

Listening to Viral Tongues: Comparing Viral Trees Using a Stochastic Context-Free Grammar

Andrey Rzhetsky* and Walter M. Fitch†

*Department of Biomedical Informatics, Columbia Genome Center, and Center for Computational Biology and Bioinformatics, Columbia University, New York; and †Department of Ecology and Evolutionary Biology, University of California, Irvine

We suggest a probabilistic method for comparing the topological features of large phylogenetic trees. Using this method, we demonstrate that a stochastic grammar can generate three influenza-subtype (A H1, A H3, and B) hemagglutinin trees used in an earlier study, with statistically similar parameters. The proposed methodology is applicable to a broad class of problems that require comparison of the topological properties of various dendrograms.

Introduction

Phylogenetic trees reconstructed from alignments of viral proteins often vary drastically in their general appearance. Some trees are wide/bushy and short (acacialike), whereas others are slim and extended (cactuslike). (If two rooted trees are of different shape but have the same number of leaves, the longest path from root to a leaf in the cactus tree will be longer than in the acacia tree.) Furthermore, it appears that some trees have mostly short edges, whereas others tend to have longer edges. For surface-exposed viral proteins, such variation in the tree shape may reflect differences in the dynamics of interaction between viral-coat proteins and the host immune system. For example, differences in shape of trees reconstructed from alignments of hemagglutinin 1 protein of several subtypes (A H1, A H3, and B) of influenza virus recently led a group of researchers to postulate that these viral subtypes evolve differently (Ferguson, Galvani, and Bush 2003) (see Figure 1).

To explain the observed differences across these viral trees, Ferguson, Galvani, and Bush (2003) built an extensive model of dynamics of amino acid replacement in viral-coat proteins, taking into account human population structure, virus–host interactions, and temporal dynamics of a viral pandemic. The resulting complex model of viral evolution had quite a few parameters that were not readily estimable from the viral protein sequences or from the viral tree data and, therefore, had to be obtained elsewhere. We suggest an additional necessary step in their analysis: testing whether the tree topologies for different subtypes are indeed significantly different. For the purpose of implementing such a test, we developed a simple probabilistic model belonging to a class of stochastic context-free grammars that are frequently used in linguistics and computer science. Our model describes generation of each viral tree with a six-parameter stochastic process. Mathematically, the model that we consider here is analogous to a model of a multitype continuous-time branching process in population genetics (e.g., Harris 1963; Athreya and Keiding 1975; Lange and Fan 1997; Athreya and Ney 2004).

Key words: stochastic context-free grammar, human influenza virus hemagglutinin, stochastic measures of tree similarity, multitype branching processes.

E-mail: wfitch@uci.edu.

Mol. Biol. Evol. 22(4):905–913. 2005

doi:10.1093/molbev/msi074

Advance Access publication December 29, 2004

Materials, Methods, and Results

In a nutshell, a grammar is a mathematical model that allows generation of a set of strings (each string is a sentence, and their collection is a language) through a series of substitutions. A subset of permitted symbols that appear only in the final strings are terminal symbols, whereas symbols that appear only in the intermediate substitutions are nonterminal. In addition to allowed symbols, each grammar comes with a set of substitution rules (production rules) that can be probabilistic or deterministic. To illustrate the concept, let us consider a toy grammar with nonterminal symbols $\{S, Y\}$, terminal symbols $\{a, b\}$, and the following production rules (the probability of each rule is shown in parentheses).

$$S \rightarrow YY \ (1.0)$$
$$Y \rightarrow ab \ (0.1)$$
$$Y \rightarrow ba \ (0.9)$$

