Statistical Alignment: Recent Progress, New Applications, and Challenges

Gerton Lunter[1], Alexei J. Drummond[1,2], István Miklós[1,3] and Jotun Hein[1]

[1] Bioinformatics group, Dept. of Statistics, Oxford Univ., Oxford, UK. {lunter,hein}@stats.ox.ac.uk
[2] Current affiliation: Dept. of Zoology, Oxford Univ., UK. alexei.drummond@zoology.oxford.ac.uk
[3] Current affiliation: Theoretical Biology and Ecology Group, Hungarian Acad. of Science and Eötvös Loránd Univ., Budapest, Hungary. miklosi@ramet.elte.hu

1.1 Introduction

Summary

Two papers by Thorne, Kishino, and Felsenstein in the early 90's provided a basis for performing alignment within a statistical framework. Here we review progress and associated challenges in the investigation of models of insertions and deletions in biological sequences stemming from this early work. In the last few years this approach to sequence analysis has experienced a renaissance and recent progress has given this methodology the potential for becoming a practical research tool. The advantages of a statistical approach to alignment include the possibility of parameter inference, hypothesis testing, and assessment of uncertainty, none of which are possible using the score-based methods that currently predominate.

Recent progress in statistical alignment includes better models, the extension of pairwise alignment algorithms to many sequences, faster algorithms, and the increased use of MCMC methods to handle practical problems. In this chapter we illustrate the statistical approach to multiple sequence alignment on a series of increasingly large data sets.

Introduction

Although bioinformatics is perceived as a new discipline, certain aspects have a long history and could be viewed as classical bioinformatics. For example, the application of string comparison algorithms to sequence alignment has a history spanning the last three decades, beginning with the pioneering paper by Needleman and Wunsch [NW70]. They used dynamic programming to maximize a similarity score based on a matching score for amino acids, and a cost function for insertion and deletions. Independently, Sankoff and Sellers in 1972 introduced an approach of comparing sequence pairs by minimizing a distance function. Their algorithm is very similar to the algorithm maximizing similarity. Sankoff and Cedergren generalized the distance minimizing approach to multiple sequences related by a phylogenetic tree. In the last three decades, these algorithms have received much attention from computer scientists and have been generalized and accelerated. Despite knowledge of exact algorithms, essentially all current multiple alignment programs rely on heuristic approximations to handle practical-sized problems. An example is the very popular Clustal family of programs. A completely different approach to alignment was introduced in 1994 by Krogh *et al.*, who used Hidden Markov Models (HMMs) to describe a family of homologous proteins. This statistical approach has proved very successful, however it was not based on an underlying model of evolution, or phylogeny.

In 1981 Smith and Waterman introduced a local similarity algorithm for finding homologous DNA subsequences that has so far remained the gold standard for the local alignment problem. The main use of local alignment algorithms is to search databases, and in this context the Smith-Waterman algorithm has proved too slow. A series of computational accelerations have been proposed, with the BLAST family of programs being the *de facto* standard in this context [AGM$^+$90].

At the same time that score-based methods were being developed for sequence alignment, parsimony methods were being used to solve the problem of phylogenetic reconstruction. The method of parsimony, which finds the minimal number of evolutionary events that explain the data, can be viewed as a special case of score-based methods. Over the last two decades the parsimony method of phylogenetic reconstruction has been criticized, and it has essentially been replaced by methods based on stochastic modelling of nucleotide, codon or amino acid evolution. This probabilistic treatment of evolutionary processes is based on explicit models of evolution, and thus give rise to meaningful parameters. In addition, these parameters can be estimated by maximum likelihood or Bayesian techniques, and the uncertainty in these estimates can be readily assessed. This is in contrast to score-based methods, where the weight or cost parameters cannot be easily estimated, or necessarily even interpreted. Because this probabilistic treatment of phylogenetic evolution is based on explicit models, it also allows for hypothesis testing and model comparison.

Despite the increased statistical awareness of the biological community in the case of phylogenetic inference, which is now fundamentally viewed as a statistical inference problem [Fel01], the corresponding problem of alignment has not undergone the same transformation, and score-based methods still predominate in this field. However, recent theoretical advances have opened up the possibility of a similar, statistical treatment of the alignment inference problem. A pioneering paper by Thorne, Kishino and Felsenstein from 1991 proposed a time-reversible Markov model for insertions and deletions (termed the TKF91 model), that allowed a proper statistical analysis for two sequences. This model provides methods for obtaining pairwise maximum likelihood sequence alignments, and estimates of the evolutionary distance between two sequences. The model can also be used to define a test of homology which is not predicated on a particular alignment of the sequences. At present, this is a test of global similarity, and although analogues of local alignment methods are possible, they have not yet been developed in the statistical alignment framework.

The recent extension of the TKF91 model to multiple sequences, and algorithmic improvements to the analysis of this model, have considerably increased the practical applicability of the model. Along with the evolutionary processes of insertion, deletion and mutation, analyzing multiple sequences additionally requires the consideration of their phylogeny. Most current alignment programs treat alignment and phylogeny separately, whereas in fact they are interdependent. A more principled approach is to estimate both simultaneously, see e.g. [Hei90, vHV97]. In this chapter we show some preliminary results on the co-estimation of phylogeny and alignment under the TKF models of evolution. For up to about four sequences, a full probabilistic treatment is feasible (see Sec. 1.4). For larger data sets, it is necessary to use approximative methods such as MCMC (see Sec. 1.5).

In conclusion, the statistical alignment framework enables a coherent probabilisitic treatment of both the sequence alignment and phylogenetic inference problems. However, challenges still remain, especially with respect to the computational problems inherent in using larger data sets, and the biological realism of the evolutionary models. In this chapter we shall review the basic model in some detail, and sketch out some recent developments and current directions of research.

1.2 The basic model

The pioneering paper by Thorne, Kishino and Felsenstein [TKF91] proposed a continuous-time evolutionary model (TKF91) for sequence insertions and deletions, as well as substitutions, that allowed a proper statistical analysis of the alignment of two sequences. This model treats insertions and deletions (indels) as single-nucleotide events, and is arguably the simplest possible continuous-time model for sequence evolution in the presence of nucleotide insertions and deletions. A major advantage of the model is that it can be treated analytically, and in fact it can be reformulated as a Hidden Markov model (HMM). This leads to alignment procedures that, using the standard HMM algorithms, are as fast as score-based approaches.

In this section we describe the TKF91 model and sketch the derivation of the transition probabilities. We introduce the extension of TKF91, termed TKF92 [TKF92], which is able to deal with artibrary-length non-overlapping indels, and which can be viewed as the statistical analogue of "affine gap penalties" in the score-based setting. Finally, we introduce the "long indel" model,

(a) ∼T∼A∼T∼A∼A∼A∼A∼A∼G∼G∼G∼ **(b)**

−	A	T	−	−	A	A	C
G	A	T	C	C	−	−	G

Fig. 1.1. (a) In the TKF91 model, a sequence is viewed as nucleotides separated by *links* (∼). Deletions originate from nucleotides, while insertions originate from links. The leftmost link is never deleted, and is called the *immortal link*. **(b)** Example of a 5-nucleotide sequence that evolved into a 6-nucleotide sequence through a series of indel and substitution events. The evolutionary outcome is summarized by an alignment showing that 3 of the ancestral nucleotides (top line) share homology with descendant nucleotides, while other nucleotides have been either deleted or inserted.

a stochastic indel process that allows for overlapping indels of arbitrary length, and discuss some approaches that approximate this process.

1.2.1 The TKF91 model

In the TKF91 model, a nucleotide sequence is modelled as a finite string of nucleotides or *letters*, separated by *links*. The string both starts and ends with a link, so that there is always one more link than there are nucleotides, see Fig. 1.1a. The insertion and deletion events are modelled as time-continuous Markov processes. Insertions of single letters originate from *links*, and occur at a rate of λ per unit of time and per link. Deletions, also of a single letter at a time, originate from the *letters*, and occur at a rate μ per unit of time per letter. Models like these are known as *birth-death processes*. We may view the sequence as consisting of a single link, followed by letter-link pairs that get inserted and deleted as little modules. In this view, the leftmost link is never deleted, and is called the *immortal link*. This immortal link ensures that the empty sequence is not a sink for the process.

Parallel to this birth-death process, the individual nucleotides are subject to a continuous-time substitution process. The original paper used Felsenstein's one-parameter model [Fel81], but this can be generalized to other models without difficulty. Similarly, in case alignments of proteins are desired, a substitution model on the amino acid alphabet is used.

Birth-death processes in which only singlet birth and deaths occur, of which the TKF91 model is an example, are automatically time-reversible by virtue of the state graph's linear topology. This fact considerably simplifies calculations. Saying that a model is time-reversible is equivalent to saying that the *detailed balance condition* holds, and this can be used to work out the equilibrium length distribution. Suppose that, at equilibrium, the probability of observing a sequence of length k is q_k. The transition rate from a length-k sequence to one of length $k-1$ is μk, since each individual nucleotide contributes a deletion rate μ. Since a sequence of length $k-1$ has k links, the transition rate in the other direction is similarly λk. Detailed balance now requires that

$$\mu k q_k = \lambda k q_{k-1} \qquad \Leftrightarrow \qquad \frac{q_k}{q_{k-1}} = \frac{\lambda}{\mu}. \tag{1.1}$$

Since the q_k are probabilities, $\sum_{k=0}^{\infty} q_k = 1$, and we have

$$q_k = \left(1 - \frac{\lambda}{\mu}\right)\left(\frac{\lambda}{\mu}\right)^k. \tag{1.2}$$

This means that $\lambda < \mu$ is a requirement to have an equilibrium length distribution. This is not surprising, since otherwise the birth rate of a length-k sequence, $\lambda(k+1)$ (there are $k+1$ links), always exceeds the death rate μk, so that sequences would tend to grow indefinitely.

Now suppose we let the TKF91 process act on a given initial sequence. After time t, the process will have resulted in a descendant sequence through a series of insertion, deletion and substitution events. Some nucleotides will have survived (though they may have undergone substitutions), and others will have been deleted or inserted. The latter will not be homologous to any nucleotide in the other sequence. This outcome can be summarized by an alignment of the ancestral and descendant sequences, where the homologous nucleotides are aligned in columns (see Fig. 1.1b).

