

## MARKOV CHAIN MONTE CARLO COMPUTATION OF CONFIDENCE INTERVALS FOR SUBSTITUTION-RATE VARIATION IN PROTEINS

ANDREY RZHETSKY<sup>1,2</sup> AND PAVEL MOROZOV<sup>1</sup>

<sup>1</sup>*Columbia Genome Center, and* <sup>2</sup>*Department of Medical Informatics, Columbia University,  
1050 St. Nicholas Avenue, Unit 109, New York, NY 10032, USA*

*{ar345, pm259@columbia.edu}*

We suggest a method implemented in a computer program, immodestly dubbed TSUNAMI, that allows us to compare two homologous protein subfamilies with respect to the distribution of substitution rates along sequences. This study furthers our earlier work on a wavelet model of rate variation (1). The current approach allows sensitive detection of subtle discordances in the selection patterns between two protein subfamilies. In addition to performing fast computation of the maximum posterior probability estimates of the relative substitution rates, the method can select the most appropriate number of wavelet parameters for a particular dataset. TSUNAMI is based on a Markov chain Monte Carlo technique, and appears to be more applicable to larger datasets than is the full likelihood-based approach.

### 1. Introduction

Amino-acid sites in real protein sequences differ in their degree of reluctance to accept an amino-acid substitution; furthermore, a particular site's degree of conservation is usually correlated with its functional importance. Therefore, analysis of relative rates of amino-acid substitution along a protein sequence is essential to our understanding of the evolution of protein function and to our ability to reconstruct phylogenetic trees.

Numerous mathematical models have been suggested to describe substitution rate variation (e.g., Durbin and colleagues (2) and the currently popular gamma-model (3, 4)). In a recent paper (1), we presented two alternative models — a wavelet and a Fourier model — that have the attractive property of flexibility in the number of parameters required to describe the rate variation across sites in a particular dataset. Here, we suggest a few enhancements that speed up computation under these models. We also demonstrate how to implement a Bayesian model-selection approach with regard to the wavelet model.

We use Markov chain Monte Carlo (MCMC) simulation to compute the posterior distributions of parameter estimates, the posterior-probability confidence intervals for substitution rates, and the maximum posterior density estimates of the parameter values. We speed-up the MCMC computation by fitting the estimates of posterior density to a three-parameter gamma distribution for each rate parameter separately. We implement an MCMC model-selection procedure and apply it to analysis of real sequence data.

## 2. Trees, likelihood, and pseudolikelihood

Before we specify the proposed modifications, we outline the conserved core of the accepted mathematical model of protein evolution.

Each set of truly homologous present-day sequences is a product of whole-gene duplications of a whole ancestral gene and local amino-acid substitutions, deletions, and insertions. For many reasons, deletion and insertion events are often excluded from the inference of a phylogeny for a set of homologous proteins, and the statistical modeling and data analyses both concentrate on only substitution events. In practice, the result is that the sites of a multiple alignment that contain at least one deletion character are excluded from analysis, so that the remaining differences among sequences can be explained completely by amino-acid substitution alone.

Researchers usually assume that amino acid substitution at each site of a protein sequence follows a Poisson process, whereas the rate of this process varies from site to site. For mathematical description, it is convenient to work with the *relative* substitution rates for sites, rather than with the absolute rate. The relative rates are simpler because the *absolute* expected number of amino substitutions per given protein site varies from one branch of the tree to another, whereas the *relative* substitution rate at each site can be conveniently assumed constant for all branches of the tree. The relative substitution rates for a dataset are usually defined such that the average relative substitution rate over all sites of each dataset is equal to 1. Then, the expected number of substitutions per a unit time at each individual site of the dataset is equal to the product of the relative rate for this site and to the average number of substitutions per site per unit time over all sites.