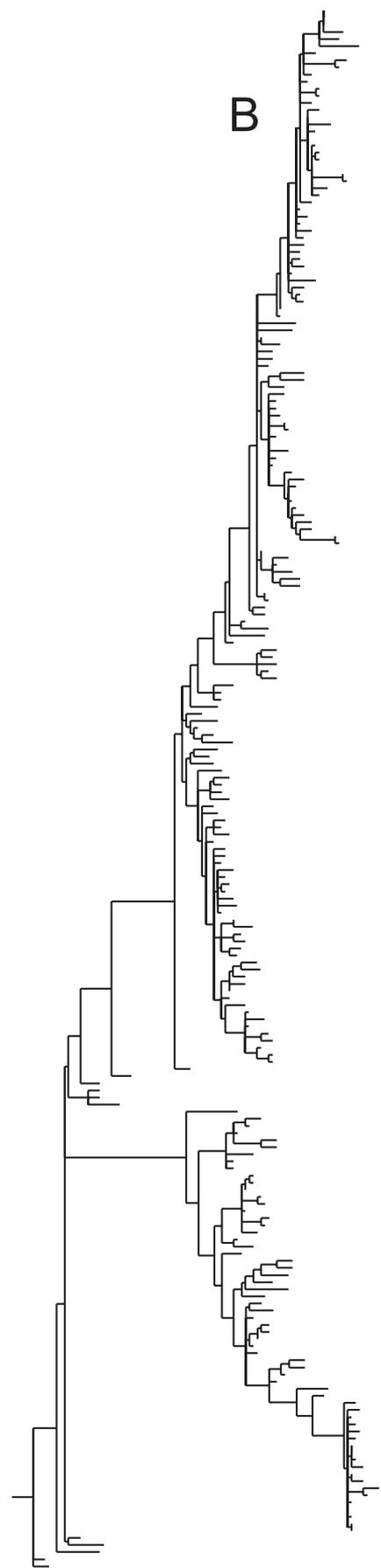
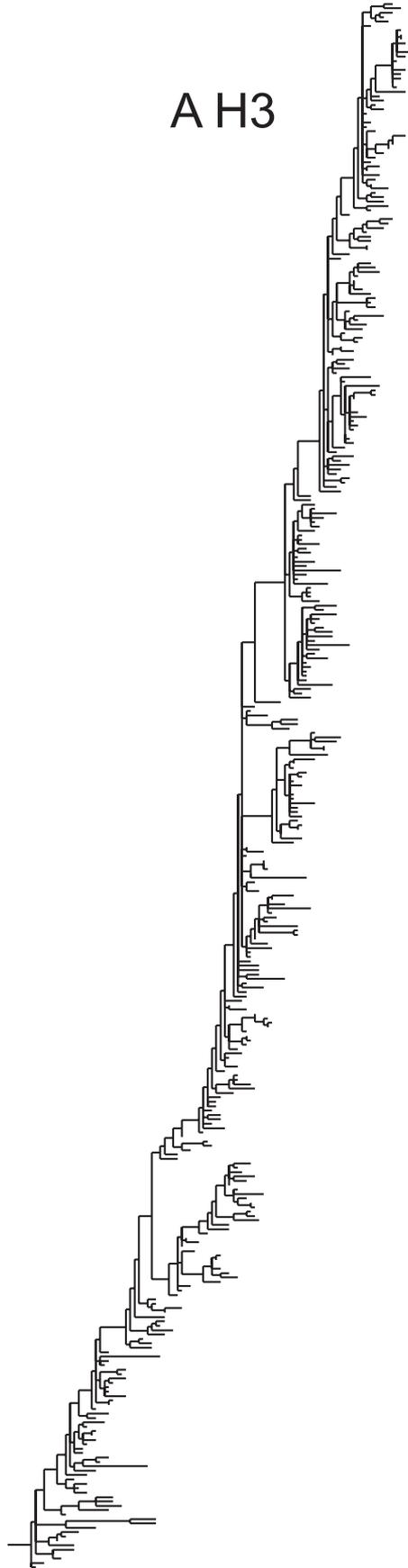
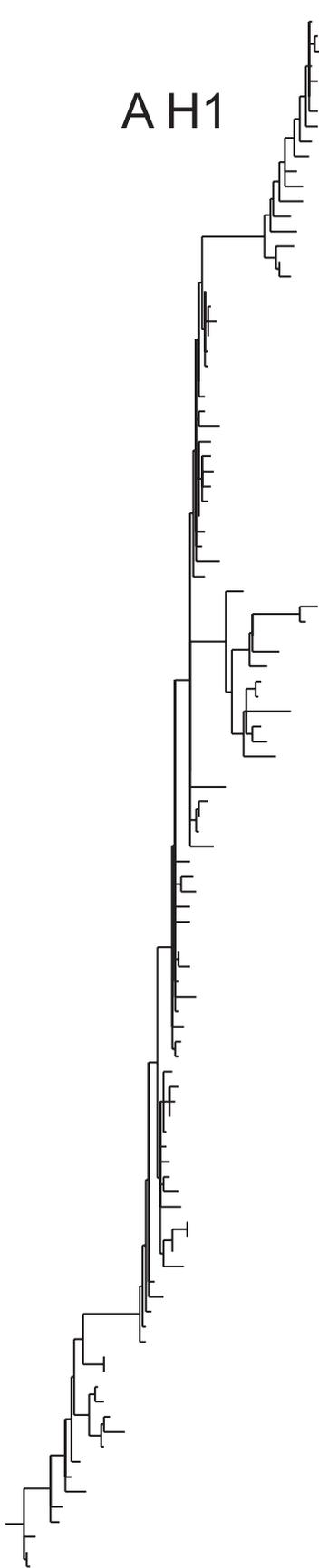
The grammar starts to generate sentences with symbol S , substitutes yy for S (the first production rule) and then equiprobably substitutes either ab or ba for each y , generating a language that comprises just four sentences (the probability of each sentence is shown in parentheses): $abba$ (0.09), $baab$ (0.09), $abab$ (0.01), and $baba$ (0.81).

A hierarchy of formal grammars, suggested by Chomsky (1956, 1959), comprises regular, context-free, context-sensitive, and unconstrained grammars, which we list here in order of increasing complexity. Each of the grammars in the list is a special case of all the grammars that follow it in the list. (For a detailed description of Chomsky's hierarchy, see e.g., Manning and Schütze [1999].) The computational cost of applying formal grammars rises quickly, from linear for regular grammars, to polynomial for context-free grammars, to exponential for more complex (context-sensitive and unrestricted) grammars. The context-free grammars are often used in practical applications because they are not only more powerful than regular grammars but also are less prohibitively expensive, in terms of computation, than the more complex grammars.

The original purpose of the formal grammars was generating sentences of a human language (such as English, see Chomsky [1956, 1959]). Thus, we can think of the viral trees generated by our stochastic grammar as languages or dialects.

Assumptions

Two different rooted trees are perceived as having different shapes for two reasons. First, in the cactus tree, if we



imagine how it grows, starting at the root, many lineages have one or no descendants. In contrast, in the growing acacia tree, most lineages have one or two descendants. Second, the cactus and acacia trees may have dissimilar distributions of branch length. For example, branches leading to nodes with offspring may be longer than branches leading to childless nodes in acacia tree but not in the cactus tree. Therefore, we make the following assumptions about the process that we are modeling:

The tree that we are generating is rooted (in the case of viral trees, we know the position of the root with certainty). The branches of the tree have two different distributions: short branches are terminal (the node ending the short branch died out), whereas the long terminal branches lead to nodes that could have continued to produce offspring if the process had run longer. The lengths of short and long branches come from two different stochastic distributions (which we assume to be either lognormal or gamma), with the mean value of short branches not exceeding the mean value of long branches.

