

# Stochastic Models of Sequence Evolution including Insertion-Deletion events

István Miklós<sup>1,2,3</sup>, Ádám Novák<sup>2</sup>, Rahul Satija<sup>2</sup>, Rune Lyngsø<sup>2</sup>, and Jotun Hein<sup>2</sup>

<sup>1</sup> Bioinformatics group, Alfréd Rényi Institute of Mathematics, Hungarian Academy of Sciences, 1053 Budapest, Reáltanoda u. 13-15, Hungary  
miklosi@renyi.hu,

<sup>2</sup> Bioinformatics Group, Department of Statistics, University of Oxford, 1 South Parks Road, OX1 3TG Oxford, UK  
miklosi@ramet.elte.hu,

<sup>3</sup> Data Mining and Search Research Group, Computer and Automation Institute, Hungarian Academy of Sciences, 1111 Budapest, Lágymányosi u. 11., Hungary

Wednesday 1<sup>st</sup> October, 2008, 22:42

**Abstract.** Comparison of sequences that have descended from a common ancestor based on an explicit stochastic model of substitutions, insertions and deletions has risen to prominence in the last decade. Making statements about the positions of insertions-deletions (abbr. indels) is central in sequence and genome analysis and is called alignment. This statistical approach is harder conceptually and computationally, than competing approaches based on choosing an alignment according to some optimality criteria. But it has major practical advantages in terms of testing evolutionary hypotheses and parameter estimation. Basic dynamic approaches can allow the analysis of up to 4-5 sequences. MCMC techniques can bring this to about 10-15 sequences. Beyond this, different or heuristic approaches must be used. Besides the computational challenges, increasing realism in the underlying models is presently being addressed. A recent development that has been especially fruitful is combining statistical alignment with the problem of sequence annotation, making statements about the function of each nucleotide/amino acid. So far gene finding, protein secondary structure prediction and regulatory signal detection has been tackled within this framework. Much progress can be reported, but clearly major challenges remain if this approach is to be central in the analyses of large incoming sequence data sets.

## 1 Introduction

Although bioinformatics has diversified enormously, certain aspects have a long history and could be viewed as classical bioinformatics. The best example must be the application of string comparison algorithms to sequence alignment. This has a history spanning the last three decades, beginning with the pioneering paper by Needleman and Wunsch [1]. They used dynamic programming to maximise a similarity score based on a matching score for amino acids, and a cost function for insertion and deletions. In 1973 Sankoff and Cedergren generalised a distance minimising 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 generalised and accelerated. A completely different approach to alignment was introduced in 1994 by Krogh *et al.* [2], who used Hidden Markov Models (HMMs) to describe a family of homologous proteins. This statistical approach has proved very successful, despite not

being based on an underlying model of evolution, or phylogeny. HMMs and their generalisations - Stochastic Context Free Grammars (SCFGs) - has since become key tools in bioinformatics, including evolutionary models.

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 [3]. In contrast to global alignment, this approach does not force the entire sequence to be aligned but rather, the score of the alignment is determined based on local similarities. 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 [4].

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. The key algorithms doing this were published by Fitch [5] and Hartigan [6] and since then implemented in a variety of programs. Over the last two decades the parsimony method of phylogenetic reconstruction has received increasing criticism, 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. The key algorithm for allowing calculation of the probability of observing a set of nucleotides on a phylogeny was published by Felsenstein in 1981 [7]. 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 [8], 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 [9] proposed a time-reversible Markov model for insertions and deletions (termed the TKF91 model), that allows 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.

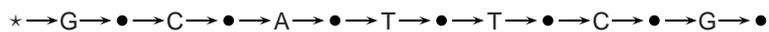
The three main types of mutations modifying biological sequences are insertions, deletions and substitutions. The TKF91 model is the simplest conceivable model involving these three types of mutations. In this model, the positions of a sequence evolve independently and identically. Each character in the sequence can be substituted with another character according to a prescribed reversible continuous-time Markov model on the possible characters. Insertion-deletions are modelled as a birth-death process with insertion and deletion rates  $\lambda$  and  $\mu$ , where insertions occur between any two characters or at the very beginning or end of the sequence. The pairwise statistical align-

ment problem is to calculate the likelihood of two sequences evolving from a common ancestor under the TKF91 model or one of its extensions, and can be solved in time  $O(L_1 L_2)$  [9, 10] where  $L_i$  is the length of sequence  $i$ . The related inference problem of computing the most likely alignment relating the two sequences can be solved with similar techniques.

The multiple statistical alignment problem is to calculate the likelihood of a set of sequences, namely, what is the probability of observing a set of sequences, given all the necessary parameters that describe the evolution of sequences. Generalising the statistical pairwise alignment algorithm to many sequences, proved considerably harder than the corresponding optimisation problem. Mike Steel solved it for the special case, when the sequences are related by a star tree [11]. This was generalised by Hein to a binary tree [12]. This paper also showed that the TKF91 process could be described by an HMM. Hein, Pedersen and Jensen in 2003 [13] further showed that recursions for statistical alignment possess a surprising asymmetry in removing the first and the last column in an alignment, which is in sharp contrast to optimisation alignment. The algorithm has  $O(5^n L^n)$  running time, where  $n$  is the number of sequences, and  $L$  is the geometric mean of the sequence lengths. This has since been improved to  $O(2^n L^n)$  by Lunter *et al.* [14, 15]. As an alternative to the time consuming exact computations, MCMC methods were first used by Churchill in 1997 [16] for general alignment and for TKF91 based alignment by Holmes and Bruno in 2001 [17].

## 2 The Basic Models

Traditional score based alignment models aim to render the evolutionary most plausible alignment optimal. However, this plausibility is not quantified as a distribution over alignments. In their pioneering paper [9] Thorne, Kishino and Felsenstein proposed a model (TKF91) for sequence evolution. The model is continuous-time and allows insertions and deletions (indels) as well as substitutions. This defines a distribution over all alignments relating two sequences, enabling a proper statistical analysis of pairwise sequence homology. The model treats all events as independent, single-nucleotide events, and is arguably the simplest continuous-time model for sequence evolution under insertion, deletion and substitution events conceivable. Though somewhat naïve, a major advantage of this simple model is that it can be solved analytically. The analytic treatment can be formulated in terms of a hidden Markov model defining a left-to-right construction of sequence alignments with the correct distribution. Here we will first describe the TKF91 model and sketch the derivations leading to the HMM formulation of the model. We then briefly introduce an extension to the TKF91 model, denoted TKF92 [18], that addresses the single nucleotide simplification by allowing non-overlapping indels of any length. This extension is the statistical alignment equivalent of the “affine gap cost” in the context of score-based alignment [19].



**Fig. 1.** The TKF91 model view of the string GCATTTCG. The immortal link is shown as ★ while the normal links are shown as •.

## 2.1 The TKF91 Model

A single nucleotide substitution or deletion event will affect a nucleotide, while an insertion event will affect the space between two nucleotides. The TKF91 model captures this by treating a nucleotide sequence as an alternating chain of nucleotides and *links*, see Fig. 1. Insertions can occur at the very beginning and end of a sequence, so the chain begins and ends with a link. The insertion-deletion process is modelled as a continuous-time Markov process on such chains. Insertions originate from links and insert a new nucleotide, drawn from the background distribution, and a new link before the following nucleotide. This happens with a rate of  $\lambda$  at each link. Deletions originate from a nucleotide and removes both the nucleotide and the following link, and occur at a rate  $\mu$  at each nucleotide. Alternatively we can consider the process as acting on letter-link pairs that get inserted and deleted. Such models are known as birth-death processes, as we have two competing processes of birth and death of entities. Note that the model leaves no way to delete the initial link, which consequently is called the immortal link. The immortal link prevents the empty sequence being a sink for the process, as a new sequence can be grown from the immortal link remaining after the initial sequence has been completely deleted. This may appear to contradict *ex nihilo nihil fit*, but one should keep in mind that nucleotide sequences usually appear in a genomic context rather than flanked by empty space.

So far we have only discussed the insertion and deletion part of the TKF91 model. Sequences can also change by substitution. This is modelled by a parallel continuous-time substitution process, that allows each letter to be substituted with another letter. In the original formulation Felsenstein's one-parameter model [20] was used, but this can be replaced with other substitution models without difficulty. The substitution process can be dealt with independently by standard single nucleotide substitution model methods, so for the remainder of this section we will focus on the insertion-deletion part of the process.

In the TKF91 birth-death process the number of entities, letter-link pairs, always increases or decreases by one. This results in a state graph with linear topology which is a sufficient condition for time-reversibility of the insertion-deletion process. Time-reversibility is equivalent to detailed balance in the equilibrium distribution. In other words, if we let  $q_k$  denote the probability of drawing a sequence of length  $k$  from the equilibrium distribution, then

$$\mu k q_k = \lambda q_{k-1} \quad \Leftrightarrow \quad \frac{q_k}{q_{k-1}} = \frac{\lambda}{\mu} \quad (1)$$

as detailed balance requires the change from length  $k$  sequences to length  $k - 1$  sequences to equal the change in the reverse direction; length  $k$  sequences have  $k$  letters where a deletion event can occur and length  $k - 1$  sequences have  $k$  links where an insertion event can occur. It follows that

$$q_k = \left(\frac{\lambda}{\mu}\right)^k q_0. \quad (2)$$

Since the  $q_k$  define a probability distribution over finite sequence lengths we also have  $\sum_{k=0}^{\infty} q_k = 1$ . Combining, we obtain

$$q_k = \left(1 - \frac{\lambda}{\mu}\right) \left(\frac{\lambda}{\mu}\right)^k \quad (3)$$

This only yields a well defined distribution (in fact, geometric distribution) for  $\lambda < \mu$  which should hardly be surprising: a sequence of length  $k$  has overall insertion rate  $\lambda(k + 1)$  (from  $k + 1$  links) and overall deletion rate  $\mu k$  (from  $k$  letters), so if  $\lambda \geq \mu$  the overall insertion rate will always exceed the overall deletion rate and sequences would grow indefinitely. Metzler introduced the Fragment Insertion-Deletion (FID) models [21] that are generalisations of the TKF92 model. These models do not need the constraint that  $\lambda$  should be smaller than  $\mu$ . In this way the stochastic process might not have an equilibrium distribution. However, Metzler showed that parameter estimation is still possible in these models.

An alignment postulates a detailed relationship between the individual nucleotides in the sequences aligned. A column containing two nucleotides states that these are homologous, *i.e.* descend from the same nucleotide in a shared ancestral sequence. As the insertion-deletion process of the TKF91 model is time-reversible, if a time-reversible substitution model is also used the entire process will be reversible. This allows us to simplify the scenario such that we can assume that one sequence evolved into the other. Assume now that we let the TKF91 process act on an initial sequence while keeping track of the fate of each nucleotide in this sequence. The outcome can then be summarised in an alignment. Each nucleotide that survived without being deleted will be homologous to a nucleotide, possibly modified by one or more substitution events, in the final sequence, and adding a column to the alignment constituted by these two nucleotides will reflect that. A nucleotide not surviving should be aligned to a gap in the final sequence, while a nucleotide that arose through an insertion should be aligned to a gap in the initial sequence. In this way the TKF91 model defines a distribution over all possible alignments of two sequences. Observe that each possible alignment will usually correspond to an infinite number of possible histories, as *e.g.* a nucleotide can be subject to an arbitrary number of substitution events.

As the nucleotides evolve independently, the probability of any particular outcome is just the product of the independent probabilities of the outcome concerning each nucleotide in the initial sequence. So if we for a given time  $t$  can determine the probability of any particular outcome for a single nucleotide, we can easily obtain the same for a nucleotide sequence. For a given nucleotide in the initial sequence two aspects will influence its alignment contribution, namely whether the nucleotide survived and how many nucleotides it left in the final sequence, either directly or through insertions that can be traced back to its associated link. Furthermore, we also need consider the insertions that can be traced back to the immortal link. The following table captures the possible outcomes, the alignment block summarising the outcome, and the notation used for the probability of the outcome.

