Gibbs sampler for statistical multiple alignment

Jens Ledet Jensen

Department of Theoretical Statistics, Institute of Mathematics,

Ny Munkegade, DK-8000 Aarhus C, Denmark.

Phone: +45 89423526   Fax: +45 86131769   Email: jlj@imf.au.dk

Jotun Hein

Department of Statistics, Oxford University,

The Peter Medawar Building for Pathogen Research,

South Parks Road, Oxford OX1 3SY, England

Abstract

For a set of sequences related by a binary tree that have evolved according to the Thorne-Kishino-Felsenstein model a Gibbs sampler is presented for simulating the ancestral sequences and their alignments. The updating step consists in updating the ancestral sequence and its three alignments within a 3-star tree. We compare the Gibbs sampler with the algorithm suggested recently by Holmes and Bruno.

Keywords: evolutionary model, EM-algorithm, hidden Markov model, Markov chain Monte Carlo, 3-star tree

Running title: Gibbs sampler for multiple alignment

Corresponding author: Jens Ledet Jensen. Phone: +45 89423526; Fax: +45 86131769

1 Introduction

In this paper we describe a Markov Chain Monte Carlo method for sampling multiple sequence alignments within the Thorne, Kishino, and Felsenstein (1991) model (*TKF-model*) on a binary tree. We use a Gibbs sampler where in each step an ancestral sequence and its alignments with the three neighbours is simulated conditional on the three neighbouring sequences. While developing the method described in this paper the article by Holmes and Bruno (2001) appeared. In that paper a Gibbs sampler is given that is computationally simpler than the one described in this paper. In each step Holmes and Bruno (2001) either update the alignment between two sequences given the two sequences or update the ancestral sequence given its alignments with the three neighbours and at the same time allowing the insertion of new letters in the ancestral sequence that are not aligned to any of the letters in the three neighbours (see Appendix A below). We compare our method with that of Holmes and Bruno (2001) in terms of mixing properties and the efficiency of the two approaches.

1.1 TKF-model

In the TKF-model (Thorne, Kishino, and Felsenstein, 1991) each letter in a sequence develops independently of the other letters according to a birth and death process with birth rate λ and death rate $\mu > \lambda$. When a new letter is born it is inserted to the right of the letter giving birth. The new letter is chosen according to a distribution π. At the very left end of the sequence is a birth process with rate λ (immigration) so that the sequence will not eventually die out. While a letter is alive it is subject to a Markovian substitution process with stationary probabilities given by π and with the transition probability of a change form w_1 to w_2 within a time span τ given by $f(w_2|w_1;\tau)$. The stationary distribution of a sequence S of length L is given by

$$P(S) = (1-\gamma)\gamma^L \prod_{i=1}^{L} \pi(S[i]), \quad \gamma = \frac{\lambda}{\mu}, \tag{1}$$

where $S[i]$ is the ith element in the sequence.

If a sequence S_1 evolves into S_2 during a time span τ we can summarise the evolution in terms of the alignment of some of the letters in S_1 with some of the letters in S_2

(survival of these letters), in terms of deletions (deaths) of some of the letters, in terms of insertions (births), and finally in terms of substitutions for the aligned letters. The TKF-model for this summary information can be reformulated as a hidden Markov model (Durbin, Eddy, Krogh, and Mitchison 1998). The three basic states for the underlying Markov chain can symbolically be given as

$$\binom{\#}{\#}, \quad \binom{\#}{-}, \quad \text{or} \quad \binom{-}{\#}, \tag{2}$$

corresponding to survival, insertion and deletion, respectively. Thus # denotes the presence of a symbol (nucleotide or amino acid) and $-$ denotes the absence of a symbol. Apart from the three states above there is also an *end state* in order to model the random lengths of the sequences. To give the transition probabilities in the Markov chain we define

$$\begin{aligned}
&\beta = \frac{1-\exp((\mu-\lambda)\tau)}{1-\gamma\exp((\mu-\lambda)\tau)}, \quad \gamma = \frac{\lambda}{\mu}, \\
&b(\#,\#) = \gamma\beta, \quad b(\#,-) = 1-b(\#,\#), \\
&b(-,\#) = 1-\frac{\beta}{1-\exp(-\mu\tau)}, \quad b(-,-) = 1-b(-,\#), \\
&s(\#) = \exp(-\mu\tau), \quad s(-) = 1-s(\#),
\end{aligned} \tag{3}$$

where $b(\cdot,\#)$ is the probability of having a birth, $b(\cdot,-)$ is the probability of not having a birth, and $s(\#)$ is the probability of survival. The transition probabilities are then,

	$\binom{\#}{\#}$	$\binom{\#}{-}$	$\binom{-}{\#}$	End
$\binom{\#}{\#}$	$b(\#,-)\gamma s(\#)$	$b(\#,-)\gamma s(-)$	$b(\#,\#)$	$b(\#,-)(1-\gamma)$
$\binom{\#}{-}$	$b(-,-)\gamma s(\#)$	$b(-,-)\gamma s(-)$	$b(-,\#)$	$b(-,-)(1-\gamma)$
$\binom{-}{\#}$	$b(\#,-)\gamma s(\#)$	$b(\#,-)\gamma s(-)$	$b(\#,\#)$	$b(\#,-)(1-\gamma)$

(4)

