ensured that I had ample opportunity to meet with other researchers in the phylogenetics community.

A second round of thanks goes to the other two members of my supervisory committee, Dr. Dave Coltman and Dr. Ziheng Yang. Dr. Yang graciously hosted my visit to his research group at University College London, which was a fantastic opportunity at the start of my graduate career. Many thanks to Dr. Coltman for welcoming me as an honourary member of his laboratory and for ensuring a solid grounding in biology.

I am very grateful to Dr. Martin Sommervile for stepping in as my 'official' supervisor upon Bruce's departure at the end of my program, and to Dr. Joseph Felsenstein for reading this thesis as the external examiner.

Thank you to the members of our small but animated journal club - Tara Fulton, Greg Wilson and Isabelle Delisle - for many helpful phylogenetic discussions. Tara also kindly provided the Carnivore data used for testing of the agreement subtree method in Chapter 5. I want to sincerely thank my office mate and travelling partner Ligia Mateiu for extensive scientific and moral support.

My funding for this research was provided by a University of Alberta Dissertation Fellowship, a Province of Alberta Graduate Fellowship, a Faculty of Medicine and Dentistry 75th Anniversary Award, and two Graduate Research Assistantships from the Faculty of Medicine and Dentistry. Additional support came from Canadian Institute for Health Research grant MOP 44064 and National Institute of Health grant HGO1988 to Dr. Rannala.

I want to acknowledge the support and encouragement that I have received from my family. Mom, Dad and Wendy (as well as the extended family) never doubted my ability to complete a PhD degree, and their confidence was invaluable, especially on the bad days. Finally, I want to thank my spouse, Darren Boss. First, for his technical role as system administrator for the lab, and his endless patience with my server requests and C++ programming questions. Even more important was his tremendous emotional support. This thesis would never have been completed without it. Thank you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Phylogenetics

A phylogeny is a branching tree diagram that describes the evolutionary history shared by a group of species. The study of phylogenies, or phylogenetics, includes the inference of evolutionary histories from empirical data, as well as development of inference methods.

Phylogenies are the underpinning of modern systematics, and the ambitious Assembling the Tree Of Life (AToL) project aims to infer the phylogeny of all biodiversity on the planet. Beyond systematics, though, a large number of other research fields routinely use phylogenetic trees. These include conservation biology (Purvis et al., 2005), divergence time estimation (Sanderson, 2002; Thorne et al., 1998) and anthropology, including linguistics (Gray and Atkinson, 2003) and the study of human origins (Garrigan and Hammer, 2006; Torroni et al., 2006). Examples important to human medicine are the use of phylogenetic techniques to infer the life history and spread of pathogens (Rambaut et al., 2001; Song et al., 2005) and to the discovery of regulatory regions (Blanchette and Tompa, 2002; Wang and Stormo, 2003)). Phylogenies of pathogen species have been offered as evidence in legal proceedings (Hillis and Huelsenbeck, 1994; de Oliveira et al., 2006). Even within systematics, phylogenies based on molecular characters have had unanticipated impact, leading to the creation of the PhyloCode (de Queiroz and Gauthier, 1994), a novel system of nomenclature based on phylogenetic relatedness.

1.2 Inferring phylogenies

1.2.1 A brief history

A phylogeny includes both a topology and a set of branch lengths. The topology is the shape of the tree, describing the pattern of shared history between the species. The branch lengths detail the amount of evolutionary change between successive speciation events in the topology. A phylogenetic tree was one of the few figures included in Darwin's *On the Origin of Species*, and for quite some time after the publication of this tree, phylogenies were inferred by simply grouping species based on some aspect of phenotypic similarity. These early trees represented topologies only, as there were no available methods for inferring the length of the branches.

Statistical methods for inferring evolutionary trees required the availability of computers. Early work included that of Edwards and Cavalli-Sforza, who used distance methods, parsimony and maximum likelihood (Edwards and Cavalli-Sforza, 1963, 1964) as well the parsimony work of Sokal, Sneath and Camin (Sneath and Sokal, 1962; Camin and Sokal, 1965). These methods pre-dated the availability of molecular sequence data, instead using morphological characters and gene frequency data. At this point in the field of phylogenetics, the development of theory advanced ahead of both computing capabilities and the widespread availability of genetic data. The situation now has reversed. The introduction of molecular data provided the ability to make more sensitive comparisons between species. It allowed for development of models of sequence evolution and the introduction of rigorous parametric methods for inferring trees. Current method development in phylogenetics strives to keep pace with the increasing quantity of both sequence data and computational resources.

Methods for inferring phylogenies require the calculation of an objective function that is then used to compare the competing phylogenetic hypotheses. One such function is the parsimony score, which judges a tree to be a better explanation of the data when fewer changes (mutations) are required on the branches. As better information about the underlying processes of mutation became available, these findings were incorporated into parsimony methods (Sankoff, 1975). This then led to the development of likelihood methods, which are based on an explicit model of evolution.

1.2.2 Evolutionary models

The likelihood of a phylogeny is the probability of the data given the phylogeny and model parameters. In order to calculate the likelihood of a phylogeny, we first need a model of evolution. Model-based methods calculate the score for a tree based on

how well the data fit both the tree and the model of evolution. The use of a model allows us to infer the tree but also to make inferences about the model itself, leading to greater understanding about the evolutionary processes underlying the data.

The models used in phylogenetic inference are Markov processes that describe the rate of change from one nucleotide (or amino acid) to another. I will limit my discussion to nucleotide models. A Markov process is a stochastic (random) process with a matrix that describes the probability of change between possible states. An important property of this type of process is that the transition to a new state depends only on the current state and not on any of the previous states. In the context of evolutionary models, the Markov process can be described using a matrix of the instantaneous rates of change between nucleotide states ordered as T, C, A, G along rows and columns:

$$Q = \begin{pmatrix} \cdot & a\pi_C & b\pi_A & c\pi_G \\ a\pi_T & \cdot & d\pi_A & e\pi_G \\ b\pi_T & d\pi_C & \cdot & f\pi_G \\ c\pi_T & e\pi_C & f\pi_A & \cdot \end{pmatrix} \tag{1.1}$$

The diagonals are $Q_{ii} = -\sum_{j \neq i} Q_{ij}$, so that the rows sum to 0. The π_i terms are the frequencies of the nucleotides and the parameters a, b, c, d, e, f are the rate parameters. Generally, we scale the rates in matrix Q so that the overall rate of substitution per unit time is 1.0. This rate matrix in Equation 1.1 describes the General Time Reversibility (GTR) model, which allows each type of change between nucleotides to have its own rate. Simpler models of evolution may combine some of the rate parameters, for example, setting $a = f$ and $b = c = d = e$ so that all transitions and all transversions have the same rate but the two classes of mutations differ.

To calculate the probability of observing nucleotides i and j at the start and end of a branch of length t, we calculate the P_{ij} elements of the transition probability matrix, $P(t)$, which describes the expected rate of change over a given interval, t, of time:

$$P(t) = e^{Qt} \tag{1.2}$$

If the overall rate in Q is 1.0, then the time is measured in units of expected substitutions. For some of the simpler models of evolution, the transition probabilities can be calculated analytically, but more complex models require the matrix exponentiation in Equation 1.2. Noting that the exponent contains t, the branch length, this calculation must be repeated for each branch length on the phylogeny.

1.2.3 Calculating the likelihood for a phylogeny

The data is an aligned set of sequences for m species and n sites. The likelihood for a phylogeny is the product of the conditional probabilities over all sites in the sequence, with the assumption that each site has evolved independently. If L_i is the conditional probability of the data at site i, the likelihood is:

$$L = \prod_{i=1}^{n} L_i \tag{1.3}$$

The probabilities for each site can be extremely small, so to avoid underflow errors (numbers that are too small for a computer to handle) we generally use the sum of the log probabilities rather than the product to obtain the log-likelihood:

$$ln(L) = \sum_{i=1}^{n} ln(L_i) \tag{1.4}$$

For a given site, we calculate the probability of the data by multiplying the transition probabilities on each branch of the phylogeny. At internal nodes (where the sequence is unknown), we sum over the four nucleotide possibiliites. For the simple case of three species shown in Figure 1.1, the probability for this single site is:

$$L_i = \sum_x \sum_y p_{yA}(t_1)\, p_{xC}(t_2)\, p_{xT}(t_3)\, p_{yx}(t_4)\, \pi_y \tag{1.5}$$

where $p_{ij}(t)$ is the transition probability from nucleotide i to nucleotide j over the branch length t, calculated using the transition probability matrix from Equation 1.2. The sums are over the four possible nucleotide states at the internal nodes X and Y and π_y is the probability of observing the given nucleotide at the root node. For larger trees, we sum over the possible nucleotides at all internal nodes. As the number of species increases, this summation over all ancestral states becomes an extremely expensive calculation.

The pruning algorithm (Felsenstein, 1981) reduces the computational complexity of this calculation by identification of common factors. To use the pruning algorithm, we move the summation signs in Equation 1.5 as far to the right as possible:

$$L_i = \sum_y \pi_y p_{yA}(t_1) \left(\sum_x p_{xC}(t_2)\, p_{xT}(t_3)\, p_{yx}(t_4) \right) \tag{1.6}$$

Using this formulation, we sum over the ancestral states at node X before moving down the tree to node Y. When we calculate the sums at node X, we store the values for the four nucleotide states (the *conditional probabilities*) at X. Then, when we calculate $p_{yx}(t_4)$, we use these stored conditional probabilities, rather than

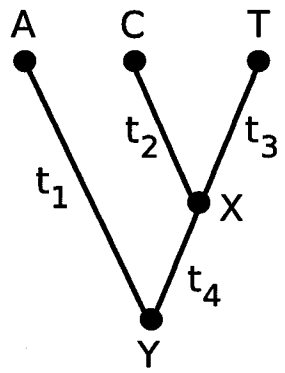


Figure 1.1: A sample phylogeny to illustrate the likelihood calculation.

repeating the calculations for the descendants of X.

To generalize this calculation for any node in any size tree, define the conditional probability, $L_c(s)$, as the probability of observing the entire subtree above node c, given that node c currently has nucleotide state s. Using the conditional probabilities, we write the marginal probability as a recursion for node c with descendants a and b and branch lengths t_a and t_b (note that I have dropped the site subscript, i, but this probability is also for a single site):

$$L_c(s) = \left[\sum_x p_{xa}(t_a)L_a(x)\right]\left[\sum_y p_{yb}(t_b)L_b(y)\right] \tag{1.7}$$

The conditional probability at a node is dependent on the length of the descendant branches and the conditional probabilities at the descendant nodes. If the descendants are tips, then the conditional probabilities for the four nucleotide states are equal to either 1.0 or 0.0, depending on whether the state matches the actual observed nucleotide. Finally, at the root of the tree, we sum over the base frequencies (from the evolutionary model) of the four nucleotide states.

1.2.4 Tree search using Maximum Likelihood

The optimal tree is the one that maximizes the likelihood function. The general strategy is to search the tree space, comparing the each tree found with the best tree discovered so far, and keeping the new tree if it has a better likelihood score. The number of possible topologies is very large, so that we cannot possibly evaluate every one, even for trees with as few as 20 species (where there are 2.22×10^{20} possible unrooted trees). Generally, we use heuristic search strategies to find the optimal trees, although these are not guaranteed to find the global optima. I will not go into details about the process (for excellent descriptions, see (Swofford et al., 1996; Felsenstein, 2004; Yang, 2006)) but I do note that this involves searching for

the best topology while also optimizing values for the model parameters and branch lengths.

Given an estimate for the maximum likelihood phylogeny, we would like some sort of measure of the quality of that phylogeny. The process of searching the tree space does not provide any such estimate, so the standard procedure is to use the nonparametric bootstrap to place confidence limits on each internal branch of the tree (Felsenstein, 1985). For each bootstrap replicate, we resample a new set of characters, with replacement, from the original sequence data and estimate the maximum likelihood tree for this new data set. The bootstrap proportion for a given partition on the tree is the number of times that the partition appears in the collection of trees inferred from the bootstrap replicates. Calculation of the bootstrap is computationally demanding, because we perform a new tree search for each new bootstrap replicate.

Bayesian methods of phylogenetic inference use the likelihood of the phylogeny as the optimality criteria, but the process of searching the tree space provides a measure of uncertainty through the posterior probabilities for partitions and for whole topologies. The application of Bayesian techniques followed more than a decade after the introduction of computationally-feasible maximum likelihood methods, and Bayesian methods were published by several groups (Yang and Rannala, 1997; Mau et al., 1999; Li et al., 2000). Before describing the specific application to phylogenetics, I first introduce the general concept of Bayesian inference.

1.3 Bayesian inference

1.3.1 General introduction

Bayesian inference uses conditional probabilities to calculate the probability of a hypothesis of interest given the available data. We can define Bayes theorem using basic relations of conditional probability:

$$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)} \tag{1.8}$$

$$\Pr(A \cap B) = \Pr(A|B)\Pr(B) \tag{1.9}$$

and

$$\Pr(B|A) = \frac{\Pr(A \cap B)}{\Pr(A)} \tag{1.10}$$

$$\Pr(A \cap B) = \Pr(B|A)\Pr(A) \tag{1.11}$$

Noting the common factor $\Pr(A \cap B)$ in both Equation 1.9 and 1.11, we can combine these two and rearrange:

$$\Pr(A|B)\Pr(B) = \Pr(B|A)\Pr(A) \tag{1.12}$$

$$\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\Pr(B)} \tag{1.13}$$

Equation 1.13 is Bayes theorem, originally proposed by the Reverend Thomas Bayes and published posthumously in 1763. The utility of this theorem is in situations where we want to know the probability of a hypothesis, A, given some data, B, but cannot calculate the probability directly. If we can instead calculate the probability of the data given the hypothesis, $P(B|A)$ and can assign an *a priori* probability to the hypothesis, $P(A)$, then we can calculate the relative probabilities of competing hypotheses. The term $P(B|A)$ is the *likelihood* of the parameter values, or hypothesis. The denominator, $P(B)$ is a scaling factor that is the expected probability of the data over all possible hypotheses (as in any probability calculation, Bayesian or otherwise).

Bayes theorem in itself is not controversial, since it follows from basic properties of conditional probability. In the application of the theory to calculate $P(A|B)$, we need to calculate both the likelihood, $P(B|A)$ and a assign a prior probability to the hypothesis, $P(A)$. It is the latter definition which is the source of much controversy.

1.3.2 Numerical integration

While the theory behind Bayes theorem is extremely simple, determining the quantity $\Pr(A|B)$ requires the integration (or summation, in the discrete case) over all possible hypotheses:

$$\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\int_A \Pr(B|A)\Pr(A)dA} \tag{1.14}$$

or

$$\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\sum_i \Pr(B|A_i)\Pr(A_i)} \tag{1.15}$$

For many problems, the calculation of the required integrals is intractable.

Therefore, practical applications of Bayesian methods to problems lacking simple analytical expressions for $P(B|A)$ had to wait for the introduction of numerical integration techniques, which sample from a distribution rather than calculating an integral directly. Monte Carlo integration, named after the famous gambling locale, is an integration technique in which we sample randomly from the distribution of interest in order to calculate the integral and summarize the distribution. Monte Carlo integration is unbiased, but can be inefficient if the majority of proposed samples have very low probability (which often occurs if the total sample space is very large). An improvement came with Markov chain Monte Carlo (MCMC), which allows us to propose samples from regions of the sample space with higher probability.

Markov chain Monte Carlo

MCMC uses a Markov chain to propose the samples for Monte Carlo integration. A Markov chain is a stochastic process where the value of any given state in the chain is dependent only on the previous state, and not on any of the earlier history of the chain. This is the same type of process used to model changes between nucleotide states in the evolutionary models described in section 1.2.2. The most common implementation of MCMC is the Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970). At each iteration of the algorithm, we propose a new state for the chain and then accept or reject the state dependent on the ratio of the *likelihood* × *prior probability* × *proposal probability* for the two states. The algorithm operates as follows:

1. Given the data, X, start with an initial state, y, selected in some fashion (for example, randomly or from the prior distribution). Calculate the likelihood, $f(X|y)$, and the prior probability, $f(y)$, of y.
2. Propose a new state, z, from the current state and calculate the probability of this proposal, $q(z|y)$, as well as the reverse proposal, $q(y|z)$.
3. Calculate the likelihood and the prior probability of z, $f(X|z)$ and $f(z)$.
4. Using the likelihood, prior and proposal probabilities for the current and proposed states, calculate the Metropolis-Hastings ratio:

$$\frac{f(X|z)}{f(X|y)} \times \frac{f(z)}{f(y)} \times \frac{q(y|z)}{q(z|y)} \tag{1.16}$$

5. Generate a uniform random number, $U \sim (0, 1)$.

6. Accept the new state, z, by setting $y = z$ if $U < \alpha(y, z)$, where:

$$\alpha(z, y) = \min\left(1, \frac{f(X|z)}{f(X|y)} \times \frac{f(z)}{f(y)} \times \frac{q(y|z)}{q(z|y)}\right) \tag{1.17}$$
$$= \min(1, \textit{likelihood ratio} \times \textit{prior ratio} \times \textit{proposal ratio})$$

   Otherwise, discard z.

7. Repeat the process for a large number of iterations until an appropriate number of samples have been collected to summarize the distribution $f(y|X)$.

The acceptance of the new state depends on the value of the M-H ratio. If the ratio is greater than 1 (if the state is better than the previous state), we always accept it. If the M-H ratio is less than 1, then we sometimes accept it (with probability α) and the rate of acceptance is proportional to the value of the M-H ratio. We are much more likely to accept a state that is only slightly worse than the current state than one that is much worse. This formulation means that the chain moves towards regions of higher probability but can also move away from a local optima.

The beauty of combining Bayes theorem with MCMC methods is the elimination of the normalizing constant, $P(B)$ in the denominator of Bayes theorem (Equation 1.13). Since $P(B)$ is a constant, it cancels out in the M-H ratio and we need not calculate it directly.

Inference of the posterior distribution

The results from a Bayesian MCMC analysis are not simply point estimates of the parameters, but an estimate of the full posterior distributions. The states of the chain from a properly-designed Metropolis-Hastings algorithm are a set of samples from the posterior distribution. If we are interested in the distribution of a parameter, we can record the list of sampled states and then use them to make inferences about the underlying distribution. The frequency of each sample value is proportional to its probability, allowing us to assign a posterior probability to a particular value or range of values. A point estimate for a Bayesian posterior distribution is the mode, which is the most frequently sampled value. A common interval estimate is the highest posterior density (HPD) interval, which is the interval containing a given percent, say 95%, of the total probability of the posterior.

Another advantage of Bayesian methods is the ability to deal with nuisance parameters - those parameters that must be specified in order to calculate the likelihood but are not specifically of interest. Bayesian inference allows us to use a model that incorporates these nuisance parameters without having to estimate the

parameters or assume specific values. We simply integrate over their posterior distribution while using the samples to infer the posterior distributions of the parameters of interest. The mechanism is the same for all parameters of the model, we can simply choose not to record the sample values for parameters that are not of interest.

1.4 Bayesian methods in phylogenetics

In the context of phylogenetics, Bayes theorem translates to

$$Pr(\tau, \theta|X) = \frac{Pr(X|\tau, \theta)Pr(\tau)Pr(\theta)}{Pr(X)} \tag{1.18}$$

where X is the data (DNA, RNA or protein sequence) and $Pr(X|\tau, \theta)$ is the likelihood of the data given the tree topology, τ, branch lengths, $\mathbf{v_i}$, and model parameters, θ. For a given topology the numerator expands to

$$Pr(X|\tau_i) = \int\int f(X|\tau_i, \theta, \mathbf{v_i}) f(v_i) f(\theta) d\mathbf{v_i} d\theta \tag{1.19}$$

where v_i are a set of branch lengths on the topology and θ_i are a set of model parameters. The double integral indicates that the likelihood for a given topology is evaluated over all possible branch lengths and values for model parameters.

Given the enormous multi-dimensional sample space (topology, branch lengths and model parameters), this integral cannot be calculated analytically and we instead use MCMC to sample from the stationary distributions of the parameters. Iterations of the MCMC modify the topology and branch lengths (often simultaneously) as well as model parameters.

1.4.1 Assigning prior distributions

In order to implement a Bayesian method for inferring phylogenies, we must specify the prior distributions on the parameters. The specification of priors is one of the major technical and philosophical differences between maximum likelihood and Bayesian methods. We hope that the likelihood function will overwhelm the influence of the prior, but this may not be the case if the data is uninformative or the model is overparameterized (Rannala, 2002).

The choice of prior can be based on existing information about the parameter (an *informative* prior) or we can attempt to choose a prior that contains as little information as possible (a *vague* prior). The birth-death prior on branch lengths is an example of the former, while a uniform branch length prior is an example of the

latter. Choosing a vague prior is more complex than it appears, as an uninformative prior under one parameterization is not equivalent under a different parameterization. A nice example of this is the contradiction created when assigning a uniform prior both to the probability of change along a branch and also to the length of the branch (Felsenstein, 2004).

Priors on the branch lengths of the phylogeny depend upon whether we assume the molecular clock. Under the clock assumption, rates of evolution are constant through the tree so that branch lengths are in units of time. Given that all of the species on the tree are sampled at the present time, this implies that the root-to-tip path is equal for all tips (such a tree is said to be *ultrameric*).

A uniform prior on branch lengths can be used for any phylogeny, irrespective of whether the branches are constrained by the molecular clock assumption. Although an option in many inference packages, the uniform prior is rarely used to due its lack of biological justification.

For clock-like trees, the most common prior is the birth-death prior, which describes the collection of branches on the tree using a linear birth-death process. There are three parameters. The birth rate, λ, describes the rate of splitting one lineage into two lineages. The death rate, μ, is the rate of extinction of a lineage. The final parameter is the species sampling, ρ, which is the proportion of total extant species that are represented in the phylogeny. The branch lengths are described in terms of the node times, or waiting times between speciation events. Since we are only interested in the tree describing currently extant species, we do not see the extinction events. This is the *reconstructed* birth-death process (Nee et al., 1994).

For trees with branches that are not constrained by the molecular clock, the most common prior is the exponential. Each branch length on the tree is assumed to be an independent random variable drawn from a single exponential distribution. This prior cannot be used on clock-like trees, as the clock constraint means that the branch lengths are not independent (since the sum of all root-to-tip paths must be equal). A exponential prior that assigns different rates to external and internal branches has also been proposed (Yang and Rannala, 2005), but has not yet been implemented into an available software package.

1.4.2 Advantages of Bayesian methods

A major advantage of Bayesian methods is that the result is an estimate of the posterior distribution of the parameters, rather than a point estimate. In the context of phylogenetics, the result is a posterior distribution of phylogenies. We can use the frequency of a specific topology to assign a probability to a phylogeny.

The posterior probability is an intuitive and well-defined measure of the quality of the hypothesis. In addition to the full phylogeny, we can also assign posterior probabilities to the individual partitions, or clades, on the tree by taking the marginal probability of the given partition over the full distribution.

The second major advantage is the ability to integrate over parameters of the model (or other parameters) while performing the phylogenetic inference. This allows for the simultaneous inference of model parameters and phylogeny, allowing the distribution of each component to be incorporated into the inference of the other.

1.4.3 Current research and thesis objectives

The choice of evolutionary model is probably the most well-researched area of Bayesian phylogenetics. The popular program ModelTest (Posada and Crandall, 1998; Posada, 2006) uses information criteria and likelihood ratio tests to help researchers choose the most appropriate models for their specific data sets. In the Bayesian framework, it is also possible to integrate over the type of model as well as the individual model parameters (Huelsenbeck et al., 2004). The effects of misspecification are fairly well understood (Bollback, 2002; Lemmon and Moriarty, 2004; Buckley, 2002; Huelsenbeck and Rannala, 2004). Underparameterization can lead to overestimation of branch lengths and also biases in partition probabilities. With an underparameterized model, high partition probabilities tend to be overestimated while low probabilities are underestimated. An overparameterized model (relative to the true model, as opposed to overparameterization in the sense of non-identifiability) can lead to reduced efficiency, but does not seem to bias the results. In studies that have examined model inadequacy, failure to include variable rates across sites caused much greater effects than assuming that transitions and transversions have the same rate or that the nucleotide frequencies are equal.

Not unrelated to model misspecification is the question of differences between Bayesian posterior probabilities and bootstrap proportions for partitions on the phylogeny. There is a tendency for posterior probabilities to be inflated relative to the bootstrap proportions, a phenomena that was noted even in the early papers on Bayesian phylogenetics (Rannala and Yang, 1996; Larget and Simon, 1999). Model specification plays a role in the differences between the two measures (Alfaro et al., 2003; Erixon et al., 2003; Suzuki et al., 2002; Wilcox et al., 2002) as does the prior distribution on branch lengths (Yang and Rannala, 2005). Also addressed in these studies is the question of whether or not we should expect the partition probabilities and bootstrap values to be equal. There are fundamental differences between the definition and calculation of the two measures of uncertainty. While we would

intuitively like to see agreement, it is unclear under which conditions, if any, the two should be equivalent.

There are a number of important recent improvements in Bayesian MCMC methods. One is the use of Metropolis-coupled Markov chain Monte Carlo (MCMCMC or MC^3) (Geyer, 1991; Huelsenbeck and Bollback, 2001) to improve convergence and mixing. In MCMCMC, we run multiple *heated* chains alongside one *cold* chain. The posterior distribution of the heated chains is modified to flatten peaks and raise valleys, meaning that the chains can more easily escape local optima. We cannot sample from these heated chains (due to the altered posterior distribution), but swapping states between chains can allow the cold chain to occasionally jump to the current state of a heated chain.

The increase in multiple gene or full genome analyses requires methods for allowing different parts of the data to have different evolutionary models (this is also possible for different codon positions within a single gene). One such method allows for heterogeneity within the data through the use of explicitly partitioned data sets (Nylander et al., 2004). Each partition can have a different model and the model parameters in each partition are inferred independently of the other partitions. A second approach is to use mixture models (Pagel and Meade, 2004), where the different models apply to different regions of the data with varying probability. Models are not explicitly assigned to partitions, but rather we allow the data to infer which model(s) most closely explain the observed data at each site.

Possibly the most promising new development is the simultaneous inference of alignment and phylogeny (Redelings and Suchard, 2005; Suchard and Redelings, 2006). The assumption of alignment accuracy has been made by virtually every phylogenetic inference method, despite the knowledge that sequence alignment is a very challenging problem.

The choice and effect of prior distributions, despite being a point of contention between Bayesians and non-Bayesians, have received surprisingly little attention. Investigations have only recently begun, including studies on the priors for branch lengths (Yang and Rannala, 2005), model parameters (Zwickl and Holder, 2004) and topology (Pickett and Randle, 2005; Brandley et al., 2006; Steel and Pickett, 2006).

Another area that requires further study is the design of MCMC algorithms, especially tree proposals and selection of tuning parameters for these proposals. The performance of MCMC methods depend on how efficiently we can move around the parameter space. The parameter space for phylogenies is complex and not well-understood, and there has been little published work on the efficiency of various tree rearrangement algorithms in the context of Bayesian MCMC. In Chapter 2, I introduce a novel tree proposal and in Chapter 4, I compare its performance with

existing methods and examine the effect of different algorithms and tuning parameters on MCMC convergence and mixing.

Bayesian methods produce an enormous amount of output in the form of sampled states for each parameter. Analysis of the output includes both detection of MCMC convergence and also summary methods for the posterior distributions. When using MCMC algorithms, we must ensure that we have run the chain long enough for the samples to be representative of the stationary distribution. Chapter 3 examines the convergence of Bayesian MCMC algorithms, with particular attention paid to the convergence of the posterior distribution of phylogenies.

Currently, programs such as MrBayes (Ronquist and Huelsenbeck, 2003) or BAMBE (Simon and Larget, 2000) summarize the posterior distributions of phylogenies using the majority rule consensus tree. One of the advantages of Bayesian inference is the generation of a posterior distribution of phylogenies, with a posterior probability for each tree. Given this wealth of output, it seems that additional methods should be explored in order to capture more of the information contained in the distribution. Chapter 5 describes a novel method for summarizing the posterior distribution of phylogenies using agreement subtrees.

These areas may have remained less well-studied for a longer time for at least two reasons. The first is likely due to maximum likelihood (ML) methods pre-dating Bayesian methods. Issues of model choice also appear in ML methods of phylogenetic reconstruction, so this was not a new subject area introduced by Bayesian inference but merely one that needed to be re-visited in the Bayesian context. The bootstrap proportion / posterior probability conflict would have been immediately apparent to users moving between ML and Bayesian methods, inviting study (and critique). In contrast, ML methods do not require prior distributions, making this unfamiliar territory. Choosing proper priors is mathematically complex and determining the effect of a prior distribution is not a simple task. For choosing priors and MCMC tuning parameters, there is no equivalent software package to ModelTest, which selects among nested models based on the data. An additional challenge is that several of the above topics require creation or modification of source code for testing purposes, instead of allowing the design of studies using only the output from existing software.

1.5 A novel software package

In order to conduct the research described above, I wrote a novel software package named BayesTrees. While much of this thesis work could have been done by modifying the code from existing software (for example, MrBayes or BAMBE), I

chose to write the code from scratch. Although time consuming, this exercise provided a far better understanding of the code structure and of the underlying theory. It precluded the need to understand and modify code written by a different researcher. Also, by having my own software, I was not affected by parallel development in another laboratory over which I had no control.

The BayesTrees package for phylogenetic inference also includes two other related programs, TreeSum and MAPminer. TreeSum calculates the topology-based statistics used in Chapter 3 for the assessment of convergence. MAPminer summarizes the posterior distribution of phylogenies using output from programs such as BayesTrees, MrBayes or BAMBE. The summary method is based upon common agreement subtrees present in the distribution of phylogenies and is described further in Chapter 5.

Appendix I details the implementation of BayesTrees, TreeSum and MAPminer.

Bibliography

Alfaro, M. E., S. Zoller, and F. Lutzoni. 2003. Bayes or bootstrap? a simulation study comparing the performance of Bayesian Markov chain Monte Carlo sampling and bootstrapping in assessing phylogenetic confidence. Mol. Biol. Evol. 20:255–66.

Blanchette, M. and M. Tompa. 2002. Discovery of regulatory elements by a computational method for phylogenetic footprinting. Genome Research 12:739–748.

Bollback, J. P. 2002. Bayesian model adequacy and choice in phylogenetics. Mol. Biol. Evol. 19:1171–80.

Brandley, M. C., A. D. Leach, D. L. Warren, and J. A. McGuire. 2006. Are unequal clade priors problematic for Bayesian phylogenetics? Syst. Biol. 55:138–46; author reply 147–51.

Buckley, T. R. 2002. Model misspecification and probabilistic tests of topology: evidence from empirical data sets. Syst. Biol. 51:509–523.

Camin, J. H. and R. R. Sokal. 1965. A method for deducing branching sequences in phylogeny. Evolution 19:311–326.

de Oliveira, T., O. G. Pybus, A. Rambaut, M. Salemi, S. Cassol, M. Ciccozzi, G. Rezza, G. C. Gattinara, R. D'Arrigo, M. Amicosante, L. Perrin, V. Colizzi, C. F. Perno, and B. S. Group. 2006. Molecular epidemiology: HIV-1 and HCV sequences from Libyan outbreak. Nature advanced online publication:1–2.

de Queiroz, K. and J. Gauthier. 1994. Toward a phylogenetic system of biological nomenclature. Trends Ecol. Evol. 9:27–31.

Edwards, A. and L. Cavalli-Sforza. 1963. The Reconstruction of evolution. Annals of Human Genetics 27:105–106.

Edwards, A. and L. Cavalli-Sforza. 1964. Reconstruction of evolutionary trees. Pages 67–76 *in* Phenetic and Phylogenetic Classification (V. Heywood and J. McNeill, eds.). Systematics Association Publ.

Erixon, P., B. Svennblad, T. Britton, and B. Oxelman. 2003. Reliability of Bayesian posterior probabilities and bootstrap frequencies in phylogenetics. Syst. Biol. 52:665–673.

Felsenstein, J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. Journal of Molecular Evolution 17:368–76.

Felsenstein, J. 1985. Confidence limits on phylogenies: An approach using the bootstrap. Evol. 39:783–791.

Felsenstein, J. 2004. Inferring Phylogenies. Sinauer, Sunderland.

Garrigan, D. and M. F. Hammer. 2006. Reconstructing human origins in the genomic era. Nat. Rev. Genet. 7:669–680.

Geyer, C. 1991. Markov chain Monte Carlo maximum likelihood. Pages 156–163 *in* Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface (E. Keramidas, ed.). Interface Foundation, Fairfax Station.

Gray, R. D. and Q. D. Atkinson. 2003. Language-tree divergence times support the Anatolian theory of Indo-European origin. Nature 426:435–439.

Hastings, W. K. 1970. Monte-Carlo sampling methods using Markov chains and their applications. Biometrika 57:97–&.

Hillis, D. M. and J. P. Huelsenbeck. 1994. Support for dental HIV transmission. Nature 369:24–25.

Huelsenbeck, J. and B. Rannala. 2004. Frequentist properties of Bayesian posterior probabilities of phylogenetic trees under simple and complex substitution models. Syst. Biol. 53:904–913.

Huelsenbeck, J. P. and J. P. Bollback. 2001. Empirical and hierarchical Bayesian estimation of ancestral states. Syst. Biol. 50:351–366.

Huelsenbeck, J. P., B. Larget, and M. E. Alfaro. 2004. Bayesian phylogenetic model selection using reversible jump Markov chain Monte Carlo. Mol. Biol. Evol. 21:1123–1133.

Larget, B. and D. L. Simon. 1999. Markov chain Monte Carlo algorithms for the Bayesian analysis of phylogenetic trees. Mol. Biol. Evol. 16:750–9.

Lemmon, A. R. and E. C. Moriarty. 2004. The importance of proper model assumption in Bayesian phylogenetics. Syst. Biol. 53:265–277.

Li, S. Y., D. K. Pearl, and H. Doss. 2000. Phylogenetic tree construction using markov chain monte carlo. J. Am. Stat. Assoc. 95:493–508.

Mau, B., M. A. Newton, and B. Larget. 1999. Bayesian phylogenetic inference via Markov chain Monte Carlo methods. Biometrics 55:1–12.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. 1953. Equation of state calculations by fast computing machines. J Chem Phys 21:1087–1092.

Nee, S., R. M. May, and P. H. Harvey. 1994. The reconstructed evolutionary process. Philos Trans R Soc Lond B Biol Sci 344:305–311.

Nylander, J. A. A., F. Ronquist, J. P. Huelsenbeck, and J. L. Nieves-Aldrey. 2004. Bayesian phylogenetic analysis of combined data. Syst. Biol. 53:47–67.

Pagel, M. and A. Meade. 2004. A phylogenetic mixture model for detecting pattern-heterogeneity in gene sequence or character-state data. Syst. Biol. 53:571–581.

Pickett, K. M. and C. P. Randle. 2005. Strange Bayes indeed: uniform topological priors imply non-uniform clade priors. Mol. Phylogenet. Evol. 34:203–211.

Posada, D. 2006. ModelTest Server: a web-based tool for the statistical selection of models of nucleotide substitution online. Nucleic Acids Res 34:W700–W703.

Posada, D. and K. A. Crandall. 1998. MODELTEST: testing the model of DNA substitution. Bioinformatics 14:817–818.

Purvis, A., J. L. Gittleman, and T. M. Brooks, eds. 2005. Phylogeny and Conservation. Cambridge University Press.

Rambaut, A., D. L. Robertson, O. G. Pybus, M. Peeters, and E. C. Holmes. 2001. Human immunodeficiency virus. Phylogeny and the origin of HIV-1. Nature 410:1047–1048.

Rannala, B. 2002. Identifiability of parameters in MCMC Bayesian inference of phylogeny. Syst. Biol. 51:754–760.

Rannala, B. and Z. Yang. 1996. Probability distribution of molecular evolutionary trees: A new method of phylogenetic inference. Journal of Molecular Evolution 43:304–311.

Redelings, B. D. and M. A. Suchard. 2005. Joint Bayesian estimation of alignment and phylogeny. Syst. Biol. V54:401–418.

Ronquist, F. and J. P. Huelsenbeck. 2003. MrBayes 3: Bayesian phylogenetic inference under mixed models. Bioinformatics 19:1572–1574.

Sanderson, M. J. 2002. Estimating absolute rates of molecular evolution and divergence times: a penalized likelihood approach. Mol. Biol. Evol. 19:101–109.

Sankoff, D. 1975. Minimal mutation trees of sequences. SIAM J. Appl. Math. 28:35–42.

Simon, D. and B. Larget. 2000. Bayesian analysis in molecular biology and evolution (BAMBE), version 2.03 beta. Department of Mathematics and Computer Science, Duquesne University.

Sneath, P. H. A. and R. R. Sokal. 1962. Numerical taxonomy. Nature 193:855–860.

Song, H.-D., C.-C. Tu, G.-W. Zhang, S.-Y. Wang, K. Zheng, L.-C. Lei, Q.-X. Chen, Y.-W. Gao, H.-Q. Zhou, H. Xiang, H.-J. Zheng, S.-W. W. Chern, F. Cheng, C.-M. Pan, H. Xuan, S.-J. Chen, H.-M. Luo, D.-H. Zhou, Y.-F. Liu, J.-F. He, P.-Z. Qin, L.-H. Li, Y.-Q. Ren, W.-J. Liang, Y.-D. Yu, L. Anderson, M. Wang, R.-H. Xu, X.-W. Wu, H.-Y. Zheng, J.-D. Chen, G. Liang, Y. Gao, M. Liao, L. Fang, L.-Y. Jiang, H. Li, F. Chen, B. Di, L.-J. He, J.-Y. Lin, S. Tong, X. Kong, L. Du, P. Hao, H. Tang, A. Bernini, X.-J. Yu, O. Spiga, Z.-M. Guo, H.-Y. Pan, W.-Z. He, J.-C. Manuguerra, A. Fontanet, A. Danchin, N. Niccolai, Y.-X. Li, C.-I. Wu, and G.-P. Zhao. 2005. Cross-host evolution of severe acute respiratory syndrome coronavirus in palm civet and human. Proc. Natl. Acad. Sci. U.S.A. 102:2430–2435.

Steel, M. and K. M. Pickett. 2006. On the impossibility of uniform priors on clades. Mol. Phylogenet. Evol. 39:585–586.

Suchard, M. A. and B. D. Redelings. 2006. BAli-Phy: simultaneous Bayesian inference of alignment and phylogeny. Bioinformatics 22:2047–2048.

Suzuki, Y., G. V. Glazko, and M. Nei. 2002. Overcredibility of molecular phylogenies obtained by Bayesian phylogenetics. Proc. Natl. Acad. Sci. U.S.A. 99:16138–43.

Swofford, D., G. Olson, P. Waddell, and D. Hillis. 1996. Phylogeny Reconstruction vol. Molecular Systematics chap. 11. 2nd ed. Sinauer Associates Inc.

Thorne, J. L., H. Kishino, and I. S. Painter. 1998. Estimating the rate of evolution of the rate of molecular evolution. Mol. Biol. Evol. 15:1647–57.

Torroni, A., A. Achilli, V. Macaulay, M. Richards, and H. J. Bandelt. 2006. Harvesting the fruit of the human mtdna tree. Trends Genet. 22:339–345.

Wang, T. and G. D. Stormo. 2003. Combining phylogenetic data with co-regulated genes to identify regulatory motifs. Bioinformatics 19:2369–2380.

Wilcox, T. P., D. J. Zwickl, T. A. Heath, and D. M. Hillis. 2002. Phylogenetic relationships of the dwarf boas and a comparison of Bayesian and bootstrap measures of phylogenetic support. Mol. Phylogenet. Evol. 25:361–71.

Yang, Z. 2006. Computational Molecular Evolution. Oxford University Press.

Yang, Z. and B. Rannala. 1997. Bayesian phylogenetic inference using DNA sequences: a Markov Chain Monte Carlo Method. Mol. Biol. Evol. 14:717–724.

Yang, Z. and B. Rannala. 2005. Branch-length prior influences Bayesian posterior probability of phylogeny. Syst. Biol. 54:455–470.

Zwickl, D. and M. Holder. 2004. Model parameterization, prior distributions, and the general time-reversible model in Bayesian phylogenetics. Syst. Biol. 53:877–888.

Chapter 2

Tree Rearrangement Algorithms

2.1 Introduction

When using Markov chain Monte Carlo methods, at each iteration of the chain we propose a new state, compare the new state to the current state and then decide whether to accept or reject the proposal. The choice of proposal method is critical to the success of the overall algorithm, and the ideal method satisfies certain general mathematical requirements, is efficient in proposing new moves and leads to fast convergence and good mixing of the MCMC chain.

There are two mathematical requirements for proposing new states - the Markov chain defined by the proposal must be both irreducible and aperiodic. The irreducibility condition states that a given state can be reached from any other state in a finite number of moves. This prevents the chain from becoming trapped in a region with no probability of moving away. The aperiodic condition requires that the chain does not visit states in a cyclic manner.

Once these mathematical considerations are satisfied, the selection of a proposal algorithm becomes as much art as science. The choice of algorithm greatly affects the mixing properties of the chain. If the proposed states are very close to the current states, we will almost always accept the new state. Such an algorithm, however, is apt to become trapped at local optima, as the moves are not large enough to move down from a peak or to cross a valley of low probability. If instead the proposed states are very different from the current state, there is an increased probability of finding other optima, but the proposed states are more likely be in regions of low probability and to be rejected. Moves that are either too small or too large cause the chain to mix poorly and can introduce an undesirably high level of correlation between the samples. While the size of the move is not easy to monitor, the acceptance rates provide a window into the proposal behaviour. A high acceptance rate indicates that most proposed moves are small and close to the

current state, while a low acceptance rate is indicative of larger proposals.

