Bayesian Phylogenetic Inference under a Statistical Insertion-Deletion Model

Gerton Lunter[1], István Miklós[1], Alexei Drummond[1], Jens Ledet Jensen[2], and Jotun Hein[1]

[1] Department of Statistics, University of Oxford,
1 South Parks Road, Oxford,
OX1 3TG, United Kingdom
{lunter,miklos,drummond,hein}@stats.ox.ac.uk
[2] Department of Mathematical Sciences, University of Aarhus
Ny Munkegade Building 530,
DK-8000 Aarhus C, Denmark
jlj@imf.au.dk

Abstract. A central problem in computational biology is the inference of phylogeny given a set of DNA or protein sequences. Currently, this problem is tackled stepwise, with phylogenetic reconstruction dependent on an initial multiple sequence alignment step. However these two steps are fundamentally interdependent. Whether the main interest is in sequence alignment or phylogeny, a major goal of computational biology is the co-estimation of both. Here we present a first step towards this goal by developing an extension of the Felsenstein peeling algorithm. Given an alignment, our extension analytically integrates out both substitution and insertion–deletion events within a proper statistical model. This new algorithm provides a solution to two important problems in computational biology. Firstly, indel events become informative for phylogenetic reconstruction, and secondly phylogenetic uncertainty can be included in the estimation of insertion-deletion parameters. We illustrate the practicality of this algorithm within a Bayesian Markov chain Monte Carlo framework by demonstrating it on a non-trivial analysis of a multiple alignment of ten globin protein sequences.

Supplementary material: www.stats.ox.ac.uk/∼miklos/wabi2003/supp.html

1 Introduction

A fundamental problem in computational biology is the inference of phylogeny given a set of DNA or protein sequences. Traditionally, the problem is split into two sub-problems, namely multiple alignment of the sequences, and inference of a phylogeny based on an alignment. Several methods that deal with one or both of these sub-problems have been developed. ClustalW and T-Coffee are popular sequence alignment packages, while MrBayes [13], PAUP* [25] and Phylip [6] all provide phylogenetic reconstruction.

G. Benson and R. Page (Eds.): WABI 2003, LNBI 2812, pp. 228–244, 2003.
© Springer-Verlag Berlin Heidelberg 2003

Although these methods can work very well, they share two fundamental problems. First, the division of the phylogenetic inference problem into multiple sequence alignment and alignment-based phylogenetic reconstruction is flawed. For instance, ClustalW computes its alignment based on a 'guide tree', the choice of which will bias any tree inference that is based on the resulting alignment. The solutions of the two sub-problems are interdependent, and ideally phylogenies and alignments should be co-estimated.

The second issue is that heuristic methods are used to deal with insertions and deletions (indels), and sometimes also substitutions. This lack of a proper statistical framework makes it impossible to accurately assess the reliability of the estimated phylogeny. Much biological knowledge and intuition goes into judging the outcomes of these algorithms.

The relevance of statistical approaches to evolutionary inference has long been recognised. Time-continuous Markov models for substitution processes were introduced more than three decades ago [15], and have been considerably improved since then [27]. The first paper on the evolutionary modelling of indel events appeared in the early nineties [26], giving a statistical approach to pairwise sequence alignment, and its extension to an arbitrary numbers of sequences related by a tree has recently been intensively investigated [23, 9, 12, 10, 19, 18]. Such methods are often computationally demanding, and full maximum likelihood approaches are limited to small trees. Markov chain Monte Carlo techniques can extend these methods to practical-sized problems.

While statistical modelling has only recently been used for multiple sequence alignment, it has a long history in population genetic analysis. In particular, coalescent approaches to genealogical inference have been very successful, both in maximum likelihood [16, 7] and Bayesian MCMC frameworks [28, 1]. The MCMC approach is especially promising, as it allows for large data sets to be tackled, as well as allowing for nontrivial extensions of the basic coalescent model, e.g. [21]. Over the short evolutionary time spans considered in population genetics, sequence alignment is generally straightforward, and genealogical inference from a fixed alignment is well-understood [5, 7, 24, 22]. On the other hand, for more divergent sequences, these approaches have difficulty dealing with indels. Not only is the alignment treated as known, but indel events are generally treated as missing data. Treating gaps as unobserved residues [4] renders them phylogenetically uninformative. However, indel events can be highly informative of the phylogeny, because of their relative rarity compared to substitution events.

In this paper, we present an efficient algorithm for computing the likelihood of a multiple sequence alignment given a tree relating the sequences, under the TKF91 model. This model combines probabilistic evolutionary models for substitution events and indel events, allowing consistent treatment of both in a statistical inference framework. The crux of the method is that all missing data (pertaining to the evolutionary history that generated the alignment) is summed out analytically. The algorithm can be seen as an extension of the celebrated peeling algorithm for substitutions [4] to include single residue indels. Summing out missing data eliminates the need for data augmentation of the

tree, a methodology referred to in the MCMC literature as *Rao-Blackwellization* [17]. As a result, we can treat indels in a statistically consistent manner with no more than a constant cost over existing methods that ignore indels. Moreover we can utilise existing MCMC kernels for phylogenetic inference, changing only the likelihood calculator. We implemented the new likelihood algorithm in the programming language Java and demonstrated its practicality by interfacing with an existing MCMC kernel for phylogenetics and population genetics [1].

The method presented in this paper represents an important step towards the goal of a statistical inference method that co-estimates phylogeny and alignment. In fact, the only component currently missing is a method for sampling multiple sequence alignments under a fixed tree. Several approaches to sampling alignments employing data augmentation have already been investigated [12, 10]. Such methods may well hold the key to a co-estimation approach, and we are currently investigating the various possibilities.

The rest of the paper is organised as follows. In Section 2 we briefly introduce the TKF91 model of single residue insertion and deletion. Section 3 forms the core of the paper. Here we first derive a recursion (the 'one-state recursion') for the tree likelihood that, unlike in the usual Hidden Markov model formulation of the TKF91 model, does not require states for its computation. We then simplify the computation of transfer coefficients to a dynamic programming algorithm on the phylogenetic tree, similar to Felsenstein's peeling algorithm. Finally we prune the recursion so that only nonzero contributions remain, thereby yielding a linear time algorithm. In Section 4, we apply our method to a set of globin sequences, and estimate their phylogeny. Section 5 concludes with a discussion.