Generation of Tree-Encoding Strings

The model that we describe here is suitable for generating any of numerous currently available tree formats. For the sake of descriptive simplicity, we have chosen to represent trees in the Newick format (Archie et al. 1986), which is itself an extension of a tree encoding suggested by the nineteenth-century English mathematician Arthur Cayley. For example, a three-species tree, shown in figure 2A, is represented in the Newick format as “((1:1,2:1):1,3:0.45)1’”. The topology of the tree is encoded as “((1,2),3),” indicating that species 1 is grouped with species 2, and species 3 is attached to the 1-2 cluster. Furthermore, branches of the tree (of lengths 1, 1, 1, 0.45, and 1) are encoded with the set of species corresponding to them—1, 2, and 3 for the leaves of the tree, and (1, 2) for the only interior branch of the tree that has the cluster 1, 2 beneath it. This encoding allows us to combine in a single string information about the tree topology and the branch lengths. The length of each branch is specified right after the encoding that corresponds to the cluster underneath that branch. In our tree-generating algorithm, we start by specifying the value of N —the size of the tree—where the size is defined as the length of the longest direct path through the tree starting at the root, or as the number of leaves minus 1.

Our context-free stochastic grammar has the following components.

Nonterminal symbols: $\{G, S, L, \#\}$,

Terminal symbols: $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 0, (,), ', : ', ; ', , ', \}$.

Production:

$G \xrightarrow{\alpha} (G:L, G:L)$ (ancestor produces two descendants)

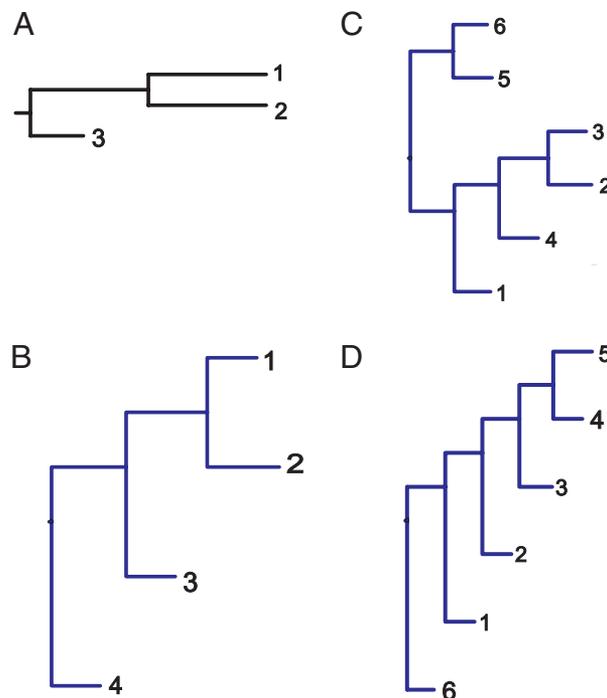


FIG. 2.—Hypothetical trees. (A) A tree for three species. (B) A tree corresponding to the Newick string generated by our grammar (see text for more information). (C) Two trees that represent the difference in tree topology that can be captured by our grammar: acacialike tree (C) and cactuslike tree (D).

$G \xrightarrow{2\beta} (G:L, \#:S)$ or $(\#:S, G:L)$ (ancestor produces one descendant)
 $G \xrightarrow{1-\alpha-2\beta} (\#:S, \#:S)$ (lineage goes extinct)
 $\# \xrightarrow{1} \text{numbers } (1:N)$

$$S \xrightarrow{f_1(x)} x,$$

$$L \xrightarrow{f_2(y)} y.$$

Note that we treat $(G:L, \#:S)$ and $(\#:S, G:L)$ as indistinguishable configurations, which is equivalent to assuming that the order within each pair of sister nodes is undefined. We considered two alternative forms of the probability densities.

$$f_i(x) = \frac{1}{\sigma_i \sqrt{2\pi}} e^{(\ln(x) - \mu_i)^2 / 2\sigma_i^2}, \tag{1}$$

and

$$f_i(x) = \frac{1}{b_i^{a_i} \Gamma(a_i)} x^{a_i-1} e^{-(x/b_i)}, \tag{2}$$

where $i = 1, 2$. These equations represent a lognormal and a gamma probability density, respectively. Therefore, we

FIG. 1.—Hemagglutinin trees of viral subtypes A H1, A H3, and B. These trees were analyzed first by Ferguson, Galvani, and Bush (2003). We recreate the trees using PAUP* (Swofford 1996) and visualized using TreeExplorer program written by Koichiro Tamura (http://evolgen.biol.metro-u.ac.jp/TE/TE_man.html).