There are many well-described methods for proposing new states for continuous numeric parameters (reviewed in Yang, 2006), such as the parameters of the evolutionary model or a single branch on the phylogeny. Proposing new phylogenies is a more complex problem. The proposal generally affects both branch lengths and topology of the tree (otherwise, separate topology and branch length moves would be required). Proposals can be local, affecting only a small region of the tree, or global, affecting either the entire tree or multiple regions. Some methods propose a topology change which then may include or induce a branch length change, while others propose branch length changes that may force topology changes. Constraints on the phylogeny, such as a molecular clock assumption, upper bounds on the branch length or tree height, and the different structure of internal, tip and root nodes, must be incorporated into the proposal.

Many of the methods used to propose new trees for Bayesian inference are also used in heuristic tree search with maximum likelihood or maximum parsimony. Their use in MCMC requires calculation of the probability of the forward and reverse moves in order to calculate the Hastings, or proposal, ratio. This can be the most challenging aspect of designing a new algorithm, particularly given the constraints mentioned above. Choices must be made about whether to design the proposal so that all moves obey the constraints, to adjust the final state to bring it between the upper and lower bounds (and calculate the appropriate correction to the Hastings ratio) or to simply reject moves that take parameters outside of the allowable boundaries.

The behaviour of the proposal algorithm affects the efficiency of the chain, and this is a particularly important consideration when modifying the phylogeny. There is a relationship between the choice of tree rearrangement mechanism and the cost of the likelihood calculation for the proposed tree. When calculating likelihood using the pruning algorithm, the conditional probability at a given node depends upon the conditional probabilities at the descendant nodes but not the ancestral nodes (Felsenstein, 1981). If a rearrangement does not change any of the descendent nodes of a particular node, then we can simply use the stored conditional probabilities rather than re-calculate the transition probabilities. Since calculating the likelihood of the tree is the most costly operation in phylogenetic MCMC, we aim to minimize the scale of this operation. Rearrangement mechanisms that affect only a small part of the tree require re-calculation of the conditional probabilities for a smaller number of nodes. This, of course, must be balanced against appropriate mixing of the MCMC algorithm in terms of tree proposals.

In this chapter, I describe some existing proposal algorithms for phylogenies as

well as a novel method. Following the descriptions is a comparison of the general efficiencies of each method. Then I define the expected distribution of topology and branch lengths for a small number of taxa and show that all of the methods return the expected distributions. A more thorough comparison of the performance of the various algorithms is given in Chapter 4.

2.2 Conventions

When discussing evolutionary trees, the terms phylogeny, topology and labelled history are common. The topology is the shape of the tree - the branching pattern that leads from the root to the tips. A labelled history, as the name implies, labels all nodes so that we can distinguish identical topologies with different branching orders. Finally, a phylogeny specifies both the branching pattern and the length of the branches.

Computer scientists and biologists often use different orientations when drawing and describing trees. In this thesis I use the biological convention, where the root is drawn at the bottom of the tree and the tips (or leaves) are at the top. Therefore, when describing tree rearrangement, I write of moving 'up' towards the tips and 'down' towards (or past) the root.

All of the algorithms described in this chapter operate on topologies either under the molecular clock assumption or with unconstrained branch lengths. For non-clock trees, the algorithms use an unrooted representation. While the biological meaning of the terms 'rooted' and 'unrooted' is clear, the distinction with respect to the computer programming implementation is slightly different. Coded tree representations utilizing a binary tree structure always include a root node, which is used to access the tree structure and as a starting point for tree traversals. Due to the ubiquitous presence of a root node, the 'unrooted' forms of the algorithms simply ignore the root node, treating the left and right branches of the root as one single branch.

As a final note, in the following descriptions of algorithms, all randomly selected nodes, branches or other parameter values are chosen based on a uniform distribution, unless specified otherwise.

2.3 Description of algorithms

2.3.1 Local

An early paper on Bayesian phylogenetics (Larget and Simon, 1999) described two algorithms, Local and Global. As suggested by their names, the Global algorithm

changes the entire tree in a single move, while the Local algorithm changes only a small portion of the tree. They were initially designed to work together, using Global at the start of an MCMC chain to find a rough estimate of the tree and Local after the burnin period in order to fine-tune the estimate. Both algorithms have a molecular clock and a non-clock version.

The Global algorithm has not been used for Bayesian phylogenetics outside of the software package described in the original paper. This is most likely due to the fact that it changes all branch lengths on the tree in every move, causing re-calculation of every conditional probability, making it a very expensive operation. Also, the Global method without the molecular clock assumption treats the left and right root branches as two separate branches in the tree, in contrast to the other algorithms described in this chapter. It is unclear what the effect of this would be when switching between Global and another algorithm during the course of a single MCMC chain. For these reasons, I chose to implement only the Local method.

Local without the molecular clock

The Local algorithm changes the branch lengths, and possibly the topology, but only in a small region of the tree. Application of the Local algorithm without the molecular clock is as follows:

1. Randomly select one of the $s-3$ internal branches of the tree. The internal nodes at the ends of the branch are X and Y. Refer to Figure 2.1 for labelling of the nodes.
2. Randomly select one of the other two nodes neighbouring X (node A in the illustration) and one of the nodes neighbouring Y (node C). There are three adjoining branches between A and C. The unselected children are nodes B and D.
3. Multiply the lengths of these three branches by a common scaling factor, $s = e^{\delta(U-0.5)}$, where $U \sim U(0,1)$ and δ is a tuning parameter.
4. Randomly select either X or Y and move it, along with its attached subtree, to a random point along the three branches. Nodes A and C remained unaffected by this last step.

The tree topology changes if the new insertion point for X or Y causes these two nodes to change position relative to nodes A and C. In either case, the Hastings ratio is s^3 because we multiply the three branches by the same scaling factor s. The original paper by Larget and Simon stated the Hastings ratio as s^2, which was later corrected (Holder et al., 2005). If the scaling factor s would take one or more of the proposed branch lengths out of range, we simply set the proposal ratio equal to zero

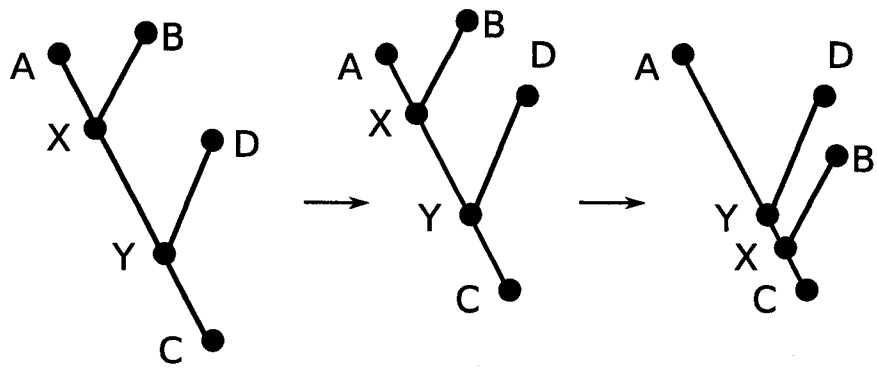


Figure 2.1: An example of a Local move without the molecular clock. The three branches AX, XY and YC are multiplied by a common scaling factor. Then, X is randomly selected to move to a new location chosen randomly between A and C. The Hastings ratio is s^3, where s is the scaling factor.

and abort the move (which is mathematically equivalent to calculating the Metropolis Hasting ratio with a prior probability of zero for the set of branch lengths).

Local with the molecular clock

In Local with the molecular clock, node X is always the child of node Y. The neighbour nodes of X are its two children (nodes A and B), while for node Y, one of the neighbours is a child (node D) and one is the ancestor (node C). The move changes the height of nodes X and Y and the links to the nodes (A,B,D). If node Y is the root of the tree, then node D does not exist and the move proceeds slightly differently. Both subtypes are detailed below:

1. Calculate the heights of the child nodes A, B and D, relative to node C (or to node Y, if Y is the root): h_A, h_B and h_D.
2. If Y is not the root of the tree select new heights for X and Y as follows:
   (a) Select the two smallest of these three heights: h_1 and h_2.
   (b) Calculate the heights of X and Y relative to node C: h_X and h_Y.
   (c) Choose two values uniformly on [0,h_1] and [0,h_2] and set h_X^* to the larger value and h_Y^* to the smaller. This means that the relative heights of X and Y do not change.
3. Otherwise, if Y is the root of the tree, select a new height for X as follows:
   (a) Modify the smallest of h_A, h_B and h_D using a multiplier proposal: $h_1^* = h_1 \times e^{\delta(U-0.5)}$.
   (b) Change the other heights deterministically based on the h1 adjustment: $h_i^* = h_i + h_1^* - h_1$ for $i = 2, 3$. The relative heights do not change.

(c) Choose a new height for X, $h_X^* \sim [0, h_2^*]$.

4. Link the children (A,B,D) to X and Y: If $h_X^* > h_1$, then the lowest child is below node X and must therefore be the child of Y. Otherwise, randomly choose one of (A,B,D) to be the child of Y and the others are children of X.

When Y is the root, the Hastings ratio is $r \times (h_1^*/h_1)$ and when Y is not the root, the Hastings ratio is simply r. The value of r depends on the initial and final height of internal node X relative to the lowest of the three child nodes A, B and D (which affects whether there is a choice about the final topology in step 4, above). Let D be the lowest node after the rearrangement. If X starts higher than D and ends lower, then there are three options for the forward move but only one for the reverse move and $r = 3/1 = 3$. Similarly, if X starts lower and ends higher, then $r = 1/3$. If the relative heights of X and D do not change, $r = 1$. MrBayes v. 3.1.2 (Ronquist and Huelsenbeck, 2003) states that there is an error in the originally published Hastings ratio for the case where Y is the root of the tree, and does not implement this aspect of the algorithm (requiring Local to be used in conjunction with another algorithm that can change the height of the tree). There is no published report detailing this issue.

2.3.2 Subtree Pruning and Regrafting

The Subtree Pruning and Regrafting (SPR) algorithm is a graph-theoretic operation that has been commonly applied to phylogenetic trees. It operates by removing a branch (and the attached subtree) from the tree and then reattaching the branch in a new location on the tree. This location can be an existing branch or, in the rooted form of the algorithm, it can be a new root location. The root option is required to allow any given move to be reversed in one step (Bordewich and Semple, 2005). The topology changes in the area surrounding the start and end attachment points, which may be in very different parts of the tree, so this is a global rearrangement. Since the branch to move and the branch containing the new location are chosen randomly, the Hastings ratio for the topology change involves only the probabilities of choosing the specific locations on the branch in the forward and reverse moves:

$$\frac{q(x|x^*)}{q(x^*|x)} = \frac{U(0, l_1)}{U(0, l_2)} = \frac{l_1}{l_2} \tag{2.1}$$

where l_2 is the length of the branch containing the new insertion point of the moved subtree and l_1 is the length of the branch containing the original location (Jow et al., 2002). Refer to Figure 2.2 for details.

In the rooted form, the branch lengths of the moved subtree are re-scaled in order to obey the molecular clock assumption and we multiply the Hastings ratio by

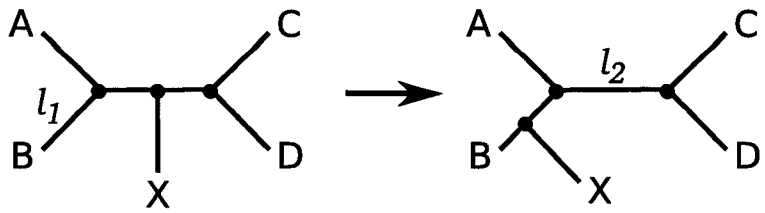


Figure 2.2: An example of a SPR move. The branch ending with node X is moved to a new location, chosen randomly from the remaining branches on the tree. The branch lengths l_1 and l_2 are used in the calculation of the Hastings ratio.

$(h'/h)^m$, where h' is the height of the subtree after the final scaling, h is the original height of the subtree and m is the number of internal nodes scaled (Rannala and Yang, 2003). The height of the tree changes if we select the root branch to move the root or if we place the selected branch in the root location.

In the unrooted form, the left and right root branches are considered as a single branch and there are no moves down past the root. I also propose a change in length to the moved branch using multiplier proposal with scaling factor $s = e^{\delta \times (U-0.5)}$ where δ is a tuning parameter and $U \sim (0, 1)$. The Hastings ratio in Equation 2.1 is then multiplied by the scaling factor, s.

2.3.3 BranchSlide

In this thesis, I present the BranchSlide algorithm, which is an extension of the SPR algorithm that allows greater control over the size of the rearrangement. The SPR move chooses the new location of the moved subtree at random, and this location may be very close to the original location, or may be in a completely different part of the topology. The BranchSlide method chooses a new location for the subtree which is a distance, $d \sim N(0, \sigma^2)$, from the original location. The use of the normal distribution to choose the distance means that most moves are small (close to the mean of zero), while the occasional move can be larger. The tuning parameter for the algorithm is the variance of the normal distribution.

BranchSlide with the molecular clock

When using the molecular clock assumption, the procedure for proposing a new tree with s taxa is as follows:

1. Randomly choose one of the $2s - 2$ branches on the tree.
2. Disconnect this branch from the tree. This can produce a single branch to move (if the descendant node is a tip) or a subtree (if the descendant is an internal node).

3. Choose a distance to move, $d \sim N(0, \sigma)$, where N is the normal distribution with mean 0 and the variance, σ is a tuning parameter. The sign of the distance indicates the initial direction: if $d < 0$, the destination node is the ancestral node, otherwise, it is the descendant node. Once the direction is determined, $d = |d|$.
4. Move the branch to a new location which is distance d away from the current location. There are three possibilities as the branch moves towards the new location:
   (a) The destination node is a tip and we reflect the unused distance over the tip, moving back down towards the root.
   (b) The destination node is the root and we choose uniformly between moving up towards the other child or down past the root (creating a new root). If we choose to move down, we reflect over an upper bound on the tree height.
   (c) The destination node is an internal node, and we choose to move in one of the two possible directions with an equal probability.
5. As we move through the tree, we subtract the traversed branch lengths from the initial distance until no distance remains. Then we reattach the branch at the new location.
6. Adjust the branch lengths of the move branch (or subtree) so that all tips are the same height. If we moved a subtree, rather than a single branch, the change in height must be scaled over the subtree.

The probability of the move is the probability of choosing the branch, times the probability of the chosen distance, times probability of choosing a direction at each internal node crossed:

$$P = \frac{1}{2s-2} \cdot \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \cdot \left(\frac{1}{2}\right)^n \tag{2.2}$$

where n is the number of internal nodes crossed. The number of branches does not change, so the first term is symmetric. The length of the tree (excluding the moved branch) does not change, so the reverse move would use the same distance. The only term in the Hastings ratio, then, comes from the final scaling of the subtree to obey the molecular clock:

$$\frac{q(x|x^*)}{q(x^*|x)} = \left(\frac{h'}{h}\right)^m \tag{2.3}$$

where h' is the height of the subtree after the final scaling, h is the original height of the subtree and m is the number of internal nodes scaled.

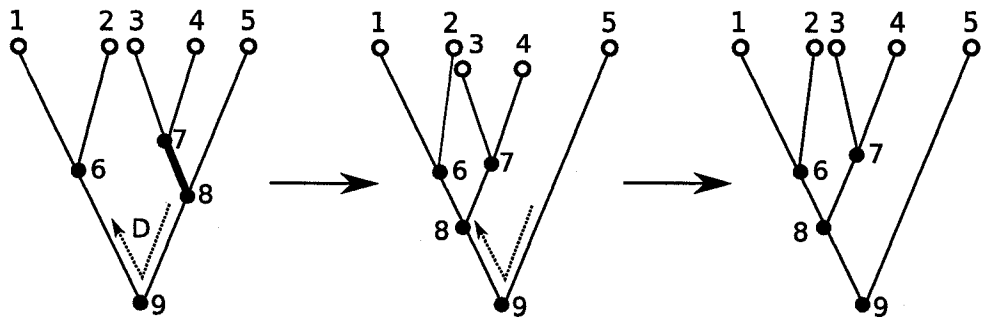


Figure 2.3: An example of a BranchSlide move with the molecular clock. Branch 7 (show in bold) is moved to a new location, with the distance D (dotted line) separating the old and new locations. After moving the branch, we pull nodes 3 and 4 up to the same height as the other tips, scaling the change in height over the branches above and below node 7.

The proposal causes a change in tree topology if the chosen distance is greater than the branch length from the insertion point of the selected branch to the destination node. The height of the tree changes if the root branch was moved or if the selected branch was moved down past the root. An example of a BranchSlide move causing a topology change is shown in Figure 2.3. After a BranchSlide move, the conditional probabilities must be recalculated for all nodes below the old and new insertion points.

BranchSlide with unconstrained branch lengths

Without the molecular clock constraint, I make the following changes to the BranchSlide algorithm:

1. Choose one of the $2s - 3$ branches on the tree, treating the two branches on either side of the root as a single branch. Disconnect the selected branch from the tree, which joins the child and ancestral branches of the target node.
2. When moving the selected branch, if we encounter the root, we simply ignore the root and move through to the other child, rather than allowing a move down past the root.
3. After reattaching the moved branch (or subtree) at the new location, propose a new length for the moved branch using multiplier proposal with scaling factor $s = e^{\delta \times (U-0.5)}$ where δ is a tuning parameter and $U \sim (0, 1)$. The Hastings ratio is then simply the scaling factor, s.

2.3.4 Other algorithms

I briefly mention several other algorithms not used in this thesis. The Tree Bisection and Reconnection (TBR) method is an SPR move where there is a new attachment point on the both the moved subtree and the remaining tree. The Nearest-Neighbour Interchange (NNI) proposal swaps two of the four subtrees on either side of an internal branch. This can also be thought of as an SPR move where the new location for the moved subtree is restricted to one of the branches that is a neighbour of the branch that contained the starting point (Bordewich and Semple, 2005). The Local algorithm can behave as an NNI move, given a choice of distances that cause a topology change.

Perusal of the source code of MrBayes v 3.1.2 reveals a number of new algorithms. Released after my development of BranchSlide, this version includes ExtendingTBR and SPR moves, which choose a new location for a branch using a probability that decreases as the branch moves farther from the originating point. The Subtree Swapper algorithm chooses two branches (or subtrees) and exchanges their position on the tree. There are no published reports detailing these methods or comparing their performance with existing move strategies, save for a single brief mention of the Subtree Swapper (Ronquist et al., 2006).

The NodeSlider algorithm (Ronquist and Huelsenbeck, 2003; Lartillot and Philippe, 2004) is similar to the Local method, but results in a smaller move. Rather than scaling the change in branch length over three branches, the change is only over two branches. The intervening node is re-located to a position that is chosen uniformly over the sum of the new length of the two scaled branches.

Noting the frequent use of the word 'similar' in the preceding paragraphs, many of the available algorithms are perturbations of the SPR / TBR / NNI class of methods, which move a branch or subtree to a new location on the tree. This basic strategy offers many options for making the rearrangements larger or smaller, depending on restrictions and tuning parameters.

2.3.5 Tuning parameters

Most of these algorithms include a tuning parameter that affects the size of the move. The effect of the tuning parameter differs for each method.

With Local and the molecular clock, the tuning parameter only comes into play when we select a branch adjacent to the root, in which case the proposed height for the child node depends on the tuning parameter of a multiplier move. Moves not involving the root have no tuning parameter. Without the molecular clock, the proposed branch lengths depend on a tuning parameter for the scaling factor. The tuning parameters for Local, then, have a small effect, both because only a small

region of the tree is changed and because some types of moves do not use a tuning parameter at all.

The Subtree Pruning and Regrafting move does not use a tuning parameter in the molecular clock version. The tuning parameter in the non-clock version affects only the proposed length of the new branch. In both versions, the choice of new location for the moved branch is completely random. Therefore, this can be a rather large move, and the size of the move is not controlled by the tuning parameter.

Finally, the BranchSlide method uses a tuning parameter for the width of the Normal distribution used to propose a new location for the moved subtree. The effect of the distance tuning parameter can be adjusted to make very large moves (moving the subtree to a very different location on the tree) or very small moves (which would generally change branch lengths only). Therefore, BranchSlide can behave either as a local or global move, depending on the tuning parameter. The effect of the parameter depends heavily on the length of the tree. If the tuning parameter is very large, the BranchSlide algorithm would behave similarly to a general SPR rearrangement (reflections at the tips and root would cause the destination branch to be chosen almost randomly). For the non-clock version, there is also a tuning parameter for the multiplier move used to scale the branch lengths in the region of the rearrangement, but the effect of this parameter is minimal compared to the variance of the Normal distribution.

At the start of the MCMC run, it may be preferable to choose a tuning parameter that allows larger moves so that the chain explores a large area of the tree space. Then, a smaller tuning parameter can fine-tune the estimate in the latter stages of the chain. The choice of optimal tuning parameter is rarely known prior to starting the MCMC run, so it is tempting to change the tuning parameter based on the acceptance rate. Unfortunately, altering the chain based on its past behaviour violates the basic principle of a Markov chain that new states are based only on the previous state. Samples taken from such a chain may not provide reliable inference about the posterior distribution (Gilks et al., 1998). It is possible, however, to adjust the tuning based on the acceptance rate in the burn-in stage and start sample collection after the tuning parameter has stabilized (Gelfand and Sahu, 1994).

Chapter 4 examines the practical aspects of choosing tuning parameters and algorithms for tree rearrangements in phylogenetic MCMC.

2.4 Testing performance

Without data, there is no likelihood term in the M-H ratio and the behaviour of the MCMC is affected only by the proposal ratio and the prior ratio. Given a prior

distribution on a parameter and a well-behaved proposal algorithm, running the method without data should return the prior distribution as the inferred posterior distribution for each parameter of interest. If the method does not return the prior distribution, the proposal is likely biased and not being corrected by the appropriate Hastings ratio. The only other explanation is that the definition of the prior is incorrect (or is not what we think it is).

This sort of test is particularly important for tree proposal algorithms. As previously mentioned, these methods are much more complex than the proposal methods for simple numerical parameters. The rearrangement is a combination of many steps, which must be correct both conceptually and in the programming implementation. Without the constraint of data, we can more easily ensure that the algorithms behave correctly at the boundaries of the parameter space (at values that would only rarely, if ever, be proposed with input data). This testing also ensures correctness of the Hastings ratio, which can have very subtle effects when erroneous, as exemplified by the elapsed time between the publication of the original Local algorithm (Larget and Simon, 1999) and the correction of its Hastings ratio (Holder et al., 2005).

For each of the algorithms described above, I ran the MCMC algorithm without data to look for return of the prior distribution on topologies and branch lenghts. Used a maximum branch length of 1.0, which allows the chain to converge to the true distribution of phylogenies in a reasonable number of iterations (less than 1000000). With a larger maximum branch length, the parameters space of the branch lengths is larger, changing the topology requires larger moves and exploration of the full tree space requires more iterations.

2.4.1 Distribution of topologies

Given a model of cladogenesis as a prior, we can derive the theoretical distribution of phylogenies and ensure that the algorithms return the expected prior distribution. For four and five taxa, the number of topologies is small enough that we expect the chain to visit all possible states and we can compare the predicted distribution with the distribution realized from the MCMC.

With 4 taxa, there are 15 possible rooted labelled histories and two possible topologies under the molecular clock: ((X,X),(X,X)) and ((X,X),X),X). Under the Yule or Equal-Rate Markov (ERM) model of cladogenesis (Yule, 1925), speciation events occur with equal probability on any given branch. For 4-taxon rooted trees, this model produces a distribution of topologies where asymmetric topologies are twice as likely as symmetric topologies. I confirmed that the algorithms were returning the correct distribution of topologies using a χ^2 test (see Table 2.1).

Table 2.1: Results of χ^2 test for distribution of 4-taxon rooted topologies. Results for each algorithm are the average of three runs of 1100000 iterations, sampling every 100 and discarding the first 100000 as burn in (for 10000 total samples). The tabulated χ^2 value ($\alpha = 0.05$, df $= 1$), used as the lower limit for statistical significance, is 5.99.

Algorithm	Symmetric	Asymmetric	χ^2
BranchSlide	20043	9957	0.2774
Local	20014	9986	0.0294
SPR	19955	10045	0.3038

Table 2.2: Results of χ^2 test for distribution of 4-taxon unrooted topologies. Results for each algorithm are the average of three runs with a total of 10000 samples per run). The tabulated χ^2 value ($\alpha = 0.05$, df $= 2$) is 7.81.

Algorithm	Tree 1	Tree 2	Tree 3	χ^2
BranchSlide	9909	10061	10030	1.2902
Local	9951	10096	9953	1.3826
SPR	9880	10185	9937	5.2965

For non-clock topologies, the position of the root is ignored and there is only one possible topology, but three possible labelled histories: (W,X,(Y,Z)), (W,(X,Y),Z) and ((W,Y),X,Z). We expect each one to be equally likely, so that ratio is 1:1:1. Table 2.2 details results from χ^2 test showing that all three algorithms return the correct distribution of trees.

2.4.2 Distribution of branch lengths

For trees constrained by the molecular clock assumption, I test the branch lengths returned under a uniform and birth-death prior. The branch lengths are defined with respect to the node times, or heights, of the branching points on the tree. In all cases, there is also an upper limit on the branch lengths. The upper bound on branches is also the upper limit for the tree height, since the ancestral branch of a one-taxon outgroup also represents the full height of the tree under the molecular clock assumption.

For a birth-death prior, the expected distribution of node times for a linear birth-death process can be generated by simulating the order statistics. I generated the expected distribution of node times using Mathematica (Wolfram Research, 2003) by simulating two random variables using the inverse transformation method described in (Yang and Rannala, 1997). This is a two-step process for each node time. First, generate a uniform(0,1) random deviate. Then, a node time, y, is:

Table 2.3: Mean and 95% confidence intervals for expected and obtained node times for the birth death process $(\lambda, \mu, \rho) = (1.63, 0.5, 1.0)$.

Method	Rep	t2	t3
Simulated	1	0.215 (0.212, 0.219)	0.524 (0.519, 0.529)
Simulated	2	0.214 (0.210, 0.218)	0.521 (0.515, 0.526)
Simulated	3	0.215 (0.211, 0.219)	0.523 (0.518, 0.528)
BranchSlide	1	0.212 (0.208, 0.215)	0.518 (0.513, 0.523)
BranchSlide	2	0.216 (0.212, 0.219)	0.520 (0.515, 0.525)
BranchSlide	3	0.213 (0.209, 0.217)	0.517 (0.512, 0.522)
Local	1	0.211 (0.207, 0.215)	0.521 (0.516, 0.527)
Local	2	0.212 (0.208, 0.215)	0.520 (0.515, 0.525)
Local	3	0.210 (0.207, 0.214)	0.517 (0.512, 0.522)

$$y = \frac{\log\{\phi - U\rho\lambda\} - \log\{\phi - U\rho\lambda + U(\lambda - \mu)\}}{\mu - \lambda} \tag{2.4}$$

where λ is the speciation rate, μ is the extinction rate, ρ is the species sampling and

$$\phi = \frac{\rho\lambda(e^{(\mu-\lambda)} - 1) + (\mu - \lambda)e^{(\mu-\lambda)}}{e^{(\mu-\lambda)} - 1} \tag{2.5}$$

Ordering these simulated values gives the expected node times for the phylogeny, conditional on the number of taxa. I repeated this sequence for two different birth-death processes for four extant taxa: $\lambda = 1.63$, $\mu = 0.5$ and $\rho = 1.0$ and $\lambda = 2.23$, $\mu = 0.5$ and $\rho = 0.5$. The lower sampling frequency gives trees with internal branches that are shorter, on average, than external branches, which is more biologically realistic. In each case, the speciation time was calculated based on the given values of μ and ρ, a root time ($t1$) of 1.0 and four extant taxa. Table 2.3 and Table 2.4 summarize the simulated and observed node times for these two birth-death processes. In each case, there are (3 x 10000) simulated node times from Mathematica and (3 x 10000) observed node times from each algorithm. For the birth-death prior, each algorithm returns node times with confidence intervals that overlap those for the simulated data.

For rooted, clock-like trees with a uniform prior on branch lengths (and the same uniform prior on the tree height), the node times are the order statistics of a uniform distribution. Similar to testing the branch lengths under a birth-death process, we check the scaled node times against the expected values for a root time of 1.0. For n order statistics of a uniform(0,θ) distribution, the expected value of of the k_{th} order statistic is:

Table 2.4: Mean and 95% confidence intervals for expected and obtained node times for the birth death process $(\lambda, \mu, \rho) = (2.23, 0.5, 0.5)$.

Method	Rep	t2	t3
Simulated	1	0.257 (0.253, 0.261)	0.573 (0.569, 0.578)
Simulated	2	0.261 (0.257, 0.265)	0.579 (0.574, 0.584)
Simulated	3	0.256 (0.252, 0.260)	0.571 (0.566, 0.576)
BranchSlide	1	0.260 (0.256, 0.264)	0.572 (0.567, 0.577)
BranchSlide	2	0.256 (0.252, 0.260)	0.569 (0.564, 0.574)
BranchSlide	3	0.257 (0.253, 0.261)	0.572 (0.567, 0.577)
Local	1	0.260 (0.256, 0.264)	0.576 (0.571, 0.581)
Local	2	0.256 (0.252, 0.260)	0.572 (0.567, 0.577)
Local	3	0.258 (0.254, 0.262)	0.576 (0.571, 0.581)

Table 2.5: Mean and 95% confidence intervals for inferred node times using the molecular clock assumption and a U(0,1) prior on branch lengths.

Method	t2	t3
BranchSlide	0.333 (0.328, 0.337)	0.667 (0.662, 0.671)
Local	0.335 (0.330, 0.340)	0.669 (0.665, 0.674)
SPR	0.332 (0.327, 0.336)	0.665 (0.660, 0.670)

$$E(X_{(k)}) = \frac{k\theta}{n+1} \tag{2.6}$$

with variance

$$Var(X_{(k)}) = \frac{k(n-k+1)\theta}{(n+1)^2(n+2)} \tag{2.7}$$

For $\theta = 1$ and $n = 2$, $E(X_{(1)}) = 1/3$ and $E(X_{(1)}) = 2/3$, each with a variance of 0.056. Table 2.5 shows the distribution of node times for each of the algorithms. All confidence intervals contain the true mean. All tests were done in triplicate, but since all results were essentially identical, I have included only one sample result for each method in the table.

For unconstrained (unrooted) trees, the root to tip distance is not constant for all tips, so the node times cannot be described as the order statistics of the prior distribution. We can instead look explicitly at the branch lengths on the tree. The most widely used prior for unconstrained trees is the exponential, as the birth-death prior requires the molecular clock assumption and the uniform prior is biologically unrealistic. Table 2.6 shows the distribution of branch lengths for the Local and

Table 2.6: Mean and variance for two branch lengths and the tree length for 4 and 5 taxon trees inferred without molecular clock constraint and using an exponential prior with rate $\lambda = 10$. Expected value of the mean is $1/\lambda = 0.1$ and variance is $1/\lambda^2 = 0.01$. Expected value of the tree length is $(2n - 3) \times 0.1$

Method	n	Internal	External 1	Tree Length
BS	4	0.0997 (0.0010)	0.1010 (0.0103)	0.503 (0.0514)
Local	4	0.0991 (0.0095)	0.1010 (0.0104)	0.499 (0.0510)
SPR	4	0.0993 (0.0102)	0.1010 (0.0102)	0.499 (0.0502)
BS	5	0.1090 (0.0114)	0.0996 (0.0098)	0.696 (0.0688)
Local	5	0.1010 (0.0102)	0.1020 (0.0103)	0.705 (0.0717)
SPR	5	0.1090 (0.0110)	0.0991 (0.0010)	0.695 (0.0697)

BranchSlide algorithms.

2.5 Conclusions

I have described a number of existing algorithms for rearranging the tree topology and branch lengths, as well as the BranchSlide method, which is an extension of subtree pruning and regrafting (SPR). The BranchSlide method could easily be extended to include other types of moves within the same framework. With a certain probability, we could choose a new attachment point on the moved subtree, making this similar to a TBR move. It is also be possible to use a distribution other than Normal to produce new values for the distance moved. For example, distances chosen from a bimodal distribution could produce larger moves (if the modes were farther from zero), or both small and large moves (with one mode at zero and one farther from zero).

Choosing the move distance based on the branch lengths of the tree means that in well-resolved areas, small distances would translate to branch length adjustments only. Since poor resolution is associated with shorter branch lengths, small distances in areas of poor resolution would more be more likely to produce moves that change both the topology and branch lengths. MrBayes implements an ExtendingSPR move that chooses the new location for the moved branch based on an extension probability that decreases with every branch crossed. I expect that this would produce less dramatic moves than BranchSlide, as the probability of crossing a branch does not depend on the length of the branch.

Each of the described algorithms have been implemented into BayesTrees. I show that running the MCMC without data returns a prior distribution of the tree topology consistent with a Yule process. For rooted trees, the methods return node

times consistent with the order statistics of the birth-death or uniform branching process chosen as a prior. For unrooted trees, the methods return the expected exponentially distributed branch lengths given a exponential prior on branch lengths.

I note that there are models other than the Yule, or ERM, process for the distribution of topologies, and these have been recently examined by Matsen (Matsen, 2006). The different models describe different branching patterns and therefore produce distributions of topologies that differ from those using the ERM model. There was early evidence that reconstructed phylogenies do not follow the ERM model (Heard, 1992; Guyer and Slowinski, 1993), and a recent study furthered this hypothesis using a large sample of trees from an online database (Blum and Francois, 2006).

The algorithms described in this thesis implicitly use the ERM process as the prior distribution. Rearrangements are performed by randomly choosing branches and moving them within the tree, which is consistent with the idea of a random branching model. Changing this underlying distribution for tree topologies could be done by explicitly applying a prior probability term for each proposed tree topology, something that would be possible only for relatively small numbers of taxa. Since some of the discrepancy between actual and theoretical tree shapes seems to be due to changes in speciation and extinction events (Mooers and Heard, 1997), it may be possible to use a birth-death prior with lineage-specific speciation and extinction rates in order to reconcile the differences. Current formulations of the birth-death prior require the molecular clock assumption, which is violated for most phylogenetic analyses. Separation of rate and time on the phylogenies would allow for the use of varying rates of speciation and extinction while also inferring a phylogeny with unconstrained branch lengths. More information about the effect of the ERM prior on our inference of phylogenies is needed before these, or other, strategies are taken to alter the prior.

Most of the described algorithms include a tuning parameter that affects the size of the move. The magnitude of the effect depends on the particular algorithm. Given the lack of discussion in the literature about tuning parameters for phylogenetic rearrangement methods, it is likely that most users do not alter the default values when performing a Bayesian phylogenetic analysis. One strategy, then, is to develop proposal methods with tuning parameters that have a small (or no) effect on MCMC mixing and convergence. A negative consequence of this strategy is that choosing relatively 'untunable' algorithms leaves us with little ability to improve a poorly performing MCMC chain. Another option is to develop methods with strong tuning parameters which can be optimized for better

performance given a specific data set. A tuning parameter with a dramatic effect may cause great difficulty with convergence if not changed from an inappropriate default value. Users should then be encouraged to examine the MCMC output carefully and make changes to the algorithm if convergence seems to be a problem.

The goal in developing proposal algorithms is to find methods that lead to faster convergence of the MCMC to the true posterior distribution and better mixing during the sampling phase. In Chapter 4, I compare these described algorithms for a number of different data sets. In order to compare the speed of convergence and quality of mixing, we need one or more diagnostic tools for measuring convergence of the MCMC algorithms. In Chapter 3, I introduce and test a number of convergence diagnostics for Bayesian phylogenetics before returning to the question of choosing and tuning tree rearrangement methods in Chapter 4.

Bibliography

Blum, M. G. B. and O. Francois. 2006. Which random processes describe the tree of life? A large-scale study of phylogenetic tree imbalance. Syst. Biol. 55:685–691.

Bordewich, M. and C. Semple. 2005. On the computational complexity of the rooted subtree prune and regraft distance. Ann. Comb. 8:409–423.

Felsenstein, J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. Journal of Molecular Evolution 17:368–76.

Gelfand, A. and S. Sahu. 1994. On Markov Chain Monte Carlo Acceleration. J Comput. Graph. Stat. 3:261–276.

Gilks, W., G. Roberts, and S. Sahu. 1998. Adaptive Markov chain Monte Carlo through regeneration. J. Am. Stat. Assoc. 93:1045–1054.

Guyer, C. and J. B. Slowinski. 1993. Adaptive radiation and the topology of large phylogenies. Evolution 47:253–263.

Heard, S. B. 1992. Patterns in tree balance among cladistic, phenetic, and randomly generated phylogenetics trees. Evolution 46:1818–1826.

Holder, M. T., P. O. Lewis, D. L. Swofford, and B. Larget. 2005. Hastings ratio of the LOCAL proposal used in Bayesian phylogenetics. Syst. Biol. 54:961–965.

Jow, H., C. Hudelot, M. Rattray, and P. G. Higgs. 2002. Bayesian phylogenetics using an RNA substitution model applied to early mammalian evolution. Mol. Biol. Evol. 19:1591–1601.

Larget, B. and D. L. Simon. 1999. Markov chain Monte Carlo algorithms for the Bayesian analysis of phylogenetic trees. Mol. Biol. Evol. 16:750–9.

Lartillot, N. and H. Philippe. 2004. A Bayesian mixture model for across-site heterogeneities in the amino-acid replacement process. Mol. Biol. Evol. 21:1095–1109.

Matsen, F. A. 2006. A geometric approach to tree shape statistics. Syst. Biol. 55:652–661.

Mooers, A. and S. Heard. 1997. Inferring evolutionary processes from phylogenetic tree shape. Q. Rev. Biol. 72:31–54.

Rannala, B. and Z. Yang. 2003. Bayes estimation of species divergence times and ancestral population sizes using DNA sequences from multiple loci. Genetics 164:1645–1656.

Ronquist, F. and J. P. Huelsenbeck. 2003. MrBayes 3: Bayesian phylogenetic inference under mixed models. Bioinformatics 19:1572–1574.

Ronquist, F., B. Larget, J. P. Huelsenbeck, J. B. Kadane, D. Simon, and P. van der Mark. 2006. Comment on "Phylogenetic MCMC algorithms are misleading on mixtures of trees". Science 312:367; author reply 367.

Wolfram Research, I. 2003. Mathematica version 5.0. Champlain, IL.

Yang, Z. 2006. Computational Molecular Evolution. Oxford University Press.

Yang, Z. and B. Rannala. 1997. Bayesian phylogenetic inference using DNA sequences: a Markov Chain Monte Carlo Method. Mol. Biol. Evol. 14:717–724.

Yule, G. U. 1925. A mathematical theory of evolution, based on the conclusions of Dr J C Willis, F R S. Philos. Trans. R. Soc. Lond., B, Biol. Sci. 213:21–87.

Chapter 3

Detecting MCMC convergence in Bayesian phylogenetics

3.1 Introduction

When we use MCMC as a numerical integration technique, the samples produced by the chain act as our estimate of the true underlying distribution. When converged, a properly constructed MCMC chain samples from the stationary distribution and the resulting output can be safely used to make inferences about various properties of the distribution. There are two aspects to convergence - the chain must both find the region(s) of highest probability and also sample sufficiently in these regions.

At the beginning of the MCMC analysis, samples from the chain are overly influenced by the starting point, which is usually chosen at random and located in a region of low probability. We should not begin using samples from the chain for inference until the effect of the starting values of the parameters is not longer evident. This ensures that samples are representative of the underlying distribution rather than being unduly influenced by the random starting values. These first samples are known as the initial transient, or more commonly, the burn-in phase.

Once we have chosen to begin sampling, the subsequent concerns are when to stop sampling and how frequently to save sampled states from the chain. The choice of total run length, or when to stop sampling, has received less attention in phylogenetics than the length of the burn-in period. A longer sampling phase more thoroughly explores the sample space, giving more accurate estimates of the properties of the distribution, particularly in terms of the variance. A longer chain will improve estimates of the stationary distribution, but requires more computing time. The optimal length will depend on the behaviour of the chain, the requirements of the study and the available resources.

The path that the chain takes while exploring the sample space greatly affects the number of iterations required for the chain to converge. Due to the nature of the MCMC, a sampled state is always dependent on the previous state, and this autocorrelation may extend across many subsequent states. A high level of autocorrelation means that the chain is moving around the parameter space slowly and requires more iterations to get a proper view of the distribution than if the samples were less correlated. We can reduce autocorrelation by modifying how the MCMC proposes new states. If the moves are too small, we accept most moves but the distance between states is small. Also, if moves are too large, then very few will be accepted and the chain becomes stuck in a single state for many iterations. A common technique to reduce autocorrelation that is not dependent upon modifying the MCMC sampler is to simply sample the chain at a frequency of less than one, so that the number of samples is less than the number of iterations. This has the additional benefit of reducing the amount of MCMC output.

A determination of convergence depends upon identifying the burn-in stage, checking that the chain is at stationarity and allowing the chain to run long enough that the samples collected can be used to estimate functions of the distribution with an appropriate level of accuracy and precision.

3.1.1 Basic strategies for diagnosing convergence

Research on MCMC convergence can be broadly grouped into two areas. First, there has been some effort to calculate the rate of convergence (or at least the bounds on the rate of convergence) and the required number of samples directly from what we know *a prioi* about the distribution and the Markov transition kernel (Rosenthal, 2002, 1995). These approaches are specific for a given implementation, and changes to the distribution or MCMC implementation require a new set of calculations. The required calculations are complex, and the bounds on convergence tend to be very loose.

The second, and much more common, research area focuses on diagnosing convergence from the output of the MCMC (reviewed in Cowles, 1996; Brooks and Roberts, 1998). The pattern of sampled states contains information about the behaviour of the chain, including autocorrelation, initial transient stages and mixing. The goal of convergence diagnostics is to detect aspects of the sampled states that indicate non-convergence. Methods can be graphical or numerical. The disadvantage of this approach is that there may be a substantial time committment to running an MCMC analysis only to find problems with convergence that require a new MCMC analysis with different parameters. However, post-run diagnostics are currently the only feasible method for problems as complex as phylogenetic

inference (and for most other problems where we would use MCMC methods).