A state emits letters in those position where we have the symbol #. Thus, in the state $\binom{\#}{-}$ a letter is emitted in sequence S_1 and the distribution of the letter is $\pi(\cdot)$, in the state $\binom{-}{\#}$ a letter is emitted in sequence S_2 also from the distribution $\pi(\cdot)$, and in the

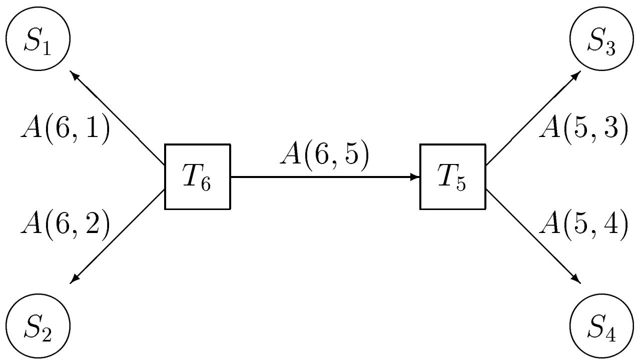


Figure 1: A tree with four observed sequences.

state $\binom{\#}{\#}$ a letter is emitted in both sequences and the distribution of the letters $\binom{w_1}{w_2}$ is $\pi(w_1)f(w_2|w_1;\tau)$.

The immigration part of the model is incorporated by saying that the Markov chain starts in the state $\binom{\#}{\#}$ and this initial state does not emit any letters (also called the immortal state).

In Hein, Jensen and Pedersen (2002) a detailed description is given of how to formulate the TKF-model on a binary tree as a hidden Markov model. We will use this below for the special case of a 3-star tree.

1.2 Notation and Gibbs idea

We have η observed sequences $S_1, \dots, S_\eta$ one for each leaf of a binary tree with ν interior nodes. The unobserved sequences at the inner nodes are denoted $T_{\eta+1}, \dots, T_{\eta+\nu}$. The root of the tree is taken as the interior node numbered $\eta+\nu$. Any interior node $\eta+1 \le i < \eta+\nu$ has an ancestor $a(i)$ among the interior nodes $i+1, \dots, \eta+\nu$ and two descendants $d_1(i)$ and $d_2(i)$ among the interior nodes $\eta+1, \dots, \eta+i-1$ and the leaves. For the root the ancestor $a(\eta+\nu)$ is replaced by a descendant. For a leaf j the ancestor $a(j)$ is among the interior nodes. An example of a tree with 4 observed sequences is given in Figure 1 and an example with 7 observed sequences is given in Figure 2.

The branch from the node $a(j)$ to the node j is numbered j so that the set of branches is $j = 1, \dots, \eta+\nu-1$. Branch number j has a length τ_j and an alignment $A(a(j), j)$ consisting of a sequence with terms as in (2).

The TKF-model gives the joint probability of all the sequences $(T_{\eta+1},\ldots,T_{\eta+\nu})$, $(S_1,\ldots,S_\eta)$, and all the alignments $A(a(j),j)$. In this paper we consider simulation of $T_{\eta+1},\ldots,T_{\eta+\nu}$ and the alignments $A(a(j),j)$ conditionally on the value of the observed sequences $(S_1,\ldots,S_\eta)$. Each step in the simulation consists in simulating a 3-star tree conditionally on the sequences at the three leaves. Thus for each $r=\eta+1,\ldots,\eta+\nu$ we consider the 3-star with interior node r and leaves $a(r), d_1(r), d_2(r)$ and simulate a new value of the sequence T_r and the alignments $A(r,a(r))$, $A(r,d_1(r))$, and $A(r,d_2(r))$, from the conditional distribution given the sequences at the three leaves. This conditional distribution is given in (13) below. For the tree in Figure 1 we simulate the 3-star tree with node 5 being the interior node and next the 3-star tree with node 6 being the interior node.

To get initial values of the interior sequences $T_{\eta+1},\ldots,T_{\eta+\nu}$ we use an algorithm for simulating a 2-star tree equivalent to the one described below for a 3-star tree. In particular we have a formula equivalent to (13) for the sequential simulation of an interior sequence. For $r=\eta+1,\ldots,\eta+\nu$ we simulate T_r given the sequences at $d_1(r)$ and $d_2(r)$.

2 3-star tree

2.1 States and transition probabilities

We consider a 3-star tree where we let T be the sequence at the interior node and let S_1, S_2, S_3 be the sequences at the three leaves. The evolutionary time distances along the branches are τ_1,τ_2,τ_3. In this section we describe the TKF-model for the 3-star tree as a hidden Markov chain. Each state is an alignment column with 4 entries, the first corresponding to the interior node, which we number by 0, and entries 1 to 3 corresponding to the 3 leaves. Each entry is either # or $-$, and a state emits a letter in those sequences where we have # in the state.