Outcome	Alignment	Probability
Nucleotide survives and adds $n - 1$ new ones	$\# - \dots -$ $\# \# \dots \#$	$p_n^H(t)$ ( $n = 1, 2, \dots$ )
Nucleotide dies but adds $n$ new ones	$\# - \dots -$ $- \# \dots \#$	$p_n^N(t)$ ( $n = 0, 1, \dots$ )
Immortal link adds $n$ new nucleotides	$* - \dots -$ $* \# \dots \#$	$p_n^I(t)$ ( $n = 0, 1, \dots$ )

As previously mentioned, we only consider the probability of the insertion-deletion part of the TKF91 model. Hence nucleotides are represented merely by the Felsenstein wildcard symbol  $\#$ , cf. [17]. A surviving nucleotide will further incur a factor of the corresponding substitution probability in the substitution model applied, and an inserted nucleotide will incur a factor of the corresponding nucleotide probability in the substitution model's equilibrium distribution. Except for the immortal link, the links have been left out for clarity.

With this notation we can now set up differential equations, also called Kolmogorov's forward equations, for the time-dependent outcome probabilities, by considering the rates at which we switch between different outcomes. For instance, the equations for the immortal link outcome probabilities  $p_n^I(t)$  are

$$\frac{d}{dt}p_n^I(t) = (n+1)\mu p_{n+1}^I(t) + n\lambda p_{n-1}^I(t) - [n\mu + (n+1)\lambda]p_n^I(t) \quad (4)$$

$$p_n^I(0) = \begin{cases} 1 & \text{for } n = 0 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where for convenience  $p_{-1}^I(t)$  is defined and equal to 0 for all  $t$ . For time  $t = 0$  no new nucleotides can yet have been inserted. For  $t > 0$  we end up with  $n$  new nucleotides by either inserting an extra nucleotide from a state with  $n - 1$  new nucleotides, or by deleting a nucleotide from a state with  $n + 1$  new nucleotides; we vacate a state with  $n$  new nucleotides by either deleting or inserting a nucleotide. Similarly

$$\frac{d}{dt}p_n^H(t) = n\mu p_{n+1}^H(t) + (n-1)\lambda p_{n-1}^H(t) - n(\mu + \lambda)p_n^H(t) \quad (6)$$

$$p_n^H(0) = \begin{cases} 1 & \text{for } n = 1 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

with  $\forall t : p_0^H(t) = 0$  and

$$\frac{d}{dt}p_n^N(t) = \mu [(n+1)p_{n+1}^N(t) + p_{n+1}^H(t)] + (n-1)\lambda p_{n-1}^N(t) - n(\mu + \lambda)p_n^N(t) \quad (8)$$

$$p_n^N(0) = 0 \quad \text{for all } n \quad (9)$$

with  $\forall t : p_{-1}^N(t) = 0$ .

These equations can be solved analytically [9], and using the following functions as shorthands

$$\beta(t) = \frac{1 - e^{(\lambda-\mu)t}}{\mu - \lambda e^{(\lambda-\mu)t}}, \quad E(t) = \mu\beta(t), \quad N(t) = (1 - e^{-\mu t} - \mu\beta(t))(1 - \lambda\beta(t)),$$

$$I(t) = 1 - \lambda\beta(t), \quad B(t) = \lambda\beta(t), \quad H(t) = e^{-\mu t}(1 - \lambda\beta(t))$$

the solution can be written succinctly as

$$p_0^N(t) = E(t) \quad (10)$$

$$p_n^N(t) = N(t)B(t)^{n-1} \quad \text{for } n > 0 \quad (11)$$

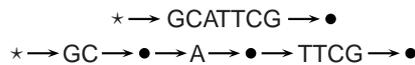
$$p_n^H(t) = H(t)B(t)^{n-1} \quad (12)$$

$$p_n^I(t) = I(t)B(t)^n \quad (13)$$

A key observation is that in all three scenarios, the factor incurred for each column corresponding to the insertion of a new nucleotide is the same, namely  $B(t)$ , regardless of the number of columns already added. This observation is the key to the HMM formulation discussed further in Appendix A.1.

## 2.2 The TKF92 Model

The main deficiency of the TKF91 model is that all events are treated as independent single nucleotide events. It is well known that substitution rates are context dependent. More severely, though, insertions and deletions rarely occur as single nucleotide events, but rather as insertions or deletions of segments of nucleotides, an issue originally addressed for score based alignment methods in [19]. In an attempt to inch towards reality Thorne *et al.* in [18] proposed a generalisation of their TKF91 model. This generalisation, the TKF92 model, allows insertions and deletions of segments with lengths drawn from an arbitrary distribution. As pointed out in the paper, the TKF92 model still falls short of reality by not allowing the inserted and deleted segments to overlap.



**Fig. 2.** Two of the 32 TKF92 model views of the string GCATTCG. Fig. 1 presents a third view.

In our exposition of the TKF91 model we took the view that deletions originate at nucleotides and also remove the associated link following the deleted nucleotide. One can equivalently view this as a deletion of a normal link that also removes the associated nucleotide in front of it. The TKF92 generalises the link model of sequences by splitting a sequence into *fragments* separated by links, rather than single nucleotides. A normal link will be associated with the preceding fragment, and deleting the link will also remove the associated fragment. Similarly, when a link spawns an insertion a fragment rather than just a single nucleotide is inserted. Focusing just on the links, the birth-death process of the TKF92 model is identical to that of the TKF91 model. Hence, transition probabilities for the link birth-death process has the same solution in the TKF92 model. The distribution over fragment lengths and the single nucleotide substitution process operate independently, hence the three elements are combined by simple multiplication.

The major weakness of the TKF92 model is that the link structure is not allowed to change over time. A fragment can thus only be deleted in its entirety. This is to some extent alleviated by taking all possible fragmentations of a sequence into account, allowing any fragment of an initial sequence to be a candidate for future deletion. However, inserted fragments cannot be partly deleted or deleted as part of a larger fragment. For example, the two event evolutionary history  $A \rightarrow ACG \rightarrow AG$  that first inserts two nucleotides at the end of an initial sequence and then deletes one of them again is not included in the TKF92 model. Still, the model offers a significant improvement in realism over the TKF91 model. Moreover, when a geometric distribution is assumed for fragment lengths, inference under the model remains as efficient as under the TKF91 model [18] by an algorithm structurally similar to Gotoh's algorithm [19] for affine gap cost in score-based alignment.

## 3 Applications

### 3.1 Inference of Phylogeny

It is well known that the score based multiple alignment methods introduce more variance into the phylogeny analysis than the tree-building methods themselves [22, 23].

The proper scoring of a multiple alignment could be given only if the phylogenetic tree on which the analysed sequences are related were known. On the other hand, the phylogenetic tree of the sequences needs an alignment of the sequences. To break out from this chicken-egg problem, the co-estimation of alignment and phylogeny would be desirable. The multiple statistical alignment approach provides an excellent framework for such a co-estimation. Indeed, to define a joint Bayesian distribution of alignments, trees and evolutionary parameters, all that is needed is some joint prior distribution of trees and evolutionary parameters. Such priors for trees and evolutionary parameters are defined in several manuscripts, for example [24, 25], or the readers might also consult with [26].

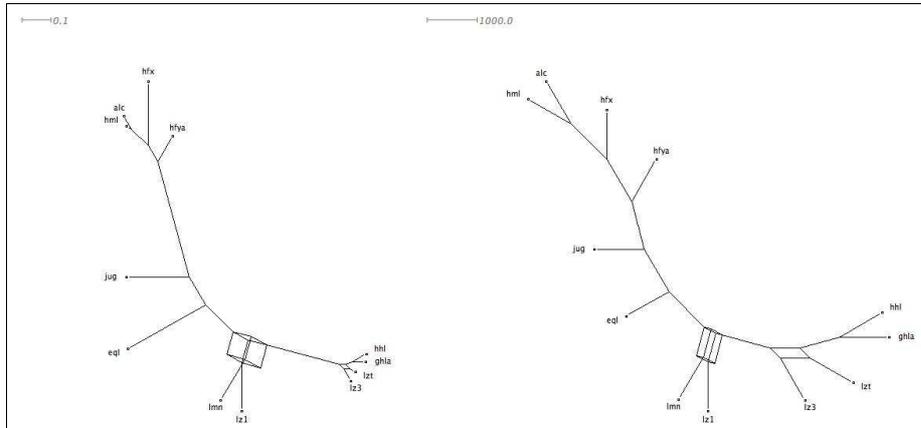
Once the Bayesian distribution is defined, a Markov chain Monte Carlo approach can be used to sample from the posterior distribution of alignments, trees and evolutionary parameters. The uninteresting coordinates might be marginalised from the distribution, for example, if the user is interested in only the phylogenies, then the alignments and evolutionary parameters might be marginalised. In this way, a set of sampled evolutionary trees might be generated. This set of trees might be summarised in several ways. One of the most informative analyses is the consensus network, see for example, Fig. 3. The consensus network contains all the *splits* that have been sampled with a probability greater than or equal to a prescribed threshold. A split is a bipartition of the set of species involved in the analysis. On a phylogenetic network, a split is represented by one or more parallel edges of the network. Parallel edges appear when there are conflicting splits in the Bayesian ensemble. Such conflicting splits are described with rectangles, cubes or hypercubes. When all but one dimension of the hypercube is collapsed, we get back a phylogenetic tree with the Bayesian support of the remaining dimension of the hypercube, which represents a split.

Dedicated software exists for analysing a Bayesian ensemble of phylogenetic trees. SplitsTree4 [27] is a software package, that provides a user-friendly graphical interface and allows results to be saved in both graphical and Nexus text file formats. It is implemented in Java and thus available across multiple platforms. The StatAlign software package [28], also implemented in Java, allows sampling of evolutionary trees from a joint distribution on trees, alignments and model parameters. The sampled trees can be saved in Nexus format, which can be read and analysed by SplitsTree4 for a statistical alignment based analysis of supported phylogenies.

### 3.2 Protein Structure Prediction

As genome sequencing costs continue their downward spiral, sequencing of organisms has become increasingly viable. A consequence of the increasing speed of accumulating biological sequences is that the gap between the number of known structures and the number of known protein sequences keeps increasing. As a result, there is a high demand for reliable computational methods and *in silico* estimation of protein structures remains one of the most challenging tasks in bioinformatics.

The central dogma of comparative bioinformatics methods for proteins is that the structures of proteins are more conserved than their amino-acid sequences. Therefore it is possible to map the structure of a sequence onto homologous sequences. However, insertions and deletions separating two homologous sequences accumulate, and hence, homologous characters in the two sequences will occupy different positions, which causes a non-trivial problem of identifying homologous positions. Sequence alignment algorithms address this problem [1, 3, 19, 32]. They maximise the similarity between aligned positions while also minimise the insertions and deletions needed to align the sequences.



**Fig. 3.** Consensus network of the Glycosyl hydrolase family 22 (lysozyme) protein sequences. The sequences have been downloaded from the Homstrad database [29], and have been analysed with the statistical alignment program described in [30]. 1000 phylogenies were sampled from the posterior distribution by subsampling from the MCMC chain, sampling the current phylogeny every 1000<sup>th</sup> once convergence had been reached. Consensus networks were obtained with Split-Tree4 [27], using the algorithm of Holland and Moulton [31]. **Left:** consensus network when considering only splits with a posterior probability 0.1 or higher. Edges indicate mean length. **Right:** the same consensus network, but in this case, edges indicate counts in the Bayesian ensemble.

The relationship between gap-penalties and similarity scores can be set such that they maximise the number of correctly aligned positions in a benchmark set of alignments [33, 34]. By contrast, stochastic models are capable to calibrate their parameters by applying a Maximum Likelihood or Bayesian approach even if no benchmark set is available. The first such stochastic models were Hidden Markov Models, which have appeared in bioinformatics almost fifteen years ago [2]. Thorne, Kishino and Felsenstein introduced continuous-time Markov models for describing insertion and deletion events [9, 18], and they showed on simulated data that the Maximum Likelihood method could correctly estimate the insertion-deletion as well as the substitution parameters with which the simulated data had been generated. The TKF models have subsequently been improved [10, 35], and have been tested for alignment accuracy on biological data [10].