These diagnostics include graphical methods, such as plotting the output parameters against iteration number, and numerical methods, such as the PSRF and Raftery-Lewis tests discussed later in this chapter. Some tests merely look for stability of the sampled states, while others also attempt to estimate the appropriate length of the sampling phase.

Why is diagnosing convergence such a hard problem? Given that we are using MCMC because we cannot analytically determine the shape of the distribution, it is simply impossible to say for certain when our sampled states match the stationary distribution. Even in the case of simulated data, we can know the true generating model and the true tree, but this information does not allow us to know the posterior distribution of phylogenies.

Rather than examine convergence to the true distribution, the various diagnostic methods look for stationarity, or stability, of the chain and infer convergence at this point. Diagnostics can help to identify the burn-in period, estimate stationarity of the output parameters after the burn-in and identify poor mixing. A very slowly mixing chain may look as if it were converged, especially if examined over a relatively short number of iterations. A chain trapped at a local optima can pass all tests for convergence if other optima have never been visited. Even once the mean appears to be stable, there may be reduction in the variance that occurs when we continue to run the chains. Autocorrelation between the sampled states biases traditional measures of variance, so most convergence diagnostics use alternate methods to estimate variance and determine optimal run length.

In most applications of MCMC methods, the parameter space is multidimensional and is usually difficult to visualize. Each chain produces many different output parameters, and the number of sampled states per parameter may be in the thousands or millions. Each parameter will have its own rate of convergence and these rates may vary greatly between parameters. Correlation between parameters can cause problems with convergence, and this can be difficult to diagnose when multidimensionality is high. It is simply not possible to examine every output parameter, particularly if we are using multiple diagnostic methods. A decision about convergence often requires a subjective decision about what is a stable enough mean, small enough variance or a flat enough plot.

3.1.2 Single versus multiple MCMC chains

One of the ongoing debates with respect to MCMC is whether a single long chain is preferable to multiple shorter chains (for example, see (Gelman and Rubin, 1992; Geyer, 1992), with discussion). Due to the naming conventions in MrBayes, the

phrase "multiple chains" in phylogenetics often refers to the coupled chains of Metropolis-coupled MCMC (MCMCMC), but here I use the more general definition, meaning more than one independent MCMC chain.

The biggest disadvantage of multiple chain methods is the number of samples discarded as burn-in. For the same data and MCMC implementation, the burn-in stage is constant, no matter the total run length. If the burn-in comprises x iterations, when using one long chain, we discard x iterations, but with m chains, we lose $m \times x$ iterations as burn-in. As the burn in fraction increases, the problem becomes more severe.

There are two major advantages of multiple MCMC chains. The first is their value in diagnosing convergence. Given that we can never say with absolute certainty that a chain has converged, if two independent chains converge to the same distribution, we have higher confidence that this is the correct distribution. If the posterior distribution contains multiple optima, it may be quite likely that one chain becomes trapped, but the probability of all chains being trapped in the same optima decreases with the number of chains.

The second advantage is a result of increased availability of computer resources, where multiple chains can be run in parallel on a cluster of computers. Given the advantages of multiple chains and the speed of current computers, the time spent on the burn-in portion of the chain is rarely a concern, except in analyses with very long burn-in times. Running multiple chains in parallel can decrease the run time over that of a single chain with the same number of total iterations.

3.1.3 History of methods in Bayesian phylogenetics

The task of diagnosing convergence in Bayesian phylogenetic inference is further complicated by the tree topology parameter. It is a categorical variable, making it unsuitable as input for the common convergence diagnostics. At the same time, it is normally the parameter that is of greatest interest in phylogenetic analysis. One strategy in phylogenetics is to assume convergence with respect to the tree topology when other parameters of the MCMC have converged (such as log likelihood of the tree or parameters of the evolutionary model). In order to explicitly monitor the convergence of the distribution of topologies, we can use numerical summary statistics based on the tree topology.

Users of Bayesian phylogenetic inference software have relied heavily on plots of log-likelihood versus iteration number to diagnose convergence, ending the burn in phase when the log likelihood stabilizes. This is despite the early introduction of alternate multiple-chain diagnostics, such as 2-dimensional scatterplots of the partition probabilities (Huelsenbeck and Bollback, 2001). The scatterplots compare

the partition probabilities from two independent chains by plotting probabilities from chain 1 on the x-axis and chain 2 on the y-axis. Points that depart from the line $x = y$ indicate partitions that do not agree between the two chains, indicating a lack of convergence. The lack of adoption of these other diagnostic tools may be due to a lack of knowledge about MCMC methods for early users of Bayesian methods, and also due to settings in initial versions of MrBayes, which ran only a single independent MCMC chain by default (noting that coupled MCMCMC chains are equivalent to only a single MCMC chain in terms of the output available for inference).

Over time, users have become more aware of the issues surrounding MCMC convergence, aided by several software tools. The current version of MrBayes includes some convergence diagnostics, and runs two independent analyses by default. The program Tracer (Rambaut and Drummond, 2005) plots time series and posterior densities, calculates autocorrelation within chains, the correlation between pairs of parameters and estimates effective sample size (based on the autocorrelation). The web-based tool AWTY (Are We There Yet?) (Wilgenbusch et al., 2004) creates a variety of plots tracking individual partition probabilies in a MCMC sample of trees.

Examples of convergence diagnostic techniques in recent empirical systematics papers include checking stabilization of partition probabilties (Brandley et al., 2005), comparing the shape of the posterior distribution of model parameters between chains and against the prior (Castoe and Parkinson, 2006) and checking for overlap of credible sets for model parameters (Zwickl and Holder, 2004). An increasing number of users are comparing results over multiple MCMC analyses to help diagnose convergence. This is by no means an exhaustive list of techniques, but it does exemplify the increasing sophistication of users of Bayesian phylogenetic methods.

Specifically addressing the issue of MCMC convergence in Bayesian phylogenetics was a recent paper by Beiko *et al* (Beiko et al., 2006). The group used two statistics: δ and ε. The first is the sum of the differences between a set of bipartitions from two independent MCMC chains. This statistic is bounded between 0 (when all partitions have the same probability in both chains) and $2(n-3)$ (when all partitions in the first chain are absent from the second chain). The second statistic, ε, is the average standard deviation across all partitions across a number of MCMC chains. MrBayes by default calculates mean standard deviation (SD) when running more than one MCMCMC analysis (i.e. setting parameter *nruns* to more than 1). One of the key results in this paper is that the burn-in period is short and easily identified, and it is increased sampling that is important to Bayesian MCMC

phylogenetic methods. This result was obtained by comparing multiple short independent chains to very long single chains (by breaking the long chain into smaller segments and comparing to the independent chains).

3.2 Numerical convergence diagnostics

There are a number of diagnostics that have been developed to test for various aspects of convergence. In this section, I describe three of the more common tests that can be applied to a list of MCMC sampled states. These are not specific to phylogenetics and can be used with the output from any MCMC analysis. All three are implemented in the CODA (Convergence diagnosis and output analysis software for Gibbs sampling output) (Best et al., 1995) and BOA (Bayesian Output Analysis) (Smith, 2005) packages for R. The three methods are each based on different underyling theory and therefore detect different potential problems with the chain output.

3.2.1 Brooks, Gelman and Rubin diagnostic

The Brooks, Gelman and Rubin diagnostic (Gelman and Rubin, 1992; Brooks and Gelman, 1998), known as the potential scale reduction factor (PSRF), is the most well-known of the three. It is the only multi-chain diagnostic, using data from any number of independent MCMC chains to diagnose convergence. The PSRF uses an analysis-of-variance technique, comparing the variance within the chains to the variance between all chains. Given m chains with n samples per chain, the within-chain variance is:

$$W = \frac{1}{m(n-1)} \sum_{i=1}^{m} \sum_{j=1}^{n} (x_{ij} - \bar{x}_{i\cdot})^2 \tag{3.1}$$

The between chain variance is:

$$B = \frac{n}{m-1} \sum_{i=1}^{m} (\bar{x}_{i\cdot} - \bar{x}_{\cdot\cdot})^2 \tag{3.2}$$

The weighted mean of the two variance measures is:

$$\hat{r}^2 = \frac{n-1}{n} W + \frac{1}{n} B \tag{3.3}$$

and the PSRF is then calculated as:

$$\hat{R} = \frac{\hat{r}^2}{W} \tag{3.4}$$

Gelman and Rubin suggest that the two measures of variance (between and within chains) should be equal and therefore the PSRF should be approximately equal to 1.0. If the chains have not yet converged to the same distribution, W will underestimate the variance and the PSRF will be greater than 1. In this case, we can potentially reduce the variance by continuing to run the MCMC for additional iterations. Brooks and Gelman (Brooks and Gelman, 1998) improved the method by adjusting for the sampling variability in the variance estimates, producing the corrected scale reduction factor (CSRF). The authors recommend that the 0.975 quantile of the CSRF should be less than 1.2.

The PSRF has been implemented in MrBayes for model parameters and branch lengths. The software correctly emphasizes that one of the assumptions of this test is that the sampled states are normally distributed, which may not be the case for all output parameters in Bayesian phylogenetics.

3.2.2 Raftery and Lewis diagnostic

Based on the input of a single MCMC chain, the Raftery and Lewis (RL) diagnostic (Raftery and Lewis, 1992a,b) detects convergence based on two-state Markov chain theory. Raftery and Lewis propose that non-convergent behaviour may be more commonly due to poor mixing (due to high correlation between samples) rather than to an insufficient burn-in period (Raftery and Lewis, 1992b). The authors deal with autocorrelation between the sampled states by thinning the samples until the chain behave as an approximate first-order Markov chain (so that there is no dependence of a sample on more than the immediately previous sample). Then, based upon the transition probabilities for this first-order chain, it calculates a burn-in factor and number of iterations required to give a user-defined accuracy for a quantile of interest. The output of the test includes the following statistics:

k	Thinning factor
M	Number of samples to discard as burn-in
N	Total number of iterations required to estimate the quantile to the desired accuracy
N_{min}	Total number of iterations required if the samples were independent
I	Dependence factor, $I = (M + N)/N_{min}$

This test fails when either the estimated number of required iterations is greater than the total number of available iterations or the dependence factor, I, is too high (the authors suggest that $I > 5.0$ is cause for concern).

3.2.3 Heidelberger and Welch diagnostic

The Heidelberger and Welch (HW) test (Heidelberger and Welch, 1981, 1983) tests both for the length of the burn-in period and the accuracy of our estimate of the mean. Due to the presence of autocorrelation in the MCMC samples, the authors choose to work in the frequency domain, rather than the time domain, by using the spectral density of the sampled states.

The test proceeds by checking if the process is distributed as a Brownian bridge (roughly speaking, as Brownian motion with a fixed start and end value). The test uses the Cramer von Mises statistic (which is commonly used in goodness-of-fit tests). If the test fails, a block of samples is deleted from the start of the (time series) process and the test is repeated. This continues until the test passes or we have discarded more than half of the samples. The remaining samples are assumed to be at stationarity.

Once the burn-in period is eliminated, the second phase of the test determines if the remaining samples are sufficient to estimate the mean of the parameter. We then construct a confidence interval for the mean, using the variance estimates from the spectral density rather than traditional variance estimates (again, due to autocorrelation). If the halfwidth of the confidence interval is less than a user-defined accuracy, the test passes. If the test fails, we can attempt to remove another sample block from the start of the run and repeat.

The assumption made in this test is that the process has indeed converged to a stationary state by the end point of the run.

3.2.4 Autocorrelation

In addition to these named tests, we can also directly calculate the autocorrelation of the time series. Given a sequence of variables, X, that are some function of the MCMC output, the autocorrelation at lag k is

$$\rho(k) = \frac{E[(X_i - \mu)(X_{i+k} - \mu)]}{\sigma^2} \tag{3.5}$$

Rather than calculating the autocorrelation separately for a number of different lag intervals, Geyer discusses methods to sum over autocorrelation for a reasonable number of different lags (Geyer, 1992). This has been applied to an MCMC method for estimating population parameters under the coalescent (Drummond et al., 2002) and implemented into the Tracer software tool (Rambaut and Drummond, 2005) as the Autocorrelation Time (ACT).

3.3 Applying convergence diagnostics in phylogenetics

Since the primary parameter of interest in Bayesian phylogenetics is the phylogeny, it would be useful to know which diagnostic methods were most sensitive to changes in the distribution of topologies. As already mentioned, the tree topology does not lend itself well to standard diagnostic methods. We must then use numerical proxies in order to examine the tree topology indirectly. In this study, I apply a number of different convergence diagnostics to a variety of traditional and non-traditional Bayesian phylogenetic output parameters. These are compared to each other and also to statistics describing the distribution of topologies (the mode and the size of the credible sets).

3.3.1 Detecting convergence using numerical output

There are a number of available output variables from a Bayesian phylogenetic analysis. Traditionally, these include log-likelihood, tree length and model parameters. I used two additional statistics, calculated from the samples phylogenies, the Branch Score and γ statistic. The Branch Score is a measure of the squared distance between two trees (Kuhner and Felsenstein, 1994), using the difference in branch length, b_i for all N partitions that appear in either tree:

$$Bs(B, B') = \sum_{i=1}^{N} \left(b_i - b'_i\right)^2 \tag{3.6}$$

If a branch exists in one tree but not the second tree, a branch length of zero is used for the second tree to indicate its absence. For simulated data, the two trees are the tree sampled in a given MCMC iteration and the true tree.

The γ statistic (Pybus et al., 2002), is a function of the branch lengths on the tree that is based on the relative position of internal nodes. It was originally developed to test hypotheses about the birth-death process underlying the phylogeny. Given the internode distances, g_i, for a phylogeny of n taxa:

$$\gamma = \frac{\left(\frac{1}{n-2}\sum_{i=2}^{n-1}\left(\sum_{k=2}^{i} k g_k\right)\right) - \left(\frac{T}{2}\right)}{T\sqrt{\frac{1}{12(n-2)}}}, T = \left(\sum_{j=2}^{n} j g_i\right) \tag{3.7}$$

We can apply any of the aforementioned numerical convergence diagnostics to these output parameters.

3.3.2 Detecting convergence using the tree topology

With most phylogenetic inference methods, we can test the accuracy of the method by comparing the inferred tree with the true tree. In Bayesian phylogenetics, we are inferring the distribution of topologies, rather than a point estimate, and even if we know the true tree and true evolutionary model, we do not know the true posterior distribution of phylogenies.

For a distribution of phylogenies, the equivalent to the mode and variance are the MAP tree and the size of the credible set of trees. The posterior probability of the MAP tree, P_{MAP}, measures the height of the mode. These are statistics that we can use to help infer whether we have converged upon a stable distribution of topologies (which we hope is the correct posterior distribution of phylogenies). Two additional statistics are described in the following sections.

Mean Square Error of Topology

Given that the tree topology is both a very interesting and complex parameter in phylogenetics (complex in the sense that it is a combination of a number of different parameters), we looked for a statistic that might capture the type of information contained in the two-dimensional partition probability plots. Ideally, this measure would summarize all of the partitions on the tree and also allow for comparison of multiple chains. With these goals, I developed a novel statistic that uses these tree partition probabilities to numerically represent the changing tree topology, the Root Mean Square Error of Topology (RMSET).

In standard statistical analysis, the RMSE is the distance, on average, of a data point from the fitted line. In this application, we will use the distance between the partition probabilities in multiple independent MCMC chains as the data points and the mean value of the probabilities as the fitted line.

Given m chains from a MCMC analysis of a data set describing a evolutionary tree with k partitions, an combined chain that contains the samples from all m chains from the full MCMC analysis and the MAP tree for whole run, RMSET statistic compares the probability of each partition on the MAP tree in the combined chain and each of the sample chains:

$$RMSET = \sqrt{\frac{1}{k}\sum_{i=1}^{k}\left(\frac{1}{m}\sum_{j=1}^{m}(p_{i\cdot} - p_{ij})^2\right)} \tag{3.8}$$

where p_{ij} is the probability of partition i in random chain j and $p_{i\cdot}$ is the probability of partition i in the combined chain. This statistic can be considered a numerical version of the two-dimensional partition probability plot, although it is an

greatly improved version of that diagnostic. The improvement comes from the ability to easily view changes in the statistic over the course of an MCMC sampling run, in simultaneously including information from all of the partitions on the tree and in combining the results from any number of MCMC chains. It also provides a numerical measure, making it possible to plot the statistic as well as to calculate further statistics and determine the theoretical distribution.

The expected distribution of this statistic can be determined by simulation. Ideally, we would like to know the value of the RMSET when all chains are sampling from the true distribution. Since the true distribution is unknown, we instead use the combined distribution of the chains. At stationarity, each of the chains should be sampling from the same distribution.

The RMSET statistic is based on differences between partition probabilities, which cannot be simulated directly. It is possible, however, to simulate distributions of topologies in order to calculate partition probabilities. Using the set of topologies present in the combined chain, I generate a pseudo replicate using bootstrapping. The general simulation procedure is as follows:

1. Using non-parametric bootstrapping, generate a new combined chain using the list of sampled trees in the starting combined chain.
2. Generate m new sample chains by subsampling the new combined chain.
3. Calculate the RMSET statistic using the new sample chains against the new combined chain.
4. Repeat steps 1, 2 and 3 to obtain 1000 replicates in total
5. Calculate a 95% confidence interval for the mean of the RMSET statistic for the given data set.
6. Determine if the calculated RMSET statistic for the original set of randomly-started MCMC chains falls within the confidence interval.

What this procedure determines is the expected distribution of the RMSET when all three chains are sampling from the same distribution (the distribution represented by the sum of all of the chains). There is no assumption that the combined chain represents the true posterior distribution of phylogenies, so this procedure does not tell us if the chains have converged to the true distribution. This is similar to any multi-chain MCMC diagnostic, which can only compare the distributions and detect whether there are differences between the chains. I should note that if the sampled trees are not independent (due to significant autocorrelation in the MCMC sampling), then comparing the simulated value of the RMSET with the calculated value is a conservative test for convergence.

Mean Standard Deviation of Partitions

Recent versions of MrBayes include a convergence diagnostic which is the average standard deviation of partition probabilities across chains. Given m chains and k total partitions in the sample of trees, the statistic is:

$$MeanSD = \frac{1}{k}\sum_{i=1}^{k}\left(\sqrt{\frac{1}{m-1}\sum_{j=1}^{m}(p_{i\cdot} - p_{ij})^2}\right) \tag{3.9}$$

where p_{ij} is the probability of partition i in chain j and $p_{i\cdot}$ is the average probability of partition i over the three chains at this point in the chain. The software includes an option to include only those partitions whose probability is greater than a specified minimum value in at least one of the chains. This prevents the summation over a large number of very poorly supporting partitions.

There are several notable differences between the MeanSD and the RMSET. First, the MeanSD can be calculated while the MCMC analysis is ongoing, as the average probability, $p_{i\cdot}$, is only based on those samples already collected. Contrast this to the RMSET, which uses $p_{i\cdot}$ as the average in the combined chain, using the combined samples from the full analysis. As implemented in MrBayes, the MeanSD is calculated over the set of partitions with probability greater than a minimum value, so that the value of k used to calculate the statistic can change as the analysis progresses. With the RMSET, I use only the set of partitions that exist on the MAP tree so that k is constant over the course of the MCMC.

3.4 Methods

In order to test various methods of detecting convergence, I used simulated data for phylogenies of various sizes and then applied the diagnostics to the results of the phylogenetic inference. I took the approach of using a smaller number of total data sets and looking at a larger range of diagnostics. This is in contrast to the recent study on MCMC convergence in phylogenetics (Beiko et al., 2006), which examined two diagnostics over a large number of data sets. Both approaches have merits. A large number of data sets increases the possibility of finding aspects of the data or phylogeny that affect convergence. In this study, I am more interested in the merits of different diagnostic tools, so using a larger number of tests against a smaller number of data sets seemed more appropriate. The ideal of course, would be a large number in both dimensions, but constraints of time and computational resources do eventually come into play.

3.4.1 Data simulation

I simulated ten birth-death topologies for each of 10, 30 and 50 taxa, using two different sampling frequencies for the birth-death process: five trees with sampling frequency $\rho = 1.0$ and five with $\rho = 0.01$. The lower sampling frequency gives phylogenies a smaller ratio of internal:external branch lengths. Shorter internal branch lengths make the phylogeny more difficult to infer. The expected number of species, n, for a birth-death process with species sampling is given by (Nee et al., 1994):

$$n = \frac{\exp[(\lambda - \mu)t]\, P(t,T)}{P(0,T)} \tag{3.10}$$

where μ is the extinction rate, T is the tree height, λ is the speciation rate and:

$$P(t,T) = \frac{\lambda - \mu}{\rho\lambda + (\lambda(1-\rho) - \mu)\, \exp(-(\lambda - \mu)(T-t))} \tag{3.11}$$

is the probability that that a single lineage alive at time t has not gone extinct at time T. Setting $\mu = 1.0$, $T = 1.0$ and n equal to 10, 30 or 50, I calculate the expected speciation rates for each tree size by solving equation 3.10 for λ using Mathematica.

Topologies and branch lengths were generated using BayesTrees, and sequences generated on these trees using the evolver package of PAML (Yang, 1997) under a Jukes-Cantor model of evolution.

The 10 taxon trees were used as a control set of analyses. Data sets of this size should converge to the true posterior distribution with respect to both numerical MCMC output parameters and also topologies.

3.4.2 Bayesian Inference

The analysis was performed under the Jukes-Cantor model, using a uniform prior on topologies and a birth-death prior on branch lengths with parameters consistent with those used to simulate the phylogenies. For each of the 60 data sets, I ran three independent MCMC chains, starting from randomly chosen trees, for 1×10^5 iterations (1×10^6 for the 10 taxon trees), sampling every 100 iterations for a total of 5000 (10000) data points. All analyses used the BranchSlide algorithm to propose new topologies for the MCMC using a tuning parameter of 0.01.

For each data set, I collected the tree topology, log likelihood and tree length, as well as the Branch Score and γ statistic (described in section 3.3.1). The purpose of collecting these two additional variables was not to make any inferences about their specific values. Instead, I wish to expand the number of output parameters beyond

the standard tree length and log likelihood measures to determine if different variables are more sensitive to the various convergence diagnostics.

3.4.3 An empirical data set

Since real data is much less clean than simulated data, I wanted to also test the methods on an empricial data set. I selected a phylogenetic analysis of Hylid frogs (treefrogs) that contained 85 taxa (Wiens et al., 2005). The data set contained both molecular and morphological characters and the authors noted a slow rate of convergence, particularly for the morphological partitions. Since BayesTrees does not implement models for morphological data or partitioned analysis, I performed the phylogenetic analysis with MrBayes version 3.1.2 using the same parameters described in the paper. I ran 2 chains of 4.4×10^6 iterations, sampling every 1000 (the original study used 2 chains of 2×10^6 after running a single exploratory chain of 4×10^6).

3.4.4 Calculating numerical diagnostics

The three numerical convergence diagnostics (PSRF, RL and HW) are implemented in the Bayesian Output Analysis (BOA) package for R (Smith, 2005). The program takes as input a whitespace delimited file for each MCMC chain, with parameter values in separate columns. I used Perl to create scripts for R that allowed the analyses to proceed using batch mode. This involved a modification to the BOA source code in order to prevent the package from waiting for a user signal from the keyboard after each summary method.

The BOA analysis produced one output file per data set containing results for the three convergence diagnostics for each of the four output parameters (log likelihood, tree length, distance and γ). I again used Perl scripts to extract the results of the three tests from the BOA output.

3.4.5 Topology-based measures

For each data set, I calculated the RMSET, the mean standard deviation (MeanSD) of the partitions, and simulated the expected value of the RMSET. I also tracked the probability of the MAP tree in each chain and the size of the 95% credible set of topologies. Ideally, the credible sets should increase at the beginning of the analysis and then stabilize at a constant size. If we are continuing to add significant numbers of topologies to the credible set, the chain has not sufficiently explored the sample space to be considered truly converged with respect to the distribution of topologies (although our estimates of the partition probabilities may be stable even with

increasing size of the credible sets).

Given that the MeanSD uses a lower probability limit for the inclusion of a partition in the analysis, I tested a range of probability limits. For the RMSET, which uses a fixed number of partitions, I calculated the value after every 500 samples, using both the cumulative list of sampled trees and only the trees sampled in that batch.

3.5 Results

I separate the results into two sections, one for the 10 taxon trees and one for the 30 and 50 taxon trees. The 10-taxon trees were included, not because I felt these would pose a challenge to the Bayesian inference method, but because I can be confident that these analyses had converged to a stable distribution of phylogenies. Therefore, results for these data sets provide a baseline measure of what to expect from the various convergence diagnostics in an ideal case.

3.5.1 Analysis of 10 taxon trees

Summary of phylogenetic inference

For each data set, I summarize the phylogenetic inference using the probability of the MAP tree, the total number of unique trees sampled and the number of trees in the 90%, 95% and 99% credible sets of trees. These statistics give a rough idea of the underlying distribution. Table 3.1 lists these results. I note that for all data sets, the results for the three independent chains are very consistent. The MAP tree is the same topology in the three chains, and is equal to the true topology in all cases. The credible sets overlap completely (if the credible sets are the same size, they contain identical topologies, otherwise the larger set contains the smaller set plus an additional topology).

Data sets 6 through 10 (simulated under a lower sampling frequency of the birth-death process) have larger credible sets of trees, on average, than data sets 1 through 5. This is as expected, since the shorter internal branch lengths should be a a more challenging inference problem. Acceptance rates were between 21% and 26% for all analyses.

To get a sense of the distribution of topologies, I also examined the change in size of credible set and value P_{MAP} over the course of the analysis. If the analysis had indeed converged to a stable distribution of topologies, the credible sets should maintain a constant size and the P_{MAP} should be stable and the equal over the three chains. For all chains, the sizes of the credible sets differed by no more than a single tree over the course of the MCMC (post burn-in). In Figure 3.1, I show the

Table 3.1: Summary of phylogenetic inference for 10 taxon trees. For each simulated data set, we report the probability of the MAP tree, the number of unique trees sampled and the size of the 90%, 95% and 99% credible sets for each of the three independent MCMC chains (all calculated after removal of the first 500 samples as burn-in). Data sets 1 through 5 are those simulated with sampling frequency, ρ, equal 1.0 and 6 through 10 have ρ=0.01.

Data set	Chain	P_{MAP}	Unique trees	Credible sets
1	1	0.5847	12	(3, 3, 3)
	2	0.5784	13	(3, 3, 3)
	3	0.5726	13	(3, 3, 3)
2	1	0.9988	10	(1, 1, 1)
	2	0.9983	10	(1, 1, 1)
	3	0.9982	15	(1, 1, 1)
3	1	0.9989	5	(1, 1, 1)
	2	0.9982	9	(1, 1, 1)
	3	0.9982	13	(1, 1, 1)
4	1	0.9991	9	(1, 1, 1)
	2	0.9986	10	(1, 1, 1)
	3	0.9986	10	(1, 1, 1)
5	1	0.9985	11	(1, 1, 1)
	2	0.9978	13	(1, 1, 1)
	3	0.9975	17	(1, 1, 1)
6	1	0.6051	20	(3, 3, 3)
	2	0.5976	20	(3, 3, 3)
	3	0.6097	15	(3, 3, 3)
7	1	0.9483	14	(1, 2, 3)
	2	0.9444	16	(1, 2, 3)
	3	0.9413	17	(1, 2, 3)
8	1	0.9977	11	(1, 1, 1)
	2	0.9979	15	(1, 1, 1)
	3	0.9978	14	(1, 1, 1)
9	1	0.6389	28	(4, 4, 7)
	2	0.6129	24	(4, 5, 8)
	3	0.6126	20	(4, 5, 5)
10	1	0.9612	19	(1, 1, 3)
	2	0.9551	17	(1, 1, 3)
	3	0.9537	19	(1, 1, 3)

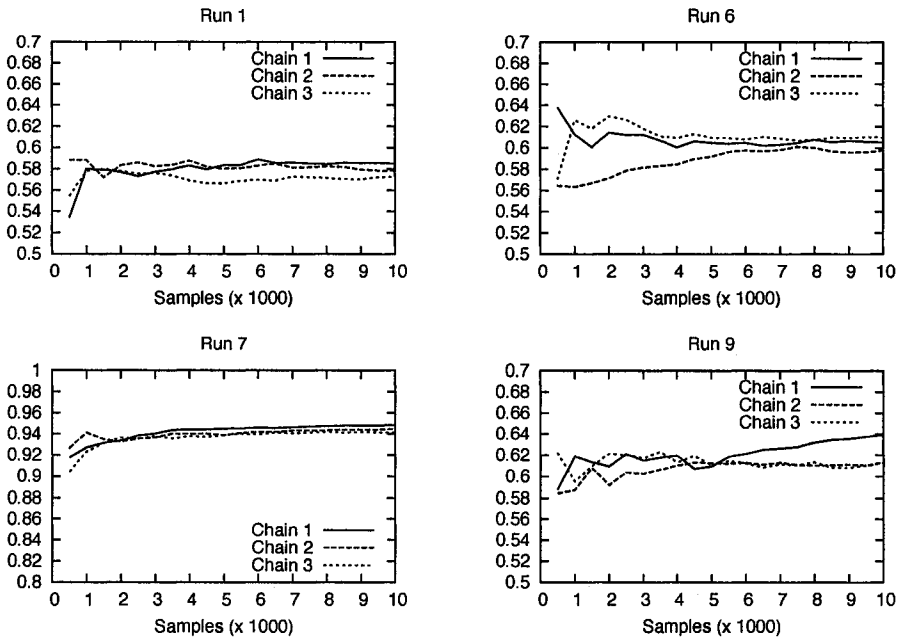


Figure 3.1: Probability of the MAP tree for four of the 10 taxon data sets.

changing probability of the MAP tree in four of the analyses (the other six are more stable than these four, with data sets 2, 3, 4 and 8 having a MAP tree with probability of nearly 1.0).

Time series plots

Table 3.2 gives the estimated burn-in evaluated by visual examination of the time series plots. The results are consistent across parameters. None of the plots indicated any other problems with the inference - parameters were visually stable following elimination of the burn-in stage.

Numerical diagnostics

In this section, I detail the results of the numerical convergence diagnostics. Note that all results here are based on the list of samples, not total iterations, so results for x iterations (samples) are equivalent to $100x$ MCMC iterations (where 100 is the sampling frequency used for the Bayesian inference).

The first diagnostic, the Potential Scale Reduction Factor (PSRF) was less than 1.00 for each of the 10 data sets for all four parameters. This is well below the 1.20 upper limit recommended by the authors of the test.

The Raftery and Lewis test outputs four results (thinning, burn-in, total

Table 3.2: Estimated burn-in from visual inspection of time series plots. Results are in terms of samples, not iterations. Sampling frequency was every 100 iterations. Estimates are rounded up to the nearest 1000 iterations.

Data set	LnL	TreeLen	Distance	γ	Mean
1	20	20	20	20	20.0
2	20	10	20	20	17.5
3	20	20	20	20	20.0
4	20	20	20	20	20.0
5	20	20	10	20	17.5
6	40	40	40	40	40.0
7	50	50	50	50	50.0
8	40	40	40	40	40.0
9	40	40	40	40	40.0
10	50	50	50	40	47.5

iterations and dependence factor) for each chain and each parameter. Table 3.4 that follows summarizes the results using mean values across chains and parameters. Data is not shown for individual chains, but I note that the results were very similar for the three chains for a given analysis.

Thinning factors are relatively small, which is not surprising given the size of the tree and the fact that this chain has already been subsampled at a frequency of 100 iterations. A thinning factor of 1.0 suggest that a smaller sampling frequency would have been possible (at a sampling frequency of 100, the samples appear independent).

Except for the log-likelihood, the burn-in values in Table 3.4 are surprisingly low, even after taking into account that these are burn-in samples, not iterations. A burn-in of 2 translates to eliminating 200 MCMC iterations, which is an order of magnitude lower than what we estimate from the time series plots (or from the HW test, see below).

The total iterations is an estimate of the number of iterations required to measure the 0.025 quantile of the given parameter to an accuracy of 0.05. This statistic is not particularly interesting in this context, as we are not actually interested in the values of these parameters, but instead on the estimates of the underlying tree topology.

The final set of RL results are the dependence factors, which are a measure of autocorrelation across the samples. The authors recommend that dependence factors should be close to 1.0 and that a result greater than 5.0 is cause for concern. Dependence factors for tree length and distance are well below this limit, while

Table 3.3: The suggested number of samples to eliminate as burn-in, from the HW test of the 10-taxon trees. All chains not reported in this table had a result of 1000 for all four parameters.

Data set	Chain	LnL	TreeLen	Distance	γ
1	1	1000	2000	1000	1000
1	2	1000	2000	1000	1000
2	2	2000	1000	2000	1000
4	1	2000	1000	1000	1000
8	2	1000	2000	1000	3000

many of the log likelihood and γ statistic values are above the limit (with some log likelihood values greater than 10.0).

Overall, I note the distinct difference between these four output parameters, even in this relatively simple phylogenetic inference problem. The log likelihood and γ parameters seem to have higher autocorrelation, causing higher thinning factors and longer burn-in times. Differences between the data sets is far less dramatic than between parameters.

For the HW tests, all chains for all analyses passed both the stationarity test and the halfwidth test (the output from BOA is simply the result "passed"). Nearly all of the analyses had a suggested number of burn-in samples equal to 1000. Since the HW tests performs the stationarity test by checking blocks that comprise 10% of the total samples, the result 1000 means that only the first block (10% of the total 10000 samples) was detected as an initial transient. In Table 3.3, I report only the chains with a result other than 1000.

	Thinning					Dependence factors				
Data set	LnL	TreeLen	Distance	γ	Mean	LnL	TreeLen	Distance	γ	Mean
1	3.3	1.0	1.0	1.7	1.8	3.7	1.0	1.0	2.0	1.9
2	3.7	1.0	1.0	3.3	2.3	4.0	1.0	1.0	4.1	2.5
3	4.0	1.0	1.0	3.7	2.4	4.6	1.0	1.0	3.6	2.6
4	3.3	1.0	1.0	2.7	2.0	3.7	1.0	1.0	2.8	2.1
5	4.3	1.0	1.0	3.0	2.3	5.4	1.0	1.0	3.3	2.7
6	5.0	1.7	1.0	1.0	2.2	6.0	1.7	1.0	1.1	2.4
7	5.0	1.0	1.0	1.0	2.0	5.5	1.1	1.0	1.1	2.2
8	4.3	1.3	1.0	1.3	2.0	4.7	1.4	1.0	1.4	2.1
9	5.0	2.0	1.0	1.0	2.3	5.5	2.0	1.0	1.1	2.4
10	4.7	1.0	1.0	1.0	1.9	4.9	1.1	1.0	1.0	2.2
Mean	4.3	1.2	1.0	2.0		4.9	1.2	1.0	2.1	
	Total Iterations					**Burn in**				
Data set	LnL	TreeLen	Distance	γ	Mean	LnL	TreeLen	Distance	γ	Mean
1	13954	3825	3782	7418	7245	10.3	2.0	2.0	4.0	4.6
2	15146	3898	3751	15188	9496	9.3	2.0	2.0	10.0	5.8
3	17300	3908	3855	13618	9670	12.0	2.0	2.0	11.0	6.8
4	13922	3782	3741	10601	8011	9.0	2.0	2.0	6.7	4.9
5	20374	3731	3897	12218	10055	11.7	2.0	2.0	8.7	6.1
6	22289	6344	3930	4018	9145	15.0	4.3	2.0	2.7	6.0
7	20715	4084	3909	4045	8188	15.0	3.0	2.3	2.7	5.8
8	17649	5249	3823	5252	7993	14.7	3.3	2.0	2.7	5.7
9	20625	7639	3866	4022	9038	15.0	4.7	2.0	2.7	6.1
10	21063	4006	3876	3928	8218	14.0	2.3	2.0	2.3	5.1
Mean	18304	4647	3842	8031		12.6	2.8	2.0	5.3	

Table 3.4: Results from the Raftery and Lewis test for convergence for the 10 taxon trees. Output includes a thinning factor, burn in, total number of iterations and dependence factors. Each cell is the average over three chains. I note that there was very good agreement between the results for each individual chain (data not shown).

Table 3.5: The RMSET and average standard deviation calculated across the three chains for the 10 taxon trees. For the MeanSD calculations, report results using different lower probability limits for including partitions in the calculation.

		MeanSD		
Data set	RMSET	Limit 0	Limit 0.10	Limit 0.20
1	1.776E-03	5.446E-04	1.547E-03	1.329E-03
2	4.007E-04	3.077E-04	4.512E-04	4.512E-04
3	4.610E-04	4.019E-04	5.280E-04	5.280E-04
4	2.828E-04	2.855E-04	3.255E-04	3.255E-04
5	3.578E-04	3.149E-04	4.267E-04	4.267E-04
6	1.745E-03	5.795E-04	1.564E-03	1.619E-03
7	9.828E-04	5.457E-04	9.834E-04	9.834E-04
8	3.723E-04	4.039E-04	3.955E-04	3.955E-04
9	5.689E-03	1.465E-03	4.572E-03	4.986E-03
10	1.113E-03	4.665E-04	1.010E-03	1.010E-03

Topology-based measures

Table 3.5 lists the final values of the RMSET and MeanSD statistics for all analyses. For the MeanSD, there are multiple values for different lower probability limits. Figure 3.2 is a graphical version of these results, more clearly showing the differences between the data sets and the statistics. For the five analyses with a single well-supported tree, the statistics have the lowest values and there is little difference between the RMSET and various MeanSD values. For the other five analyses, the RMSET (using the partitions present on the MAP tree) is consistently larger than any of the MeanSD values. This is a result of averaging over a smaller number of total partitions. The largest values are for data set 9, which also has the largest credible set of trees and greatest variation in the probability of the MAP tree over the three chains.

The lowest values are for the MeanSD with a lower limit of 0, meaning that all partitions are included and the calculation averages over a large number of very low probability partitions. This illustrates how the choice of partition limit can affect the calculation of the topology statistics. The inclusion of a greater number of partitions lowers the value of the statistic, and this lower value is primarily due to the number of low probability partitions rather than low variability between partition probabilities across the chains.

It is also possible to plot the changing value of the topology statistics over the course of the MCMC analysis. I calculated the RMSET and the MeanSD after each block of 500 sampled trees (50000 iterations). In this first calculation, blocks are additive, so that the statistics include a larger sample of trees after each block. The

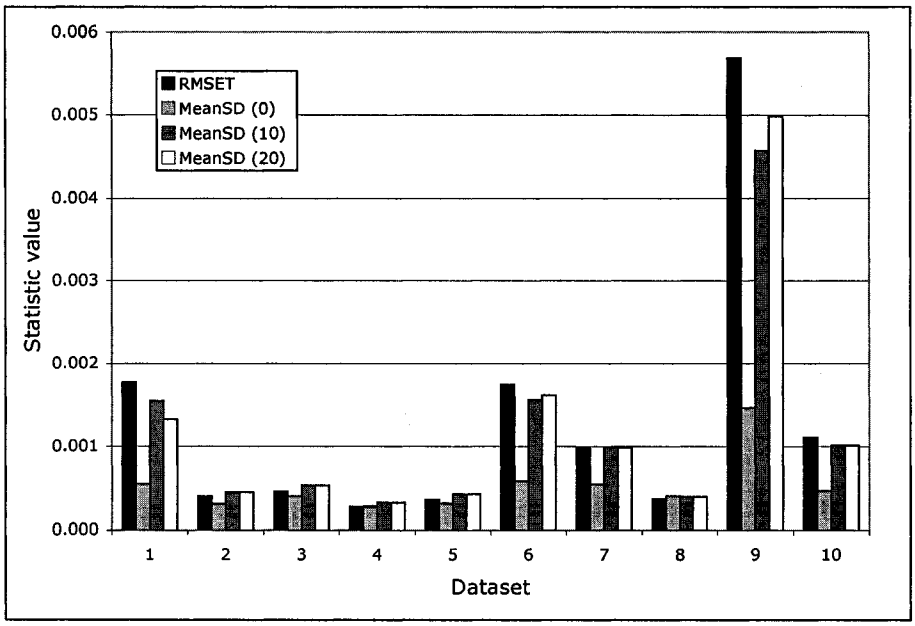


Figure 3.2: A comparison of topology-based convergence diagnostics. Each result is the final value recorded at the end of the MCMC. The value in brackets for the MeanSD result is the probability limit for inclusion of partitions.

plots in Figure 3.3 show the topology statistics for a selected number of the ten data sets - the three with the highest RMSET values (1, 6 and 9) as well as data set 2 (the lowest) for comparison. The plots show a large reduction in the topology variance at the start of the run and then a gradual reduction as the run progresses. The shape of the RMSET and the MeanSD are very similar. The rough shape of the plots for data sets 1, 6 and 9 is due to the chains moving between the different topologies present in the posterior distribution.

The RMSET is a measure of the variance between the distribution of topologies in the three chains. To help distinguish between the effect of increasing sample size and of the changing distributions of topologies, I calculated the RMSET using constant-sized batches of sampled trees. Figure 3.4 illustrates plots of this calculation for selected data sets. The batch size is 500 trees. In these plots, the initial transient is now very easy to identify and we can see that for most data sets it does not extend outside of the first batch. The decreasing RMSET appears to be primarily due to increasing sample size, since the RMSET in each batch does not, on average, decrease relative to the previous batch. Instead, we see much more random behaviour of the batched RMSET values. This technique allows us to visualize the amount of noise in the changing distribution of topologies.