2 The TKF Model

The TKF91 model is a continuous time reversible Markov model for the evolution of nucleotide (or amino acid) sequences. It models three of the main processes in sequence evolution, namely *substitutions*, *insertions* and *deletions* of characters, approximating these as single-character processes. A sequence is represented by an alternating string of *characters* and *links*, connecting the characters, and this string both starts and terminates with a link. We adopt the view that insertions originate from links, and add a character-link pair to the *right* of the original link; deletions originate from characters and have the effect of removing the character and its right link. (This view is slightly different but equivalent to the original description, see [26].) In this way, subsequences evolve independently of each other, and the evolution of a sequence is the sum of the evolutions of individual character-link pairs. The leftmost link of the sequence has no corresponding character to its left, hence it is never deleted, and for this reason it is called the *immortal link*.

Since subsequences evolve independently, it is sufficient to describe the evolution of a single character-link pair. In a given time-span τ, this evolves into a sequence of characters of finite length. Since insertions originate from links, the first character of this sequence may be homologous to the original one, while

Fate:	Probability:	Label:
$C \to C\#^{n-1}$	$e^{-\mu\tau}(1-\lambda\beta(\tau))(\lambda\beta(\tau))^{n-1}$	$H_\tau B_\tau^{n-1}$
$C \to \#^n$	$(1-e^{-\mu\tau}-\mu\beta(\tau))(1-\lambda\beta(\tau))(\lambda\beta(\tau))^{n-1}$	$N_\tau B_\tau^{n-1}$
$C \to -$	$\mu\beta(\tau)$	E_τ
$\star \to \star\#^n$	$(1-\lambda\beta(\tau))(\lambda\beta(\tau))^n$	$(1-B_\tau)B_\tau^n$

Table 1. Possible fates after time τ of a single character (denoted C), and of the immortal link (denoted $\star$), and associated probabilities. The first three lines refer to (1) the ancestral character surviving (with 0 or more newly inserted), (2) the ancestral character dying after giving birth to at least one newly inserted one, and (3) the death of the ancestral character and all of its descendants.

subsequent ones will be inserted characters and therefore non-homologous. Table 1 summarises the corresponding probabilities. On the right-hand side of the arrow in the column labelled "Fate", C denotes a character homologous to the original character, whereas #'s denote non-homologous characters. The immortal link is denoted by $\star$ and other links are suppressed. All final arrangements can be thought of as being built from five basic "processes" which we call Birth, Extinction, Homologous, New (or Non-homologous) and Initial (or Immortal). These processes are labelled by their initials, and each corresponds to a specific probability factor as follows:

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

where parameters λ and μ are the birth rate per link and the death rate per character, respectively, and in order to have a finite equilibrium sequence length, we require $\lambda < \mu$. We followed [26] in using the abbreviation

$$\beta(\tau) := \frac{1-e^{(\lambda-\mu)\tau}}{\mu-\lambda e^{(\lambda-\mu)\tau}}. \quad (2)$$

In a tree, time flows forward from the root to the leaves, and to each node of the tree we associate a time parameter τ which is set equal to the length of the incoming branch. For the root, $\tau = \infty$ by assumption of stationarity at the root, and the resulting equilibrium length distribution of the immortal link sequence is geometric with parameter $B_\infty = \lambda/\mu$ (where length 0 is possible); other links will have left no descendants since $H_\infty = N_\infty = 0$.

Because the TKF91 model is time reversible, the root placement does not influence the likelihood (Felsenstein's "Pulley Principle", [4]). Although the algorithms in this paper do not *look* invariant under root placement at all, in fact they are. This follows from the proofs, and we have used it to check the correctness of our implementations.

In the original TKF91 model, a simple substitution process known as the Felsenstein-81 model [4] was used. It is straightforward to replace this by more

general models for substitutions of nucleotides or amino acids [11]. In the present paper, when a new non-homologous character appears at a node (as the result of a B or N process), it is always drawn from the equilibrium distribution; if a character at a node is homologous to the character at its immediate ancestral node, then the probability of this event is given by the chosen substitution model.

3 Computing the Likelihood of a Homology Structure

In the previous Monte Carlo statistical alignment papers, the sampled missing data were either unobserved sequences at internal nodes [14], or both internal sequences and alignments between nodes [12]. In both cases the underlying tree was fixed. Here we introduce the concept of *homology structure*, essentially an alignment of sequences at leaves, without reference to the internal tree structure. We present a new algorithm that allows us to compute, under the TKF91 model, the likelihood of observing a set of sequences and their homology structure, given a phylogeny and evolutionary parameters. Missing data in this case are all substitution events and indel events compatible with the observed data, all of which are analytically summed out in linear time. In contrast to previous MCMC approaches [12, 14], we need not store missing data at internal tree nodes, and we can change the tree topology without having to resample missing data. This enables us to consider the *tree* as a parameter, and efficiently sample from tree space.

3.1 Definitions and Statement of the One-State Recursion

Let $A_1, A_2, ...A_m$ be sequences, related to a tree T with vertex set V. Let a_i^j denote the jth character of sequence A_i, and let A_i^k denote its k long prefix.

A *homology structure* $\mathcal{H}$ on $A_1, \ldots, A_m$ is an equivalence relation $\sim$ on the set of all the characters of the sequences, $C = \{a_i^j\}$. It specifies which characters are homologous to which. The evolutionary indel process generating the homology structure on the sequences imposes constraints on the equivalence relations that may occur. More precisely, the equivalence relation $\sim$ has the property that a total ordering, $<_h$, exists on C such that

$$\begin{aligned} a_i^x =_h a_j^y &\iff a_i^x \sim a_j^y, \\ a_i^x <_h a_i^y &\iff x < y \end{aligned} \tag{3}$$

In particular, these imply that characters of a single sequence are nonhomologous. The ordering $<_h$ corresponds to the ordering of columns of homologous characters in an alignment. Note that for a given homology structure, this ordering may not be unique, see Figure 1. This many-to-one relationship of alignment to homology structure is the reason for introducing the concept of homology structure, instead of using the more common concept of alignment.