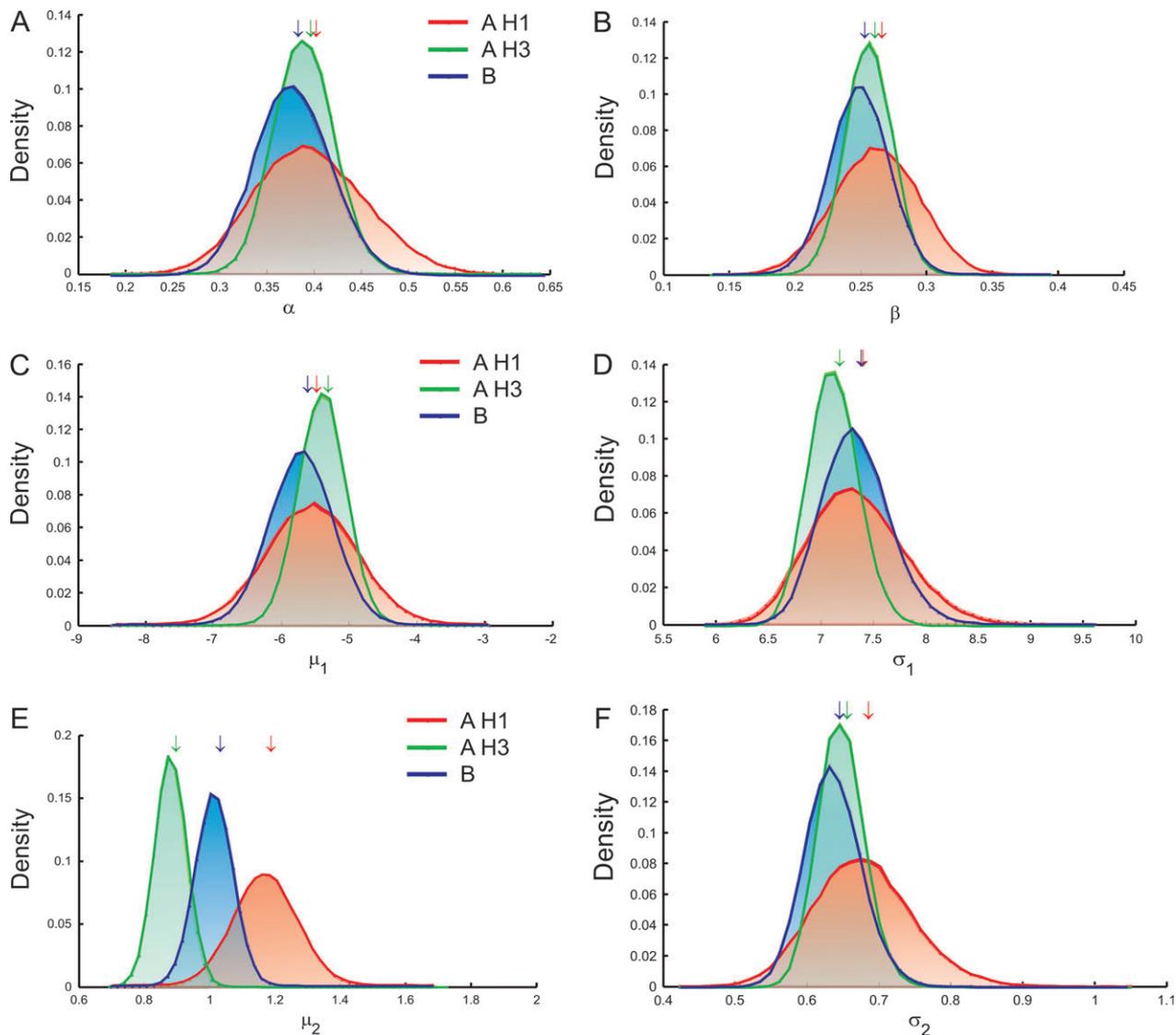


FIG. 3.—Analysis of viral trees under the lognormal-density with 6-parameters. Arrowheads indicate a posterior mean for different data sets.

have two versions of a six-parameter model, with parameters $\{\alpha, \beta, \mu_1, \sigma_1, \mu_2, \sigma_2\}$ and $\{\alpha, \beta, a_1, b_1, a_2, b_2\}$, respectively.

We then suggest the following six-step algorithm, which generates a string.

Step 1. Choose a value of N and begin with the string “S;”

Step 2. For $i = 1, N$ repeat

Apply one of the four production rules to a randomly chosen symbol G , selecting rules with probabilities specified by parameters α and β ; replace only one symbol G at each iteration.

If there are no more symbols G remaining, proceed to step 4.

End

Step 3. If the string has symbols G , do replacement; $G \rightarrow \#$. (Else, skip to step 4.)

Step 4. Assign $j = 1$.

While there are symbols $\#$ in the string

Find the leftmost symbol $\#$ in the string and substitute for it number j .

Increment j by 1.

End

Step 5. While there are symbols S in the string

Find the leftmost symbol S .

Sample a real number x from distribution $f_1(x)$.

Substitute symbol S with number x .

End

Step 6. While there are symbols L in the string

Find the leftmost symbol L .

Sample a real number y from distribution $f_2(y)$.

Substitute symbol L with number y .