The other main advantage of stochastic models above automatic parameter estimation is that such models can provide posterior probabilities for each estimated alignment column as well as for the whole alignment, and these posterior probabilities correlate with the probability for the alignment column being correctly aligned [10, 26, 36].

The uncertainty in the sequence alignment can be slightly reduced by simultaneously aligning more than two sequences. During the last twenty years, much effort has been put in developing accurate multiple sequence alignment methods. Although efficient algorithms exist for any type of pairwise alignment problem, the multiple sequence alignment problem is proven to be hard. Indeed, Wang and Jiang proved that the optimal multiple sequence alignment problem under the sum-of-pairs scoring protocol is NP-hard [37]. Although it is not formally proved, it is strongly believed that

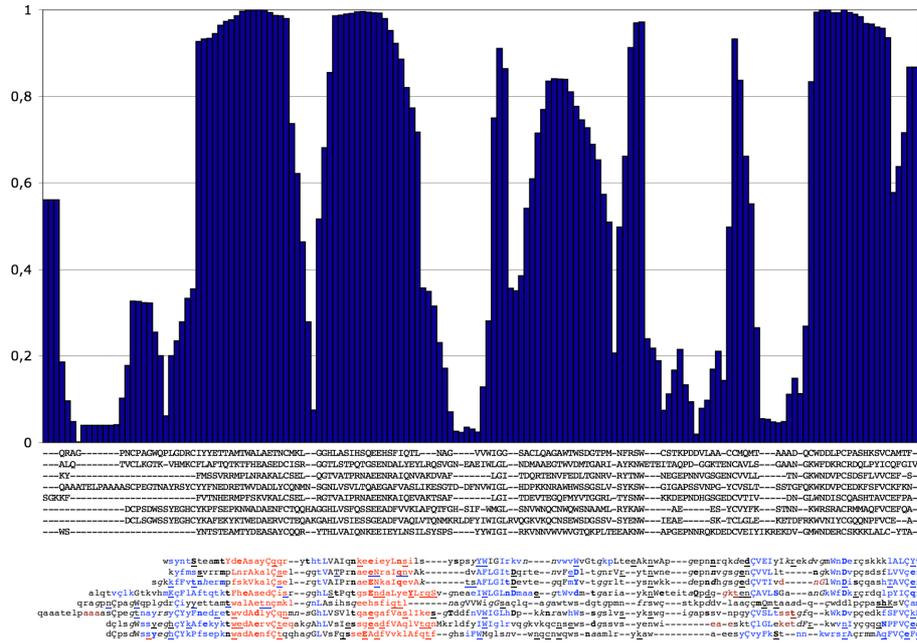
the statistical approach to multiple sequence alignment is algorithmically not simpler than score-based approaches. Since it is unlikely that fast algorithms exist for any type of exact multiple sequence alignment problem, heuristic approaches have become widespread. Profile-HMM methods [38, 39] align sequences to a profile-HMM instead of each other, and the multiple sequence alignment is obtained by aligning sequences together via a profile-HMM. Since the jumping and emission parameters of the HMM are learned from the data, this approach needs many sequences for parameter optimisation. Furthermore, profile-HMMs do not consider evolutionary relationships amongst sequences, and hence, they cannot properly handle over-representation of evolutionary groups.

Iterative sequence alignment approaches have been introduced for score-based methods in the eighties [40, 41] and have recently been extended for stochastic methods [36, 42] using the transducer theory [43, 44]. The drawback of iterative approaches is that in each iteration, they consider only a single, locally optimal alignment that might not lead to a globally optimal alignment. Additionally, the statistical methods naturally underestimate the uncertainty of posterior probabilities as they consider only locally optimal partial solutions.

The Markov chain Monte Carlo (MCMC, see also Section A.2) approach represents a third way to attack the multiple stochastic alignment problem. In statistical alignment, it was first introduced for assessing the Bayesian distribution of evolutionary parameters of the TKF91 model aligning two sequences [45], and has subsequently been extended to multiple sequence alignment [14, 15, 17, 46–48]. The central theory of Markov chain Monte Carlo [49, 50] states that the Markov chain will be in the prescribed distribution after infinite number of random steps. Obviously, we cannot wait infinite many steps in practice, and therefore the success of MCMC methods depends on fast convergence: if the Markov chain converges quickly to the prescribed distribution, the bias of samples from the Markov chain after a limited number of steps will be negligible. There are two classic approaches for checking convergence of MCMC, one is to measure autocorrelation in the log-likelihood trace or a few other statistics of the Markov chain, the second is to run several parallel chains with different random starting points [51].

Since practical methods and software packages for the multiple statistical alignment problem have been introduced only in the last five-six years, a large-scale, comprehensive analysis on the performance of methods for protein structure prediction has been published only recently [30]. In that paper, Miklós *et al.* presented a survey on how stochastic alignment methods can be used for protein secondary structure predictions. The authors applied homology modelling, namely they mapped the known secondary structure of one of the sequences onto the other sequence(s) via the best (multiple) alignment or the Bayesian distribution of (multiple) alignments. The prediction can be based on pairwise or multiple alignments and in both cases, either only a single, optimal alignment or the whole posterior distribution of alignments is used for prediction. In this way, they presented four different protocols for protein structure prediction. The four methods have been tested on protein structure families from the HOMSTRAD database [29]. This database contains 1031 families of protein sequences, for each family, a structural multiple alignment is given, and the alignments are annotated in JOY format [52]. The JOY format tells – amongst others – the known secondary structures of the sequences. They were interested in the question how much one can gain by involving more sequences and the posterior distribution of the alignments into the secondary structure prediction.

In all cases, posterior probabilities of aligning characters in pairwise or multiple alignments correlated with the probability that the secondary structure predictions based



**Fig. 4.** Top: The Maximum Posterior Decoding alignment of C-type lectin sequences estimated from MCMC samples following the protocol described by Miklós *et al.* [30]. Posterior probabilities of alignment columns are indicated for each alignment column. Bottom: The multiple structural alignment of the same sequences as given in the HOMSTRAD database [29] in JOY format [52]. Beta-strands are indicated by blue characters, alpha helices are indicated by red ones.

on the alignments in question were correct. See Fig. 4 for an example of predicted multiple alignment with posterior probabilities for alignment columns. The authors also found that pairwise alignment methods are under-confident on predicting alpha helices and beta sheets, namely, posterior probabilities of alignment columns are lower than the actual probability that the structure prediction based on the alignment column is correct, while they are over-confident on predicting  $3_{10}$  helices, *i.e.*, posterior probabilities for these alignment columns are greater than the probabilities that the secondary structure prediction for these amino acids is correct. Multiple alignment methods provide slightly more reliable predictions about their reliability of secondary structure predictions – they are less over-confident on  $3_{10}$  helix predictions.

Secondary structure predictions can be given based on single, optimal pairwise or multiple alignments and also based on the posterior ensemble of alignments. In the latter case, posterior probabilities are closer to the probabilities that the secondary structure prediction is correct, especially when the structure prediction is based on the posterior distribution of multiple sequence alignments.

The multiple sequence alignment is the Holy Grail of bioinformatics [53] since what “one or two homologous sequences whisper ... a full multiple sequence alignment shouts out loud” [54]. The experiments show that multiple sequence alignments not only highlight conserved positions better than pairwise alignments, but they also more reliably indicate the reliability of their prediction capabilities.

This extra information could be exploited in three-dimensional protein structure prediction: high posterior probabilities indicate the regions of the sequence alignment where the alignment accuracy is significantly better than the average alignment accuracy. These parts can be used as a reliable scaffold in homology modelling. On the remaining, unreliable parts, homology modelling is expected to have a low quality, and hence the spatial structure of these regions should be predicted with alternative methods, like *ab initio* threading methods (*e.g.* [55–57]).

It is worth mentioning that the alignment methods applied do not consider any information about how secondary structures evolve. It is well-known that different secondary structure elements follow different substitution processes, and this difference in the substitution pattern can be used for secondary structure prediction [58]. It is fairly straightforward to incorporate into current statistical alignment methods *a priori* knowledge on the substitution, insertion and deletion processes of secondary structures, and we expect that such combined approaches will have a better performance in structure prediction. Furthermore, secondary structures can be predicted not only in a comparative way, but also using a single sequence, based on the statistical properties of the amino acids in different secondary structure types, see *e.g.* [59, 60]. Potential prior distributions for secondary structure elements might be derived from such statistics and might be used in Bayesian analysis.

The running time of the methods obviously increases with the complexity of the background models, and analyses utilising such combined methods currently take too long to be applicable for everyday use on personal computers. However, the speed of processors keeps increasing exponentially following Moore’s law, and will soon reach a level when it won’t pose a barrier to such combined approaches. Moreover, there are also promising channels to improve the running time of the methods. The standard approach for statistical multiple alignment is going to be MCMC, and current implementations make use of very basic tricks only, like the alignment window cut algorithm described in Section A.3. Several groups are working on making MCMC alignment methods more efficient and achieving faster mixing, and significant improvements are expected in the coming years.

### 3.3 Signal Prediction

The identification of conserved DNA sequences by comparative genome sequence analysis has been widely used to annotate both protein-coding and gene regulatory elements in a wide variety of taxa [61–65]. Alignment based phylogenetic footprinting [66] approaches assume that regulatory elements in non-coding regions are subject to purifying selection, and therefore will exhibit higher levels of conservation than surrounding neutral sequence. Numerous phylogenetic footprinting approaches have been developed and successfully applied to detect conserved regulatory elements in diverse taxa [61–63]. A popular approach used in the creation of University of California Santa-Cruz (UCSC) Genome Browser conservation scores, phastCons, implements a hidden Markov model (HMM) with a hidden state for conserved regions, and a hidden state for non-conserved regions [65]. Conserved elements are predicted by fitting the HMM to an alignment by maximum likelihood. The algorithm, however, assumes a single “perfect” alignment, and by ignoring the possibility of alignment uncertainty, these predictions are highly sensitive to both alignment errors and regions where alternate alignments may describe the true evolutionary history. A dependence on a single alignment may be particularly harmful when searching for regulatory motifs, such as transcription factor binding sites (TFBS), which are difficult to reliably align due to their short lengths (6-15

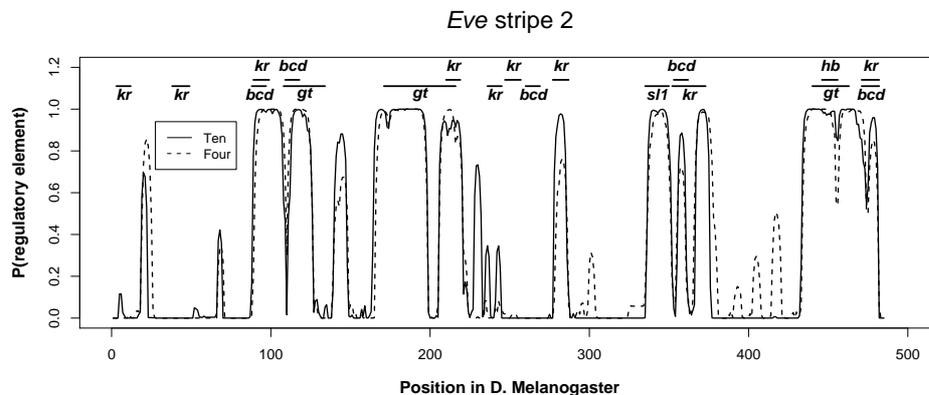
nucleotides) and tolerance for degenerate nucleotides [67]. Recent studies have noted that single-alignment phylogenetic footprinting approaches often produce inaccurate or inconsistent results depending on the alignment method used, and have called for new techniques capable of controlling for alignment error and uncertainty [23, 61, 68–70].

