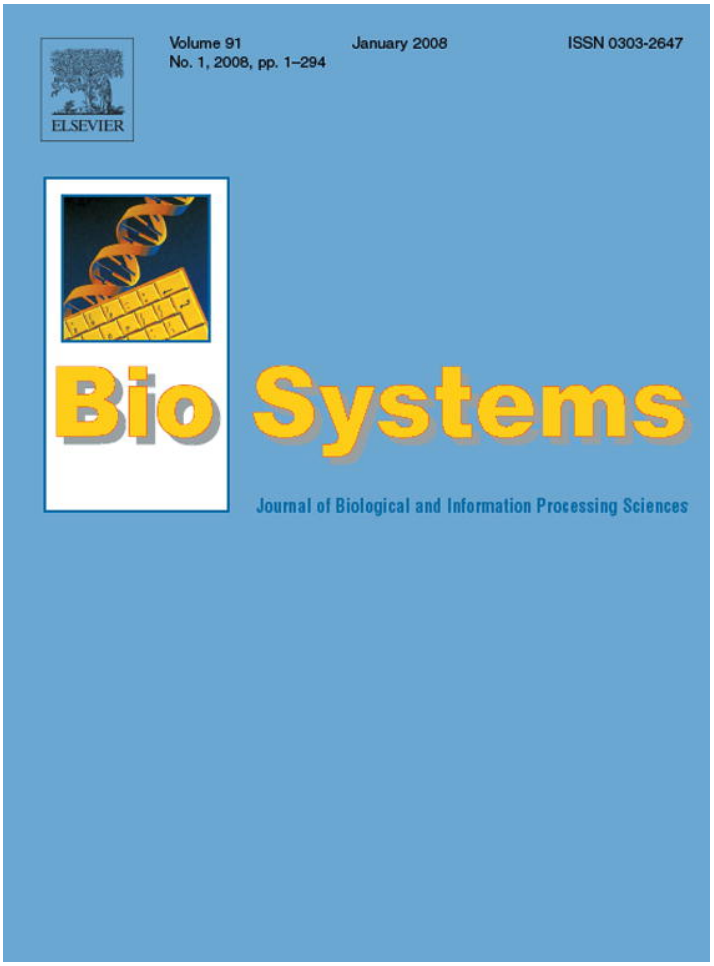

Available online at www.sciencedirect.com
ScienceDirect
BioSystems 91 (2008) 94–107
BioSystems
www.elsevier.com/locate/biosystems

Phylogenetic tree construction using sequential stochastic approximation Monte Carlo

Sooyoung Cheon [a], Faming Liang [b,*]

[a] *Department of Public Health Sciences, University of Virginia, Charlottesville, VA 22908-0717, USA*
[b] *Department of Statistics, Texas A&M University, College Station, TX 77843-3143, USA*

Received 28 January 2007; received in revised form 9 June 2007; accepted 14 August 2007

Abstract

Monte Carlo methods have received much attention recently in the literature of phylogenetic tree construction. However, they often suffer from two difficulties, the curse of dimensionality and the local-trap problem. The former one is due to that the number of possible phylogenetic trees increases at a super-exponential rate as the number of taxa increases. The latter one is due to that the phylogenetic tree has often a rugged energy landscape. In this paper, we propose a new phylogenetic tree construction method, which attempts to alleviate these two difficulties simultaneously by making use of the sequential structure of phylogenetic trees in conjunction with stochastic approximation Monte Carlo (SAMC) simulations. The use of the sequential structure of the problem provides substantial help to reduce the curse of dimensionality in simulations, and SAMC effectively prevents the system from getting trapped in local energy minima. The new method is compared with a variety of existing Bayesian and non-Bayesian methods on simulated and real datasets. Numerical results are in favor of the new method in terms of quality of the resulting phylogenetic trees.

Keywords: Bayesian phylogeny; Curse of dimensionality; Local trap; Markov chain Monte Carlo; Stochastic approximation Monte Carlo

1. Introduction

Phylogenetic trees have been used in biology for a long time to graphically represent evolutionary relationship among species and genes. During the past several decades, a variety of methods have been developed to construct phylogenetic trees, including neighbor joining (Saitou and Nei, 1987), minimum evolution (Kidd and Sgaramella-Zonta, 1971; Rzhetsky and Nei, 1992), maximum parsimony (Fitch, 1971; Maddison, 1991), maximum likelihood (Felsenstein, 1981, 1993; Kishino et al., 1990; Salter and Pearl, 2001), Bayesian methods (Rannala and Yang, 1996; Yang and Rannala, 1997; Mau and Newton, 1997; Mau et al., 1999; Larget and Simon, 1999; Newton et al., 1999; Li et al., 2000), among others. Holder and Lewis (2003) provide an excellent review for these methods. Neighbor joining is a heuristic, distance-based method, whose principle is to construct a tree by successive pairing of the taxa with the smallest distance (nearest neighbors). Since some information is lost in compressing sequences into distances, it may be hard for the neighbor joining method to produce reliable phylogenetic trees for divergent sequences. However, it usually performs well when the divergence between sequences is low. The minimum evolution method is also distance-based. It seeks the tree with the smallest sum of branch lengths, which can be estimated under the least-squares criterion from the distance matrix. Like

* Corresponding author. Tel.: +1 979 8458885; fax: +1 979 8453144.
E-mail address: fliang@stat.tamu.edu (F. Liang).

0303-2647/$ – see front matter © 2007 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.biosystems.2007.08.003

minimum evolution, maximum parsimony and maximum likelihood are only criteria for evaluating given phylogenetic trees. In practice, they are always combined with a search algorithm looking for the best tree. The best tree under maximum parsimony is the tree that requires the minimum number of mutations that could possibly produce the data, and the best tree under maximum likelihood is the tree that is the most likely to have occurred given the observed data and the assumed model of evolution.

Bayesian methods seek to draw inferences based on the posterior distribution of the phylogenetic tree. The posterior distribution can be simulated using Markov chain Monte Carlo (MCMC) algorithms (Liu, 2001). The best tree usually refers to the maximum *a posteriori* (MAP) tree or the consensus tree (Larget and Simon, 1999). Bayesian methods have several advantages over other methods. First of all, they account for the uncertainty embedded in the construction of phylogenetic trees automatically. Secondly, they can be used to infer the models of sequence mutation. Thirdly, they make analysis of large data sets more tractable.

Due to the complexity of genealogy, the above phylogenetic tree construction methods suffer from two difficulties. The first one is the curse of dimensionality. As the number of taxa increases, the number of all possible trees will increase at a rate $((2n-3)!/[2^{n-2}(n-1)!] \approx [(2n-3)/e]^{n-1})$, which is much faster than the exponential rate. As a consequence, the searching time for the optimal tree will increase drastically as the number of taxa increases. The second one is the local-trap problem; that is, the method tends to find a local optimal tree near the starting point. This problem typically appears in applications of the maximum parsimony, maximum likelihood, and Bayesian methods. A number of authors have noticed this difficulty and have tried to apply some advanced MCMC algorithms to resolve it. For example, Huelsenbeck and Ronquist (2001) and Altekar et al. (2004) applied parallel tempering (Geyer, 1991) to the problem, and Feng et al. (2003) employed the multiple-try Metropolis algorithm (Liu et al., 2000). However, no authors have addressed the two difficulties simultaneously.

In this paper, we propose a new method for phylogenetic tree construction. The new method attempts to overcome the two difficulties simultaneously by making use of the sequential structure of phylogenetic trees (as described in Section 3) in conjunction with stochastic approximation Monte Carlo (SAMC) simulations (Liang et al., 2007). As demonstrated by Liang (2003), an appropriate use of the sequential structure of a system can provide substantial help to reduce the curse of dimensionality in simulations from the system. As shown by Liang et al. (2007), SAMC can effectively prevent the system from getting trapped in local energy minima. The new method is compared with a variety of existing Bayesian and non-Bayesian methods on simulated and real datasets. Numerical results are in favor of the new method in terms of quality of the resulting phylogenetic trees.

The remaining part of this paper is organized as follows. In Section 2, we provide a brief description for Bayesian phylogeny analysis. In Section 3, we describe the new phylogenetic tree construction method. In Section 4, we apply the new method to simulated and real datasets. In Section 5, we conclude the paper with a brief discussion.