In a typical model, the substitution process is assumed to be homogeneous over time; that is, the intensity matrix,  $\mathbf{Q}$ , is postulated to be the same for all branches of the true evolutionary tree. Traditionally, the branch lengths of a tree are measured in terms of the number of amino-acid substitutions per site. As long as  $\mathbf{Q}$  remains constant, the expected lengths of edges are allowed to deviate from the molecular-clock behavior (when all expected root-to-tip distances are equal). The matrix of probabilities of substituting any amino acid for another amino acid over a time interval  $t$  at the  $i$ th site of protein alignment is computed as a matrix exponential of the product  $\mathbf{Q} t r_i$ , where  $r_i$  is the relative substitution rate at the  $i$ th site. Since  $\mathbf{Q}$  and  $t$  appear in the likelihood equation as an inseparable product, it is convenient to scale  $\mathbf{Q}$  such that the mean number of substitution per a unit of time is equal to 1 (e.g., see ref. 4). We compute the full likelihood of a tree as a conditional probability of the data given the model and a set of parameter values. In the case of just two sequences, 1 and 2, we compute the likelihood for the  $x$ th site as

$$L_x = \sum_i \pi_i P(i \rightarrow s_1(x), t_1 r_x) P(i \rightarrow s_2(x), t_2 r_x)$$

In this equation,  $\pi_i$  is the expected frequency of  $i$ th amino acid in the common ancestor of sequences 1 and 2, respectively;  $s_1(x)$  and  $s_2(x)$  are the integers (with

value between 1 and 20) corresponding to amino acids at the  $x$ th site of sequences 1 and 2, respectively, and notation  $i \rightarrow j$  indicates substitution of amino acid with label  $i$  with amino acid with label  $j$ .  $P(i \rightarrow s_1(x), t_1 r_x)$  is the probability of observing the ancestral amino acid  $i$  substituted with amino acid  $s_1$  after time  $t$ ; and  $r_x$  is the relative substitution rate at the  $x$ th site. Under a time-reversible model equation for likelihood over all sites simplifies to  $L = P(s_1(x) \rightarrow s_2(x), (t_1 + t_2)r_x)$ .

When we consider more than two sequence simultaneously, the equation for the likelihood function follows the tree topology. We must compute and multiply the probabilities of transitions along each branch of the tree, and then sum such products over all possible values of the unknown amino acids that belong to unobservable ancestral sequences in the interior nodes of the tree. There is a simple and elegant mapping from a ‘parentheses’ encoding of a tree to the matrix equation for  $L_x$ . For example, for a hypothetical tree with just four pending vertices - 1, 2, 3, and 4,  $((1, 2), 3), 4$  - we obtain  $L_x = \pi_x \times (\mathbf{P}_x^{(6)} \times (\mathbf{P}_x^{(5)} \times (\mathbf{P}_{S_1(x)}^{(1)} \bullet \mathbf{P}_{S_2(x)}^{(2)} \bullet \mathbf{P}_{S_3(x)}^{(3)} \bullet \mathbf{P}_{S_4(x)}^{(4)}))$ , where  $\pi_x$  is a row vector of expected frequencies of amino acids at the root of the tree,  $\mathbf{P}_x^{(i)} = \exp(\mathbf{Q} r_x t_i)$  is 20-by-20 matrix of transition probabilities corresponding to the  $i$ th branch of the tree,  $\mathbf{p}_j^{(i)}$  is the  $j$ th column of this matrix,  $\mathbf{A} \times \mathbf{B}$  indicates a regular matrix product, and  $\mathbf{A} \bullet \mathbf{B}$  indicates an elementwise product of two matrices of equal dimensionality,  $\mathbf{A} \bullet \mathbf{B} = (a_{ij} \times b_{ij})$ . In complex expressions, operators of elementwise multiplication have a precedence higher than that of operators of the regular matrix multiplication. Clearly, the parentheses pattern is the same in both expressions, commas in tree expression are mapped to pairwise multiplication operators, numerals that represent pending vertices are mapped into corresponding column vectors, and empty spaces between parentheses are mapped to operators of multiplication followed by a full transition probability matrix corresponding to an interior branch. (Matrix notation for likelihood computation comes in handy in MatLab programming, since MatLab computes matrix expressions much faster than it does equivalent scalar expressions.) For a more advanced description of maximum-likelihood analyses in phylogenetics and of the substitution models, please refer to other sources (e.g., ref. 2, pp.192-232: refs. 5, 6).