Finally, I plot the mean of 1000 simulated values of the RMSET ($RMSET_S$) against the calculated values in Figure 3.5. As $RMSET_S$ is the expected value with

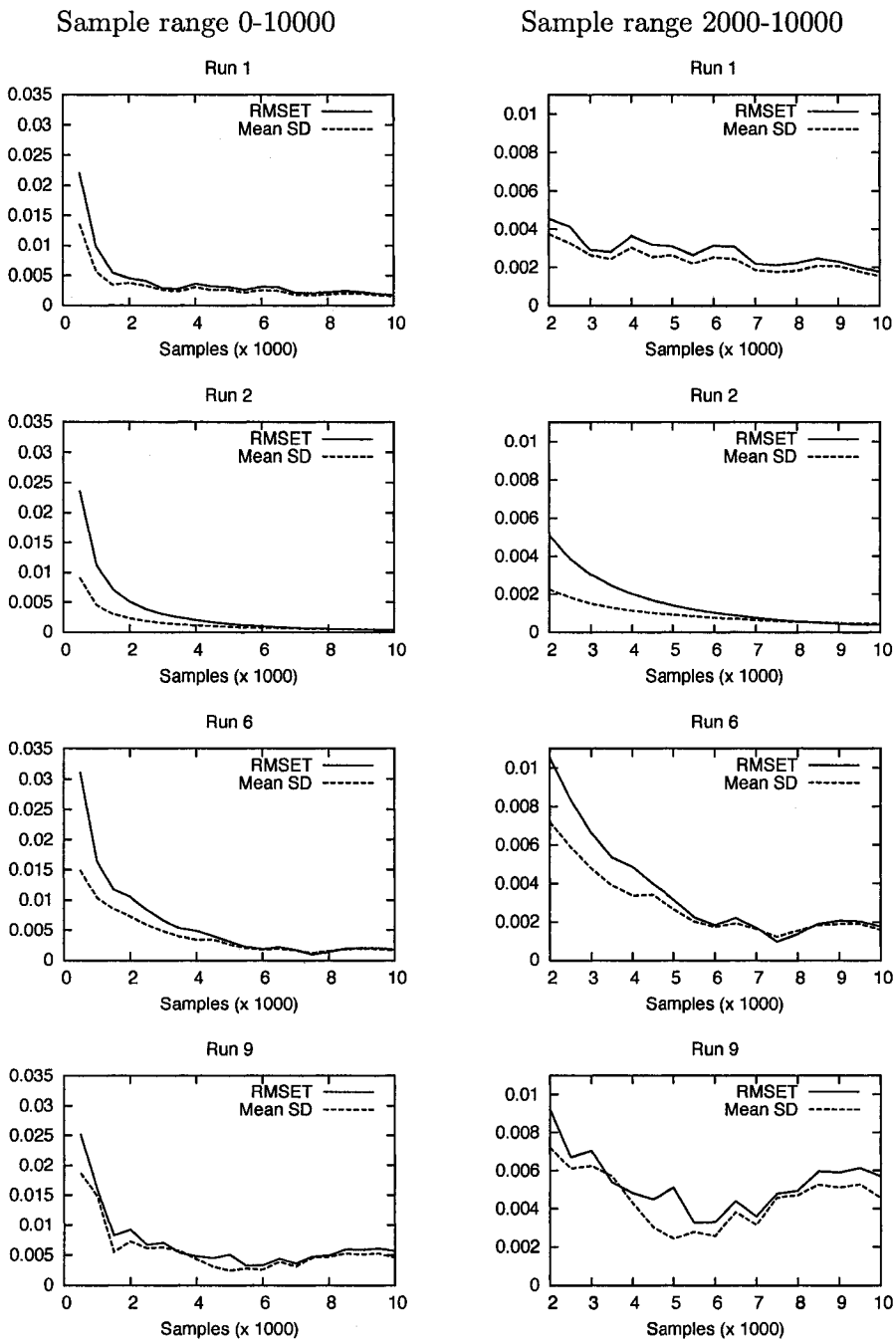


Figure 3.3: RMSET and MeanSD for selected 10-taxon data sets. The MeanSD is calculated using partitions with probability greater than 10%. The two plots in each row are the same analysis, differing only in the scale of the x-axis (in order to show additional detail). Scale of the y-axis scale in constant in each column.

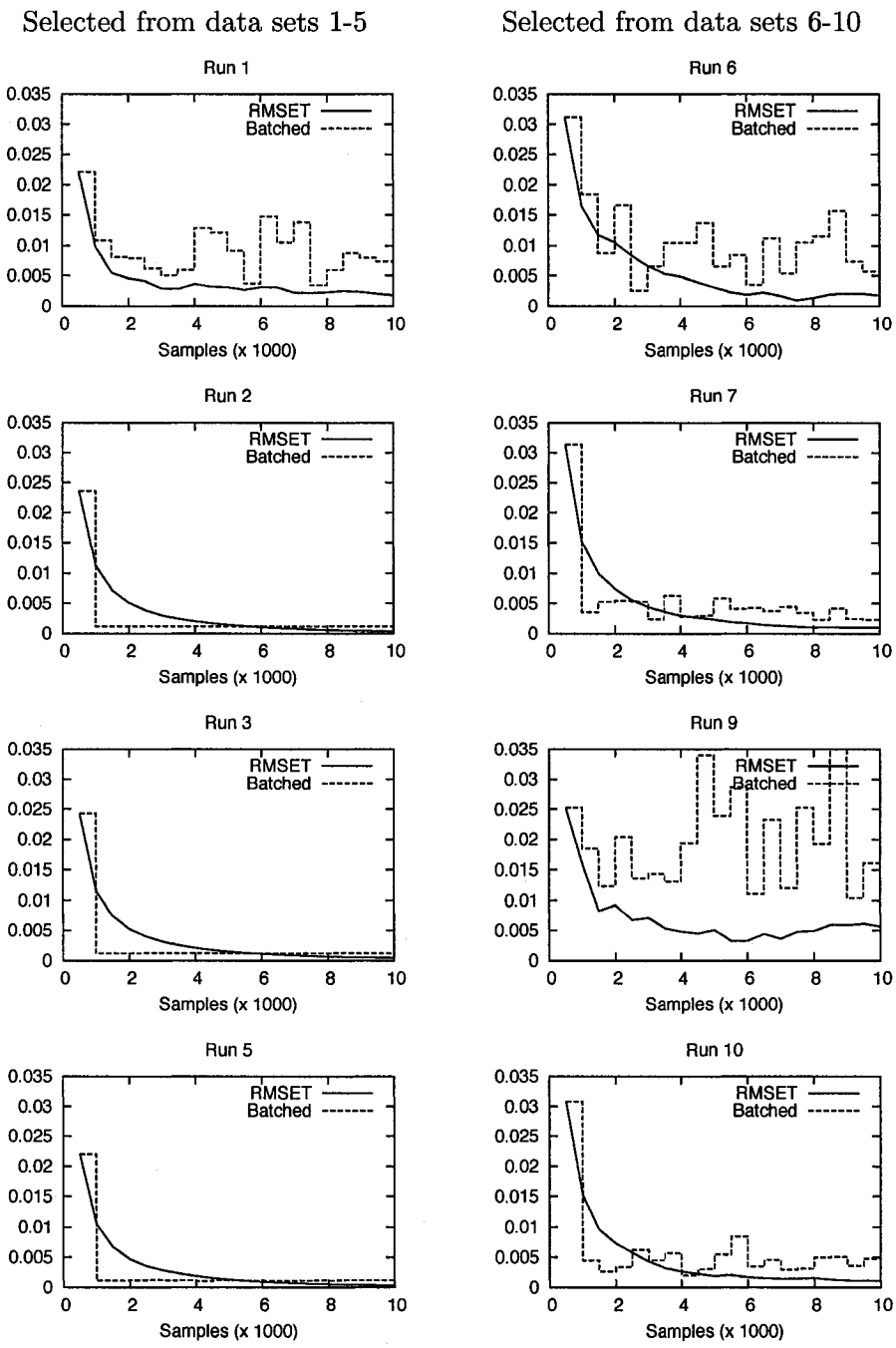


Figure 3.4: RMSET for the 10-taxon data sets. The statistic is calculated using constant batch size and increasing batch size. Left plots are selected from data sets 1 through 5, right plots from 6 through 10.

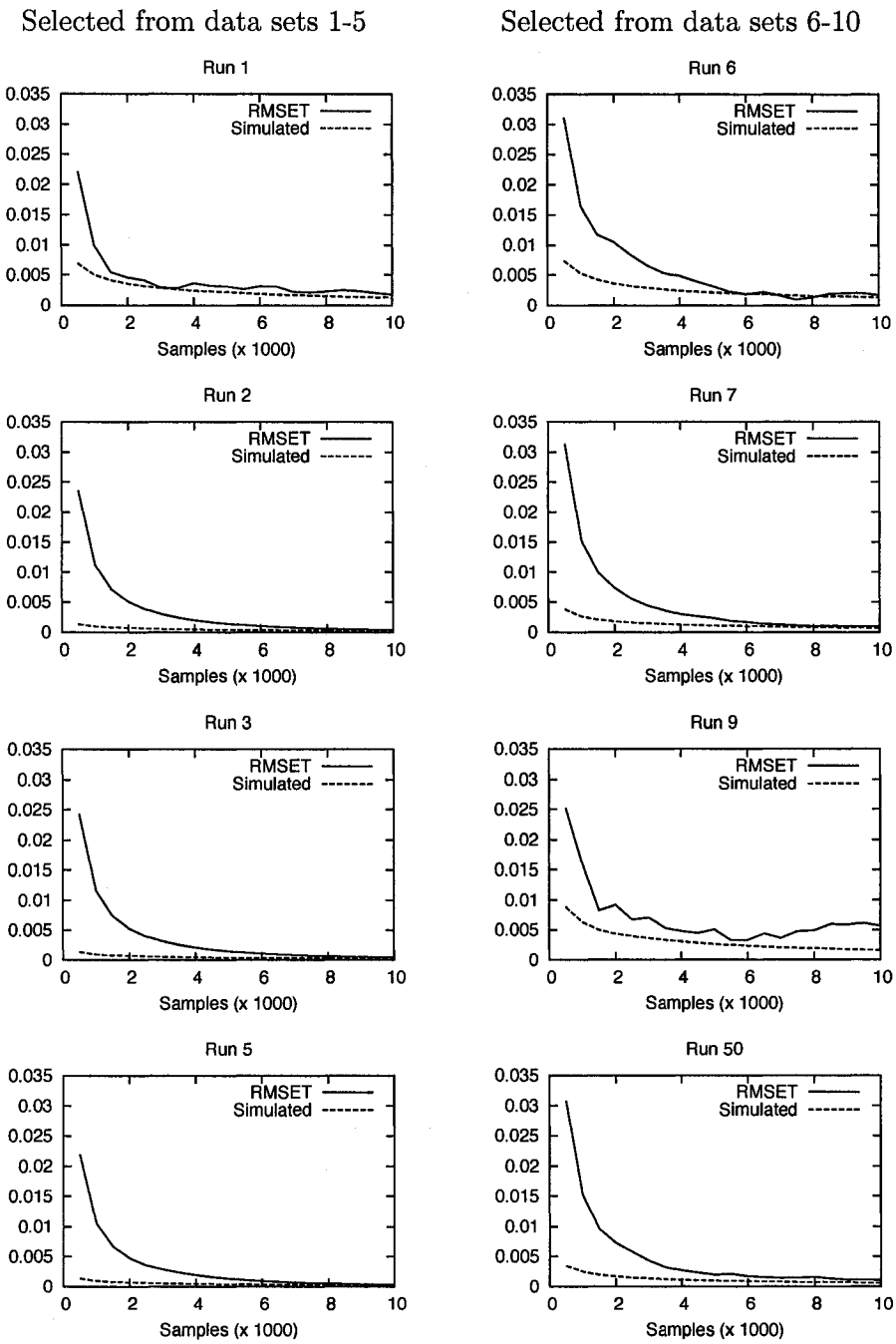


Figure 3.5: Comparison of simulated and calculated RMSET values for the 10-taxon data sets. Left plots are selected from data sets 1 through 5, right plots from 6 through 10.

Table 3.6: Simulated values of the RMSET ($RMSET_S$) for the 10 taxon trees, with bounds for 95% confidence interval. Also includes calculated value, $RMSET_C$, and the difference ($RMSET_C - RMSET_S$) for comparison. Rows are sorted from largest to smallest difference.

Data set	$RMSET_S$	LCI	UCI	$RMSET_C$	Difference
9	1.559E-03	1.521E-03	1.596E-03	2.936E-03	1.377E-03
6	1.275E-03	1.239E-03	1.312E-03	1.944E-03	6.691E-04
7	6.755E-04	6.613E-04	6.898E-04	9.334E-04	2.578E-04
10	6.209E-04	6.080E-04	6.339E-04	8.368E-04	2.158E-04
8	3.602E-04	3.502E-04	3.702E-04	5.540E-04	1.938E-04
1	1.248E-03	1.209E-03	1.287E-03	1.356E-03	1.079E-04
5	2.021E-04	1.976E-04	2.065E-04	2.826E-04	8.057E-05
4	1.899E-04	1.850E-04	1.948E-04	2.288E-04	3.887E-05
2	1.785E-04	1.741E-04	1.829E-04	2.137E-04	3.524E-05
3	2.119E-04	2.065E-04	2.173E-04	2.391E-04	2.716E-05

all chains sampling from the same distribution, the plot of this statistic illustrates the reduction in variance due to increasing sample size (without any of the time series structure from the MCMC analysis). The confidence intervals for the simulated statistic are extremely narrow, such that including the trace for the upper and lower limits simply makes the trace for the mean appear as a slightly thicker line. The narrow width of the confidence intervals is given in Table 3.6, which also compares the calculated and simulated values for the RMSET.

Summary

Application of these diagnostics to the 10 taxon trees illustrates their variable sensitivity. The PSRF and HW stationarity tests did not indicate any problems, nor did the time series plots. The Raftery and Lewis diagnostics indicate fairly high autocorrelation for some output parameters. The topology statistics are very small for all data sets, indicating good estimation of the partition probabilities (although they come to a stable value much later than the numerical output parameters). Despite the low RMSET and MeanSD values, it is still very easy to see differences between the data sets.

3.5.2 Analysis of 30 and 50 taxon trees

Summary of inference

The phylogenetic inference results are presented in Tables 3.7 and 3.8. As expected, the larger trees generally have smaller P_{MAP} values and larger credible sets of trees.

With the 50 taxon trees, there is great variability between the data sets. Compare data set 3 with $P_{MAP} \approx 0.97$ and a single tree in the 95% credible set with data sets 9 and 10, with $P_{MAP} < 0.05$ and hundreds of trees in the credible sets. Again, all chains appear very similar with respect to these measures.

Acceptance rates were between 17% and 22% for the 30 taxon trees, and between 12% and 20% for the 50 taxon trees.

The two tests for stationarity did not indicate any problems with the analyses. PSRF values were less than 1.01 for all analyses (which is slightly larger than the 1.00 limit for the 10 taxon trees), and the HW stationarity test passed in all cases.

Burn-in times

Table 3.9 gives the mean burn-in times, averaged over the four output parameters and three chains. The RL test estimates a slightly shorter burn-in than what is indicated by the time series plots, although this average result masks the fact that the results are extremely variable across parameters (but consistent across chains). Burn-in values for the tree length and branch score were consistently less than 10.0 for the 30 taxon trees (and less than 15 for the larger trees), while values for the log likelihood and γ statistic were nearly an order of magnitude higher. The HW estimates are expected to be higher due to the block-based nature of the test, but the fact that some chains had burn-in values greater than 500 samples (1 block) means that this test is the most conservative.

Autocorrelation

The dependence factors from the RL test for all 30 and 50 taxon data sets are summarized in Table 3.10. On average, the dependence factors increase with increasing size of the phylogeny, and while many factors are below the 5.0 upper limit, there is at least one parameter above this limit for each of the data sets. Similar to the burn-in estimates, there is also a significant difference between the different input parameters. The dependence factors for the log-likelihood are consistently higher, while γ is wildly variable (compare γ for the 50 taxon data sets 1 through 5 with data sets 6 through 10). In contrast to the 10 taxon trees, there is also noticeable variability between the chains for a given analysis. The between-chain variance is larger for larger values of the dependence factors and this trend is more significant for the 50 taxon trees than for the 30 taxon trees.

As the measures of autocorrelation from the RL test seem to vary significantly across chains and parameters, I also calculated the autocorrelation time (ACT) using Tracer. The measures are not expected to be identical, but trends should be the same. Table 3.11 compares the mean and variance across chains for the two

Table 3.7: Summary of phylogenetic inference for 30 taxon trees. For each simulated data set, I report the probability of the MAP tree, the total number of unique trees sampled, the size of the 90%, 95% and 99% credible sets for each of the three independent MCMC chains (all calculated after removal of the first 500 samples as burn-in). Data sets 1 through 5 are the data sets simulated with sampling frequency, ρ, equal 1.0 and 6 through 10 have ρ=0.01.

Data set	Chain	P_{MAP}	Unique trees	Credible sets
1	1	0.318444	22	(6, 8, 13)
1	2	0.288	28	(6, 8, 14)
1	3	0.303778	30	(6, 8, 13)
2	1	0.428222	10	(3, 3, 5)
2	2	0.409556	9	(3, 3, 6)
2	3	0.400222	11	(3, 3, 6)
3	1	0.327778	45	(12, 17, 29)
3	2	0.349778	40	(12, 16, 28)
3	3	0.312	41	(12, 17, 27)
4	1	0.620889	4	(3, 3, 3)
4	2	0.636444	5	(3, 3, 3)
4	3	0.606	3	(3, 3, 3)
5	1	0.536	9	(3, 3, 3)
5	2	0.526889	7	(3, 3, 3)
5	3	0.520889	9	(3, 3, 3)
6	1	0.170222	10	(8, 8, 9)
6	2	0.172667	11	(8, 8, 9)
6	3	0.180667	9	(8, 8, 9)
7	1	0.316444	35	(12, 16, 26)
7	2	0.308444	33	(13, 17, 26)
7	3	0.294889	33	(12, 15, 25)
8	1	0.347556	26	(6, 9, 17)
8	2	0.350444	25	(6, 9, 15)
8	3	0.370667	22	(6, 9, 16)
9	1	0.295778	27	(11, 14, 21)
9	2	0.274889	31	(11, 14, 22)
9	3	0.265556	30	(10, 14, 21)
10	1	0.370444	17	(5, 6, 9)
10	2	0.358	18	(5, 6, 11)
10	3	0.355333	21	(5, 6, 11)

Table 3.8: Summary of phylogenetic inference for 50 taxon trees. For each simulated data set, we report the probability of the MAP tree, the number of unique trees sampled and the size of the 90%, 95% and 99% credible sets for each of the three independent MCMC chains (all calculated after removal of the first 500 samples as burn-in). Data sets 1 through 5 are the data sets simulated with sampling frequency, ρ, equal 1.0 and 6 through 10 have ρ=0.01.

Data set	Chain	P_{MAP}	Unique trees	Credible sets
1	1	0.5967	8	(3, 3, 5)
1	2	0.5900	8	(3, 3, 5)
1	3	0.5489	9	(3, 4, 5)
2	1	0.3502	49	(10, 15, 27)
2	2	0.3660	48	(11, 16, 29)
2	3	0.4100	42	(10, 15, 28)
3	1	0.9764	5	(1, 1, 2)
3	2	0.9640	6	(1, 1, 3)
3	3	0.9727	5	(1, 1, 2)
4	1	0.0967	228	(86, 122, 185)
4	2	0.0984	197	(84, 110, 157)
4	3	0.0982	225	(83, 116, 180)
5	1	0.1271	217	(62, 98, 172)
5	2	0.1409	224	(63, 103, 179)
5	3	0.1687	220	(58, 97, 175)
6	1	0.4278	45	(11, 15, 27)
6	2	0.4453	53	(11, 16, 29)
6	3	0.4467	44	(11, 16, 26)
7	1	0.2240	159	(48, 70, 122)
7	2	0.2178	161	(50, 77, 125)
7	3	0.2071	161	(54, 80, 127)
8	1	0.4296	43	(4, 8, 22)
8	2	0.4411	31	(3, 7, 17)
8	3	0.3873	52	(7, 14, 31)
9	1	0.0393	472	(204, 289, 427)
9	2	0.0347	451	(203, 282, 406)
9	3	0.0324	498	(217, 308, 453)
10	1	0.0331	882	(516, 657, 837)
10	2	0.0431	822	(465, 597, 777)
10	3	0.0362	835	(483, 610, 790)

Table 3.9: Estimated burn-in from various diagnostics for 30 and 50 taxon trees. Results are in terms of samples, not iterations and are an average over the four parameters and three chains. For times series, estimates were rounded up to the nearest 10 samples (30 taxa) or 100 samples (50 taxa) before averaging.

Taxa	Data set	Raftery Lewis	Time Series	Heidelberger Welch
30	1	70.0	44.8	500.0
30	2	72.5	42.7	500.0
30	3	70.0	41.4	541.7
30	4	70.0	96.8	500.0
30	5	67.5	50.4	541.7
30	6	70.0	28.8	500.0
30	7	72.5	36.2	500.0
30	8	70.0	26.5	500.0
30	9	75.0	32.8	500.0
30	10	70.0	37.8	500.0
50	1	293.3	127.5	500.0
50	2	255.0	130.0	500.0
50	3	188.8	125.0	500.0
50	4	227.1	130.0	500.0
50	5	185.3	115.0	500.0
50	6	190.3	152.5	541.7
50	7	175.0	150.0	541.7
50	8	135.0	150.0	875.0
50	9	130.5	135.0	625.0
50	10	151.9	130.0	500.0

Table 3.10: The RL dependence factors for the 30 and 50 taxon trees. The value for each analysis is averaged over the three independent chains. This is a measure of autocorrelation.

Taxa	Data set	LogL	TreeLen	Distance	γ	Mean
30	1	19.3	2.2	2.0	15.1	9.6
30	2	25.6	1.8	2.1	17.5	11.7
30	3	21.8	2.2	2.1	19.3	11.4
30	4	14.3	2.7	1.6	21.2	9.9
30	5	19.7	1.7	1.7	19.8	10.7
	Mean	20.1	2.1	1.9	18.6	
30	6	20.4	2.7	2.2	3.2	7.1
30	7	23.1	3.5	2.7	3.4	8.2
30	8	19.6	2.8	2.3	5.4	7.5
30	9	25.1	3.5	2.4	2.9	8.5
30	10	24.2	2.7	2.3	2.8	8.0
	Mean	22.5	3.0	2.4	3.5	
50	1	19.7	2.7	3.3	45.7	17.9
50	2	21.7	2.2	4.1	57.1	21.3
50	3	29.0	2.6	2.8	80.9	28.8
50	4	29.1	2.6	3.9	28.9	16.1
50	5	47.2	2.7	3.6	42.6	24.0
	Mean	29.4	2.5	3.5	51.0	
50	6	20.1	3.9	3.7	6.0	8.4
50	7	39.3	3.5	3.2	6.2	13.0
50	8	35.4	4.0	3.4	6.8	12.4
50	9	38.0	4.0	3.5	6.1	12.9
50	10	17.4	4.0	2.4	7.8	7.9
	Mean	28.0	3.7	3.3	15.5	

Table 3.11: Comparison of RL dependence factors and autocorrelation time as a measure of autocorrelation within the MCMC chains. Mean and variance are calculated across the three independent chains for each data set.

			RL Dependence factors		Autocorrelation time	
Taxa	Data set	Parameter	Mean	Variance	Mean	Variance
30	1	LogL	19.32	9.02	3.36	0.05
30	1	TreeLen	2.21	0.13	2.10	0.05
30	1	Distance	1.95	0.03	2.55	0.01
30	1	γ	15.11	62.78	3.36	0.02
30	10	LogL	24.22	0.06	3.55	0.13
30	10	TreeLen	2.68	0.43	3.40	0.13
30	10	Distance	2.32	0.07	3.24	0.08
30	10	γ	2.75	0.10	5.06	0.53
50	1	LogL	19.70	1138.01	5.86	0.57
50	1	TreeLen	2.66	0.27	3.18	0.18
50	1	Distance	3.31	1.98	5.18	0.14
50	1	γ	45.74	1843.05	7.13	0.04
50	10	LogL	17.41	884.67	10.92	4.40
50	10	TreeLen	3.98	0.08	5.55	0.44
50	10	Distance	2.42	0.00	7.80	0.05
50	10	γ	7.77	32.52	16.43	3.38

autocorrelation measures for selected analyses. For both the RL dependence factors and the ACT, values are higher for log likelihood and γ statistic than for tree length and distance, particularly for the 50 taxon trees. Results for both statistics show that the variance between chains tends to increase with an increasing mean autocorrelation value. The variance values for the RL dependence factors are orders of magnitude greater than the ACT in the larger trees (see particularly the results for the log likelihood and γ statistic for the 50 taxon trees).

Topology-based measures

Calculated and simulated values of the topology-based measures, RMSET and MeanSD, are given in Tables 3.12 and 3.13. On average, the value of the statistics increases with increasing tree size. Again, confidence intervals for the simulated $RMSET_S$ are extremely narrow, such that no calculated value falls within the interval.

Figures 3.6 and 3.7 illustrates the different topology measures for four of the 30 taxon data sets. Analyses include calculated and simulated values of the RMSET and the changing P_{MAP} and credible set size. These data sets are converging on a

Table 3.12: Difference between simulated and calculated RMSET values for 30 taxon trees. Rows are sorted from largest to smallest difference.

Data set	$RMSET_S$	LCI	UCI	$RMSET_C$	Difference
9	0.00192	0.00188	0.00197	0.00561	0.00369
2	0.00141	0.00137	0.00145	0.00487	0.00347
3	0.00194	0.00190	0.00199	0.00538	0.00343
1	0.00176	0.00172	0.00180	0.00402	0.00226
6	0.00179	0.00174	0.00183	0.00363	0.00184
8	0.00184	0.00180	0.00188	0.00366	0.00182
4	0.00145	0.00141	0.00149	0.00306	0.00161
5	0.00143	0.00139	0.00147	0.00282	0.00139
10	0.00166	0.00162	0.00170	0.00278	0.00112
7	0.00201	0.00196	0.00206	0.00302	0.00101

Table 3.13: Difference between simulated and calculate RMSET values for 50 taxon trees. Rows are sorted from largest to smallest difference.

Data set	$RMSET_S$	LCI	UCI	$RMSET_C$	Difference
8	0.00182	0.00179	0.00186	0.00952	0.00770
10	0.00258	0.00255	0.00261	0.00872	0.00614
5	0.00204	0.00201	0.00206	0.00774	0.00570
7	0.00215	0.00212	0.00218	0.00679	0.00464
4	0.00219	0.00216	0.00222	0.00634	0.00415
2	0.00185	0.00182	0.00188	0.00587	0.00402
3	0.00137	0.00133	0.00140	0.00484	0.00347
6	0.00193	0.00189	0.00196	0.00512	0.00320
9	0.00238	0.00235	0.00241	0.00534	0.00296
1	0.00161	0.00158	0.00165	0.00431	0.00270

fairly stable distribution of topologies, seen by the stable sizes of the credible sets. There is still some variability between the chains, seen in the decreasing RMSET statistic and changing probability of the MAP tree. The initial batch of RMSET values are quite large, then the statistic drops off smoothly after the first 500 samples. The RMSET statistic approaches the simulated value at the end of the MCMC. I note that, while these plots appear much smoother than the 10-taxon plots, this is due to a very different y-axis scale.

Figures 3.5.2 and 3.5.2 illustrates the same set of topology diagnostic plots for four of the 50 taxa data sets - the two first and two last listed in Table 3.13. The first figure, data sets 1 and 8, show fairly narrow credible sets and relatively high P_{MAP} values. The high value of RMSET difference for data set 8 appears to be due to a change in the distribution of chain 3 near the end of the analysis - note the shift in credible set. The second sets of plots, data sets 9 and 10, contrast two analyses with very wide credible sets and low P_{MAP} values. The shape of the RMSET plot is not particularly sensitive to the increasing size of the credible sets in these analyses, indicating that these additional trees are not increasing the variation in the of the partition probabilities.

3.5.3 Analysis of treefrog phylogeny

To illustrate the ability of these convergence diagnostics to detect serious non-convergence, I examined the MCMC results from the empirical data set of Hylid frogs. In this analysis, the probability of the MAP tree is low (0.0030 in chain 1 and 0.0025 in chain 2) and the credible sets are large. The 95% credible sets contains approximately 2600 trees and the 99% credible sets are nearly identical to the total number of unique sampled trees. There is no discernible difference between the chains.

Numerical diagnostics for this analysis do indicate some problems with convergence. The PSRF values were 1.001 for the log-likelihood and 1.041 for the tree length. Neither result is considered a failure based on the upper limit of 1.20 for the test. Despite thus result, visual inspection of the time series plots indicates that the two sample chains for the tree length have visually different traces (Figure 3.10). The log-likelihood plot does not look unusual (not shown). This is the only data set examined in this study where the time series plot of one parameter is dramatically different than that of a different output parameter.

Results of the Raftery and Lewis test are shown in Table 3.14. This test does not indicate any significant difference between the two output parameters, although the results between the two chains do show a greater than two-fold difference with respect to the autocorrelation measure.

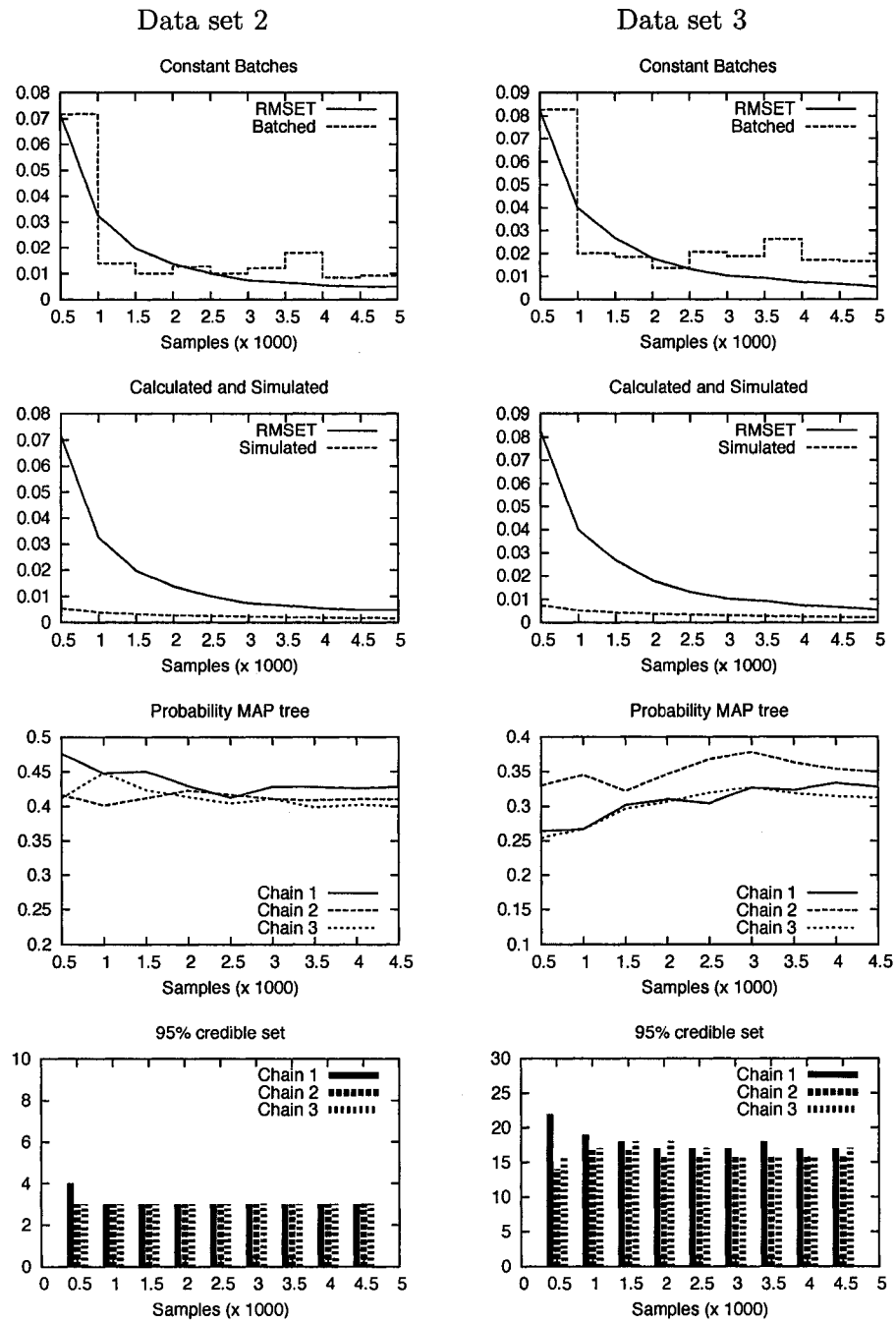


Figure 3.6: 30 taxon topology measures. Row 1: Calculated RMSET with increasing and constant-sized batches of samples. Row 2: Calculated and simulated RMSET. Row 3: Probability of the MAP tree over the course of the MCMC for the three chains. Row 4: Size of 95% credible set for the three chains.

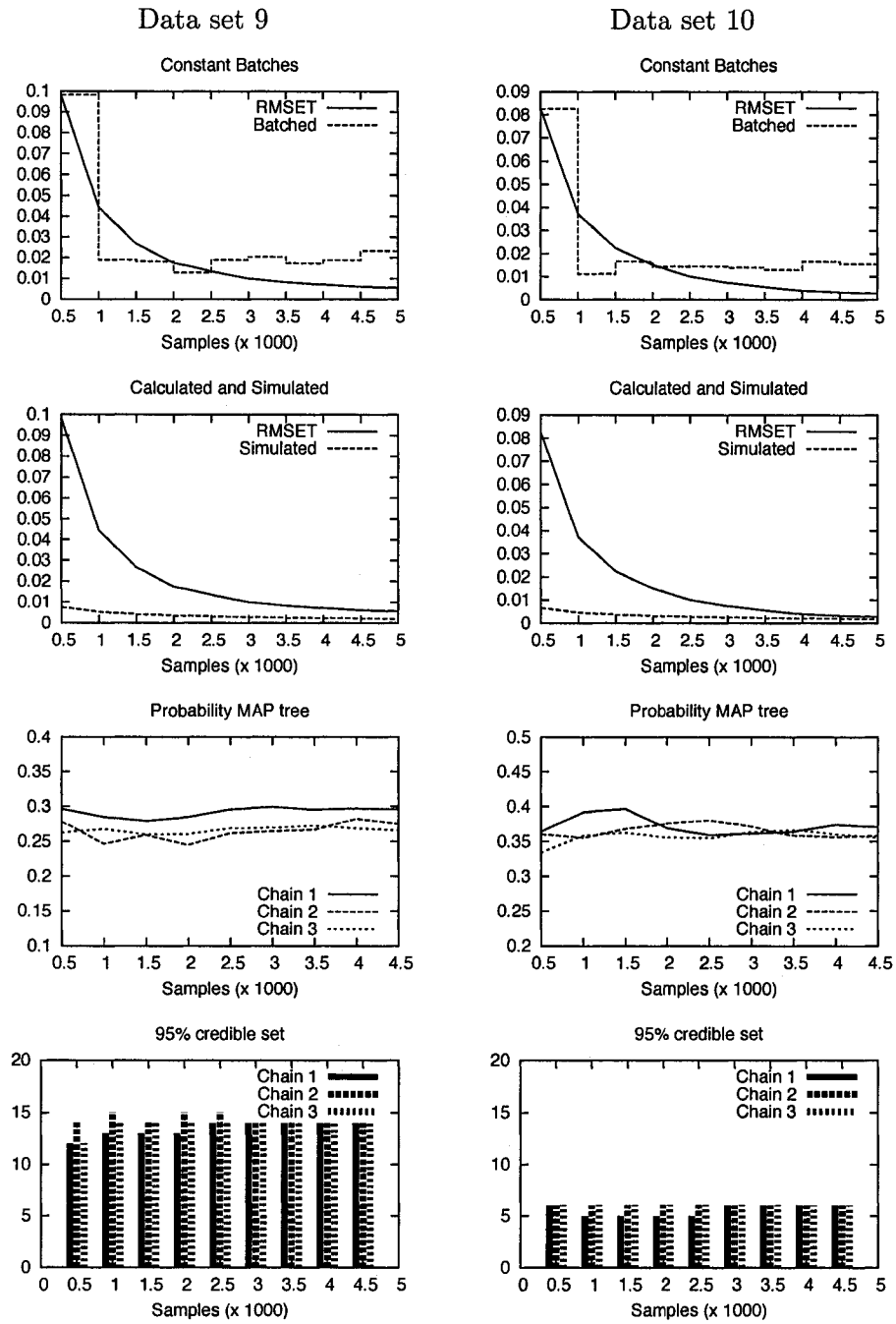


Figure 3.7: 30 taxon topology measures. Row 1: Calculated RMSET with increasing and constant-sized batches of samples. Row 2: Calculated and simulated RMSET. Row 3: Probability of the MAP tree over the course of the MCMC for the three chains. Row 4: Size of 95% credible set for the three chains.

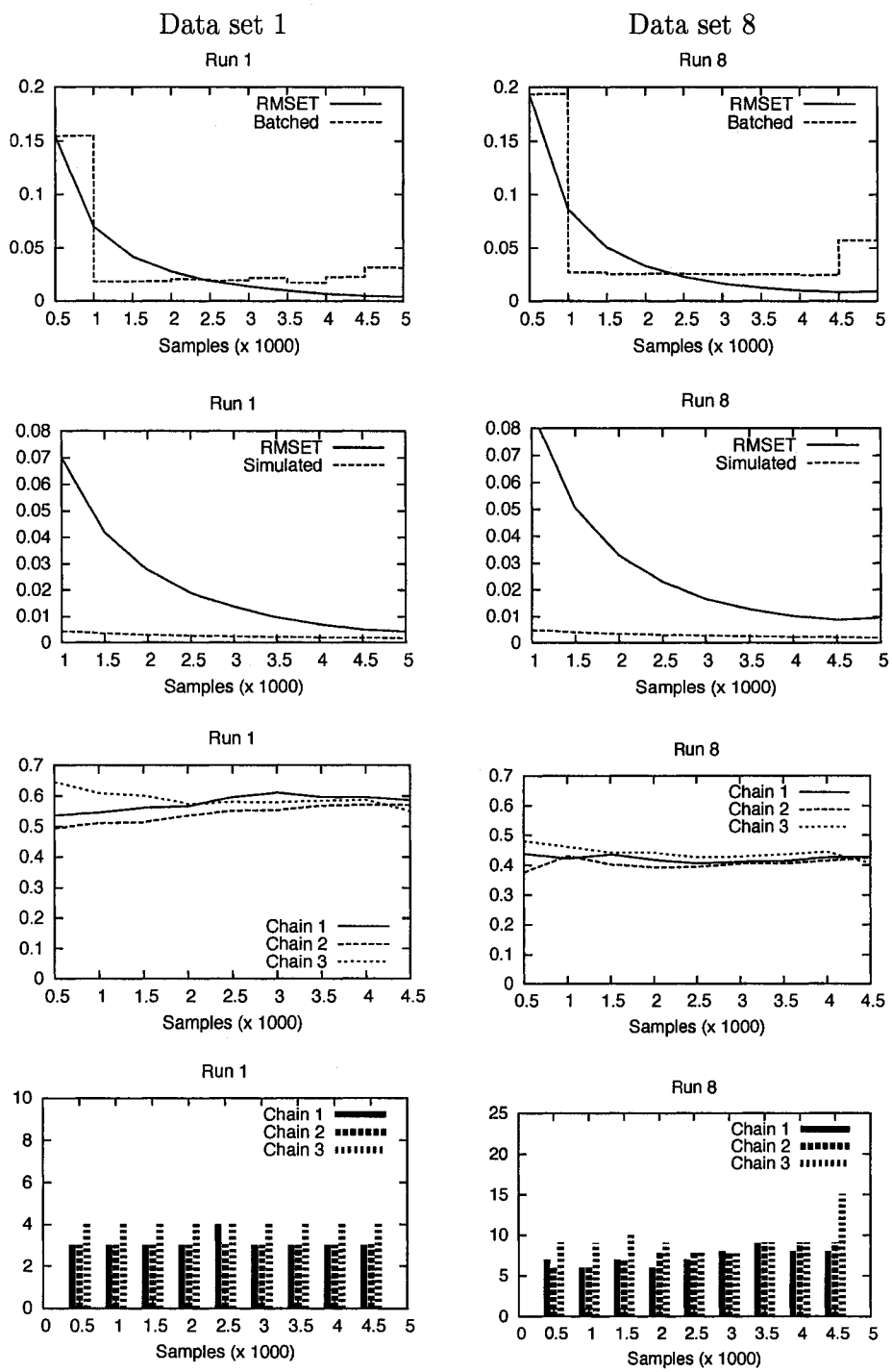


Figure 3.8: 50 taxon topology measures. Row 1: Calculated RMSET with increasing and constant-sized batches of samples. Row 2: Calculated and simulated RMSET. Note altered range on x-axis to show additional detail due to very high value at start of analysis. Row 3: Probability of the MAP tree over the course of the MCMC for the three chains. Row 4: Size (number of topologies) of the 95% credible set for the three chains.

