A “Long Indel” Model For Evolutionary Sequence Alignment

I. Miklós, G. A. Lunter, and I. Holmes

Department of Statistics, University of Oxford, Oxford, U.K.

We present a new probabilistic model of sequence evolution, allowing indels of arbitrary length, and give sequence alignment algorithms for our model. Previously implemented evolutionary models have allowed (at most) single-residue indels or have introduced artifacts such as the existence of indivisible “fragments.” We compare our algorithm to these previous methods by applying it to the structural homology dataset HOMSTRAD, evaluating the accuracy of (1) alignments and (2) evolutionary time estimates. With our method, it is possible (for the first time) to integrate probabilistic sequence alignment, with reliability indicators and arbitrary gap penalties, in the same framework as phylogenetic reconstruction. Our alignment algorithm requires that we evaluate the likelihood of any specific path of mutation events in a continuous-time Markov model, with the event times integrated out. To this effect, we introduce a “trajectory likelihood” algorithm (Appendix A). We anticipate that this algorithm will be useful in more general contexts, such as Markov Chain Monte Carlo simulations.

Introduction

Stochastic models of amino acid and nucleotide substitution are fundamental to many applications in bioinformatics, including database searching, pairwise sequence alignment, evolutionary tree construction, domain profiling, and (most recently) “phylogenetic footprinting” (Durbin et al. 1998). Yet, despite extensive progress in researching such point substitution models, the evolutionary treatment of insertion and deletion events (indels) has lagged behind (Thorne et al. 1991; Hein et al. 2000; Hein 2001; Holmes and Bruno 2001; Miklós & Toroczkai 2001; Metzler et al. 2001; Lunter et al. 2003; Metzler 2003). A measure of this lag is that all indel models so far analyzed assume either that each deletion event affects only a single residue or that the sequence is comprised of independently evolving fragments. Both these assumptions are clearly unrealistic and may cause systematic bias (Hein et al. 2000; Holmes and Bruno 2001).

To date, the canonical model for biological sequence evolution with indels has been the TKF91 model (Thorne et al. 1991). This model describes the evolution of a finite sequence and allows only single-residue indel events. TKF91 alignment therefore uses a *global* scoring scheme with a *linear* penalty function for gap sizes. The TKF91 model has been extensively analyzed (Hein et al. 2000) and developed into a multiple alignment algorithm, both in full likelihood (Hein 2001; Lunter et al. 2003) and Markov Chain Monte Carlo (MCMC) settings (Holmes and Bruno 2001).

In contrast to TKF91 alignment, many computational biologists use a *local* scheme with *affine* gap penalties. In evolutionary terms, affine gap costs correspond roughly to a geometric length distribution for each single indel event. Previously, the closest evolutionary equivalent to this has been the TKF92 model. In this model, the sequence is assumed to consist of finite-length indivisible fragments, and the indel process acts on fragments rather than residues. This introduces hidden information in the form of fragment boundaries whose locations must be inferred. The realism of these invisible boundaries is questionable, and they may potentially bias multiple alignment (Thorne et al. 1992).

When indels affect single residues only, the fate of each residue in a given sequence is independent: we can chop a pairwise alignment into independently evolving zones by making a cut before every ancestral residue (Thorne et al. 1991). Modeling alignments with long insertion events but single-residue deletions is tractable (if mathematically complex) because each ancestral residue still corresponds to an independent zone (Miklós and Toroczkai 2001). However, finding exact probabilities for alignments with long deletion events is difficult: for any two ancestral residues, there is a finite probability that they will be deleted in a single event, so the alignment cannot be split into independent zones by cutting after each ancestral residue.

Later in this paper (see *Models*), we present a new *long indel model* as an alternative to the TKF91 model, introducing a new notation for evolutionary models that we call the *rate grammar*. We then show (see *Algorithm*) that, in an idealized model where the evolving sequence is assumed to be part of an infinitely long sequence, the pairwise alignment can still be split into independent zones; not after every ancestral residue, but after every *surviving* ancestral residue; that is, every column in the alignment that does not contain a gap. Because no deletion event can cross such a boundary, the zones are independent. The probability of observing a zone can be estimated, not analytically (as with the TKF91 and TKF92 models), but by directly summing the probability of short mutation trajectories such as that shown in Figure 1. We have implemented our model and compared its performance to the TKF91, TKF92, and Gotoh algorithms (Gotoh 1982), using for a benchmark the structurally informed alignments from the HOMSTRAD database (Mizuguchi et al. 1998), as described in the section titled *Evaluation*. Our results, given in *Results*, show that the long indel model outperforms TKF91 and TKF92 both at alignment and evolutionary distance estimation, and its performance is comparable with that of Gotoh, while providing much more information about sequence comparison (confidence levels). In the *Discussion* we present applications of our theory.

Key words: Stochastic modeling of molecular evolution, Structural alignment, Maximum Likelihood evolutionary time estimation.

E-mail: miklos@stats.ox.ac.uk.

Mol. Biol. Evol. 21(3):529–540. 2004
DOI: 10.1093/molbev/msh043
Advance Access publication December 5, 2003
Molecular Biology and Evolution vol. 21 no. 3

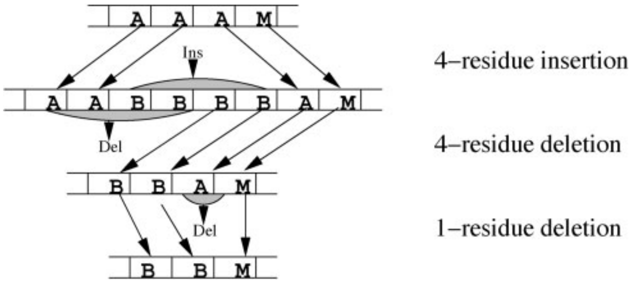

FIG. 1.—An example three-event trajectory for a zone that changes length from four resides to three (outcome $B^i_{3.2}$; see section titled *Algorithm*). By definition, the final ancestral residue in the zone (the M) cannot be deleted, whereas every other ancestral residue (the A's) *must* be deleted.

As mentioned above, our alignment algorithm directly enumerates finite-length "trajectories," such as the one shown in figure 1. A trajectory is here defined to be the set of all mutation histories with a given sequence of mutation events; that is, the actual timing of each individual mutation event is integrated out of the trajectory likelihood. As models of molecular sequence evolution become increasingly complex, and the chances of obtaining closed-form solutions to these models become correspondingly slim, we anticipate that such trajectory-counting approaches as we have used (based either on direct enumeration, MCMC sampling, or some other method) will become increasingly necessary. Accordingly, in Appendix A, we present a complete algorithm for computing trajectory likelihoods.

Models

Let Ω be a finite alphabet, let Ω^L be the set of all sequences over Ω of length L, and let $\Omega^* = \cup_{L=0}^{\infty}\Omega^L$ be the set of sequences of any length. For sequences S, $T \in \Omega^*$, let $|S|$ denote the sequence length, ST the concatenation of S and T, and S_n the nth symbol, for $1 \leq n \leq |S|$.

SID Models

The evolutionary models we consider are continuous-time Markov processes whose state space Φ is the set of all sequences, $\Phi = \Omega^*$. We consider in particular a class of models called substitution/insertion/deletion (SID) models. These models allow local mutations only, including point substitutions and multi-residue indels. The rates of substitution, insertion, and deletion events are given, respectively, by $\rho_S(S^L,S^X,S^Y,S^R)$, $\rho_I(S^L,S^I,S^R)$, and $\rho_D(S^L,S^D,S^R)$, where S^X, S^Y are the incoming and outgoing substitution sequences (each of length 1), S^I,S^D are the inserted or deleted sequences, and S^L,S^R are the sequences flanking the mutation.

Thus, the instantaneous rate $R(S,S')$ with which sequence S mutates to sequence S' is given by

$$\begin{aligned} R(S,S') = & \sum_{\substack{S^L,S^R,|S^X|=1,\\ |S^Y|=1,S^X\neq S^Y}} \rho s(S^L,S^X,S^Y,S^R) \\ & \times \delta(S = S^LS^XS^R)\delta(S' = S^LS^YS^R) \\ & + \sum_{s^L,S^R,|S^I|>0} \rho_I(S^L,S^I,S^R) \\ & \times \delta(S = S^LS^R)\delta(S' = S^LS^IS^R) \\ & + \sum_{S^L,S^R,|S^D|>0} \rho_D(S^L,S^D,S^R) \\ & \times \delta(S = S^LS^DS^R)\delta(S' = S^LS^R), \end{aligned} \tag{1}$$

for $S \neq S'$. Here $\delta(S = S')$ is 1 if $S = S'$ and 0 if $S \neq S$. Note that we sum over all applicable mutations; thus, the total rate of mutating sequence CAAG to sequence CAG is R(CAAG,CAG) = ρ_D(C,A,AG) + ρ_D(CA,A,G), because there are two indistinguishable A's that can be deleted.