Below we give an algorithm for calculating the likelihood of the observed data, namely, the sequences with their homology structure. By definition, this

Fig. 1. (Left:) Two alignments representing the same homology structure: residues are homologous if they appear in the same column. These alignments may represent different evolutionary histories, all of which we include in our recursion. Note that this ambiguity is rare in biological sequence alignments. (Right:) Due to the evolutionary process acting on the sequences, homology relationships (arrows) will never 'cross' as depicted. This restriction on the equivalence relation $\sim$ is codified by $<_h$ (see text).

likelihood is the sum of the likelihoods of all evolutionary scenarios resulting in the observed data. In previous works ([12, 14]), it was shown that an evolutionary scenario can be described as a path in a multiple-HMM, so that the likelihood of a homology structure can be calculated using a multiple-HMM. However, this straightforward calculation is infeasible for practical-sized biological problems, since the number of states in the HMM grows exponentially with the number of sequences [18].

In this subsection we show that a so-called *one-state* recursion exists for calculating this likelihood, given evolutionary parameters. In subsequent sections we give an efficient algorithm based on this recursion.

Definitions Let t_r denote the subtree whose root is $r \in V$. Let Ω be the (nucleotide or amino acid) alphabet. An *event* e is a labelling of the nodes of a subtree t_r with $\mathtt{B}_\alpha$, $\mathtt{H}_\alpha$, $\mathtt{N}_\alpha$, $\mathtt{E}$, where $\alpha \in \Omega$ is a character, subject to the following conditions: There is a birth of a character α ($\mathtt{B}_\alpha$) at the root $r = r_e$ of the event, and only there, and if a node is labelled $\mathtt{E}$ then all its descendants are labelled $\mathtt{E}$ as well [18], see Figure 2. In this way, an event codifies the fate (see Table 1) of a single nucleotide born at $r \in T$, but ignoring all subsequent births it may have spawned. We let $e(n)$ denote the symbol labelling the node n. We say that an event e *emits a character* α at a leaf n if $e(n) = \mathtt{H}_\alpha$, $\mathtt{N}_\alpha$ or $\mathtt{B}_\alpha$. The *emission vector* v_e is an m-dimensional vector, and its ith coordinate is 1 if e emits a character into the ith leaf, and 0 otherwise. The *probability factor* $p(e)$ of an event e is the product of probabilities associated to the label of each node (see Table 1), including nucleotide equilibrium probabilities (for $\mathtt{B}_\alpha, \mathtt{N}_\alpha$) or substitution probabilities (for $\mathtt{H}_\alpha$), except that an $\mathtt{E}$ label counts for 1 if its parent is also labelled $\mathtt{E}$. We call these probability *factors* as they do not add up to 1 in an obvious way. However, if we calculate probabilities of observing sequences at leaves using these probability factors, the obtained probabilities do add to 1 (summation is over all observable sequences).

We end this section with a description of which events are allowed given a homology structure:

Definition 1 (Agreement with homology). *An event e* agrees with the homology structure *if the following holds:*

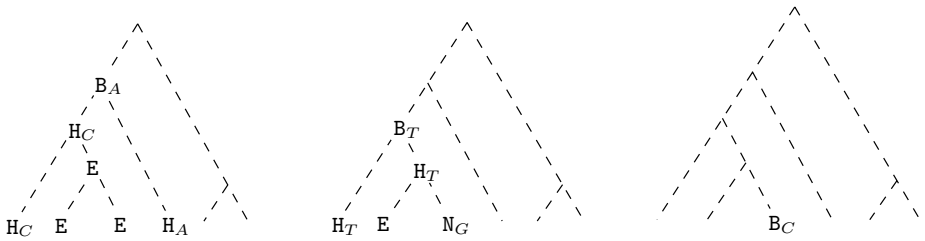

Fig. 2. Three possible events. The first two events emit two characters; in e they are homologous (and at least one mutation event occurred, changing an A into a C), while in e' they are inhomologous by the the N ("new") label; see also Section 2.

1. *If e emits a character $a \in C$, then it emits all $a' \in C$ with $a \sim a'$, and the nodes along the path connecting a and a' are not labelled* $\mathtt{N}_\alpha$.
2. *If e emits characters a and a', and $a \not\sim a'$, then at least one of the nodes along the path connecting them is labelled* $\mathtt{N}_\alpha$.

Statement of the One-State Recursion Let $P(\mathbf{K})$ denote the probability of emitting the prefixes $A^{\mathbf{K}} := (A_1^{K_1}, ..., A_m^{K_m})$ and such that their homology agrees with the given homology structure $\mathcal{H}$. The following equation holds:

$$P(\mathbf{K}) = \left(\prod_{n \in T} (1 - B_n) \right) \sum_{(e_1,\ldots,e_n) \in \mathcal{A}} p(e_1) \cdots p(e_n), \tag{4}$$

where $\mathcal{A}$ is the set of sequences of legal events (see App. A for a definition) that emit $A^{\mathbf{K}}$ and agree with the homology structure, and the factor in front derives from the immortal link. Note that for brevity we wrote B_n instead of $B_{l(n)}$, where $l(n)$ is the length of the incoming branch. Our goal is to find a recursion in terms of $P(\mathbf{K})$. To formulate our result, we need one more definition:

Definition 2. *A set of events $\{e_1, \ldots, e_l\}$ is a* nested set *if, for $i \neq j$, we have that $r_{e_i} \in t_{e_j} \implies e_j(r_{e_i}) = \mathtt{E}$.*

Let $M_l^{\mathbf{K}}$ denote the set of nested sets of l events, where each event is in $\mathcal{E}(\mathbf{K})$. Here $\mathcal{E}(\mathbf{K})$ is the set of events e that emit characters that extend the prefixes to $A^{\mathbf{K}}$ (i.e. from $A^{\mathbf{K}-v_e}$), and that agree with the homology structure $\mathcal{H}$.