(a)

$t=0:$	$\star$	$-$	$\#$	$\#$	$-$	$\#$	$\#$	$-$	$\#$
$t=\tau:$	$\star$	$\#$	$\#$	$\#$	$\#$	$-$	$-$	$\#$	$\#$
Probability:	I_τ	B_τ	H_τ	H_τ	B_τ	E_τ	N_τ		H_τ

(b)

$-$	$\#$	$\#$	$-$	$-$	$\#$	$\#$	$\#$
$\#$	$\#$	$\#$	$\#$	$\#$	$-$	$-$	$\#$

(c)

$t=-\infty:$	$\star$						$\#$	$\#$	$\#$	$\cdots$
$t=0:$	$\star$	$\#$	$\#$	$\#$	$\#$	$\#$	$-$	$-$	$-$	
Probability:	I_∞	B_∞	B_∞	B_∞	B_∞	B_∞	E_∞	E_∞	E_∞	$\cdots$

Fig. 1.2. Example of an evolutionary history for two sequences, and the associated probability according to the TKF91 model. **(a)** Example history for five nucleotides evolving into a length-6 sequence. Note that the event N_τ, where a nucleotide dies but not before giving birth to a new, non-homologous nucleotide, is represented by two columns in an alignment. Conditional on the ancestral sequence, the probability for this history is $I_\tau B_\tau^2 H_\tau^2 N_\tau E_\tau$. **(b)** The alignment resulting from this history. Since alignments summarize only the homology relationships between sequences, certain columns can be swapped without altering the meaning of the alignment (and different evolutionary histories may give rise to the same alignment). **(c)** Summary of the probabilities for a length-five sequence at equilibrium (that is, after an infinitely long time). The last columns are added for illustration; the ancestral sequence at $t=-\infty$ is unknown, but this makes no difference since $E_\infty = 1$ (the probability of a nucleotide being deleted tends to 1 if we wait long enough). Therefore, the probability of observing a length-5 sequence at equilibrium is $I_\infty B_\infty^5 = (1-\lambda/\mu)\,(\lambda/\mu)^5 = q_5$.

Because all nucleotides evolve independently, the probability of a particular outcome at time t, conditioned on the ancestral sequence, can be calculated by simply multiplying the probabilities of the outcomes of the individual nucleotides. For a given nucleotide, there are two sets of possible outcomes we want to distinguish, namely those where the ancestral nucleotide survives, and those where it is deleted. To complete the description, we also need the probabilities for births emanating from the immortal link:

Outcome:		Probability:
$\#\ -\ \cdots\ -$ $\#\ \#\ \cdots\ \#$	(Homologous nucleotide survives, with $n-1$ new ones)	$p_n^H(t) \quad (n=1,2,\ldots)$
$\#\ -\ \cdots\ -$ $-\ \#\ \cdots\ \#$	(Ancestor was deleted, leaving n new nucleotides)	$p_n^N(t) \quad (n=0,1,\ldots)$
$\star\ -\ \cdots\ -$ $\star\ \#\ \cdots\ \#$	(Immortal link gives rise to n new nucleotides)	$p_n^I(t) \quad (n=0,1,\ldots)$

Here # denotes a nucleotide, and we adopt the usual convention that nucleotides appearing in a column are homologous, with the ancestor appearing above the descendant. We do not explicitly write the links, except the immortal link which is denoted by a $\star$. It is now possible to set up differential equations, known as *Kolmogorov's forward equations*, for the time-dependent outcome probabilities, by considering the rate at which a state is populated from other states, and the rate at which it populates other states. For instance, the equations for $p_n^I(t)$ are

$$\frac{\mathrm{d}}{\mathrm{d}t}p_n^I(t) = (n+1)\mu p_{n+1}^I + n\lambda p_{n-1}^I - [n\mu + (n+1)\lambda]\, p_n^I(t), \tag{1.3}$$

$$p_n^I(0) = 1 \ \text{ for } n=0, \quad 0 \text{ otherwise}, \tag{1.4}$$

where p_{-1}^I is defined to be 0. These equations for a classic birth-death process are solved by

$$p_n^I(t) = (1-\lambda\beta(t))\,[\lambda\beta(t)]^n, \qquad \text{where} \quad \beta(t) = \frac{1-e^{(\lambda-\mu)t}}{\mu - \lambda e^{(\lambda-\mu)t}} \tag{1.5}$$

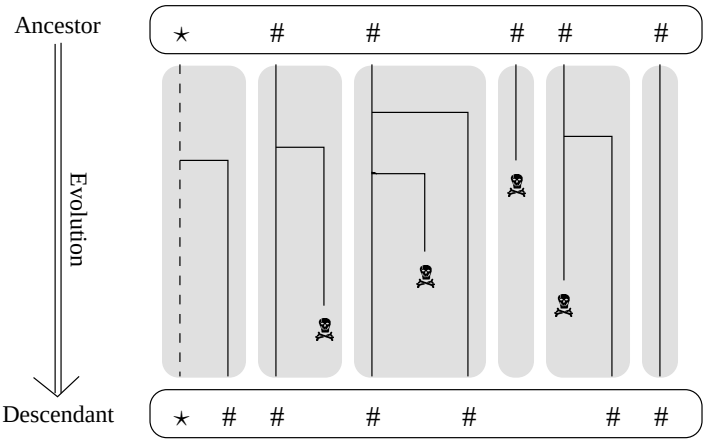


Fig. 1.3. One possible evolution of a sequence under the TKF91 model, resulting in the outcome represented in Fig. 1.2a, and summarized by the alignment of Fig. 1.2b. In this example the immortal link ($\star$) gave birth to a new nucleotide that survived; its neighbouring ancestral nucleotide gave rise to a new nucleotide that did not survive; and so on. Note that this detailed evolution contains far more information than the outcome as depicted in Fig. 1.2a (and far more than we can observe). The associated outcome probability includes contributions of all possible evolutions compatible with the outcome.

The differential equations for the other probabilities are more involved, but can also be solved analytically [TKF91]. In terms of the following abbreviations,

$$\begin{aligned} &B_\tau = \lambda\beta(\tau), && E_\tau = \mu\beta(\tau), && N_\tau = (1 - e^{-\mu\tau} - \mu\beta(\tau))(1 - \lambda\beta(\tau)), \\ &H_\tau = e^{-\mu\tau}(1 - \lambda\beta(\tau)), && I_\tau = 1 - \lambda\beta(\tau), \end{aligned} \tag{1.6}$$

the solutions are:

$$p_0^N(t) = E_\tau, \tag{1.7}$$
$$p_n^N(t) = N_\tau B_\tau^{n-1}, \qquad (n > 0) \tag{1.8}$$
$$p_n^H(t) = H_\tau B_\tau^{n-1}, \qquad (n > 0) \tag{1.9}$$
$$p_n^I(t) = I_\tau B_\tau^n. \tag{1.10}$$

See figure 1.2 for an example of how to calculate the probability of a particular evolutionary history.

1.2.2 The TKF92 model

The most obvious drawback of the TKF91 model, as already noted in the original paper, is that insertions and deletions occur one letter at a time. In reality, many indel events involve more than a single nucleotide. In 1992, Thorne, Kishino and Felsenstein introduced an improved version of their model, designed to model indel events of more than a single letter [TKF92]. This model, referred to as TKF92, differs from the TKF91 model by acting on sequence *fragments* instead of single nucleotides. The fragments themselves are not observed, and their length is randomly distributed according to a geometric distribution, with parameter ρ. This approach leads to a model that can still be treated analytically, and is a reasonably good approximation of the actual observed indel length distribution.

The approximation that is made in the model, and that makes it possible to analytically compute the probabilities, is that the fragments (and their size) are supposed to stay fixed over the entire evolutionary history of the sequence. This assumption, made for technical reasons, is clearly not realistic. However, things are not as bad as they might seem. In the same way that the TKF91 model sums over all possible alignments, the TKF92 model also sums over all possible *fragment* assignments. Effectively, this means that indels of any length may occur at any position in the sequence, but such indels may not, in the course of evolutionary history, overlap. See Fig. 1.8 for an alignment under the TKF91 and TKF92 models.

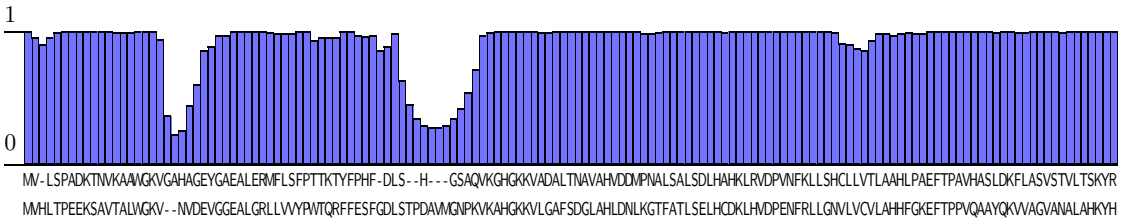


Fig. 1.4. The most likely pairwise alignment of human α and β hemoglobin, according to the TKF91 model. The vertical bars indicate posterior column probabilities, i.e. the proportion of alignments that include that particular column (weighted according to the posterior probability under the model). See Sec. 1.3.3 for the algorithms used to calculate the alignment and posteriors. The log likelihood of observing this alignment under the TKF91 model, using Maximum Likelihood parameters for these sequences ($\lambda = 0.03718$, $\mu = 0.03744$, $t = 0.91618$, see [HWK$^+$00]), is -735.859. This low likelihood reflects the relatively high sequence divergence, and the fact that it is very unlikely for the ancestor of these sequences to have evolved by chance; however, the log likelihood of observing both sequences by chance independently is far smaller still, $-401.372 - 418.764 = -820.136$, giving strong support to the hypothesis that these sequences are homologous.

(a)

$t = 0$:	$\star$	#	#	#	−	−	−	−	#	#	#	#	#	#
$t = \tau$:	$\star$	#	#	#	#	#	#	#	#	#	#	#	#	#
Probability:	I_τ	$H_\tau\rho^2(1-\rho)$			$B_\tau\rho^3(1-\rho)$				$H_\tau\rho^3(1-\rho)$				$H_\tau\rho(1-\rho)$	

(b)

#	#	#	−	−	−	−	#	#	#	#	#	#
#	#	#	#	#	#	#	#	#	#	#	#	#

Fig. 1.5. An evolutionary history according to the TKF92 model. **(a)** One possible fragmentation into fragments of size 3, 4, 4 and 2 respectively, and the associated probability for this evolutionary history. **(b)** The alignment resulting from this history. Many different fragmentations contribute to this alignment.

1.2.3 Parameters of the TKF models

Although the TKF91 model has two parameters, λ and μ, their ratio is in practice fixed by the sequence length. Indeed, if we maximize the likelihood

$$q_L = \left(1 - \frac{\lambda}{\mu}\right)\left(\frac{\lambda}{\mu}\right)^L \tag{1.11}$$

in terms of λ/μ, for a fixed sequence length L, we find that the maximum is obtained for

$$\frac{\lambda}{\mu} = \frac{L}{L+1}. \tag{1.12}$$

For maximum likelihood parameter estimates, it is therefore not meaningful to estimate λ and μ independently, but rather to fix their ratio based on the average of the sequence lengths that are to be aligned, and estimate just one free parameter.