Comparative approaches that analyse an alignment distribution have been previously implemented to predict regulatory elements from pairwise comparisons. [64] analysed orthologous sequences from the human and mouse genomes by analysing a distribution of alignments using the Bayes Block Aligner [71], although the algorithm did not consider separate models for regulatory elements and background sequence and simply examined posterior alignment column probabilities. [72] developed MORPH, a framework which detects and aligns instances of known motifs, specified by position state weight matrices (CITE), by summing over all possible alignments of two species. While this approach has the additional requirement that binding site motifs must have been previously characterised, the authors report that binding site predictions are robust to alignment ambiguities.

[73] combined statistical alignment and phylogenetic footprinting to create SAPF, a software package with the following features. The SAPF model considers a multiple alignment HMM built by placing an HMM transducer [43] on each branch of a phylogenetic tree, and then doubling the number of states to model both quickly and slowly evolving regions. Thus, a state path through the final SAPF HMM represents both a multiple sequence alignment and an annotation of each alignment column as either quickly evolving (neutral sequence) or slowly evolving (functional sequence).

The authors demonstrate how the combined technique outperformed the analysis of a single alignment when analysing both simulated datasets and *cis*-regulatory modules in *Drosophila* species. The differential in accuracy was found to be especially high when there was uncertainty in the alignment of functional regions of the sequences. The authors also demonstrate the potential for significant improvement when analysing four species compared to analysing only a pairwise alignment. However, as the number of sequences to analyse increases, the number of states in the HMM increases exponentially and as a result, SAPF can analyse up to four sequences. While the potential benefit of adding more sequence data is highly dependent on the evolutionary distances between species in the dataset, recent simulation studies have demonstrated how greater numbers of species can increase the specificity of functional element recognition [70, 74]. Additionally, [74] proposed the simple rule that for a given evolutionary distance, the number of genomes scales inversely for element length. Therefore, while two genomes may be sufficient for detecting long conserved exons, three to fifteen genomes may be needed to confidently detect TFBS.

To address this problem, Satija *et al.* (unpublished manuscript) have created BigFoot: a statistical aligner and phylogenetic footprinter that works on large datasets. BigFoot utilises Markov Chain Monte Carlo methods to simultaneously sample from a posterior distribution over alignments and annotations. The authors demonstrate how adding additional sequences can significantly improve predictive accuracy, especially with regards to sensitivity to weakly conserved binding sites, and the nucleotide resolution for predicting the exact boundaries of the TFBS. For datasets in both vertebrates and *Drosophila*, BigFoot outperforms traditional alignment-based phylogenetic footprinting tools by correcting for alignment error and ambiguity. The authors also report that the joint model improves the accuracy of the multiple sequence aligner by grouping together long segments of weakly conserved bases to correctly align an entire binding site, while a naive aligner may scatter different regions of the binding site into multiple regions of the overall alignment.



**Fig. 5.** BigFoot results for the *eve* stripe 2 enhancer when analysing four homologous sequences from four *Drosophila* species and when analyzing ten homologous sequences from ten *Drosophila* species including the previous four. For each nucleotide in the *D. melanogaster* sequence, Bigfoot outputs the probability that the nucleotide is part of a functional element and is subject to purifying selection. Experimentally verified binding sites in *D. melanogaster* for the transcription factors, Bicoid (bcd), Hunchback (hb), Kruppel (kr), Giant (gt), and Sloppy-paired 1 (s11) are shown above the posterior probabilities. Increasing the number of species in the analysis results in higher posterior probabilities in many experimentally verified binding sites, and increases the nucleotide resolution when identifying the precise locations for the TFBS.

In order to contain the complexity of the overall model, both SAPF and BigFoot assume that all sequences in an alignment column must contain the same annotation as either slowly or quickly evolving. A useful improvement to these models would relax this constraint, allowing the model to appropriately represent the gain and loss of functional regions in parts of the tree. Other potential improvements might involve an increase in complexity of the binding site model. SAPF and BigFoot model TFBS as slowly evolving regions. While this simplistic approach has been shown to successfully detect functional regions, more advanced models, such as position state weight matrices, that take into account sequence specific features may increase predictive power if they can be applied to multiple sequences.

## 4 Challenges for statistical alignment

### 4.1 Many Sequences

The advantage of including more sequences in an alignment based investigation has already been highlighted, with multiple alignment better capturing prediction reliability for protein secondary structure predictions [30] and allowing reliable detection of shorter features [74]. However, adding more sequences significantly increases the complexity of the alignment problem [75]. This necessitates the use of heuristics, that will make the multiple statistical alignment problem more tractable but at the expense of only offering an approximate solution. Here we discuss the possible approaches and associated challenges when the underlying methodology is solving the recursions of

statistical alignment [13] by dynamic programming, postponing sampling based approaches to a later point.

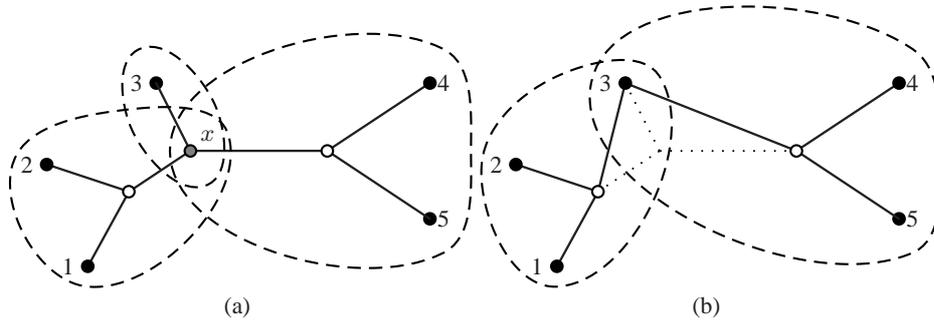
**Approximating the summation** The fundamental technique used for solving (multiple) statistical alignment, whether finding the optimal alignment or the data probability, is dynamic programming similar to basic score-based (multiple) alignment methods. For a data set  $S$  of  $n$  sequences dynamic programming tables of dimension  $n$  are populated with partial results. These partial results corresponds to alignments of initial parts of the sequences in  $S$  [43, 76] (or, for some algorithms, terminal parts), with separate tables capturing the different states the alignment process can be in. This technique results in time and, often more problematically, space requirements of  $\Omega(L^n)$ , requirements that become prohibitively steep for even a modest number of sequences. Evidently there is no way to sidestep this complexity without eliminating large chunks of the dynamic programming tables from the computation. This may still produce the correct optimal alignment, but for data probability we will inevitably ignore some contributions in the sum over all possible evolutionary histories. The challenge is to identify large parts of the dynamic programming tables that can be ignored with only a negligible effect on the final result.

Without large insertion or deletion events, at any point in a plausible alignment the fraction of each sequence to the left of the point will usually be roughly the same. This observation is utilised in the simplest way of restricting the dynamic programming tables, known as *banding* [77]. Here positions in different sequences are only allowed to align if their difference is less than some predetermined bound. The banding technique is available in the HMM compiler HMMoC [78] and subsequently used for statistical alignment in [79]. This works well as long as all plausible alignments fall in a narrow band along the main diagonal of the dynamic programming tables. However, as overall insertion and deletion rates increase a wider and wider band has to be applied to capture all plausible alignments sufficiently well, diminishing the gains in efficiency.

A slightly more intelligent way to define banding, building on decades of progress on speeding up score-based multiple alignment, uses an optimal score-based alignment rather than the main diagonal to define the course of the band alignments are restricted to [80]. Though banding around an optimal score-based alignment spine goes some way to adapt to the input data, it will still fail when there are large regions where the alignment is poorly defined or when there are alternative close to optimal alignments with significantly different trajectories in the dynamic programming tables, e.g. if the input sequences contain repeats. A fully adaptive restriction should attempt to allow all plausible alignments to be considered, rather than just alignments close to one predetermined or inferred alignment.

For real data a plausible multiple alignment will usually induce a plausible pairwise alignment for any pair of sequences. Conversely, if we identify all the plausible pairwise alignments for all pairs of sequences, formalised as *alignment envelopes* in [81], this defines a natural restriction on the dynamic programming tables – only consider entries that result in plausible alignment columns in all (or most) induced pairwise alignments, see e.g. [82]. As we progress up through the evolutionary tree relating the input sequences, however, pairwise alignments may become increasingly unreliable as the basis for restricting the dynamic programming tables.

Despite the dynamic programming approach to multiple statistical alignment sketched above not being a progressive alignment method, it may be advantageous to use a progressive approach, in particular since evolution along the branches of a tree is an inherent part of the model. Solving the statistical alignment problem for the full data set will



**Fig. 6.** Breaking sequence dependencies. (a) If the sequence at internal node  $x$  is known, the full data set probability decomposes into a product of independent probabilities of the three subsets indicated; this probability is conditional on the sequence at  $x$ . (b) Perturbing the tree by making node 3 an internal node allows us to decompose the unconditional data set probability into just two independent probabilities of the subsets indicated, albeit under a modified tree relating the sequences.

usually be by far more demanding than solving it for the sequences in any subtree of the assumed evolutionary tree. Hence, it is worthwhile to align sequences of subtrees at lower levels in a progressive manner to guide the restrictions at higher levels. More formally, for any node in the evolutionary tree first recursively align the sequences in each subtree of the node, extract alignment envelopes for each subtree, merge the alignment envelopes into a restriction on the dynamic programming tables for all sequences in the (sub)tree rooted at the node, and finally align the sequences subject to the table restrictions. In [43] it is discussed how restrictions can effortlessly be incorporated into the dynamic programming computations, mentioning restriction to a single alignment as example. The approach also bears close resemblances to the TreeAlign algorithm for score-based alignment [83], with alignment envelopes replacing alignment graphs.

**Approximating the phylogeny** The reason for the steep growth in time and space requirements of multiple sequence statistical alignment is the inherent dependency between a set of sequences related by a tree. Without knowledge of the state of the internal nodes in the tree, every possible column aligning positions from some or all sequences is possible and should be considered in an exact approach. With gaps between two consecutive positions in a sequence also corresponding to a unique position, there are  $\Theta(2^n L^n)$  possible alignment columns so simply enumerating them would account for the complexity of the problem in itself. One approach to circumvent this problem is to restrict the set of alignment columns considered, as outlined above. This will work for a while, but it is hard to imagine a restriction sufficiently judicious to avoid the exponential growth with number of sequences while still maintaining good approximation: imagine extending a data set of  $n$  sequences with one extra sequence by augmenting plausible alignment columns for the  $n$  sequences with positions from the new sequences – if on average each plausible alignment column has more than 1 plausible augmentation, we will still see a growth exponential in the number of sequences.

An alternative approach is to try to break the dependencies between the input sequences. If we knew the state, *i.e.* the sequence, at an internal node of the tree relating the sequences, the alignment problem would decompose into independent alignment

problems for the subtrees attached to this internal node, cf. Fig. 6(a): due to the Markov property of the evolutionary model, the sequences in different subtrees will be independent conditional on the sequence in the internal node. In most cases it will be infeasible to attempt to utilise this by fixing a (small set of) plausible sequence(s) at one or more internal nodes. Even ignoring variation in length of plausible sequences, if just a constant fraction of the positions have two or more choices of plausible characters we would again see an exponential growth, this time in sequence length, of plausible candidates we need to consider to expect a good approximation.

For some nodes in the tree we only have one plausible sequence, however. These are the leaf nodes corresponding to the observed sequences. So by perturbing the tree to make some of the nodes corresponding to observed sequences internal, as illustrated in Fig. 6(b), we can break dependencies without having to consider exponential large sets of plausible sequences at any nodes. For example, in Fig. 6 the multiple statistical alignment problem on five sequences is decomposed into two independent multiple statistical alignment problems on three sequences.