Theorem 1 (One-state recursion). *The following equation holds for $P(\mathbf{K})$:*

$$P(\mathbf{K}) = \sum_{l=1}^{2m} \sum_{\{e_1,e_2,\ldots e_l\} \in M_l^{\mathbf{K}}} (-1)^{l-1} P\left(\mathbf{K} - \textstyle\sum_{j=1}^{l} v_{e_j}\right) \prod_{i=1}^{l} p(e_i) \tag{5}$$

Proof: See appendix A ■

3.2 A Reverse Traversal Algorithm for the Transition Factor

Theorem 1 by itself does not give a fast algorithm for calculating the likelihood of a homology structure, as the number of terms in the summation (5) grows exponentially with the number of leaves m. As a first step to arrive at an efficient algorithm, we group terms in (5) as follows:

$$P(\mathbf{K}) = \sum_{v \in \{0,1\}^m} T_v^{\mathbf{K}} P(\mathbf{K} - v) \tag{6}$$

Here v runs over all 2^m length-m vectors with entries 0 or 1 (in section 3.3 we reduce the v-summation). The *transition factors* $T_v^{\mathbf{K}}$ are sums of expressions of the form $(-1)^{l-1} \prod_{i=1}^{l} p(e_i)$. Previously we showed how to calculate these transition factors in linear time when there is no homology structure to be observed [18]. In this section we describe a similar algorithm for the present case.

Note that the zero vector is among the vectors summed over in (6), so that $P(\mathbf{K})$ also appears on the right-hand side. Solving the resulting linear equation results in a proper recursion, and conceptually amounts to summing out all non-emitting events. The relation between this approach and existing ones, see e.g. [3, 23], is discussed in more detail in [18].

To derive the dynamic programming recursion, the main observation is that a nested set of events $\{e_1, \ldots, e_l\}$ can be represented by a labeling of a single tree, since for every node n there is at most one e_i with $e_i(n) \neq \mathtt{E}$. By labeling each node n with the unique non-E label among the $e_i(n)$ (or with $\mathtt{E}$ if none exists), the roots of the events can be recovered as they are precisely the nodes labeled B_α. The term $(-1)^{l-1} p(e_1) \cdots p(e_l)$ that corresponds to a nested set represented in this way, is calculated by multiplying the probabilities for the labels at all nodes according to (1) with the following two caveats: (a) a node labeled $\mathtt{E}$ carries a factor 1 if its parent is also labeled $\mathtt{E}$; (b) a B_α attracts a minus sign to account for the factor $(-1)^{l-1}$, and an additional factor E if its parent is not labeled $\mathtt{E}$. Finally, summing over all possible nested sets is done in linear time with a dynamic programming or 'pruning' algorithm similar to Felsenstein's one [4].

To take the homology structure into account, we need to restrict the tree labelings to those that produce emissions compatible with the given homology, amounting to implementing Definition 1. Input to the algorithm is the total emission vector v, and a numerical vector h that encodes the homology structure of v such that $h_i = h_j \neq 0 \iff a_i^{\mathbf{K}_i} \sim a_j^{\mathbf{K}_j}$, and h_i is zero whenever $v_i = 0$. For example, for $\{e_1, \ldots, e_l\} \in M_l^{\mathbf{K}}$ we have $v = \sum_i v_{e_i}$ and we could set $h = \sum_i i v_{e_i}$. Those leaves with $h_i = h_j$ are said to belong to the same *homology class*.

The algorithm proceeds as follows. First we compute for each homology class the minimum spanning tree of its leaves. If two such spanning trees intersect, no nested sets corresponding to v and satisfying the homology contraints exist, and $T_v^{\mathbf{K}} = 0$. Otherwise, for each node n we set $h(n)$ to be the label of the homology class whose tree contains n, and 0 otherwise:

Algorithm 1 (Computing homology spanning trees)
Input: *Homology vector* $h = (h_1, \ldots, h_m)$*, tree* T*.*
Output: *Either* $''T_v^{\mathbf{K}} = 0''$*, or homology class* $h(n)$ *for each node* n *in* T*.*
Algorithm:

$h(n) \leftarrow h_i$ *for the index* i *corresponding to leaf* n*;* 0 *for non-leaf nodes.*
$c(n) \leftarrow 1$ *if* $h(n) \neq 0$*;* 0 *otherwise.*
$m(j) \leftarrow \#\{i|h_i = j\}$ *(multiplicity of homology class* j*)*
`For` *all nodes* n *in postorder traversal (i.e. leaves first):*
  `If` $c(n) \neq m(h(n))$*, then:*
    `If` $h(a(n)) = 0$ *or* $h(a(n)) = h(n)$*, then:*
      $h(a(n)) \leftarrow h(n)$
      $c(a(n)) \leftarrow c(a(n)) + c(n)$
    `Else:`
      `Return` $''T_v^{\mathbf{K}} = 0''$
    `EndIf`
  `EndIf`
`EndFor`
`Return` $h(\cdot)$

In this algorithm, $a(n)$ denotes the ancestor of node n. The symbol $\leftarrow$ ('becomes') is the assignment operator.

Our recursion is in terms of quantities $F_H(\alpha, n)$, $F_N(\alpha, n)$ and $F_E(n)$, which are related to (1) character α at node n being homologous to at least one character at the leaves of t_n in the case of F_H; (2) that character being non-homologous to all characters at the leaves of t_n in the case of F_N; (3) no character existing at node n in the case of F_E.