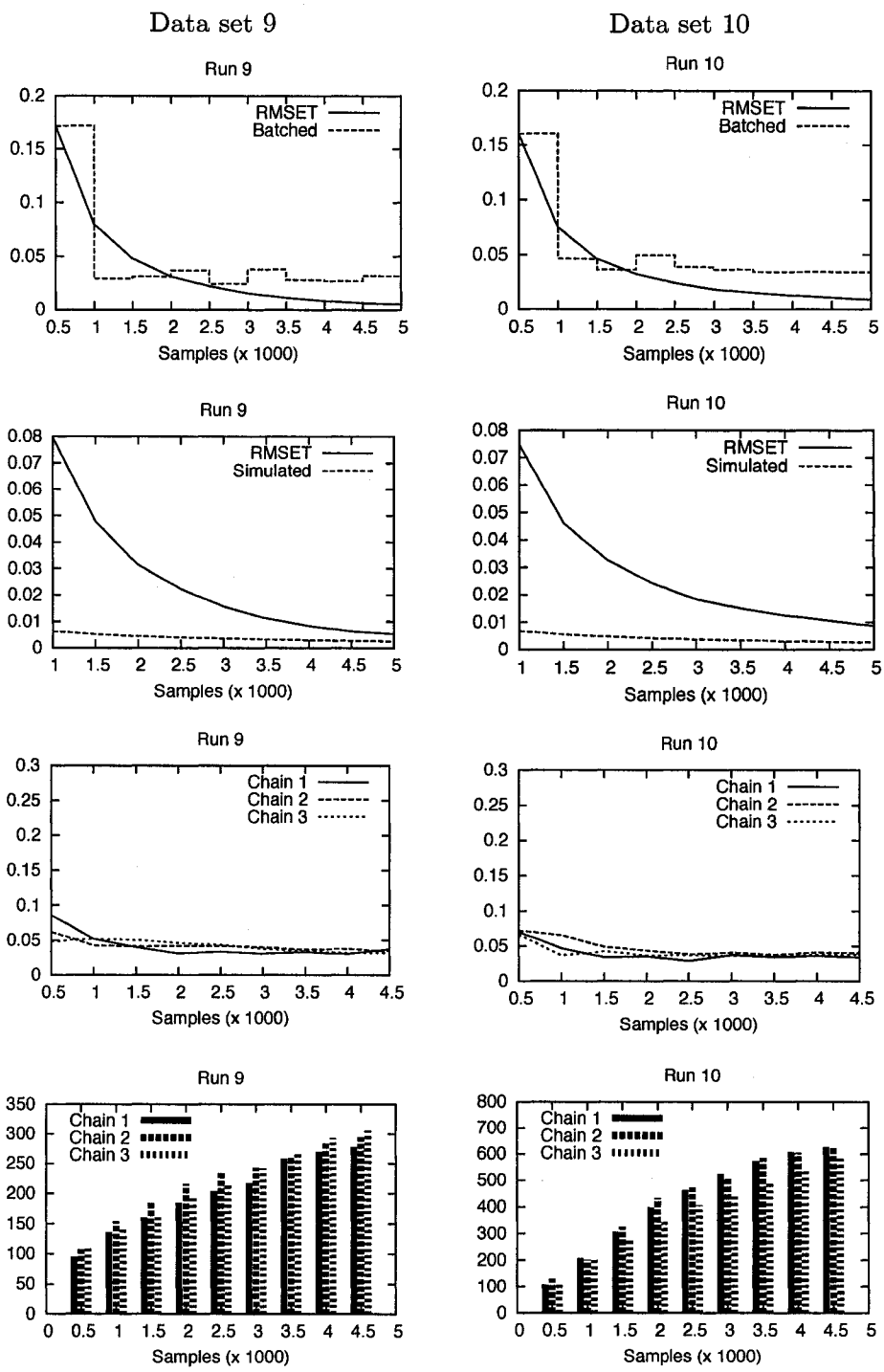


Figure 3.9: 50 taxon topology measures. Row 1: Calculated RMSET with increasing and constant-sized batches of samples. Row 2: Calculated and simulated RMSET. Row 3: Probability of the MAP tree over the course of the MCMC for the three chains. Row 4: Size (number of topologies) of the 95% credible set for the three chains.

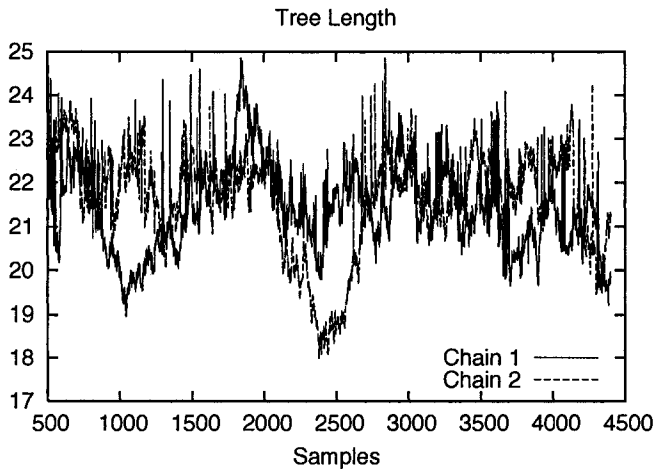


Figure 3.10: The tree length plot for the treefrog phylogeny. The first 500 samples are not shown, although inclusion of this burn-in period only increases the y-axis range by 3 units.

Table 3.14: Results of the RL test for the frog phylogeny.

Chain	Parameter	Thinning	Burn-in	Dependence factor
1	LnL	8	56	14.56
1	TreeLen	10	130	15.69
2	LnL	8	208	38.67
2	TreeLen	14	168	41.44

Table 3.15: Results of the HW test for the frog phylogeny.

Chain	Parameter	Stationarity test	Burn-in	Halfwidth Test
1	LnL	passed	440	passed
1	TreeLen	failed	2640	failed
2	LnL	passed	440	passed
2	TreeLen	failed	2640	failed

The HW test results are in Table 3.15. Again, the tree length samples are problematic, failing the stationarity test (and, by default, the halfwidth test). In contrast to the RL results, the HW test varies dramatically between the two output parameters but is identical for the two independent chains.

Figure 3.11 illustrates the topology measures and the shape of the distribution. The difference between the calculated and simulated value of the RMSET indicates that there are still large differences between the chains in terms of partition probabilities. This is confirmed by plots of the MAP tree probability and credible set size, which have not yet stabilized after 4.4 million iterations.

For this data set, the single chain diagnostics indicate a problem with convergence, but the multiple chain diagnostic does not. The differences between diagnostics illustrate the utility of both multiple diagnostics and of monitoring multiple output parameters.

3.6 Discussion

3.6.1 Identifying burn-in period

The analyses converged quickly to the stationary distribution, as judged by the numerical output parameters. This is consistent with previous expectations for general MCMC methods (Raftery and Lewis, 1992a) and for Bayesian phylogenetics (Beiko et al., 2006). Visual analysis of time series plots, which is the most common method for detecting the burn-in period, was generally consistent with the numerical diagnostics. For all of the data sets, the burn-in estimated from the log-likelihood plot did not differ from that estimated using plots of the other output parameters.

The Raftery and Lewis test underestimated the burn-in for the small trees (when autocorrelation is very low), but was very similar to the graphical estimates for the larger trees. The Heidelberger and Welch test seemed to overestimate burn-in. Given the short length of the burn-in period relative to the full length of the chain, an overestimate of burn-in is not critical (and much preferred to an underestimate). For studies involving a large number of analyses, numerical

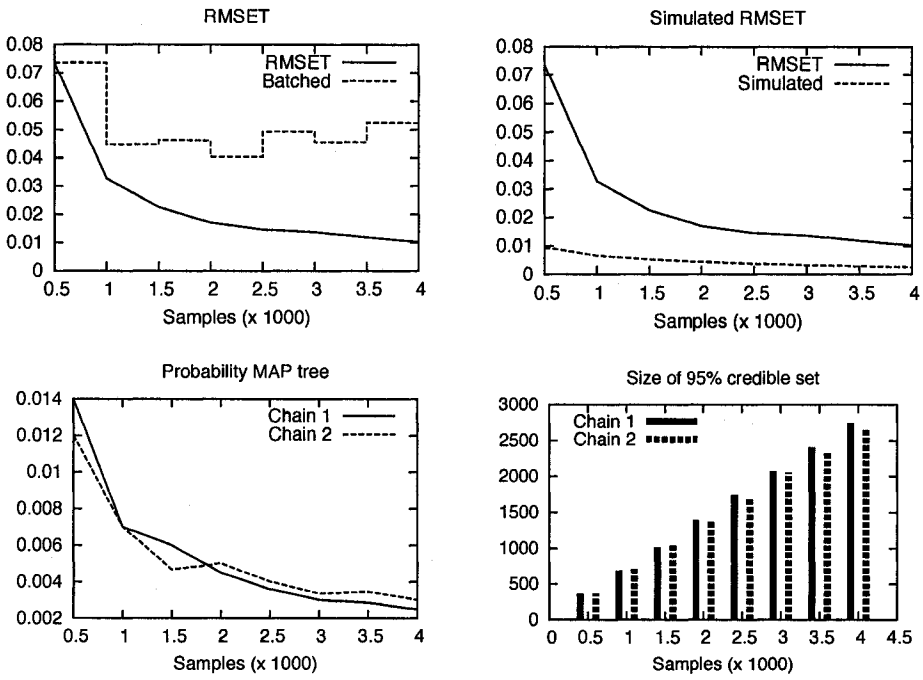


Figure 3.11: Topology measures for treefrog phylogeny. Top Left: Calculated RMSET with increasing and constant-sized batches of samples. Top Right: Calculated and simulated RMSET. Bottom left: Probability of the MAP tree over the course of the MCMC for the two chains. Bottom right: Size (number of topologies) of the 95% credible set for the two chains.

measures are a suitable alternative to visually inspecting each analysis. I also note that I used a mean burn-in over chains and parameters in the tables of values, but selecting the maximum value over all results would be a more conservative strategy.

I note that when judging the different diagnostics, it is important to consider the structure of the test. With the Raftery-Lewis test, calculating the number of burn-in iterations is based on properties of the thinned first-order Markov process, rather than being based on direct examination of the sampled states. With HW, the burn-in is dependent on the stationarity of the samples collected at the end of the MCMC. Although the HW test results indicate higher burn-in values than RL or time series plots, this is partially due to batching of results. If the size of the HW batch is 500, then a burn-in value of 1000 is consistent with a value between 500 and 1000 from the other tests.

For the simulated data sets, none of the tests for stationarity (time series plots, HW stationarity test and the PSRF statistic) indicated any problems, although some of these diagnostics were able to detect convergence problems in the empirical data set. The differences between the results for different diagnostics illustrates the need for multiple tests for convergence.

3.6.2 Autocorrelation and mixing

While convergence to the stationary distribution was not a problem for the simulated data sets, the analysis of mixing behaviour was more interesting.

Autocorrelation is a measure of how well the chain is mixing. One measure of autocorrelation are the dependence factors from the Raftery and Lewis test, which are based on the the subsampling that would be required to have the chain behave as a first-order Markov process. Figure 3.12 shows the dependence factors with increasing tree size for the three tree sizes and four output parameters. Both the mean and the variance of the dependence factors increases with tree size. A certain amount of autocorrelation is to be expected in phylogenetic MCMC analysis, given the topology parameter. Most proposals of new states modify only a portion of the tree, so that output parameters based on the topology will be affected in the location of the topology or branch length change but not in the remaining parts of the tree. This effect will be greater with increasing numbers of taxa, so that more sparse MCMC sampling may be required for analyses of larger phylogenies, along with a longer total chain length.

There was great variability in the RL dependence factors across parameters and chains, particularly for the larger trees when autocorrelation was high. For some of the 50 taxon trees, the dependence factors were so large (and variable across chains) that this test seemed unreliable. Due to these results for the RL test,

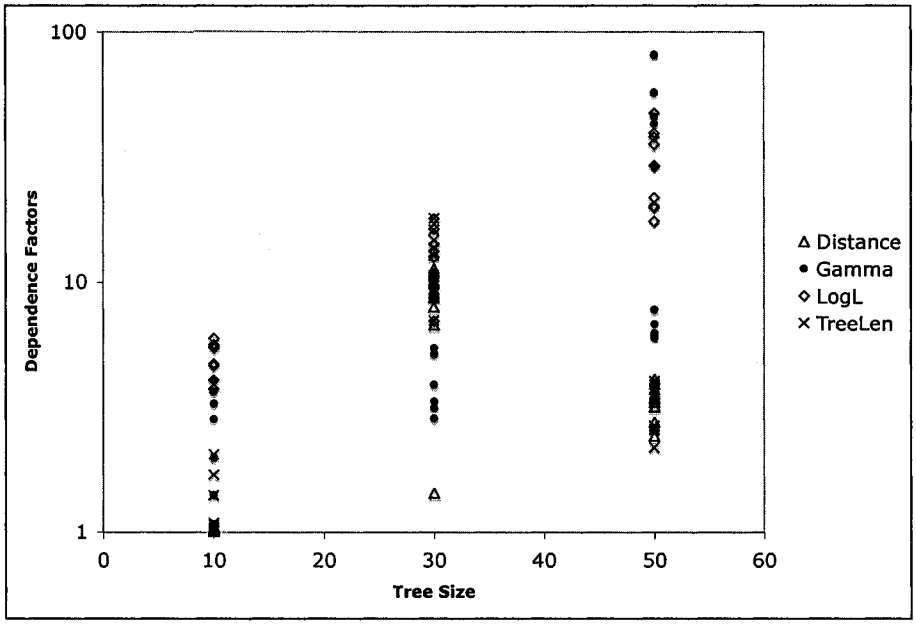


Figure 3.12: Increasing autocorrelation with tree size (number of taxa), as measured by the Raftery and Lewis dependence factors. Note that y-axis is logarithmic.

autocorrelation was also examined using the autocorrelation time (ACT), which is a sum of the autocorrelation function over various lag intervals. The trends were the same with both measures, although the ACT did not show the huge variation between chains that appeared in the RL dependence factors for the larger trees. The Raftery and Lewis test is based upon a first-order Markov chain constructed from the original sampled states, and this construction may be more problematic when autocorrelation is very high. With these data sets, some autocorrelation values were an order of magnitude greater than the upper limit defined by the test authors as cause for concern.

The larger trees had greater autocorrelation, and results for the log likelihood and γ statistic were greater than that for the Branch Score (distance) and tree length. The result for tree size (number taxa) is not unexpected. We only modify a portion of the tree topology and branch lengths at each iteration, and the fraction of the tree left unchanged will tend to be larger as the number of taxa increases, leading to increased autocorrelation in measures based on the tree topology.

3.6.3 Utility of topology-based measures

The final set of diagnostic were those based on the partition probabilities. What are these measures telling us? As a starting point, their decreasing value even after

other parameters have stabilized indicates that the standard output parameters of an MCMC analysis are not very sensitive to changes in the tree topology. The RMSET comes to a stable value only in the smallest trees examined. For the larger trees, the RMSET continues to decline right to the end of the analysis, despite the stability of all other output parameters within the first five thousand iterations.

The topology-based measures give an estimate of the total variance of partition probabilities over the course of the MCMC analysis. A higher value is correlated with greater uncertainty about the distribution of topologies and other parameters. A lower value for the RMSET (or MeanSD) indicates less variability in the results, but a higher value does not indicate that the analysis has not converged. A less-informative data set may have a broad credible set as the true distribution of phylogenies, and in this case, the RMSET would be relatively large at convergence. For this reason, comparison of simple RMSET magnitudes across different data sets should not be used to rank the convergence of the analyses. A better way of thinking about these diagnostics would be that a lower RMSET value for the same data set analyzed under a different set of MCMC parameters would indicate an improvement in the phylogenetic inference with respect to the distribution of trees.

Simulating the distribution of the RMSET provides a more objective measure for a single analysis. As the independent chains converge on the same distribution, the calculated value approaches the simulated value. This does not prove that we are sampling from the true posterior distribution, but having multiple chains with the same distribution increases our confidence that we are close to the true distribution. The difference between the calculated and simulated values of the RMSET can then be used as a comparison across different data sets.

Comparison of the RMSET or MeanSD with the changing probability of the MAP tree and the size of the credible sets allows us to see that stability of partition probabilities does not indicate that we have converged to a stable posterior distribution of topologies. Both statistics can have a very small value and be decreasing even as the credible set of topologies increases in size. This is a positive result, as is indicates that our estimate of the phylogeny can be very good even without convergence to the posterior distribution of phylogenies.

Finally, a note about the differences between the RMSET and the MeanSD. The two diagnostics can be close in value with the correct choice of probability limit for the MeanSD. If this limit is set too low, the result for the MeanSD can be artificially low value due to the inclusion of many low probability partitions (a false positive result). The RMSET measure does not depend on a user-defined limit, which makes it more robust. However, it does not lend itself to calculation as the analysis progresses (due to the comparison with the overall mean across chains for the entire

MCMC analysis).

3.6.4 Conclusions

To diagnose convergence, we can monitor numerical output parameters, partition probabilities and the posterior distribution of trees. The rate of convergence of the three types of output is in this same order, and the posterior distribution of phylogenies may not converge in a satisfactory number of iterations. Numerical parameters can be monitored for stationarity and mixing using the diagnostics examine in this study. Stationarity for the various output parameters, however, does not indicate convergence of the distribution of topologies or of the partition probabilities.

What are the goals of Bayesian phylogenetic methods? Generally, when using MCMC we hope that the parameters of interest converge to their stationary distribution. Therefore, convergence of the posterior probability distribution of trees seems to be the obvious answer to this question. Unfortunately, with data sets of reasonable size, this is an unreasonable hope. There are examples of analyses that show convergence to a distribution of phylogenies for relatively small numbers of taxa (Rokas et al., 2003; Pagel et al., 2004), but most studies are satisfied with convergence of the numerical parameters. The failure of Bayesian phylogenetic methods to converge to a stable distribution of phylogenies has been recognized previously (Hillis et al., 2005; Beiko et al., 2006). This is due to the sheer size of the tree space and, for many data sets, the large number of reasonably good trees. Therefore, we look to convergence of the partition probabilities and use these as a method of summarizing the information in the distribution of phylogenies.

As has been stated in many previous studies of convergence diagnosis, there is no simple answer and there will never be a single test or statistic that can reliably signal convergence of an MCMC chain to the stationary distribution. I propose a three stage process for checking output from phylogenetic inference:

1. Identify the burn-in stage and eliminate samples collected during this phase of the MCMC. Visual inspection of the time series plots seems to work as well as any of the other methods tested, although numerical tests are more efficient for large number of analyses.
2. Test for stationarity using output parameters such as the log likelihood and tree length. Again, time series plots can be of assistance here, as can numerical methods such as the PSRF and the HW test. How does the distribution of topologies and the probability of the MAP tree change over the course of the MCMC?
3. Examine mixing behaviour. Calculating autocorrelation values (and again,

examination of time series) can identify poorly mixing chains. If the chain appears stable but mixing is poor, then we need more samples or we need to adjust proposal algorithms.

4. Verify precise estimation of the partition probabilities. Multiple chains should give similar estimates of the partition probabilities, verified with low values of the RMSET or MeanSD. The values of the statistics should decrease over the course of the MCMC, but may not reach a stable value.

I note the repetition about utility of time series plots in the preceding list. As stated in a classic text on time series analysis, "the first, and most important, step in any time-series analysis is to plot the observations against time" (Chatfield, 1989). Plotting against the full scale of x and y-axis can identify the burn-in period, and then a change in scale can detect more subtle problems with stationarity or mixing.

It is possible to have truly objective criteria for diagnosing convergence? Stability of the MCMC chains is the general aim, and the definition of "stable" is far from clear. Numerical diganostics such as the RL, HW and PSRF tests have tried to address this issue by defining an additional criteria, such as determining the number of samples required to calculate a given statistic to a given level of accuracy. This is, of course, assuming that we are interested in the value of that particular statistic. With phylogenetics, it is unclear what statistic and what level of accuracy we should use. An accurate value of the tree length does not necessarily indicate that the current number of topology samples are sufficient for calculation of partition probabiliites. For the partition-based MeanSD, MrBayes suggests a fixed cut-off of 0.10. The value of topology-based diagnostics like the MeanSD and RMSET are dependent on the particular data set. The same value for different data sets, while quantifying the variance in the same way, is not guaranteed to be diagnostic for convergence of the analyses.

The choice of convergence diagnostics will depend on the scope of individual study. When performing a single analysis (such as the case of a systematist inferring the phylogeny of a single data set) the optimal diagnostics may be different than a study that involves hundreds of data sets. The former user may prefer graphical methods, while the latter study is more amenable to numerical methods, which are more easily automated and summarized.

The output from even a relatively simple Bayesian inference of a single gene tree will have a topology parameter, plus 2s-2 (or 2s-3) branch lengths, 2-6 model parameters of the rate matrix and parameters describing rate variability. The advent of partitioned analyses (Nylander et al., 2004) can dramatically increase this count, multiplying the number of parameters mentioned above by the number of partitions (genes and codon positions). Parameter output files for this type of

analysis can have hundreds of columns, making close inspection quite unrealistic.

Software tools such as BOA and CODA and the phylogenetic-specific Tracer and AWTY are useful for the analysis of convergence in Bayesian phylogenetic MCMC methods. Most of these methods require the user to manually input each individual MCMC chain into the software in order to perform diagnostics (I modified the BOA code in order to run in batch mode, but this is a non-standard application). There is need, though, for additional methods with an increased level of automation in order to deal with analyses with large numbers of output parameters or with studies that involve a large number of data sets. This will involve a trade-off between efficiency of the output analysis and loss of information from detailed examination of time series plots and other graphical methods.

Bibliography

Beiko, R. G., J. M. Keith, T. J. Harlow, and M. A. Ragan. 2006. Searching for convergence in phylogenetic Markov chain Monte Carlo. Syst. Biol. 55:553–565.

Best, N. G., M. K. Cowles, and S. K. Vines. 1995. CODA: Convergence diagnosis and output analysis software for Gibbs sampling output, version 0.3. MRC Biostatistics Unit, Cambridge.

Brandley, M. C., A. Schmitz, and T. W. Reeder. 2005. Partitioned Bayesian analyses, partition choice, and the phylogenetic relationships of scincid lizards. Syst. Biol. 54:373–390.

Brooks, S. P. and A. Gelman. 1998. General methods for monitoring convergence of iterative simulations. J Comput Graph Stat 7:434–455.

Brooks, S. P. and G. O. Roberts. 1998. Convergence assessment techniques for Markov chain Monte Carlo. Stat. Comput. 8:319–335.

Castoe, T. A. and C. L. Parkinson. 2006. Bayesian mixed models and the phylogeny of pitvipers (Viperidae: Serpentes). Mol. Phylogenet. Evol. 39:91–110.

Chatfield, C. 1989. The Analysis of Time Series: An Introduction. Fourth ed. Chapman and Hall.

Cowles, M. 1996. Markov chain Monte Carlo convergence diagnostics: A comparative review. J Am Stat Assoc 91:883–904.

Drummond, A. J., G. K. Nicholls, A. G. Rodrigo, and W. Solomon. 2002. Estimating mutation parameters, population history and genealogy simultaneously from temporally spaced sequence data. Genetics 161:1307–20.

Gelman, A. and D. Rubin. 1992. Inference from iterative simulation using multiple sequences. Stat. Sci. 7:457–72.

Geyer, C. 1992. Practical Markov Chain Monte Carlo. Stat. Sci. 7:473–483.

Heidelberger, P. and P. Welch. 1981. A spectral method for confidence interval generation and run length control in simulations. Commun. ACM 24:233–245.

Heidelberger, P. and P. D. Welch. 1983. Simulation run length control in the presence of an initial transient. Oper Res 31:1109–1144.

Hillis, D. M., T. A. Heath, and K. S. John. 2005. Analysis and visualization of tree space. Syst. Biol. 54:471–482.

Huelsenbeck, J. P. and J. P. Bollback. 2001. Empirical and hierarchical Bayesian estimation of ancestral states. Syst. Biol. 50:351–366.

Kuhner, M. K. and J. Felsenstein. 1994. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. Mol. Biol. Evol. 11:459–68.

Nee, S., R. M. May, and P. H. Harvey. 1994. The reconstructed evolutionary process. Philos Trans R Soc Lond B Biol Sci 344:305–311.

Nylander, J. A. A., F. Ronquist, J. P. Huelsenbeck, and J. L. Nieves-Aldrey. 2004. Bayesian phylogenetic analysis of combined data. Syst. Biol. 53:47–67.

Pagel, M., A. Meade, and D. Barker. 2004. Bayesian estimation of ancestral character states on phylogenies. Syst. Biol. 53:673–684.

Pybus, O. G., A. Rambaut, E. C. Holmes, and P. H. Harvey. 2002. New inferences from tree shape: numbers of missing taxa and population growth rates. Syst. Biol. 51:881–888.

Raftery, A. and S. Lewis. 1992a. How many iterations in the Gibbs sampler? Pages 763–773 *in* Bayesian Statistics 4 (J. Bernardo, J. Berger, A. Dawid, and A. Smith, eds.). Oxford University Press, Oxford.

Raftery, A. and S. Lewis. 1992b. [Practical Markov Chain Monte Carlo]: Comment: One long run with diagnostics: Implementation strategies for Markov Chain Monte Carlo. Statictical Science 7:493–497.

Rambaut, A. and A. Drummond. 2005. Tracer verson 1.3.

Rokas, A., B. L. Williams, N. King, and S. B. Carroll. 2003. Genome-scale approaches to resolving incongruence in molecular phylogenies. Nature 425:798–804.

Rosenthal, J. 2002. Quantitative convergence rates of Markov chains: A simple account. Elec. Comm. Prob. 7:123–128.

Rosenthal, J. S. 1995. Convergence rates for Markov chains. SIAM Review 37:387–405.

Smith, B. J. 2005. Bayesian Output Analysis program (BOA), version 1.1.5. http://www.public-health.uiowa.edu/boa.

Wiens, J. J., J. W. Fetzner, C. L. Parkinson, and T. W. Reeder. 2005. Hylid frog phylogeny and sampling strategies for speciose clades. Syst. Biol. 54:778–807.

Wilgenbusch, J., D. Warren, and D. Swofford. 2004. AWTY: A system for graphical exploration of MCMC convergence in Bayesian phylogenetic inference. http://ceb.csit.fsu.edu/awty.

Yang, Z. 1997. PAML: a program package for phylogenetic analysis by maximum likelihood. Comput Appl Biosci 13:555–6.

Zwickl, D. and M. Holder. 2004. Model parameterization, prior distributions, and the general time-reversible model in Bayesian phylogenetics. Syst. Biol. 53:877–888.

Chapter 4

Selection and tuning of tree proposal algorithms

4.1 Introduction

The algorithms used to propose new states in an MCMC analysis can greatly influence the rate of convergence and quality of mixing. In Bayesian phylogenetic MCMC methods, the choice of tree proposal algorithms is likely to have a large effect on convergence to the posterior distribution of phylogenies. Some tree proposal algorithms only propose states close to the current state by perturbing a small region of the tree (local moves), while others can propose much more drastic changes that affect multiple regions in the tree (global moves). Methods with tuning parameters allow adjustment of the move size, and powerful tuning parameters can allow a single algorithm to function both as a local and global method.

There are several ways to choose the tuning parameters for an MCMC analysis. We can simply select a parameter (or set of parameters) and perform the analysis, knowing that the choice may not be optimal for the data set. We can perform several exploratory runs to determine the best tuning parameters before starting a full analysis, or we can develop automated methods for choosing the parameters. The choice of method is generally based on the acceptance rate of the chain. If the acceptance rates are too large, then the proposals should be smaller (and larger when acceptance rates are too high).

One of the difficulties with automated methods of choosing algorithms and tuning parameters is the fact that the underlying process is a Markov chain. The proposal mechanisms define the transition kernel, or the probability of moving between states in the chain. A Markov chain requires that the proposal of a given state depends only on the immediately previous state and not on the earlier history

of the chain. If we change the transition kernel based on the samples we have already collected, the process is no longer a Markov chain because the samples are dependent on the earlier history. This means that functions based on the sampled states are not guaranteed to be consistent (Gilks et al., 1998).

It is possible to change the transition kernel during sample collection, but only in a way that does not depend on the collected samples. We can insert a smaller or larger move (by altering a tuning parameter, changing the type of move or by swapping chains in MCMCMC), but the choice of proposal strategy must be either fixed before we start the analysis or determined in a stochastic fashion during the analysis.

Another strategy is to alter the proposals during the burn-in stage using the existing sampled states. After a specified number of iterations, or once the acceptance rate is within a given range, we then stop updating the kernel and start collecting samples (Gelfand and Sahu, 1994). Adjusting the transition kernel using the sampled states is valid during the burn-in stage (when we do not retain the samples for inference about the posterior distribution). The kernel remains fixed during the sampling phase. In the context of Bayesian phylogenetics, BAMBE (Larget and Simon, 1999) used this strategy, but MrBayes does not.

The choice of algorithm and of tuning parameter is an area that deserves further study. How should we choose between the tree proposal algorithms described in Chapter 2 and what are the optimal tuning parameters for a given algorithm? At the start of an MCMC analysis, we want a method that converges quickly to the stationary distribution. This will likely involve large proposals. However, the method that gives fastest convergence may not provide optimal mixing once we are at stationarity. An even more fundamental question is how to compare the performance of different algorithms or tuning parameters. The effect of a slow rate of convergence is straightforward, causing a longer burn-in period. Slow convergence is inefficient, as we are required to discard a larger fraction of the total samples. The effect of poor mixing in Bayesian phylogenetic inference is less understood. How does poor mixing affect our estimates of the tree topology? How do we diagnose poor mixing? How robust are Bayesian phylogenetic MCMC methods to changes in tuning parameters?

There were two goals in this study. The first was to investigate the effect of algorithm choice and tuning parameters on MCMC convergence and mixing. This was accomplished by analyzing a number of data sets under a variety of algorithms and tuning parameter and applying convergence diagnostics discussed in Chapter 3 to the MCMC output. The second goal was to further test the convergence diagnostics against a wider range of MCMC conditions than those seen in Chapter

Table 4.1: Taxa and characters in data sets used in this chapter. Runs 1, 3, 6 and 8 are simulated data sets, and the remaining are empirical.

Data	Taxa	Characters
Runs 1,3,6,8	30	5000
Baldwin	35	452
Winkworth	35	610
McCracken	39	1116
Yuan	42	1167
101_SC	101	1858

3. I also briefly explore the idea of dynamically modifying tuning parameters during burn-in, but this is a difficult problem that warrants further study.

4.2 Methods

4.2.1 Data

This study used both simulated and empirical data. The simulated data is taken from Chapter 3 and includes four 30 taxon data sets, two simulated under the low sampling frequency and two under a high sampling frequency.

Empirical data poses a more challenging inference problem than simulated data. Research on fast maximum likelihood (ML) methods has begun to compile and test benchmark data sets (Stamatakis et al., 2005; Hordijk and Gascuel, 2005). In this chapter, I use several data sets tested in these papers, including a large 101 taxon fungal tree and three smaller data sets from the Hordijk study on SPR moves (Yuan et al., 2005; McCracken and Sorenson, 2005; Winkworth et al., 2005), noting that the Winkworth paper describes two different data sets - I have included both. Details of the data sets are given in Table 4.1. The large fungal tree (data set 101_SC) is noted to be a difficult convergence problem, with several rogue taxa that are not strongly supported in a single location on the tree. Using data sets that have previously been used in method testing allows for verification of results and comparisons between methods.

The combination of simulated and empirical data allows for the comparison of algorithms and tuning parameters under a variety of inference conditions.

4.2.2 Bayesian inference

The simulated data sets were inferred as described in Chapter 3. For the empirical data sets, inference was under the HKY model (Hasegawa et al., 1985). This model

was used for two reasons. First, it is the same model used in the ML studies that also used these data sets. This allowed comparison of likelihoods (to ensure that BayesTrees had not simply converged to a non-optimal mode). Second, I wanted to focus on integration of the tree topology and branch lengths and eliminate the issue of convergence of the model parameters.

The range of algorithms and tuning parameters for each data set is described in Table 4.2.

For the Local and SPR algorithms, the tuning parameter affects the multiplier proposal used to propose new branch lengths. Given a current value, x, and a tuning parameter, b, the new length, x' is generated using Larget and Simon's multiplier proposal $x' = x \times \exp^{2ln(b)\times(0.5-U)}$ (Larget and Simon, 1999). I describe the tuning parameter in terms of b rather than $\lambda = 2ln(b)$ because it is conceptually simpler: b is the largest possible multiplier (i.e. if $U = 1.0$, then $x' = bx$). The smallest possible multipler, when $U = 0$, is $1/b$.

For BranchSlide, the tuning parameter is the variance of the Normal distribution used to propose distances between the old and new location of moved subtrees. With this algorithm, a much larger range of tuning parameters is available than for Local and SPR.

I performed the Bayesian analysis using BayesTrees, setting kappa to the mean value estimated from a million iterations of a single MCMC chain and estimating the base frequencies from the data. The model parameters were identical for all runs for a given data set. The prior on branch length was exponential ($\lambda = 10, \mu = 0.1$) with branches unconstrained by the molecular clock. The simulated analyses were run for 50000 iterations, and the empirical for 1000000 iterations. Sampling was every 100. Some analyses were extended to a larger number of iterations (details in Results).

During the MCMC, I collected the log likelihood and tree length parameters. The gamma statistic used in Chapter 3 assumes that the trees are ultrameric, which is not true for the empirical data sets. The distance measure (BranchScore) requires a use of a reference tree for comparison, and while the true tree is available for simulated data, this is not the case for the empirical data sets (it would be possible, although computationally expensive, to compute the Branch Score against all of the previously sampled trees). The results from Chapter 3 indicate that all four of these output parameters give similar results for burn-in analysis, and for mixing, the log likelihood and gamma statistic were similar (with higher autocorrelation), as were the distance and tree length.

Table 4.2: Algorithms and tuning parameters used for phylogenetic inference. The small empirical data sets are the Baldwin, McCracken, Winkworth and Yuan data sets, while 101_SC is the large benchmark data set. Each analysis consisted of 3 independent MCMC chains.

Data	Algorithm	Analyses	Tuning
simulated	BranchSlide	4	0.005, 0.01, 0.02, 0.05
small empirical	BranchSlide	5	0.005, 0.01, 0.02, 0.05, 0.1
	Local	2	1.1, 2.0
	SPR	2	1.1, 2.0
101_SC	BranchSlide	4	0.01, 0.1, 0.5, 1.0, 2.0
	Local	2	1.1, 2.0
	SPR	2	1.1, 2.0

4.2.3 Convergence analysis

As a starting point, I examined time series plots of log likelihood and tree length for departures from stationarity and to determine the burn-in period. Results from this visual analysis were confirmed using the PSRF and HW tests.

For runs that had reached stationarity, I then check mixing using autocorrelation of the tree length and log likelihood and acceptance rates for the topology proposals. Autocorrelation was measured by the sum of lag autocorrelations (ACT statistic implemented in Tracer (Rambaut and Drummond, 2005)), which gave more consistent results than the RL autocorrelation estimates in Chapter 3.

I assessed convergence of the posterior distribution of phylogenies using the RMSET statistic and the size of the credible sets. These topology measures were compared to the stationarity and autocorrelation results for the log likelihood and tree length to determine the effect of rate of convergence and mixing on the inference of the posterior distribution of phylogenies.

4.3 Results

4.3.1 Rate of convergence

Not surprisingly, there were no problems with reaching stationarity for the simulated data sets. The acceptance rates were negatively correlated with the tuning parameter, and runs with higher acceptance rates had longer burn-in periods (due to smaller proposals). Burn in periods and acceptance rates were very consistent across the runs for the same tuning parameter. Table 4.3 lists the results. The numerical diagnostics (PSRF and HW stationarity test) indicated a pass for all runs.

For the small empirical data sets, examination of the time series plots after one

Table 4.3: Average burn in iterations and acceptance rates for simulated data sets under various tuning parameters of the BranchSlide algorithm. Burn-in determined from visual inspected of time series plots.

Tuning	Burn in	Acceptance rate
0.005	12250	0.35
0.01	6250	0.21
0.02	5750	0.11
0.05	2750	0.05

million iterations indicated some problems with convergence. The Local algorithm was insufficient for convergence for almost all of the data sets. Although most chains had stabilized, comparison of independent chains showed that many of chains were stuck in local optima. Application of the SPR algorithm alone also caused poor convergence, but the problem was instead a very slow rate of convergence. Most chains had not reached a stable value by the end of the analysis, although the likelihood continued to rise smoothly, without becoming stuck in a local optima. Convergence of the BranchSlide runs depended on the tuning parameter, although most runs converged quickly.

All of these results were confirmed using the HW stationary test and the PSRF. Any analysis that displayed a problem with the time series plots also failed the numerical convergence diagnostic tests. With the single-chain HW stationarity test, two of the runs with multiple modes visible on the time series plot passed the stationarity test, while the multiple chain PSRF diagnostic was able to detect this problem. The PSRF also diagnosed problems in some runs that appeared to have converged with respect to the time series plots. These contradictory results were generally in cases with large move parameters and small acceptance rates.

Table 4.4 gives the number of burn in iterations, acceptance rates and PSRF results for the various algorithms and tuning parameters for the small empirical trees. As expected, acceptance rates decreased with increasing move sizes (larger tuning parameters).

For runs that had converged, the log likelihood was close to that described in the fast ML study (Hordijk and Gascuel, 2005). We don't expect exact agreement due to differences in model parameters (the exact values of κ and base frequencies used for the HKY model is not specified in their study), as well as the fact that the Bayesian method produces a posterior distribution of likelihood values rather seeking the single best likelihood score.

Table 4.4: Acceptance rate and burn in times for the initial 1 million MCMC iterations for each data set. Burn in times are estimated from time series plots of log likelihood and tree length. Algorithms are in the first row (BS = BranchSlide), with tuning parameters below. Results are in terms of iterations, not samples. MM = multiple modes and NC = not converged. An 'X' as a result for the PSRF indicates a value above the 1.2 upper limit.

		BS	BS	BS	BS	BS	Local	Local	SPR	SPR
Data set	Parameter	0.005	0.01	0.02	0.05	0.1	1.1	2.0	1.1	2.0
Baldwin	LogL	7000	5000	5000	4000	6000	MM	200000	NC	70000
Baldwin	TreeLen	7000	5000	5000	4000	6000	MM	200000	NC	70000
Baldwin	Acceptance Rate	0.58	0.45	0.35	0.25	0.17	0.57	0.44	0.05	0.05
Baldwin	PSRF						X	X	X	X
McCracken	LogL	9000	12000	5000	5000	5000	MM	150000	NC	150000
McCracken	TreeLen	MM	150000	180000	180000	40000	MM	MM	NC	100000
McCracken	Acceptance Rate	0.42	0.31	0.23	0.14	0.09	0.45	0.37	0.04	0.03
McCracken	PSRF	X	X	X	X	X	X	X	X	X
Winkworth	LogL	6000	4000	4000	4000	4000	MM	MM	NC	50000
Winkworth	TreeLen	6000	4000	4000	4000	4000	MM	MM	NC	50000
Winkworth	Acceptance Rate	0.46	0.34	0.23	0.15	0.10	0.4	0.3	0.05	0.05
Winkworth	PSRF							X	X	X
Yuan	LogL	16000	5000	5000	7000	8000	MM	MM	NC	80000
Yuan	TreeLen	8000	5000	6000	7000	12000	MM	MM	NC	150000
Yuan	Acceptance Rate	0.30	0.20	0.13	0.07	0.05	0.27	0.15	0.01	0.009
Yuan	PSRF					X	X	X	X	X

Table 4.5: Results for stationarity tests of the 101_SC data set. Each run was started from the a final tree of the initial analysis with BranchSlide tuning parameter 2.0 (which appeared to have converged, based on time series analysis). The HW results indicate passed ('P') or failed ('F') for each of the three chains.

Algorithm	Tuning	PSRF	HW logL	HW treelen
BS	0.01	1.023	(P, P, P)	(P, P, P)
BS	0.1	1.089	(P, P, P)	(P, P, P)
BS	0.5	1.671	(P, P, P)	(P, F, P)
BS	1.0	1.982	(F, F, F)	(P, P, P)
BS	2.0	4.075	(F, F, F)	(F, F, F)
Local	1.1	1.229	(P, P, P)	(P, P, P)
Local	2.0	1.133	(P, P, P)	(P, F, P)
SPR	1.1	2.408	(F, F, F)	(F, F, F)
SPR	2.0	2.050	(F, F, F)	(F, F, F)

The general result was that Local and SPR runs were not yet converged but that most of the BranchSlide runs appeared to be at stationarity (the exception was the set of McCracken runs, which seemed to have problems reaching stationarity for all MCMC proposals, particularly with respect to the tree length). To ensure sufficient iterations to study mixing properties, I extended the BranchSlide runs for an additional 0.5 million iterations. For these extended analyses, two of the McCracken data sets still failed the PSRF stationarity test, while the HW test and times series plots did not detect any problems with any of the runs.

For the 101 taxon tree, reaching stationarity was difficult. Figure 4.1 displays the log likelihood plots for several different algorithms and tuning parameters. The only run that converged quickly used BranchSlide with a tuning parameter of 2.0 (the tree length is approximately 12). Neither of the Local runs converged. One of the SPR runs did appear to reach stationarity at the end of the run, but the rate of convergence was extremely slow. Figure 4.1 also includes plots from the same data set run using MrBayes with two different MCMCMC temperature parameters. Even using MCMCMC, there was a lack of convergence across all three independent chains.

As many of the original 101_SC runs did not converge, I repeated each one using the algorithms and tuning parameters specified in *Methods*, but starting from the last state of one of the runs that did appear to reach stationarity (from the time series plots). This provided a constant number of iterations for each algorithm and tuning parameter. Testing of these repeated analysis indicated that the analyses were not at stationarity. Time series plots are given in Figure 4.2. Results for the PRSF and HW stationarity tests are given in Table 4.5.

One point to emphasize here is the utility of multiple chains for diagnosing convergence. Without the other chains as a comparison, some of these traces would appear converged, particularly in a shorter window of iterations. This was also true for the small empirical data sets (although time series plots of these runs are not shown). Comparison of the numerical diagnostics also illustrate this point. All chains of an analysis can pass the HW stationarity test individually but fail when combined together and analyzed with the PSRF test.

4.3.2 Autocorrelation and mixing

Figure 4.3 shows the ACT of the log likelihood against tuning parameter and acceptance rate for the four smaller empirical data sets. Autocorrelation declines with decreasing move size (and higher acceptance rates). The stacked points in the upper graph (corresponding to the set tuning parameters) shift in the second graph when we substitute acceptance rates for tuning parameters. An identical tuning parameter operates differently on each data set, giving a different acceptance rate (although the trend is the same). Plots of the tree length autocorrelation display the same trend as the log likelihood. These results confirm that, for the BranchSlide algorithm, we should be able to set tuning parameters and control autocorrelation through the acceptance rate, but show that there will be some variation between data sets.