A more readable notation for this model is

$$S^LS^XS^R \xrightarrow{\rho_S(S^L,S^X,S^Y,S^R)} S^LS^YS^R \qquad |S^X| = 1,\ |S^Y| = 1,\ S^X \neq S^Y; \tag{2}$$

$$S^LS^R \xrightarrow{\rho_I(S^L,S^I,S^R)} S^LS^IS^R \qquad |S^I| > 0; \tag{3}$$

$$S^LS^DS^R \xrightarrow{\rho_D(S^L,S^D,S^R)} S^LS^R \qquad |S^D| > 0. \tag{4}$$

We call this notation a *rate grammar*, because it is similar to a stochastic grammar (Durbin et al. 1998); indeed, the only differences are (1) that it has continuous-time evolutionary rates associated with every rule, rather than discrete-time transformation probabilities; and (2) in contrast to a stochastic grammar, there are no "terminal" symbols; rather, every residue is a "nonterminal," because evolution is a nonterminating process. Using rate grammars, it is possible to formally describe a wide variety of evolutionary processes (including duplication, inversion, and translocation) as well as quite general MCMC algorithms for sampling trajectories through the state space of such models.

Returning to straightforward SID models, if ρ_S, ρ_I, and ρ_D are independent of S^L and S^R (i.e., $\rho_S \equiv \rho_S(S^X,S^Y)$, $\rho_I \equiv \rho_I(S^I)$, and $\rho_D \equiv \rho_D(S^D)$) then we say that the SID model is *context-independent*. Note, however, that a context-independent rate grammar does not have quite the same meaning as a context-free grammar in the Chomsky sense, because a context-free grammar would not allow multi-residue deletions, but a context-independent rate grammar allows such deletions as long as they occur at a rate independent of the flanking sequence.

TKF91

The TKF91 model (Thorne et al. 1991) is a reversible context-independent SID model (see above) with the following indel rates:

$$\rho_I(S) = \lambda q(S_1) \qquad \text{if } |S| = 1; 0 \text{ otherwise.} \tag{5}$$

$$\rho_D(S) = \mu \qquad \text{if } |S| = 1; 0 \text{ otherwise.} \tag{6}$$

Here $q(x)$ is the equilibrium residue distribution, while λ and μ are constant insertion and deletion rates, with $\lambda < \mu$.

Let $\upsilon(n)$ be the equilibrium length distribution. For reversibility, detailed balance requires that $\lambda\upsilon(n) = \mu\upsilon(n+1)$ and so $\upsilon(n) = \gamma^n(1-\gamma)$ where $\gamma = \lambda/\mu$. That is, the equilibrium length distribution is geometric with parameter γ and mean $\gamma/(1-\gamma)$. The full sequence equilibrium distribution takes the form $\pi(S) = \upsilon(|S|)\prod_{k=1}^{|S|} q(S_k)$.

Because the TKF91 model does not allow multiple residues to be deleted instantaneously, the fates of any two ancestral residues are independent. This permits an exact solution of the transition probabilities by dynamic programming (Thorne et al. 1991).

TKF92

The TKF92 model is a variation of the TKF91 model (Thorne et al. 1992). In TKF92, the sequence consists of fixed-length indivisible fragments, each fragment containing a geometrically distributed number of residues. This is equivalent to replacing the finite alphabet of residues, Ω, with an infinite alphabet of sequence fragments, Ω^*, so that the state space of the model is now $\Phi = (\Omega^*)^*$—i.e., the set of sequences of sequences. The fragment substitution matrix ρ_S is set up to allow only point substitutions within each fragment.

A TKF92 evolutionary state is not a sequence of residues, but rather a sequence of sequences; the observed, "biological" sequence is recovered by concatenating all the fragments, using the map $f(S) = S_1S_2 \cdots S_{|S|}$. Fragments are not observed, and alignment probabilities, as well as the maximally contributing alignment, are obtained by summing out all possible fragmentations of the observed sequences. (We use the standard Viterbi algorithm for finding the maximally contributing alignment, and the summation over fragmentations is done implicitly by a careful design of the TKF92 HMM.) The advantage of TKF92 is that it allows the simultaneous deletion of multiple residues (in the same fragment) while retaining transition probabilities that can be calculated exactly, as per TKF91 (Thorne et al. 1992). However, this comes at the expense of introducing hidden information that may bias alignment.

Long Indel Model

We return now to models with state space $\Phi = \Omega^*$, where the evolutionary state is a sequence (rather than a sequence of sequences, as in TKF92).

The *long indel* model is a time-reversible, context-independent SID model, as described earlier. It generalizes TKF91 by allowing instantaneous deletion of arbitrarily long subsequences, without requiring that these subsequences form an indivisible fragment.

The long indel model has the following rates:

$$\rho_I(S) = \lambda_{|S|}\prod_{k=1}^{|S|} q(S_k), \tag{7}$$

$$\rho_D(S) = \mu_{|S|}, \tag{8}$$

where λ_k,μ_k are the rates of k-residue insertions and deletions, and q and ρ_S are as defined as in the earlier description of TKF91.

Again, let $\upsilon(n)$ be the equilibrium length distribution. Time reversibility implies detailed balance, which requires that $\lambda_k\upsilon(n) = \mu_k\upsilon(n+k)$ for all n and k, implying both that $\upsilon(n) = \gamma^n(1-\gamma)$ and that $\lambda_k/\mu_k = \gamma^k$, where $\gamma = \lambda_1/\mu_1$. That is, not only is the equilibrium length distribution geometric with parameter γ, but so is the ratio λ_k/μ_k as a function of k. Again, $\pi(S) = \upsilon(|S|)\prod_{k=1}^{|S|} q(S_k)$.

Note that we have full freedom in choosing the deletion rate function (which then fixes the insertion rates). For simplicity, however, we choose the rate for k-residue deletions to be a geometric function of k with parameter r, so $\mu_k = \mu(1-r)^2r^{k-1}$ and $\lambda_k = \gamma\mu(1-r)^2(\gamma r)^{k-1}$. The reason for the normalization factor $(1-r)^2$ is that it makes the total deletion rate per site come out as $\sum_{k=1}^{\infty} k\mu_k = \mu$, so that μ has the same meaning as in the TKF91 and TKF92 models, where $\mu = \sum_{k=1}^{\infty} k\mu_k$ is the total deletion rate per site. It can be seen that TKF91 emerges as a special case of the geometric long indel model when $r = 0$.

In sequence alignment terms, with this choice of indel rates our model is the evolutionary analog of Gotoh's affine gap algorithm (Gotoh 1982). Note, however, that, although the size of indel events is geometrically distributed, the size of gaps in an alignment will only be geometrically distributed when the two sequences are sufficiently close so that there is only one expected indel event per observed gap.

In the above treatment, we have adopted a reversible model purely for technical convenience. Compared to irreversible models, reversible models have roughly half as many parameters; furthermore, they permit the use of an unrooted phylogenetic tree, which for us means that we can treat one sequence as the ancestor and the other as descendant with no influence on the outcomes (Durbin et al. 1998).

Although reversibility is a common assumption in molecular evolutionary analyses, we believe that it must be treated with skepticism. Although there is anecdotal evidence that nucleotide substitution processes are often close to reversible (Bruno and Arvestad 1997), there is no reason why this should, in general, be the case. There are many ways in which a realistic indel model could violate detailed balance: for example, insertion events might typically be small and frequent, and deletion events might be rare and large (Holmes and Durbin 1998). Similarly plausible irreversibilities can be conjectured for substitution models, particularly if the substitution rate depends on the sequence context, as, e.g., in C_PG depression.

Infinite Sequence

Although the long indel model has some convenient theoretical properties, it also has some that are decidedly inconvenient. For example, consider all deletion events on sequence S that start at residue n; in other words, deletions of the form $S^LS^DS^R \rightarrow S^LS^R$ where $|S^L| = n-1$. Call such an event a *rightward deletion* of residue n. The total rate of all rightward deletions of residue n is $\sum_{k=1}^{|S|+1-n} \mu_k$. Clearly this is dependent on n: the total rightward deletion rate is lower when n is near the right-hand end of the sequence, so that the probability that the residue escapes rightward deletion after time t depends on the history of the flanking

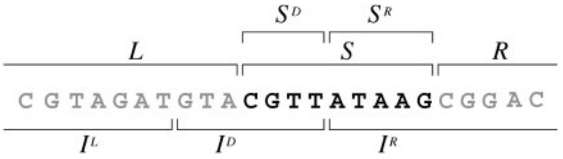

FIG. 2.—Embedding a sequence S in an infinite sequence $I = LSR$ increases the effective rate of deletions at the ends of S. Here, a deleted segment I^D partially overlaps the flanking sequences (shown in red) and is observed as a deletion of S^D.

sequence. This is bad news for dynamic programing alignment methods, which require conditional independence between different parts of the alignment.