End

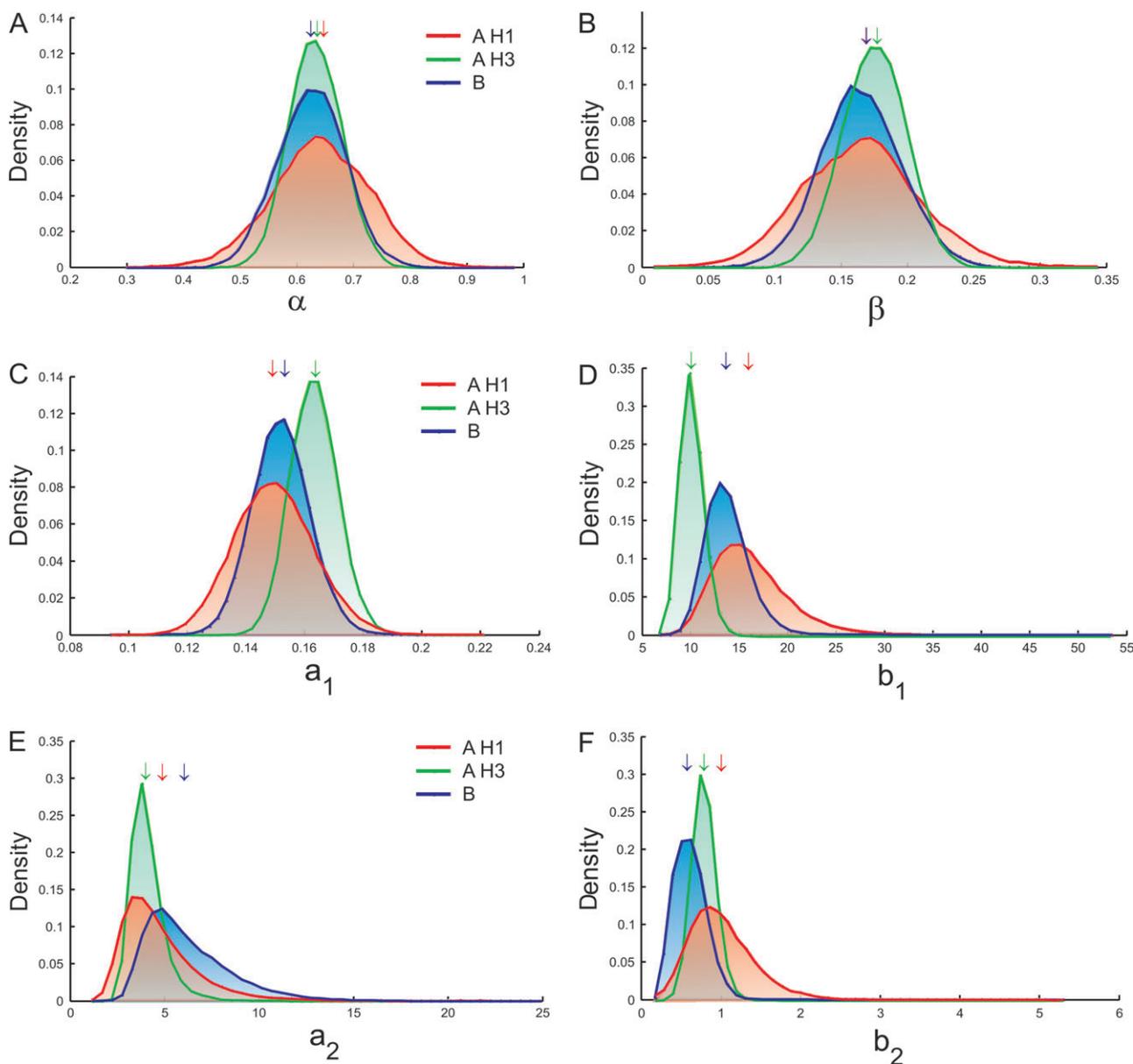


FIG. 4.—Analysis of viral trees under the gamma-density six-parameter model. Arrowheads indicate the a posteriori mean for different data sets.

As an example, we show the generation of a tree of height 3 (see figure 2B).

We start with a string “G;” (step 1 of the algorithm).

G;

Apply substitution $G \xrightarrow{2\beta} (G:L, \#:S)$ or $(\#:S, G:L)$ (step 2)

(G:L, #:S);

Again, substituting $G \xrightarrow{2\beta} (G:L, \#:S)$ or $(\#:S, G:L)$ (step 2)

((G:L, #:S):L, #:S);

Apply one more substitution corresponding to step 2 (remember that, $G \xrightarrow{2\beta} (\#:S, G)$ and $G \xrightarrow{2\beta} (G, \#:S)$ are indistinguishable)

(((\#:S, G:L):L, #:S):L, #:S);

Because the target depth 3 has been reached, apply substitution $G \rightarrow \#$ (step 3 of the algorithm).

(((\#:S, #:L):L, #:S):L, #:S);

Substitute symbols # with consecutive integers (step 4).

(((\mathbf{1:S, 2:L}):L, \mathbf{3:S}):L, \mathbf{4:S});

Substitute symbols S with real numbers sampled from f_1 (using a lognormal distribution, step 5).

(((\mathbf{1: 1.000, 2:L}):L, \mathbf{3:0.989}):L, \mathbf{4:0.987});

Substitute symbols L with real numbers sampled from f_2 (using a lognormal distribution, step 6).

(((\mathbf{1: 1.000, 2:1.453}):1.641, \mathbf{3:0.989}):1.511, \mathbf{4:0.987});

Our Newick tree string is ready to be visualized (see figure 2B). The advantage of using the Newick format is that we can apply directly various programs to convert the tree string into a tree picture. To illustrate application of this algorithm, let us consider two trees that were generated

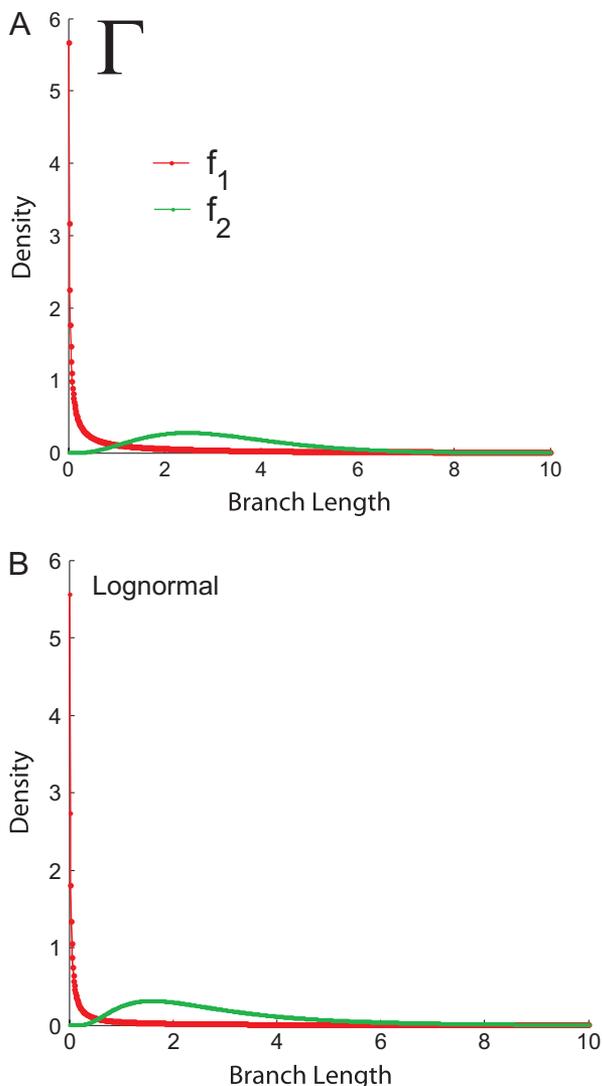


FIG. 5.—Branch length densities estimated under gamma (A) and lognormal (B) models with three parameters.

by two grammars and that are different in only the values of parameters α and β (see figure 3): The values of α approaching 1 tend to produce short, bushy trees (see figure 2C), whereas values of β approaching 0.5 tend to produce tall, skinny trees (see figure 2D).

Calculating Likelihood

First, we must go through the list of interior nodes of the tree. For this application, the viral tree must be rooted. This constraint is usually not a problem, because when viral strains are collected during successive epidemics, each viral strain comes with the exact date it was collected, so the root often can be placed to the edge leading to the strain that was collected the earliest. For each interior node, we can identify two incidental branches of the tree: the right branch of length $e_{i,r}$ and the left branch of length $e_{i,l}$. It is important only that we distinguish two branches; the labels that we use are irrelevant.