2. Bayesian Phylogeny Analysis

A phylogenetic tree can be represented as a rooted binary tree. Each node with descendants represents the most recent common ancestor of the descendants, and the root represents the most common ancestor of all the entities at the leaves of the tree. In general, a phylogenetic tree of n leaves has $n-2$ internal nodes (excluding the root node) and $2n-2$ branches. The length of branch represents the distance between two end node sequences and is often calculated from a model of substitution of residues over the course of evolution.

Suppose that the problem under consideration is to construct a phylogenetic tree for n nucleotide sequences (taxa). The problem for protein sequences is similar. The nucleotide sequences can be arranged as a $n \times N$ matrix, where N is the common number of sites or the common length of the sequences. The data can be viewed as a realization of a stochastic process that has evolved along the branches of an unknown phylogenetic tree. By assuming that evolution among sites is independent conditional on the given genealogy, modeling is reduced to a single site. There are several evolutionary models for nucleotides: one parameter model (Jukes and Cantor, 1969) in which nucleotide substitutions have equal probabilities; two-parameter model (Kimura, 1980) which allows for different rates of transitional and transversional events; Felsenstein model (1981) which adds three parameters to the Jukes–Cantor model by allowing the stationary probabilities to be different; HKY85 model (Hasegawa et al., 1985) which possesses a general stationary distribution of the nucleotides and different rates for transition and transversion events. In this paper, we consider the HKY85 model, for which the elements of the transition probability matrix are

given by

$$Q_{j|i}(t) = \begin{cases} \pi_j + \pi_j\left(\frac{1}{\lambda_j} - 1\right)\mathrm{e}^{-\alpha t} + \left(\frac{\lambda_j - \pi_j}{\lambda_j}\right)\mathrm{e}^{-\alpha\gamma_j t} & \text{if } i = j, \\ \pi_j + \pi_j\left(\frac{1}{\lambda_j} - 1\right)\mathrm{e}^{-\alpha t} - \left(\frac{\pi_j}{\lambda_j}\right)\mathrm{e}^{-\alpha\gamma_j t} & \text{if } i \neq j \text{ (transitional event)}, \\ \pi_j(1 - \mathrm{e}^{-\alpha t}) & \text{if } i \neq j \text{ (transversional event)}, \end{cases} \quad (1)$$

where t is the evolution time or the branch length of a phylogenetic tree, α the evolutionary rate, $\lambda_j = \pi_A + \pi_G$ if base j is a purine (A or G) and $\pi_C + \pi_T$ if base j is a pyrimidine (C or T), $\gamma_j = 1 + (\kappa - 1)\lambda_j$, and κ is a parameter responsible for distinguishing between transitions and transversions. The stationary probabilities of the model are π_A, π_C, π_G, and π_T for the nucleotides A, C, G, and T, respectively. The HKY85 model contains five free parameters, namely, α, κ, π_A, π_C, and π_G, which satisfy the constraints $\alpha > 0$, $\kappa > 0$, $0 < \pi_A, \pi_C, \pi_G < 1$, and $0 < \pi_A + \pi_C + \pi_G < 1$.

Let $\omega = (\tau, h, \phi)$ denote a phylogenetic tree, where τ denotes the tree topology, h denotes the branch length vector, and ϕ denotes the vector of parameters of the evolutionary model. The likelihood of the tree can be calculated using the pruning method proposed by Felsenstein (1981). The pruning method produces a collection of partial likelihoods of subtrees, starting from the leaves and working recursively to the root for each site. Let $\mathcal{S} = \{A, C, G, T\}$ denote the set of nucleotides. For site k of a leaf e, define $L_e^k(i) = 1$ if state i matches the base found in the sequence and 0 otherwise, where i indexes the elements of $\mathcal{S}$. At site k of an internal node v, the conditional probability of descendant data given state i is

$$L_v^k(i) = \left(\sum_{j\in\mathcal{S}} L_u^k(j) Q_{j|i}(h_{vu})\right) \times \left(\sum_{j\in\mathcal{S}} L_w^k(j) Q_{j|i}(h_{vw})\right), \quad i \in \mathcal{S},$$

where u and w denote the two children nodes of v, and h_{ab} denotes the length of the branch ended with the nodes a and b. The likelihood function of the tree can then be written as

$$L(\omega|D) = \prod_{k=1}^{N}\sum_{i\in\mathcal{S}} \pi_0(i) L_\rho^k(i), \quad (2)$$

where D denotes the observed sequences of n taxa, ρ denotes the root node, and π_0 is the initial probability distribution assigned to the ancestral root sequence. In all simulations of this paper, π_0 is set to the observed frequency of the nucleotides of the given sequences.

Let $f(\omega)$ denote the prior distribution of ω. The posterior distribution of the phylogenetic tree can then be formed as

$$f(\omega|D) \propto L(\omega|D) f(\omega). \quad (3)$$

Various samplers can then be employed to sample from this posterior. For example, BAMBE (Larget and Simon, 1999) employs the Metropolis–Hastings (MH) algorithm (Metropolis et al., 1953; Hastings, 1970), and MrBayes (Huelsenbeck and Ronquist, 2001) employs the parallel tempering algorithm. Inferences can then be drawn based on the samples simulated from the posterior distribution.

In this paper, we follow Mau et al. (1999) and Larget and Simon (1999) to place a uniform prior on ω. This can be understood that we place a uniform prior on both the parameter space of the HKY model and the joint space of tree topology and branch lengths induced by the HKY model; that is, we set $f(\phi) \propto 1$ and $f(\tau, h|\phi) \propto 1$. Under this setting, the MAP tree coincides with the maximum likelihood (ML) tree. Since the main purpose of this paper is to introduce a new phylogenetic tree construction method and the Bayesian inference for phylogenetic trees has been well explored by many other authors, e.g., Huelsenbeck and Ronquist (2001) and Larget and Simon (1999), our numerical reports will focus on the MAP trees in this paper.

3. Sequential Stochastic Approximation Monte Carlo

In this section, we describe a new phylogenetic tree construction method, which makes use of the sequential structure of phylogenetic trees in conjunction with the stochastic approximation Monte Carlo algorithm (Liang et al., 2007). In the following, the new method will be abbreviated as SSAMC. SSAMC is not only useful for phylogeny analysis, but also useful for many other problems with sequential structures, such as multiloci genetic linkage analysis (Irwin et al., 1994), traveling salesman problem (Wong and Liang, 1997), and image analysis (Besag and Green, 1995).

3.1. Buildup Ladder Construction

SSAMC consists of two steps, the buildup ladder construction and the SSAMC simulation. A buildup ladder (Wong and Liang, 1997; Liang, 2003) comprises a sequence of systems of different dimensions. Typically, we have

$$\dim(\mathcal{X}_1) < \dim(\mathcal{X}_2) < \cdots < \dim(\mathcal{X}_m),$$

where $\mathcal{X}_i$ denotes the sample space of the ith system. The principle of buildup ladder construction is to approximate the original system by a system with a reduced dimension; the reduced system is again approximated by a system with a further reduced dimension, until one reaches a system of a manageable dimension. That is, the corresponding system is able to be sampled easily by a local updating algorithm, such as the MH algorithm and the Gibbs sampler (Geman and Geman, 1984). The solution of the reduced system is then extrapolated level by level until the target system is reached.

For phylogeny problems, the buildup ladder can be constructed as follows. Intuitively, we want to first approximate the shape of a phylogenetic tree using a small number of taxa, and then add other taxa to the tree locally. To achieve this goal, the taxa are first ordered in the following way. Let A and A^c denote the sets of ordered and not yet ordered taxa, respectively. Starting A with an arbitrary taxon; the next taxon added to A should be the one with the maximum distance from the starting taxon; the third taxon should be the one with the maximum distance from the set A, and so on. Here the distance between a taxon and a set of taxa is defined as the minimum distance of the taxon to any taxon in the set, i.e., $\min_{j\in A} d_{ij}$ with d_{ij} denoting the distance between taxon i and taxon j. The ordering procedure can be summarized as follows.

(a) Calculate the pairwise distance matrix (d_{ij}) of the taxa. For example, the distance can simply be the number of different nucleotides or the alignment score calculated according to the substitution scoring matrices, e.g., PAM matrices (Dayhoff et al., 1978) or BLOSUM matrices (Henikoff and Henikoff, 1992) for protein sequences, and the nucleotide substitution matrix (Chiaromonte et al., 2002) for DNA sequences.

(b) Order the taxa sequentially. The next taxon added to A is the taxon $k\ (\in A^c)$ which satisfies the condition: there exist a taxon $m \in A$ such that

$$d_{km} = \max_{i\in A^c} \min_{j\in A} d_{ij}. \tag{4}$$

If there are several taxa all satisfying the above condition, choose one randomly.