It would be convenient to have a rightward deletion rate that is independent of the position along the sequence. This condition is satisfied in infinite sequences, in which the rightward deletion rate of any residue according to the long indel model is a constant, $\sum_{k=1}^{\infty} \mu_k = (1-r)\mu$. One way to achieve a position-independent rightward deletion rate is therefore to consider the sequence S as being embedded in an infinite sequence, $I = LSR$, where $L, R \in \Omega^{\infty}$. Here L and R are infinite flanking sequences that we do not observe.

Embedding S in an infinite sequence I has consequences for indels at the ends of S. Consider, for example, a deletion event in I.

$$I^L I^D I^R \to I^L I^R, \tag{9}$$

where the deleted subsequence I^D overlaps both S and the flanking sequence L (fig. 2). From the point of view of the embedded sequence S, this is equivalent to the event

$$S^D S^R \to S^R, \tag{10}$$

i.e., a deletion of the $|S^D|$ leftmost residues of S. Many deletions in I correspond to this deletion event in S. Thus, there is an effectively increased rate of deletions at both ends of S.

The form of the deletion rate $\rho_D(S^L,S^R,S^D)$ that is effectively experienced by sequence S must be modified from equation (8) to take account of these end effects. If both $|S^L| > 0$ and $|S^R| > 0$, then $\rho_D = \mu_{|S^D|}$ (as before); if $|S^L| = 0$ or $|S^R| = 0$ (but not both), then $\rho_D = \sum_{k=|S^D|}^{\infty} \mu_k$; whereas if both $|S^L| = |S^R| = 0$, then $\rho_D = \sum_{j=0}^{\infty} \sum_{k=|S^D|+j}^{\infty} \mu_k$. Substituting in the geometric deletion rate $\mu_k = \mu(1-r)^2 r^{k-1}$, we have

$$\rho_D(S^L,S^D,S^R) = \begin{cases} \mu(1-r)^2 r^{|S^D|-1} & \text{if } |S^L|>0, |S^R|>0;\\ \mu(1-r) r^{|S^D|-1} & \text{if one of } |S^L|, |S^R| \text{ is } 0;\\ \mu r^{|S^D|-1} & \text{if } |S^L| = |S^R| = 0. \end{cases} \tag{11}$$

To keep the model reversible, we set

$$\rho_I(S^L,S^I,S^R) = \rho_D(S^L,S^I,S^R)\gamma^{-|S^I|}\prod_{k=1}^{|S^I|} q(S^I_k). \tag{12}$$

Thus the effective insertion rate at the ends of S is also increased relative to the rate inside S. An interpretation of this is that, with a certain rate, the boundaries of S are extended into the infinite flanking sequence. This effect, which balances deletions extending into the flanking sequence, is responsible for the apparent extra "insertions" at the ends.

Table 1
Symbolic Representation of the Four Types of Chop Zone According to Whether They Adjoin the Left and/or Right Flanking Sequence (⋯) and Notation for Their Probabilities

$L_{ij} = P\left(\cdots \begin{array}{ccc\|} \#^i & -^j & \mathrm{M} \\ -^i & \#^j & \mathrm{M} \end{array}\right)$	$N_{ij} = P\left(\begin{array}{ccc\|} \#^i & -^j & \mathrm{M} \\ -^i & \#^j & \mathrm{M} \end{array}\right)$
$R_{ij} = P\left(\begin{array}{ccc} \#^i & -^j & \cdots \\ -^i & \#^j & \end{array}\right)$	$B_{ij} = P\left(\cdots \begin{array}{ccc} \#^i & -^j & \cdots \\ -^i & \#^j & \end{array}\right)$

NOTE.—These probabilities are conditional on observing the i (or $i + 1$) ancestral nucleotides. The # signs represent unaligned residues; M pairs represent aligned residues, and vertical bars represent chop zone boundaries.

Algorithm

Like the dynamic programming (DP) algorithms for the TKF models, the DP for the long indel model makes use of independence between different parts of the alignment, allowing us to "chop" the alignment into independent zones after every aligned ancestor-descendant residue pair. The justification for this runs as follows. Because of the embedding in an infinite sequence (see above), the total rightward deletion rate for residue n is independent of n or $|S|$. Independence of $|S|$ implies that the survival probability of residue n does not depend on indel events that start to the right of n. Moreover, the existence of an aligned residue pair somewhere left of n implies that no deletion events crossed that point in the sequence, so that the survival probability is also conditionally independent on the events left of that point. Therefore, if we chop an alignment after each aligned pair, the probability of the alignment simplifies to a product of conditionally independent probabilities for each chopped zone.

Let us consider the following alignment:

$$\cdots \begin{array}{ccc|cccc|cccc|c|cccc} \mathrm{A} & \mathrm{A} & \mathrm{G} & \mathrm{T} & - & \mathrm{A} & \mathrm{T} & - & - & - & \mathrm{G} & \mathrm{T} & \mathrm{A} & \mathrm{C} & \mathrm{C} & \mathrm{G} \\ - & - & \mathrm{C} & - & \mathrm{G} & - & \mathrm{T} & \mathrm{G} & \mathrm{C} & \mathrm{G} & \mathrm{G} & \mathrm{T} & - & - & - & - \end{array} \cdots$$

Here ⋯ denote the (unobserved) infinite flanking sequences, and we have indicated the places at which the alignment is chopped using vertical bars (|). Because of dependencies caused by overlapping indels, we cannot do dynamic programing algorithm inside a chop zone; we can only calculate the likelihood of whole chop zones. Within a chop zone, the sequence ordering of insertions and deletions is not specified. In other words, we regard the above alignment as indistinguishable from

$$\cdots \begin{array}{ccc|cccc|cccc|c|cccc} \mathrm{A} & \mathrm{A} & \mathrm{G} & \mathrm{T} & \mathrm{A} & - & \mathrm{T} & - & - & - & \mathrm{G} & \mathrm{T} & \mathrm{A} & \mathrm{C} & \mathrm{C} & \mathrm{G} \\ - & - & \mathrm{C} & - & - & \mathrm{G} & \mathrm{T} & \mathrm{G} & \mathrm{C} & \mathrm{G} & \mathrm{G} & \mathrm{T} & - & - & - & - \end{array} \cdots$$

where the fifth and sixth columns have been swapped, because these alignments represent the same evolutionary homology between the residues of the two sequences.

There are four kinds of chop zone, distinguished by whether they border the left or right flanking sequence, neither of them, or both. The conditional probabilities of observing these chop zones are denoted by L_{ij}, R_{ij}, N_{ij}, and B_{ij}, respectively (see table 1). The indices specify the fate of nucleotides within the chop zone, namely the deletion of

i residues, which are replaced by j newly inserted residues. Chop zones that do not border the right flanking sequence (L_{ij} and N_{ij}) furthermore end in a single aligned residue pair; the others do not.

To calculate alignment likelihoods, we also need the probability $p_t(x \to y)$ that residue x mutated to residue y in time t, found as usual by exponentiating the substitution rate matrix, and the residue equilibrium probability distribution $q(\omega)$. As with the TKF91 model, we separate the likelihood into a product of independent substitution and indel probabilities. Thus, conditional on the ancestral sequence, the probability of our example alignment is the following product of probabilities:

$$\begin{aligned} & L_{2,0} \times p_t(\mathrm{G} \to \mathrm{C}) \\ & \quad \times N_{2,1} \times q(\mathrm{G}) p_t(\mathrm{T} \to \mathrm{T}) \\ & \quad \times N_{0,3} \times q(\mathrm{G}) q(\mathrm{C}) q(\mathrm{G}) p_t(\mathrm{G} \to \mathrm{G}) \\ & \quad \times N_{0,0} \times p_t(\mathrm{T} \to \mathrm{T}) \\ & \quad \times R_{4,0} \end{aligned} \tag{13}$$

The full alignment probability is obtained by multiplying this by the equilibrium probability of the ancestral sequence, π(AAGTATGTACCG) (see the section titled *Long Indel Model*).

Finite Trajectory Approximation

To calculate alignment likelihoods, we need to compute probabilities for the four kinds of chop zone listed in table 1. In each case, this involves enumerating the different chains of events, or *trajectories*, that give rise to the particular outcome, and then calculate the appropriate transition probability for the zone.