The drawback is that we are using a different tree to relate the input sequences. We can to some extent assess the effect this will have. The multiple alignment problem is a special case of the problem known as the Steiner tree problem: given a universe  $\mathcal{U}$  and a set of points  $S \subset \mathcal{U}$ , find a tree of minimal length that connects all the points in  $S$ . In the multiple alignment case,  $\mathcal{U}$  corresponds to all finite sequences. A tree connecting the points in  $S$  is called a  $k$ -restricted Steiner tree if the subtrees obtained by splitting the tree at all internal nodes corresponding to an element of  $S$  each contains at most  $k$  elements from  $S$ . So the tree in Fig. 6(a) corresponds to an unrestricted Steiner tree on the five observed sequences, while the tree in Fig. 6(b) corresponds to a 3-restricted Steiner tree. Constructing a  $k$ -restricted Steiner tree from unrestricted Steiner trees on subsets of at most  $k$  elements is the state-of-the-art approach to approximating the minimum Steiner tree length. Robins and Zelikovsky [84] presents an algorithm finding a  $k$ -restricted Steiner tree of length at most 1.55 times the minimum length of an unrestricted Steiner tree. The approximation ratio of 1.55 is only obtained for very large  $k$ , though; for values of  $k$  more realistic in the multiple alignment setting the ratio is somewhere between 1.8 and 1.9. Furthermore, the problem considered is for additive distance, rather than probabilities that are multiplied. We can convert to additive distances by changing to negative log-scale, but this also changes the ratio to an exponent – the guarantee becomes computing a probability of at least  $p^{1.55}$  where  $p$  is the true probability of the input sequences. Still, even if the exact probability is not adequately approximated, it may not be unreasonable to assume a strong correlation between approximated and true distributions of alignments and model parameters.

Another consequence of promoting observed sequences to internal nodes is that all possible alignments are no longer possible. Similarly to progressive alignment methods, initial choices – in this case of which observed sequences to promote – will to some extent dictate the possible future alignments. For example, if we are using the tree depicted in Fig. 6(b) to relate the input sequences, alignments postulating homology between characters in e.g. sequences 1 and 5 but with no homologous character in sequence 3 will not be possible. To some extent this problem might be alleviated by marginalising over or sampling a set of  $k$ -restricted trees relating the input sequences. However, this will introduce a new challenge of extracting a representative alignment, whether it is the most likely alignment or the alignment maximising the expected number of correct alignment columns. When only one tree is used, these problems can be solved independently for each subset of interdependent sequences, and then using the observed sequences at internal nodes as scaffolds for merging alignments. With multiple

trees we need to optimise over the combined effect, which does not have a straightforward solution – exhaustive enumeration of possible alignments or alignment columns will again introduce exponentially large sets of possibilities to consider.

## 4.2 Realistic Models

The TKF92 model is a substantial improvement over the TKF91 model, as it allows indel events involving more than one nucleotide. The main assumptions of the model are that

- indel events do not overlap, and
- 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. Below we focus on relaxing the former assumption, although the proper modelling of the actual indel length distribution (see e.g. [85]) is very likely at least as important for alignment accuracy. We refer to this 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 probability of the total outcome does not factorise 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 [35] 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.

An even more realistic approximation to the long indel model is possible, although it needs computationally more demanding algorithms. In [10] 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 [10] for more details.

Another important issue is that substitutions are not independent from each other. In nucleotide sequences, it is well known that the substitution rate of a nucleotide depends on the neighbours. The most known example is that the CpG motifs are frequently methylated, making the methylated C mutate more frequently to T. The context-dependent substitution on its own is very hard to model, see for example [86–88]. To our best knowledge, there is no published model and corresponding inference tool that could model both context-dependent substitution and insertion-deletion process.

In case of proteins, the substitution rate depends on the secondary structure [58] and also on the interaction amongst amino acids that are close in the three dimensional space. Since such amino acids might be at different positions in the sequence, it is again extremely hard to develop a computationally tractable model. Indeed, even modelling the substitutions are very hard, and the statistical inference of such a model needs

Markov chain Monte Carlo methods, since analytical solutions for the proposed model are not known [89].

In case of RNA sequences, the substitution rates are different in single and double stranded parts. The base-pairing nucleic acids are co-evolved which can be easily modelled with a dinucleotide substitution model, if the secondary structure is known. However, adding structure-dependence to the insertion-deletion process is not easy at all. For two sequences, Holmes introduced a model and corresponding algorithm for likelihood calculation [90] that has subsequently been accelerated [81]. However, this model aligns sequences to a Stochastic Context Free Grammar instead of explicitly modelling the sequence evolution of RNA sequences [91]. This allows an accelerated likelihood calculation with  $O(L^3)$  running time, where  $L$  is the length of the sequences. This is a significant improvement compared with the running time of an algorithm that calculates the likelihood for an explicit model. The running time of this later algorithm is  $O((L_1 L_2)^3)$ , which is definitely computationally intractable.

## 5 Conclusions

Although statistical alignment has made enormous progress in recent years, its clear superiority over optimisation based alignment methods also makes it imperative that it takes on further challenges, where methods haven't been developed yet. There are at least 6 major areas to be pursued: aligning very large numbers of sequences, possibly even thousands; aligning complete genomes, preferably beyond pairwise alignment; local statistical alignment; combining statistical alignment with annotation problems; more realistic substitution and indel models; Lastly, combining statistical alignment with other data types than sequences and using existing knowledge. In more detail, work in these areas should address the following:

1. Aligning a large number of sequences is a necessity, since very large number of sequences will be available and it will be ideal to analyse all data. Even when the problem at hand could be solved by analysing a smaller number of sequences, it is convenient that the smaller subset is selected by an algorithm and not by a laborious series of analysis and re-analysis by a user.
2. Many applications focus on complete genomes and taking a stochastic modelling approach would here have major advantages. This can be pursued in different ways. A realistic approach is to use existing tools for genome alignment and then re-align using the more advanced method of statistical alignment. Finding the homologous regions in different genomes might be hard due to possible paralogs [92]. Realignment defines a neighbourhood around a given alignment and explores that in great depth. Such an approach is being pursued by Lunter and Satija (unpublished manuscript). This has not been generalised to more genomes, but clearly would be feasible for at least a small handful of genomes. Aligning genomes where individual genes are viewed as atomic, also known as genome rearrangement, has been addressed both as an optimisation problem and as a statistical inference problem. Combining the two levels of statistical alignment – insertions, deletions and substitutions within genes and gene rearrangement and copying at the genome level – is theoretically possible, but at present likely to be computationally infeasible.
3. The framework of statistical alignment has at present no concept corresponding to local alignment, which is of great value in sequence comparison and at the core of programs such as BLAST. The natural model for this would imply large scale deletions-insertions of random sequences, that would leave islands of homology

behind. Whether it is possible to do inference in such a model remains to be seen and possibly a model would have to be tailored around making computations feasible. Extending the statistical alignment framework to local alignment is a real and practical issue in data analysis, however, for instance when searching upstream regions of genes for regulatory signals.

4. Annotation is one of the big areas of sequence analysis, where key categories are: protein genes, RNA secondary structure, regulatory signals and finally the nature of selection on a specific position. Normally this is done by a stochastic description of the item of annotation as a hidden structure, where the state of the observation (sequences) depends on the state of the hidden structure. Markov Models and Stochastic Context Free Grammars are popular in this context. Normally a pre-given alignment is assumed. However, in principle and to great advantage annotation and alignment could be combined. Section 3.3 considers the simplest possible such combination, where alignment has been combined with annotation of fast and slow for each state. This combination is based on an approximation, not an explicit model of the evolution of a sequence with annotation. However, for most purposes this approximation is most likely very good. This application has unambiguously illustrated the value of summing out alignments. Thus, combining alignment and other kinds of annotation would be highly valuable.
5. Statistical alignment has proven so computationally challenging, that investigating goodness of fit of existing models has had second priority. At present this prioritisation seems justified. This is equivalent to the statement, that it is better to analyse many sequences with a slightly incorrect model, than few sequences with the correct model. However, at some point details about the substitution and insertion-deletion process must be re-investigated and the statistical framework of statistical alignment required again to prove its worth by answering questions about these processes unanswerable by optimisation alignments.
6. The analysis of most biological problems, will use a variety of approaches and increasingly a “sequences only” approach will be a rarity. Alignment and modelling of sequences with no perceived annotation of relevance, *i.e.* having no function, is of limited interest. For functional sequences, experimental data relating to this function may be available, and formulating models combining sequence evolution with functional impact of sequences can significantly strengthen inference. For instance, co-observing expression levels could be very informative when looking for signals controlling mRNA expression. Other obvious examples of this would be knowledge of transcripts for protein gene annotation, base pairing information for RNA structure annotation and ChIP-on-chip information for regulatory signal annotation.

Other areas of extensions could be suggested, but at least some would be too hard to address by anything but heuristic means for the foreseeable future. One example could be combining alignment and recombination, replacing the evolutionary tree relating sequences with structures capturing recombination. Both phenomena are important in viruses and virus analysis could benefit from a combined solution. However, each of these problems are current challenges in their own right, and attempts at combining them may seem overly ambitious at the present time.

Modelling and algorithmic advances will only be of value to the community if it results in flexible and user-friendly software. Making software that integrates all components will be a major, but necessary challenge. The present packages took years to develop and can handle at most 15 sequences and are in general based on the most basic models (TKF91/92) and possibly simple rate heterogeneity. Clearly, what is needed is a

software organisation that can be improved as new models and algorithms are described and as new modules are written. StatAlign [28] takes some steps in this direction, but much more is needed.

The continued rise of statistical alignment will be very interesting to follow. Starting with the Needleman-Wunsch algorithm in 1970, the score-based approach had almost total dominance for three decades. Statistical alignment only arose as an idea in 1986 [93], but was initially of little use and formalised by a basic model containing several flaws. In the early 1990s Thorne, Kishino and Felsenstein introduced more satisfactory models [9, 18]. Usefulness was still limited by difficulties in introducing even more realistic models and the lack of a multiple sequence solution. Hidden Markov Models were introduced into sequence alignment in 1994 [2], but in a totally non-evolutionary way. Around 2000 breakthroughs in multiple statistical alignment and discovery of the use of HMM and transducer theory in model development saw the use and research in statistical alignment take off in a major way. We are clearly only at the beginning of this revolution in sequence analysis.

## 6 Acknowledgements

This work was supported by BBSRC grant BB/C509566/1. I.M. was also supported by a Bolyai postdoctoral fellowship and an OTKA grant F 61730.

## References

1. Needleman S, Wunsch C. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol.* 1970;48(3):443–53.
2. Krogh A, Brown M, Mian I, Sjolander K, Haussler D. Hidden Markov models in computational biology: Applications to protein modeling. *J Mol Biol.* 1994;235:1501–1531.
3. Smith T, Waterman M. Identification of common molecular subsequences. *J Mol Biol.* 1981;147(1):195–7.
4. Altschul S, Gish W, Miller W, Myers E, Lipman D. Basic local alignment search tool. *J Mol Biol.* 1990;215(3):403–410.
5. Fitch W. Toward defining the course of evolution: minimum change for a specific tree topology. *Syst Zool.* 1971;20:406–416.
6. Hartigan J. Minimum evolution fits to a given tree. *Biometrics.* 1973;29:53–65.
7. Felsenstein J. Evolutionary trees from DNA sequences: A maximum likelihood approach. *Journal of Molecular Evolution.* 1981;17(6):368–376.
8. Felsenstein J. The troubled growth of statistical phylogenetics. *Systematic Biology.* 2001;50:465–467.
9. Thorne J, Kishino H, Felsenstein J. An evolutionary model for maximum likelihood alignment of DNA sequences. *J Mol Evol.* 1991;33(2):114–24.
10. Miklós I, Lunter GA, Holmes I. A 'long indel' model for evolutionary sequence alignment. *Mol Biol Evol.* 2004;21(3):529–540.
11. Steel M, Hein J. Applying the Thorne-Kishino-Felsenstein model to sequence evolution on a star-shaped tree. *Appl Math Let.* 2001;14:679–684.
12. Hein J. An Algorithm for Statistical Alignment of Sequences Related by a Binary Tree. In: *Pacific Symposium on Biocomputing.* vol. 6; 2001. p. 179–190.
13. Hein J, Jensen J, Pedersen C. Recursions for statistical multiple alignment. *PNAS.* 2003;100(25):14960–14965.
14. Lunter G, Miklós I, Drummond A, Jensen J, Hein J. Bayesian phylogenetic inference under a statistical indel model. *Lecture Notes in Bioinformatics.* 2003;2812:228–244.
15. Lunter G, Miklós I, Drummond A, Jensen J, Hein J. Bayesian Coestimation of Phylogeny and Sequence Alignment. *BMC Bioinformatics.* 2005;6:83.