Apart from the end state there are two classes of states given by

$$\begin{pmatrix} \# \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} - \\ x_1 \\ x_2 \\ x_3 \end{pmatrix}, \tag{5}$$

where $x_i \in \{\#, -\}$, $i=1,2,3$. The first set of states correspond to having a letter in the

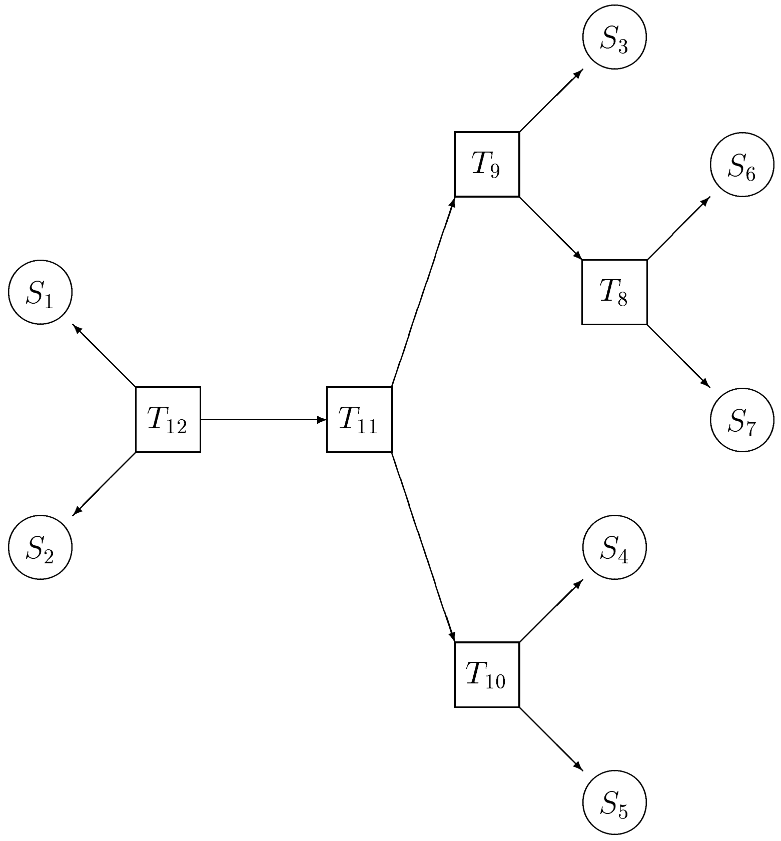


Figure 2: A tree with seven observed sequences.

ancestral sequence T and recording the possible survival ($x_i = \#$) or nonsurvival ($x_i = -$) in leaf i. The second set of states correspond to births (insertions) at the three leaves and we require that $x_j = \#$ for at least one j. From a state in the first set we can go to any state in the two sets. From a state x in the second set we can go to any state in the first set, but only to states y in the second set for which $x_j = -$ implies that $y_j = -$, $j = 1, 2, 3$. The set of 15 states in (5) are denoted by Ξ.

To state the transition probabilities we define $\beta(j)$, $b(\#, \#; j)$, $b(\#, -; j)$, $b(-, \#; j)$, $b(-, -; j)$, $s(\#; j)$, and $s(-; j)$ as in (3) with τ replaced by τ_j, $j = 1, 2, 3$. The transition probability $p(x,y)$ of going from the state x to the state y is then

	$y_0 = \#$	$y_0 = -$	$y = \text{End}$
$x_0 = \#$	$B(\#,\#)\gamma\left(\prod_{j=1}^{3} s(y_j;j)\right)$	$B(\#,-)$	$B(\#,\#)(1-\gamma)$
$x_0 = -$	$B(-,\#)\gamma\left(\prod_{j=1}^{3} s(y_j;j)\right)$	$B(-,-)$	$B(-,\#)(1-\gamma)$

(6)

with

$$B(\#,\#) = \prod_{j=1}^{3} b(x_j,-;j), \quad B(\#,-) = \prod_{j=1}^{3} b(x_j,y_j;j),$$
$$B(-,\#) = \prod_{\{j:x_j=\#\}} b(\#,-;j), \quad B(-,-) = \prod_{\{j:x_j=\#\}} b(\#,y_j;j).$$

The initial state has # on all the four entries and do not emit any letters.

A state x of the form (5) emits a letter in those sequences j for which $x_j = \#$. Let the emitted letters be $w = (w_j : j = 0,1,2,3,4)$, where w_j is the empty set if $x_j = -$. Then the probability of w given the state x is

$$p_e^0(w|x) = \begin{cases} \pi(w_0)\prod_{\{j:x_j=\#\}} f(w_j|w_0;\tau_j) & x_0 = \# \\ \prod_{\{j:x_j=\#\}} \pi(w_j) & x_0 = - \end{cases} \tag{7}$$

We will also be using the marginal probability of (w_1,w_2,w_3) given the state x obtained by summing over w_0 in the previous expression

$$p_e((w_1,w_2,w_3)|x) = \begin{cases} \sum_{w_0} \pi(w_0)\prod_{\{j:x_j=\#\}} f(w_j|w_0;\tau_j) & x_0 = \# \\ \prod_{\{j:xj=\#\}} \pi(w_j) & x_0 = - \end{cases} \tag{8}$$

2.2 Simulating a 3-star tree

We denote the length of the sequence S_j by L_j, $j = 1,2,3$. A subsequence starting in a and ending in b is denoted $S_j[a:b]$. If $a > b$ we interpret $S_j[a:b]$ as the empty set. For column vectors u and v with integer entries we let $S[u:v]$ denote the three subsequences $S_j[u_j:v_j]$, $j = 1,2,3$. For a state x we define a vector $l(x)$ and a number $t(x)$ by

$$l(x)_j = 1(x_j = \#), \quad j = 1,2,3, \quad \text{and} \quad t(x) = 1(x_0 = \#).$$

Finally, we let I denote the state having # at all four entries and let $\mathcal{E}$ be the end state.