In this paper, the distance matrix of the taxa is calculated according to the pairwise alignment scores. It is apparent that the buildup ladder depends on the choice of the starting taxon and the taxon addition rule used in step (b). The effect of different buildup ladders on SSAMC simulations will be discussed at the end of Section 3.2.

3.2. Sequential SAMC Simulation

Suppose a buildup ladder has been constructed for a set of taxa. Let $D_1, \ldots, D_L$ denote L subsets of taxa, $D_1 \subset D_2 \subset \cdots \subset D_L$, where D_i contains the first $|D_i|$ taxa in the buildup order and $D_L = D$ contains all taxa of the dataset. SSAMC is then employed to sample simultaneously from the following distributions,

$$f(\omega_i|D_i) = \frac{1}{Z_i} L(D_i|\omega_i), \quad i = 1, \ldots, L, \tag{5}$$

where ω_i is the tree constructed for the taxa contained in D_i, $f(\omega_i|D_i)$ is the posterior distribution of ω_i, and Z_i is the unknown normalizing constant of $f(\omega_i|D_i)$. The uniform prior is also placed on the partial tree at each level of the buildup ladder. The reason why we need to simultaneously generate samples from a series of partial trees will be discussed at the end of this section. The SSAMC simulation consists of two stages, the normalizing constant ratio estimation and the target sample generation, which are described in order as follows.

To use the stochastic approximation Monte Carlo (SAMC) algorithm to estimate the ratios of the normalizing constants $Z_1, \ldots, Z_L$, we need to define two types of Monte Carlo moves, namely, the moves for updating a partial tree at a given buildup level and the moves for jumping between neighboring levels. In this paper, the moves used in Larget and Simon (1999) are adopted for updating partial trees at a given buildup level. Those moves include three parts: the part for updating model parameters, the part for updating branch lengths, and the part for rearranging tree topologies. Larget and Simon (1999) describe the moves for both types of trees with and without molecular clocks. Our tree corresponds to the case without molecular clocks. Refer to Larget and Simon (1999) for the details.

For jumping between neighboring levels, we design two operators, extrapolation and projection, which are depicted by Fig. 1. Let t denote the current iteration, let

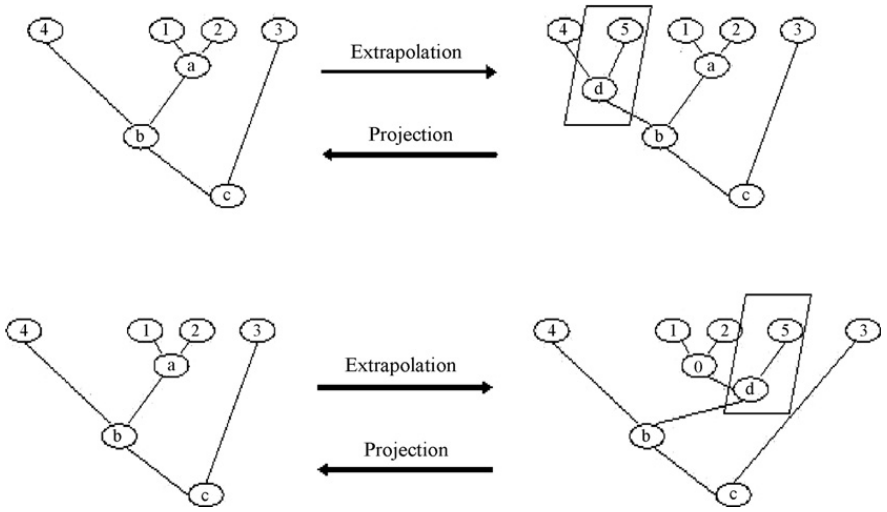

Fig. 1. Illustrative graphs for extrapolation and projection operators when a taxon is sampled from leaves (upper panel) and internal nodes (lower panel). Extrapolation (left → right): insert taxon '5' and internal node 'd' to the current tree. Projection (right → left): delete taxon '5' and the corresponding internal node '*d*' from the current tree.

$i^{(t)}$ denote the current buildup level, let $\omega_i^{(t)}$ denote the current tree, and let ω^* denote the proposed tree. The projection operator is simple. It just removes the leaves belonging to the set $D_{i^{(t)}} \setminus D_j$ (with $j = i^{(t)} - 1$) and the corresponding internal nodes. The corresponding transition probability is $T(\omega_i^{(t)} \to \omega^*) = 1$. The extrapolation operator is a little bit complex. Let D^* denote the set of all leaves and internal nodes of the partial tree $\omega_i^{(t)}$. The operator proceeds as follows.

Initialize $T(\omega_i^{(t)} \to \omega^*) = 1$, and do the following steps for each taxon $k \in D_j \setminus D_{i^{(t)}}$ (with $j = i^{(t)} + 1$):

(a) Sample a node l with probability $p_{kl} = \mathrm{e}^{-d_{kl}/\tau_s} / \sum_{l'=1}^{|D^*|} \mathrm{e}^{-d_{kl'}/\tau_s}$, $l \in D^*$, where τ_s is called the insertion temperature. A high insertion temperature corresponds to a random insertion, whereas a low insertion temperature corresponds to the nearest neighbor insertion. In this paper, we set $\tau_s = 0.6$ in all simulations.
(b) Add the taxon k to the tree as a sibling leaf of l and set $D^* \leftarrow D^* + \{k,$ parent node of k and $l\}$. The position of the parental node of k and l is sampled uniformly on the branch between l and its current parental node.
(c) Calculate the extrapolation probability by setting $T(\omega_i^{(t)} \to \omega^*) \leftarrow T(\omega_i^{(t)} \to \omega^*) \times p_{kl}/b_{lk}$, where b_{lk} denotes the length of the branch between l and its parental node before adding taxa k to the tree.

Let $\theta_i^{(t)}$ denote an estimate of $\log(Z_i)$ obtained at iteration t. The new tree ω^* generated by the extrapolation or projection operators is accepted or rejected with the probability:

$$\min\left\{1, \frac{\mathrm{e}^{\theta_i^{(t)}}}{\mathrm{e}^{\theta_j^{(t)}}} \frac{L(\omega^*|D_j)}{L(\omega_i^{(t)}|D_{i(t)})} \frac{T(\omega^* \to \omega_i^{(t)})}{T(\omega_i^{(t)} \to \omega^*)} \frac{\tilde{T}(D_j \to D_{i(t)})}{\tilde{T}(D_{i(t)} \to D_j)}\right\}, \tag{6}$$

where $\tilde{T}(\cdot \to \cdot)$ denotes the proposal probability of the extrapolation and projection moves. If it is accepted, make the updating $i^{(t+1)} \leftarrow j$ and $\omega_j^{(t+1)} \leftarrow \omega^*$; otherwise, set $i^{(t+1)} \leftarrow i^{(t)}$ and $\omega_i^{(t+1)} \leftarrow \omega_i^{(t)}$.

SSAMC algorithm. Let $\tilde{T}_{mk} = \tilde{T}(D_m \to D_k)$ denote the proposal probability for a transition from level m to level k. In this paper, $\tilde{T}$ is specified as a tridiagonal matrix with the elements $\tilde{T}_{1,1} = \tilde{T}_{L,L} = 2/3$, $\tilde{T}_{1,2} = \tilde{T}_{L,L-1} = 1/3$, and $\tilde{T}_{m,m+1} = \tilde{T}_{m,m-1} = \tilde{T}_{m,m} = 1/3$. Initialize the estimates of $\log(Z_m)$ s by setting $t = 0$ and $\theta_1^{(0)} = \cdots = \theta_L^{(0)} = 0$. Generate an arbitrary tree at the first level of the buildup ladder, and set $i^{(0)} = 1$. One iteration of SSAMC consists of the following steps:

(a) *Level proposing*: Generate level j according to the stochastic matrix $\tilde{T}$ and the current level $i^{(t)}$.
(b) *Tree updating*:
  (b.1) If $j = i^{(t)}$, update the model parameters, branch lengths, and tree topology iteratively as in Larget and Simon (1999).
  (b.2) If $j = i^{(t)} + 1$, propose an extrapolation operation along the buildup ladder, and accept the proposed tree with a probability calculated in (6).
  (b.3) If $j = i^{(t)} - 1$, propose a projection operation along the buildup ladder, and accept the

proposed tree with a probability calculated in (6).