16. Churchill G. Monte Carlo Sequence Alignment. In: Proceedings of RECOMB97; 1997. p. 93–97.
17. Holmes I, Bruno W. Evolutionary HMMs : a Bayesian approach to multiple alignment. *Bioinformatics*. 2001;17(9):803–820.
18. Thorne J, Kishino H, Felsenstein J. Inching toward reality: an improved likelihood model of sequence evolution. *J Mol Evol*. 1992;34(1):3–16.
19. Gotoh O. An improved algorithm for matching biological sequences. *J Mol Biol*. 1982;162:705–708.
20. Felsenstein J. Evolutionary trees from DNA sequences : a maximum likelihood approach. *J Mol Evol*. 1981;17:68–376.
21. Metzler D. Statistical alignment based on fragment insertion and deletion models. *Bioinformatics*. 2003;19(4):490–499.
22. Goldman N. Effects of sequence alignment procedures on estimates of phylogeny. *BioEssays*. 1998;20:287–290.
23. Wong K, Suchard M, Huelsenbeck J. Alignment uncertainty and genomic analysis. *Science*. 2008;319(5862):473–6.
24. Rannala B, Yang Z. Probability distribution of molecular evolutionary trees: a new method of phylogenetic inference. *J Mol Evol*. 1996;43:304–311.
25. Drummond A, Nicholls G, Rodrigo A, Solomon W. Estimating Mutation Parameters, Population History and Genealogy Simultaneously From Temporally Spaced Sequence Data. *Genetics*. 2002;161(3):1307–1320.
26. Durbin R, Eddy S, Krogh A, Mitchison G. Biological sequence analysis. Probabilistic models of proteins and nucleic acids. Cambridge University Press; 1998.
27. Huson D, Bryant D. Application of Phylogenetic Networks in Evolutionary Studies. *Mol Biol Evol*. 2006;23(2):254–267.
28. Novák Á, Miklós I, LyngsøR, Hein J. StatAlign: An Extendable Software Package for Joint Bayesian Estimation of Alignments and Evolutionary Trees. *Bioinformatics*. 2008;.
29. Mizuguchi K, Deane C, Blundell T, JP O. HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Science*. 1998;7:2469–2471.
30. Miklós I, Novák Á, Dombai B, Hein J. How reliably can we predict the reliability of protein structure predictions? *BMC Bioinformatics*. 2008;9:137.
31. Holland B, Moulton V. Consensus Networks: A Method for Visualising Incompatibilities in Collections of Trees. *Lecture Notes in Computer Science, Proceedings of WABI2003*. 2003;2812:165–176.
32. Waterman M, Smith T, Beyer W. Some biological sequence metrics. *Advan Math*. 1976;20:367–387.
33. Waterman M. Parametric and ensemble sequence alignment algorithms. *Bulletin of Mathematical biology*. 1994;5(4):743–767.
34. Kececioğlu J, Kim E. Simple and Fast Inverse Alignment. *Lecture Notes in Computer Science*. 2006;3909:441–455.
35. Knudsen B, Miyamoto M. Sequence alignments and pair hidden Markov models using evolutionary history. *J Mol Biol*. 2003;333:453–460.
36. Löytynoja A, Milinkovitch M. A hidden Markov model for progressive multiple alignment. *Bioinformatics*. 2003;19(12):1505–1513.
37. Wang L, Jiang T. On the complexity of multiple sequence alignment. *J Comp Biol*. 1994;1(4):337–348.
38. Karplus K, Barrett C, Hughey R. Hidden Markov Models for Detecting Remote Protein Homologies. *Bioinformatics*. 1998;14(10):846–856.
39. Eddy S. Profile Hidden Markov Models. *Bioinformatics*. 1998;14:755–763.
40. Hogeweg P, Hesper B. The alignment of sets of sequences and the construction of phyletic trees: An integrated method. *J Mol Evol*. 1984;20(2):175–186.
41. Feng D, Doolittle R. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J Mol Evol*. 1987;25:351–360.
42. Löytynoja A, Goldman N. An algorithm for progressive multiple alignment of sequences with insertions. *PNAS*. 2005;102(30):10557–10562.

43. Holmes I. Using guide trees to construct multiple-sequence evolutionary HMMs. *Bioinformatics*. 2003;19(90001):147–157.
44. Bradley R, Holmes I. Transducers: An Emerging Probabilistic Framework for Modeling Indels on Trees. *Bioinformatics*. 2007;Doi:10.1093/bioinformatics/btm402.
45. Metzler D, Fleissner R, von Haeseler A, Wakolbinger A. Assessing variability by joint sampling of alignments and mutation rates. *J Mol Evol*. 2001;53:660–669.
46. Fleissner R, Metzler D, von Haeseler A. Simultaneous Statistical Multiple Alignment and Phylogeny Reconstruction. *Systematic Biology*. 2005;54:548–561.
47. Redelings B, Suchard M. Joint Bayesian estimation of alignment and phylogeny. *Syst Biol*. 2005;50:401–418.
48. Suchard M, Redelings B. BAli-Phy: simultaneous Bayesian inference of alignment and phylogeny. *Bioinformatics*. 2006;22(16):2047–2048.
49. Metropolis N, Rosenbluth A, Rosenbluth M, Teller A, Teller E. Equations of state calculations by fast computing machines. *J Chem Phys*. 1953;21(6):1087–1091.
50. Hastings W. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*. 1970;57(1):97–109.
51. Ronquist F, Huelsenbeck J. MrBayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics*. 2003;19(12):1572–1574.
52. Mizuguchi K, Deane C, Johnson M, Blundell T, Overington J. JOY: protein sequence-structure representation and analysis. *Bioinformatics*. 1998;14:617–623.
53. Gusfield D. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge University Press; 1997.
54. Hubbard T, Lesk A, Tramontano A. Gathering them into the fold. *Nature Structural Biology*. 1996;3:313.
55. Skolnick J, Kolinski A, Kihara D, Betancourt M, Rotkiewicz P, M B. Ab initio protein structure prediction via a combination of threading, lattice folding, clustering, and structure refinement. *Proteins*. 2002;44(S5):149–156.
56. Wu S, Skolnick J, Zhang Y. Ab initio modeling of small proteins by iterative TASSER simulations. *BMC Biology*. 2007;5:17.
57. Zhou H, Skolnick J. Ab Initio Protein Structure Prediction Using Chunk-TASSER. *Biophysical Journal*. 2007;93:1510–1518.
58. Goldman N, Thorne J, Jones D. Using Evolutionary Trees in Protein Secondary Structure Prediction and Other Comparative Sequence Analyses. *J Mol Biol*. 1996;263(2):196–08.
59. Kneller D, Cohen F, Langridge R. Improvements in Protein Secondary Structure Prediction by an Enhanced Neural Network. *J Mol Biol*. 1990;214:171–182.
60. Garnier J, Gibrat JF, B R. GOR secondary structure prediction method version IV. *Methods in Enzymology*. 1996;266:540–553.
61. Stark A, Lin M, Kheradpour P, Pedersen J, Parts L, Carlson J, et al. Discovery of functional elements in 12 *Drosophila* genomes using evolutionary signatures. *Nature*. 2007;450(7167):219–232.
62. Cliften P, Sudarsanam P, Desikan A, Fulton L, Fulton B, Majors J, et al. Finding Functional Features in *Saccharomyces* Genomes by Phylogenetic Footprinting. *Science*. 2003;301(5629):71–76.
63. Boffelli D, McAuliffe J, Ovcharenko D, Lewis K, Ovcharenko I, Pachter L, et al. Phylogenetic Shadowing of Primate Sequences to Find Functional Regions of the Human Genome. *Science*. 2003;299(5611):1391–1394.
64. Wasserman W, Palumbo M, Thompson W, Fickett J, Lawrence C. Human-mouse genome comparisons to locate regulatory sites. *Nature Genetics*. 2000;26:225–228.
65. Siepel A, Bejerano G, Pedersen J, Hinrichs A, Hou M, Rosenbloom K, et al. Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome Research*. 2005;15(8):1034.
66. Tagle D, Koop B, Goodman M, Slightom J, Hess D, Jones R. Embryonic epsilon and gamma globin genes of a prosimian primate (*Galago crassicaudatus*). Nucleotide and amino acid sequences, developmental regulation and phylogenetic footprints. *J Mol Biol*. 1988;203(2):439–55.