Suppose we want to calculate $N_{3,2}$. If A represents a deleted ancestral residue, M the conserved ancestral residue and B an inserted residue, then the zone starts as the four-residue sequence AAAM and ends as the three-residue sequence BBM. One valid, three-event trajectory generating this outcome is shown in figure 1.

Abstractly, a trajectory is viewed as a sequence of events, and the configuration of the sequence (within the chop zone) in between events is referred to as the *state* of the sequence at that moment. To calculate a trajectory probability, we need the rate for all events in the trajectory, as well as the *exit rate* for each state visited (the exit rate of a state is defined as the total rate of mutation events for that state—i.e., the sum of the outgoing transition rates). Because the required probability is conditional on no deletion event crossing the left chop zone boundary, this includes only rightward deletions that originate in the current chop zone. However, to calculate the exit rate we must take into account all such deletions, including those that continue into neighboring chop zones on the right. To make sure the last residue pair of a chop zone remains homologous, insertions are assigned to the chop zone of the residue to the right of the insertion, except at the end of the sequence, where it is assigned to the last chop zone. Other insertions (or deletions) are taken not to change the state.

Exact calculation of these probabilities involves solving some partial differential equations for a generating function (Miklós and Toroczkai 2001), and it becomes very complicated, particularly when deletions are involved. Instead of using exact results, we approximated the chop zone probabilities by bounding the number of indel events, and the indel lengths per event. This reduces the infinite sum over all trajectories to a finite sum. We find empirically that, in the parameter regime of the alignments we studied, it is sufficient to allow at most three indel events, and indel lengths of at most 100. (Gaps exceeding 100 residues are extremely rare in the database we used for testing, accounting for less than 0.1% of all gaps, and the probability of more than three indel events overlapping, assuming an indel rate of $\mu = 0.1$ and evolutionary distance 3, is less than $(1 - e^{-3\times 0.1})^4 < 0.005$ per site.)

Using this finite trajectory approximation, we can sum the likelihoods of the zone trajectories directly. A recursive algorithm for computing the likelihood of an individual trajectory is given in Appendix A.

Dynamic Programming

To compute the joint likelihood of two sequences, we sum over all chop zone assignments. The following DP algorithm achieves this. Let P^i_j be the sum of probabilities of all partial alignments of the first i residues of ancestral sequence A with the first j residues of descendant sequence B, where the last characters of the two subsequences are homologous, and conditional on observing the ancestral sequence, then

$$\begin{aligned} P^i_j = {} & L_{i-1,j-1} p_t(A_i \to B_j) \prod_{k=1}^{j-1} q(B_k) \\ & + \sum_{n=0}^{i-2} \sum_{m=0}^{j-2} P^{i-n-1}_{j-m-1} N_{nm} p_t(A_i \to B_j) \prod_{k=j-m}^{j-1} q(B_k), \end{aligned} \tag{14}$$

where the rightmost product is 1 if $m = 0$. The probability of observing B conditional on A is

$$\begin{aligned} p_t(B \mid A) = {} & B_{|A|,|B|} \prod_{k=1}^{|B|} q(B_k) \\ & + \sum_{n=0}^{|A|-1} \sum_{m=0}^{|B|-1} P^{|A|-n}_{|B|-m} R_{nm} \prod_{k=|B|-m+1}^{|B|} q(B_k). \end{aligned} \tag{15}$$

The joint likelihood of the two sequences is $P_t(A,B) = P_t(B \mid A)\pi(A)$. These formulas can be simplified somewhat by taking the equilibrium distribution probabilities outside the recursion. Let $F^i_j = P^i_j / \prod_{k=1}^{j} q(B_k)$, then

$$F^i_j = \frac{p_t(A_i \to B_j)}{q(B_j)} \left(L_{i-1,j-1} + \sum_{n=0}^{i-2} \sum_{m=0}^{j-2} F^{i-n-1}_{j-m-1} N_{nm} \right), \tag{16}$$

and the joint likelihood $P_t(A,B)$ of observing A and B becomes

$$\left(B_{|A|,|B|} + \sum_{n=0}^{|A|-1} \sum_{m=0}^{|B|-1} F^{|A|-n}_{|B|-m} R_{nm} \right) \pi(A) \prod_{k=1}^{|B|} q(B_k), \tag{17}$$

The time complexity of this recursion is $O(|A|^2|B|^2)$, because we need to look back to the start of each sequence

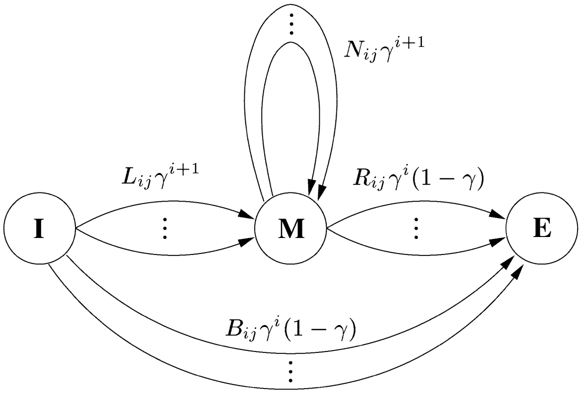

FIG. 3.—A hidden Markov model formulation of the long indel model. The emission probabilities (associated to transitions) are not included. The parameter $\gamma = \lambda_1/\mu_1$ is the parameter governing the geometric equilibrium length distribution.

to calculate each entry. However, it is straightforward to use corner-cutting methods (Hein et al. 2000), as the likelihood of big "unalignable" regions is negligible. Among other approximations, this involves dropping terms involving B_{ij}, corresponding to completely unaligned sequences. With such corner cutting, the running time can be reduced to $O(nm)$. In practice, the finite-trajectory approximation of the probability functions takes roughly the same time as the DP itself, for the event length cutoff of 3 that we used.

The long indel model can be formulated as a hidden Markov model (HMM) with three states, Initial, Middle, and End, and transitions that correspond to the chop zones (fig. 3). In this formulation, emissions are associated to transitions [the "Mealy machine" view (Durbin et al. 1998)]. The DP algorithm for computing the joint likelihood is then simply the forward algorithm for HMMs, and general HMM algorithms for sampling and finding the most likely path can be used straightforwardly.

Evaluation

To evaluate the various alignment algorithms, we used the structural alignments from the HOMSTRAD database as a benchmark (Mizuguchi et al. 1998). We randomly selected $\frac{2}{3}$ of the pairwise alignments as a training set. The remaining $\frac{1}{3}$ served as a test set. Alignment accuracy was measured by residue-pair overlap; see e.g., Holmes and Durbin (1998). Because we compare global alignment methods, we trimmed the sequences to the subsequence bounded by the first and last aligned amino acid pair.

The alignments in the HOMSTRAD database have sequence similarity ranging from 3% to 99%, with 10% quantiles at 14% and 83%. For the Gotoh alignment algorithm, we used the standard BLOSUM62 matrix, which gave similar but slightly better results than other BLOSUM matrices. We optimized gap opening and extension parameters to maximize the Gotoh algorithm's accuracy on the training set.

To test the probabilistic models, we needed a substitution rate matrix. Using the Dayhoff matrix as initial guess, we obtained maximum likelihood (ML) evolutionary distances t for the training set alignments. We then estimated ML rate matrix by Expectation Maximization (Holmes and Rubin 2002), normalizing the matrix for one expected mutation per site at $t - 1$. This procedure was iterated until both the evolutionary distances and the rate matrix converged. The resulting rate matrix was used for all probabilistic models.

We obtained the deletion rate parameters, μ and (for the TKF92 and long indel model) r by ML estimation on alignments from the training set, using the time estimates described above. The insertion rate parameter λ was allowed to vary for each sequence pair to make the expected sequence length equal the average length of the two sequences under consideration. We refer to μ and r as the *indel rate parameters*.

For the test set, we fixed the indel rate parameters and we performed a ML parameter estimation of the evolutionary distance t for each pair of sequences, summing over all possible alignments. We then recovered to maximum likelihood alignment using the Viterbi algorithm (Durbin et al. 1998).

For the long indel model, we also implemented a program computing the posterior labeling (Durbin et al. 1998) of each aligned column of the Viterbi alignment, that is, the probability of the aligned column given the data.

Alignment software for the long indel model is available on request (miklos@stats.ox.ac.uk).

Results

The gap opening and extension parameters that maximize the accuracy of Gotoh's algorithm on our training set are (15, 2), using the BLOSUM62 matrix, giving an 82.2% overlap on the test set. These parameters are higher than the NCBI-recommended parameters (11, 1); for these parameters Gotoh's algorithm performs slightly worse, with 80.9% overlap.