(c) *Estimate updating*: Set $\theta_i^{(t+1)} \to \theta_i^{(t)} + \gamma_t$, where γ_t is called the gain factor and satisfies the conditions:

$$\sum_{t=0}^{\infty} \gamma_t = \infty, \qquad \sum_{t=0}^{\infty} \gamma_t^2 < \infty. \tag{7}$$

Under mild conditions, Liang et al. (2007) showed that $e^{\theta_i^{(t)}}/e^{\theta_j^{(t)}}$ converges to Z_i/Z_j almost surely. A remarkable feature of SSAMC is that it will not get stuck at a buildup level. This is due to the intrinsic self-adjusting mechanism of the algorithm. If a level transition proposal is rejected, the weight of the current level will increase by the factor e^{γ_t}, and thus a level transition proposal will less likely be rejected in the next iteration. This process can repeat until a success of level transition occurs.

Liang et al. (2007) pointed out that SAMC tends to sample from each level equally when t becomes large. This relates directly to the choice of the gain factor and the diagnosis for the convergence of the simulation. In this paper, we set:

$$\gamma_t = \frac{t_0}{\max(t_0, t)}, \quad t = 0, 1, 2, \ldots, \tag{8}$$

with $t_0 > 0$ being a user tuning parameter. It is easy to verify that (8) satisfies the conditions (7). The choice of t_0 is problem dependent. The more complex the problem is, the higher value of t_0 one should use. A practical guideline for the choice of t_0 is to examine flatness of the histogram of the samples drawn at different levels. A histogram is said flat if the sampling frequency at each buildup level is not less than 80% of the average sampling frequency over levels. If the histogram is not flat, SSAMC should be rerun with a large value of t_0 or a large number of iterations. In this paper, we set $t_0 = 10{,}000$ for all simulations. Refer to Liang et al. (2007) for more discussions on the choice of t_0.

Suppose that SSAMC has converged in a run. To generate MCMC samples from the target distribution $f(\omega_L|D_L)$, we can continue to run SSAMC by fixing the gain factor to zero; that is to simulate from (5) in the style of simulated tempering (Marinari and Parisi, 1992; Geyer and Thompson, 1995) with the estimated normalizing constants. The complete tree samples generated in this stage can be used for making a Bayesian inference about the phylogeny. In fact, only the complete tree samples generated in this stage need to be retained for the Bayesian inference. Note that if we are only interested in the MAP tree, this stage can be skipped. We also note that the samples generated in the stage of normalizing constant estimation can not be used for the Bayesian inference due to the inhomogeneous and adaptive nature of the simulation process.

SSAMC is different from the parallel tempering algorithm used in MrBayes (Huelsenbeck and Ronquist, 2001). Parallel tempering works by simulating in parallel a sequence of distributions scaled by different temperatures, while keeping the sample space unchanged at each level. SSAMC works with a buildup ladder with dimension increasing along the ladder. The use of buildup ladder can provide substantial help to reduce the curse of dimensionality for simulations of complex systems. This has been argued theoretically by Liang (2003) based on the Rao–Blackwellization theorem (Liu, 2001, p. 27), which suggests that one should carry out analytical computation as much as possible in order to improve efficiency of the simulations. Note that the distribution defined at a lower buildup level can be regarded as a marginal distribution of the distribution defined at a higher buildup level. Hence, we prefer to sample simultaneously from a series of partial trees instead of complete trees. As shown by our numerical results, the improvement made by SSAMC over MrBayes and BAMBE can be significant in both simulation times and quality of the solutions. The latter two methods work on the space of complete trees directly.

SSAMC is also different from the stepwise addition methods, e.g., the quartet-puzzling algorithm (Strimmer and von Haeseler, 1996; Strimmer et al., 1997). The quartet-puzzling algorithm inserts taxa into an already constructed partial tree sequentially and deterministicly, while SSAMC does this stochastically by moving back and forth along a buildup ladder. At the low levels of the buildup ladder, where only a small number of taxa are involved, the simulation can easily reach its equilibrium, and this can help the system to escape from local traps effectively. Intuitively, when the system is trapped in a local energy minimum at a high level, instead of struggling for numerous iterations at that level, the system can move to a lower buildup level to update its configuration there and then moves back. Hence, SSAMC is less likely to get trapped in local energy minima than is the quartet-puzzling algorithm. A similar idea has been implemented in simulated tempering (Marinari and Parisi, 1992) in which the system tries to escape from local energy minima by moving back and forth along a temperature ladder. Both temperature and buildup ladders lead to a series of systems with increasing complexity. Simulated annealing (Kirkpatrick et al., 1983) also bears a similar idea to escape from local energy minima, but in which the moves are unidirectional, from higher to lower tem-

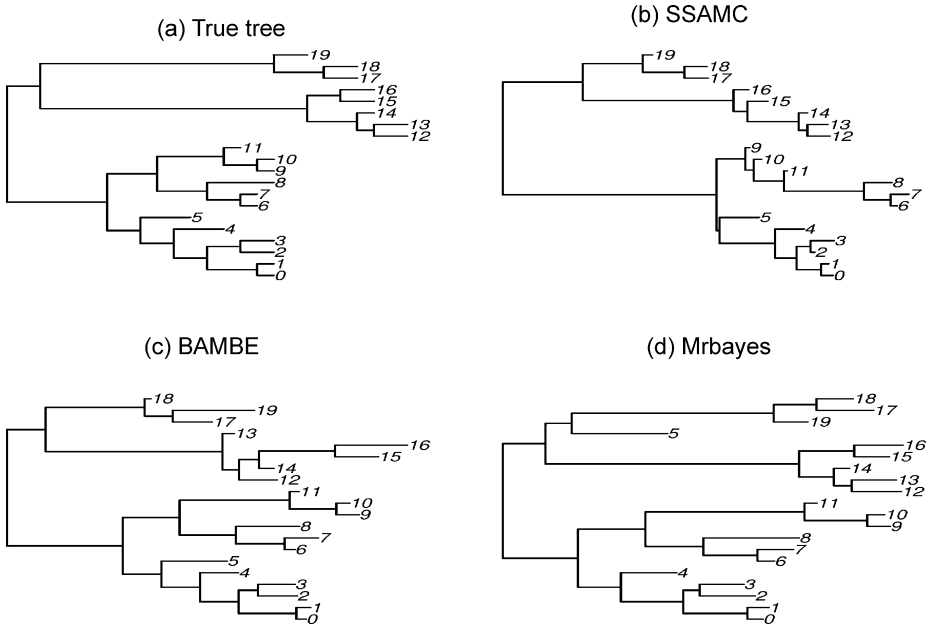

Fig. 2. Comparison of the phylogenetic trees produced by SSAMC, BAMBE, and MrBayes for the simulated example. The respective log-likelihood values of the trees are (a) −4209.44, (b) −4196.09, (c) −4197.68, and (d) −4198.19.

peratures. In addition, the self-adjusting mechanism of SSAMC can provide substantial helps for the system to transit smoothly between neighboring levels in the first stage of the simulation.

In the second stage, SSAMC seeks to sample from the following mixture distribution:

$$\frac{1}{L}\sum_{m=1}^{L} f(\omega_m|D_m).$$

The samples generated at the highest buildup level correspond to complete trees, and are approximately distributed as $f(\omega_L|D_L)$ when the number of iterations becomes large. Other distributions $f(\omega_1|D_1), \ldots, f(\omega_{L-1}|D_{L-1})$ only work as trial distributions for sampling from complete trees. Hence, the choice of the buildup ladder will not affect the equilibrium of the simulation at the highest buildup level, although it may affect the efficiency of the simulation. As mentioned before, the buildup ladder construction depends on the choice of the starting taxon and the taxon addition rule. Our experience shows that given the taxon addition rule (4), the effect of starting taxon on SSAMC simulations is really minor. Hence, we suggest to choose the starting taxon randomly. However, the effect of the taxon addition rule can be significant. An arbitrary buildup ladder may cause difficulties for state transitions between some pairs of neighboring levels. Hence, the taxon addition rule should be carefully selected to ensure that the systems on neighboring levels resemble each other largely and thus the state transitions between them can be made smoothly.

4. Numerical Examples

4.1. A Simulated Example

In this study, we test the ability of SSAMC to find high quality phylogenetic trees. A total of 20 nucleotide sequences were generated according to a given tree (shown in Fig. 2(a)), a given root sequence (shown in Table 1), and the HKY85 model with parameters $\kappa = 2$, $\alpha = 1$, and $\pi_A = \pi_C = \pi_G = \pi_T = 0.25$. The length of each sequence is 200.