### 3. Wavelets and the wavelet model

A wavelet decomposition of a discrete function is a mathematical procedure that is used frequently in signal processing and statistical modeling. In addition to their abstract beauty, wavelets have the attractive property of allowing us to compress data by identifying and setting to zero most of the wavelet coefficients that make only small contributions to the signal (7).

We briefly introduce wavelets for readers new to this subject. Wavelets are local discrete functions. They are “local” in that each function has a well-defined *domain*, and outside of each wavelet, there is a contiguous subset of sites that have zero values. We chose for this study the simplest Haar wavelet, which has value of +1 at the

first half and a value of -1 in the second half of its domain. A wavelet domain length is always a power of 2 (2, 4, ..., full sequence) - which is why we had to extend the actual protein sequences in this analysis to the nearest power of 2 by adding dummy unvaried sites. Denoting by the  $l$  the sequence length, a complete set of wavelets in our analysis contains  $l/2$  wavelets with domain length 2, which cover the sequences without overlapping way;  $l/4$  wavelets with domain length 4, which again cover complete sequence but do not overlap each other; and so on, until we reach two wavelets with domain length  $l/2$  and a single wavelet with domain length  $l$ .

In our previously described wavelet model (1), relative substitution rates,  $\{r_x\}$ , where  $x$  indicates the site number, are assumed to be the following function of wavelet parameters,  $\{a_i\}$ :

$$r_x = \left(1 + \sum_y a_y \Psi(x, y)\right) \div \left(1 + \sum_i \sum_j a_i \Psi(i, j)\right)$$

In the present study, we change this definition slightly by dropping the normalization:

$$r_x = 1 + \sum_y a_y \Psi(x, y)$$

We can make this change because, when all wavelet parameters,  $\{a_j\}$ , are equal to zero, the average relative rate for all sites is equal to 1, as required. Further, by adding a padding of constant sites to the dataset to increase the total number of homologous sites to the nearest power of 2, and by forbidding combinations of wavelet parameters that produce negative relative rates, we can modify  $\{a_j\}$  while still preserving the normalization of the relative rates. (Do not worry about the dummy sites; at the end of the analysis, they are discarded, and the relative rates for remaining true sites are properly renormalized.) The possibility of avoiding renormalization speeds up the probability computation significantly. We obtain this speedup because, in the absence of renormalization, we have to recompute site likelihoods only for those sites that have their rates changed and, due to local nature of wavelets, the number of such sites is on average small. Renormalization of relative rates would force us to recompute all site likelihoods.

In addition to computing the honest likelihood function, we will analyze another objective function, pseudolikelihood (PL), which we define as follows:

$$PL = \prod_{i < j} (L_{ij})$$

Based on the observation that the pseudolikelihood is a product of honest two-sequence likelihoods, we expect that the mean values of rate estimates that we obtain by maximizing the pseudolikelihood function will be similar to those of the full likelihood. Since pseudolikelihood has only one product - whereas the full likelihood has a sum of numerous products - it is much faster to compute. Pseudolikelihood has the virtue that it does not depend explicitly on the tree topology. Moreover, because the

computation of pseudolikelihood is much cheaper, we can increase the number of sequences under analysis and thus can reach (almost) any desired level of precision of relative-rate estimation.

The pitfall is that the pseudolikelihood function corresponds to a rigorous likelihood function under a improbable model: Instead of assuming that the set of proteins evolved according to a full tree, this model assumes that the proteins evolved as a set of *independent* trees, each with two sequences. Since each sequence appears in  $(n-1)$  pairs, each is treated as  $(n-1)$  independent sequences. As a result, the amount of information obtained from the data is grossly overestimated, and estimates with perilously narrow confidence intervals are obtained. Nevertheless, we conjecture that the pseudolikelihood is appropriate for estimating wavelet parameters and for selecting an appropriate wavelet model.