We introduce some notation. For a node n, n_l and n_r denote the left and right descendant, respectively. We abbreviate $\delta_{n,m} := \delta_{h(n),h(m)}$, where the second δ is the usual Kronecker delta, i.e. $\delta_{n,m} = 1$ if $h(n) = h(m)$, 0 otherwise. Let $p_n(\alpha \to \gamma)$ denote the probability that character α evolves into γ in time l_n, which is the length of the incoming branch to node n, and let $\pi(\alpha)$ denote the equilibrium distribution of characters. We finally introduce the following abbreviations, where we again write H_n, N_n, B_n, E_n for $H_{l(n)}, N_{l(n)}, B_{l(n)}, E_{l(n)}$ resp., where $l(n)$ is the length of node n's incoming branch:

$$H(n,\alpha) = \sum_{\gamma\in\Omega} F_H(\gamma, n) H_n p_n(\alpha \to \gamma), \tag{7}$$

$$N(n,\alpha) = F_E(n)E_n + \sum_{\gamma\in\Omega} F_N(\gamma, n) H_n p_n(\alpha \to \gamma) + \sum_{\gamma\in\Omega} \left[F_H(\gamma, n) + F_N(\gamma, n)\right] \left[N_n - E_n B_n\right] \pi(\gamma), \tag{8}$$

$$E(n) = F_E(n) - \sum_{\gamma\in\Omega} \left[F_H(\gamma, n) + F_N(\gamma, n)\right] B_n \pi(\gamma). \tag{9}$$

Algorithm 2 (Computing $T_v^{\mathbf{K}}$) *With the notation as above, and $p(n)$ computed by Algorithm 1, the transition factor in (6) associated to the sequences $A_1, \ldots, A_m$ related by a tree T and homology structure $\mathcal{H}$, is $T_v^{\mathbf{K}} = -E(r)$, where r is the root of T. The terms $F_E(r)$, $F_H(\gamma, r)$ and $F_N(\gamma, r)$ are computed recursively as follows. If $h(n) = 0$ and n is an internal node, then*

$$F_H(\alpha, n) = H(n_l, \alpha)N(n_r, \alpha) + N(n_l, \alpha)H(n_r, \alpha), \tag{10}$$
$$F_N(\alpha, n) = N(n_l, \alpha)N(n_r, \alpha), \tag{11}$$
$$F_E(n) = E(n_l)E(n_r). \tag{12}$$

If $h(n) \neq 0$ and n is an internal node, then

$$\begin{aligned} F_H(\alpha, n) = {} & [H(n_l, \alpha)\delta_{n,n_l} + N(n_l, \alpha)(1 - \delta_{n,n_l})] \times \\ & [H(n_r, \alpha)\delta_{n,n_r} + N(n_r, \alpha)(1 - \delta_{n,n_r})], \end{aligned} \tag{13}$$
$$F_N(\alpha, n) = F_E(n) = 0. \tag{14}$$

If n is a leaf node, then

$$F_H(\alpha, n) = 1 \quad \text{if } a_n^{K_n} = \alpha, \quad 0 \text{ otherwise}; \tag{15}$$
$$F_N(\alpha, n) = 0; \tag{16}$$
$$F_E(n) = 1 \quad \text{if } v_n = 0, \quad 0 \text{ otherwise}. \tag{17}$$

Note that we abused notation by confusing (leaf) nodes and sequence indices.

3.3 Finding the Prefix Vectors K

For many vectors $\mathbf{K}$, the quantity $P(\mathbf{K})$ will vanish, since the corresponding sequence prefixes $A^{\mathbf{K}}$ cannot occur while at the same time agreeing with the homology structure. Secondly, for many vectors v the transition factor $T_v^{\mathbf{K}}$ will similarly vanish, for the same reason. Restricting to those v and $\mathbf{K}$ that actually contribute dramatically increases the efficiency of the algorithm.

All this depends on the homology structure; the various paths through $\mathbf{K}$-space that the algorithm traverses correspond to the various possible orderings $\leq_h$ that correspond to the underlying homology structure. In fact, if none of the characters are homologous to any other (corresponding to an alignment with only single-character columns), all of the $\mathbf{K}$-values are valid, and all of the vectors v have to be considered. In practice, however, such alignments will have very low likelihood and can safely be ignored.

As the final part of our algorithm, we give the top-level subroutine that traverses the set of $\mathbf{K}$-vectors for which $P(\mathbf{K}) \neq 0$. Input to the algorithm is a homology structure, which is represented by an alignment, or more precisely, as a sequence of vectors $(c_1, c_2, \ldots, c_n)$, where the jth coordinate of c_i is 1 precisely if the jth character in the ith column in the alignment is a residue, and 0 if it is a gap. The algorithm is independent of the alignment chosen to represent the homology structure.

Algorithm 3 (Traversing contributing prefix vectors)
Input: *Vectors* $(c_1, \ldots, c_n)$*; sequences* $A_1, \ldots, A_m$*; tree* T*; cutoff* N
Output: *Likelihood of sequences under given homology structure.*
Algorithm:

Mark all c_i *'free';* $\mathbf{K} \leftarrow (0, \ldots, 0)$*;* $P(\mathbf{K}) \leftarrow \prod_{n \in T}(1 - B_n)$
`While` *not all* c_i *marked 'used':*
  Let s_i *be the maximal, increasing subsequence satisfying:* c_{s_i} *'free'* $\forall i$
  Mark c_{s_k} *'possible' iff* $c_{s_k} \cdot \sum_{i=1}^{k-1} c_{s_i} = 0$
  `If` *there are no more than* N *'possible' vectors:*
    `For` *all subsets* $C = \{c'_1, \ldots, c'_n\}$ *of 'possible' vectors:*
      Construct $v = \sum_{k=1}^{n} c'_k$ *and* $h = \sum_{k=1}^{n} k c'_k$*; compute* $T_v^{\mathbf{K}+v}$
      $P(\mathbf{K} + v) \leftarrow P(\mathbf{K} + v) + T_v^{\mathbf{K}+v}(1 - T_{\mathbf{0}}^{\mathbf{0}})^{-1} P(\mathbf{K})$
    `EndFor`
  `Else:`
    `Return` *"Likelihood too small"*
  `EndIf`
  $k \leftarrow \max\{i | c_i$ *labeled 'possible'*$\} \cup \{\infty\}$
  Mark c_k *'used';* $\mathbf{K} \leftarrow \mathbf{K} + c_k$
  `For` *all* $i > k$ *for which* c_i *is labeled 'used':*
    Mark c_i *'free';* $\mathbf{K} \leftarrow \mathbf{K} - c_i$
  `EndFor`
`EndWhile`
`Return` *"Likelihood=*$P(\mathbf{K})$*"*

In practice, for biologically meaningful alignments and with the cutoff N in place, the algorithm is linear, although it has slower worst-case behaviour.