Assuming that our tree has M interior nodes, the likelihood (the probability of the data given the model and the model parameters) of our tree under our model can be written in the following way:

$$L(\Theta | T, \mathbf{e}) = \prod_{i=1}^M P(e_{i,r}, e_{i,l}). \quad (3)$$

To explain how to compute probability (3), let us introduce notations, $P(x \leftarrow f_1)$ and $P(x \leftarrow f_2)$, which stand for the “probability that x is sampled from distribution f_1 ” and “probability that x is sampled from distribution f_2 ,” respectively. Then, the likelihood value for each pair of edges can be written in the following way. If both the right and left branches are terminal, the likelihood is

$$\begin{aligned} P(e_{i,r}, e_{i,l}) = & \alpha P(e_{i,r} \leftarrow f_1) P(e_{i,l} \leftarrow f_1) \\ & + \beta P(e_{i,r} \leftarrow f_1) P(e_{i,l} \leftarrow f_2) \\ & + \beta P(e_{i,r} \leftarrow f_2) P(e_{i,l} \leftarrow f_1) \\ & + (1 - \alpha - 2\beta) P(e_{i,r} \leftarrow f_2) P(e_{i,l} \leftarrow f_2). \end{aligned} \quad (4)$$

If only one branch is terminal but the other one is not, we have

$$\begin{aligned} P(e_{i,T}, e_{i,N}) = & \alpha P(e_{i,T} \leftarrow f_1) P(e_{i,N} \leftarrow f_1) \\ & + \beta P(e_{i,T} \leftarrow f_1) P(e_{i,N} \leftarrow f_2), \end{aligned} \quad (5)$$

where subscripts T and N refer to terminal and nonterminal branches, respectively.

Finally, if both branches are nonterminal, the expression reduces to

$$P(e_{i,r}, e_{i,l}) = \alpha P(e_{i,r} \leftarrow f_1) P(e_{i,l} \leftarrow f_1). \quad (6)$$

Equation (4) has four terms because, when we work with a real tree, we do not know with certainty the actual production rule used to generate a pair of terminal tree branches, and, therefore, we have to sum through the probabilities that the forklike topological motif would be generated, using all possible productions for nonterminal symbol G .

The information provided in the preceding paragraphs should be sufficient to allow anyone proficient in computer programming to implement model analyses with the maximum-likelihood method. Such analyses would produce point estimates of the parameter values. In addition to these estimates, we would like to compute credible intervals (Bayesian equivalents of confidence intervals in classical statistics [e.g., see Howson and Urbach {1993}]). This computation would allow us to make statements about the significance of the difference in parameter estimates for different viral trees. Alternatively, we could use estimates of the information matrix at the point of the maximum likelihood to determine approximate confidence intervals. This second method, however, is known to produce liberal confidence intervals and is based on a frequently violated assumption, for real-life sample sizes, of the approximate normality of the likelihood function.

Thus, to determine Bayesian credible intervals, we would like to estimate the posterior density $P(\Theta | data)$.

Table 1
A Posteriori Mean Estimates and 95% Credible Intervals for Parameter Values of the Six-Parameter Model with Log-Normal Distribution of Branch Lengths

	α	β	μ_l	σ_l	μ_2	σ_2
A.h1: AM	0.40	0.26	-5.56	7.35	1.17	0.68
A.h1: 95% CI	[0.29 0.52]	[0.20 0.32]	[-6.89 -4.25]	[6.53 8.32]	[0.98 1.36]	[0.55 0.82]
A.h3: AM	0.39	0.26	-5.40	7.13	0.88	0.65
A.h3: 95% CI	[0.33 0.45]	[0.22 0.29]	[-6.09 -4.72]	[6.68 7.61]	[0.79 0.97]	[0.59 0.71]
B: AM	0.38	0.25	-5.71	7.33	1.02	0.64
B: 95% CI	[0.30 0.46]	[0.21 0.29]	[-6.63 -4.80]	[6.75 7.99]	[0.90 1.13]	[0.56 0.72]

NOTE.—Bold typeface indicates parameter estimates that are significantly different for different data sets. AM = a posteriori mean; CI = credible interval.

Calculation of Posterior Densities Using Markov Chain Monte Carlo

We compute the posterior density using the Bayes' theorem:

$$\begin{aligned}
 P(\Theta | data) &= \frac{P(data | \Theta)P(\Theta)}{P(data)} \\
 &= \frac{P(data | \Theta)P(\Theta)}{\int_{\text{admissible values of } \Theta} P(data | \Theta)P(\Theta)d(\Theta)}, \quad (7) \\
 \Theta &= (\alpha, \beta, \mu_1, \sigma_1, \mu_2, \sigma_2) \text{ or} \\
 \Theta &= (\alpha, \beta, a_1, b_1, a_2, b_2).
 \end{aligned}$$

To use equation (7), we have to specify the prior distribution of parameter values. As people often do in such cases, we can assume that no prior information is available about the parameter values, except that

$$\begin{aligned}
 \alpha + 2\beta &\leq 1, \\
 \alpha &\geq 0, \\
 \beta &\geq 0, \\
 -\Omega &\leq \mu_i \leq \Omega, \\
 0 &\leq \sigma_i \leq \Omega, \\
 0 &\leq a_i \leq \Omega, \\
 0 &\leq b_i \leq \Omega.
 \end{aligned} \quad (8)$$

where Ω is a large positive number that is less than positive infinity. Then, we can use uniform prior distributions for all parameters subject to constraints (8). Given a sufficient volume of data, the results of the computation of the posterior distribution under weakly informative priors would be similar to those under uninformative ones. As long as the prior

distributions occur both in the numerator and denominator of equation (7) and are uniform, they cancel out and our only concern is computing the likelihood function.