#### 4. MCMC random walk

The idea of MCMC is startlingly simple yet powerful (8-11). We need only to organize a random walk through the space of parameter values such that a condition of *detailed balance* (or reversibility) is satisfied. More specifically, for the Metropolis–Hastings algorithm (8, 9), the system should go through a series of random states. We denote by  $X_t$  the state of the system at iteration  $t$ . Then, for each state  $X_t$ , we define a *proposal* distribution  $q(\cdot|X_t)$  from which a new candidate state will be sampled randomly. The candidate new state  $Y$  is accepted with probability

$$\alpha(X,Y) = \min[1, p(Y)q(X|Y) / \{p(X)q(Y|X)\}].$$

If the new state  $Y$  is not accepted, the system remains in the old state. To implement MCMC, we should be able to compute  $p(X)$  and  $p(Y)$  as the likelihood or pseudolikelihood of the corresponding state, and to compute the *proposal* probabilities  $q(X|Y)$  and  $q(Y|X)$ . A *healthy MCMC* satisfies the *reversibility* condition  $\pi(Y)q(X|Y)\alpha(Y,X) = \pi(X)q(Y|X)\alpha(X,Y)$ , as a result of definition of  $\alpha(X,Y)$ .

In our case, the parameter space is defined by the tree branch lengths (or, in the case of the pseudolikelihood, by the values of pairwise distances between sequences), by the wavelet parameters, and (optionally) by additional parameters of the amino acid substitution model chosen for simulation. In the data analysis described later in this study, we used the simplest model of amino-acid substitution - the Poisson model (12). The approach accommodates more sophisticated models easily.

A Bayesian statistical analysis requires explicit formulation of a prior distribution on parameter values and alternative statistical models (in our case, the alternative models are different tree topologies and distinct subsets of non-zero wavelet parameters). In this study, we used *uninformative prior distributions*; that is, we assumed *a priori* that all models and all parameter values are equally plausible. The posterior probability that we obtain in the end is therefore essentially an integration under the

likelihood function.

The restrictions on parameter values are as follows. Tree branch lengths (and, obviously, distances between sequences) must be nonnegative. The wavelet parameters in the general case can assume both positive and negative values. To restrict the parameter space and to solve the normalization problem simultaneously, we introduced a trick. We restricted values of the wavelet parameters to be nonnegative. Non-negative values of wavelet parameters would roughly correspond to a set of models where relative rates are non-increasing from the left to the right. (Obviously, it is always possible to arrange sites of a sequence alignment to achieve a non-increasing order of the relative substitution rates.) Simultaneously, we introduced a random swapping of rates between pairs of sites.

In our MCMC sampling, we always update one parameter at a time. For each parameter, we first define a proposal distribution, and sample a new value from this distribution. Therefore, the old parameter-value vector,  $X$ , is different from the new proposal vector,  $Y$ , by the value of only one parameter. Next, we compute pseudolikelihood values for  $X$  and  $Y$ , corresponding to probabilities  $\pi(X)$  and  $\pi(Y)$  in Equation 1, and probabilities  $q(X|Y)$  and  $q(Y|X)$  (we give the details of this computation later in this section). We accept the new state,  $Y$ , with probability  $\alpha(X, Y)$  (see Equation 1); if we do not accept it, we remain in state  $X$ . After we finish the update trials for all parameters, we save the current vector of parameter values.

We can estimate the posterior densities of parameter values with precision that depends on only the number of MCMC iterations – these posterior densities are computed as the frequencies of the parameter values that are observed in the random walk.

The individual parameter values in our analysis are updated in the following way.

**Tree branch lengths (the computation is the same for pairwise distances).**

Before simulation, we define the *maximum-branch-length jump*,  $\delta_b$ . Given the current state  $\delta_{old}$  for a particular branch length, we select either the plus or the minus direction with probability 0.5. **If the direction is minus**, we sample a uniformly distributed value from interval  $[\max(0, \delta_{old} - \delta_b), \delta_{old}]$ , then set  $q(X|Y)/q(Y|X) = [1/\min(\delta_{old}, \delta_b)]/[1/(1/\delta_b)]$ . **If the direction is plus**, we sample a uniform random value ( $\delta_{new}$ ) from interval  $[\delta_{old}, \delta_{old} + \delta_b]$ , and set  $q(X|Y)/q(Y|X) = [1/\min(\delta_b, \delta_{new})]/[1/\delta_b]$ .