Results for the Local and SPR algorithms are not included on this plot. Autocorrelation for those methods was extremely high (higher than any of the BranchSlide runs) with the exception of Local with a tuning parameter of 2.0, which had autocorrelation values lower than any of the BranchSlide runs. The SPR moves had acceptance rates that were approximately one-half that of the largest BranchSlide tuning parameter (see Table 4.4). Acceptance rates for Local, however, were in the same range as the smaller BranchSlide tuning parameters, so acceptance rate alone does not explain differences in autocorrelation values.

We expect that autocorrelation would be a problem with moves that are too large and those that are too small. For BranchSlide, none of the tuning parameters tested were small enough to show an increase in autocorrelation at the lower end. The Local algorithm did display this phenomena, with the small tuning parameter giving very high autocorrelation, although the acceptance rates were in the same range as the small BranchSlide moves.

For the simulated data sets, autocorrelation again decreases with increasing acceptance rates, although the range of autocorrelation values is smaller than for the empirical data (despite a similar number of taxa). Figure 4.4 shows autocorrelation for both output parameters in simulated and empirical data. With

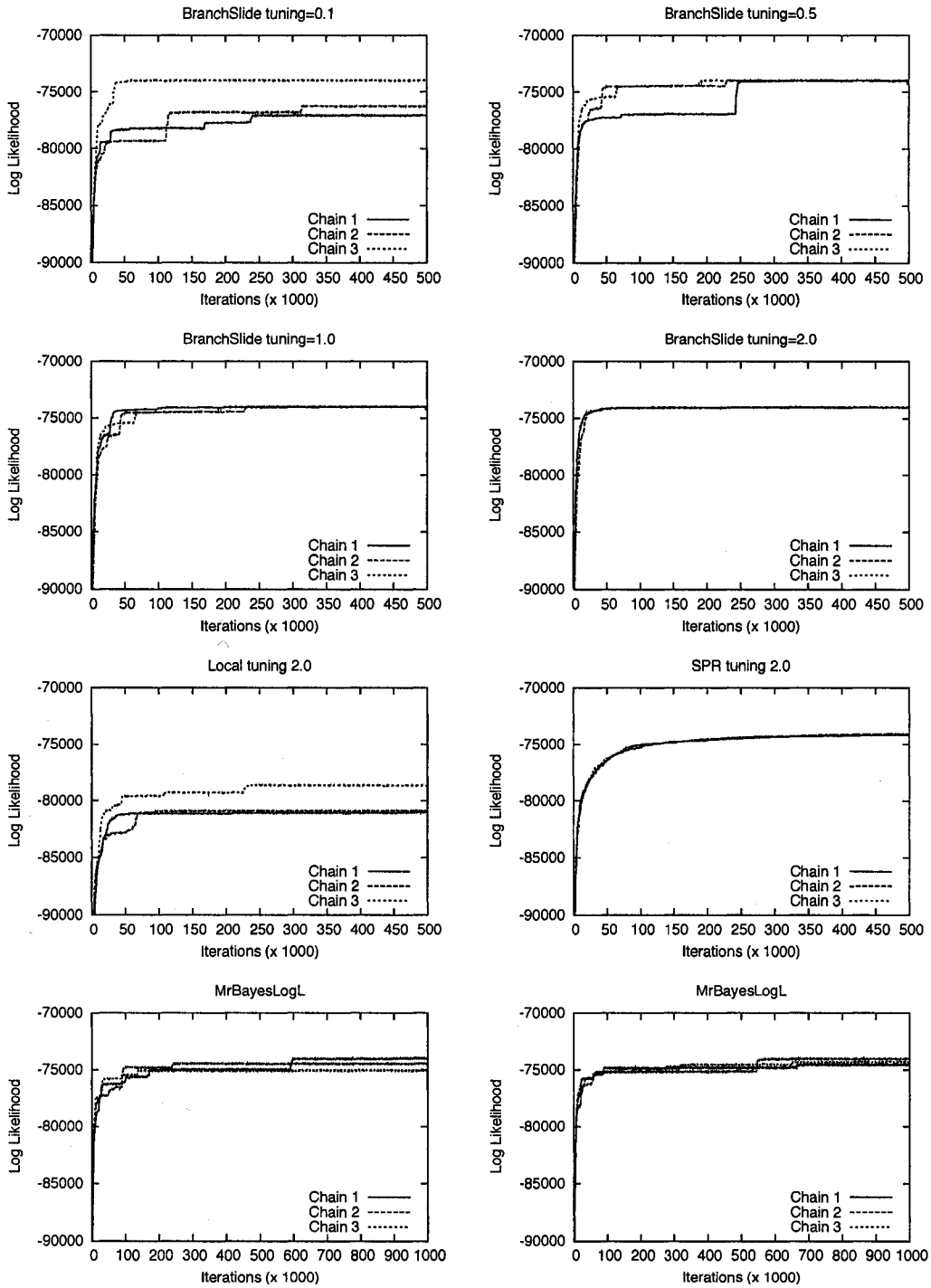


Figure 4.1: Convergence of the 101_SC data set when analyzed using different algorithms and tuning parameters. The top four plots are different tuning parameters for the BranchSlide algorithm. The third row is two runs using Local and SPR with tuning=2.0 and the bottom plots are three independent MCMCMC chains from MrBayes with two temperature (T) values (left is T=0.2, right is T=0.05).

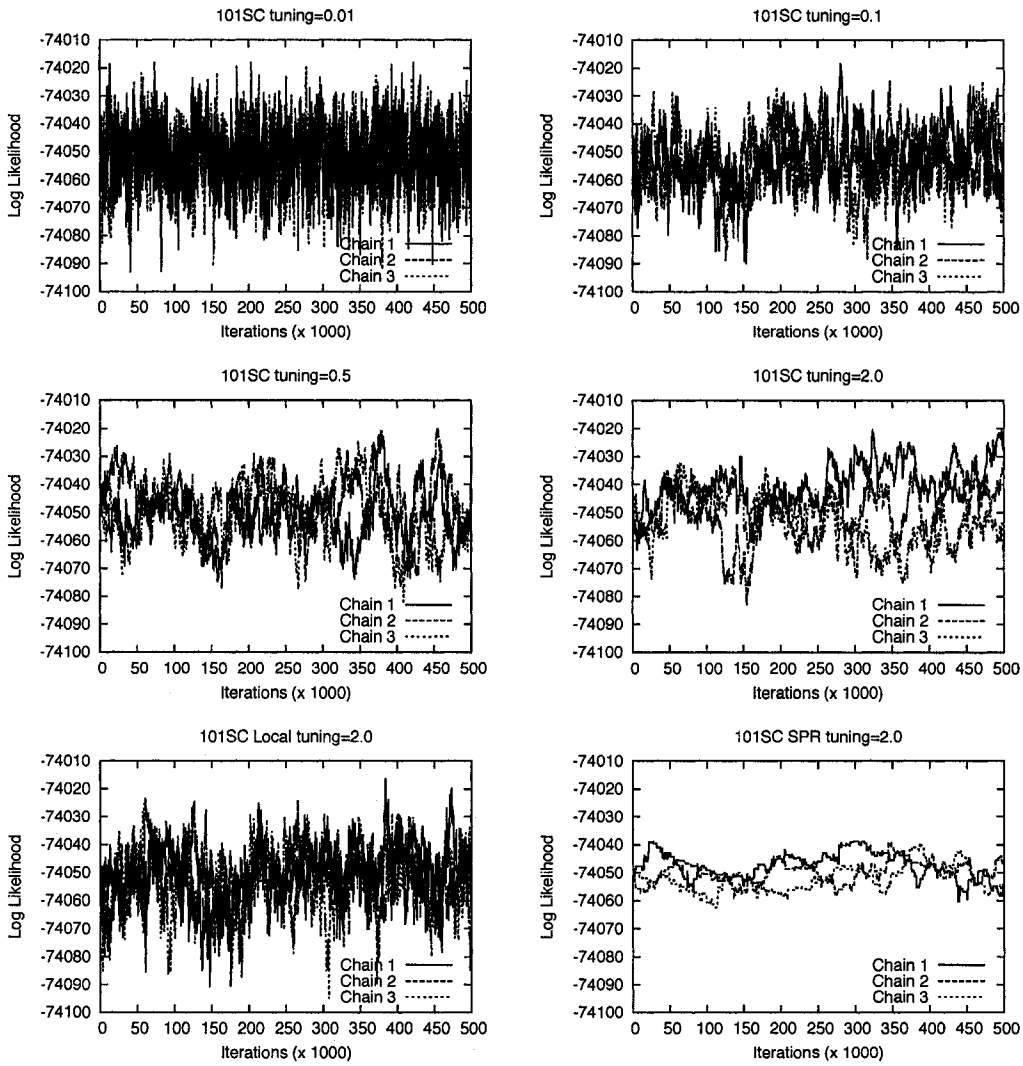
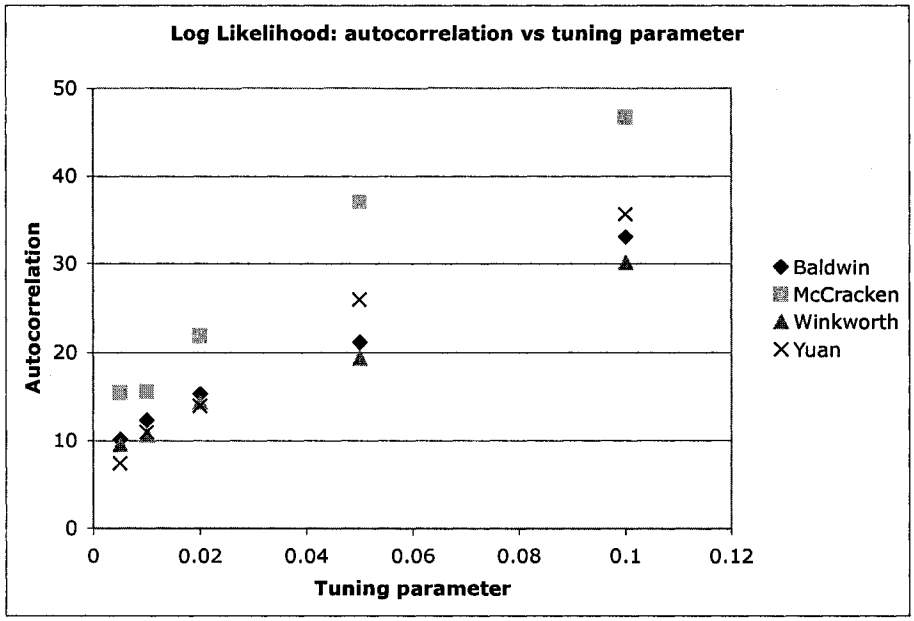


Figure 4.2: Mixing behaviour of the 101_SC data set when analyzed using different algorithms and tuning parameters but starting from the same topology and set of branch lengths. The top four plots are different tuning parameters for the BranchSlide algorithm, and the bottom two are Local and SPR.



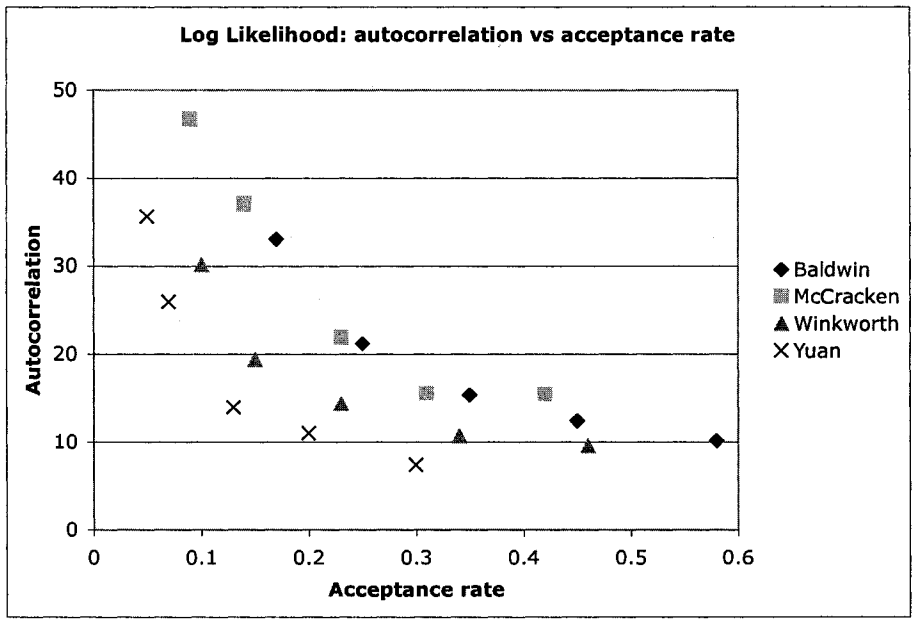


Figure 4.3: Relationship between autocorrelation, tuning parameters and acceptance rate. Each point is an average over three independent chains.

simulated data, autocorrelation for the two parameter is similar for larger acceptance rates, but the values for the log likelihood are greater for lower acceptance rates. With the empirical data, there is no clear trend with the two parameters. This is true even if we separate the points in Figure 4.4 by data set. The highest posterior density (HPD) intervals for the tree length are much narrower for the simulated data than for the empirical data, which may be one cause of the differences seen between the two types of data.

4.3.3 Tree topology measures

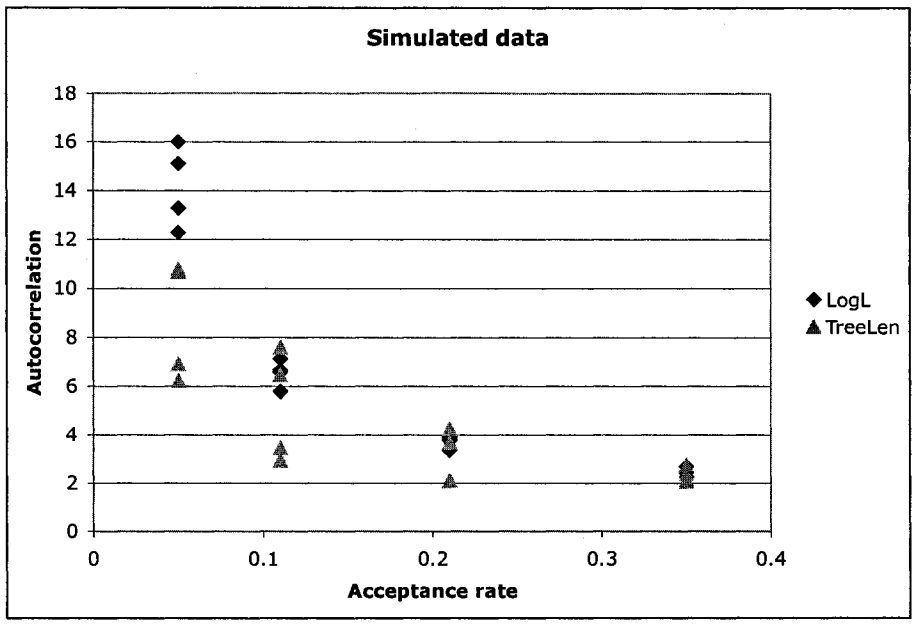
What is effect of different tree proposals on our estimates of the tree topology? For the simulated data, the MCMC does converge on a fairly stable set of topologies. The analyses seems robust to changes in the proposal methods (see Table 4.6). The total number of unique trees sampled, the size of the credible sets and the probability of the MAP tree did not display any systemic change across tuning parameters. Overlap between the three sample chains and the combined chain appears to be good based on size (credible sets for the combined chain have the same number of trees as each of the sample chains).

There is no trend for the value of the RMSET calculated at the end of the run as compared to the tuning parameter. However, Figure 4.5 illustrates that while the value at the end of the run is very similar, there is variation in the shape of the plot. This is consistent with results from previous sections, where all runs appeared to have reached stationarity, but burn-in periods and mixing behaviour differs across the tuning parameters.

For the small empirical data sets, while numerical output parameters appear to have converged, the number of unique sampled trees is approximately equal to the total number of sampled trees for most of the runs, meaning that we have not yet converged to a stable posterior distribution of trees.

Table 4.7 lists the final calculated values of the RMSET and MeanSD. All values are below 0.01 except for the McCracken data sets (which are an order of magnitude higher than the other runs). The McCracken runs are also those that displayed the longest burn-in periods and had failures with the stationarity tests. The probability of the MAP tree is very low for all of these data sets, so that the MeanSD statistic is calculated over a larger number of partitions that the RMSET and has a lower value (noting the single exception with the smallest tuning parameter and the Baldwin data set).

Figure 4.6 illustrated the differences in RMSET traces over the course of the run for different tuning parameters and the same data set. For the Baldwin and McCracken data sets, where almost every sampled tree was unique, there is no



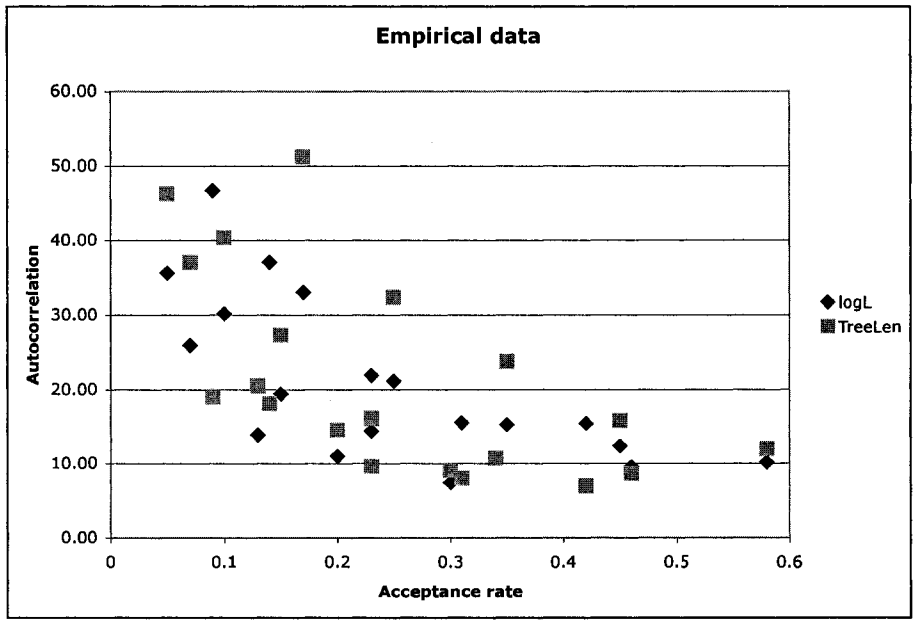


Figure 4.4: Differences in autocorrelation in the two output parameters. With simulated data, the two parameters are similar when autocorrelation is low, but the increase in ACT for the log likelihood is greater as acceptance rates decrease. With empirical data, there is no difference between the two parameters, and overall, autocorrelation values are larger and more variable.

Table 4.6: Effect of MCMC mixing on size and overlap of credible sets for simulated data as well as calculation of the RMSET statistic. Total trees is the total number of unique sampled trees in all three chains. Combined column is the 95% credible set for all three combined chains. If the chains are sampling from the same distribution, the credible sets will be the same size in all four cases (chains 1 through 3 and the combined chain).

Data set	Tuning	P_{MAP}	Total trees	95% credible sets, chain 1	2	3	Combined	RMSET
1	0.005	0.280	35	8	8	8	8	0.00430
	0.01	0.293	32	8	8	8	8	0.00402
	0.02	0.300	28	8	8	8	8	0.00313
	0.05	0.300	22	8	8	8	8	0.00307
3	0.005	0.313	44	16	17	17	17	0.00466
	0.01	0.326	46	17	16	17	17	0.00538
	0.02	0.321	45	18	17	16	17	0.00388
	0.05	0.320	44	16	16	17	17	0.00498
6	0.005	0.176	14	8	8	8	8	0.00470
	0.01	0.170	12	8	8	8	8	0.00362
	0.02	0.170	18	9	8	8	8	0.00313
	0.05	0.169	25	9	8	8	8	0.00846
8	0.005	0.358	26	8	9	8	9	0.00544
	0.01	0.354	27	9	9	9	9	0.00366
	0.02	0.364	26	8	8	7	8	0.00538
	0.05	0.358	26	8	9	9	8	0.00252

Table 4.7: RMSET and MeanSD for the small empirical data sets.

	Baldwin		McCracken	
Tuning	RMSET	MeanSD	RMSET	MeanSD
0.005	0.0028	0.0032	0.0344	0.0186
0.01	0.0056	0.0039	0.0108	0.0066
0.02	0.0033	0.0035	0.0218	0.0115
0.05	0.0037	0.0034	0.0421	0.0206
0.1	0.0043	0.0039	0.0201	0.0126
	Winkworth		Yuan	
Tuning	RMSET	MeanSD	RMSET	MeanSD
0.005	0.0029	0.0005	0.0045	0.0023
0.01	0.0023	0.0004	0.0047	0.0020
0.02	0.0027	0.0005	0.0039	0.0022
0.05	0.0044	0.0005	0.0065	0.0031
0.1	0.0063	0.0007	0.0065	0.0028

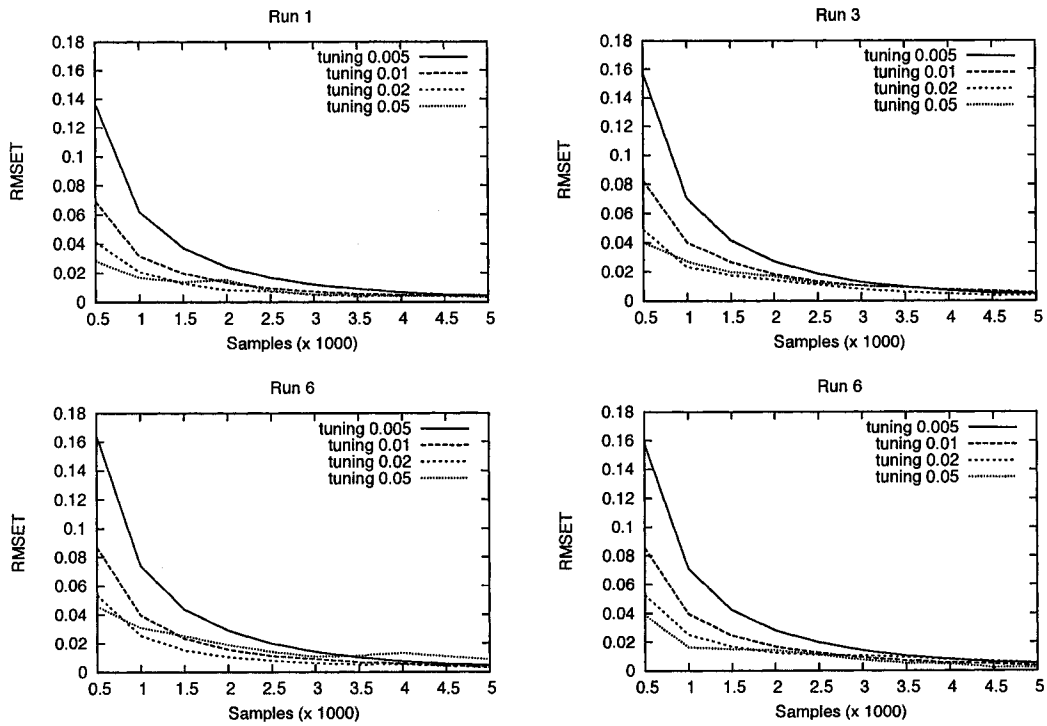


Figure 4.5: RMSET traces for the simulated data sets. All tuning parameters converge to a very similar RMSET value, but the shape of the plot differs between tuning parameters. The first 500 sampled trees were removed as burn-in, which is conservative based on the burn-in judged from the time series plots.

apparent correlation between tuning parameters / acceptance rates and the RMSET value. With the other two data sets, larger values of RMSET are associated with higher tuning parameters (and therefore lower acceptance rates and more autocorrelation). The Winkworth and Yuan data sets did show some shape in the posterior distribution of phylogenies. Table 4.8 details the results for credible sets and probability of the MAP tree. As the tuning parameters increase, the total number of trees explored decreases (as does the size of the credible sets).

The Yuan data was the most informative, and there in an interesting trend over the different tuning parameters. With higher tuning parameters, the estimates from a single chain appear to be more precise, but comparisons between the chains indicate that the overlap between the chains is minimal. The larger tuning parameters produce lower acceptance rates and higher autocorrelation, and the chains explore the tree space less thoroughly. The other data sets did not show this trend, which may simply be due to a lack of information in the data.

Again, these result confirms the need for comparisons across multiple chains. With a single chain analysis, the results would simply look more precise with larger tuning parameters (and the acceptance rates are not low enough to indicate a serious problem).

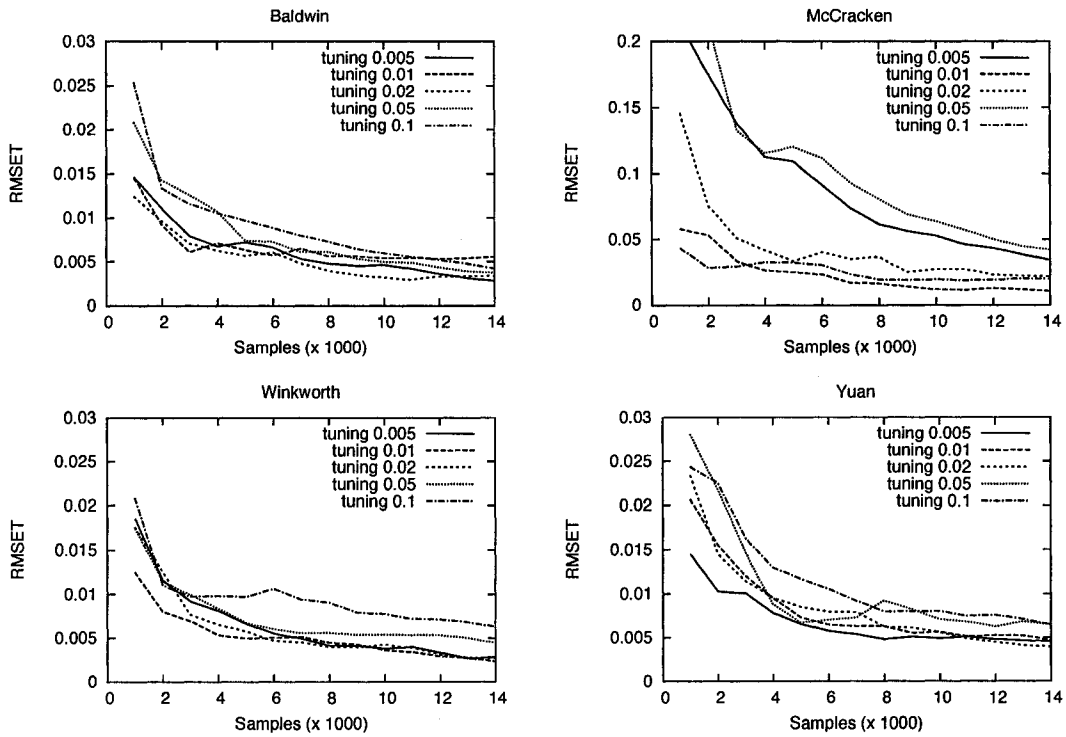


Figure 4.6: RMSET under various tuning parameters for the extended MCMC analyses of the Baldwin, McCracken, Winkworth and Yuan data sets. Note difference in y-axis scale for the McCracken results.

Table 4.8: Effect of MCMC mixing on size and overlap of credible sets for empirical data. Total trees is the total number of unique sampled trees in all three chains (out of a total of 5000 total samples). Combined column is the 95% credible set for all three combined chains. If the chains are sampling from the same distribution, the credible sets will be the same size in all four cases (chains 1 through 3 and the combined chain). Difference is the difference between the sum of the three chains and the combined chain (measuring the amount of overlap between the chains.

Run	Tuning	P_{MAP}	Total trees	95% credible sets, chain 1	2	3	Combined	Difference
Baldwin	0.005	2.38E-05	41998	13300	13300	13300	39900	0
	0.01	2.38E-05	41998	13300	13300	13300	39900	0
	0.02	2.38E-05	41998	13300	13300	13300	39900	0
	0.05	2.38E-05	41998	13300	13300	13300	39900	0
	0.10	4.76E-05	41988	13297	13297	13297	39889	0
McCracken	0.005	2.38E-05	41998	13300	13300	13300	39900	0
	0.01	2.38E-05	41998	13300	13300	13300	39900	0
	0.02	2.38E-05	41998	13300	13300	13300	39900	0
	0.05	7.14E-05	41935	13280	13278	13279	39836	1
	0.10	7.14E-05	41421	13116	13088	13119	39322	1
Winkworth	0.005	7.14E-05	40222	13066	13090	13058	38121	1093
	0.01	9.52E-05	40180	13072	13075	13080	38081	1146
	0.02	9.52E-05	40125	13033	13057	13091	38026	1155
	0.05	1.10E-04	39654	12904	12952	12902	37555	1203
	0.10	9.52E-05	39126	12680	12719	12647	37027	1019
Yuan	0.005	7.69E-03	6341	2471	2769	2666	4242	3644
	0.01	7.11E-03	5819	2397	2272	2498	3720	3447
	0.02	8.41E-03	5700	2154	2326	2344	3601	3223
	0.05	7.33E-03	5356	1980	2053	2114	3257	2890
	0.10	7.95E-03	4669	1699	1681	1655	2814	2221

4.4 Discussion

4.4.1 Effect of proposal method on convergence and mixing

When the data was very informative, as in the case of the simulated data, the Bayesian inference was robust to changes in the proposal method. The burn-in period was shorter for larger moves, but autocorrelation was higher (indicating poorer mixing). Smaller moves had longer burn-in and lower autocorrelation. However, while changes in rate of convergence and mixing behaviour were apparent, they did not have an effect on the size of the credible set, the probability of the MAP tree or the estimation of partition probabilities (as measured by the RMSET statistic).

With the empirical data, which was a more challenging inference problem, the differences between proposals became more apparent. Very small moves (such as those produced by the Local algorithm or BranchSlide with a small tuning parameter) took a long time to converge and were apt to be trapped in local optima. Very large moves (using SPR) converged very slowly but avoided local optima. For the small empirical trees, the range of values of the BranchSlide parameter tested gave similar results for the rate of convergence.

For the large fungal data set (101_SC), the effect of proposal method greatly affected the rate of convergence. Only the largest tuning parameter for the BranchSlide algorithm quickly reached a stable value for the log likelihood over all three chains. However, when all proposal methods started from the same tree, detailed analysis of the results indicated that none of the analyses were at stationarity, and those that displayed the best mixing behaviour used proposal methods with the smallest moves.

As the smaller trees did reach stationarity, I was able to study the effect of proposal methods on MCMC mixing and also the effect of mixing behaviour on inference of the posterior distribution of phylogenies. As with the simulated data, smaller moves were correlated with higher acceptance rates, although the variability in acceptance rates across data sets for a given tuning parameter was larger than for simulated data. Autocorrelation of the log likelihood and tree length were negatively correlated with acceptance rate (although the Local algorithm with small tuning parameter did show the expected high autocorrelation with very small moves), and again, there was more variability in the empirical data than simulated.

When examining the convergence of the partition probabilities and tree topology, it is challenging to see any effect of MCMC mixing when the distribution of topologies is very flat. In the Baldwin and McCracken data sets, nearly every sampled tree is unique and there is no trend when comparing the RMSET or

credible sets against acceptance rates or autocorrelation. When the data are more informative (as in the Winkworth and Yuan data sets), the RMSET increases with increasing autocorrelation and the credible sets of tree decrease in size. This indicates that poor mixing does have a negative effect on our ability to infer the tree topology. The Yuan data set was the most informative, and comparison of the credible sets across the three chains indicates that high autocorrelation causes the MCMC chain to explore the tree space inefficiently, reducing the size of credible set for each chain and reducing the overlap in sampled trees across the chains. This was reflected in higher values for the RMSET and MeanSD for higher autocorrelation and lower acceptance rates. It would be interesting to look for other empirical data sets that also displayed this trend.

I want to emphasize that in most of these analyses, the value of the RMSET and MeanSD statistics were very low (less than 0.01, which is the upper limit suggested by MrBayes). Even when mixing was poor, causing increased variability of the partition probabilities between chains, the differences between the different proposal methods were small. Unless the problems with convergence are serious (as for the McCracken and 101_SC data sets) our estimates of the partition probabilities are fairly robust to changes in convergence and mixing of the MCMC.

4.4.2 Differences between algorithms

Roughly speaking, SPR always proposes a large move, and Local always proposes small moves. BranchSlide can propose both small and large moves, even with a single tuning parameter. Not surprisingly, then, the effect of tuning parameters varied considerably across algorithms. With BranchSlide, the different tuning parameters produced very difference acceptance rates and autocorrelation values. Even with the largest tuning parameters tested, the behaviour of BranchSlide was significantly different than SPR, and with small tuning parameters, it differed from Local.

The effect of tuning parameters for SPR was limited, and this algorithm has very low acceptance rates with both the large and small tuning parameters. The effect of tuning parameters for the Local algorithm was larger than expected. With a small tuning parameter, Local had very high autocorrelation, and was the only method to show high autocorrelation with small moves. Local proposes changes to the branch length, which may also induce a change in topology. A change in topology is more likely with a larger change in branch lengths, so the very small tuning parameter will rarely change the topology. Compare this behaviour to SPR, which changes the tree topology with nearly every move, and also proposes a change in branch length. Only the branch length change is dependent on the tuning parameter.

The differences in convergence and mixing behaviour is not entirely dependent on tuning parameters. Certain parameterizations give similar acceptance rates between the different algorithms and yet the MCMC convergence and mixing was not equivalent.

4.4.3 Autocorrelation

Even with small moves and high acceptance rates (and a sampling frequency of 100), there is still fairly high autocorrelation in phylogenetic MCMC. This is due to the structure of the tree topology parameter. Our proposals alter only a portion of the tree at each iteration, and output parameters calculated from the tree will be identical in regions of the tree that have not changed. Autocorrelation appears to increase as the data becomes less informative. The simulated data had much lower autocorrelation than the empirical data, despite similar number of taxa. The Yuan and Winkworth data also had lower autocorrelation than the Baldwin and McCracken data sets.

Given the structure of phylogeny parameter, it is unlikely that autocorrelation can be greatly reduced through changes in the transition kernel. One method of dealing with high autocorrelation is to increase the sampling frequency to reduce dependence between subsequent sampled states. The increase in sampling frequency should also be accompanied by an increase in total run length, to ensure a sufficient number of sampled states. The increase in run length is a more important component than the increased sampling frequency. High autocorrelation does not mean that inferences based on every sampled state are inaccurate, just that we need a larger total number samples than if they were independent. Some authors suggest that subsampling an MCMC chain is not justified except to reduce storage requirements (MacEachern and Berliner, 1994). Storage is a concern with phylogenetics, as the tree topology and branch lengthes are stored as a string, and this string increases in size as the number of taxa increases. For example, 10000 samples from a single chain with 35 taxa required 8.5 MB of storage, while the same number of trees with 101 taxa required 24 MB of storage.

4.4.4 Multiple chains

If mixing is poor, a single chain can give false results, both in terms of accuracy and precision. The analysis may appear to be at stationarity while simply trapped in a local optima that is not the global optima. I found many examples of this behaviour in the empirical data sets, and this would be impossible to detect with only one chain. As confirmation, the test for stationarity across multiple chains (the PSRF) was more sensitive than the single chain diagnostic (the HW test).

With respect to the inference of the tree topology, a poorly mixing chain does not as thoroughly explore the tree space. This results in a smaller credible set which appears to be a more precise result unless compared to a one or more additional independent chains. Statistics to detect this behaviour compare partition probabilities across chains and cannot be calculated with a single chain.

4.4.5 Optimal proposals

When comparing the BranchSlide, Local and SPR algorithms, results indicate that neither Local or SPR when used on their own provide fast convergence or good mixing. The advantage of the BranchSlide algorithm is that a single tuning parameter can provide a range of move sizes, due to selection of the distance to move the subtree from a Normal distribution. If we set the tuning parameter for BranchSlide to produce fast convergence, however, the proposal is not optimal for the mixing phase. For informative data, the results are robust to the choice of proposal, so a single tuning parameter can be used throughout the analysis. With more challenging data sets, this may not be sufficient.

The strategy used by MrBayes is to use a combination of algorithms. For example, for the analysis of the 101_SC data, 75% of the iterations used an algorithm that proposes larger moves and the remaining iterations used the Local algorithm. During the burn-in phase, the Local moves are not very useful, but during mixing, they likely increase the overall acceptance rate and improve mixing over solely using the larger proposals.

MrBayes also uses MCMCMC, which can allow for the injection of large moves into the MCMC chain when we swap between heated and cold chains (Altekar et al., 2004). A similar idea is that of using small world networks as a proposal (Guan et al., 2006), where most of the moves are local but we include an occasional random draw. These types of proposal strategies are designed to deal with multiple regions of high probability. This occurs often in problems such as image analysis, and can also be a problem in phylogenetics with multiple modes for topology (Mossel and Vigoda, 2005) or branch lengths (Chor et al., 2000; Stefankovic and Vigoda, 2006).

MCMCMC methods are expensive, as each single independent chain requires one cold chain plus x heated chains, and the cost increases dramatically if we want to run multiple independent chains. For the large difficult data set analyzed in this study, MCMCMC did not converge as fast as simple MCMC with a large proposal strategy. Recent papers have suggested that MCMCMC may not be justified for phylogenetics (Pagel et al., 2004; Beiko et al., 2006). Results in these studies indicate that the method can be helpful in reaching stationarity, but then swaps between chains are very infrequent during the mixing phase. If we can use different

tuning parameters with simple MCMC to get the same rate of convergence, then MCMCMC is inefficient. Completely random moves, as used in small world proposals, are probably not recommended for phylogenetics, given the immense size of the tree space, but the injection of larger moves into an MCMC analysis should be further studied.

4.5 Conclusions

The choice of proposal algorithm and tuning parameter can dramatically affect the rate of convergence and quality of mixing in Bayesian phylogenetic MCMC methods. Generally, larger proposals are associated with shorter burn-in periods, lower acceptance rates and higher levels of autocorrelation. As the size of the proposal decreased, the burn-in period is longer but mixing is better, with lower autocorrelation between the sampled states.

For each of the data sets, the optimal method for fast convergence does not give the best results in terms of mixing. If the data is informative, however, then Bayesian phylogenetic MCMC is robust to changes in proposal methods and we can use the same proposal for both the convergence and mixing phases without any noticeable effect on inference of the phylogeny. This was the case with the simulated data.

The empirical data displayed a greater range of results than the simulated data. The variability of acceptance rates for a given tuning parameter was larger across data sets. Autocorrelation was higher and was more variable, even though the trees had similar numbers of taxa. Increased autocorrelation decreased the ability of the MCMC to explore the tree space, as judged by narrower credible sets of topologies. Despite the higher autocorrelation and visible differences in the posterior distributions, estimates of the partition probabilities was still very good as judged by the low values of the RMSET statistic (the exception was a single data set with high values of the RMSET but also failures with numerical convergence diagnostics).

Analysis of a large fungal data set known to be a hard problem indicated that convergence can be very sensitive to proposal algorithms and that MCMC with large proposals can outperform MCMCMC in difficult inference problems. It also illustrated that the difference between optimal algorithms for the mixing and convergence phases can be very large for hard problems.

The analysis of several of the data sets with tuning parameters dynamically adjusted based on acceptance rate indicated that this strategy was not straightforward (data not shown). The acceptance rate is slow to stabilize after adjusting the tuning parameter, causing a large delay between successive

adjustments and a very long setup stage. There are a number of parameters required for this strategy, including an initial tuning parameter, the number of iterations between adjustments and the optimal acceptance rates for the setup and sampling phases. Automated methods for selecting tuning parameters may simply require too many other parameters to be justified over simply setting the tuning parameters to static values based on short preliminary analyses.

If setting parameters dynamically is hard, and the optimal tuning parameters for convergence and mixing are quite different, perhaps one solution is to initialize Bayesian analyses with partially optimized starting trees. The increase in fast ML methods (for example, the programs PHYML, RAxML and Garli (Guindon and Gascuel, 2003; Stamatakis et al., 2005; Zwickl, 2006)) provides one source for starting trees. Currently, we seem to be optimizing Bayesian methods for fast convergence rather than optimal mixing, and this strategy would allow Bayesian methods to focus on optimal exploration of the stationary distribution near the global optima. I note that using a ML starting tree may bias the Bayesian inference if the ML tree happens to be in a local, rather than a global, optima.

In this study, I fixed the model parameters and integrated only over topology and branch lengths. This was done intentionally to isolate the issue of convergence of the posterior distribution of phylogenies. Of course, phylogenetic inference is greatly influenced by the choice of model parameters. Further studies should infer both model parameters and phylogeny and determine the effect of convergence of model parameters on the inference of the posterior distribution of phylogenies.

For some analyses that use a phylogeny as input (such as calculation of divergence times or detection of residues undergoing selection), it is important to have accurate estimates of the branch lengths on the tree. The topology statistics examined here are based on partition probabilities, not branch lengths, and while I do examine the tree length, this is not very informative with respect to individual branches on the tree. It should be possible to develop statistics that incorporate the branch lengths in order to have a more sensitive picture about convergence of the branches than we can get from the tree length alone.

Future work with choosing proposal mechanisms should examine combinations of proposals and continue to judge the relative utility of MCMC versus MCMCMC. If we can improve basic MCMC methods so that MCMCMC is not required, this would increase the efficiency of Bayesian phylogenetic methods. Finally, differences between optimal algorithms for speed of convergence and optimal mixing indicate that further exploration of dynamically-set tuning parameters is justified to allow for tuning of algorithms in each of these phases.

Bibliography

Altekar, G., S. Dwarkadas, J. P. Huelsenbeck, and F. Ronquist. 2004. Parallel Metropolis coupled Markov chain Monte Carlo for Bayesian phylogenetic inference. Bioinformatics 20:407–415.