Maximum likelihood time estimates (MLTEs) of evolutionary separations for the HOMSTRAD pairwise alignments, estimated using a point substitution model, yielded times between 0.014 and 4.23 (in units of expected substitutions per site) with 10% quantiles at 0.37 and 2.26. We refer to these estimates as *Homstrad MLTEs*. We also calculated time MLTEs for unaligned sequences, using the various indel models to sum over all alignments, referring to these individually as *TKF91 MLTEs*, *TKF92 MLTEs*, and *Long indel MLTEs*, and collectively as *model-based MLTEs*. The relationship between Homstrad MLTE and long percentage identity was approximately linear (data not shown).

For TKF91, TKF92, and the Long indel model, we obtained evolutionary parameters μ and r by Maximum Likelihood (table 2). In addition, we endowed the Long indel model with a mixed geometric distribution for the indel rates,

$$\mu_k = \mu[\alpha(1 - r_1)^2 r_1^{k-1} + (1 - \alpha)(1 - r_2)^2 r_2^{k-1}], \qquad (18)$$

where we estimated $\alpha = 0.40$, $r_1 = 0.55$, $r_2 = 0.90$ by counting observed indels in the training set. We then

Table 2
Estimated Evolutionary Parameters for Evolutionary Models

Alignment Method	μ	r	α
TKF91	0.043	—	—
TKF92	0.038	0.67	—
Long indel	0.049	0.543	—
Long indel, mixed geometric	0.095	0.55; 0.9	0.4

optimized μ on total overlap. The overall performance of the various alignment algorithms is summarized in table 3.

As an alternative to fixing the indel parameters μ and r for the entire test set, we also computed maximum likelihood values for the indel parameters, and time, for each sequence pair individually. Because of computing constraints, we only did this for a subset of our full test set. Results were similar to the results for the given procedure (data not shown).

The relationships between the model-based and Homstrad MLTEs are shown in figure 4. The TKF91 time estimates often diverged to infinity, probably as a result of a bad model fit. This problem was less pronounced in TKF92, and all but absent for the long indel model. All model-based estimates of divergence times tend to be lower than estimates based on Homstrad alignments, with least-squares slopes (on data with outliers removed) in the range 0.75–0.78 for the three models, all significantly different from 1. The hypothesis that the slopes for all three models were equal could not be rejected at a 5% level.

Assessed on HOMSTRAD overlap, the TKF91 model is the least accurate alignment method, though it is comparable to TKF92. The long indel model is clearly better, and as good as the simple Gotoh algorithm. A heuristic explanation to the weak performance of TKF91 and TKF92 is that in the absence of strong homology, TKF91 tends to give very fragmented alignments, whereas TKF92 alignment is at the other extreme: it prefers as few fragments as possible.

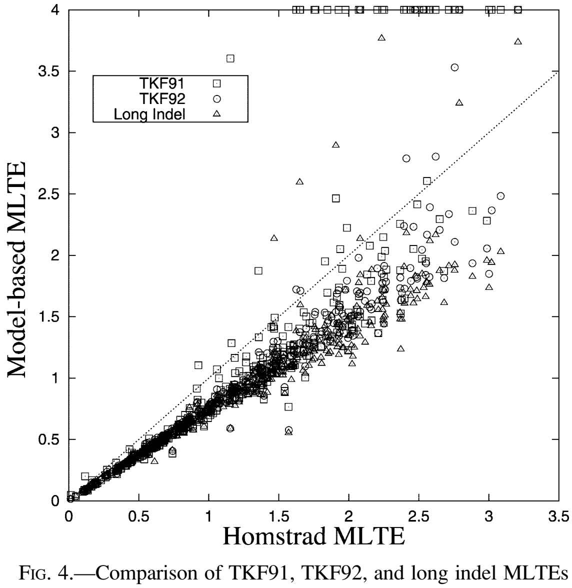

FIG. 4.—Comparison of TKF91, TKF92, and long indel MLTEs (y-axis) with Homstrad MLTEs (x-axis), see *Results* for definitions. The dotted line is $x = y$. The TKF time parameters sometimes ran away during ML parameter estimation, and thus appear at the very top of the graph. Note that the model-based MLTEs tend to be lower than the Homstrad MLTE, for all models. Small local database misalignments could cause such an effect; see *Discussion*.

Table 3
Performance of Alignment Methods, as Measured by Alignment Accuracy or "Overlap," the Percentage of Alignment Columns Identical to Those of the HOMSTRAD Structural Alignments

Alignment Method	Training Set Optimization[a]	Test Set Overlap (%)
TKF91	ML	73.8
TKF92	ML	75.9
Gotoh (BLOSUM62)	NCBI defaults	80.9
Long indel	ML	81.1
Long indel, mixed geometric	Accuracy	82.1
Gotoh (BLOSUM62)	Accuracy	82.2

[a] Parameters were optimized over a training set to maximize either likelihood or overlap. In addition, for the Gotoh algorithm we used NCBI (National Center for Biotechnology Information) defaults for gap opening and gap extension parameters.

At higher sequence identity all models perform much better. This can clearly be seen for the long indel model in figure 5, which plots the overlap for the long indel and Gotoh algorithms as a function of Homstrad MLTE.

We plotted the Viterbi alignment together with its posterior labeling, and we indicated the correctly aligned columns for several sequence pairs. We found that posterior labeling is a good indicator of correctness of alignment (fig. 6).

Discussion

Using rate grammar notation, we have presented an evolutionary model that allows multiple-residue indels without introducing hidden information such as fragment boundaries. We described alignment algorithms for our long indel model, using a finite trajectory approximation.

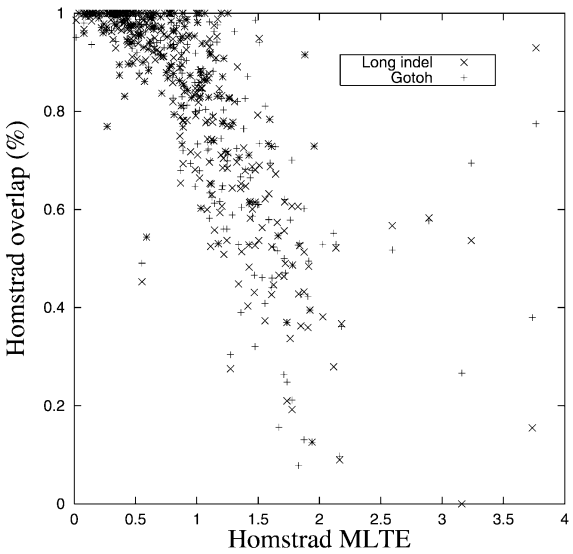

FIG. 5.—Accuracy of Gotoh and long indel alignment algorithms, as a function of Homstrad MLTE.