The TKF92 model has one extra parameter, ρ, parametrizing the geometric fragment length distribution. Fragments drawn from this distribution have an expected length of $\frac{1}{1-\rho}$. In the TKF92 model, the parameters λ and μ refer to the indel rate *per fragment*. To allow a meaningful comparison it is useful to introduce new parameters λ' and μ' that specify the average indel rates *per site*, and which are related the parameters λ and μ by

$$\lambda = (1-\rho)\lambda', \qquad \mu = (1-\rho)\mu'. \tag{1.13}$$

Note that TKF91 is a special case of TKF92, obtained by setting $\rho = 0$. This corresponds to a degenerate fragment length distribution where all fragments have length 1. For an example how to calculate the likelihood given a fragmentation, see Figure 1.5.

1.2.4 The "Long Indel" model

The TKF92 model is a substantial improvement over the TKF91 model, as it allows indel events involving more than one nucleotide. The main assumptions that go into the model are (1) that indel events do not overlap, and (2) that the indel lengths are geometrically distributed. A natural, more general evolutionary model would relax these two assumptions, specifically, by allowing indel events to overlap, and by allowing an arbitrary indel length distribution. Here we focus on relaxing the former assumption, although the proper modelling of the actual indel length distribution (see e.g. [QG01]) is probably at least as important for alignment accuracy. We refer to the more general model as the "long indel" model. In its general form, no closed-form solution of the outcome probabilities are known, even for a geometric indel length distribution. The main difficulty is that by allowing overlapping indel events, the fates of neighbouring nucleotides become entangled over time, so that the probility of the total outcome does not factorize into individual nucleotide outcome probabilities, as is the case for the TKF models.

To arrive at a tractable implementation of this model, some kind of approximation is necessary. Knudsen and Miyamoto [KM03] develop an approximation that is analytically no more complex than the TKF models: their pairwise alignment algorithm takes $O(L^2)$ time, where L is the sequence length. In fact their model is formulated as an HMM with the same topology as the TKF models are commonly formulated in, and differs only in the transition probabilities. It is satisfying that, in contrast to TKF92, this indel model is derived from first principles, but given its similar structure, it is unclear how much it improves upon TKF92.

If one is willing to use computationally more demanding algorithms, then an even more realistic approximation to the long indel model is possible. In [MLH04] an approximation is used that allows each indel event to overlap with up to two others, and allows an arbitrary indel length distribution to be used. The corresponding pairwise alignment algorithm has time complexity $O(L^4)$, making the algorithm unsuitable for e.g. large database searches. However, single pairwise alignments can still be computed relatively quickly, and on a set of trusted alignments based on known 3D protein structure, this model outperformed TKF92. See [MLH04] for more details.

1.3 Pairwise alignment

In this section we describe how the TKF models are used in practical pairwise sequence alignment algorithms. First we describe an intuitive dynamic programming recursion, which however has a high computational complexity. More efficient recursion exist, and we describe in detail one that is based on the formulation of the TKF models in terms of hidden Markov models. The additional structure makes it easier to describe the various algorithms that are based on it, and pave the way for the multiple alignment algorithms later on.

1.3.1 Recursions for the likelihood of two homologous sequences

Let us now turn to the task of calculating the likelihood of homology, that is, the likelihood that two sequences have evolved from a common ancestor. Because of the time-reversibility of the TKF91 model, this is equilvalent to the likelihood that one sequence evolved into the other, in twice the time that separates the ancestor from the two descendants (below referred to as τ). This likelihood is, by definition, the total probability corresponding to all evolutionary histories that are consistent with the observed sequences. Obviously, there are extremely many of these evolutionary histories, so a direct evaluation of this sum is impractical. However, a dynamic programming approach is possible that computes this sum in reasonable time.

In the following, $P(i, j)$ is the likelihood of the length-i prefix of the ancestral sequence evolving into the length-j prefix of the descendant sequence. For instance, $P(0, 0) = I_\infty I_\tau$, since the probability of observing the empty ancestral sequence is I_∞, while the probability of the empty sequence evolving, in time τ, into the empty sequence again is I_τ. The dynamic programming solution now consists of computing $P(i, j)$ in terms of previously computed $P(i', j')$. By filling a table, all values can then be computed in reasonable time.

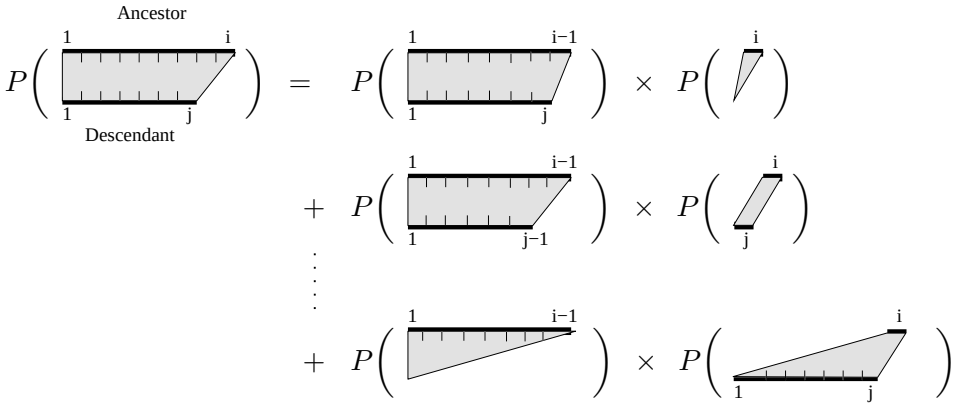


Fig. 1.6. A graphical illustration of a dynamic programming recursion for the TKF91 likelihoods. The left-hand side is the likelihood $P(i, j)$ defined in the text, horizontal black bars represent sequences, and the grey area represents "evolution". Since individual nucleotides evolve independently, the likelihood for the length-i ancestral prefix to evolve into the length-j descendant prefix factorizes into several other prefix likelihoods, and probabilities of the ith residue to evolve into descendant subsequences.

As we saw in Sec. 1.2.1, each ancestral nucleotide evolves independently of the others, and a single ancestral nucleotide can evolve into $0, 1, 2, \ldots$ descendant nucleotides (which may or may not be homologous to the ancestor). For the recursion, this means that we can express $P(i, j)$ in terms of $P(i-1, j-k)$ with $k = 0, 1, \ldots, j$ (corresponding to outcomes with k descendant nucleotides), multiplied by the probability of a particular evolution of the last ancestral nucleotide. The contribution of the indel process to these probabilities is given in Sec. 1.2.1. This must be multiplied by (1) the probability B_∞ of observing one additional ancestral nucleotide; (2) the equilibrium probability of the particular nucleotide observed; (3) the probability of that nucleotide evolving into the descendant nucleotide (in case of a homologous descendant nucleotide); and (4) the nucleotide equilibrium probabilities of any nonhomologous descendant nucleotides. The resulting algorithm is illustrated in Fig. 1.6.

As the algorithm is formulated here, its running time is cubic in the sequence length. However, due to the geometric tails of the outcome probabilities p_n^N, p_n^H and p_n^I as functions of n, the recursion may be reformulated so that it only uses a bounded lookback, resulting in an algorithm that has quadratic time complexity [TKF91]. In this context, this is just an algebraic trick, and is reminiscent of the method used by Gotoh in 1982 for reducing the time complexity of a score-based sequence alignment algorithm with affine gap penalties (see [Got82]). However, there is a more meaningful and conceptual way to look at this. It turns out that the TFK91 model can be viewed as an instance of what is known as a hidden Markov model (HMM). From that point of view, the algorithm derived using the algebraic trick becomes the well-known forward algorithm for HMMs, and more algorithms are immediately applicable, such as the Viterbi algorithm for determining the most likely path through the chain, corresponding to the most likely alignment supported by the model. In the next section we will develop this point of view in more detail.

1.3.2 A hidden Markov model formulation of TKF models

The outcome probabilities $p_n^H(t)$ and $p_n^N(t)$ of the TKF91 model are geometric functions of n. As a result, we can construct a graph, with probabilities on each of its edges, such that each path through the graph corresponds uniquely to a particular outcome, and the product of all the probabilities encountered is precisely the probability of that outcome. The graph shown in Figure 1.7a has all these properties, and generates alignments according to the TKF91 model. Such graphs are called *Markov models* if the outgoing probabilities sum to 1, for each of the states. This is accomplished in Figure 1.7 by multiplying and dividing by a factor $1 - B_\tau$ at certain positions, in such a way that the total probability of any closed path from the start state to the end state does not change. We

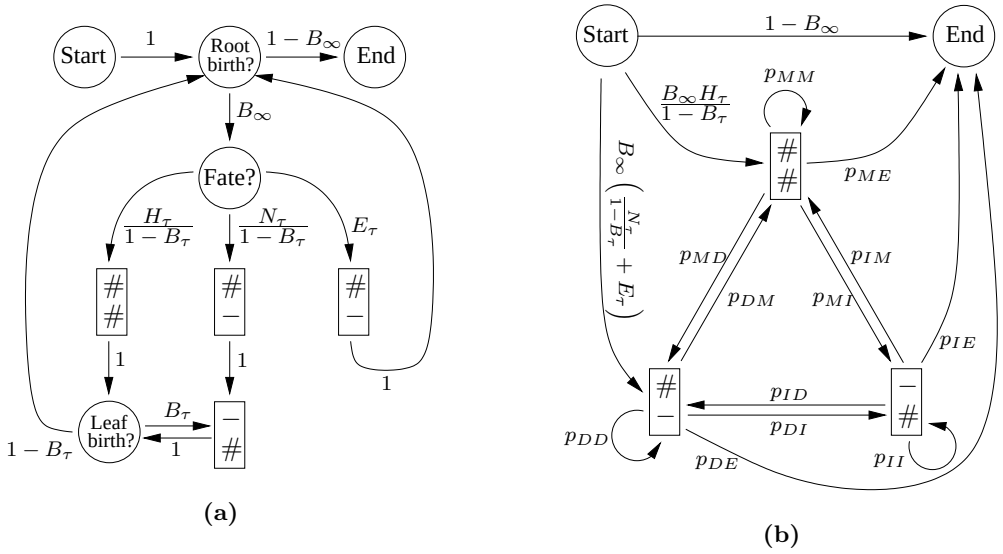


Fig. 1.7. Two HMM formulations of the TKF91 model. **(a)** Direct translation of TKF91 probabilities into an HMM. A factor $1-B_\tau$ is needed for three transitions, to make all outgoing transition probabilities add up to 1. **(b)** Another HMM that is emission-equivalent to (a). The two states $\begin{smallmatrix}\# \\ - \end{smallmatrix}$ were merged into one, and all non-emitting states removed, leaving a fully-connected three-state HMM. Note that the evolutionary indel model by Knudsen and Myamoto [KM03] is formulated using an HMM with exactly the same topology.