For large trees, equation (7) is too cumbersome for exact analytical treatment, and researchers commonly use an approximation to the integral. An estimate of an integral of a function can be obtained with a Monte Carlo integration. In its simplest version, the Monte Carlo integration involves generation of random points within a hypercube that encapsulates a surface of interest (the surface corresponds to the function under the integral sign). By estimating the proportion of points under the surface to the total number of randomly generated points and by knowing the volume of the hypercube, we can estimate the volume under the surface with arbitrarily high precision.

The Markov chain Monte Carlo (MCMC) algorithm is a more computationally efficient version of a stochastic integration (Metropolis et al. 1953; Hastings 1970; Gilks, Richardson, and Spiegelhalter 1996). It is based on the observation that Markov chains of the class called ‘‘positive ergodic Markov chains,’’ have the important property of converging to their stationary distribution starting at an arbitrary point in state space. Thus, we can start a time-reversible Markov process at an arbitrary point of the parameter space, then let it run for a large number of iterations. In this application, we used 100,000 iterations of full update of the parameter values after MCMC reached stationarity. Eventually, the stochastic process will reach the condition where all states will be visited in proportion to the stationary distribution of the chain. Therefore, if we can design a time-reversible Markov chain that has the required probability distribution as a stationary distribution (in our case this is $P(\alpha, \beta, \mu_1, \sigma_1, \mu_2, \sigma_2 | data)$ to estimate the distribution, we need only to compute the frequency with which each state of the Markov chain occurs in a long simulation.

Table 2
A Posteriori Mean Estimates And 95% Credible Intervals for Parameter Values of the Six-Parameter Model with a Gamma Distribution of Branch Lengths

	α	β	a_l	b_l	a_2	b_2
A.h1: AM	0.64	0.16	0.15	16.29	4.86	1.01
A.h1: 95% CI	[0.47 0.81]	[0.08 0.25]	[0.12 0.17]	[10.32 25.86]	[2.17 11.01]	[0.37 2.01]
A.h3: AM	0.63	0.18	0.16	10.23	4.11	0.79
A.h3: 95% CI	[0.54 0.72]	[0.13 0.22]	[0.15 0.18]	[8.13 12.85]	[2.85 6.33]	[0.51 1.10]
B: AM	0.62	0.16	0.15	13.82	6.20	0.62
B: 95% CI	[0.51 0.74]	[0.11 0.23]	[0.13 0.17]	[10.24 18.70]	[3.27 11.72]	[0.30 1.05]

NOTE.—AM = a posteriori mean; CI = credible interval.

Table 3
A Posteriori Mean Estimates and 95% Credible Intervals for Parameter Values of a Three-Parameter Version of the Model with a Gamma Distribution of Branch Lengths

	γ	a	b
A.h1: AM	0.99	0.18	15.54
A.h1: 95% CI	[0.96 1.]	[0.15 0.20]	[10.73 22.66]
A.h3: AM	0.997	0.19	10.04
A.h3: 95% CI	[0.989 1.]	[0.18 0.21]	[8.33 12.16]
B: AM	0.995	0.19	12.91
B: 95% CI	[0.982 1.]	[0.17 0.20]	[10.11 16.50]

NOTE.—AM = a posteriori mean; CI = credible interval.

More specifically, given the current values of parameters Θ , we need to define a probabilistic way to sample new parameter values Θ^* . We chose $q(\Theta^* | \Theta)$ as a uniform one-dimensional probability function that is symmetrical when parameter values are far from the boundary of the admissible values for the parameter and asymmetrical when its values are close to the boundary. We updated the parameters in a stepwise fashion, such that Θ differed from Θ^* by the value of a single parameter. We computed the probability of accepting the new parameter values Θ^* given old values Θ in the following way:

$$A(\Theta^*, \Theta) = \min \left[1, \frac{L(\Theta^*)q(\Theta | \Theta^*)}{L(\Theta)q(\Theta^* | \Theta)} \right]. \quad (9)$$

We made the actual decision whether to accept the new state by generating a random value of a uniformly distributed random variable; if the generated value was smaller than $A(\Theta^*, \Theta)$, we accepted the new state. $L(\Theta)$ in equation (9) stands for the likelihood value computed for parameter values Θ .

We updated one parameter at a time using the following transition function, which, as we mentioned, is symmetric uniform far from boundaries and is asymmetric uniform at the boundaries of the parameter values.

$$q(\theta^* | \theta) = \begin{cases} \frac{1}{2\theta}, & \text{if } (\theta < \psi \text{ and } \theta^* < \theta), \\ \frac{1}{2\psi}, & \text{if } (\theta > \psi \text{ and } \theta^* < \theta) \\ \text{or } (\theta + \psi \leq \theta_{\max} \text{ and } \theta^* > \theta), \\ \frac{1}{2(\theta_{\max} - \theta)}, & \text{if } (\theta + \psi > \theta_{\max} \text{ and } \theta^* > \theta). \end{cases} \quad (10)$$

Parameters ψ , θ , θ^* , and θ_{\max} represent the maximum jump size, the old value of a parameter, the new value of the parameter, and the upper limit for the parameter, respectively. (Note that we assume that the boundaries of the region, 0 and θ_{\max} , are excluded from the admissible region.)

Data Analysis

We applied our estimation procedure to the same three data sets used by Ferguson, Galvani, and Bush (2003). The data sets A.H1, A.H3, and B contain alignments of 104, 357, and 220 sequences (see figure 1), respectively. Following Ferguson, we estimated phylogenetic trees with the

maximum-parsimony method using PAUP* (Swofford 1996). To root the trees, we used as outgroups sequences X00027_A/USSR/90/77, A/Oita/3/83, and AB027392_B/Aichi/70/81 for viral subtypes A H1, A H3, and B, respectively. The resulting trees were saved in a Newick format and analyzed under the model described in the *Introduction*. All programs for this analysis were written in MatLab.