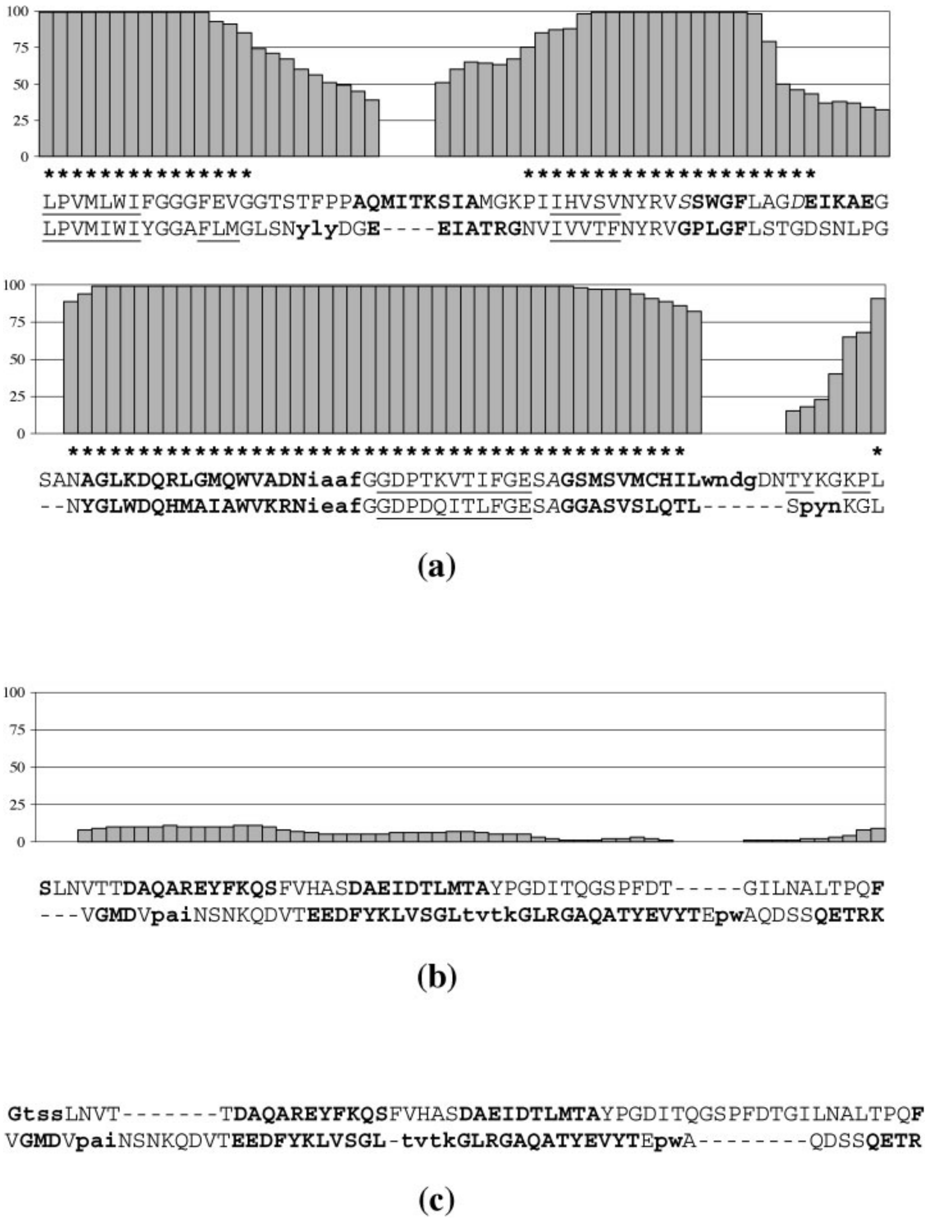

FIG. 6.—Maximum likelihood alignment of triacylglycerol hydrolase (*Candida rugosa*) and bile-salt–activated lipase (*Bos taurus*) using the long indel model, and posterior probabilities ("reliabilities") for individual aligned columns. Bold uppercase characters represent alpha helices, bold lowercase characters represent 3_{10} helices, and underlined characters represent beta-sheets. Correctly aligned columns are marked with asterisks. *a*. Part of the alignment with high overlap with the HOMSTRAD structural alignment, showing that posterior probabilities are good indicators of correctness of alignment. *b*. Part of the alignment without overlap with the structural alignment, corresponding to a marked decrease in posterior column probabilities. This section is hard to align as sequences contain repetitive alpha helices in this region, and the Viterbi alignment shifted these alpha helices, causing no correctly aligned residues for this part. *c*. HOMSTRAD structural alignment of the sequence segment of *b*.

We implemented and tested the TKF91, TKF92, and long indel models on structural alignments from HOMSTRAD.

Our data suggest that the long indel model gives better time estimates than TKF92, which gives better estimates than TKF91. An implication of this result is that the long indel model is preferable for molecular phylogeny. In terms of alignment accuracy, the ordering is TKF91 $<$ TKF92 $<$ long indel $\approx$ Gotoh. The high accuracy of the relatively simple Gotoh algorithm is unexpected, and demonstrates that this algorithm remains a powerful tool for molecular biologists.

Although there is no significant difference in alignment accuracy between Gotoh and the long indel model, the latter provides more information about the alignment. It allows us to compute posterior probabilities ("reliabilities") of individual alignment columns (Durbin et al. 1998). We find that these reliabilities are good predictors of correctness of alignment (see fig. 6). The ability to assign reliabilities to parts of the alignment is a major advantage of using probabilistic, as opposed to score-based, models.

It is interesting to note that the divergence time estimates obtained from the substitution model on

HOMSTRAD structural alignments are systematically higher than those obtained from the evolutionary models, which combine substitution and indel events (fig. 4). This is surprising, because heterogeneous mutation rates along the sequence can be expected to bias the estimates from HOMSTRAD alignments toward lower values, because regions experiencing a low mutation rate have more alignable parts. Furthermore, all models investigated estimate divergence times by considering both insertion-deletions and substitutions. Therefore, unlike in score-based alignment methods, aligning more similar amino acids by introducing more gaps does not automatically yield lower time estimates. However, alignments based on structural similarity might not always reflect homology, as small shifts can occur because of the spatial extent and periodicity of structural elements; see figure 7 for an example. This biases the time estimates based on HOMSTRAD alignments toward higher values, and this effect seems to dominate.

The long indel alignment algorithm is rather slow, but speedups are possible. Specifically, the current algorithm uses an arbitrary emission-length Pair HMM. We anticipate that our model can be reasonably well approximated with a single-character emission Pair HMM, and this will yield an algorithm with time complexity $O(L^2)$, like TKF91.

Our evaluation method used the HOMSTRAD alignments as a guide to trim the sequences to the alignable regions. This means that the absolute alignment accuracies of table 3 are probably overestimates. We chose this method because the alignment algorithms we implemented are global in nature; however, we plan to develop local versions. Note that because of the long indel model's increased indel rate at both sequence ends, this data preparation method puts it at a slight disadvantage compared with the other methods.

Although Gotoh's algorithm has an advantage in our comparison because Gotoh uses optimized parameters, whereas for the evolutionary models we estimate parameters by maximum likelihood, accuracies for the long indel model with estimated parameters closely approach those of Gotoh's algorithm (81.1% vs. 82.2%). This shows the utility of ML parameter estimation for the long indel model. The ability to use such objective parameter estimation methods is a general advantage of probabilistic models over score-based ones.

Using a hybrid method of optimization and curve fitting, we estimated parameters for a mixed geometric indel rate distribution, resulting in a further improved alignment accuracy (to 82.1%). It is interesting that Gotoh reaches this accuracy without such sophisticated models. To test whether Gotoh's success was due to the BLOSUM62 matrix, we tried forcing our long indel algorithm to use BLOSUM62, but this did not perform any better than our ML rate matrix. One factor that might be relevant, however, is that affine gap penalties were used in constructing the HOMSTRAD structural alignments. These particular penalties will bias the resulting gap length distribution, which will naturally favor Gotoh's algorithm.

In general, there may be theoretical limits to the accuracy obtainable with respect to databases such as HOMSTRAD: the homology signal may be too weak in places (Holmes and Durbin 1998), or structural and sequence homology may be mutually inconsistent. Nonetheless, the enhanced performance of the long indel model over TKF91 and TKF92 is grounds for encouragement. Its performance is comparable with that of Gotoh's algorithm. In addition, it allows for inference of evolutionary divergence, and it provides information on the reliability of different parts of the alignment. This information can be useful, e.g., for 3D structural modeling, where reliable regions can be used as a seed for homology modeling.

By sampling alignments in local neighborhoods on a phylogenetic tree, a multiple-sequence Markov Chain Monte Carlo (MCMC) version of our alignment algorithm can be constructed [Holmes and Bruno 2001; Jensen and Hein 2002]. In this context, the trajectory likelihood algorithm of Appendix A will again be useful. As the corpus of data from large-scale sequencing projects grows and our understanding of molecular evolution deepens, the incorporation of this understanding leads to more realistic models, that often cannot be solved analytically. This is especially common when the state space of such models is large, as is the case for whole-sequence models. In such situations, MCMC approaches can be used (Pedersen and Jensen 2001; Robinson et al. 2003). Since the algorithm of Appendix A applies to general discrete-state continuous-time Markov models, it is useful for any MCMC procedure that samples trajectories of such models [Robinson et al., 2003]. So, with our algorithm, it is possible to avoid sampling event times (which are true nuisance variables) and thus to improve the performance of the MCMC procedure.

The long indel model is the first evolutionary model to incorporate realistic long indel events without introducing hidden information. As such, it should find applications in alignment, phylogenetic analysis and profiling.

Appendix A: Trajectory Likelihoods

In this appendix we describe how to calculate likelihoods of histories and trajectories through some state space. Here, we use the terminology that a trajectory is a path through state space, $\phi_0 \to \phi_1 \to \cdots \to \phi_N$, while a history is a path with times $t_1, \ldots, t_N$ for each change of state.