The multiple alignment for a 3-star tree is given through the states $x^0, x^1, \ldots, x^N$, where $x^0 = I$ is the initial state that do not emit any letters, $x^i \in \Xi$, $i = 1, \ldots, N$, and x^{N+1} is the end state. Here N is random. Let

$$L^i = l(x^1) + \cdots + l(x^i), \quad t^i = t(x^1) + \cdots + t(x^i).$$

Thus L_i is the part of the sequences in S explained by the first i states of the alignment. We can write the joint probability of the sequences and the alignment as

$$\begin{aligned} &P(N = n, x^1, \ldots, x^n, T, S) \\ &= p(x^n, \mathcal{E}) \prod_{i=1}^{n} p(x^{i-1}, x^i) p_e^0(T[t^{i-1}+1 : t^i], S[L^{i-1}+1 : L^i] | x^i), \end{aligned} \tag{9}$$

where n and $x^1, \ldots, x^n$ are such that

$$L^n = l(x^1) + \cdots + l(x^n) = L,$$

with L being the vector of lengths of the sequences. Note, that if we sum this expression over the possible letters of the ancestral sequence T the term p_e^0 is replaced by the term

$$p_e(S[L^{i-1}+1 : L^i] | x^i).$$

To obtain the marginal probability of a part of the alignment we introduce the function $F(K|x^0)$, where K is a column vector of integers and x^0 is any state among (5), given by

$$\begin{aligned} &F(K|x^0) \\ &= \sum_{n=0}^{\infty} \sum_{x^1, \ldots, x^n \in \Xi : K + L^n = L} p(x^n, \mathcal{E}) \prod_{i=1}^{n} p(x^{i-1}, x^i) p_e(S[K + L^{i-1}+1 : K + L^i] | x^i), \end{aligned} \tag{10}$$

where the inner sum is zero if there is no $x^1, \ldots, x^n$ with $K + L^n = L$. In particular, $F(K|x) = 0$ if there exists j with $K_j > L_j$. This function gives the marginal probability of the sequences $S[K+1 : L]$ given that the initial state is x^0. In particular the marginal probability of the sequences at the three leaves is

$$P(S) = F(0|I). \tag{11}$$

From (9) and (10) we find the following marginal probability

$$\begin{aligned} &P(x^1, \ldots, x^k, T[1 : t^k], S) \\ &= \left\{ \prod_{i=1}^{k} p(x^{i-1}, x^i) p_e^0(T[t^{i-1}+1 : t^i], S[L^{i-1}+1 : L^i] | x^i) \right\} F(L^k | x^k), \end{aligned}$$

and using (11) we get

$$\begin{aligned} & P(x^1, \ldots, x^k, T[1:t^k]|S) \\ = \; & \left\{ \prod_{i=1}^{k} p(x^{i-1}, x^i) p_e^0(T[t^{i-1}+1:t^i], S[L^{i-1}+1:L^i]|x^i) \right\} \frac{F(L^k|x^k)}{F(0|I)}. \end{aligned} \tag{12}$$

Dividing (12) by the same expression with k replaced by $k-1$ we obtain

$$\begin{aligned} & P(x^k, T[t^{k-1}+1:t^k]|S, x^1, \ldots, x^{k-1}, T[1:t^{k-1}]) \\ = \; & p(x^{k-1}, x^k) p_e^0(T[t^{k-1}+1:t^k], S[L^{k-1}+1:L^k]|x^k) \frac{F(L^k|x^k)}{F(L^{k-1}|x^{k-1})}. \end{aligned} \tag{13}$$

From (13) we can sequentially simulate $(x^1, T[1:t^1]), (x^2, T[t^1+1:t^2]), \ldots$ if $F(K|x)$ is known for any x and any $K \leq L$.

In order to calculate $F(K|x)$ we make a recursion from (10). We do this by separating the sum into the sum over x^1 and the sum over the remaining variables. For $K_j \leq L_j$, $j = 1, 2, 3$, and $K_j < L_j$ for at least one j we get

$$F(K|x) = \sum_{x \in \Xi} p(x, z) p_e(S[K+1:K+l(z)]|x) F(K+l(z)|z), \tag{14}$$

and for $K = L$ we find

$$F(L|x) = p(x, \mathcal{E}) + p(x, D) F(L|D), \quad D = \begin{pmatrix} \# \\ - \\ - \\ - \end{pmatrix}. \tag{15}$$

The recursion (14) is solved in the following way. If $F(\tilde{K}|x)$ has been found for all x and all $\tilde{K}$ with $\tilde{K} \geq K$ and $\tilde{K}_j > K_j$ for at least one j, we first find $F(K|D)$ from

$$F(K|D)(1 - p(D, D)) = \sum_{z \in \Xi, z \neq D} p(D, z) p_e(S[K+1:K+l(z)]|x) F(K+l(z)|z),$$

and next find $F(K|x)$, $x \neq D$, from (14). The start of the recursion is given by

$$\begin{aligned} F(L|D) &= \frac{p(D, \mathcal{E})}{1 - p(D, D)}, \\ F(L|x) &= p(x, \mathcal{E}) + p(x, D) F(L|D), \quad x \neq D. \end{aligned}$$

Note that when we have simulated the alignment $x^1, \ldots, x^N$ for the 3-star with interior node r and leaves $a(r)$, $d_1(r)$, and $d_2(r)$ we can immediately read off the alignments $A(r, a(r))$, $A(r, d_1(r))$, and $A(r, d_2(r))$.