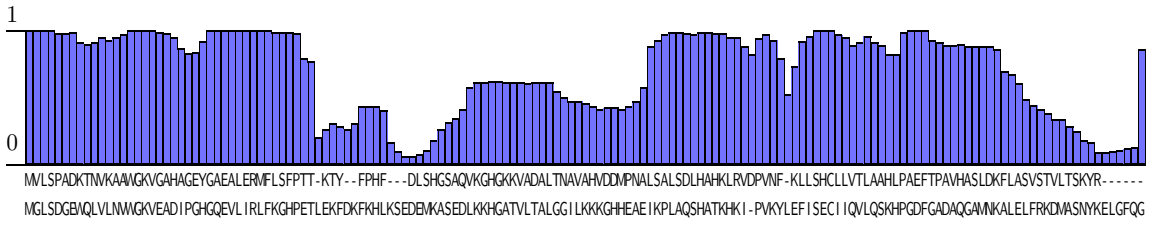
use this Markov model as a *hidden* Markov model (HMM), because we treat only the sequences as known, while the alignment structure is regarded as unknown. This unknown information is encoded by the path taken through the Markov chain, while the emitted sequences of nucleotides are given. We refer to [DEKM98] for more information about HMMs.

By manipulating the graph of Fig. 1.7a, the number of states can be reduced to just 3 (apart from the Start and End states), see Fig. 1.7b. This reduces the time and memory complexity of the HMM algorithms (especially when more sequences are considered, see below). Because of the algebraic manipulations, the transition probabilities take a more complicated form, and are listed in Table 1.1. Henceforth, when we refer to the TKF91 HMM, we are referring to the reduced HMM of Fig. 1.7b.

	Transition probabilities	
Edge:	TKF91	TKF92
p_{MM}	$B_\infty H_\tau$	$\rho + (1-\rho)B_\infty H_\tau$
p_{MI}	B_τ	$(1-\rho)B_\tau$
p_{MD}	$B_\infty(N_\tau + E_\tau(1-B_\tau))$	$(1-\rho)B_\infty(N_\tau + E_\tau(1-B_\tau))$
p_{ME}	$(1-B_\tau)(1-B_\infty)$	$(1-\rho)(1-B_\tau)(1-B_\infty)$
p_{II}	B_τ	$\rho + (1-\rho)B_\tau$
p_{IM}	$B_\infty H_\tau$	$(1-\rho)B_\infty H_\tau$
p_{ID}	$B_\infty(N_\tau + E_\tau(1-B_\tau))$	$(1-\rho)B_\infty(N_\tau + E_\tau(1-B_\tau))$
p_{IE}	$(1-B_\tau)(1-B_\infty)$	$(1-\rho)(1-B_\tau)(1-B_\infty)$
p_{DD}	$E_\tau B_\infty$	$\rho + (1-\rho)E_\tau B_\infty$
p_{DM}	$E_\tau H_\tau B_\infty[N_\tau + E_\tau(1-B_\tau)]^{-1}$	$(1-\rho)E_\tau H_\tau B_\infty[N_\tau + E_\tau(1-B_\tau)]^{-1}$
p_{DI}	$N_\tau[N_\tau + E_\tau(1-B_\tau)]^{-1}$	$(1-\rho)N_\tau[N_\tau + E_\tau(1-B_\tau)]^{-1}$
p_{DE}	$E_\tau(1-B_\tau)(1-B_\infty)[N_\tau + E_\tau(1-B_\tau)]^{-1}$	$(1-\rho)E_\tau(1-B_\tau)(1-B_\infty)[N_\tau + E_\tau(1-B_\tau)]^{-1}$

Table 1.1. Transition probabilities in the HMM of Fig. 1.7b, for the TKF91 and TKF92 models. The probabilities for TKF91 are obtained from those of TKF91 by multiplying all transition probabilities by $1-\rho$, and adding a self-transitions with probability ρ to every state.

(a)



(b)

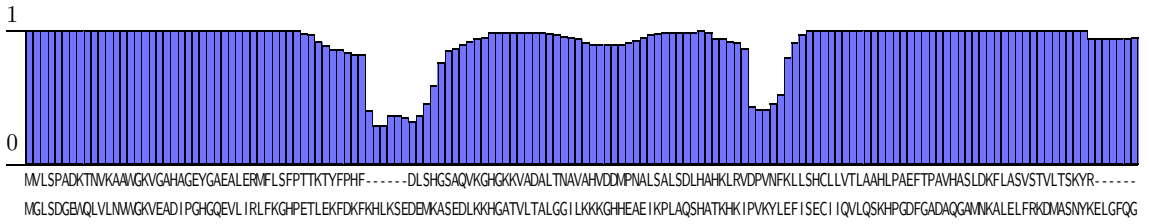


Fig. 1.8. Viterbi alignment of human α hemoglobin with human myoglobin, under **(a)** the TKF91 model, and **(b)** the TKF92 model (with parameter $\rho = 0.44$). Clearly, the TKF92 model fits the data much better, generally assigning higher posteriors to the alignment (maximum log likelihoods for TKF91 and TKF92 are -825.25 and -817.25, respectively). Note the ≈ 6 aligned columns that show a sudden decrease in posterior probability in the TKF92 alignment, where the corresponding TKF91 alignment has two small indels. The TKF92 model is reluctant to include many individual indels, preferring a single large one. Although the maximum likelihood path is the one without any indels in that region, alignments with indels contribute significantly to the total likelihood, indicating that the homology implied by the alignment there should treated with caution. This is a good example of what additional information can be obtained from the posterior column probabilities.

Starting from this HMM formulation of TKF91, it is straightforward to transform it into the HMM for TKF92. This is done by adding a self-transition (with probability ρ) to each state, which accounts for the geometric fragment length distribution. To compensate, all other transition probabilities (including the existing self transitions) are multiplied by $1-\rho$, making outgoing probabilities add up to 1 again. See Table 1.1 for the explicit transition probabilities.

1.3.3 Algorithms

The above formulation of the TKF models in terms of HMMs allows us to use standard HMM algorithms, such as the Forward-Backward algorithm, and the Viterbi algorithm. For a detailed explanation of these algorithms we refer to [DEKM98]; here we focus on their application.

Applied to the TKF HMMs, the Forward (or Backward) algorithm calculates the total likelihood of one sequence to have evolved from another. In fact, in most cases, we want to calculate the likelihood of an unknown root sequence having evolved, independently, into two observed modern sequences. Because of the time-reversibility of the model, the position of the root on the branch connecting these two sequences does not influence the likelihood, and therefore these two likelihoods are equal. This symmetry property is known as Felsenstein's Pulley Principle [Fel81].

The Forward and Backward algorithms compute the total probability of all paths through the Markov chain that emit the observed sequences. This can be used for homology testing [Hei01], and to estimate evolutionary parameters [TKF91], such as the divergence time and the indel rate, by Maximum Likelihood. The Viterbi algorithm is the HMM analogue of the Needleman-Wunsch [NW70] score-based alignment algorithm, and is traditionally the main workhorse for doing inference in hidden Markov models. The algorithm finds the most probable (that is, the Maximum Likelihood) path to emit the given sequences, and this path codifies the alignment of the sequences.

From the intermediate results from both the Forward and Backward algorithms, it is possible to compute the posterior probability of passing through any given state, conditional on emitting the

observed sequences. Figures 1.4 and 1.8 show examples of Viterbi alignments, and corresponding posterior state probabilities on the Viterbi paths. For the alignment models, these are interpreted as the posterior probability of observing an individual column in the alignment. These posteriors are therefore indicators of the local "reliability" of an alignment. They add important information to the simple "best" answer obtained for example by the Viterbi algorithm, and can be seen as the alignment equivalent of confidence intervals for simple numerical parameter estimates. In practical examples, there are very many alignments that contribute to the total likelihood, and the most likely alignment may contribute only a very small fraction. This makes a single best answer not very informative, and the local reliability measure indicates which parts of the alignment can be trusted, and which parts are essentially random, giving a quantitative underpinning of the notion of "unalignable region" [Lee01].

Although the Viterbi algorithm, computing the Maximum Likelihood path, is ubiquitously used for alignment inference, it should here be mentioned that there is no one-to-one relationship between paths through the Markov chain and alignments. More than one evolutionary history can give rise to a single alignment, see Figure 1.2 for an example. Note that for the output of the Viterbi algorithm, the exact topology of the HMM is important, and in general, two HMMs may be *emission* equivalent without being *path* equivalent. An example is provided by the TKF92 versions of the HMMs of Fig. 1.7, which are derived from those by adding self-transitions with probability ρ to each (emitting) state. Paths through Fig. 1.7a codify the sequence fragmentation, while in Fig. 1.7b the sequence fragmentation is analytically summed out, and cannot be deduced from the path. Their hidden information differs, but the observables (the emitted nucleotides) follow exactly the same distribution. The result is that the Forward or Backward algorithms give the same answers, but the Viterbi algorithm is biased toward alignments with more indels if the HMM of Fig. 1.7a is used.

Although not much of a problem for pairwise alignment, the non-equivalence of paths and alignments turns up again, and more seriously, in the case of alignments on trees. One way of dealing with this problem is to explicitly look for the most probable alignment, and keep track of all paths that contribute to it [Kro97]. Unfortunately, the resulting algorithm is very slow. Another method that recovers a "best" alignment from an HMM, without relying on path reductions, is *posterior decoding* [DEKM98]. The idea is to first compute posterior probabilities for each possible column that may appear in the alignment, and then find the alignment that maximizes the combined posterior column probability. This can be done efficiently using Dynamic Programming, which is the same strategy that underlies the Forward, Backward and Viterbi algorithms. Although there is no guarantee that the alignment obtained in this way is the most probable one, in practice this method gives very good results. Another advantage of the method is that it is also applicable in Markov chain Monte Carlo settings (see Section 1.5), where the Viterbi algorithm cannot be used, but estimates of posterior column probabilities are available.

1.4 Multiple statistical alignment

The simultaneous alignment of several sequences can reveal conserved motifs much more sensitively than a pairwise alignment can. This assists in the alignment of more distantly related sequences, and the detection of functional sites. Unfortunately, multiple alignment is a computationally hard problem, and certain particular cases are known to be NP-hard [WJ94]. Furthermore, the problems of multiple alignment and phylogenetic inference are closely interlinked: to properly align a set of homologous sequences, it is necessary to know their phylogeny, and vice versa [Hei90, vHV97]. Keeping this interrelatedness in mind, we will nonetheless focus mostly on alignments. We do not discuss the various interesting approaches developed for phylogenetic reconstruction, and will return to this topic only at the end of this section where we discuss co-estimation of alignment and phylogeny.