Beiko, R. G., J. M. Keith, T. J. Harlow, and M. A. Ragan. 2006. Searching for convergence in phylogenetic Markov chain Monte Carlo. Syst. Biol. 55:553–565.

Chor, B., M. D. Hendy, B. R. Holland, and D. Penny. 2000. Multiple maxima of likelihood in phylogenetic trees: an analytic approach. Mol. Biol. Evol. 17:1529–1541.

Gelfand, A. and S. Sahu. 1994. On Markov Chain Monte Carlo Acceleration. J Comput. Graph. Stat. 3:261–276.

Gilks, W., G. Roberts, and S. Sahu. 1998. Adaptive Markov chain Monte Carlo through regeneration. J. Am. Stat. Assoc. 93:1045–1054.

Guan, Y., R. Felibner, P. Joyce, and S. Krone. 2006. Markov chain Monte Carlo in small world. Stat. Comput. 16:193–202.

Guindon, S. and O. Gascuel. 2003. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. Syst. Biol. 52:696–704.

Hasegawa, M., H. Kishino, and T. Yano. 1985. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. Journal of Molecular Evolution 22:160–74.

Hordijk, W. and O. Gascuel. 2005. Improving the efficiency of SPR moves in phylogenetic tree search methods based on maximum likelihood. Bioinformatics 21:4338–4347.

Larget, B. and D. L. Simon. 1999. Markov chain Monte Carlo algorithms for the Bayesian analysis of phylogenetic trees. Mol. Biol. Evol. 16:750–9.

MacEachern, S. N. and L. M. Berliner. 1994. Subsampling the Gibbs sampler. Am. Stat. 48:188–190.

McCracken, K. and M. Sorenson. 2005. Is homoplasy or lineage sorting the source of incongruent mtDNA and nuclear gene trees in the stiff-tailed ducks (Nomonyx-Oxyura)? Syst. Biol. 54:35–55.

Mossel, E. and E. Vigoda. 2005. Phylogenetic MCMC algorithms are misleading on mixtures of trees. Science 309:2207–2209.

Pagel, M., A. Meade, and D. Barker. 2004. Bayesian estimation of ancestral character states on phylogenies. Syst. Biol. 53:673–684.

Rambaut, A. and A. Drummond. 2005. Tracer verson 1.3.

Stamatakis, A., T. Ludwig, and H. Meier. 2005. RAxML-III: a fast program for maximum likelihood-based inference of large phylogenetic trees. Bioinformatics 21:456–463.

Stefankovic, D. and E. Vigoda. 2006. Phylogeny of mixture models: Robustness of maximum likelihood and non-identifiable distributions. http://arxiv.org/abs/q-bio/0609038.

Winkworth, R., D. Bryant, P. Lockhart, D. Havell, and V. Moulton. 2005. Biogeographic interpretation of splits graphs: least squares optimization of branch lengths. Syst. Biol. 54:56–65.

Yuan, Y.-M., S. Wohlhauser, M. Mller, J. Klackenberg, M. Callmander, and P. Kpfer. 2005. Phylogeny and biogeography of exacum (gentianaceae): a disjunctive distribution in the Indian ocean basin resulted from long distance dispersal and extensive radiation. Syst. Biol. 54:21–34.

Zwickl, D. J. 2006. Genetic algorithm approaches for the phylogenetic analysis of large biological sequence datasets under the maximum likelihood criterion. Ph.D. thesis The University of Texas at Austin.

Chapter 5

Summarizing a posterior distribution of phylogenies

5.1 Introduction

When we use Bayesian phylogenetic inference to generated a posterior distribution of phylogenies, we generally need a method to summarize the information in the distribution. In a systematics study, a summary method may be needed to graphically portray the relationships between species on a tree. An evolutionary study may require one or more trees in order to study such topics as divergence times (Sanderson, 2002; Thorne et al., 1998), selection (Yang et al., 2005) or phylogeography (Knowles and Maddison, 2002; Hewitt, 2001). In either case, the information in the full posterior distribution of trees must be sufficiently reduced for the required purpose.

While the complexity of the phylogeny as a parameter makes traditional statistical methods problematic, this is certainly not due to a lack of information contained in the phylogenies, leading us to develop creative ways of summarizing and comparing an input list of trees. In Chapters 3 and 4, results indicate that there is significant signal with respect to the topology and partition probabilities, even when the posterior distribution of phylogenies is quite broad.

A common starting point for summarizing any distribution is to report summary statistics that include such concepts as the mean, mode and variance. For a posterior distribution of phylogenies, a natural point estimate is the mode of the distribution, or the maximum *a posteriori*, or MAP, tree. An interval estimator is the credible set of trees. In an ideal situation, the credible set would be small and there would be a significant proportion of the posterior probability assigned to the

[0]A version of this chapter has been submitted for publication. Cranston and Rannala 2006. Systematic Biology.

MAP tree.

We rarely see this sort of reporting in the phylogenetics literature. Why is this the case? First, the probability of the MAP tree may be quite low, even if the marginal posterior probabilities on most of the internal nodes are high. This can happen if the posterior distribution is broad, giving credible sets with a large number of trees. A large credible set is unlikely to produce a single MAP tree with high probability. In this case, the distribution of trees is relatively flat and the probability of the single best tree may not be much greater than any number of other trees.

The second reason may be that reporting a credible sets of trees is not as intuitive as a credible set for a continuous numerical parameter. The credible set of trees defines the number of trees contained in the set, giving a measure of the overall spread of the distribution. However, there is no information about the relationship between the trees or how widely they differ from one another. Contrast this to a posterior distribution of a continuous parameter with a single mode, where the credible set contains a range of values that fall between a well-defined minimum and maximum point.

Instead, the most common method for reporting a posterior distribution of phylogenies is the majority rule consensus (MRC) tree as a point estimate with partition probabilities at each internal node as a measure of the uncertainty. The MRC tree is constructed by combining all partitions with probability greater than 0.5 from the list of partition probabilities. This often results in multifurcations being introduced into the tree in order to combine low probability binary partitions into a single well-supported multifurcating node. Since the MRC is a combination of sampled partitions, it is possible (although unlikely) that the entire tree was never actually sampled during the MCMC, meaning that we cannot assign a posterior probability to the whole tree. If construction of the MRC involves collapsing nodes to produce multifurcations, then it is certain that we did not sample the tree and there is no probability for the MRC tree. Given that one of the advantages of a Bayesian analysis is the ability to assign probabilities to trees, it is preferable to retain this measure of support for the tree when presenting the results.

Whether we use the MAP tree, the MRC tree or a different consensus method, reducing the distribution to a single tree fails to adequately describe the full distribution. Although providing marginal probabilities for a tree, or a single clade, can improve the information content, other summary methods can describe aspects of the distribution not captured by such point estimates.

If the credible set of trees is large and the probability of the MAP tree is small, what information can we obtain from the distribution? This situation may occur if

there is a lack of information, conflicting signals or possibly due to a lack of convergence of the MCMC. If we can eliminate the issue of convergence (which I note is a non-trivial process for many data sets), then either there is simply not enough information in the data to infer a single strongly supported tree or there is an underlying evolutionary process that does not support a single tree for the full set of taxa. In any case, additional types of analysis can elucidate information about the support for various parts of the evolutionary history even if we cannot place a high probability value on a single tree.

5.2 Summary using tree pruning

One way to extract additional information from the distribution of phylogenies is to simplify the distribution until we find a well-supported MAP tree. This is akin to finding an underlying well-supported 'skeleton tree' within the full posterior distribution of trees. While sampling trees using MCMC, we expect that portions of the tree will remain nearly fixed while integrating over the uncertainty. We can find this constant tree by removing uncertain taxa and leaving those taxa that are well-supported by a large percentage of the input trees. This type of approach is similar to the so-called maximum agreement subtree (MAST) methods (Finden and Gordon, 1985). Given a list of trees, the maximum agreement subtree is the largest subtree that is contained in all of the input trees. This subtree (or set of subtrees) is generally found using heuristic search and even for three input trees, the problem is NP-complete (Amir and Keselman, 1997), meaning that a polynomial-time solution is not available.

The MAST tree, by definition, must agree with each of the input trees. This means that any taxa with ambiguous relationships are stripped from the input trees, irrespective of the support values for the ancestral nodes. When requiring strict agreement, the resulting subtree may contain only a very small subset of the taxa in the starting dataset, or the a MAST tree may not exist at all.

Rather than look for a strict MAST tree contained in every one of the input trees, I instead propose a method that searches for agreement subtrees that may be present in only some of the input trees. This is similar to the body of literature describing frequent subtree mining (reviewed in Chi et al., 2005). There are two properties that differentiate phylogenies from other many of the types of trees and networks used in the subtree mining algorithms. The first is that phylogenies are unordered, that is, there is no information in the left-right orientation of nodes. Whether we draw a given taxa as a left or the right descendant of an ancestral node does not affect the uniqueness of the tree. The second, more important, property is

that ancestral nodes are only interesting when they have descendant nodes. If we remove two sister taxa from a phylogeny, then the ancestral node is no longer of interest, and we no longer include it on the tree. This differs from situations in non-phylogenetic applications where we are interested in properties of lower-level nodes even if there are no higher-level descendents. Frequency subtree mining has been addressed specifically in the context of phylogenies (Shasha et al., 2004) but only to look for common pairs of taxa in trees with different sets of taxon labels. The problem addressed in this study uses a posterior distribution of phylogenies as input, which always contain the same set of taxon label in each tree.

To search for agreement subtrees in a posterior distribution, we can use the posterior probabilities to score the various subtrees. The posterior distribution of trees contains a count of each tree in proportion to its probability, so we prefer agreement subtrees that are present in a larger proportion of the original sampled trees. In effect, we are weighting the agreement subtrees using the sum of the posterior probabilities of the input trees that agree with the subtree. An example of this strategy is shown in Figure 5.1.

Summarizing the set of trees using pruned subtrees, in addition to producing estimates of the best-supported subtrees, also helps to identify so-called rogue taxa. These are species that appear in multiple relationships with other taxa in the trees, and are of particular concern when more than one relationship has non-negligible probability. Removing these taxa from the posterior distribution will have a greater effect on the MAP tree probability than taxa that have well-supported relationships on the input trees. For example, given the posterior distribution of trees in Table 5.1, we can prune off different taxa from the trees and compare the resulting distributions. The original distributions and those resulting from pruning taxon 2, 6 or 16, are shown in Figure 5.2. Although pruning any of these taxa improves the probability of the MAP tree and decreases the width of the credible sets, taxon 6 gives the greatest improvement, therefore we prefer removal of taxon 6 over the other two taxa. This result indicates that the placement of taxon 6 on the tree is less well-supported than the placement of taxon 2 or 16, which is not immediately obvious from studying the posterior distribution of phylogenies.

This can then be extended to greater numbers of pruned taxa. For a more complex distribution of trees (with more starting taxa or greater initial spread), removing a single taxa may not be sufficient to produce a subtree with high probability. If this is the case, pruning additional taxa will further collapse the input trees until a well-supported skeleton is discovered.

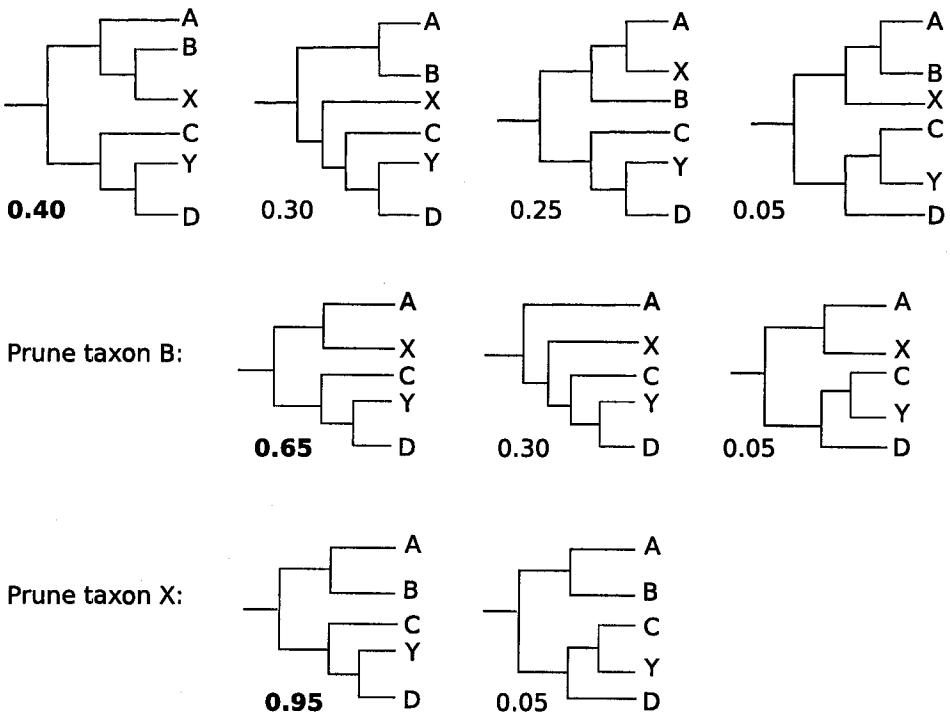


Figure 5.1: Example of how the pruning method can increase the probability of the MAP tree. The original posterior distribution is shown as the top of the figure, with the MAP probability labeled in bold. We can increase the P_{MAP} be pruning off any single taxon. In this example, we prefer to prune taxon X rather than taxon B due to the higher probability of the resulting subtree.

Table 5.1: A sample posterior distribution of phylogenies.

Probability	Tree
0.6029	((((((2,(6,(16,27))),(7,(48,(43,44)))))
0.1637	(((((((2,6),(16,27)),(7,(48,(43,44)))))
0.0832	((((((6,(2,(16,27))),(7,(48,(43,44)))))
0.0646	((((((2,(16,(6,27))),(7,(48,(43,44)))))
0.0254	(((((((2,16),(6,27)),(7,(48,(43,44)))))
0.0194	((((((2,(27,(6,16))),(7,(48,(43,44)))))
0.0123	((((((2,(6,(16,27))),(7,(44,(43,48)))))
0.0045	((((((16,(6,(2,27))),(7,(48,(43,44)))))
0.0040	((((((6,(27,(2,16))),(7,(48,(43,44)))))
0.0040	((((((16,(2,(6,27))),(7,(48,(43,44)))))
0.0030	((((((6,(16,(2,27))),(7,(48,(43,44)))))
0.0024	((((((16,(27,(2,6))),(7,(48,(43,44)))))
0.0021	(((((((2,6),(16,27)),(7,(44,(43,48)))))
0.0020	(((((((6,16),(2,27)),(7,(48,(43,44)))))
0.0015	(((((((2,6),(16,27)),(7,(43,(44,48)))))
0.0015	((((((2,(16,(6,27))),(7,(43,(44,48)))))
0.0012	((((((27,(6,(2,16))),(7,(48,(43,44)))))
0.0005	((((((6,(2,(16,27))),(7,(43,(44,48)))))

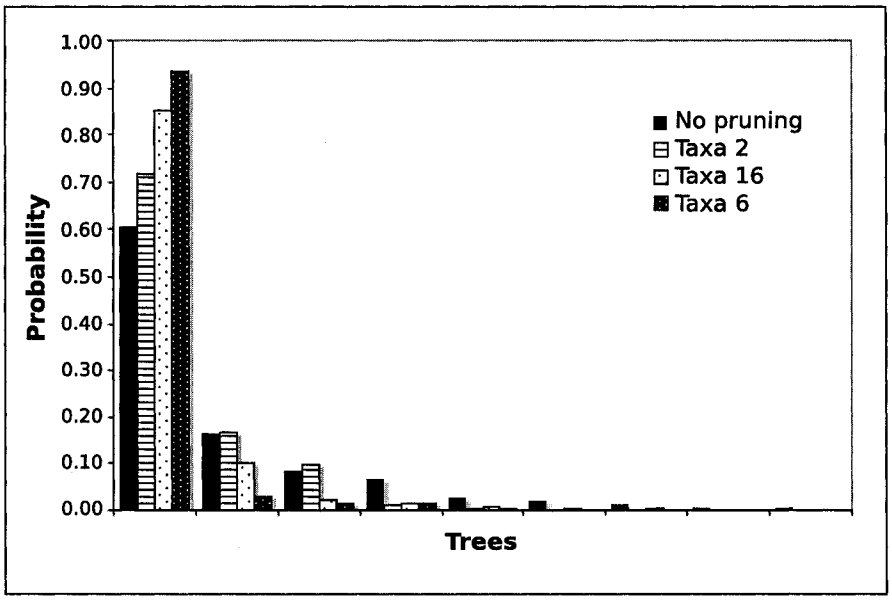


Figure 5.2: Posterior distribution changes after pruning. I show only the ten most probable trees (others have negligible probability). The three patterned series show the effects of pruning off a single taxa from the posterior distribution. In this particular case, pruning taxa 6 gives a MAP tree with the highest probability, 0.95.

5.3 Theory

Assume the existence of a vector of R tree topologies, $\bar{\tau} = \{\tau_i\}$, $i = 1, 2, ...R$, generated by a Bayesian MCMC phylogenetic inference algorithm. The probability of the MAP tree, M, can be calculated as follows:

$$P_{MAP}(\tau) = \frac{1}{R} \sum_{i=1}^{R} I_M(\tau_i) \tag{5.1}$$

where

$$I_M(\tau) = \begin{cases} 1 & \text{if } \tau_i = M \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

The objective is to identify k taxa (from a total of S taxa) that, when eliminated from each tree in $\bar{\tau}$, improve the support for a single best tree. I set a target posterior probability for a MAP tree based on $S - k$ taxa and then attempt to minimize k.

Let the taxa indices be $\bar{S} = \{1, ..., S\}$, where S is the total number of taxa included in the phylogenetic analysis. A potential subset of taxa is $S_k = i_1, ..., i_k$ where $i_j \in \{1, ..., S\}$ and $i \neq j$. I then prune each of the taxa in this subset from each of the input trees to obtain a new set of trees, $\tau[\bar{S}_k] = \{\tau_i[S_k]\}$, where $\tau_i[S_k]$ is the subtree obtained by removing the set of taxa S_k from tree, τ_i.

Once I have the pruned list of trees, which is a sample from the distribution of subtrees, I can find the mode of the distribution:

$$P_{MAP}(\tau[S_k]) = \frac{1}{R} \sum_{i=1}^{R} I_M(\tau_i[S_k]) \tag{5.3}$$

I start with a small value for k, prune various combinations S_k and see which sets most improve the MAP tree probability. For even a moderate number of taxa, the number of distinct sets is enormous - equivalent to the number of combinations of k items chosen from S items. An exhaustive search is possible for at least $k < 3$, but as k increase, the number of possible combinations is far too large, especially as S, the number of taxa in the input trees, increases. Therefore, I developed two stochastic search algorithms to search for sets of taxa that, when pruned, improve the probability of the MAP subtree. One of the algorithms is an MCMC search and the other uses Threshold Accepting. Both use the same general strategy of selecting a set of k taxa, S_k, to prune from the tree, perturbing the set to create S_k', and determining if this new set improves the probability of the MAP tree as compared to the original set. If removing a particular set of taxa improves P_{MAP}, then I keep this set for the next iteration. If not, then I sometimes keep the new set, according

to a set of rules that differs between the two algorithms. I describe the details of each method below.

5.3.1 MCMC algorithm

The MCMC algorithm uses the relative probability of the new MAP tree against the former MAP probability as an objective function for a Metropolis-Hastings proposal ratio.

The MCMC algorithm is implemented as follows:

1. Set number of species removed = k
2. Choose a subset of k species, S_k, from the total list of S species (number of possible subsets = $\binom{S}{k}$)
3. For each unique tree in the sampled set, remove the species in S_k from the tree
4. Calculate the initial probability of the map tree, $P_{MAP}(\tau[S_k])$
5. Choose number of iterations, i, based on total number of possible subsets
6. Start the MCMC loop and run for i iterations:
   (a) Create a new list of k species = S'_k, by removing one or more species from S_k and replacing them with an equal number of species from the remaining list of n taxa
   (b) Remove the species in S'_k from the original list of trees
   (c) Calculate $P_{MAP}(\tau[S'_k])$
   (d) If $\left(\frac{P_{MAP}(\tau[S'_k])}{P_{MAP}(\tau[S_k])}\right) >$ uniform(0,1) then accept the new list; else discard and keep the old list
7. If $max(P_{MAP}) < limit$, then $k = k + 1$ and repeat from 1; else quit

Proposing the new list of k species to remove, S'_k, involves moving a number of taxa from the current list, S_k, into a holding vector of unremoved taxa and then moving an equivalent number of taxa from the holding vector into S'_k. The number of taxa moved in each step is chosen from a Poisson distribution with rate $0.5(k-1)$.

The prior probability of removing a given set of taxa is uniform. The proposal ratio is symmetric, as I randomly choose taxa to remove and to replace. Within a single MCMC loop, the size of the sets (k) does not change. Therefore, the Metropolis-Hastings (M-H) ratio consists only of comparing the objective function, which is the probability of the MAP tree for the given sets of taxa: $P_{MAP}(\tau[S_k])$. The probability of accepting a 'worse' taxa set is then proportional to the M-H ratio.

When the algorithm discovers a new optimum, I discard any saved results and store the new subtree as well as the list of pruned taxa. I keep all of the sets of taxa (and the resulting pruned subtrees) that have the a P_{MAP} equal to the optimum value.

5.3.2 Threshold Accepting

In Threshold Accepting (Dueck and Scheuer, 1990), a new state is always accepted if it is within a certain distance, or threshold, of the current state. The threshold is relatively large at the start of the algorithm, allowing exploration in a large region of the sample space and movement between local optima. As the algorithm progresses, the threshold is progressively lowered until the method only accepts solutions that are extremely close to the current solution. Threshold Accepting is related to the Simulated Annealing (SA) algorithm (Kirkpatrick et al., 1983). In SA, acceptance is based on the function $e^{\triangle E/T}$, where $\triangle E$ is the different in objective functions between the two states and T is a temperature parameter. TA simplifies the SA strategy by always accepting if $\triangle E$ is within a certain threshold. In this method, I accept a new set of taxa if the difference $[P'_{MAP} - P_{MAP}]$ is less than the set threshold. For my purposes, this strategy works well because the P_{MAP} values are constrained to the range [0,1] and the threshold can be set to a well-defined difference in probability. TA also eliminates the cost of exponentiation in SA and random number generation present in SA and MCMC.

The TA algorithm is implemented as follows:

1. Set number of species removed = k
2. Choose a subset of k species, S_k, from the total list of S species (number of possible subsets = $\binom{S}{k}$)
3. For each unique tree in the sampled set, remove the species in S_k from the tree
4. Calculate the initial probability of the map tree, $P_{MAP}(\tau[S_k])$
5. Choose number of iterations, i, based on total number of possible subsets
6. Start TA loop for i iterations:
   (a) Set threshold = t and increment = i
   (b) Start loop for this threshold:
      i. Create a new list of k species = S'_k, by removing one or more species from S_k and replacing them with an equal number of species from the remaining list of n taxa
      ii. Remove the species in S'_k from the original list of trees
      iii. Calculate $P_{MAP}(\tau[S'_k])$
      iv. If $[(P_{MAP}(\tau[S'_k]) - P_{MAP}(\tau[S_k])] > -t$ then accept the new list; else discard and keep the old list
   (c) $t = t - i$ (if $t = 0$, exit loop)
7. If $max(P_{MAP}) < limit$, then $k = k + 1$ and repeat from 1; else quit

Changing the threshold sequence allows for fine-tuning of the algorithm to the particular data set being analyzed. Depending on the starting P_{MAP} and the

breadth of the distribution, different initial threshold and decrement values will be appropriate. The starting threshold should be chosen so that the initial acceptance rate is approximately 80% (Dueck and Scheuer, 1990). The acceptance rate should then decrease as the threshold decreases. If the acceptance rate is too high, then we accept too many moves and the procedure behaves more like a random search, without moving towards an optimum. If the acceptance rate is too low, the likelihood of getting trapped in a local optimum increases.

5.3.3 Implementation and output

The method is implemented as MAPminer, a C++ program that takes as input either a Nexus tree block containing a posterior distribution of phylogenies (such as the *.trprobs summary file from MrBayes) or any file containing a list of unweighted phylogenetic trees in newick format (such as the *.t output file from MrBayes). In the latter case, the program will calculate the posterior distribution of trees from the unweighted list before beginning the pruning algorithm. The trees can be rooted or unrooted. It is also possible to pool samples from multiple runs. The user specifies the probability limit for the MAP tree (*prlimit*) and the maximum number of taxa to remove (max_k). The program exits when it reaches *prlimit* or max_k, whichever comes first. The number of iterations and the length of the burn-in can also be adjusted, and specific taxa can be excluded from pruning (in cases where you are interested in the most well-supported subtree that contains a particular taxa or group of taxa).

For each value of k (number of taxa removed), the program outputs the list of subtrees with maximal P_{MAP}. There may be more than one subtree with the same maximum value for P_{MAP} if more than one set of taxa can be pruned to give the same probability for the MAP tree. For each taxon, the method summarizes the frequency that the taxon is removed from each of the best subtrees. For example, if half of the best subtrees lack a given taxon, then the frequency for that taxon is 0.50.

The running time of the algorithm depends on the number of taxa in the input trees, the number of unique input trees and the shape of the distribution. A more disperse distribution contains a larger number of unique trees to search and will also likely require a larger number of taxa to be pruned, meaning that the runtime will be longer than for a more sharply peaked distribution. Although I did not study this explicitly, the analysis time for the pruning algorithm is likely to be positively correlated with the amount of time required for the initial phylogenetic inference.

5.4 Methods

To illustrate use of the method, I analyzed the posterior distribution of trees from analysis of both simulated and empirical data. For simulated data, I generated five different phylogenies of 50 taxa using a birth-death process (speciation rate = 8.5, extinction rate = 0.5 and sampling frequency = 0.01). For each phylogeny, I simulated 5000 sites under the Jukes Cantor model of evolution using the evolver package of PAML (Yang, 1997). Phylogenetic inference was performed with MrBayes v. 3.1.2 (Huelsenbeck and Ronquist, 2001; Ronquist and Huelsenbeck, 2003) using the known evolutionary model and a birth-death prior on branch lengths. I note that in this study, the details of the phylogenetic inference method are not critical, as the goal is simply to produce a distribution of trees for the post-run analysis, rather than infer the phylogenies themselves.

Using the output from the phylogenetic inference, I ran MAPminer using both algorithms in order to summarize the posterior distribution. The limit on number of taxa to prune was set at 10 (20% of the total taxa) and the desired limit for the probability of the MAP tree was 95%. For small numbers of removed taxa ($k = 3$ and $k = 4$), I also performed an exhaustive search in order to compare the true frequent subtrees with those found by the stochastic searches.

The empirical data was the posterior distribution of phylogenies for a data set of 85 Carnivore species (Fulton and Strobeck, 2006). I note that I did not perform the phylogenetic analysis, instead obtaining the MrBayes output files directly from the authors of the original paper and using these files as input for the MAPminer program.

5.4.1 MCMC settings

Initial testing using the MCMC algorithm with posterior distributions from the simulated data displayed an extremely high acceptance rate (greater than 90%). In light of the data, this high acceptance rate is expected. The objective function uses the posterior probabilities, which are proportional to the likelihood of the trees. We know *a priori* that the list of input trees only contains trees selected as reasonable by the original phylogenetic inference. The range of likelihood values for these trees is much smaller than for the full tree space, so comparing the posterior probilities based on these likelihoods should very often accept a proposed state. However, this makes the algorithm inefficient, as the search becomes more like a random Monte Carlo search than a directed MCMC search. In order to increase the sensitivity of the method, I altered the acceptance procedure so that I accept if:

$$\left(\frac{P_{MAP}(\tau[S'_k])}{P_{MAP}(\tau[S_k])}\right)^x > U(0,1) \tag{5.4}$$

where x is a small integer. This has the effect of exaggerating the differences between P_{MAP} values and reducing the acceptance rate. In a standard MCMC application, altering the posterior distribution in such a way would prevent sampling from the chain (similar to the inability to sample from a heated chain in Metropolis-coupled MCMC (Geyer, 1991)). In this application, however, the goal of the MCMC algorithm is to search for optimal subtrees, not to sample from the space of subtrees, so the modification simply has the effect of making the search more efficient.

5.4.2 TA settings

The TA algorithm requires an initial choice of threshold and a decrement value. I tested starting threshold values of 0.1 and 0.2, with decreases of 0.05, 0.025 and 0.02. Either starting value produced acceptance rates in an appropriate range. The number of iterations was 4000 to 5000 per threshold decrement so that the total number of iterations was 20000 (equivalent to the number used for the MCMC method, so that the two algorithms could be fairly compared).

5.5 Results

5.5.1 Phylogenetic inference

The phylogenetic inference results from the five runs are summarized in Table 5.2. Each posterior distribution contained 10000 total trees, with the first 1000 discarded as burn-in based on plots of the log-likelihood and tree length. Changing the burn-in to 5000 did not significantly alter the size of the credible set or the probability of the MAP tree. For the five posterior distributions, the probability of the MAP trees ranged from 0.055 to 0.887 and the size of the 95% credible set from 11 to 434 trees. Run 3 is a narrow, peaked distribution, while Runs 4 and 5 are much more dispersed with a very low probability on the MAP tree. The other two runs fall in between these extremes. The majority rule consensus trees for each run contain at least 4 nodes with uncertain resolution (posterior probability of the clade is less than 100%), and all but Run 3 contain a multifurcation in the MRC tree. These results give a sufficiently variable set of posterior distributions for testing the tree pruning algorithms.

Table 5.2: Summary of phylogenetic inference. For each simulated data set, I report the probability of the MAP tree, the size of the 50%, 90% and 95% credible sets, the total number of unique sampled trees and the number of nodes in the MRC tree with less than 100% posterior probability.

Run	P_{MAP}	Size of credible sets (50,90,95)%	Nodes < 100%	Multifurcations
1	0.206	(23, 34, 58)	4	1
2	0.136	(42, 59, 89)	5	1
3	0.887	(2, 4, 11)	5	0
4	0.055	(131, 178, 267)	6	1
5	0.072	(204, 286, 434)	9	1

Table 5.3: Summary of pruning results from the five posterior distributions of phylogenies.

run	starting P_{MAP}	final P_{MAP}	Taxa removed	Equivalent subtrees
1	0.206	0.9726	5	2
2	0.136	0.6604	15	1
3	0.887	0.9575	4	2
4	0.055	0.6964	15	1
5	0.072	0.9334	15	4

5.5.2 Pruning trees for simulated data

I first present pruning results for the five distributions using the TA algorithm with starting threshold 0.1 and threshold decrement 0.02. For each of the input posterior distributions, the number of best subtrees for a given number of taxa ranges from 1 to 18 (see Table 5.3). Subtrees with $P_{MAP} > 0.95$ were found in only two of the five distributions. Figure 5.3 illustrates detailed results for Run 1, one of the two distributions where the method discovered well-supported subtrees.

It is possible to summarize the taxa present in the subtrees. If the method finds a single subtree, then the output includes that subtree and the list of taxa removed from the tree. If there is more than one equivalent subtree, the output lists the subtrees and the taxa removed to produce each subtree. For multiple subtrees, the output also includes the fraction of subtrees that do not contain that taxa. If a taxa is always absent, then it has lower overall support in the original distribution of phylogenies.

Multiple equivalent subtrees result when different combinations of taxa produce subtrees with the same probability. This is the case when multiple taxa have the

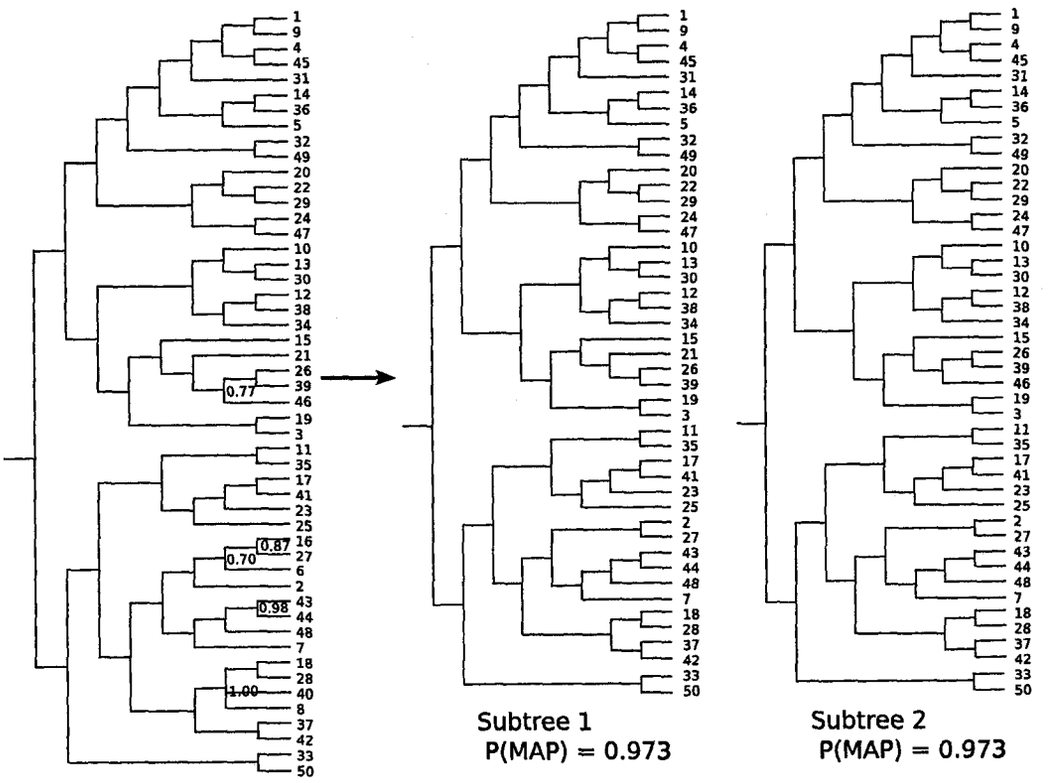


Figure 5.3: Majority rule consensus tree and best agreement subtrees for Run 1. Nodes without explicit posterior probabilities on the MRC tree have probability 1.00. Note the low probability on the original MAP tree (P=0.206), despite relatively high posterior probabilities on the MRC tree. The taxon frequencies indicate the frequency that the taxon is pruned to produce the subtrees.

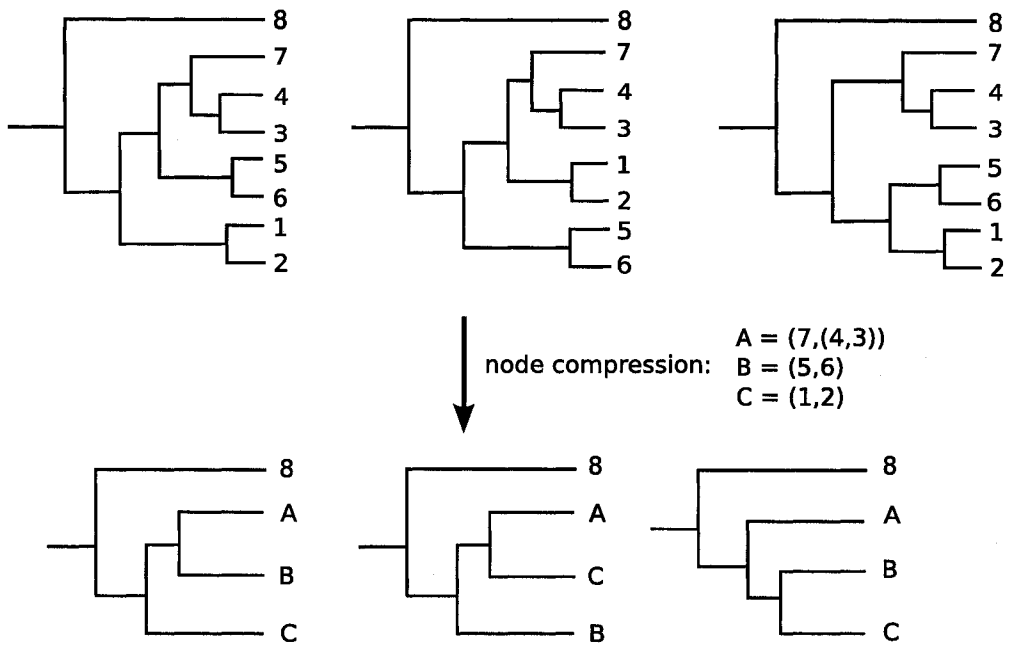


Figure 5.4: Pre-processing step that compresses nodes with posterior probability of 100%.

same resolution within the tree (in terms of the marginal posterior probabilities of clades). This can happen with distantly related taxa, or it can happen for taxa in the same clade when the poor resolution is on the ancestral node. To separate these two issues, I added an optional pre-processing step to the algorithm. Before starting the pruning, this step identifies all internal nodes that have 100% posterior probability on both the node itself and on all descendent internal nodes. Then, it collapses these nodes (and all descendent nodes), replacing them with a single marker node. This is justified because removing a taxon descendent to one of these internal nodes has no effect on the probability of the MAP tree (removing such a taxon cannot collapse any of the input trees, since 100% of the input trees contain the same pattern). An example of this strategy is shown in Figure 5.4. The pre-processing step also greatly increases the speed of the algorithm, since it excludes taxa from the search that cannot improve P_{MAP}. The disadvantage is that this makes the results more difficult to summarize, as the marker nodes are treated as one taxa, while we are in reality removing the entire clade (such that the output trees may not be the same size). Weighting the nodes seems like a possible solution, but this makes the proposal strategy more complicated.

I suggest performing an initial search with a relatively small number of iterations and a large upper limit on k, the number of taxa removed. From these

Table 5.4: Comparison of MCMC and TA algorithms for five posterior distributions of trees. The k value is the largest number of taxa removed for that run. For each algorithm, the table lists the optimum P_{MAP} found by each algorithm. The number in brackets is the number of subtrees with the given probability. The best result is the one that first maximizes P_{MAP} and then the number of subtrees. The last row summarizes the number of times that algorithm gave the best results. Details about settings for each algorithm are in the text.

run	k	MCMC1	MCMC2	MCMC3	TA1	TA2	TA3
1	5	0.919 (1)	0.963 (1)	0.973 (1)	0.973 (2)	0.973 (3)	0.973 (2)
2	10	0.557 (1)	0.623 (1)	0.623 (1)	0.623 (18)	0.623 (1)	0.628 (4)
3	5	0.955 (1)	0.945 (3)	0.955 (1)	0.955 (2)	0.955 (2)	0.958 (1)
4	10	0.489 (1)	0.489 (1)	0.528 (1)	0.586 (3)	0.525 (1)	0.571 (1)
5	10	0.698 (1)	0.698 (1)	0.639 (1)	0.862 (1)	0.682 (1)	0.782 (1)
all		0	0	0	2	1	2

results, determine a smaller range of k values that give probabilities near the desired range. Then, perform an intensive search, with a larger number of iterations in the smaller range of k values. The algorithm should run until no further changes are observed in the optimal P_{MAP} or the number of equivalent subtrees.

5.5.3 Comparison of algorithms

I compared the performance of the MCMC and TA algorithms, using each posterior distribution of trees and the described range of implementation parameters for each method. Performance was judged based on three nested criteria: 1. the maximum P_{MAP} tree found; 2. the number of subtrees with this probability; and 3. the number of iterations required to find the optimal solution. The best method found subtrees with the highest P_{MAP} and the largest number of equivalent subtrees in the smallest number of iterations. The results are summarized in Table 5.4. The TA algorithm was far more consistent in both finding an optimal P_{MAP} and finding the largest number of subtrees with that P_{MAP}. Of the five runs, the TA algorithm found the optimal solution in all five cases. The time requirement for each of the algorithms was very similar (data not shown).

For $k = 3$ and $k = 4$ (where k = number of removed taxa), I also compared the two algorithms with the results from an exhaustive search. This gave 5 runs * 2 values $k = 10$ comparisons. In 9 out of 10 comparisons, both stochastic search methods found an optimal P_{MAP} equal to that from the exhaustive search, measured to 4 significant digits. In these 9 comparisons, MCMC found all of the equivalent subtrees in 8 comparisons, while TA found all of the equivalent subtrees in all comparisons. Table 5.5 details these results.

Table 5.5: Comparison of stochastic and exhaustive search strategies for the five posterior distributions of trees. Similar to Table 5.4, each cell contains the optimal P_{MAP} and the number of equivalent subtrees in brackets. For each algorithm, the table lists the optimum P_{MAP} found by each algorithm.

run	k	Exhaustive	MCMC	TA
1	3	0.7000 (1)	0.7000 (1)	0.7000 (1)
1	4	0.9143 (2)	0.9143 (2)	0.9143 (2)
2	3	0.3946 (4)	0.3946 (4)	0.3946 (4)
2	4	0.5154 (2)	0.5154 (2)	0.5154 (2)
3	3	0.9405 (2)	0.9405 (2)	0.9405 (2)
3	4	0.9454 (2)	0.9452 (1)	0.9435 (2)
4	3	0.2987 (2)	0.2987 (2)	0.2987 (2)
4	4	0.3883 (4)	0.3883 (4)	0.3883 (4)
5	3	0.2578 (8)	0.2578 (8)	0.2578 (8)
5	4	0.3303 (8)	0.3303 (6)	0.3303 (8)

5.5.4 Empirical data

The original phylogenetic analysis of the Carnivora data set produced a consensus tree with good resolution between most of the major groups but with a lack of resolution within groups(Fulton and Strobeck, 2006). Figure 5.5 shows the well-resolved relationship between the family-level groups. The posterior distribution of phylogenies contained 20000 total sampled trees from 2 MCMC chains, with the first half of each run discarded as burn-in. The MRC tree contains 5 multifurcations and 13 nodes with marginal posterior probability less than 0.95. The posterior distribution of phylogenies was very flat, with a probability of the MAP tree equal to 0.001 and the 50, 90 and 95% credible sets containing 3471, 7472 and 7972 trees each.