Because the probability of an event occurring in an infinitesimal time interval is vanishingly small, individual histories have infinitesimal likelihood, but finite likelihood density. Suppose that the model starts in state ϕ_0 at time $t_0 = 0$. At time t_1, it mutates to state ϕ_1; at time t_2, it mutates to ϕ_2; and so on up to ϕ_N at time $t_N \leq T$. Setting $t_{N+1} = T$, the likelihood density of this history for the time interval $0 \leq t \leq T$, conditional on the initial state, is

$$f(\{\phi_0, t_1, \phi_1, \ldots, t_N, \phi_N\} \mid \phi_0) = e^{-\zeta_0(t_1-t_0)} r_1 e^{-\zeta_1(t_2-t_1)} r_2 \cdots r_N e^{-\zeta_N(t_{N+1}-t_N)}, \qquad (19)$$

where $\zeta_i = -R(\phi_i,\phi_i)$ is the total mutation rate (the *exit rate*) of state ϕ_i, and $r_i = R(\phi_{i-1},\phi_i)$ is the rate of the specific mutation $\phi_{i-1} \to \phi_i$. The likelihood of a trajectory is obtained by integrating the history likelihood density

```
DMGPQPRAEASWQFFMS-DKPLRLAVSL
--PGPQPTAETTRQFLMSDKPLHLEASL
```

FIG. 7.—Part of HOMSTRAD structural alignment of *Bos taurus* S-arrestin (amino acids 1–27) and beta-arrestin 1 (amino acids 1–26), clearly shifted with respect to their homology.

over the N variables $t_1 \ldots t_N$, where $t_0 \leq t_1 \leq \cdots \leq t_N \leq T$, and we set $t_0 = 0$:

$$P(\phi_0 \to \cdots \to \phi_N; T \mid \phi_0) = \int_{t_1=t_0}^{T} \cdots \int_{t_N=t_{N-1}}^{T} f(\{\phi_0, t_1 \ldots t_N, \phi_N\}) \mathrm{d}t_1 \ldots \mathrm{d}t_N. \tag{20}$$

Note that the likelihood is invariant under permutation of the exit rates $\zeta_0, \ldots, \zeta_N$. An equivalent, recursive formula for this likelihood is

$$P(\phi_0 \to \cdots \to \phi_N; T \mid \phi_0) = \begin{cases} \int_0^T P(\phi_0 \to \cdots \to \phi_{N-1}; t \mid \phi_0) & \\ \quad \times\, r_N e^{-\zeta_N (T-t)} \mathrm{d}t & \text{if } N > 0 \\ e^{-\zeta_0 T} & \text{if } N = 0, \end{cases} \tag{21}$$

with ζ_i and r_i defined as before.

A subtlety arises in the following analysis if there is *degeneracy* in the set of exit rates; that is, if any of the rates $\zeta_0, \ldots, \zeta_N$ are equal. To account for this, we suppose that $\{\xi_0, \ldots, \xi_M\}$ is the corresponding set of rates *with duplicates removed* (so $M \leq N$), and let $d_n + 1$ be the multiplicity of the nondegenerate rate ξ_n among the rates $\zeta_0, \ldots, \zeta_N$; thus, exit rate ξ_0 occurs $d_{0\,+\,1}$ times, exit rate ξ_1 occurs $d_1 + 1$ times, and so on; therefore $\sum d_i = N - M$.

As a trial solution, we write the likelihood in the form

$$P(\phi_0 \to \cdots \to \phi_N; T \mid \phi_0) = \left(\prod_{n=1}^{N} r_n \right) \sum_{n=0}^{M} e^{-\xi_n T} \sum_{k=0}^{d_n} c_n^k T^k, \tag{22}$$

where c_n^k is the coefficient of T^k in the polynomial multiplying $e^{-\xi_n T}$. These coefficients can be computed by a recursive algorithm that we now describe. The algorithm makes use of the following identity for $x \neq 0$, $k \geq 0$:

$$\frac{(-x)^{k+1}}{k!} \int_0^T t^k e^{tx}\, \mathrm{d}t = 1 - e^{Tx} \sum_{i=0}^{k} \frac{(-Tx)^i}{i!}, \tag{23}$$

which can be derived by writing the left-hand side as I_{k+1} whereupon integration by parts shows that $I_{k+1} = I_k - e^{Tx} \frac{(-Tx)^k}{k!}$, taking $I_0 = 1$. This identity is used as follows. Suppose the c_n^k are known for some trajectory $\{\phi_0, \ldots, \phi_N\}$, and we wish to calculate new coefficients for the trajectory $\{\phi_0, \ldots, \phi_{N+1}\}$ which is one step longer. Using the recursive definition of the trajectory likelihood we get

$$\begin{aligned} P(\phi_0 \to \cdots \to \phi_{N+1}; T \mid \phi_0) &= \left(\prod_{n=1}^{N} r_n \right) \int_{t=0}^{T} \left(\sum_{n=0}^{M} e^{-\xi_n t} \sum_{k=0}^{d_n} c_n^k t^k \right) r_{N+1} e^{-\zeta(T-t)} \mathrm{d}t \\ &= \left(\prod_{n=1}^{N+1} r_n \right) \sum_{n=0}^{M} \sum_{k=0}^{d_n} c_n^k e^{-\zeta T} \int_{t=0}^{T} t^k e^{(\zeta - \xi_n)^t} \mathrm{d}t, \end{aligned} \tag{24}$$

where we wrote $\zeta = \zeta_{N\,+\,1}$. For fixed n, the inner summation over k depends on whether $\zeta = \xi_n$ or not. If they are different, the result is

$$\sum_{k=0}^{d_n} c_n^k \left(e^{-\zeta T} \frac{k!}{(\xi_n - \zeta)^{k+1}} - e^{-\xi_n T} \sum_{i=0}^{k} \frac{k! T^i}{i! (\xi_n - \zeta)^{k-i+1}} \right), \tag{25}$$

whereas when $\zeta = \xi_n$ we get

$$\sum_{k=0}^{d_n} c_n^k e^{-\xi_n T} \frac{T^{k+1}}{k+1} \tag{26}$$

Now if ζ is not in $\{\xi_0, \ldots, \xi_M\}$ we set $\xi_{M+1} = \zeta$ and $M' = M + 1$, otherwise $M' = M$. Writing the solution in the form (22) with coefficients $c_n^{k'}$, we get for $n = 0, \ldots, M'$:

$$c_n^{k'} = - \sum_{i=k}^{d_n} c_n^i \frac{i!}{k! (\xi_n - \zeta)^{i-k+1}}, \qquad (\zeta \neq \xi_n, k = 0, \ldots, d_n) \tag{27}$$

$$c_n^{0'} = \sum_{m \neq n} \sum_{i=0}^{d_m} c_m^i \frac{i!}{(\xi_m - \zeta)^{i+1}}, \qquad (\zeta = \xi_n) \tag{28}$$

$$c_n^{k'} = \frac{c_n^{k-1}}{k} \qquad (\zeta = \xi_n; 1 \leq k \leq d_n + 1), \tag{29}$$

so that if we set $d_n^{'} = d_n$ if $\zeta \neq \xi_n$, and $d_n^{'} = d_n + 1$ if $\zeta = \xi_n$, we recover the form of the trial solution (22), and the $d_n^{'}$ indeed correspond to the multiplicities of the rates ξ_n as asserted. This leads to the following algorithm:

Algorithm 1 (*Trajectory likelihood*)

Input: *Transition rates* $r_1, \ldots, r_N$; *exit rates* $\zeta_0, \ldots, \zeta_N$; *time* T.

Output: *Probability* $P(\phi_0 \to \cdots \to \phi_N; T \mid \phi_0)$ *for the trajectory with rates* r_i *and* ζ_i.

Algorithm:

```
$M \leftarrow 0$; $\xi_0 \leftarrow \zeta_0$; $d_0 \leftarrow 0$; $c_0^0 \leftarrow 1$.
For $i$ from 1 to $N$, do the following:
    If $\zeta_i \notin \{\xi_0, \ldots, \xi_M\}$, then:
        $M \leftarrow M + 1$; $\xi_M \leftarrow \zeta_i$; $d_M \leftarrow 0$
    Else:
        $d_j \leftarrow d_j + 1$, for the $j$ satisfying $\zeta_i = \xi_j$.
    EndIf
    For $n$ from 0 to $M$, then for $k$ from 0 to $d_n$, do
    the following:
        If $\xi_n \neq \zeta_i$, then:
            $u_n^{k'} \leftarrow - \sum_{j=k}^{d_n} c_n^j \frac{j!}{k!(\xi_n - \zeta_i)^{j-k+1}}$
```

```
      Else:
        If $k = 0$, then:
          $u_n^{k'} \leftarrow \sum_{m \neq n} \sum_{j=0}^{d_m} c_m^j \frac{j!}{(\xi_m - \zeta_i)^{j+1}}$
        Else:
          $u_n^{k'} \leftarrow c_n^{k-1}/k$
        EndIf
      EndIf
    EndFor ($n$ and $k$)
    $c_n^k \leftarrow u_n^{k'}$, for $n = 0, \ldots, M$ and $k = 0, \ldots, d_n$.
  EndFor ($i$)
  Return $\left(\prod_{n=1}^{N} r_n\right) \sum_{n=0}^{M} e^{-\xi_n T} \sum_{k=0}^{d_n} c_n^k \, T^k$
```

This algorithm has computational time complexity $O(N^2)$. Two special cases are worth noting, because the coefficients can be obtained in closed form. If no two rates ζ_n, ζ_m are equal, then $d_n = 0$ and $c_n^0 = \prod_{m \neq n}(\zeta_m - \zeta_n)^{-1}$. If *all* the rates ζ_n are identical, then $d_0 = N$ and $c_0^N = 1/N!$ while $c_0^k = 0$ for $k < N$; apart from a factor r_n/ζ_n, this is just the Poisson distribution for the number of mutation events N.