SSAMC was first applied to this dataset. Suppose that the taxa have been ordered according to the pairwise alignment scores, and a buildup ladder has been con-

Table 1
The root sequence for the simulated example

ATGAACCCTT	ACATCCTAAT	AACCCTTCTT	TTCGGACTAG	GTCTAGGAAC
TACAATTACA	TTTGCAAGCT	CCCACTGACT	CCTTGCTTGA	ATAGGCCTTG
AACTAAACAC	CCTCGCTATT	ATCCCACTGA	TAGCCCAACT	CCACCACCCC
CGGGCAGTCG	AAGCTACCAC	AAAATACTTC	CTCACCCAAG	CTGCTGCCGC

Table 2
Relative sampling frequencies of the buildup levels for the simulated example

Level	Frequency (%)	Level	Frequency (%)	Level	Frequency (%)	Level	Frequency (%)
1	100.8321	2	100.6020	3	100.4161	4	100.2512
5	99.8520	6	99.4255	7	98.9422	8	98.7080
9	100.7606	10	100.1461	11	100.4045	12	99.6597

structed as follows: $D_1 = \{1^*, \ldots, 4^*\}$, $D_2 = \{1^*, \ldots, 5^*\}$,. . ., $D_7 = \{1^*, \ldots, 10^*\}$, $D_8 = \{1^*, \ldots, 12^*\}$,. . ., and $D_{12} = \{1^*, \ldots, 20^*\}$, where $1^*, \ldots, 20^*$ denote the ordered taxa. Note that at the latter level of the buildup ladder, the taxa tend to be inserted into different branches of the partial tree, and thus more taxa can be added at one level. For a large dataset, this can make the number of buildup levels be substantially smaller than the number of taxa. SSAMC was run 5 times independently, and each run consists of 2×10^6 iterations. Table 2 shows the relative sampling frequencies obtained in one run, where the relative sampling frequency of level i is defined as $N_i/(N/L) \times 100\%$, and N_i and N denote the sampling frequency of level i and the total number of iterations of the run, respectively. The approximate equality of sampling frequencies at each buildup level indicates that the choices of t_0 and the total number of iterations are appropriate for this example. Fig. 2(b) shows the MAP tree sampled among the five runs.

For comparison, two Bayesian phylogeny software, MrBayes and BAMBE, were also applied to this example. Each one was run 5 times independently with its default parameter setting, and each run consisted of 2×10^6iterations. Fig. 2(c) and (d) shows the MAP trees found by BAMBE and MrBayes among the five runs, respectively. The comparison indicates that the tree constructed by SSAMC most closely resembles to the true tree for this example. Fig. 3 shows the progression curves of the best log-likelihood values produced by the above three methods. It indicates that BAMBE tends to get trapped in a local energy minimum very quickly, and MrBayes and SSAMC perform better in this respect. However, during these runs MrBayes fails to produce better trees than those produced by SSAMC.

For a thorough comparison, non-Bayesian methods, neighbor joining, minimum evolution, maximum parsimony, and maximum likelihood, were also tried for this example. These methods have been implemented in many software, e.g., PHYLIP (http://evolution.genetics.washington.edu/phylip.html), MEGA (http://www.megasoftware.net), PAUP (http://paup.csit.fsu.edu/index.html), and APE (http://cran.r-project.org). The best trees produced by them are shown in Fig. 4. It is easy to see that these trees are inferior to the tree produced by SSAMC in terms of similarity to the true tree.

The Bayesian analysis has been done for the parameters of the HKY85 model for this example. SSAMC was run again for this dataset with the same parameter settings as those used above except that the gain factor was set to zero during the last 10^6iterations, and the samples generated during the last 10^6 iterations were collected for the Bayesian analysis. The run was repeated five times. The numerical results are summarized in Table 3. For comparison, the samples generated by MrBayes and BAMBE during the last 10^6 iterations of the above runs were also collected for the Bayesian analysis. Since each site of the taxon sequences is modeled equally in this example, the parameter α is restricted to be 1. This is the same for all the three methods under comparison. Table 3 shows that SSAMC produces the highest averaged log-likelihood value and the most accurate estimate

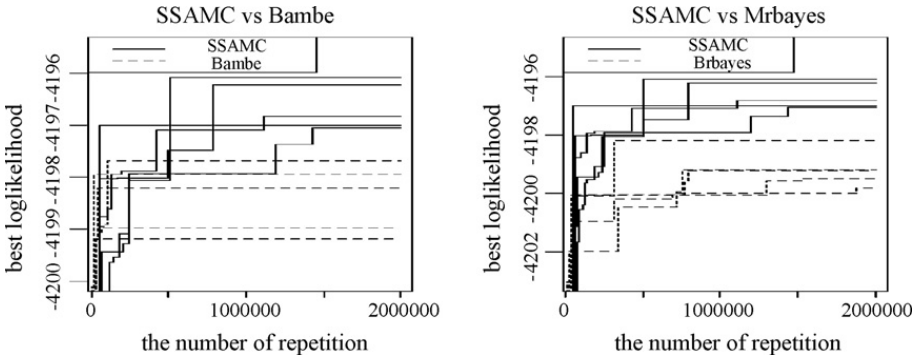

Fig. 3. Comparison of the progression curves of the best log-likelihood values produced by SSAMC, MrBayes, and BAMBE. Left: SAMC vs. BAMBE. Right: SSAMC vs. MrBayes.

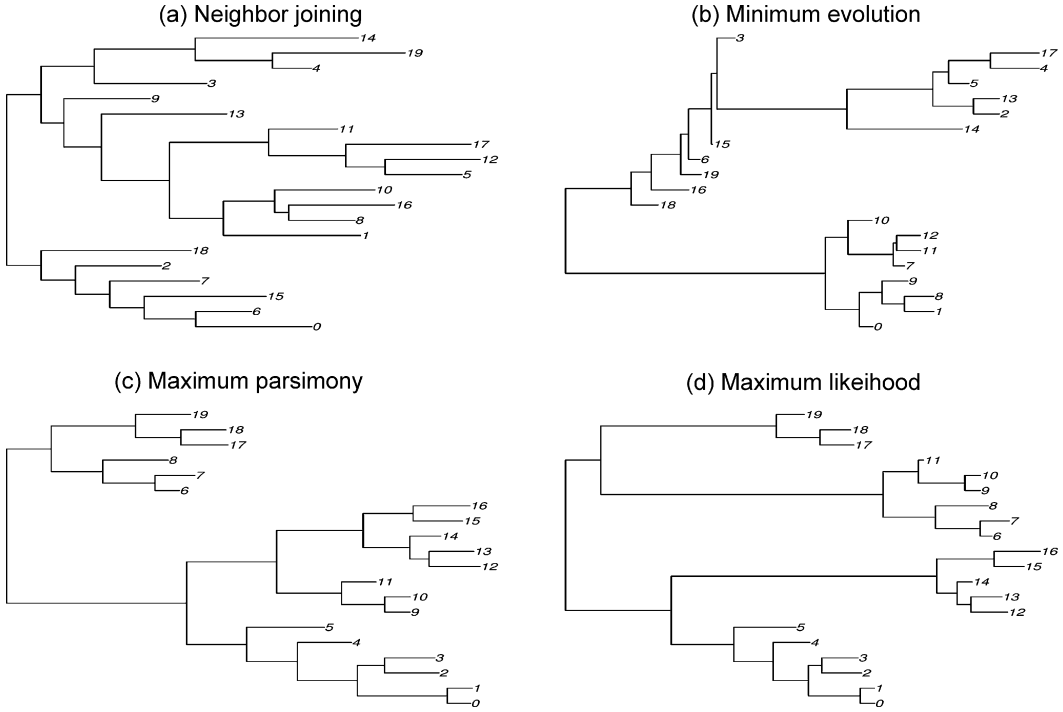

Fig. 4. Best trees produced by (a) neighbor joining, (b) minimum evolution, (c) maximum parsimony, and (d) maximum likelihood (with the log-likelihood value −4197.61) for the simulated example. Tree (d) was produced using the software APE with the HKY85 model, while others were produced using the software PHYLIP with default parameter settings.

for κ among the three methods. Note that the improvement made by SSAMC is significant in terms of averaged log-likelihood values and estimates of κ, although the absolute differences are small. Finally, we point out that the estimates of the nucleotide frequencies produced by the three methods are similar, as they are mainly determined by the data.