In the 1970s, Sankoff introduced the first multiple alignment algorithm [San75], and since then many other algorithms have been proposed. Most of these are "score-based", and use a score function that assigns a 'goodness' to particular multiple alignments (and sometimes phylogenies). The algorithms then find the best alignment by optimizing this score function. Because of the large number of possible alignments, full optimizations are practically impossible, and several clever heuristics

have been introduced to find reasonable solutions in reasonable time. Successful programs include ClustalW [TDT94], PSI-Blast [AMS+97], DiAlign [Mor99] and T-Coffee [NHH00].

A drawback of score-based approaches is that it is hard to justify the parameter settings of the score function – or indeed, the score function itself. This is one reason why probabilistic approaches are becoming more popular. Instead of assigning a score, these methods assign a probability to alignments, making it easier to train a model on data, and find parameters by techniques such as maximum likelihood. Two popular probabilistic approaches, both based on HMMs, are HMMER [Edd01] and SAM [KBH98]. Another important advantage of probabilistic models is that they provide estimates of the uncertainty in the final answer, such as posterior column probabilities for alignments, and confidence intervals for parameter estimates. An example of a probabilistic progressive multiple alignment method is [LM03], which has since been extended to include structure-dependent evolution (Löytynoja and Goldman, pers. comm.). Another example is Mitchison [Mit99], who estimates phylogeny and alignment simultaneously using an MCMC sampler, in a probabilistic framework.

However, probabilistic models also have some problems. Such models are mostly phenomenological, describing the data, but not explicitly making statements about the process that generated it. In particular, the evolutionary relationships between the sequences are often treated heuristically. Parameters of phenomenological models are linked to observables, not to the evolutionary process, making it difficult to interpret parameter values. For correct modelling, one should ideally re-estimate parameters for every dataset with different evolutionary parameters.

An evolutionary approach is based on a model of sequence evolution, from which a probabilistic model for the observed sequences is derived. In this way, the parameters of the model (such as indel and substitution rates, divergence time) are meaningful, and can be estimated using the same methods as for probabilistic models. The TKF91 and TKF92 models fit in this framework. Algorithmically, the approach is not very different from probabilistic or even score-based methods, and encounters the same problems. Full likelihood methods are possible only for very limited number of sequences, and after that, approximations and heuristics are necessary. One particularly useful approximation method is Markov Chain Monte Carlo (MCMC). These methods generate samples from the posterior distribution of alignments, thereby disregarding alignments that are very unlikely.

1.4.1 Multiple alignment and Multiple HMMs

The first step in extending statistical alignment to multiple sequences was taken by Steel & Hein [SH01], who provided an algorithm to align sequences related by a star tree (a tree with a single internal node). This was soon extended to arbitrary phylogenetic trees [Hei01], with an algorithm with time complexity $O(L^{2n})$, where L is the mean sequence length, and n the number of sequences. For star trees, this running time was subsequently reduced to $O(4^n L^n)$ [Mik02]. These results used rather complicated algebraic manipulations to derive the algorithms, and when it was realised that for two sequences, the TKF91 model can be described as a pair-HMM [MFWvH01, Hei01, HB01], the extension to multiple sequences became much easier. Holmes and Bruno [HB01] showed how to construct a multiple-HMM describing the evolution of an ancestral sequence and its two descendants. Subsequently, Hein, Jensen and Pedersen showed how to generate a multiple-HMM for TKF91 on an arbitrary phylogenetic tree [HJP03]. The details concerning the construction of these multiple HMMs are beyond the scope of this book, but to give a flavour of the techniques involved we give a single example for three sequences in Fig. 1.9.

We can loosely argue that this multiple-HMM correctly generates multiple alignments according to TKF91. First, note that each path from the start state to the end state corresponds to a multiple alignment. From the start state, the chain first jumps to a silent state next to the state emitting a character to the all sequences, which models 'births' emanating from the immortal link. Eventually the process reaches the rightmost silent state, where a decision is made whether there is a new root birth. If there is, a decision tree with transition probabilities α_i and $1-\alpha_i$ decides on which branches this nucleotide survives, after which subsequent births associated to the surviving nucleotides are introduced. It can be verified that the path probabilities equal the probabilities that the TKF91 model assigns to the corresponding alignments, a task we gladly leave to the reader.

In the same vein, TKF92 can be extended to multiple alignments on trees. The simplest way to do this is by adding self-transitions to the HMM of Fig. 1.9. This fixes fragmentations over the

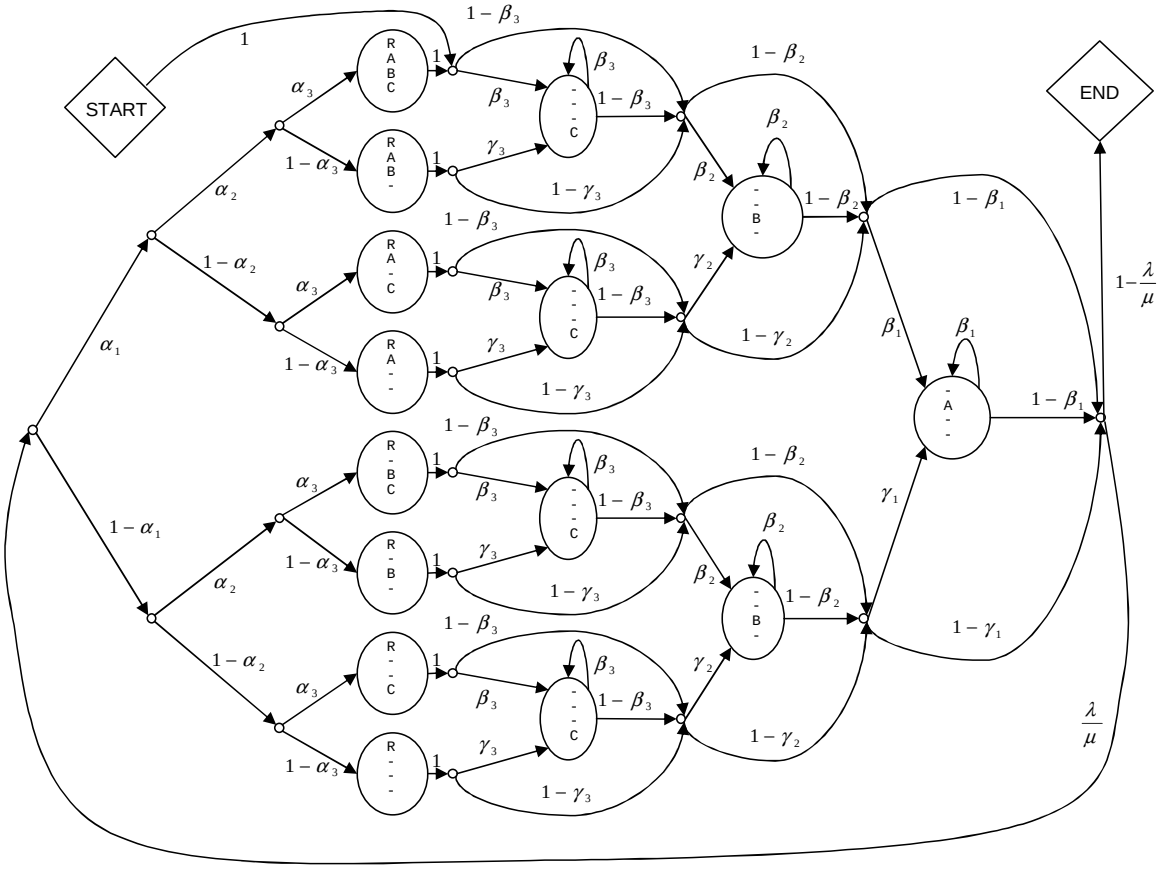


Fig. 1.9. Multiple-HMM decribing the evolution of three sequences related to a star-tree, under the TKF91 model. The following abbreviations are used: $\alpha_i = e^{-\mu t_i}$, $\beta_i = (\lambda - \lambda e^{(\lambda-\mu)t_i})/(\mu - \lambda e^{(\lambda-\mu)t_i})$, $\gamma_i = (1 - \alpha_i)^{-1}(1 - e^{-\mu t_i} - \beta_i)$, where t_i is the length of the branch descending to tip i in the phylogenetic tree. Big circles are states that emit the column shown according to the underlying substitution model; R, A, B and C represent characters in the root sequence and the three observed sequences, respectively. Small ellipses represent silent states. (See [HB01])

entire phylogenetic tree, so that indels cannot overlap even if they occur on separate branches, clearly creating undesirable correlations between independent subtrees. A better behaviour is obtained if the three-state TKF92 HMM is used as building block on each of the branches, and communicate sequences (not fragmentations) at internal nodes. Holmes introduced the concept of *transducers*, or conditionally normalized pair HMMs describing the evolution along a branch, which provides an algorithmic way to construct multiple-HMMs on a tree [Hol03]. This leads to an HMM with the same number of states as before, but one that allows overlapping indels as long as they occur on separate branches.

As an aside, note that, in contrast to the fixed-fragmentation TKF92 model, likelihoods now depends on the number and position of internal nodes along a branch. In fact, even introducing a node of degree 2 (i.e. a node with one incoming and one outgoing branch) changes the model. By increasing the density of such degree-2 nodes, the model eventually converges to the Long Indel model, allowing arbitrary overlapping indels. Unfortunately, the number of HMM states increases exponentially with the number of nodes, so that adding such degree-2 nodes is an impractical way of approximating the Long Indel model.