**Wavelet parameters.** We define the maximum jump for the  $i$ th wavelet parameter as  $\delta_a/\Delta_i$ , where  $\Delta_i$  is the total number of sites where the  $i$ th wavelet has a nonzero value, and  $\delta_a$  is a parameter common for all wavelets. Further, we define as  $\Delta_i^-$  and  $\Delta_i^+$  the *sets of sites* where the  $i$ th wavelet is negative and where it is positive, respectively. We define  $\rho^-$  as the currently minimal relative substitution  $\rho$  for the set of sites belonging to  $\Delta_i^-$ , and we define  $\rho^+$  as the current minimal relative substitution rate for  $\Delta_i^+$ . We further define  $A_1 = \min(\rho^+, \delta_a/\Delta_i, a_{old})$ , and  $A_2 = \min(\rho^-, \delta_a/\Delta_i)$ .

We begin sampling the new value of each wavelet parameter by selecting, with equal probability, the positive or the negative direction of parameter-value change. **If the direction is negative**, we sample a random uniformly distributed value,  $a_{new}$

from interval  $[\max(0, a_{old} - A_1), a_{old}]$ . The ratio of probabilities  $q(X|Y)$  and  $q(Y|X)$  is computed as  $[1/\min((\Delta_i^+ + a_{new} - a_{old}), a_{new}, \delta_a/\Delta_i)]/[1/A_1]$ . **If the direction is positive**, we sample a uniformly distributed value ( $a_{new}$ ) from interval  $[a_{old}, a_{old} + A_2]$ , then set  $q(X|Y)/q(Y|X) = \{1/\min((\Delta_i^+ + a_{new} - a_{old}), a_{new}, \delta_a/\Delta_i)\}/\{1/A_2\}$ .

**Permutation of the sites.** Although the order of the sites is not a parameter of the model, performing site permutation without changing wavelet coefficients does affect values of the likelihood and pseudolikelihood and helps us to increase significantly the speed at which we reach the equilibrium distribution. As we go through all current occupants of the sites, we attempt to interchange each site with some other randomly chosen site. If change is accepted, the swapped sites inherit each other's relative rates. In this case, state  $X$  differs from state  $Y$  by the positions of only two sites. Clearly, in this case,  $q(X|Y) = q(Y|X)$  and the acceptance ratio in Equation 1 depend on only  $\pi(Y)$  and  $\pi(X)$ .

**Jumps between trees.** MCMC analysis of phylogenetic trees was first implemented relatively recently. In their this original application (13), Mau, Newton, andarget used an ingenious tree-encoding scheme, where stochastic changes in tree branch lengths led to changes in tree topology. It appears to us that this method can change no more than one tree partition at a time. (Each tree can be represented as a set of *partitions* of leaves of the tree. Elimination of any interior branch of the tree generates two disjoint subsets of the leaves, and every such pair of subsets is called a partition. A set of  $n-3$  partitions uniquely defines the topology of an unrooted bifurcated tree with  $n$  leaves.) In our computer application, we allow the user to specify the *maximum* number of tree partitions that can be changed in one step. The actual number of the tree partitions to be changed at each tree update was drawn from a uniform distribution between zero and the maximum. This strategy allowed the MCMC process to make either large or small changes to the tree topology, as required by the particular dataset. We implement generation of a *random tree topology that differed from an input tree topology by a specified number of partitions* as a random grouping of subtrees that we generated by removing a set of adjacent partitions from a tree. In each case, the partitions to be eliminated were chosen at random. Next, among the chosen partitions, groups of adjacent partitions were identified. We repeated the process of partition removal and generation sequentially for such partition groups, considering one group at a time.

**Avoidance of entrapment in local optima.** In all analyses of data with our MCMC method, we used multiple distinct starting points for each simulation type to avoid the danger of the stochastic process becoming trapped in a local maximum, leading to extremely biased estimates of the posterior probability.

## 5. Model selection

The wavelet model has a pleasing property that it allows us to drop unimportant parameters without renormalizing relative rates. But *which* parameters are unimportant

tant?