4 Results

The indel peeling algorithm of Section 3.3 provides a method for calculating the likelihood $L = \Pr\{A, \mathcal{H} | T, Q, \lambda, \mu\}$ of observing the sequences with their homology structure ('alignment') given the tree and model parameters. Here A are the amino acid sequences, $\mathcal{H}$ is their homology structure, T is the tree including branch lengths, Q is the substitution rate matrix, and λ, μ are the amino acid birth and death rates. To demonstrate the practicality of the new algorithm for likelihood calculation we undertook a Bayesian MCMC analysis of ten globin protein sequences. We chose to use the standard Dayhoff rate matrix to describe the substitution of amino acids. For the purpose of this example we generated a homology structure using T-Coffee. Given this homology structure, we co-estimated the parameters of the TKF91 model, and the tree topology and branch lengths. To do this we sampled from the posterior,

$$h(\mu, T) = \frac{1}{Z} \Pr\{A, \mathcal{H} | T, Q, \lambda, \mu\} f(T, \lambda, \mu). \tag{18}$$

Here Z is the unknown normalising constant. We chose the prior distribution on our parameters, $f(T, \lambda, \mu)$, so that T was constrained to a molecular clock, and

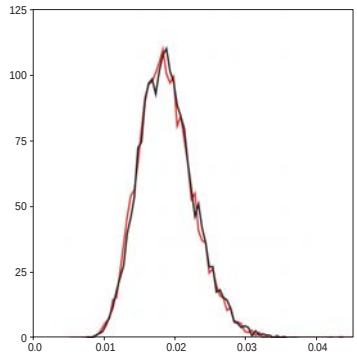

Fig. 3. Estimated posterior densities of the death rate μ sampled according to h (see text), for two independent runs, showing good convergence. Sampled mean is 0.0187; the 95% highest posterior density (HPD) interval was estimated to be $(0.0114, 0.0265)$.

$\lambda = \mu L/(L+1)$ to make the expected sequence length under the TKF91 model agree with the observed lengths. Here L is the geometric average length of the globin sequences, We assume a molecular clock to gain insight into the relative divergence times of the alpha-, beta- and myoglobin families. In doing so we incorporate insertion-deletion events as informative events in the evolutionary analysis of the globin family.

The posterior density h is a complicated function defined on a space of high dimension. We summarise the information it contains by computing the expectations, over h, of various statistics of interest. We estimate these expectations by using MCMC to sample from h. Figure 3 depicts the marginal posterior density of the μ parameter for two independent MCMC runs, showing convergence. Figure 4 depicts the maximum *a posteriori* (MAP) estimate of the phylogenetic relationships of the sequences. This example exhibits only limited uncertainty in the tree topology, however we observed an increased uncertainty for trees that included divergent sequences, such as bacterial and insect globins (results not shown).

The estimated time of the most recent common ancestor of each of the alpha, beta and myoglobin families are all mutually compatible (result not shown), suggesting that the molecular clock hypothesis is at least approximately valid. Analysis of a four sequence dataset demonstrate consistency in μ estimates between MCMC and previous ML analyses [18] (data not shown). Interestingly, the current larger dataset supports a lower value of μ. This is probably due to the fact that no indels are apparent within any of the subfamilies despite a considerable sequence divergence.

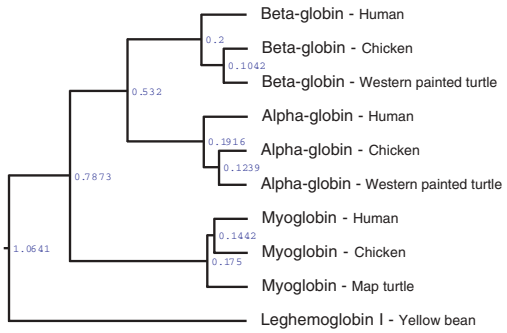

Fig. 4. The maximum *a posteriori* (MAP) estimate of the globin tree. The node heights are given in expected substitutions per site. Notice that the alpha and beta chain sub-families both support the traditional ordering of birds, turtles and mammals, while the three myoglobin sequences support an unconventional phylogeny. This inconsistent signal from myoglobin has been previously observed [8]. The marginal posterior probability (estimated from the MCMC chain) for the monophyly of human and chicken myoglobin was 83.1%, followed by the conventional grouping of turtle and chicken at 11.9%. The third topological arrangement of myoglobin occurred the remaining 5% of the time, suggesting significant homoplasy in this sub-family.

5 Discussion

In this paper we presented a method that extends Felsenstein's peeling algorithm to incorporate insertion and deletion events, under the TKF91 model. This renders indel events informative for phylogenetic inference. Although this incurs considerable algorithmic complications, the resulting algorithm is still linear-time for biological alignments (see also Figure 1).

It should be stressed that the two MCMC analyses of the globin data set were purely illustrative of the practicality of the algorithm described, and no novel biological results were obtained. The two MCMC runs undertaken required only about 3 hours of CPU time each on a 1.25 GHz G4 Apple Macintosh, using an unoptimised implementation of the algorithm, and the estimated number of independent samples (estimated sample size, ESS) obtained for the posterior probability were good at 650 and 820 respectively (see [1] for methods). The estimated ESSs for the death rate, μ, were 4800 and 3900 respectively. We expect analyses of data sets of around 50 sequences to be readily attainable with only a few days computation.

Our method is intended as the first step towards a full co-sampling approach of phylogeny and alignment. The only remaining issue is to combine the current method with a sampling strategy for alignments. Proper sampling algorithms have already been developed [12, 10], varying mainly in the way data augmentation is employed. These or similar methods of data augmentation may prove

helpful in certain stages of an MCMC kernel that solves the full problem of phylogeny/alignment co-estimation. However, in the context of the algorithm described herein it may be possible to avoid data augmentation entirely and still achieve efficient co-estimation of phylogeny and alignment through a simple Metropolis-Hastings proposal scheme that perturbs homology structures directly.