The results of our analyses are shown in figures 3–5 and in tables 1 and 2. In a nutshell, parameter estimates for all three data sets are essentially identical under both versions of our six-parameter model (all differences are non-significant). One respect in which two data sets (the A H1 and A H3 subtypes) are different from each other is the branch length distributions (see figures 3E and table 1). For the A H1 and A H3 subtypes, the branch-length densities f_2 have significantly (at the 95% level) different mean values (see table 1). Note also that, for all three data sets, the estimated densities f_1 and f_2 have significantly different parameter estimates (see figure 5 and tables 1 and 2); this difference indicates that the model in which the tree edge lengths are described by two different distributions agrees well with the data.

Discussion

It is easy to generalize our approach to fit a wide spectrum of problems related to statistical comparison of tree topologies. For example, we can consider a context-free grammar where generation of a tree-encoding string involves production rules that insert a spectrum of larger “motifs” of a tree. Such an approach is potentially applicable to testing arbitrarily complicated hypotheses in comparison of frequencies of topological motifs in treelike structures. There is no particular reason, other than computational complexity and ease of programming, for demanding that all production rules insert tree fragments of equal size; rather, we can define a spectrum of fragment sizes. Furthermore, biological applications may require that tree edges be sampled from more than two distinct distributions. We might also need to substitute, for a lognormal (or gamma) distribution of branch lengths, a different family of distributions, such as a negative binomial distribution for discrete data. (Strictly speaking, all edge lengths estimated with the maximum-parsimony method are discrete valued; other tree-making methods, such as the neighbor-joining algorithm [Saitou and Nei 1987], produce real-valued branch-length estimates.) For example, in our MCMC experiments, the gamma-density version of our six-parameter model appeared to fit the data significantly more closely than did the lognormal version: The average log-likelihoods under these two models for the same data set were different by 85 log-likelihood units for the B subtype data set. This result suggests that our model might be improved substantially by a study of alternative model setups.

The suggested methodology is not limited to viral trees or even to trees inferred from gene or protein sequences. It should be applicable to a general class of dendrograms utilized in social and natural sciences. As in the current application, the null hypothesis that two or more trees were generated by the same stochastic grammar is compared with

the hypothesis that the different trees were produced by significantly different grammars.

Our model also bears kinship to the Galton-Watson trees, where each node in a growing tree can produce k offspring ($k = 0, 1, 2, 3, \dots$) with a probability p_k that is the same for all nodes. We can convert our model to a special case of a Galton-Watson tree by setting $p_2 = \alpha$, $p_1 = 2\beta$, and $p_0 = 1 - p_1 - p_2$, or, by doing substitution $p_2 = \gamma^2$, $p_1 = 2\gamma(1-\gamma)$, $p_0 = (1-\gamma)^2$, to one-parameter version. The most common definition of the Galton-Watson trees does not allow for correlations between topological elements and edge lengths, such as we have in our model; however, we can reformulate the Galton-Watson model to account for edge-length variation. Context-free grammars that add large fragments of tree topology (e.g., a grammar using the following production rule $G \rightarrow ((G:L, G:L):L, (G:S, G:S):S):L$) are no longer equivalent to the Galton-Watson trees.

To exclude the possibility that our model is too parameter-rich to detect a difference in shape among the influenza trees, we analyzed a simplified one-parameter Galton-Watson model as just described, where $p_2 = \gamma^2$, $p_1 = 2\gamma(1-\gamma)$, $p_0 = (1-\gamma)^2$, and assumed that all edges of the tree were sampled from the same distribution. We found that this model also failed to detect significant differences among the trees (see table 3). Therefore, we currently cannot reject the hypothesis that all three viral trees considered in this study are identical in terms of overall tree topology (i.e., that they were generated by the same stochastic grammar).

Acknowledgments

This study was supported by grants from the National Institutes of Health, the National Science Foundation, the Department of Energy, and the Defense Advanced Research Projects Agency to A.R. We thank Dr. N. Ferguson for providing us with the three sets of aligned hemagglutinin sequences of human influenza, two anonymous referees, and Ms. Lyn Dupré, for providing numerous valuable comments that significantly improved the paper.

Literature Cited

- Archie, J., W. H. E. Day, W. Maddison, C. Meacham, F. J. Rolf, D. Swofford, and J. Felsenstein. 1986. Newick tree format., <http://evolution.genetics.washington.edu/phylib/newicktree.html>.
- Athreya, K. B., and N. Keiding. 1975. Estimation theory for continuous-time branching processes. University of Copenhagen, Institute of Mathematical Statistics, Copenhagen.
- Athreya, K. B., and P. Ney. 2004. Branching processes. Dover Publications, Mineola, N.Y.
- Chomsky, N. 1956. Three models for the description of language. *IRE Trans. Inform. Theory* **2**:113–124.
- . 1959. On certain formal properties of grammars. *Inform. Control* **1**:91–112.
- Ferguson, N. M., A. P. Galvani, and R. M. Bush. 2003. Ecological and immunological determinants of influenza evolution. *Nature* **422**:428–433.
- Gilks, W. R., S. Richardson, and D. J. Spiegelhalter. 1996. Markov chain Monte Carlo in practice. Chapman & Hall/CRC, New York.
- Harris, T. E. 1963. The theory of branching processes. Springer, Berlin.
- Hastings, W. K. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**:97–109.
- Howson, C., and P. Urbach. 1993. Scientific reasoning : the Bayesian approach. Open Court, Chicago.
- Lange, K., and R. Z. Fan. 1997. Branching process models for mutant genes in nonstationary populations. *Theor. Popul. Biol.* **51**:118–133.
- Manning, C. D., and H. Schütze. 1999. Foundations of statistical natural language processing. MIT Press, Cambridge, Mass.
- Metropolis, S. C., A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. 1953. Equation of state calculations by fast computing machines. *J. Chem. Phys.* **21**:1087–1092.
- Saitou, N., and M. Nei. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.* **4**:406–425.
- Swofford, D. L. 1996. PAUP*: phylogenetic analysis using parsimony (*and other methods). Sinauer Associates, Sunderland, Mass.

William Martin, Associate Editor

Accepted December 21, 2004