A technical problem with the above generated multiple-HMMs is that they may contain *silent states* that do not emit any characters, or emit only into (unobserved) internal nodes. (An example of a silent state is the `R/-/-/-` state in Fig. 1.9.) These states create self-references (or loops) in the

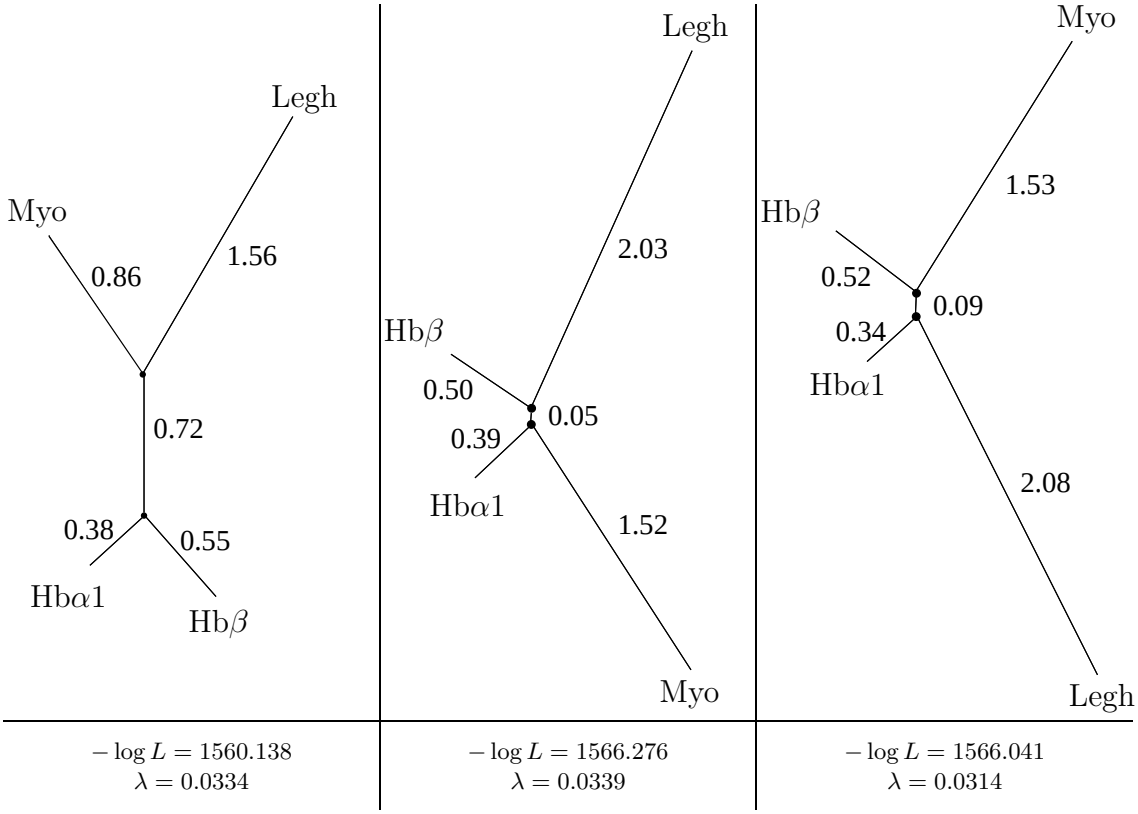


Fig. 1.10. Maximum likelihood trees relating human $\alpha 1$ and β hemoglobins, myoglobin, and bean leghemoglobin, for all three topologically distinct trees, total likelihood values (L) and insertion rates (λ), under the TKF91 model. The numbers next to the branches refer to branch lengths in units of expected number of substitutions per site. As substitution rate matrix Dayhoff's PAM matrix was used. As expected, the most likely tree is the one that groups human alpha and beta hemoglobin together. The other trees are close to degenerate, with only a very short segment connecting the internal nodes, again suggesting that these phylogenies are incorrect. The tree likelihoods combine all possible alignments of the four sequences, in contrast to most other methods that rely on a single alignment, preventing inaccuracies in a single alignment to bias the phylogeny inference (see [LMSH03]).

state graph, and need to be eliminated before the Markov chains can be used in algorithms. The technique of silent state elimination is well-known in the HMM literature [Edd01], and it involves solving a set of linear equations. See [LMSH03] for more details.

1.4.2 Algorithms for multiple sequence alignment

After eliminating silent states, we can calculate the joint probability of a set of sequences related by a phylogeny by the standard Forward and Backward algorithms, calculate the posterior probability of particular alignment columns, and we can find the most likely alignment with the Viterbi algorithm [DEKM98]. An example is presented in Figure 1.11.

A practical problem that besets algorithms for multiple-HMMs, is that the time and memory complexity increases rapidly with the number of sequences. Two factors contribute to this rapid increase: (1) the dimension of the dynamic progamming (DP) table is equal to the number of sequences n, and (2) the number of states S of the multiple HMM itself grows exponentially with the number of sequences. Generally, the basic algorithms have a time and memory complexity of $O(S^2L^n)$ and $O(SL^n)$ respectively. For TKF92 and TKF92, the number of states S is of the order $\sqrt{5}^n$ [LMSH03]. One implementation of the TKF91 model for 4 sequences uses 47 states, and 1293 transitions between them, so that for sequences of length 150, naive implementations would require about $2 \cdot 10^{10}$ memory positions and 10^{12} floating point operations.

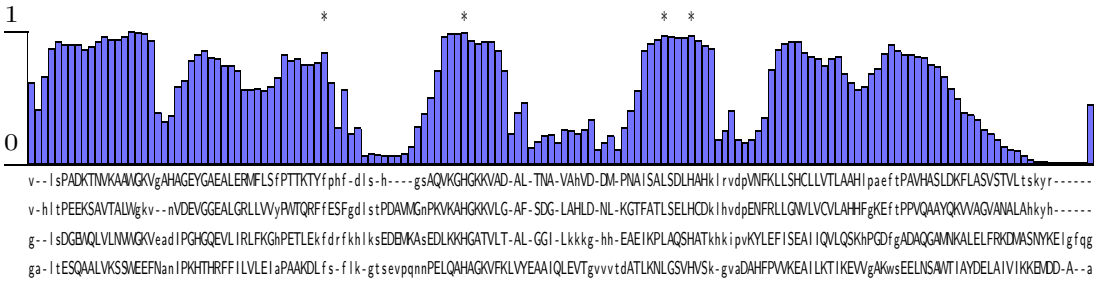


Fig. 1.11. The Viterbi alignment of α and β human hemoglobin, human myoglobin, and leghemoglobin (Lupinus Luteum) for the first phylogenetic tree in Figure 1.10. The log-likelihood of this alignment (one of those included in the tree likelihood of Fig. 1.10) is -1593.223. The column posteriors vary considerably, and clearly point to several highly conserved domains, punctuated by much less conserved regions. Amino acids that participate in α helices are shown in upper case; asterisks denote the four conserved residues that coordinate the heme group.

The TKF91 model has some surprising symmetries that allow the Forward algorithm based on the three-state pair HMM to be reduced to a one-state recursion [HWK$^+$00]. This algebraic reduction results in a recursion that contains negative coefficients, so that it cannot be interpreted as a Markov chain anymore. Nevertheless, similar reductions are possible on trees, also resulting in a one-state recursion, resulting in an algorithm to compute the total likelihood using L^n memory positions, with a running time of order $O(2^n L^n)$. See [LMSH03] for details.

The tricks involved in the reduction seem unique to TKF91, and for TKF92 and similar models, we have to resort to general algorithms. In the following section, we discuss a number of modifications to the original Forward-Backward and Viterbi algorithms, and some corner cutting methods, that make full likelihood methods possible in practice.

Multiple Forward-Backward

In practice, memory resources are often the limiting factor, and strategies to reduce memory usage are therefore of great practical importance. For the Forward and Backward algorithms, if only the total likelihood is required, one can relinquish DP table entries dynamically, resulting in memory requirements of the order SL^{n-1}, not SL^n. To compute posterior column probabilities, the straightforward algorithm computes the full DP table using both the Forward and Backward algorithm, and therefore requires order SL^n memory. However, if the posteriors for a particular alignment are required, a more careful implementation can still compute this using order SL^{n-1} memory, by relinquishing during the iterations all DP table entries that are not referenced by either the alignment of interest or new DP table entries [Hir75, MD02]. Even with these leaner implementations, the computational complexity is still very high. Further reduction in space can be achieved by heuristic *corner cutting* methods. Such methods are well-known in score-based alignment approaches [Ukk85, Mye86, CL88, LAK89, WMMM90], and we here describe their counterparts for HMMs.

In practice, only a small region of the DP table is responsible for the dominant contribution to the total likelihood. This region often consists of a well-defined "spine" corresponding to the Maximum Likelihood alignment, and close neighbours. If this "contributing region" were known, the recursion could be confined to it, resulting in a considerable speed-up [HWK$^+$00] and a negligible loss of total likelihood. The problem is clearly circular however, as the contributing region can only be determined after the full DP table has been computed. Having said this, heuristic methods for selecting the contributing region exist that work very well in practice, for example based on full pairwise alignments.

The likelihood that is computed in this way is, by construction, a lower bound for the actual likelihood, since each time the DP recursion refers outside the contributing region, probability 0 is used instead of the true (small but nonzero) probability. It is possible to also compute an upper bound, using the same contributing region. This sandwiches the actual likelihood between two bounds

and provides, if these bounds are tight enough, an effective a-posteriori proof that the Maximum Likelihood alignment lies within the contributing region. The method is based on calculating the alignment likelihood of m known sequences and $n-m$ sequences of unknown composition and length, on an n-leaved phylogenetic tree. A recursion of memory complexity SL^m exists that computes the sum of alignment probabilities over all possible alignments of these sequences, where for a particular alignment, this probability is maximized over the sequence composition of the unknown sequences. This obviously gives an upper bound for the total alignment likelihood of the n sequences, and one which is considerably better than the likelihood of simply aligning the m known sequences on an m-leaved tree. Moreover, it gives upper bounds for each of the DP table entries in the n-dimensional table, by projecting to the smaller m-dimensional table. By taking the minimum over all combinations of m sequences out of the n given ones, good upper bounds are obtained for the entire DP table. The final upper bound for the alignment probability is obtained by performing the DP recursion on the contributing region, and using the m-sequence based upper bound whenever the recursion refers to an entry outside that region. This approach was used to compute the alignment likelihoods and Maximum Likelihood trees depicted in Fig. 1.10.

Multiple Viterbi

The method of "shaving off" a dimension of the DP table in the Forward-Backward algorithm cannot be used for the Viterbi algorithm, as it contains a backtracking loop to find the most likely path, which may visit any part of the DP table. A clever idea due to Hirschberg [Hir75] reduces the memory requirements to order SL^{n-1} in a different way, at the cost of an increase in time complexity by only a constant factor. The algorithm consists of a standard Viterbi that however does not retain its DP table, and stops halfway. A "backward"-Viterbi algorithm then starts at the other end, and again stops halfway. Using their outputs, the central state of the Viterbi path is determined, but no backtracking is possible. However, with the central state known, the Viterbi recursion can be performed again but now constrained to two DP tables of size roughly $(L/2)^n$. The same strategy is used again in the smaller tables, until after several recursive divisions the full Viterbi path is found. The algorithm runs in time proportional to

$$S^2L^n \times \left[1 + \left(\frac{1}{2}\right)^{n-1} + \left(\frac{1}{2}\right)^{2(n-1)} + \cdots\right] \;=\; S^2L^n \frac{2^{n-1}}{2^{n-1}-1}, \tag{1.14}$$

an increase of at most a factor 2. Unfortunately, Hirschberg's algorithm does not perform so well if it is combined with constraints to a contributing region. Such regions usually lie close to the diagonal of the DP table, and the Hirschberg halving strategy takes off almost nothing from such an essentially one-dimensional contributing region. The use of table constraints is highly desirable however, as the algorithm otherwise becomes impractical already for as little as four sequences

Another strategy, termed "bushy Viterbi", has the same memory usage as Hirschberg's algorithm and the same constant time penalty, but can be combined with the contributing region strategy as well. The idea is to combine the two stages of the Viterbi algorithm into one, and do backtracking on-the-fly. For this to work, each state requires an additional pointer to the state it refers to, and a reference count. The algorithm keeps optimal paths for each state in the current $n-1$-dimensional DP table slice. Whenever a slice is completed, all reference counts in the previous slice are decreased by one, and those that are not referenced by states in the current slice are removed. The table entries these states refer to have their reference counts decreased as well, and when they reach zero, the entries are removed in turn, and so on. Since the optimal paths for the various states quickly coalesce, the set of all paths is in practice very tree-like, as most coalescense events occurring close to the tips, and requires not much more memory beyond the L^{n-1} DP table entries. By doing the garbage collection only occassionally, the time complexity is also not much more than ordinary Viterbi. This algorithm was used to calculate the Viterbi alignment of Fig. 1.11.