One way to make this determination is to extend our MCMC simulation to make *reversible jumps* between alternative models (14). In our case, the reversible jump is relatively easy to implement, because we can generate models nested to the full wavelet model by changing to zero those parameter values that are near zero, and holding them at zero until the random process reverts. (Recall, that in the full model, the number of wavelet parameters is equal to the number of alignment sites minus 1.)

We perform the walk through the models by switching the relative rate-variation parameters on and off using the MCMC scheme just described. That is, we go through the list of all wavelet parameters and try to change each one's state (on to off or vice versa).

For each switched-on wavelet parameter, we first determined whether we can switch it off — that is, whether we can set it to zero without generating negative relative-rate values. We cannot switch off parameter  $a_i$  if the result would be negative values of relative rates. In other words, we are not allowed to eliminate the parameter if  $a_i > \min(\Delta_i^+, \delta_a/\Delta_i)$ , where the values of  $\Delta_i^+$  and  $\delta_a/\Delta_i$  are defined for each rate parameter as described previously.

If we *are* allowed to switch off a parameter, we do switch it off with probability  $p_{off}$  (usually set to 0.5). We denote by  $q_{on}$  the probability of switching parameter on. We set  $q(X|Y) / q(Y|X) = (q_{on}/\min(\delta_a/\Delta_i, \Delta^- + a_{old})) / p_{off}$ .

If the parameter is switched off, we switch it on with probability  $q_{on}$  (usually also 0.5). Further, we sample a new parameter value from interval  $[0, \min(\delta_a/\Delta_i, \Delta_i^-)]$  and set  $q(X|Y) / q(Y|X) = p_{off}/(q_{on}/\min(\delta_a/\Delta_i, \Delta_i^-))$ .

To test the described methodology, we applied it to the data that we used in our earlier study (1).

## 6. Data analysis

Being a bit unimaginative, we applied the new tools to exactly the same data that we used in our earlier work (1). Here, space permits us to describe the analysis of only one of the datasets in detail. Curious readers will find in our earlier paper (1) a detailed discussion of the origins and genesis of the datasets. We implement the MCMC with true likelihood and pseudolikelihood calculation, as well as the model selection, in a set of programs - collectively dubbed TSUNAMI (a name inspired by the association with the wavelet models). TSUNAMI is written for MatLab 5.3.1 (R11) and is available on request from the authors.

### 6.1. Immunoglobulin datasets

Originally, we selected the immunoglobulin variable regions datasets for  $\kappa$  and  $\lambda$  sub-families of light chains for two reasons (1). The first reason is that such variable regions of immunoglobulins are notorious for extreme variation in substitution rates.

All sites in the variable regions are classified into two major groups: the hypervariable sites that define the specificity of the antibody, and the framework-region sites that bear responsibility for maintaining the basic structure of the variable domain. As their name suggests, hypervariable sites are expected to have an extremely high relative substitution rate, whereas framework-region sites are expected to have, on average, a low rate. The second reason for our choice is that we hoped to show a significant difference between substitution rates distribution in  $\kappa$  and  $\lambda$  subfamilies of immunoglobulins. In the previous study (1), although we demonstrated the first point successfully, we were not able to show a significant difference between two datasets.

To our delight, the pseudolikelihood rate-variation plots for the immunoglobulin subfamilies  $\kappa$  and  $\lambda$  were remarkably similar to those that we obtained with the honest likelihood-based MCMC.

### *6.2. Confidence intervals for relative rates obtained with the pseudolikelihood are narrower than those computed with the full likelihood*

In contrast to our original results, which we obtained with the full likelihood-based MCMC, pseudolikelihood-based confidence intervals for the relative substitution rates in the  $\kappa$  and  $\lambda$  immunoglobulin datasets indicated a significant difference between two datasets. Direct comparison demonstrated that the posterior maximum density confidence intervals are significantly narrower for the pseudolikelihood function. As we explained earlier, the pseudolikelihood computation implicitly assumes an improbable mathematical model, so the confidence intervals computed with the pseudolikelihood are misleading and should not be used for dataset comparison.