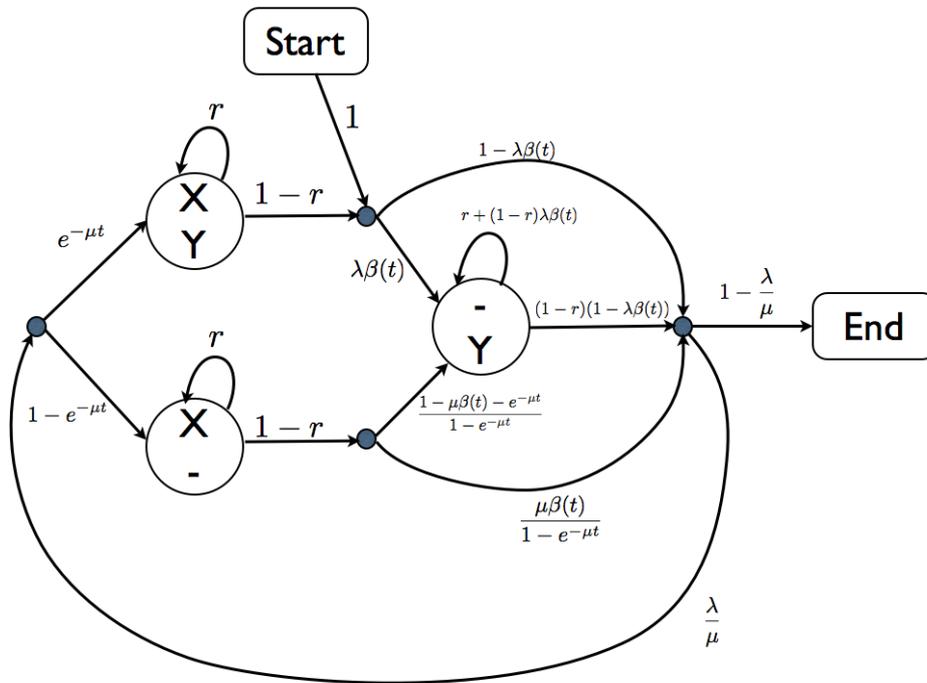
67. GuhaThakurta D. Computational identification of transcriptional regulatory elements in DNA sequence. *Nucleic Acids Research*. 2006;34(12):3585.
68. Pollard D, Moses A, Iyer V, Eisen M. Detecting the limits of regulatory element conservation and divergence estimation using pairwise and multiple alignments. *BMC Bioinformatics*. 2006;7:376.
69. Lunter G, Rocco A, Mimouni N, Heger A, Caldeira A, Hein J. Uncertainty in homology inferences: Assessing and improving genomic sequence alignment. *Genome Res*. 2007;.
70. Fan X, Zhu J, Schadt E, Liu J. Statistical power of phylo-HMM for evolutionarily conserved element detection. *BMC Bioinformatics*. 2007;8:374.
71. Zhu J. Bayesian adaptive sequence alignment algorithms. *Bioinformatics*. 1998;14(1):25–39.
72. Sinha S, He X. MORPH: Probabilistic alignment combined with hidden Markov models of cis-regulatory modules. *PLoS Comput Biol*. 2007;10.
73. Satiya R, Pachter L, Hein J. Combining statistical alignment and phylogenetic footprinting to detect regulatory elements. *Bioinformatics*. 2008;24(10):1236.
74. Eddy S. A model of the statistical power of comparative genome sequence analysis. *PLoS Biology*. 2005;3(1):e10.
75. Lunter G, Drummond A, Miklós I, Hein J. *Statistical Alignment: Recent Progress, New Applications, and Challenges*. *Statistical Methods in Molecular Evolution*. 2005;.
76. Lunter G, Miklós I, Song Y, Hein J. An efficient algorithm for statistical multiple alignment on arbitrary phylogenetic trees. *J Comp Biol*. 2003;10(6):869–889.
77. Chao KM, Pearson W, Miller W. Aligning two sequences within a specified diagonal band. *Computer Applications in the Biosciences (CABIOS)*. 1992;8(5):481–487.
78. Lunter G. HMMoC—a compiler for hidden markov models. *Bioinformatics*. 2007;23(18):2485–2487.
79. de Groot S, Mailund T, Lunter G, Hein J. Investigating selection on viruses: a statistical alignment approach. *BMC Bioinformatics*. 2008;9:304.
80. Hein J, Wiuf C, Knudsen B, Moller M, Wibling G. Statistical alignment: computational properties, homology testing and goodness-of-fit. *J Mol Biol*. 2000;302:265–279.
81. Holmes I. Accelerated probabilistic inference of RNA structure evolution. *BMC Bioinformatics*. 2005;6:73.
82. Do C, Mahabhashyam M, Brudno M, Batzoglou S. ProbCons: probabilistic consistency-based multiple sequence alignment. *Genome Research*. 2005;15:330–340.
83. Hein J. Unified approach to alignment and phylogenies. *Methods in Enzymology*. 1990;183:626–645.
84. Robins G, Zelikovsky A. Improved steiner tree approximation in graphs. *Proceedings of the 11th Annual Symposium on Discrete Algorithms (SODA)*. 2000;.
85. Qian B, Goldstein R. Distribution of indel lengths. *Proteins: struc func gen*. 2001;45:102–104.
86. Lunter G, J H. A nucleotide substitution model with nearest-neighbour interactions. *Bioinformatics*. 2004;20:i216–i223.
87. Arndt P, CB B, Hwa T. DNA sequence evolution with neighbor-dependent mutation. *J Comp Biol*. 2003;10:313–322.
88. Pedersen AM, Jensen J. A dependent rates model and MCMC based methodology for the maximum likelihood analysis of sequences with overlapping reading frames. *Mol Biol Evol*. 2001;18:763–776.
89. Robinson D, Jones D, Kishino H, Goldman N, Thorne J. Protein evolution with dependence among codons due to tertiary structure. *MolBiol Evol*. 2003;20:1692–1704.
90. Holmes I. A probabilistic model for the evolution of RNA structure. *BMC Bioinformatics*. 2004;5:166.
91. Holmes I, Rubin G. Pairwise RNA structure comparison using stochastic context-free grammars. In: *Pacific Symposium on Biocomputing*; 2002. .
92. Nye T. Modelling the evolution of multi-gene families. *Statistical Methods in Medical Research*. 2008;.
93. Bishop M, Thompson E. Maximum likelihood alignment of DNA sequences. *J Mol Biol*. 1986;190(2):159–165.

94. Liu J. Monte Carlo strategies in scientific computing. Springer; 2001.  
 95. Green P. Reversible jump Markov Chain Monte Carlo computation and Bayesian model determination. Biometrika. 1995;82:711–732.

## A Technical Appendix

### A.1 HMM Connections

Both the TKF91 and the TKF92 models can be transformed into a pair Hidden Markov Model, see Fig. 7 for the pair-HMM representing the TKF92 model. The transition and emission probabilities of the pair-HMM are parametrised with the parameters of the TKF92 model, and for any pair of sequences  $A$  and  $B$ , the emission probability of  $A$  as root sequence and  $B$  as derived sequence by the pair-HMM equals the observation probability of  $A$  and  $B$  in the TKF92 model. Moreover, the Viterbi path of the pair-HMM represents the most likely alignment of the two sequences in the TKF92 model.



**Fig. 7.** The TKF92 model [18], interpreted as a Markov model.  $\lambda$  is the insertion rate,  $\mu$  is the deletion rate,  $r$  is the parameter of the geometric distribution of inserted and deleted fragments, and  $\beta(t) = \frac{1 - e^{-(\lambda - \mu)t}}{\mu - \lambda e^{-(\lambda - \mu)t}}$ . Emission probabilities of character  $X$  to the root sequence and/or character  $Y$  to the derived sequence are given by the continuous-time substitution model of the TKF92 model: the probability of joint emission of two characters equals the joint observation probability of two characters in the substitution model and single emission follows the equilibrium distribution of the substitution process.

Holmes and Bruno [17] showed how to construct a multiple-HMM describing the evolution of an ancestral sequence and its descendants evolving on an evolutionary tree. Rather than giving a rigorous proof why this stochastic process can be described as a multiple-HMM, we explain it on a simple example for three sequences, see Fig.8. The extension to any number of sequences and evolutionary tree should be obvious, although the technical details are quite tedious.

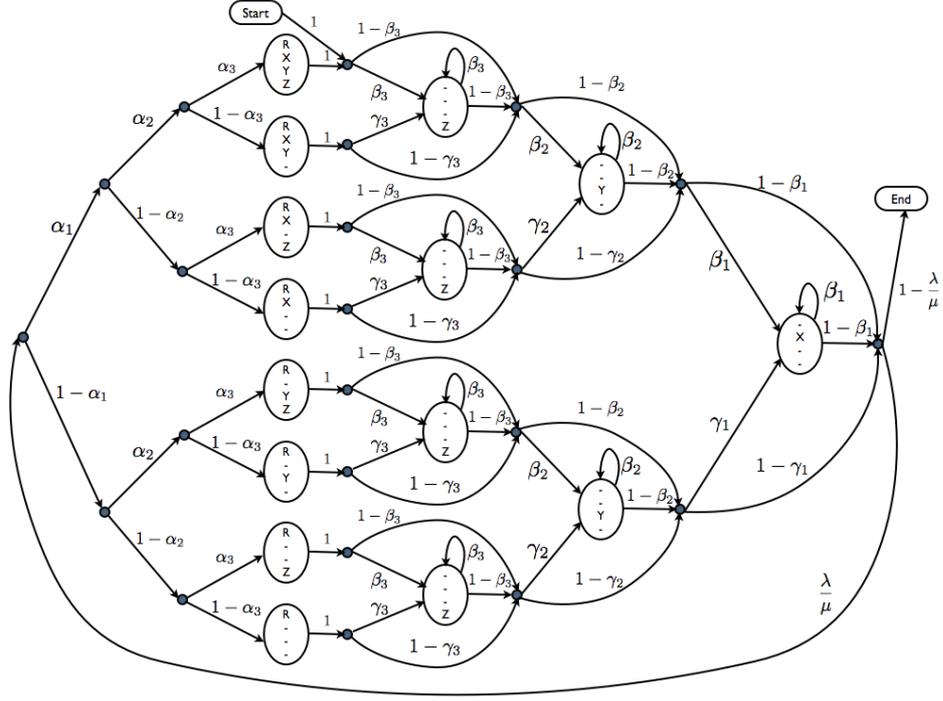
First, note that each path from the start state to the end state corresponds to a multiple alignment. From the start state, the chain first jumps to a silent state next to the state emitting a character to all sequences. This jump models “births” emanating from the immortal link. Eventually, after considering independent insertions on each branch of the star-tree, 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  $\alpha_i$  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 states deleting a root sequence character on one or more branches in the HMM of Fig.8, as was done for the deletion state in Fig. 7. This fixes fragmentations over the 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 normalised pair HMMs describing the evolution along a branch, which provides an algorithmic way to construct multiple-HMMs on a tree [43, 44]. 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.

## A.2 Theoretical Background of MCMC

The optimal multiple sequence alignment problem with sum-of-pairs scoring scheme is proven to be NP-hard [37]. Although a similar proof does not exist for the case of multiple statistical alignment, it is a strong conjecture that the multiple statistical alignment problem is also NP-hard. Therefore, there is little hope for a fast algorithm that guarantees optimal multiple alignment in a statistical alignment framework. In lieu of exact deterministic algorithms, stochastic optimisation methods, especially Markov chain Monte Carlo methods have been widespread. The Markov chain Monte Carlo methods construct a Markov chain that converges to a prescribed distribution. In case of statistical alignment, this prescribed distribution is typically the joint posterior distribution of multiple alignments, evolutionary trees and model parameters.

**The Metropolis-Hastings algorithm** Metropolis *et al.* [49] published the first Markov chain Monte Carlo algorithm that was investigated by Hastings [50], hence called the Metropolis-Hastings algorithm. The Metropolis-Hastings algorithm is used to construct a Markov chain that converges to a prescribed distribution  $\pi$  over state space  $S$ . The algorithm can tailor any Markov chain to converge to  $\pi$  if the following characteristics are true for the Markov chain:



**Fig. 8.** Multiple-HMM describing the evolution of three sequences related by a star-tree, under the TKF91 model. The following abbreviations are used:  $\alpha_i = e^{\mu t_i}$ ,  $\beta_i = \frac{\lambda - \lambda e^{(\lambda - \mu)t_i}}{\mu - \lambda e^{(\lambda - \mu)t_i}}$ ,  $\gamma_i = \frac{1 - e^{-\mu t_i} - \beta_i}{1 - \alpha_i}$ , where  $t_i$  is the length of the branch descending to vertex  $i$  in the phylogenetic tree. Big circles are states that emit the column shown according to the underlying substitution model; R, X, Y and Z represent characters in the root sequence and the three observed sequences, respectively. Small circles represent silent states. See also [17].

- If the Markov chain can step to  $y$  from  $x$ , then it also can step to  $x$  from  $y$ . More formally,

$$\forall x, y \in S : T(y | x) \neq 0 \Rightarrow T(x | y) \neq 0 \quad (14)$$

where  $T(y|x)$  is the probability that the chain moves to states  $y$  given that it is in state  $x$ .

- The chain is irreducible, *i.e.* there is a path with positive probability between any pair of states. If  $P$  is the transition matrix of the Markov chain, *i.e.*  $T(y | x) = P_{x,y}$ , then this property can be formalised as

$$\forall x, y \in S \exists n : \langle \mathbf{1}_x^T P^n | \mathbf{1}_y \rangle \neq 0 \quad (15)$$

where  $\mathbf{1}_x$  is a vector containing 1 in coordinate  $x$ , and 0 in all other coordinates,  $T$  denotes transposition, and  $\langle \cdot | \cdot \rangle$  denotes scalar product.

The Metropolis-Hastings algorithm is comprised of two steps. The first step is called *proposal*, in which a random  $y$  is drawn from the conditional distribution  $T(\cdot | x)$ . In the

second step, a random decision is made to decide whether or not the proposed  $y$  is accepted. A random number  $u$  is drawn from the uniform distribution  $U[0, 1]$  and the next state of the Markov chain is  $y$  if

$$u \leq \min \left\{ 1, \frac{\pi(y)T(x|y)}{\pi(x)T(y|x)} \right\} \quad (16)$$

otherwise the next state of the Markov chain also will be  $x$ . It is easy to show that the so-generated Markov chain converges to the prescribed distribution  $\pi$ , see *e.g.* [94].