3 Complexity and mixing

3.1 Mixing

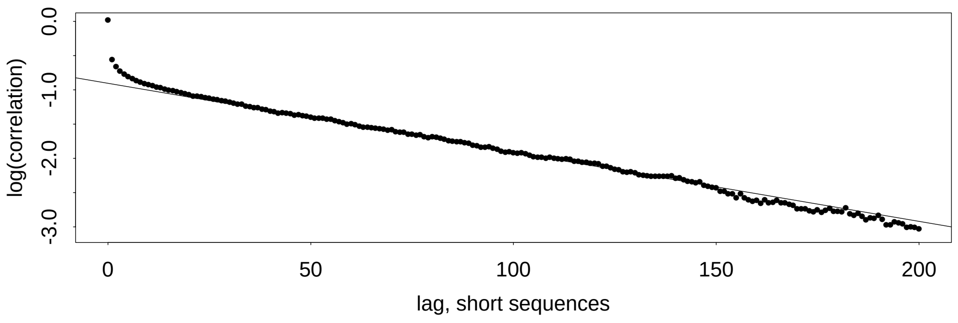
We first consider the problem of simulating the ancestor and the three alignments of a 3-star tree. Details of the simulation experiments are given in Appendix B. For a 3-star tree our algorithm is designed to simulate directly form the conditional distribution given the three observed sequences. When simulating according to the method of Holmes and Bruno (2001) (see Appendix A below for a description) we find that the mixing is not fast. In Figure 3 is a plot of the first 200 autocorrelations on a logarithmic scale based on 100000 simulated values of the number of deletions that are not followed by an insertion in branch 2 (one value correspond to one round of updating the three alignments and updating the ancestral sequence). The upper part of the plot is for three sequences of lengths around 75 and the lower plot is for three sequences of lengths around 150. We see that apart from an initial phase there seems to be an exponential decrease of the autocorrelations. We have estimated the slope using the correlations for lags 50 to 150. Based on the first 50 values and the exponential decrease we have also estimated the sum of the autocorrelations, $\sum_k r_k$. When calculating the variance of the average $\bar{x} = \sum_{i=1}^{n} x_i$ based on n simulated values we have that $n\text{Var}(\bar{x}) \approx 1 + \sum_k r_k$ in the limit $n \to \infty$. If for example $1 + \sum_k r_k = 100$ the interpretation is that we need to simulate $100n$ observations in order to have the same precision as compared with the situation of n independent values when n is large.

In Table 1 we have given the result for the algorithm of Holmes and Bruno (2001) for a 3-star tree as well as for the tree in Figure 1 with 4 observed sequences and for the tree in Figure 2 with 7 observed sequences.

For the algorithm reported in this paper there is by construction no correlation for the 3-star tree and in the other cases there is very little correlation. For lag 1 the correlation is of order 0.06 and for higher lags the correlations seems to be almost zero.

3.2 Complexity

Without any refinements our algorithm has complexity L^3, where L is a typical length of a sequence, due to the calculation of $F(K|x)$ via the recursion (14). This can be reduced



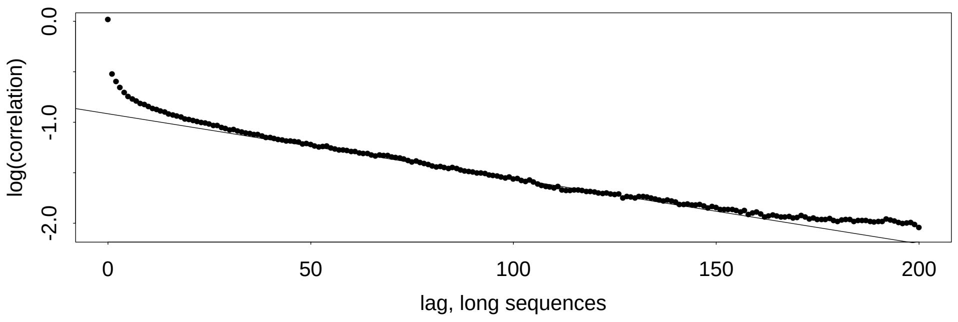


Figure 3: Log of autocorrelations for the number of deletions along branch 2.

Length	3-star, 75	3-star, 150	4-seq, 75	4-seq, 150	7-seq, 75
Slope on log scale	-0.0101	-0.0064	-0.0061	-0.0061	-0.0055
$1 + 2\sum_k r_k$	82	127	130	140(112)	176

Table 1: Correlations for the Holmes and Bruno (2001) algorithm.

since $F(K|x)$ will be practically zero outside a band around a 'typical alignment'. For the runs reported in this paper we have simply taken a fixed band around a line in the three dimensional space. Similarly, the algorithm of Holmes and Bruno (2001) has complexity L^2 and this can be reduced by using a band only. For the runs with sequences of length approximately 75 we have used a band of width 20 and for the runs with sequences of length approximately 150 we have used a band of width 30. A rough calculation gives the following complexity measures

$$\begin{aligned} \text{Holmes and Bruno} \quad &: \quad k_2 \times 3 \times L \times w + k_3 \times L \\ \text{Ours} \quad &: \quad k_3 \times 15 \times L \times w \times w \end{aligned}$$

where w is the width of the band used, k_2 is the number of branches, and k_3 is the number of interior nodes. The numbers 3 and 15 are the number of states for aligning two sequences and for aligning a 3-star tree, respectively. These complexity measures seem to be in good agreement with the actual CPU times reported in Table 2.