### *6.3. Marginal posterior distributions appear to be gamma-distributions*

Using pseudolikelihood MCMC estimates of marginal posterior distributions for both relative substitution rates and tree branch lengths we could afford the computation of many thousands of MCMC iterations, and thus obtained very smooth estimates of probability-density functions. We tried fitting these estimated density functions to several potentially suitable density functions: lognormal, Weibull, extreme value, and gamma. The *three-parameter gamma density*

$$p(x) = \frac{(x - \gamma)^{\alpha - 1} \exp(-(x - \gamma)/\beta)}{\Gamma(\alpha)\beta^\alpha}$$

gave a perfect fit to the posterior distributions for both types of parameter estimates. Posterior distributions obtained with the full likelihood were similar to their pseudolikelihood-generated counterparts and also appeared to be perfect gamma-distributions, although they were based on an order of magnitude smaller number of MCMC iterations and therefore were not as smooth (see Figure 1A-D).

The main use of the posterior distributions in our applications is for determining

the maximum posterior-probability estimates of parameters (each estimate of this kind corresponds to the mode of a marginal posterior distribution) and for inferring the maximum posterior density confidence intervals. Both inferences can be done with far greater precision when results of MCMC computations are fitted to an analytically defined density function (see Figure 1E). This is especially important for the inference of confidence intervals, because in the absence of an analytical function an accurate analysis of the tails of posterior distributions requires an enormous number of MCMC iterations.

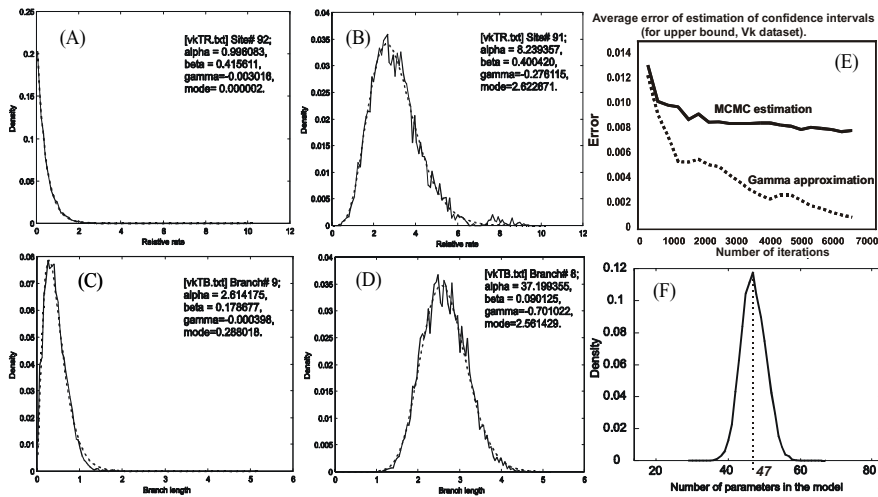


Figure 1. Full likelihood MCMC-estimated marginal posterior density functions (shown with continuous lines) (A and B) for relative rate variation parameters and C and D for the branch-length parameters. The dotted lines show the fitted gamma-densities functions. (E) Average error of estimation of maximum posterior density confidence intervals with (dotted line) and without (solid line) gamma-density approximation. Gamma-approximation significantly improved the precision of the estimates for the same number of MCMC iterations. (F) Distribution of wavelet parameter number in model selection with pseudolikelihood function; 19,000 MCMC iterations were used for computing this distribution.

#### 6.4. Model selection

The MCMC program designed for model selection behaved in a similar way with all analyzed datasets. We describe in detail only the analysis of immunoglobulin  $\kappa$

sequences.

Pseudolikelihood-based MCMC iteration started with a large number of model parameters ( $> 65$  - for a set of sequences where only 57 sites were variable!). After about a thousand burn-in iterations, the random walk reached what appeared to be an equilibrium state. The mode of the rate parameter number distribution (see Figure 1F) was equal to **47**, which is intuitively a reasonable value. A few thousands more MCMC iterations showed that the average pseudolikelihood slowly increased and that the mode of the rate parameter number distribution moved steadily towards the smaller values. Therefore, we concluded that we are looking at an example of pathologically slow convergence of an MCMC random walk.