As was mentioned in [26], it would be desirable to have a statistical sequence evolution model that deals with 'long' insertions and deletions, which is the statistical counterpart of affine gap penalties in score-based alignment methods. We have made progress on a full likelihood method for statistical sequence alignment under such an evolutionary model [20], but this method seems not to be directly extendable to trees. We believe that here too, Markov chain Monte Carlo approaches, combined with data augmentation, will be the key to practical algorithms.

Acknowledgements

This research is supported by EPSRC (code HAMJW) and MRC (code HAMKA).

A Proof of Theorem 1

Let $\leq_p$ be the partial ordering on V for which $n_1 \leq_p n_2$ iff $n_1 \in t_{n_2}$. We denote $n_1 <>_p n_2$ if n_1 and n_2 are incomparable, and we denote $n_1 \not\leq_p n_2$ if either $n_1 >_p n_2$ or $n_1 <>_p n_2$. Also we introduce a total ordering $\leq$ of the nodes which is an arbitrary refinement of the partial ordering, namely, $n_1 \leq n_2$ implies that $n_1 \leq_p n_2$ or $n_1 <>_p n_2$.

A *state* S is a function $V \to \{0,1\}$, with the property that the root of the tree is labelled with 1, and $S(n_1) = 0 \implies S(n_2) = 0$ for all $n_2 <_p n_1$. The state is used to keep track of where subsequent births may occur (i.e. everywhere except after extinctions `E`), and to avoid over-counting of independent histories in disjoint subtrees. The *initial state* is the state labelling all nodes with 1. The *action* of an event e on a state S' is defined to be $S = S' * e$, with

$$S(n) = \begin{cases} S'(n) & \text{if } n > r_e \\ 0 & \text{if } n < r_e \text{ and } n <>_p r_e \\ 1 & \text{if } n \leq_p r_e \text{ and } e(n) \neq \texttt{E} \\ 0 & \text{if } n \leq_p r_e \text{ and } e(n) = \texttt{E} \end{cases} \tag{19}$$

The first two lines make sure that histories in disjoint subtrees are not counted twice, by disallowing new births in disjoint subtrees at one 'side' of the current event. The last two lines implement the restriction that new births are not allowed after extinction (`E`) events. If an event e occurs in state S, the state becomes $S*e$ after the event. An event e is a *legal event* in a state S iff $S(r_e) = 1$.

Let $P_S(\mathbf{K})$ denote the probability of emitting the prefixes $A^{\mathbf{K}}$ by legal events agreeing with the homology structure, starting from the initial state, such that the state after the last event is S. A recursion in terms of $P_S(\mathbf{K})$ is easy:

$$P_S(\mathbf{K}) = \sum_{S'} \sum_{\substack{e \in \mathcal{E}(\mathbf{K}): \\ S' * e = S \ \wedge \ S'(e)=1}} P_{S'}(\mathbf{K} - v_e) p(e) \tag{20}$$

Here $\mathcal{E}(\mathbf{K})$ is the set of events e that emit characters extending the prefixes from $A^{\mathbf{K}-v_e}$ to $A^{\mathbf{K}}$, and that agree with the homology structure, and as initial condition we have $P_I(\mathbf{0}) = \prod_{n \in T}(1 - B_n)$, with I the initial state (assigning 1 to each node). The correctness of the recursion is discussed in detail in [18]. In the end, the quantity of interest $P(\mathbf{K})$ is obtained by summing over all states:

$$P(\mathbf{K}) = \sum_{S} P_S(\mathbf{K}) \tag{21}$$

This approach to calculate the likelihood of a set of sequences is very similar to the forward-backward algorithm of HMMs [2], and it is the straightforward extension of the original formulation of the dynamic programming algorithm given in [26]. We now combine (20) and (21) to get:

$$P(\mathbf{K}) = \sum_{\{e_1\} \in M_1^{\mathbf{K}}} \sum_{S:S(r_{e_1})=1} P_S(\mathbf{K} - v_{e_1}) p(e_1) \tag{22}$$

We can rewrite this as

$$P(\mathbf{K}) = \sum_{\{e_1\} \in M_1^{\mathbf{K}}} P(\mathbf{K} - v_{e_1}) p(e_1) - \sum_{\{e_1\} \in M_1^{\mathbf{K}}} \sum_{S:S(r_{e_1})=0} P_S(\mathbf{K} - v_{e_1}) p(e_1) \tag{23}$$

The theorem can be derived from (23) using the following lemma recursively:

$$\begin{aligned} &\sum_{\{e_1,e_2,\ldots e_l\} \in M_l^{\mathbf{K}}} \sum_{S:S(r_{e_i})=0 \forall i} P_S\left(\mathbf{K} - \textstyle\sum_{i=1}^{l} v_{e_i}\right) \prod_{i=1}^{l} p(e_i) = \\ &\quad = \sum_{\{e_1,e_2,\ldots e_{l+1}\} \in M_{l+1}^{\mathbf{K}}} P\left(\mathbf{K} - \textstyle\sum_{i=1}^{l+1} v_{e_i}\right) \prod_{i=1}^{l+1} p(e_i) - \\ &\quad - \sum_{\{e_1,e_2,\ldots,e_{l+1}\} \in M_{l+1}^{\mathbf{K}}} \sum_{S:S(r_{e_i})=0 \forall i} P_S\left(\mathbf{K} - \textstyle\sum_{i=1}^{l+1} v_{e_i}\right) \prod_{i=1}^{l+1} p(e_i) \end{aligned} \tag{24}$$

Before we prove (24), we show that it indeed leads to the proof of Theorem 1. When we apply the recursion (24) to (23) $k-1$ times, we end up with the equation

$$\begin{aligned} P(\mathbf{K}) = &\sum_{l=1}^{k} \sum_{\{e_1,e_2,\ldots,e_l\} \in M_l^{\mathbf{K}}} (-1)^{l-1} P\left(\mathbf{K} - \textstyle\sum_{i=1}^{l} v_{e_i}\right) \prod_{i=1}^{l} p(e_i) \\ &+ (-1)^k \sum_{\{e_1,e_2,\ldots,e_k\} \in M_k^{\mathbf{K}}} \sum_{S:S(r_{e_i})=0 \forall i} P_S\left(\mathbf{K} - \textstyle\sum_{i=1}^{k} v_{e_i}\right) \prod_{i=1}^{k} p(e_i) \end{aligned} \tag{25}$$

However, $M_k^{\mathbf{K}}$ eventually becomes empty, since no two events in a nested set share a root, and there are $2m-1$ nodes in the tree (recall that m is the number of leaves). This implies the theorem.