**Partial importance sampling** Since the target distribution of MCMC is typically a high-dimensional, complicated, non-Euclidian space, the proposed new state of the chain is usually drawn by a series of random decisions. Consequently the same state  $y$  sometimes can be proposed from  $x$  in several different ways, and hence it is not so easy to calculate  $T(y|x)$  which is, by definition, the sum of the probabilities of the possible series of random decisions that yields  $y$  from  $x$ . When calculating this sum is computationally demanding, it might be worthwhile taking an alternative approach. If  $y$  can be proposed from  $x$  in several ways, for each possible way  $w$  there is a backproposal way  $w'$  yielding  $x$  from  $y$ , and the mapping from proposal to backproposal ways is a bijection, then the Metropolis-Hastings ratio in Eq. (16) can be replaced by

$$\min \left\{ 1, \frac{\pi(y)T(x, w'|y)}{\pi(x)T(y, w|x)} \right\} \quad (17)$$

where  $T(y, w|x)$  denotes the probability that state  $y$  is proposed from  $x$  by way of  $w$  and  $w'$  is the backproposal way corresponding to  $w$ . A proof that the Metropolis-Hastings ratio in Eq. (16) can be replaced with the ratio in Eq. (17) without changing the stationary distribution of the Markov chain can be found in [15].

One situation where we may find several ways to propose one state from another state is when the states of the Markov chain are described as vectors, and the proposal procedure first chooses a subset of the coordinates before drawing random new values for the selected coordinates. If the procedure allows new values identical to old values to be drawn for one or more selected coordinates, any superset of the changed coordinates could have been the selected coordinates. A bijection between proposal and backproposal ways can be easily constructed, however, by agreement of the set of selected coordinates. The procedure of first selecting a random subset of coordinates and then updating the selected coordinates is called *Partial Importance Sampling* [15], and can be viewed as the discrete version of Green's Reversible Jump MCMC [95].

### A.3 MCMC in Practice

**A case study for MCMC on the TKF92 model** Since the joint distribution of alignments, trees and parameters is a high dimensional distribution that is too complicated for direct, analytical inference, Markov chain Monte Carlo [49, 50] has been used for sampling from the posterior distribution. One of the key questions here is how far we can go with the analytical calculations. For the biologically less reliable, but computationally more tractable TKF91 model [9], Lunter *et al.* developed a fast algorithm [14, 15] that calculates the likelihood of an evolutionary tree and a multiple sequence alignment of observed sequences. A similar fast algorithm in the case of the TKF92 model is unknown, and hence, more data augmentation is necessary when working with the TKF92 model.

This data augmentation includes sequences associated to the internal nodes and pairwise sequence alignments of neighbour nodes associated to the edges of the evolutionary tree. Since the likelihood of substitution events can be efficiently calculated with Felsenstein’s algorithm [20], only the distribution of conditional likelihoods are stored – also known as “Felsenstein’s wildcards” [17] – at internal nodes of the evolutionary tree. We call this structure an *extended alignment*.

In the paper of Miklós *et al.* [30], the Markov chain performs a random walk on the space comprising the following components:

- Edge lengths of the tree
- Model parameters
- Extended alignment, described above
- Tree topology

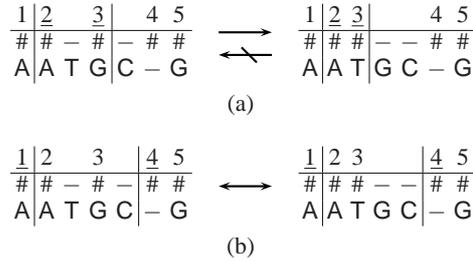
The applied Metropolis-Hastings moves change one of the components randomly, each component selected with a fixed, prescribed probability that was chosen to maximise the mixing of the Markov chain.

Standard techniques were used for modifying edge lengths and parameters in the model, see [15]. Changing the alignment is the most time-consuming event, since the running time of proposing a new alignment is proportional to the product of the lengths of the aligned sequences. A possible solution is modifying only a part of the alignment (“subalignment”), which decreases the running time of this type of proposal. Although it also decreases the mixing of the Markov chain, the overall performance of the Markov chain in terms of total computational time improves [15,45]. The subalignment is specified by a subtree and by the first and last column of the selected alignment region (“window”) of the root node of this subtree. This window is extended to all nodes on the subtree, thus selecting a partial multiple alignment which should be altered. However, since the Markov chain walks on extended alignments, it is a non-trivial question how to propose a random subalignment in a way to maintain the reversibility of the move, which is required by the Metropolis-Hastings algorithm, see Eq. (14). The trick lies in the observation that if the borders of the selected window at the root node are marked with the neighbouring Felsenstein wildcards that *are not* within the window, then regardless of insertions or deletions at the beginning or end of the new alignment, the same window will be available for selection in the new alignment. Hence, the original alignment will be available for (back)proposal from the new alignment. If, on the other hand, the first and last Felsenstein wildcards *within* the window had been chosen to indicate the borders of the window, the proposal might not always be reversible – for an example, see Fig. 9(a).

The distribution of window lengths is set such that the expected running time of an alignment changing step in the Markov chain grows approximately linearly with the lengths of the sequences.

Sequences are iteratively realigned on the selected subtree within the selected window. In each iteration, the new alignment is drawn by the Forward-Backward sampling algorithm [26] with a pair-HMM with ancestral states (“HMM3”), see Fig. 10. We opted not to use the pair-HMM corresponding to the background model, since that would have seven non-silent states, while the model applied has only four states after null-cycle elimination. This reduction of the number of states causes a speed boost of a factor of four to the calculation of proposal probability of the alignment change. The deviation from the TKF92 model did not cause low acceptance ratio for the alignment changing moves.

Nearest neighbour interchanges (NNI) were used for altering the topology as described in [25], which transform a rooted subtree in the way shown on Fig. 11. Since



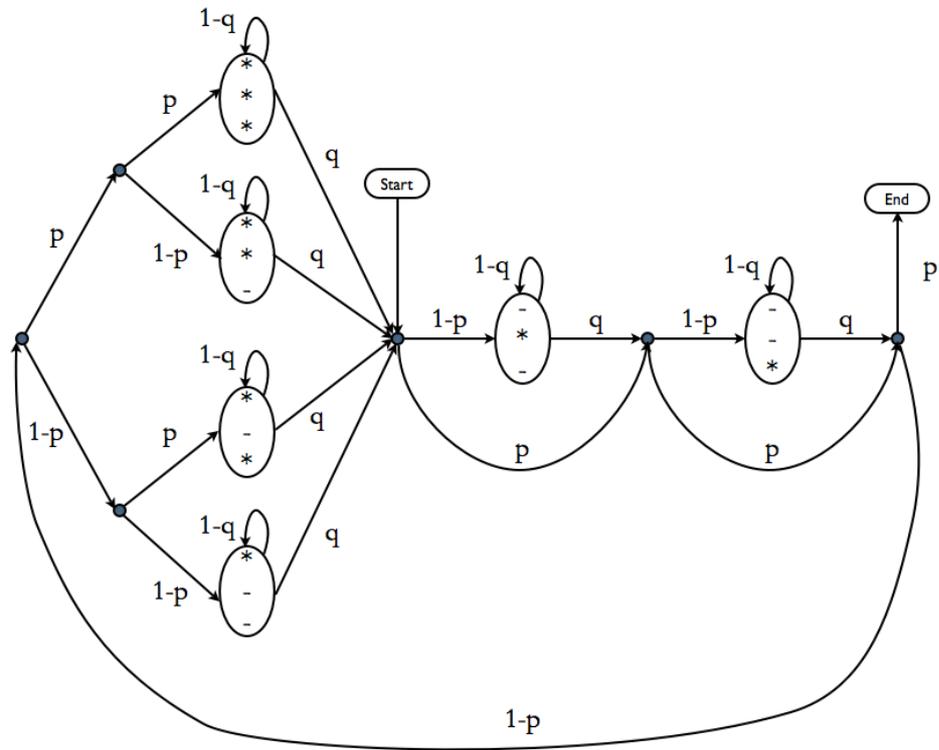
**Fig. 9.** (a) If the window borders are indicated by the first and last ancestral Felsenstein wildcard within the window (indicated as underlined), a proposed alignment could lead to a situation from which the original alignment could not be obtained by the same rules. (b) If the window borders are indicated by neighbouring ancestral Felsenstein wildcards that are not within the window and will not to be realigned, no possible alignment will lead to such a situation, and there will always be a positive probability for backproposal of the original alignment.

the alignment loses its validity after a topology change, the six affected sequences on the quartet are realigned after each nearest neighbour interchange move – the five pairwise alignments are obtained by first aligning A and F to G using the HMM3 shown above, then S and D to P the same way and finally P to G using the pair-HMM.

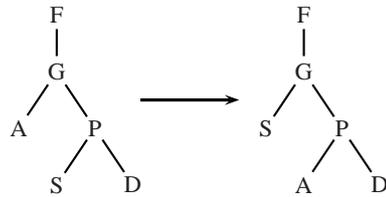
**Fast partial importance sampling for changing tree topologies** When the tree topology is changed in the above-mentioned MCMC, the extended alignment has to be changed, too, since a new tree topology might be inconsistent with the current extended alignment. Drawing a new alignment takes  $O(L^2)$  running time, where  $L$  is the average length of the sequences. Nevertheless, new alignments are proposed using the Forward-Backward sampling algorithm [26], which needs special representation of real numbers to avoid underflow errors, and hence it is a relatively slow algorithm. Instead, a faster method has been implemented by Novák *et al.* [28] based on local realignments, explained below.

Inconsistency happens at insertions and deletions, while homologous positions are indifferent to topology changes. Therefore if one would like to use NNI (Nearest Neighbour Interchange) operations to change the tree topology, one might think that it sufficient to realign those segments of the extended alignment in which insertions-deletions happen. However, new homologous positions might arise during realignment, for which the reversibility rule in Eq. (14) would not hold. Therefore, a non-zero probability for realigning homologous positions is required.

Having discussed this, the proposed algorithm for topology change is working as follows: An internal edge is selected uniformly, and two random subtrees are chosen, one for each end of the selected edge. An NNI operation is performed by swapping these two subtrees. The internal edge defines a quartet (a subtree with four leaves). The characters that are homologous on the quartet are marked, and with a small probability they are selected for realignment. All non-homologous positions are also selected for realignment. The selected positions define a set of windows in the extended alignment that have to be realigned. This set of windows defines the way a new state in the Markov chain is proposed. After realigning the selected windows, the same windows can be selected with non-zero probability in the new alignment, hence the backproposal to the



**Fig. 10.** The pair-HMM that is used to realign sequences of the selected subtree. In all runs,  $p$  was set to 0.99 and  $q$  was set to 0.6. Emission probabilities followed the corresponding substitution model.



**Fig. 11.** Effect of a single Nearest Neighbour Interchange step on a rooted subtree. An NNI move changes the sibling (S) of a daughter with her aunt (A). F, A, S and D may or may not be leaf nodes.

original alignment has non-zero probability as required. The amount of time spent on the realignment is

$$\sum_f l^2(f) \tag{18}$$

where the sum is over the windows  $f$ , and  $l(f)$  is the length of the window. Since the sum of the length of the windows is less than or equal to the length of the extended alignment, and the extended alignment typically gives rise to numerous windows, the running time of this protocol might be significantly less than the time needed to realign the entire sequences. The modified Metropolis-Hastings ratio is calculated as specified in Eq. (17), which also significantly reduces the computational time. Indeed, the transition probability  $T(y|x)$  might be tedious to calculate due to the combinatorial explosion of possible sets of windows that might be selected for realignment, but subsequently not changing the alignment in these windows.

According to experience, this new protocol not only decreases the time needed for one MCMC step, but it also increases the acceptance ratio. A possible explanation for this is that the alignment is changed only with a small probability if no change is necessary. If the entire sequences were realigned, a proposal comprising a preferred topology change and an accidentally unlikely novel alignment might be rejected due to the unlikely alignment part. Since only a part of the alignment is changed, the chance to perturb the alignment in an unfavourable way is decreased. Furthermore, the windows where insertions and deletions happened are changed with probability 1. If the sequences can be aligned better in these windows on the new tree topology, the proposal kernel will propose this new alignment with high probability.