Exploratory pruning analysis (with various threshold annealing parameters and a small number of iterations) indicated that subtrees with probablities near 50% could be found by pruning approximately 30 taxa from the tree. This is a large percentage of the total taxa, but is to be expected given the very broad initial distribution. I then performed a more extensive analysis, with total number of pruned taxa ranging from 25 to 35 and a larger number of iterations. This data set was much more sensitive to the TA settings for initial threshold and increment than were the simulated data sets. With an initial P_{MAP} of 0.0006, the starting threshold needed to be 0.01 or lower for reasonable results. I performed eight separate analysis with starting thresholds ranging from 0.005 to 0.02 and increments that were 0.1% of the starting threshold.

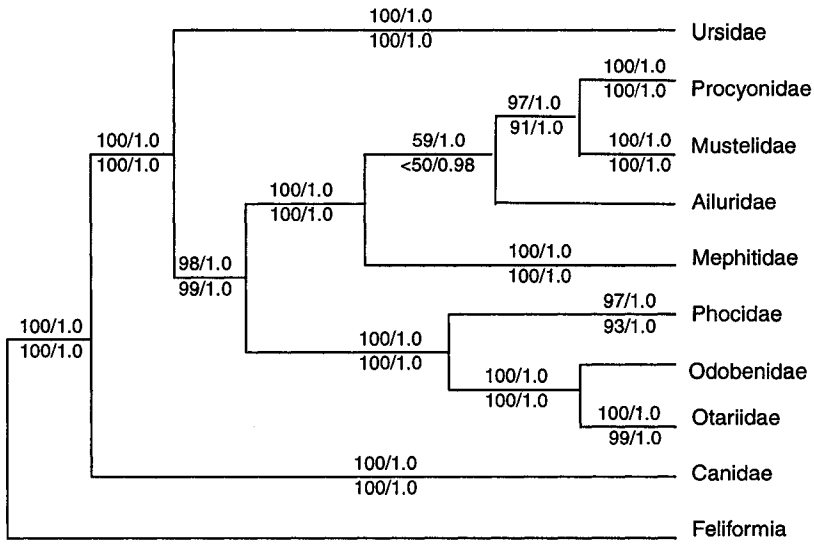


Figure 5.5: Family-level subtree for the 85-taxa Carnivora phylogeny. The second number above each branch is the posterior probability for the full data set (the first number if the bootstrap proportion from ML analysis, and below the branches are the support values for a second analysis with fewer genes). Reproduced from Fulton and Strobeck (2006).

The best result was a subtree with $P_{MAP} = 0.9558$ after the removal of 28 taxa (leaving a tree with 57 taxa). In three independent runs, MAPminer found seven unique subtrees of size 57 with probabilities greater than 0.95. The original tree and the subtrees are shown in Figure 5.6. Of the 85 original taxa, there were 19 that were always absent from the subtrees and 50 that were present in all of the high probability subtrees. The 16 remaining taxa were present in some but not all of the seven subtrees. The 19 'always absent' taxa are definite candidates for further sequencing efforts or for removal from the data set before additional phylogenetic inference.

This empirical data set illustrates an extreme case. The starting posterior distribution of phylogenies was extremely flat, with very low probability on the MAP tree and several thousand trees in the 90% credible set. As could be expected, the discovery of a well-supported skeleton tree within the distribution required the removal of a relatively large number of taxa compared to the results I saw in the simulated data sets. However, the algorithm was still able to discover several very well-supported subtrees containing 2/3 of the original taxa in the data set.

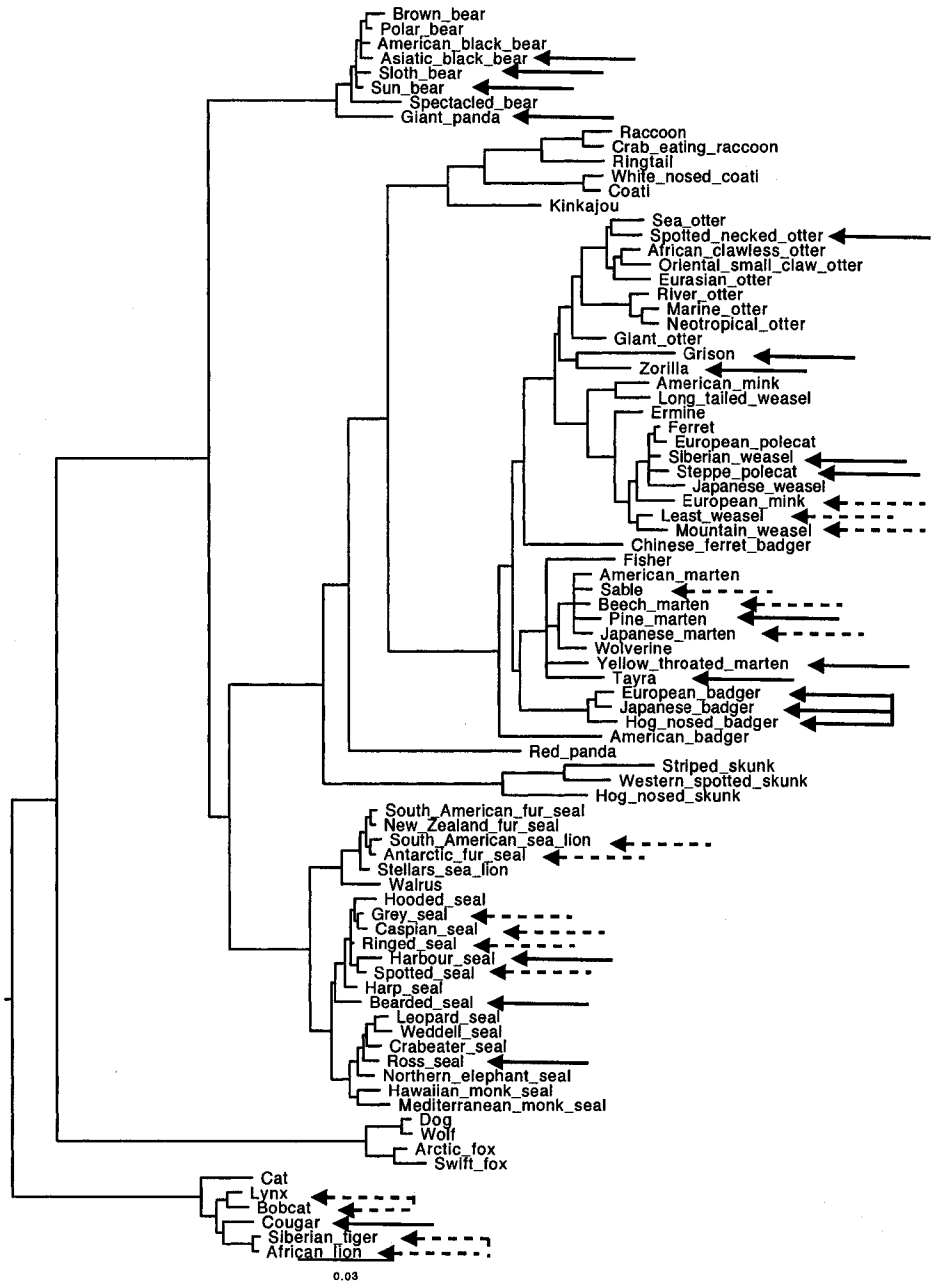


Figure 5.6: MRC tree of the Carnivora Solid arrows mark taxa that are removed in all 57-taxon subtrees with probability greater than 0.95. Dashed arrows mark taxa marked that are removed from some, but not all of the subtrees. Joined arrows indicate groups with 100% posterior probability, but uncertain position of the group as a whole on the tree.

5.6 Discussion

One of the advantages of Bayesian inference is that it produces, not just a point estimate, but a full posterior distribution for the parameters of interest. The posterior probability of particular parameter value (or particular phylogeny) gives a mathematically well-defined and intuitive measure of the support for that value. One obvious summary statistic for a distribution trees is then the most probable tree, the MAP tree. Another commonly used summary is the majority-rule consensus tree. In either case, relying solely on the a single point estimate does not adequately describe the full posterior distribution. The MAP tree often has low overall probability. When reporting a MRC tree, high partition probabilities on a majority of internal nodes in the MRC tree do not imply that there is a strongly supported MAP tree and a narrow distribution of trees. Even with most partition probabilities approaching 100%, there may still be a very large number of unique trees in the credible set.

The ideal result for a Bayesian phylogenetic inference would be a single well-supported tree, defined by a high posterior probablity. In reality, many data sets return a large credible set of trees and no single tree with high probability. The tree pruning method provides a list of the largest well-supported subtrees that exist within the posterior distribution of phylogenies. The quality of the subtrees is determined by the sum of the posterior probabilities of the input trees that agree with a given subtree. By pruning taxa from the input trees, we can search for optimal agreement subtrees and produce a modified posterior distribution of phylogenies with narrower credible sets and higher probabilities on the MAP trees. The MAP tree (the most probable tree) is the most natural point estimate for summarizing a posterior distribution and is more natural in a Bayesian context than using consensus trees. Bayesian methods provide us with probabilities for entire trees, while the use of consensus techniques causes the whole-tree probabilty to be unreported, or lost.

I implement and compare two different algorithms for the subtree search. The MCMC methodology is likely familiar to most users of Bayesian inference, with proposed solutions accepted in proportion to a Metropolis-Hastings ratio of the proposed and current objective functions. The second algorithm is the threshold accepting (TA) algorithm, which accepts proposed solutions that are within a certain threshold of the current solution, and then progressively lowers the threshold until it searches only in the region of the optimal solution. It too uses a Metropolis acceptance step, but with a different objective function and probability of acceptance. TA is strictly an optimization algorithm, rather than a sampling algorithm that can provide a picture of the underlying distribution. The

performance of the TA algorithm was superior, both when compared directly to the MCMC algorithm and when both methods were compared to an exhaustive search (the "true" result).

The possible outcomes of this type of analysis are threefold. First, the method can produce a single subtree with the specified probability, meaning that the posterior distribution of trees contains one well-supported skeleton tree and a unique set of taxa to prune. Second, there may be multiple subtrees with the same probability (or very similar probabilities). This result can occur when resolving multifurcations, or when removing taxa from a clade where the unresolved nodes are deeper in the tree, so that resolution involves removing entire clades of taxa rather than individual taxa. A pre-processing node compression step can simplify the pruning in such situations. Finally, the method may reach the upper limit on the number of taxa to remove and exit without finding any well-supported subtrees. There can be two reasons for this result. The data may simply not be informative enough to support even a subtree within the distribution, in which case a re-evaluation of the input data may be required. The other possibility is non-convergence of the Markov chain Monte Carlo in the original phylogenetic inference. I encourage users to ensure that the phylogenetic inference method has converged with respect to the log likelihood, model parameters and other output.

Rather than use the probability of the MAP tree as an end point, it would also be possible to use the size of the credible set. We would then run the analysis until the number of trees in the 95% credible set was less than a specified limit (or until we reached the maximum number of allowed pruned taxa). This may be a more useful strategy if the results from the Bayesian phylogenetic inference are being used in a program that takes a set of trees as input.

This pruning method shares some properties with the Reduced Consensus methods (Wilkinson, 1994, 1996) for improving bootstrap values on trees. The Reduced Consensus methods create a profile of subtrees based on common n-taxon statements (rooted trees) or partitions (unrooted trees) in the original set of trees. The original method (Wilkinson, 1994) was strict, requiring agreement between all input trees, but a later majority-rule method (Wilkinson, 1996) allowed less than 100% bootstrap support on the subtrees. In contrast, my tree pruning method operates with entire subtrees, which are more informative than n-taxon statements or partitions. This was previously recognized as a better solution (Sanderson and Schaffer, 2002). Using entire subtrees means that we can place support values both on the full tree and on partitions within the tree. In addition, the methods proposed by Wilkinson have "quite severe limitations on the numbers of taxa and numbers of trees that can be analyzed" (Wilkinson, 1996). MAPminer can accept thousands of

input trees with at least 100 taxa. Also, by limiting the output to the largest agreement subtrees, we avoid the problem of exponential growth of the number of trees in the Reduced Consensus profile with increasing number of taxa in the input trees (Bryant, 1997).

I want to emphasize the important distinction between performing a phylogenetic analysis without a given taxon and the post-analysis pruning method described here. The addition of taxa to a phylogenetic inference problem is known to improve accuracy of the inference (Rannala et al., 1998; Greybeal, 1998; Zwickl and Hillis, 2002). Studies by (Rosenberg and Kumar, 2001) and (Pollock et al., 2002) explicitly compare the accuracy of trees inferred from a subset of taxa with pruned trees derived from the inference of the full set of taxa. Although the magnitude of the effect is disputed (Rosenberg and Kumar, 2003; Hillis et al., 2003), the studies do indicate that the pruned trees have lower error rates than trees analyzed with only a subset of the data.

While there is information gained from each taxon in the original phylogenetic inference, the inclusion of some taxa may disproportionately complicate the post-run analysis. This may be particularly worrisome if the added taxa are not the ones of greatest interest to the study. For example, taxa may have been added in an attempt to break up long branches, or simply because the sequences were available. As the number of sequences in public databases continues to grow, it is ever easier to use larger taxon sets to infer trees. This type of post-inference summary allows all available taxa to be included in the original analysis. The summary method can then identify problematic taxa and rank these taxa according to their instability on the input trees. Information about specific taxa can be used to direct future efforts, for example, obtaining additional sequence for the most unstable taxa.

My aim in this study is to present a novel method for summarizing the posterior distribution of phylogenies and to encourage developers and users of Bayesian phylogenetic inference to investigate a variety of methods. A summary method may simply involve reducing the distribution to one single tree as a point estimate, such as the MAP tree or the MRC tree. In contrast, we can summarize the entire distribution in a network structure, which retains every partition relationship present in the full distribution at the expense of a more complex interpretation (for example, Huson and Bryant, 2006). Between these two extremes, there is great potential for other methods which balance simplicity of interpretation with maximal information content in ways that are appropriate to the desired application of the phylogenetic results.

The MAPminer method, based on frequent agreement subtrees within the posterior distribution, provides individual well-supported binary trees than can be

easily reported or input into other software packages for secondary analyses. A posterior distribution with wide credible sets requires a larger number of taxa to be pruned from the trees in order to discover a well-supported agreement subtree within the distribution. The absence of well-supported subtrees indicates a lack of information in the posterior distribution of phylogenies. One benefit of this particular post-run analysis is that it allows the original inference of the phylogeny to proceed with all of the available data, yet allows the summary to contain only the results that describe well-supported binary trees.

Bibliography

Amir, A. and D. Keselman. 1997. Maximum agreement subtree in a set of evolutionary trees: Metrics and efficient algorithms. SIAM J. Comput. 26:1656–1669.

Bryant, D. 1997. Building trees, hunting for trees and comparing trees: Theory and methods in phylogenetic analysis. Ph.D. thesis University of Canterbury.

Chi, Y., R. R. Muntz, S. Nijssen, and J. H. Kok. 2005. Frequent subtree mining - an overview. Fundamenta Inform. 66:161–198.

Dueck, G. and T. Scheuer. 1990. Threshold accepting: A general purpose optimization algorithm. J. Comput. Phys. 90:161–175.

Finden, C. R. and A. D. Gordon. 1985. Obtaining common pruned trees. J. Classif. 2:255–276.

Fulton, T. L. and C. Strobeck. 2006. Molecular phylogeny of the Arctoidea (Carnivora): Effect of missing data on supertree and supermatrix analyses of multiple gene data sets. Mol. Phylogenet. Evol. 41:165–181.

Geyer, C. 1991. Markov chain Monte Carlo maximum likelihood. Pages 156–163 *in* Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface (E. Keramidas, ed.). Interface Foundation, Fairfax Station.

Greybeal, A. 1998. Is it better to add taxa or characters to a difficult phylogenetic problem? Syst. Biol. 47:9–17.

Hewitt, G. M. 2001. Speciation, hybrid zones and phylogeography - or seeing genes in space and time. Mol. Ecol. 10:537–549.

Hillis, D. M., D. D. Pollock, J. A. McGuire, and D. J. Zwickl. 2003. Is sparse taxon sampling a problem for phylogenetic inference? Syst. Biol. 52:124–126.

Huelsenbeck, J. P. and F. Ronquist. 2001. MrBayes: Bayesian inference of phylogenetic trees. Bioinformatics 17:754–755.

Huson, D. and D. Bryant. 2006. Application of phylogenetic networks in evolutionary studies. Mol. Biol. Evol. 23:254–267.

Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi. 1983. Optimization by simulated annealing. Science 220:671–680.

Knowles, L. L. and W. P. Maddison. 2002. Statistical phylogeography. Mol. Ecol. 11:2623–2635.

Pollock, D. D., D. J. Zwickl, J. A. McGuire, and D. M. Hillis. 2002. Increased taxon sampling is advantageous for phylogenetic inference. Syst. Biol. 51:664–671.

Rannala, B., J. P. Huelsenbeck, Z. Yang, and R. Nielsen. 1998. Taxon sampling and the accuracy of large phylogenies. Syst. Biol. 47:702–710.

Ronquist, F. and J. P. Huelsenbeck. 2003. MrBayes 3: Bayesian phylogenetic inference under mixed models. Bioinformatics 19:1572–1574.

Rosenberg, M. and S. Kumar. 2001. Incomplete taxon sampling is not a problem for phylogenetic inference. Proc. Natl. Acad. Sci. USA 98:10751–10756.

Rosenberg, M. and S. Kumar. 2003. Taxon sampling, bioinformatics, and phylogenomics. Syst. Biol. 52:119–124.

Sanderson, M. J. 2002. Estimating absolute rates of molecular evolution and divergence times: a penalized likelihood approach. Mol. Biol. Evol. 19:101–109.

Sanderson, M. J. and H. B. Schaffer. 2002. Troubleshooting molecular phylogenetic analysis. Ann. Rev. Ecol. Syst. 33:49–72.

Shasha, D., J. Wang, and S. Zhang. 2004. Unordered tree mining with applications to phylogeny. Pages 708–719 *in* Data Engineering, 2004. Proceedings. 20th International Conference on.

Thorne, J. L., H. Kishino, and I. S. Painter. 1998. Estimating the rate of evolution of the rate of molecular evolution. Mol. Biol. Evol. 15:1647–57.

Wilkinson, M. 1994. Common cladistic information and its consensus representation: Reduced Adams and reduced cladistic consensus trees and profiles. Syst. Biol. 43:343–368.

Wilkinson, M. 1996. Majority-rule reduced consensus trees and their use in bootstrapping. Mol. Biol. Evol. 13:437–444.

Yang, Z. 1997. PAML: a program package for phylogenetic analysis by maximum likelihood. Comput Appl Biosci 13:555–6.

Yang, Z., W. S. W. Wong, and R. Nielsen. 2005. Bayes empirical Bayes inference of amino acid sites under positive selection. Mol. Biol. Evol. 22:1107–1118.

Zwickl, D. and D. Hillis. 2002. Increased taxon sampling greatly reduces phylogenetic error. Syst. Biol. 51:588–598.

Chapter 6

Conclusions and future directions

This thesis introduces a novel algorithm for proposing trees in Bayesian phylogenetic Markov chain Monte Carlo (MCMC), tests this algorithm against existing methods, assesses the utility of various convergence diagnostics for MCMC output and finally, develops a novel method to summarize the phylogenies sampled during an MCMC analysis.

When assessing convergence of Bayesian phylogenetic MCMC methods, numerical output parameters converge faster than partition probabilities, which converge faster than the distribution of phylogenies. For many data sets, the phylogenies are not likely to converge to a stable posterior distribution, due to the sheer size of the sample space of reasonable trees. Our estimates of the topology, as measured through variability of partition probabilities between chains, do not seem to be overly sensitive to a lack of convergence of the full posterior with respect to entire trees.

Numerical convergence diagnostics, such as the Potential Scale Reduction Factor, Raftery-Lewis and Heidelberger-Welch tests (Gelman and Rubin, 1992; Raftery and Lewis, 1992; Heidelberger and Welch, 1981), are useful to detect burn-in, confirm stationarity and examine mixing. Estimation of the burn-in phase using these diagnostics generally gives results that are consistent with the common practice of using the log likelihood time series plot to judge the length of the burn-in. It is prudent to also check diagnostics calculated using the tree length, as these can be significantly different than the log likelihood results for some data sets. Stationarity testing using numerical diagnostics was more sensitive than time series analysis, and tests based on multiple chains were more sensitive than those based on a single MCMC chain.

The partition-based statistics (RMSET and MeanSD) are much more sensitive to the convergence of the topology and branch lengths than are the standard MCMC output parameters (log likelihood and tree length). The values of these statistics continue to decline long after the whole-tree parameters have reached their stationary distribution, indicating that a long sampling phase is required after burn-in of the MCMC chains.

The quality of the sampling phase is dependent on the mixing behaviour of the chain, and autocorrelation between samples is one measure of this behaviour. Autocorrelation in phylogenetic MCMC is significant, likely due to the size of the phylogeny parameter and the fact that only a portion of this parameter is modified at each iteration of the chain. Larger trees displayed higher autocorrelation, which is consistent with an increasing portion of the tree remaining unchanged between iterations. Smaller moves sizes and higher acceptance rates were associated with lower levels of autocorrelation.

The BranchSlide algorithm introduced here is an extremely tunable proposal algorithm that can be parameterized to behave as either a local or global proposal method. The choice of move size from the Normal distribution means that the algorithm can generate both small and large moves without any change in the tuning parameter, which explains the improved mixing over solely large or small proposals. The move size is also dependent on the branch lengths of the tree, so a small distance may still induce a topology move in regions of poor resolution (with short branch lengths). The challenge with an algorithm such as BranchSlide is the choice of tuning parameter, which is dependent on both the tree length and the difficulty of the inference problem. This algorithm could easily be extended through a change to the distribution used to select the move size. For example, we could use a mixture of Normal distributions, with the mean of one distribution closer to zero and one farther away (the second distribution giving larger moves, on average).

For informative data sets, Bayesian phylogenetic MCMC methods are very robust to changes in the methods for proposing new trees. For more challenging inference problems, however, the proposal strategy can greatly affect both convergence and mixing. The best algorithm for fast convergence is not the same as the method for optimal mixing, and the difference between the optimal methods seems to increase with the difficulty of the inference problem. One strategy used to improve mixing and convergence in phylogenetics is Metropolis-coupled Markov Chain Monte Carlo (MCMCMC, or MC^3) (Geyer, 1991; Huelsenbeck and Bollback, 2001). In this study, however, simple MCMC with large proposals was more effective than MCMCMC with the most challenging data set. Although a larger number of examples are required, this result does agree with recent papers that question the

efficiency of MCMCMC methods (Beiko et al., 2006; Pagel et al., 2004).

The lack of convergence of the posterior distribution of phylogenies does not indicate an absence of information in the distribution, as evidenced by good estimates of the partition probabilities, even with an extremely flat posterior distribution of topologies. This lead to the development of a novel summary methods for the posterior distribution of phylogenies that aimed to extract one or more well-supported subtrees from within the distribution. By relaxing the requirement of strict agreement for maximum agreement subtrees, I was able to extract entire topologies with very high posterior probability from a starting distribution that did not contain any trees with high probability. This method is useful for the presentation of results from a Bayesian analysis, and also helps to identify taxa as candidates for further sequencing efforts.

A large component of the research described in this thesis was the development of a novel software package for Bayesian phylogenetic inference. Appendix A describes the implementation of this software. Despite the popularity of MrBayes, the existence of this and other software packages for Bayesian inference (see Table 6.1) points to a need for tools to enable scientists to develop their own software implementations. While it is possible to modify the source code of another package, a better solution would be to collaborate on the development of programming classes for the basic functionailty of phylogenetic inference (refer to Appendix A for a description of object-oriented programming and the class structure). There are two advantages to this sort of collection. Classes can be easily combined to create new functionality without the need to write code for the basic structure of the program. For example, there are certainly hundreds, if not thousands, of different programming implementations for the set of common evolutionary models. This is time that could be better spent developing software for novel theoretical developments. The second advantage would be higher confidence in the mathematical and programming accuracy if there were multiple authors and users contributing to the development. The PAL Java library (Drummond and Strimmer., 2001) is one step in this direction, and the forthcoming release of the object-oriented MrBayes 4.0 is another.

The advent of very fast maximum likelihood methods such as PHYML, RAxML and Garli (Guindon and Gascuel, 2003; Stamatakis et al., 2005; Zwickl, 2006) may have an effect on the future use of Bayesian methods. While the performance of Bayesian methods is good for smaller trees (Williams and Moret, 2003), for large trees, these ML methods greatly outperform in terms of speed (measuring inference of 100-200 taxon trees in terms of seconds). This makes them very attractive for systematists who simply want to infer a single tree for large data sets. In contrast,

Table 6.1: A list of available software packages for Bayesian inference of phylogenies.

Name	Reference
BAli-Phy	Suchard and Redelings (2006)
BAMBE	Simon and Larget (2000)
BayesPhylogenies	Pagel and Meade (2004)
MrBayes	Ronquist and Huelsenbeck (2003)
mcmctree	Yang (2002)
PhyBayes	Aris-Brosou and Yang (2002)

Bayesian methods are not ideally suited to simply finding a point estimate of the phylogeny (although they have often been used for this purpose). The advantages of integrating over the posterior distribution of phylogenies are the ability to quantify the uncertainty using the posterior probabilities and the ability to simultaneously infer other evolutionary parameters. This allows uncertainly in the phylogeny to be incorporated into our estimates of other parameters (and, conversely, allows the uncertainty in their distribution to inform the inference of the phylogeny). Bayesian phylogenetic methods are likely to become increasingly important in areas involving the application of distributions of trees to other questions about evolution. For example, concurrent inference of alignment and phylogeny is well-suited to Bayesian methods (Redelings and Suchard, 2005), as is the estimation of divergence times (Drummond et al., 2006; Rannala and Yang, 2003).

As novel applications in phylogenetics continue to proliferate and the size of data sets increases, it is imperative that we continue research on the basic functionality of Bayesian methods, including choice of prior distributions, proposal algorithms, MCMC convergence analysis, effect and detection of mixture models and optimization of partitioning schemes.

Bibliography

Aris-Brosou, S. and Z. Yang. 2002. Effects of models of rate evolution on estimation of divergence dates with special reference to the Metazoan 18S ribosomal RNA phylogeny. Syst. Biol. 51:703–714.

Beiko, R. G., J. M. Keith, T. J. Harlow, and M. A. Ragan. 2006. Searching for convergence in phylogenetic Markov chain Monte Carlo. Syst. Biol. 55:553–565.

Drummond, A. and K. Strimmer. 2001. PAL: An object-oriented programming library for molecular evolution and phylogenetics. Bioinformatics 17:662–663.

Drummond, A. J., S. Y. W. Ho, M. J. Phillips, and A. Rambaut. 2006. Relaxed phylogenetics and dating with confidence. PLoS Biol 4:e88.

Gelman, A. and D. Rubin. 1992. Inference from iterative simulation using multiple sequences. Stat. Sci. 7:457–72.

Geyer, C. 1991. Markov chain Monte Carlo maximum likelihood. Pages 156–163 *in* Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface (E. Keramidas, ed.). Interface Foundation, Fairfax Station.

Guindon, S. and O. Gascuel. 2003. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. Syst. Biol. 52:696–704.

Heidelberger, P. and P. Welch. 1981. A spectral method for confidence interval generation and run length control in simulations. Commun. ACM 24:233–245.

Huelsenbeck, J. P. and J. P. Bollback. 2001. Empirical and hierarchical Bayesian estimation of ancestral states. Syst. Biol. 50:351–366.

Pagel, M. and A. Meade. 2004. A phylogenetic mixture model for detecting pattern-heterogeneity in gene sequence or character-state data. Syst. Biol. 53:571–581.

Pagel, M., A. Meade, and D. Barker. 2004. Bayesian estimation of ancestral character states on phylogenies. Syst. Biol. 53:673–684.

Raftery, A. and S. Lewis. 1992. [Practical Markov Chain Monte Carlo]: Comment: One long run with diagnostics: Implementation strategies for Markov Chain Monte Carlo. Statictical Science 7:493–497.

Rannala, B. and Z. Yang. 2003. Bayes estimation of species divergence times and ancestral population sizes using DNA sequences from multiple loci. Genetics 164:1645–1656.

Redelings, B. D. and M. A. Suchard. 2005. Joint Bayesian estimation of alignment and phylogeny. Syst. Biol. V54:401–418.

Ronquist, F. and J. P. Huelsenbeck. 2003. MrBayes 3: Bayesian phylogenetic inference under mixed models. Bioinformatics 19:1572–1574.

Simon, D. and B. Larget. 2000. Bayesian analysis in molecular biology and evolution (BAMBE), version 2.03 beta. Department of Mathematics and Computer Science, Duquesne University.

Stamatakis, A., T. Ludwig, and H. Meier. 2005. RAxML-III: a fast program for maximum likelihood-based inference of large phylogenetic trees. Bioinformatics 21:456–463.

Suchard, M. A. and B. D. Redelings. 2006. BAli-Phy: simultaneous Bayesian inference of alignment and phylogeny. Bioinformatics 22:2047–2048.

Williams, T. and B. Moret. 2003. An investigation of phylogenetic likelihood methods. *in* Proceedings of 3rd IEEE International Symposium on Bioinformatics and Bioengineering (BIBE'03), Bethesda, MD.

Yang, Z. 2002. Phylogenetic analysis by maximum likelihood (PAML). version 3.13.

Zwickl, D. J. 2006. Genetic algorithm approaches for the phylogenetic analysis of large biological sequence datasets under the maximum likelihood criterion. Ph.D. thesis The University of Texas at Austin.

Appendix A

A Bayesian phylogenetic inference package

This Appendix gives a brief description of BayesTrees, the Bayesian phylogenetics inference package used throughout this thesis. The BayesTrees package includes three programs. The first is BayesTrees itself, which performs Bayesian phylogenetic inference. The second is TreeSum, which calculates the topology-based diagnostics used in Chapter 3. Finally, MAPminer implements the frequent subtree mining method described in Chapter 5. The entire package is written using object-oriented C++, so I first introduce the object-oriented programming concept.

A.1 Object-oriented programming

BayesTrees is written in C++, using an object-oriented programming (OOP) approach. In this style of programming, the code is compartmentalized into *classes* which contain groups of related variables and methods (functions). For example, the **Model** class contains all of the variables and methods for defining the evolutionary model, proposing new states for model parameters and calculating transition probabilities. To use a class, you create an *object* of that class. Upon creation of the object, a *constructor* method initializes all of the variables in the class and a *destructor* deletes any used memory when the object is no longer needed. Objects can come in and out of memory as they are needed and not needed during run-time. Compartmentalization of the code makes sharing code easier, makes the code more readable and reduces duplication.

The class can contain both private and public member variables and methods. This allows the programmer to isolate aspects of the class from the user (where the user can be another class). For example, the **Node** class contains pointers to the left, right and ancestral nodes as well as the branch lengths between these nodes.

These variables are *private* and can be accessed only through *public* methods. If we try to set an illegal branch length or change a branch between two nodes that are not connected, the public *changeLength* method performs error checking and deals with any illegal conditions, rather than allowing the user of the class to directly set the Node pointers or branch length field. There is no need to remember to perform the error checking when writing new code that changes a branch on the tree - the only way to perform this operation is through the public method in **Node**.

A class can be completely independent, or can contain other objects (classes that contain other classes are an example of *composition*). For example, the **Tree** class includes a collection of **Node** objects, and the **Phylogeny** class contains a **Tree**, a **Model** and a **Data** object. Figure A.1 describes the classes in BayesTrees and their relationships.

OOP also includes the idea of *inheritance*, where we create a new class using an existing class as a template. If there is a class whose methods you want to use, but you want additional functionality, you can create a new class that is derived from the base class. The derived class can use all of the functions of the base class, and the code added to the derived class does not affect the base class. Examples in BayesTrees include the **Tip** and **Internal** classes, which are derived classes of **Node**, and each of the tree rearrangement classes, which are derived from a generic **Proposal** class. This allows much greater efficiency when writing new implementations. Inheritance keeps the base class clean, which is particularly important if there are multiple developers on a project. The same base class can be used for multiple, unrelated projects through the creation of different derived classes.

There are a number of other standard relationships available to link classes together and improve their interaction, known as *Design Patterns* (Gamma et al., 1995). For example, I utilize a Factory pattern for generating tree rearrangement algorithms. The **Phylogeny** class simply requests an algorithm, and the **AlgorithmFactory** class returns the appropriate proposal method based on the specified type and the prior on branch lengths. The **Phylogeny** class simply receives a generic **Proposal** and can call the generic methods *rearrange* or *changeTuning* without having to know about the specific implementation. This makes it simple to add an additional tree proposal - there are no changes to the code in the **Proposal**, **Tree**, **Phylogeny** or **MCMC** classes.

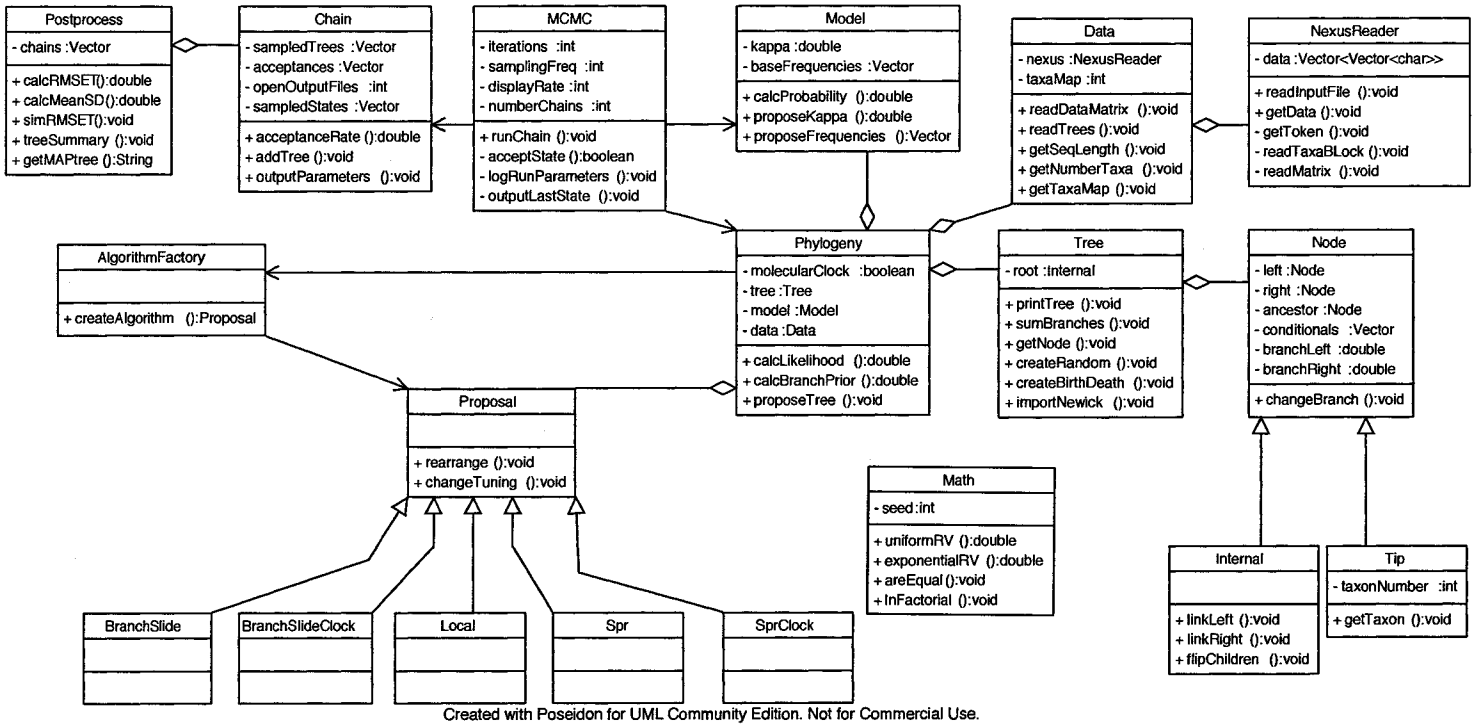


Figure A.1: The BayesTrees classes and their relationships. Each box is a class with the class name, list of sample variables and list of sample methods. Not all variables and methods are included, simply a few to give an idea of the function of the class. Arrows indicate relationships between the classes. Open triangles indicate inheritance, while diamonds indicate composition. The other arrows simply indicate that the classes communicate with each other. The Math class is a utility class that is called by nearly every other method (but adding these arrows would have made the diagram unwieldy).

Another common pattern is the Singleton, which ensures that only a single object of the class is in existance at any one time. This is useful for the **Math** class, since we can initialize the seed for random number generated once, then not worry that it is reset every time we try to create a new **Math** object. Singletons can also reduce memory consumption, since we create one instance of a class and reuse that instance rather than creating multiple instances. **Model, Math, Data** and all of the tree rearrangement classes are all Singletons.

An good example of object-oriented techniques is the Nexus Class Library (NCL) (Lewis, 2003), widely used to read data from Nexus files in BayesTrees, MrBayes, BEAST and others. Another is the PAL library (Drummond and Strimmer., 2001), which is a collection of classes for phylogenetics and evolutionary biology written in the Java programming language.

A.2 BayesTrees

BayesTrees reads data in Nexus format (Maddison et al., 1997). It currently accepts only nucleotide data. Missing characters and gaps are treated equivalently (as the unknown character 'N' with conditional likelihoods of 1.0 for all four nucleotide states). Avaiable tree proposal methods include the BranchSlide, Local and SPR algorithms described in Chapter 2. Currently, only a small number of evolutionary models are implemented, including Jukes-Cantor, F81, K2P and HKY85.

The program runs from the command line and is non-interactive. It reads analysis parameters from a BayesTrees block located in a parameter file provided at startup. The package includes a Perl script that prompts the user for options and creates the parameter file containing the BayesTrees block in the appropriate format. A sample block is as follows:

```
begin BayesTrees;
iterations      1000000          Number of MCMC iterations
sampling        100              Sampling frequency
setup           0                Number of setup iterations, when setting
                                 tuning parameters dynamically
displayRate     1000             Frequency to display current state to stan-
                                 dard output
numberchains    1                Number of independent MCMC chains
seed            random           Seed for random number generation
algorithm       b                Proposal algorithm (branchslide,local,spr)
tuning          1                Tuning parameter for proposal
sites           0 0              Start and end sites
model           4                Model type
kappa           2                Fixed value of kappa
baseFreqs       empirical        Estimate base frequencies from data
laststate       false            Start from where a previous run left off?
prior           exponential 10   Branch length prior
extension       none             File extension for output files
importtree      true             Import at starting tree?
startingtree    tree.txt         File containing start tree
end BayesTrees;
```

To run BayesTrees, type the following at the command line:

```
./BayesTrees datafile parameterfile
```

The input files do not have to be in the current directory, but if they are not, you must provide either the full or relative path to the files. While running, BayesTrees creates a set of output file. For each chain, there are two files for MCMC output, one for the phylogenies (`*.trees`) and one for the other sampled states, such as log likelihood, tree length, model parameters, as well as the acceptance rates (`*param.txt`). Other output includes a `*log.txt` file with the name of the data file and the analysis parameters, which is useful for future reference. A `*lastState.txt` file stores the last state for all analysis parameters, so that the run can be restarted (for example, if additional iterations are judged to be needed due to a lack of convergence).

The BayesTrees package utilizes the GNU Scientific Library (GSL) (Galassi, 2006) for mathematical functions such as random number generators, probability density functions, floating point comparisons and factorials. It uses the NCL to read Nexus files. Both of these libraries are licensed under the GNU General Public License (GNU GPL), meaning that the code for BayesTrees is also open source.

A.3 TreeSum

The TreeSum program includes methods for summarizing the posterior distribution of phylogenies, including calculation of tree probabilities and partition probabiliites. It calculates the RMSET and MeanSD and simulates the RMSET statistic. Finally, it can output the total unique trees sampled, the size of the credible set and the probability of the MAP tree at various points throughout the MCMC analysis.

TreeSum uses the **Postprocess, Data, Nexus, Tree** and **Node** classes from BayesTrees. The input data are the sampled phylogenies from an MCMC analysis (for example, the *.t or *.trees files). To run TreeSum, type the following at the command line:

```
./TreeSum chain1file chain2file chain3file ...
```

If you start TreeSum without the filenames, the program will prompt for the number of chains and the location of the files.

A.4 MAPminer

The MAPminer program implements the frequent subtree mining method detailed in Chapter 5. It uses many of the same classes that are part of the BayesTrees package, including **Postprocess, Data, Nexus, Tree** and **Node**. It implements a separate **Pruning** class and a separate *main* method.

The input data is either a raw output file of trees from an MCMC analysis (for example, the *.t file from MrBayes or BayesTrees) or a posterior distribution of phylogenies (the .trprobs file from MrBayes or BayesTrees). It requires the file to be in Nexus format and the trees in Newick format. Trees can be rooted or unrooted.

Again, the program is non-interactive and retrieves input parameters from a file. A Perl script is included to create this file. The input parameters are as follows:

To run MAPminer, type the following at the command line:

```
./MAPminer datafile parameterfile
```

If you omit either of the input file names, MAPminer will prompt for the location of the files.

The method outputs the best trees found for each number of pruned taxa and the posterior probability of each of the trees.

Bibliography

Drummond, A. and K. Strimmer. 2001. PAL: An object-oriented programming library for molecular evolution and phylogenetics. Bioinformatics 17:662–663.

Galassi, M. 2006. GNU scientific library reference manual (2nd ed.). http://www.gnu.org/software/gsl/.

Gamma, E., R. Helm, R. Johnson, and J. Vlissides. 1995. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.

Lewis, P. O. 2003. NCL: a C++ class library for interpreting data files in NEXUS format. Bioinformatics 19:2330–2331.

Maddison, D. R., D. L. Swofford, and W. P. Maddison. 1997. NEXUS: an extensible file format for systematic information. Syst. Biol. 46:590–621.