	3-star,75	3-star,150	4-seq,75	4-seq,150	7-seq,75
	$w = 20$	$w = 30$	$w = 20$	$w = 30$	$w = 20$
H & B	4.99	15.04	9.94	27.77	18.77
Ours	158.5	775.4	330.3	1524.0	830.2
Ratio	32	52	33	55	44
$(1 + 2\sum_k r_k)$/Ratio	2.6	2.5	3.9	2.6	4.0

Table 2: CPU running times for 100 rounds of updating of the alignment. The bottom row gives the efficiency of the algorithm in this paper as compared to the algorithm in Holmes and Bruno (2001).

From the bottom row of Table 2 we see that in all the runs our algorithm is more efficient for estimating mean values. Furthermore, if we can design a version of the algorithm that uses a fixed band irrespectively of the lengths of the sequences then the efficiency of our algorithm as compared to that of Holmes and Bruno (2001) will increase with the lengths of the sequences.

4 Maximum likelihood estimation

4.1 Full likelihood for sequences and the alignments

For two sequences S_1 and S_2 with an alignment $A = \{z^1, \ldots, z^n\}$, where z^i is one of the states from (2), the probability of the sequence S_2 and the alignment A given the sequence S_1 is

$$\begin{aligned} &P(S_2, A|S_1) \\ &= \tilde{p}(z^n, \mathcal{E}; \tau) \prod_{i=1}^{n} \tilde{p}(z^{i-1}, z^i; \tau) \tilde{p}_e^c(S_2(L_2^{i-1}+1 : L_2^i)|S_1(L_1^{i-1}+1 : L_1^i), z^i; \tau). \quad (16) \end{aligned}$$

Here L_1 and L_2 are the lengths of the sequences, $\tilde{p}(\cdot,\cdot;\tau)$ is the transition probability from (4), and $\tilde{p}_e^c$ is a conditional emission probability

$$\tilde{p}_e^c(w_2|w_1,z;\tau) = \begin{cases} f(w_2|w_1;\tau) & z = \binom{\#}{\#} \\ 1 & z = \binom{\#}{-} \\ \pi(w_2) & z = \binom{-}{\#}, \end{cases} \tag{17}$$

We can next state the full likelihood $L_f(\theta)$ for the multiple alignment on the tree, that is, the joint probability of $S_1,\ldots,S_\eta$, $T_{\eta+1},\ldots,T_{\eta+\nu}$, and all the pairwise alignments $A(a(j),j)$, as a function of the parameters θ defining the model. Using the notation $S_{\eta+j} = T_{\eta+j}$ we find

$$L_f(\theta) = P(T_{\eta+\nu}) \prod_{j=1}^{\eta+\nu-1} P(S_j, A(a(j),j)|T_{a(j)};\tau_j), \tag{18}$$

where $P(T_{\eta+1})$ is calculated as in (1).

The marginal likelihood $L_m(\theta)$ based on the observed sequences S_r, $r = 1,\ldots,\eta$, is obtained by summing $L_f(\theta)$ over the ancestral sequences and their alignments. The ratio $L_m(\theta_2)/L_m(\theta_1)$ can be calculated as a mean value

$$\frac{L_m(\theta_2)}{L_m(\theta_1)} = E_{\theta_1}\left(\frac{L_f(\theta_2)}{L_f(\theta_1)}|S_r, r=1,\ldots,\eta\right). \tag{19}$$

Thus we can use the Gibbs sampler from Section 3 to generate samples from the conditional distribution given S_r, $r = 1,\ldots,\eta$, and thereby approximate (19). However, unless θ_2 is close to θ_1 the ratio $L_f(\theta_2)/L_f(\theta_1)$ will have a very large variance growing exponentially in the length of the sequences. Instead we use the EM-algorithm described next.

4.2 Simulated EM-algorithm

To reduce the number of parameters we first estimate the stationary probabilities π from the empirical frequencies in the observed sequences $S_1,\ldots,S_\eta$. Also we estimate $\gamma = \lambda/\mu$ from the average length of the observed sequences

$$\begin{aligned} \hat{\pi}(a) &= \sum_{j=1}^{\eta}\sum_{i=1}^{L_j} 1(S_j[i]=a) / \sum_{j=1}^{\eta} L_j, \\ \hat{\gamma} &= \frac{\bar{L}}{1+\bar{L}}, \quad \bar{L} = \frac{1}{\eta}\sum_{i=1}^{\eta} L_i. \end{aligned} \tag{20}$$