Later, MrBayes and BAMBE were re-run independently 10 times, and each run consists of 2×10^7 iterations, a 10-fold increase of the total number of iterations performed in previous runs. However, neither MrBayes nor BAMBE can produce a better log-likelihood value than that produced by SSAMC in the previous runs.

A CPU cost analysis has been done for SSAMC, BAMBE and MrBayes. BAMBE has the simplest program structure among the three algorithms. It is a single Markov chain algorithm, and the state (or tree) undergoes a single MH update per iteration. MrBayes is a population MCMC algorithm in which five Markov chains are run in parallel. At each iteration, the state of each chain is updated once besides the state swapping operations between chains. Hence, the CPU time cost by an iteration of MrBayes is five times more than that cost by an iteration of BAMBE. Like BAMBE, SSAMC is a single chain algorithm, and its state only undergoes a single update at each iteration. Since in most iterations it works on partial trees, on average SSAMC costs less CPU time per iteration than BAMBE does. This can be seen in Table 3. Recall that both BAMBE and MrBayes work on the space of complete trees. To figure out the impact of the number of taxa on CPU times, the three algorithms were applied to three more datasets with 10, 30 and 40 taxa, which are generated using the same procedure as described above. Each algorithm was run once for each dataset, and each run consisted of 2×10^6 iterations.

Table 3
Bayesian analysis for the parameters of the HKY85 model used for the simulated example

Methods	CPU (m)	Averaged log-likelihood	κ	π_A	π_G	π_C	π_T
SSAMC	19.2	−4202.06 (0.11)	1.85 (0.03)	0.254 (5.5e−5)	0.225 (4.8e−5)	0.262 (3.9e−5)	0.258 (4.1e−5)
BAMBE	25.6	−4208.45 (0.09)	1.77 (1.9e−3)	0.253 (3.8e−5)	0.225 (2.8e−5)	0.263 (3.7e−5)	0.259 (2.6e−5)
MrBayes	138.9	−4202.64 (0.05)	1.77 (3.2e−4)	0.255 (4.6e−5)	0.227 (3.8e−5)	0.261 (6.2e−5)	0.257 (8.5e−5)

CPU: CPU time (in minutes) cost by a single run of the algorithm on an Intel Pentium III computer. Each of the other entries of the table is calculated by averaging over five independent runs, and the number in the parentheses represent the standard deviation of the corresponding average.

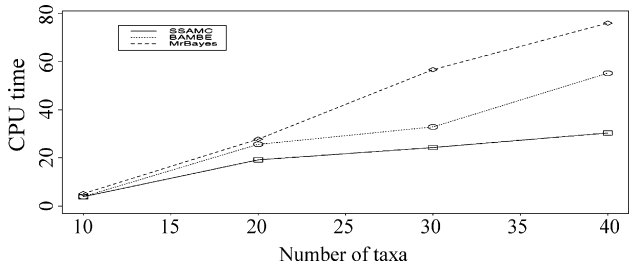

Fig. 5. CPU times cost by a single run (2×10^6iterations) of SSAMC, BAMBE and MrBayes.The CPU time of MrBayes has been rescaled by dividing by 5.

Fig. 5 shows the CPU times used by each algorithm. For convenience, the CPU times used by MrBayes have been rescaled by dividing by 5 in the plot. Among the three algorithms, SSAMC has the lowest marginal CPU cost with respect to the number of taxa. As the number of taxa increases, the CPU time used by each algorithm is more and more dominated by the part used for likelihood evaluation, so the CPU saving created by SSAMC is more and more significant. Note that even with shorter CPU times, the solutions found by SSAMC are still better than those found by the other two algorithms. This can be seen from Table 4, which summarizes the computational results of the three algorithms for both the datasets with 30 and 40 taxa. For each dataset each algorithm was run five times, and each run consisted of 2×10^6iterations.

We have also examined the impact of the sequence length on CPU times for the three algorithms. Since the sequence length has only a linear effect on the CPU time required for likelihood evaluation for each algorithm, the sequence length will not change the marginal CPU cost established above with respect to the number of taxa. For example, our experiments show that for the case of 20 taxa, the ratios of the CPU times per iteration used by SSAMC and BAMBE are always around 0.8 regardless of the sequence length varies from 150 to 250.

4.2. Cichlid Fishes

In this subsection, we analyzed aligned protein-coding mitochondrial DNA sequences obtained from 32 species of cichlid fishes (Kocher et al., 1995). Table A.1(in Appendix A) shows the tribal classification of 32 species of African cichlid fish. Taxa 1–5 form a flock from Lake Malawi. The remainder from Lake Tanganyika (6–31 taxa) constitute a Tanganyikan flock. The Malawi, Ectodini, and Lamprologini tribes are represented by the letters *A*, *C*, and *D*, respectively. Class *B* consists of taxa 6–9, a combination of most of Tropheini and one species of Limnochromini. Classes $E = \{22, 23, 24, 26, 27\}$ and $F = \{28, 29, 30, 31\}$ are convenient conglomerations (pseudoclades) of the remaining tribes. Taxa {25} is not grouped. Taxon 32 is an outgroup from cichlid America. Each DNA sequence consists of 1044 sites. Identical nucleotides are observed on 567 sites, and the nucleotides on the remaining sites are used for phylogenetic tree construction.

SSAMC was first applied to this example. The buildup ladder was constructed as follows: $D_1 = \{1^*, 2^*, 3^*, 4^*\}$, $D_2 = \{1^*, \ldots, 5^*\}$,. . ., $D_7 = \{1^*, \ldots, 10^*\}$, $D_8 = \{1^*, \ldots, 11^*, 12^*\}$,. . ., $D_{18} = \{1^*, \ldots, 31^*, 32^*\}$, where $1^*, \ldots, 32^*$ denotes the ordered taxa. SSAMC was run five times and each run consists of 10^6 iterations. Table 5 shows the relative sampling frequencies obtained in one run. The approximate equality of the relative sampling frequencies at each buildup level implies the appropriateness of the parameter settings for SSAMC. The best log-likelihood value found among the five runs is −7726.51. For comparison, MrBayes and BAMBE were also applied to this example. Each software was run five times, and each run consists of 10^6 iterations. The best log-likelihood values produced by them are −7876.98 and −7888.57, respectively. Fig. 6 compares the MAP trees found by the three methods. As reported below, the best log-likelihood value produced by the maximum likelihood method is −7881.18. These values are all significantly lower than that produced by SSAMC.

For a thorough comparison, the non-Bayesian methods, neighbor joining, minimum evolution, maximum parsimony, and maximum likelihood, were also applied to this example. Fig. 7 shows the trees constructed by

Table 4
Computational results produced by SSAMC, BAMBE and MrBayes for the datasets of 30 and 40 taxa

Methods	Thirty taxa			Forty taxa		
	Best	Average	S.D.	Best	Average	S.D.
SSAMC	−4714.79	−4716.77	0.58	−5854.50	−5856.14	0.42
BAMBE	−4787.08	−4789.52	1.91	−5857.70	−5858.32	0.29
MrBayes	−4718.71	−4719.59	0.42	−5857.74	−5857.74	0.37

Let l_i denote the maximum log-likelihood value produced by an algorithm during the ith run. "Best" denotes the best log-likelihood values produced among five runs, i.e., $\max_{i=1}^{n} l_i$; "Average" denotes the average of l_i's, i.e., $\bar{l} = \sum_{i=1}^{5} l_i/5$; "S.D." denotes the standard deviation of $\bar{l}$.

Table 5
The relative sampling frequency of the buildup levels for African cichlids

Level	Frequency (%)	Level	Frequency (%)	Level	Frequency (%)	Level	Frequency (%)
1	103.1456	2	102.9891	3	102.8426	4	102.8495
5	102.6669	6	102.5925	7	102.0664	8	101.7361
9	100.6926	10	98.6258	11	98.1661	12	97.8469
13	97.8040	14	95.8876	15	97.8977	16	96.1086
17	98.0418	18	98.0403				

them. All trees shown in Figs. 6 and 7 have a fair degree of similarity. Each has clades A, B, C, D, and F in common. The greatest disparity between estimates involves the attachment of taxa from clade E. Interestingly, all methods occur in placing the clade B closer to the clade A.