Full likelihood-based model selection showed similar tendencies - it started with large number of parameters (about 70) that, after approximately 500 iterations, dropped below 57. We were not able to study full likelihood model selection for as many iterations as we completed with the pseudolikelihood version, but we could see that it would take hundreds, if not thousands, of additional iterations for the MCMC random walk to reach equilibrium.

Although we are currently unable to prove this conjecture generally, we suspect that the wavelet-model selection with pseudolikelihood-based MCMC gives results similar to those of MCMC model selection with the full likelihood. We expect that the pseudolikelihood version of model selection is slightly more conservative - that is, it favors more parameter-rich models - than the full-likelihood version of the same procedure. We expect it to be more conservative because the pseudolikelihood function corresponds to the assumption that data contain much more information than they do under the full likelihood model. Therefore, we conjecture that it is acceptable to use the pseudolikelihood function for the wavelet-model selection.

#### *6.5. MCMC with jumps among alternative trees.*

For all datasets that we analyzed with MCMC that involved jumps between phylogenetic trees, the optimal tree was reached surprisingly fast (usually in fewer than 100 MCMC iterations, even when the starting tree was different from the optimum tree at every partition). After reaching the optimum tree, the MCMC process was always trapped there permanently. There must exist datasets that contain little information about the correct tree, however; for them, the MCMC process should switch indecisively among alternative trees until the end of simulation.) Therefore, the results that we obtained earlier (1) using MCMC with a fixed (optimum) tree should be identical to the results obtained with full MCMC.

## **7. Conclusion**

We developed and implemented a new computational tool that raises our hopes that

we can make useful the wavelet model of rate variation. Although we displayed an apparent obsession with the analysis of proteins in this study, the method presented here is also applicable to evaluation of rate variation in nucleotide sequences.

In the future we could modify the model-selection routine to include *reversible jumps* from a wavelet model to a gamma-model (15) and back. Such jumps should make possible model selection in a more general framework.

The exercises that we performed in this study with the wavelet model should be completely reproducible with a Fourier model (1), although the computational price would be higher due to the non-local nature of the discrete Fourier decomposition.

The statistical comparison of rate variations in two homologous protein subfamilies appears to have been almost completely overlooked (one exception is Gu's method (16)) by other published mathematical approaches. Thus, our algorithm is an important addition to the current arsenal of research tools in genomics and evolutionary biology.

## 8. References

1. Morozov, P., Sitnikova, T., Churchill, G., Ayala, F. J. & Rzhetsky, A. (2000) *Genetics* **154**, 381-395.
2. Durbin, R., Eddy, E., Krogh, A. & Mitchison, G. (1998) *Biological Sequence Analysis*. (Cambridge University Press, Cambridge).
3. Golding, G. B. (1983) *Mol Biol Evol* **1**, 125-142.
4. Yang, Z., Goldman, N. & Friday, A. E. (1995) *Systematic Biology* **44**, 384-399.
5. Felsenstein, J. (1981) *J Mol Evol* **17**, 368-376.
6. Tavaré, S. (1986) *Lectures in Mathematics in the Life Sciences* **17**, 57-86.
7. Daubechies, I. (1988) *Wavelets* (S.I.A.M., Philadelphia).
8. Metropolis, S. C., Rosenbluth, A., Rosenbluth, M., Teller, A. & Teller, E. (1953) *J Chem Phys* **21**, 1087-1092.
9. Hastings, W. K. (1970) *Biometrika* **57**, 97-109.
10. Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. (1983) *Science* **220**, 671-680.
11. Gilks, W. R., Richardson, S. & Spiegelhalter, D. J. (1996), (Chapman & Hall/CRC, New York).
12. Zuckerkandl, E. & Pauling, L. (1965) in *Evolving Genes and Proteins*, eds. Bryson, V. & Vogel, H. J. (Academic Press, New York), pp. 97-166.
13. Mau, R., Newton, M.A., & Larget, B. (1996) Technical Report #961. Department of Statistics, University of Wisconsin at-Madison, Madison, OH.
14. Green, P. J. (1994), (Dept. of Matematics, University of Bristol., Bristol, Geat Britain).
15. Yang, Z. (1993) *Mol Biol Evol* **10**, 1396-1402.
16. Gu, X. (1999) *Mol Biol Evol* **16**, 1664-1674.