The expected amount of evolutionary time spent in a particular state ϕ can be found using a variation of this recursion.

Appendix B: Consistency Relations

Because of the complicated combinatorics involved in computing the chop zone probabilities N_{ij}, L_{ij}, R_{ij}, and B_{ij}, it is useful to have some consistency checks:

$$\gamma^{i+1} X_{ij} = \gamma^{j+1} X_{ji} \qquad \text{(where X is one of } L, N, R, B\text{)}, \tag{30}$$

$$\sum_{i,j=0}^{\infty} L_{ij} \gamma^{i+1} + \sum_{i,j=0}^{\infty} B_{ij} \gamma^{i} = 1, \tag{31}$$

$$\sum_{i,j=0}^{\infty} N_{ij} \gamma^{i+1} + \sum_{i,j=0}^{\infty} R_{ij} \gamma^{i} = 1, \tag{32}$$

$$\sum_{i=0}^{\infty} L_{0,i} = e^{-\sum_{k=1}^{\infty} k\mu_k}, \qquad \sum_{i=0}^{\infty} N_{0,i} = e^{-\sum_{k=1}^{\infty} \mu_k}, \tag{33}$$

$$\sum_{i=0}^{\infty} R_{0,i} = 1, \qquad \sum_{i,j} N_{i,j} = 1. \tag{34}$$

Equations (30) express detailed balance, and hold because of reversibility of the model. Equations (31) and (32) assert that the exit probability of Markov states I and M resp. are 1. The survival probabilities of the first nucleotide of a left or central chop zone are related to chop zone probabilities by (33). Equations (34) finally sum out all possibilities at the end of the sequence when there are no nucleotides to delete, and all possible chop zones for an infinitely long sequence, respectively.

Equations (30) also hold for finite-order approximations; the others are only true for the exact probabilities, but are useful nonetheless to spot errors.

Appendix C: Posterior Likelihood

As with generic HMMs, it is possible to compute the posterior likelihood of particular alignment columns by calculating the likelihood of visiting a certain state using the Forward-Backward algorithm (Durbin et al. 1998). As a consequence of employing the Mealy machine view, to compute the reliability of an unaligned residue, we have to sum over all possible *transitions* associated to this possibility. Because many transitions may contribute to the emission of an unaligned residue, this is an expensive operation; however, there exists a dynamic programming solution.

Let F_j^i be the “forward” recursion as given in the section titled, *Dynamic Programming*, and G_j^i the result of the corresponding Backward algorithm. If U_i is the posterior probability that ancestral residue i is unaligned, then

$$U_1 = \pi(A) q(B) \left(\sum_{i=1}^{|A|-1} \sum_{j=0}^{|B|-1} L_{ij} \frac{P_t(A_{i+1} \rightarrow B_{j+1})}{q(B_{j+1})} G_{j+1}^{i+1} + B_{|A|,|B|} \right) \tag{35}$$

$$\begin{aligned} U_{n+1} = U_n + \Bigg(& \sum_{j=0}^{|B|-1} \left(F_{j+1}^{n} R_{|A|-n,|B|-j-1} - L_{n-1,j} \frac{P_t(A_n \rightarrow B_{j+1})}{q(B_{j+1})} G_{j+1}^{n} \right) \\ & + \sum_{k=0}^{|B|-2} \sum_{i=1}^{|A|-n-1} \sum_{j=0}^{|B|-k-2} F_{k+1}^{n} N_{ij} \frac{P_t(A_{n+i+1} \rightarrow B_{k+j+2})}{q(B_{k+j+2})} G_{k+j+2}^{n+i+1} \\ & - \sum_{k=0}^{|B|-2} \sum_{i=0}^{n-1} \sum_{j=0}^{k} F_{k-j+1}^{n-i} N_{ij} \frac{P_t(A_{n+1} \rightarrow B_{k+2})}{q(B_{k+2})} G_{k+2}^{n+1} \Bigg) \pi(A) q(B) \end{aligned} \tag{36}$$

Here $q(B) = \prod_{i=1}^{|B|} q(B_i)$. A similar recursion exists for the unaligned residues of the descendant sequence. The running time for these recursions is the same as for the Forward and Backward recursions.

Acknowledgments

This research was funded by grants from the Engineering and Physical Sciences Research Council (EPSRC) (code HAMJW) and the Medical Research Council (MRC) (code HAMKA). The authors thank Jotun Hein for support and encouragement. I.M. started his contribution to this work in the Collegium Budapest, which he thanks for its generous support. The idea of using an infinite sequence was suggested by Jeffrey Thorne. Two anonymous referees are thanked for their valuable comments.

Literature Cited

Bruno, W. J., and L. Arvestad. 1997. Estimation of reversible substitution matrices from multiple pairs of sequences. J. Mol. Evol. **45**:696–703.

Durbin, R., S. Eddy, A. Krogh, and G. Mitchison. 1998. Biological sequence analysis: Probabilistic models of proteins and nucleic acids. Cambridge University Press, Cambridge, U.K.

Gotoh, O. 1982. An improved algorithm for matching biological sequences. J. Mol. Biol. **162**:705–708.

Hein, J. 2001. An algorithm for statistical alignment of sequences related by a binary tree. Pp. 179–190 *in* Pac. Symp. Biocomp., R. B. Altman, A. K. Dunker, L. Hunter, K. Lauderdale, and T. E. Klein, eds, World Scientific, Singapore.

Hein, J., C. Wiuf, B. Knudsen, M. B. Moller, and G. Wibling. 2000. Statistical alignment: computational properties, homology testing and goodness-of-fit. J. Mol. Biol. **302**:265–279.

Holmes, I., and W. J. Bruno. 2001. Evolutionary HMMs: a Bayesian approach to multiple alignment. Bioinformatics **17**:803–820.

Holmes, I., and R. Durbin. 1998. Dynamic programming alignment accuracy. J. Comp. Biol. **5**:493–504.

Holmes, I., and G. M. Rubin. 2002. An expectation maximization algorithm for training hidden substitution models. J. Mol. Biol. **317**:757–768.

Jensen, J., and J. Hein. 2002. Gibbs sampler for statistical multiple alignment. Technical Report 429, Department of Theoretical Statistics, University of Aarhus, Denmark.

Lunter, G. A., L. Miklós, Y. S. Song, and J. Hein. 2003. An improved algorithm for multiple alignment on arbitrary phylogenetic trees. J. Comp. Biol. **10**:869–889.

Metzler, D., 2003. Statistical alignment based on fragment insertion and deletion models. Bioinformatics **19**:490–499.

Metzler, D., R. Fleißner, A. Wakolbinger, and A. von Haeseler. 2001. Assessing variability by joint sampling of alignments and mutation rates. J. Mol. Evol. **53**:660–669.

Miklós, I., and Z. Toroczkai. 2001. An improved model for statistical alignment. Pp. 1–10 *in* First workshop on algorithms in bioinformatics. Springer-Verlag, Berlin, Heidelberg.

Mizuguchi, K., C. M. Deane, T. L. Blundell, and J. P. Overington. 1998. HOMSTRAD: a database of protein structure alignments for homologous families. Protein Sci. **7**:2469–2471.

Pedersen, A. M., and J. L. Jensen. 2001. A dependent-rates model and an MCMC-based methodology for the maximum-likelihood analysis of sequences with overlapping reading frames. Mol. Biol. Evol. **18**:763–776.

Robinson, D. M., D. T. Jones, H. Kishino, N. Goldman, and J. L. Thorne. 2003. Protein evolution with dependence among codons due to tertiary structure. Mol. Biol. Evol. **20**:1692–1704.

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

———. 1992. Inching toward reality: an improved likelihood model of sequence evolution. J. Mol. Evol. **34**:3–16.

Manolo Gouy, Associate Editor

Accepted October 23, 2003