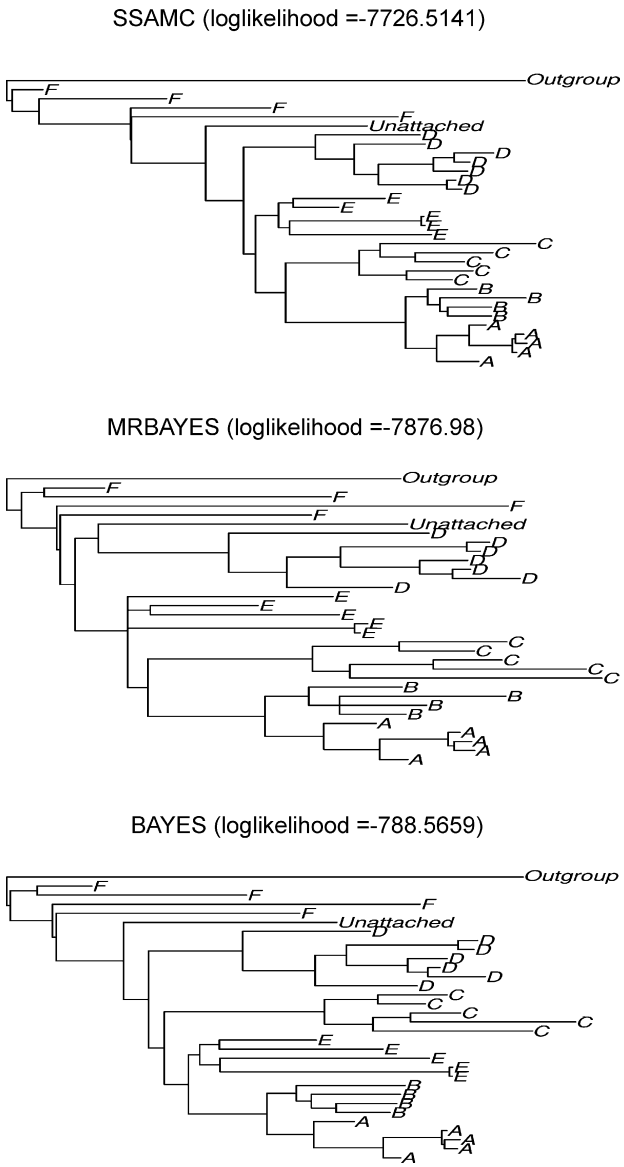

Fig. 6. Comparison of the MAP trees produced by SSAMC, MrBayes, and BAMBE for African cichlid fish example. The respective log-likelihood values are −7726.51, −7876.98, and −7888.57.

5. Discussion

In this paper, we have proposed SSAMC as a new method for phylogenetic tree construction. SSAMC makes use of the sequential structure of phylogenetic trees in conjunction with stochastic approximation Monte Carlo simulations. SSAMC is compared with a variety of existing phylogeny estimation methods on simulated and real datasets. Numerical results are in favor of the new method in terms of quality of the phylogenetic trees.

As described in the text, SSAMC simulations consist of two stages, the normalizing constant ratio estimation stage and the posterior sample generation stage. If we are only interested in the MAP trees, the second stage can be skipped. In this case, the tempering technique (Kirkpatrick et al., 1983; Geyer, 1991; Marinari and Parisi, 1992) can be combined with SSAMC by simulating from a sequence of rescaled distributions defined as follows:

$$f'(\omega_i|D_i) = \frac{1}{Z'_i}[L(D_i|\omega_i)]^{1/\tau_i}, \quad i = 1, \ldots, L, \qquad (9)$$

where $\tau_1 \geq \cdots \geq \tau_L$ denotes a pre-specified temperature ladder, and Z'_i is the normalizing constant of the distribution. A low value of τ_L will force sampling to focus on the high density region of the posterior distribution of the complete tree.

SSAMC is different from the sequential particle filter (SPF) algorithm (e.g., Grassberger, 1997; Chopin, 2002) in several aspects, although they both work on a sequence of systems with different dimensions. Firstly, SPF is an importance sampling-based algorithm, and suffers from progressive degeneracy. As sampling moving on from lower to higher dimensional levels, fewer and fewer samples, also known as particles in the context of sequential importance sampling, retain significant weights. The reweighting or resampling step does not protect from degeneracy: it just saves further calculation time by getting rid of samples with nonsignificant weights; and replaces high weights with numerous replicates of an unique sample, thereby introducing high

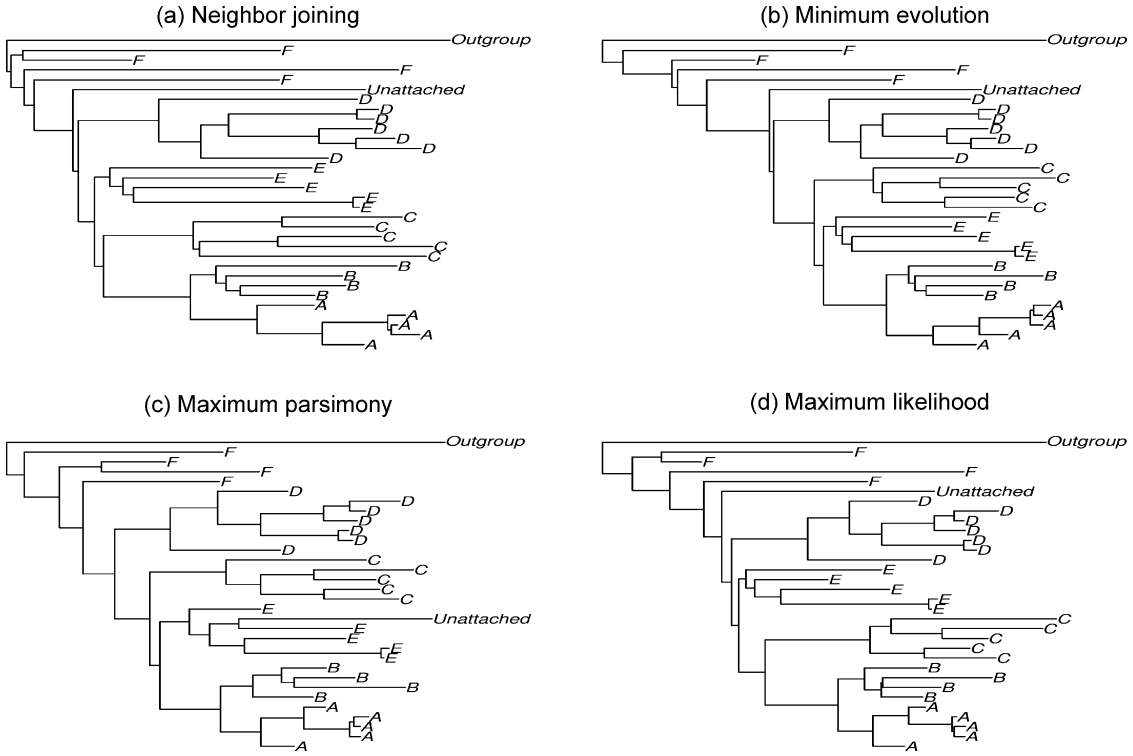

Fig. 7. Best trees produced by (a) neighbor joining, (b) minimum evolution, (c) maximum parsimony, and (d) maximum likelihood (with the log-likelihood value −7881.18) for the Cichlid fishes example. Tree (d) was produced using the software APE with the HKY85 model, while others were produced using the software PHYLIP with default parameter settings.

correlations between samples. This difficulty has been automatically overcome by SSAMC which works in the context of MCMC and moderates the acceptance of new samples with the Metropolis–Hastings rule. Secondly, SSAMC automatically provides a look ahead mechanism in inserting new taxa into partial trees by sampling back and forth along the buildup ladder. Whilst SPF can only move forward along the buildup ladder. A potential advantage of SPF over SSAMC is its flexibility: SPF allows for non-reversible level transition moves. However, it requires the ability to compute the associated importance weights, and this is impossible for almost all non-reversible level transition moves.

Acknowledgments

Liang's research was partially supported by by grants from the National Science Foundation (DMS-0405748) and the National Cancer Institute (R01-CA104620).

Appendix A

See Table A.1 .