1.5 Monte Carlo approaches

The major difficulty with statistical alignment has been in extending it to practical problem sizes. Alignments of tens or hundreds of sequences are routinely required in standard bioinformatics and phylogenetics settings. Exact techniques for statistical alignment are restricted to 4 or 5 sequences [LMSH03]. In this section we review Monte Carlo approaches that promise to considerably extend the domain of application of statistical alignment.

1.5.1 Statistical alignment using MCMC and TKF91

A number of researchers have been motivated to develop MCMC sampling algorithms to extend the use of the TKF91 model into the realms of practical multiple sequence alignment. The first such effort was by Holmes & Bruno [HB01] who produced an MCMC approach to statistical alignment under the TKF91 model conditional on a fixed tree topology and branch lengths. They used data-augmentation techniques to include paired-sequence alignments (henceforth referred to as branch-alignment) on each branch of the tree as well as inferred sequences at internal nodes. The proposal distribution they used consisted of two Gibbs sampling moves that resampled (1) a branch-alignment conditional on the two adjacent sequences (one of which might be an inferred sequence), and (2) a sequence at an internal node conditional on the three adjacent branch-alignments (while allowing insertion of characters unaligned to any of the three neighbours). Both of these moves involve sampling a subspace of the augmented problem from the exact conditional probability. This method was followed by another Gibbs sampler [HJP03] that reduced the state-space by not requiring the branch-alignments to be retained between successive states. This was achieved by using a more computationally intensive Gibbs move that resampled an internal sequence conditional only on the three neighbouring sequences. Their algorithm is $O(L^3)$ in the length of the sequence, as opposed to the $O(L^2)$ move of Holmes & Bruno. However the authors demonstrated that their algorithm's superior mixing more than made up for the extra computational time. In terms of effectively independent samples per CPU second the Hein and Jensen method appeared to be an improvement. Both of theses methods relied on EM optimization for values of the rates of substitution, insertion and deletion. Theoretically these parameters could easily be Metropolis sampled as part of the algorithm.

A third group has used MCMC to sample pairs of sequences [MFWvH01, Met03]. This work focuses on including alignment uncertainty into estimates of branch lengths. While they do not address the full problem of multiple alignments, they were the first to demonstrate the feasibility of a full Bayesian approach to co-sampling alignments and evolutionary parameters.

1.5.2 Removing the requirement for data-augmentation

One of the reasons that data-augmentation was required for the above MCMC methods was that the likelihood of the whole tree could not be efficiently calculated without internal sequences. A better solution would be to have an analogue of the Felsenstein peeling algorithm [Fel81], which would analytically sum out the sequences and gaps at internal nodes. With such an algorithm, the state could simply consist of the tree topology together with the homology structure (multiple sequence alignment) at the tips. No branch-alignments or internal sequences are then required. Not only would this considerably simplify the extension of the statistical alignment problem to co-estimation (the tree topology can be sampled without worrying about disturbing augmented data), but it should also reap computational benefits in the same way that the Hein & Jensen method did over the Holmes & Bruno one.

Surprisingly, a peeling method for the TKF91 model on a binary tree is not only possible, but is also computationally very cheap.We used this method to include indels as informative events in phylogenetic inference [LMD$^+$03], and it is the basis of the co-estimation method described below.

1.5.3 Example of co-estimation

Previous methods applying MCMC to statistical alignment problems did not sample evolutionary trees. The recent development the TKF91 peeling method mentioned above removes the requirement

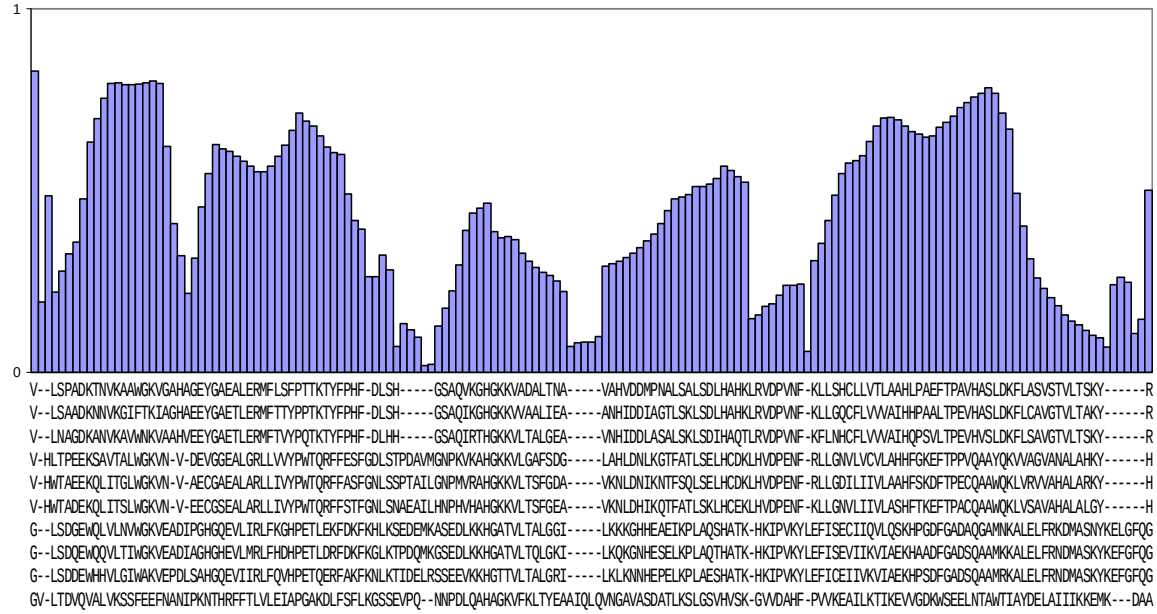


Fig. 1.12. The maximum posterior decoding of an alignment of ten globins (alpha hemoglobin [human; chicken; turtle], beta hemoglobin [human etc.], myoglobin [human etc.] and bean leghemoglobin). Estimates of posterior column probabilities were obtained by co-sampling phylogenetic trees and alignments through MCMC, using an alignment proposal distribution in windows of varying sizes, and a linear-time likelihood calculator for the TKF91 model in trees. For the MCMC run these results are based on, the estimated sample size was about 80. The column posteriors qualitatively agree with the analytic posteriors for the maximum likelihood alignment, based on just four of the ten globins (see Fig. 1.11).

for data augmentation, making tree-change proposals very simple. However, this ease of manipulating the tree comes with a drawback: without data augmentation it does not appear to be possible to perform Gibbs sampling on the alignment. Instead, other sampling methods are required, and careful design is needed for good performance. We have developed a partial importance sampler, which has good mixing properties in terms of estimated sample size (ESS) per CPU cycle. This method uses a stochastic score-based approach to propose new alignments. The proposal distribution is reshaped into the posterior distribution by standard Metropolized importance sampling techniques. We used the program BEAST written in Java as MCMC inference engine [DNRS02, DR03].

In more detail, the method works as follows. Given a multiple alignment, a random window is selected for modification, and a new subalignment in this window is proposed by a stochastic version of a score-based progressive alignment method. In this stochastic alignment method, sequences and profiles are progressively aligned using a pairwise algorithm, guided by the tree of the current MCMC state. In each iteration of the stochastic alignment, the dynamic programming table is filled as in the deterministic case by using linear gap penalties and standard similarity matrices. The stochastic element appears during the traceback phase. At each step during traceback, a random decision is made, biased towards the highest-scoring alternative. If the three alternatives have scores a, b and c respectively, the algorithm chooses between the alternatives with probabilities proportional to x^a, x^b and x^c, respectively, where $x > 1$. The stochastic path chosen determines the proposed alignment. It can be shown that all possible alignments of the subsequences can be proposed in this manner, and the proposal and back-proposal probabilities can be calculated relatively easily.

To get a reversible Markov chain, all window sizes must be proposed with a non-zero probability. We used a truncated geometric window-size distribution, but other distributions can also be used. The parameters that appear in this stochastic aligner algorithm, such as the average window size, determine the proposal distribution but do not influence the resulting posterior distribution. However, they do influence the efficiency of the MCMC sampler. For example, if the basis of exponentiation, x, is small, the proposal distribution will be flat, leading to small acceptance ratio. When x is too large, the proposal distribution will be too narrow, resulting in bad mixing behaviour

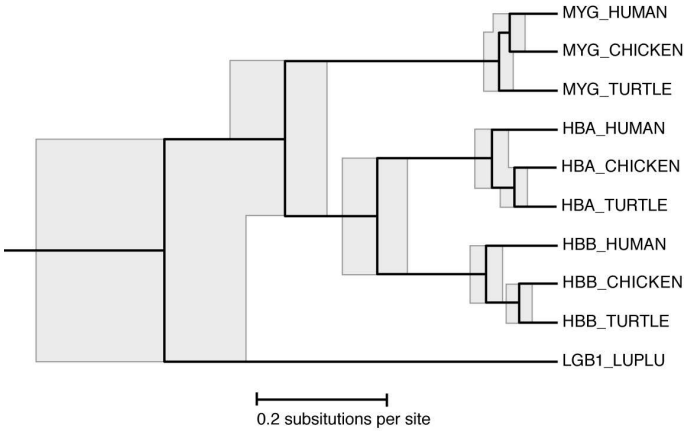


Fig. 1.13. The maximum posterior tree (black) relating the ten globins of Fig. 1.12, and 95% confidence intervals of the node heights (grey boxes). Most of the tree's topology is well determined, with the exception of the myoglobin subtree. Note that this relatively unresolved topology differs from the more well-defined topologies down the alpha and beta hemoglobin branches, that both conform to the accepted phylogeny of human, chicken and turtle.

if the distribution is far from the target distribution. The gap penalty value has a similar effect: if it is small, many alignments have similar probability of being proposed, while a big penalty results in a proposal distribution containing very few alignments.