When π and γ has been fixed the full likelihood (18) can, apart form a data dependent term, be written as

$$\begin{aligned} L_f(\theta) &= \prod_{j=1, j\neq \eta+1}^{\eta+\nu} \Big\{ b(\#,\#;j)^{N(\#,\#;j)} b(\#,-;j)^{N(\#,-;j)} \\ &\quad \times b(-,\#;j)^{N(-,\#;j)} b(-,-;j)^{N(-,-;j)} s(\#;j)^{N(\#;j)} s(-;j)^{N(-;j)} \\ &\quad \times \prod_{w_1,w_2} f(w_2|w_1;j)^{K(w_1,w_2;j)} \Big\}, \end{aligned} \tag{21}$$

where

$$\theta = (\mu, \psi, \tau_j : j = 1, \dots, \eta+\nu-1)$$

with ψ the parameters in the substitution matrix. Here $N(\#,\#;j)$ counts the number of times we have the term $b(\#,\#;j)$ in the transition probabilities (see (4) in the alignment $A(a(j), j)$. All the other counts $N(\cdot)$ are defined similarly, and $K(w_1, w_2; j)$ is the number of substitutions of w_1 by w_2 along the branch j. To use the EM-algorithm we must simulate the mean values of all the count statistics in the conditional distribution given the observed sequences and under the parameter value θ_1, say. A new value θ_2 is then found by maximising (21) with the counts replaced by their mean values.

We have used an iterative procedure to maximize (21). We first find, with $\phi = (\mu, \psi)$,

$$\hat{\tau}(\phi), \quad j = 1, \dots, \eta+\nu-1,$$

for a fixed value of ϕ, and next find a new value of ϕ by maximising

$$L_f(\tilde{\phi}, \hat{\tau}(\phi) : j = 1, \dots, \eta+\nu-1)$$

with respect to $\tilde{\phi}$. The reason for this procedure is that finding $\hat{\tau}(\phi)$ is a one dimensional search problems since L_f factorizes for a fixed value of ϕ.

We have tried the above method on a 3-star tree. For a 3-star tree we can compare with the estimates obtained by maximising the likelihood function directly. We have considered a situation with ψ fixed and thus we maximize with respect to $(\mu, \tau_1, \tau_2, \tau_3)$. In Table 3 is the result from 30 steps of the simulated EM-algorithm, where in each step we simulate the ancestral and its alignments 1000 times. Included is also the average $\bar{l}_f$ of the full log likelihood function, where $l_f = \log(L_f)$ with L_f given in (18), as well as the marginal log likelihood l_m based on the three observed sequences. From Table 3 we see that the marginal likelihood is very flat in a large region of the parameter space.

	μ	τ_1	τ_2	τ_3	$\bar{l}_f$	l_m
Start	0.100	0.80	0.80	0.80	19.41	-1.95
Iteration 5	0.085	1.15	0.87	1.21	-2.65	-0.43
Iteration 10	0.079	1.27	0.83	1.39	-6.58	-0.25
Iteration 15	0.078	1.32	0.76	1.48	-4.44	-0.17
Iteration 20	0.077	1.35	0.68	1.53	-1.82	-0.12
Iteration 25	0.077	1.38	0.63	1.57	-0.87	-0.09
Iteration 30	0.078	1.39	0.61	1.59	0.00	-0.08
MLE	0.106	1.55	0.28	1.62		0.00

Table 3: EM-algorithm for a 3-star tree.

For the tree in Figure 1 we have made 30 steps in the simulated EM-algorithm. The results can be seen in Table 4

	μ	τ_1	τ_2	τ_3	τ_4	τ_5	$\bar{l}_f$
Start	0.100	0.80	0.80	0.80	0.80	0.80	-1.68
Iteration 5	0.101	0.94	0.79	0.71	0.72	0.75	4.83
Iteration 10	0.102	1.01	0.76	0.68	0.69	0.71	7.40
Iteration 15	0.107	1.09	0.72	0.70	0.67	0.67	3.58
Iteration 20	0.110	1.11	0.66	0.69	0.66	0.67	3.14
Iteration 25	0.113	1.14	0.64	0.70	0.66	0.67	1.34
Iteration 30	0.115	1.15	0.62	0.69	0.65	0.66	0.00

Table 4: EM-algorithm for the tree in Figure 1.

5 Discussion

We have shown that it is feasible to simulate multiple alignments within the TKF-model using a Gibbs sampler where in each step a 3-star tree is updated. The Gibbs sampler seems to be mixing very rapidly. Contrary to this the algorithm suggested in Holmes and Bruno (2001) seems to have long mixing times. For the runs reported here the algorithm of Holmes and Bruno (2001) is less efficient than the algorithm suggested in this paper.

Via the EM-algorithm we can obtain maximum likelihood estimates of the parameters of the model.

The Gibbs sampler suggested in this paper is not restricted to the exact form of the TKF-model. A more general hidden Markov model simply implies a different state space and different transition probabilities for the 3-star tree in Section 2. In particular one may wish to include the possibility of going to the immortal state instead of the *end* state. This will introduce an extra parameter in the model so that $\gamma = \lambda/\mu$ is no longer determined by the lengths of the sequences.

The algorithm produces samples from the conditional distribution of alignments and ancestral sequences given the observed sequences. One can therefore estimate the probabilities of different evolutionary events. The algorithm does not point to one particular alignment although in a long run of the algorithm one can of course choose the alignment with the highest value of the full likelihood function.