Table A.1
The tribal classification of 32 species of African cichlid fish (Kocher et al., 1995)

Label	Species name	Tribe	Clade
1	*Pseudotropheus zebra*	Malawi	A
2	*Buccochromis lepturus*	Malawi	A
3	*Champsochromis spilorhynchus*	Malawi	A
4	*Lethrinops auritus*	Malawi	A
5	*Rhamphochromis* sp.	Malawi	A
6	*Lobochilotes labiatus*	Tropheini	B
7	*Petrochromis orthognathus*	Tropheini	B
8	*Gnathochromis pfefferi*	Limnochromini	B
9	*Tropheus moorii*	Tropheini	B
10	*Callochromis macrops*	Ectodini	C
11	*Cardiopharynx schoutedeni*	Ectodini	C
12	*Opthalmotilapia ventralis*	Ectodini	C
13	*Xenotilapia flavipinnus*	Ectodini	C
14	*Xenotilapia sima*	Ectodini	C
15	*Chalinochromis popeleni*	Lamprologini	D
16	*Julidochromis marlieri*	Lamprologini	D
17	*Telmatochromis temporalis*	Lamprologini	D
18	*Neolamprologus brichardi*	Lamprologini	D

Table A.1 (*Continued*)

Label	Species name	Tribe	Clade
19	*Neolamprologus tetracanthus*	Lamprologini	D
20	*Lamprologus callipterus*	Lamprologini	D
21	*Lepidiolamprologus elongatus*	Lamprologini	D
22	*Perissodus microlepis* 1	Perissodini	E
23	*Perissodus microlepis* 2	Perissodini	E
24	*Cyphotilapia frontosa*	Tropheini	E
25	*Tanganicodus irsacae*	Eretmodini	Unattached
26	*Limnochromis auritus*	Limnochromini	E
27	*Paracyprichromis brieni*	Cyprichromini	E
28	*Oreochromis niloticus*	Tilapiini	F
29	*Tylochromis polylepis*	Tylochromini	F
30	*Boulengerochromis microlepis*	Tilapiini	F
31	*Bathybates* sp.	Bathybatini	F
32	*Cichlasoma citrinellum* 28	Central America	Outgroup

References

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

Besag, J., Green, P.J., 1995. Spatial statistics and Bayesian computation. J. R. Stat. Soc. Ser. B55, 25–37.

Chiaromonte, F., Yap, V., Miller, W., 2002. Scoring pairwise genomic sequence alignments. Pacific Symp. Biocomput. 7, 115–126.

Chopin, N., 2002. A sequential particle filter method for static models. Biometrika 89, 539–551.

Dayhoff, M.O., Schwartz, R.M., Orcutt, B.C., 1978. A model of evolutionary change in proteins. In: Dayhoff, M.O. (Ed.), Atlas of Protein Sequence, Structure, vol. 5, Suppl. 3. National Biomedical Research Foundation, Washington, DC, pp. 345–352.

Felsenstein, J., 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. J. Mol. Evol. 17, 368–376.

Felsenstein, J., 1993. PHYLIP (Phylogenetic Inference Package), Version 3.5. University of Washington, Seattle.

Feng, X., Buell, D.A., Rose, J.R., Waddell, 2003. Parallel algorithms for Bayesian phylogenetic inference. J. Parallel Distrib. Comput. 63, 707–718.

Fitch, W.M., 1971. Toward defining the course of evolution: minimal change for a specific tree topology. Syst. Zool. 20, 406–416.

Geman, S., Geman, D., 1984. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. IEEE Trans. Pattern Anal. Mach. Intell. 6, 721–741.

Geyer, C.J., 1991. Markov Chain Monte Carlo Maximum Likelihood. In: Keramigas, E.M. (Ed.), Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface. Interface Foundation, Fairfax, pp. 156–163.

Geyer, C.J., Thompson, E.A., 1995. Annealing Markov chain Monte Carlo with applications to ancestral inference. J. Am. Stat. Assoc. 90, 909–920.

Grassberger, P., 1997. Pruned-enriched Rosenbluth method: simulations of θ polymers of chain length up to 1000000. Phys. Rev. E 56, 3682–3693.

Hasegawa, M., Kishino, H., Yano, T., 1985. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. J. Mol. E 22, 160–174.

Hastings, W.K., 1970. Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57, 97–109.

Henikoff, S., Henikoff, J.G., 1992. Amino acid substitution matrices from protein blocks. Proc. Natl. Acad. Sci. U.S.A. 89, 10915–10919.

Holder, M., Lewis, P.O., 2003. Phylogeny estimation: Traditional and Bayesian approaches. Nat. Rev.: Genet. 4, 275–284.

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

Irwin, M., Cox, N., Kong, A., 1994. Sequential imputation for multilocus linkage analysis. Proc. Natl. Acad. Sci. U.S.A. 91, 11684–11688.

Jukes, T., Cantor, C., 1969. Evolution of protein molecules. In: Munro, H.N. (Ed.), Mammalian Protein Metabolism. Academic Press, New York.

Kidd, K.K., Sgaramella-Zonta, L.A., 1971. Phylogenetic analysis: concepts and methods. Am. J. Hum. Genet. 23, 235–252.

Kimura, M., 1980. A simple method for estimating evolutionary rate in a finite population due to mutational production of neutral and nearly neutral base substitution through comparative studies of nucleotide sequences. J. Mol. Biol. 16, 111–120.

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

Kishino, H., Miyata, T., Hasegawa, M., 1990. Maximum likelihood inference of protein phylogeny and the origin of chloroplasts. J. Mol. Evol. 31, 151–160.

Kocher, T.D., Conroy, J.A., MacKaye, K.R., Stauffer, J.R., Lockwood, S.F., 1995. Evolution of NADH dehydrogenase submit 2 in East African cichlid fish. Mol. Phylogenet. Evol. 4, 420–432.

Larget, B., Simon, D., 1999. Markov chain Monte Carlo algorithms for the Bayesian analysis of phylogenetic trees. Mol. Biol. Evol. 16, 750–759.

Li, S., Pearl, D., Doss, H., 2000. Phylogenetic tree construction using Markov chain Monte Carlo. J. Am. Stat. Assoc. 95, 493–508.

Liang, F., 2003. Use of sequential structure in simulation from high dimensional systems. Phys. Rev. E 67, 56101–56107.

Liang, F., Liu, C., Carroll, R.J., 2007. Stochastic approximation in Monte Carlo computation. J. Am. Stat. Assoc. 102, 305–320.

Liu, J.S., 2001. Monte Carlo Strategies in Scientific Computing. Springer.

Liu, J.S., Liang, F., Wong, W.H., 2000. The use of multiple-try method and local optimization in Metropolis sampling. J. Am. Stat. Assoc. 95, 121–134.

Maddison, D.R., 1991. The discovery and importance of multiple islands of most parsimonious trees. Syst. Zool. 40, 315–328.

Marinari, E., Parisi, G., 1992. Simulated tempering: a new Monte Carlo scheme. Europhys. Lett. 19, 451–458.

Mau, B., Newton, M.A., 1997. Phylogenetic inference for binary data on dendrograms using Markov chain Monte Carlo. J. Comput. Graph. Stat. 6, 122–131.

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

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E., 1953. Equation of state calculations by fast computing machines. J. Chem. Phys. 21, 1087–1091.

Newton, M.A., Mau, B., Larget, B., 1999. Markov chain Monte Carlo for the Bayesian analysis of evolutionary trees from aligned molecular sequences. In: Seillier-Moseiwitch, F. (Ed.), Statistics in Molecular Biology and Genetics: selected proceedings of a

1997 joint AMS-IMS-SIAM summer conference on statistics in molecular biology. IMS Lecture Notes-Monograph Series, vol. 33. Hayward, California: Institute of Mathematical Statistics; Providence, Rhode Island: American Mathematical Society, c1999, pp. 156–163.

Rannala, B., Yang, Z., 1996. Probability distribution of molecular evolutionary trees: a new method of phylogenetic inference. J. Mol. E 43, 304–311.

Rzhetsky, A., Nei, M., 1992. A simple method for estimating and testing minimum evolution trees. Mol. Biol. E 9, 945–967.

Saitou, N., Nei, M., 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol. Biol. E 4, 406–425.

Salter, L.A., Pearl, D.K., 2001. Stochastic search strategy for estimation of maximum likelihood phylogenetic trees. Syst. Biol. 50, 7–17.

Strimmer, K., von Haeseler, A., 1996. Quartet-puzzling: a quartet maximum-likelihood method for reconstructing tree topologies. Mol. Biol. Evol. 13, 964–969.

Strimmer, K., Goldman, N., von Haeseler, A., 1997. Bayesian probabilities and quartet-puzzling. Mol. Biol. Evol. 14, 210–211.

Wong, W.H., Liang, F., 1997. Dynamic weighting in Monte Carlo and optimization. Proc. Natl. Acad. Sci. U.S.A. 94, 14220–14224.

Yang, Z., Rannala, B., 1997. Bayesian phylogenetic inference using DNA sequences: a Markov chain Monte Carlo method. Mol. Biol. E 14, 717–724.