Figures 1.12 and 1.13 illustrate the results of this co-estimation method on a set of 10 globin sequences. These pictures were produced from two MCMC runs with a total chain length of 10,000,000, and a burnin of 500,000. The basis of exponentiation x was chosen to be 1.5, and the mean window size was 40 amino acids. We used BLOSUM62 matrix and gap penalty -10.

1.6 Discussion

Recent progress in the development of statistical alignment methods, and especially the emergence of practical algorithms, has made it possible to treat the problem of sequence alignments as a statistical inference problem, estimate evolutionary indel parameters, and quantify alignment uncertainties. This development shows parallels with the success of statistical methods for phylogenetic inference in the last two decades.

Several aspects of statistical alignment methods have seen important progress: methods for pairwise alignment have been generalized to multiple sequences; more realistic insertion/deletion models have been proposed; hidden Markov model theory has conceptually simplified many algorithms; and MCMC methods have considerably extended the domain of application. These successes are due to, and resulted in, a growing interest in statistical alignment problems over the last few years [HWK$^+$00, HB01, Hei01, SH01, MFWvH01, Mik02, JH03, LMD$^+$03, LMSH03, Met03, HJP03, KM03, Hol03, MLH04]. Just three years ago, pairwise alignment was just about a feasible task for statistical alignment. At present the limit has been pushed up to about ten sequences. Much larger data sets are routinely of interest, and there is clearly a need for cleverly designed MCMC algorithm to tackle such problems.

The possibility of assessing the goodness-of-fit of a given statistical alignment model is a strength of probabilistic approaches, and allows for data-driven model improvements. Many such challenges remain, such as the inclusion of more biological realism in the models, incorporating for example indel rate heterogeneity, and variable substitution rates. Although heterogeneity of substitution processes has been extensively explored in the context of phylogenetic inference, it is largely unexplored in the context of sequence alignment. Perhaps even more importantly, the development of user-friendly software will be essential to make the methods appeal to a wider audience.

Sequence alignment is often just the first step in any analysis. Most current methods, such as comparative gene finding and RNA secondary structure prediction, but also phylogenetic inference,

assume a prior and fixed alignment. These methods can be combined with statistical alignment, either by a full co-estimation procedure, or simply by using a sample of alignments, or by incorporating the column reliabilities as weights. Such a hybrid approach would reduce the bias introduced by assuming exact knowledge of sequence homology, and at the same time increase the sensitivity by focusing on reliable data, and work in this direction is already in progress, see e.g. [HJ03].

The understanding of molecular evolution today owes much to the development of adequate evolutionary models. We hope that statistical alignment will contribute to this fundamental understanding in coming years.

Acknowledgements

The authors wish to thank Ian Holmes, Yun Song, Arnt von Haeseler, Bjarne Knudsen, Dirk Metzler, Korbinian Strimmer and Anton Wakolbinger for helpful remarks and stimulating discussions. The authors acknowledge support from EPSRC, and the MRC grant HAMKA. I.M. was further supported by a Békésy György postdoctoral fellowship.

References

[AGM+90] S. F. Altschul, W. Gisha, W. Miller, E. W. Meyers, and D. J. Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215(3):403–410, 1990.

[AMS+97] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped-BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, 25:3389–3402, 1997.

[CL88] H. Carillo and D. Lipman. The multiple alignment problem in biology. *SIAM J. Appl. Math.*, 48:1073–1082, 1988.

[DEKM98] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis.* Cambridge University Press, 1998.

[DNRS02] A. J. Drummond, G. K. Nicholls, A. G. Rodrigo, and W. Solomon. Estimating mutation parameters, population history and genealogy simultaneously from temporally spaced sequence data. *Genetics*, 161(3):1307–1320, 2002.

[DR03] A. J. Drummond and A. Rambaut. BEAST v1.0.3. http://evolve.zoo.ox.ac.uk/beast/, 2003.

[Edd01] S. Eddy. HMMER: Profile hidden Markov models for biological sequence analysis (http://hmmer.wustl.edu/), 2001.

[Fel81] J. Felsenstein. Evolutionary trees from DNA sequences: a maximum likelihood approach. *J. Mol. Evol.*, 17:368–376, 1981.

[Fel01] J. Felsenstein. The troubled growth of statistical phylogenetics. *Systematic Biology 50*, 50:465–467, 2001.

[Got82] O. Gotoh. An improved algorithm for matching biological sequences. *J. Mol. Biol.*, 162:705–708, 1982.

[HB01] I. Holmes and W. J. Bruno. Evolutionary HMMs: a Bayesian approach to multiple alignment. *Bioinformatics*, 17(9):803–820, 2001.

[Hei90] J. Hein. A unified approach to phylogenies and alignments. *Meth. Enzym.*, 183:625–644, 1990.

[Hei01] J. Hein. An algorithm for statistical alignment of sequences related by a binary tree. In *Pac. Symp. Biocomp.*, pages 179–190. World Scientific, 2001.

[Hir75] D. S. Hirschberg. A linear space algorithm for computing maximal common subsequences. *Communications of the ACM*, 18:341–343, 1975.

[HJ03] A. Hobolth and J. L. Jensen. Applications of hidden Markov models for comparative gene structure prediction. Technical Report MPS-RR 2003-35, MaPhySto, 2003.

[HJP03] J. Hein, J. L. Jensen, and C. N. S. Pedersen. Recursions for statistical multiple alignment. *PNAS*, 100(25):14960–14965, 2003.

[Hol03] I. Holmes. Using guide trees to construct multiple-sequence evolutionary HMMs. *Bioinformatics*, 19:i147–i157, 2003.

[HWK+00] J. Hein, C. Wiuf, B. Knudsen, M. B. Møller, and G. Wibling. Statistical alignment: Computational properties, homology testing and goodness-of-fit. *J. Mol. Biol.*, 302:265–279, 2000.

[JH03] J. L. Jensen and J. Hein. Gibbs sampler for statistical multiple alignment. *Stat. Sinica*, 2003+. (in press).

[KBH98] K. Karplus, C. Barrett, and R. Hughey. Hidden Markov Models for detecting remote protein homologies. *Bioinformatics*, 14:846–856, 1998.

[KM03] B. Knudsen and M. M. Miyamoto. Sequence alignments and pair Hidden Markov Models using evolutionary history. *J. Mol. Biol.*, 333:453–460, 2003.

[Kro97] A. Krogh. Two methods for improving performance of a HMM and their application for gene finding. In T. Gaasterland, P. Karp, K. Karplus, C. Ouzounis, C. Sander, and A. Valencia, editors, *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology*, pages 179–186. AAAI Press, 1997.

[LAK89] D. J. Lipman, S. F. Altschul, and J. D. Kececioglu. A tool for multiple sequence alignment. *Proc. Natl. Acad. Sci. USA*, 86:4412–4415, 1989.

[Lee01] M. S. Y. Lee. Unalignable sequences and molecular evolution. *Trends Ecol. Evol.*, 16:681–685, 2001.

[LM03] A. Löytynoja and M. C. Milinkovitch. A hidden Markov model for progressive multiple alignment. *Bioinformatics*, 19:1505–1513, 2003.

[LMD+03] G. A. Lunter, I. Miklós, A. Drummond, J. L. Jensen, and J. Hein. Bayesian phylogenetic inference under a statistical indel model. In *Proceedings of WABI'03*, volume 2812 of *Lec. Notes in Bioinformatics*, pages 228–244, 2003.

[LMSH03] G. A. Lunter, I. Miklós, Y. S. Song, and J. Hein. An efficient algorithm for statistical multiple alignment on arbitrary phylogenetic trees. *J. Comp. Biol.*, 10(6):869–889, 2003.

[MD02] I. M. Meyer and R. Durbin. Comparative ab initio prediction of gene structures using pair HMMs. *Bioinformatics*, 18(10):1309–1318, 2002.

[Met03] D. Metzler. Statistical alignment based on fragment insertion and deletion models. *Bioinformatics*, 19(4):490–499, 2003.

[MFWvH01] D. Metzler, R. Fleißner, A. Wakolbinger, and A. von Haeseler. Assessing variability by joint sampling of alignments and mutation rates. *J. Mol. Evol.*, 53:660–669, 2001.

[Mik02] I. Miklós. An improved algorithm for statistical alignment of sequences related by a star tree. *Bul. Math. Biol.*, 64:771–779, 2002.

[Mit99] G. Mitchison. A probabilistic treatment of phylogeny and sequence alignment. *J. Mol. Evol.*, 49:11–22, 1999.

[MLH04] I. Miklós, G. A. Lunter, and I. Holmes. A "long indel" model for evolutionary sequence alignment. *Mol. Biol. Evol.*, 21(3):529–540, 2004.

[Mor99] B. Morgenstern. DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment. *Bioinformatics*, 15:211 – 218, 1999.

[Mye86] E. W. Myers. An O(ND) difference algorithm and its variations. *Algorithmica*, 1:251–266, 1986.

[NHH00] C. Notredame, D. Higgins, and J. Heringa. T-Coffee: A novel method for multiple sequence alignments. *J. Mol. Biol.*, 302:205–217, 2000.

[NW70] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequences in two proteins. *J. Mol. Biol.*, 48:443–453, 1970.

[QG01] B. Qian and R. A. Goldstein. Distribution of indel lengths. *Proteins: struc. func. gen.*, 45:102–104, 2001.

[San75] D. Sankoff. Minimal mutation trees of sequences. *SIAM J. Appl. Math.*, 28:35–42, 1975.

[SH01] M. Steel and J. Hein. Applying the Thorne-Kishino-Felsenstein model to sequence evolution on a star-shaped tree. *Appl. Math. Let.*, 14:679–684, 2001.

[TDT94] J.D. Thompson, Higgins D.G., and Gibson T.J. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22:4673–4680, 1994.

[TKF91] J. L. Thorne, H. Kishino, and J. Felsenstein. An evolutionary model for maximum likelihood alignment of DNA sequences. *J. Mol. Evol.*, 33:114–124, 1991.

[TKF92] J. L. Thorne, H. Kishino, and J. Felsenstein. Inching toward reality: an improved likelihood model of sequence evolution. *J. Mol. Evol.*, 34:3–16, 1992.

[Ukk85] E. Ukkonen. Algorithms for approximate string matching. *Information and Control*, 64:100–118, 1985.

[vHV97] A. von Haeseler and M. Vingron. Towards integration of multiple alignment and phylogenetic tree construction. *J. Comp. Biol.*, 4(1):23–34, 1997.

[WJ94] L. Wang and T. Jiang. On the complexity of multiple sequence alignment. *J. Comp. Biol.*, 1:337–348, 1994.

[WMMM90] S. Wu, U. Manber, G. Myers, and W. Miller. An O(NP) sequence comparison algorithm. *Information Processing Letters*, 35:317–323, 1990.