Appendix A: Holmes and Bruno algorithm

In Holmes and Bruno (2001) two different updating steps are used. The first one is the ordinary simulation of an alignment of two sequences S_1 and S_2. This means that a set of states $x^1, \ldots, x^N$ of the form (2) is simulated. Here the kth is simulated from the distribution

$$\begin{aligned} & P(x^k|S, x^1, \ldots, x^{k-1}) \\ = \;& \tilde{p}(x^{k-1}, x^k)\tilde{p}_e(S[L^{k-1}+1 : L^k]|x^k)\frac{\tilde{F}(L^k|x^k)}{\tilde{F}(L^{k-1}|x^{k-1})}, \end{aligned}$$

where $\tilde{p}(\cdot,\cdot)$ is the transition probability from (4), $\tilde{p}_e$ is the emission probability

$$\tilde{p}_e\left(\begin{pmatrix} w_1 \\ w_2 \end{pmatrix}|x\right) = \begin{cases} \pi(w_1)f(w_2|w_1;\tau) & x = \binom{\#}{\#} \\ \pi(w_1) & x = \binom{\#}{-} \\ \pi(w_2) & x = \binom{-}{\#}, \end{cases}$$

and $\tilde{F}$ satisfies for $k \neq L$ the recursion

$$\begin{aligned} \tilde{F}(K|x) \; = \; & \sum_{z, K+l(z)\le L} \tilde{F}(K+l(z)|z) \\ & \times \tilde{p}(x,z)\tilde{p}_e(S_2[K_2+1 : K_2+l_2(z)]|S_1[K_1+1 : K_1+l_1(z)], z), \end{aligned}$$

where z is one of the states in (2) and $\tilde{p}_e(w_2|w_1, z)$ is given as in (17) with the first two lines multiplied by $\pi(w_1)$. The recursion is started by $\tilde{F}(L|x) = \tilde{p}(x, \mathcal{E})$.

The second updating step in Holmes and Bruno (2001) can be explained as follows. We consider a 3-star tree where we have an alignment along each of the three branches. These three alignments are translated into an alignment $\bar{x}^1, \ldots, \bar{x}^{\bar{N}}$ with states $\bar{x}^i$ of the form (5). We first remove all those $\bar{x}^i$ that equals D from (15). This gives us the reduced set of states $x^1, \ldots, x^N$. Then we insert new states equal to D in between x_{i-1} and x^i, the length m_i of the inserted states has distribution

$$P(m_i = k) = \begin{cases} \frac{p(x^{i-1},D)p(D,x^i)}{p(x^{i-1},x^i)+p(x^{i-1},D)p(D,x^i)} & k = 0 \\ \frac{p(x^{i-1},x^i)(1-p(D,D))}{p(x^{i-1},x^i)+p(x^{i-1},D)p(D,x^i)} p(D,D)^{k-1} & k > 0. \end{cases}$$

Finally we update, for $i = 1, \ldots, N$, the ancestral letter at the interior node corresponding to the state x^i. This means that if $x_0^i = -$ there is no letter to update, and if $x_0^i = \#$ we choose a letter according to the distribution

$$p(w_0) \propto \pi(w_0) \prod_{\{j: x_j^i = \#\}} f(S_j[L_j^{i-1} + 1]|w_o; \tau_j).$$

Appendix B: Details of simulation experiments

In the simulations we model sequences of nucleotides so that the possible letters are A, G, C, T. The substitution process is described by the rate matrix

$$q(w_1, w_2) = \pi(w_2)\psi^{1tv(w_1,w_2)}/c,$$

where 1_{tv} is one if substituting w_2 for w_1 is a transversion and 1_{tv} is zero otherwise. The scaling constant c was taken to be $\pi(G) + \psi(\pi(C) + \pi(T))$ so that there is a unit rate for a change of the letter A.

We considered the tree in Figure 1 with 4 observed sequences. We generated sequences from the TKF-model using the parameter settings

$$\mu = 0.1, \quad \lambda = 0.099, \quad \psi = 0.2, \quad \pi = (0.2, 0.3, 0.2, 0.3),$$
$$\tau_1 = \tau_2 = \tau_3 = \tau_4 = \tau_5 = 0.8.$$

For the investigation of the mixing properties we generates sequences of lengths $(75, 74, 79, 79)$ and $(156, 147, 145, 146)$, respectively. When simulating the ancestral sequences and their alignments we took γ and π as given in (20) and used the true values of μ, ψ, and took the branch lengths to be $\tau = 0.8$. As initial value for the ancestral sequences we simply took $T_5 = S_1$ and $T_6 = S_4$. When simulating from the tree in Figure 2 similar choices were made.

References

Durbin, R., Eddy, S., Krogh, A. and Mitchison, G. (1998). *Biological Sequence Analysis: Probabilistic Models of Protein and Nucleic Acids.* Cambridge University Press, Cambridge, UK:

Hein, J., Jensen, J.L. and Pedersen, C.N.S. (2002). Recursions for statistical multiple alignment. Research Report No. 425. Department of Theoretical Statistics, University of Aarhus.

Holmes, I. and Bruno, W.J. (2001). Evolutionary HMMs: A Bayesian Approach to Multiple Alignment. *Bioinformatics* **17**, 803-820.

Thorne, J.L., Kishino, H. and Felsenstein, J. (1991). An evolutionary model for maximum likelihood alignment of DNA sequences. *J. Mol. Evol.* **33**, 114-124.