To prove the lemma, we first apply (20) to the left-hand side of (24) to get

$$\sum_{\substack{\{e_1,e_2,\ldots e_l\}\\ \in M_l^{\mathbf{K}}}} \sum_{\substack{S:\\ S(r_{e_i})=0 \forall i}} \sum_{S'} \sum_{\substack{e\in\mathcal{E}(\mathbf{K}-\sum_{i=1}^{l} v_{e_i}):\\ S'*e=S \ \wedge \ S'(r_e)=1}} P_{S'}\left(\mathbf{K} - v_e - \sum_{i=1}^{l} v_{e_i}\right) p(e) \prod_{i=1}^{l} p(e_i) \quad (26)$$

The main observation is that for the events $\{e_1,\ldots,e_l\}$ and e over which the summation extends, we have that $\{e_1,\ldots,e_l,e\} \in M_{l+1}^{\mathbf{K}}$, and $r_e \not\leq_p r_{e_i}$ for all i. To show the latter, suppose that $r_e \leq_p r_{e_i}$ for a particular i, then since $S(r_{e_i}) = 0$ we have $S(r_e) = 0$, contradicting $S = S' * e$, therefore r_e must be a greatest element in the partial ordering. To show the former, note that from the action of e on S' it follows that $e(r_{e_i})$ must be $\mathtt{E}$ for all $r_{e_i} <_p r_e$, which implies that $\{e_1,\ldots,e_l,e\}$ is a nested set. The events in a nested set have non-overlapping emissions, so that $e \in \mathcal{E}(\mathbf{K} - \sum_{i=1}^{l} v_{e_i})$ implies that $e \in \mathcal{E}(\mathbf{K})$. From this it follows that $\{e_1,\ldots,e_l,e\} \in M_{l+1}^{\mathbf{K}}$.

Since the summand of (26) only involves S' and the events, the above observation implies that we can simplify (26) to a sum of the expression

$$P_{S'}\left(\mathbf{K} - \textstyle\sum_{i=1}^{l+1} v_{e_i}\right) \displaystyle\prod_{i=1}^{l+1} p(e_i) \quad (27)$$

over some $\{e_1,\ldots,e_{l+1}\} \in M_{l+1}^{\mathbf{K}}$, and some states S'. We claim the sum actually extends over all nested sets in $M_{l+1}^{\mathbf{K}}$, and all states S', except those for which $S'(r_{e_i}) = 0$ for all i. That these should be excluded is clear as e in (26) satisfies $S'(e) = 1$. Conversely, let S' be given and suppose $S'(e_i) = 1$ for at least one i, then for e choose the event whose root is maximal in the total ordering among the r_{e_i} for which $S'(r_{e_i}) = 1$. This event is legal for S', and yields a state $S = S' * e$ for which $S(r_{e_i}) = 0$ for all remaining e_i; moreover it is the only event among the $\{e_1,\ldots,e_{l+1}\}$ that has these two properties. This finishes the proof of the lemma, and of the theorem. ■

References

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

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

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

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

[5] J. Felsenstein. Estimating effective population size from samples of sequences: Inefficiency of pairwise and segregating sites as compared to phylogenetic estimates. *Genetical Research Cambridge*, 59:139–147, 1992.

[6] J. Felsenstein. PHYLIP version 3.5c. Dept. of Genetics, Univ. of Washington, Seattle, 1993.
[7] R. C. Griffiths and S. Tavare. Ancestral inference in population genetics. *Statistical Science*, 9:307–319, 1994.
[8] S. B. Hedges and L. L. Poling. A molecular phylogeny of reptiles. *Science*, 283(5404):945–946, Feb 12 1999.
[9] J. Hein. An algorithm for statistical alignment of sequences related by a binary tree. In *Pac. Symp. Biocomp.*, pages 179–190. World Scientific, 2001.
[10] J. Hein, J. L. Jensen, and C. N. S. Pedersen. Recursions for statistical multiple alignment. Technical Report 425, Dept. of Theor. Stat., Univ. of Aarhus, January 2002.
[11] J. Hein, C. Wiuf, B. Knudsen, M. B. Møller, and G. Wibling. Statistical alignment: Computational properties, homology testing and goodness-of-fit. *J. Mol. Biol.*, 302:265–279, 2000.
[12] I. Holmes and W. J. Bruno. Evolutionary HMMs: a Bayesian approach to multiple alignment. *Bioinformatics*, 17(9):803–820, 2001.
[13] J. P. Huelsenbeck and F. Ronquist. MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics*, 2001.
[14] J.L. Jensen and J. Hein. Gibbs sampler for statistical multiple alignment. Technical Report 429, Dept. of Theor. Stat., U. Aarhus, September 2002.
[15] T. H.. Jukes and C. R. Cantor. Evolution of protein molecules. In Munro, editor, *Mammalian Protein Metabolism*, pages 21–132. Acad. Press, 1969.
[16] M. K. Kuhner, J. Yamato, and J. Felsenstein. Estimating effective population size and mutation rate from sequence data using Metropolis-Hastings sampling. *Genetics*, 140(4):1421–1430, 1995.
[17] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2001.
[18] G. A. Lunter, I. Miklós, Y. S. Song, and J. Hein. An efficient algorithm for statistical multiple alignment on arbitrary phylogenetic trees. *J. Comp. Biol.*, 2003. In press.
[19] I. Miklós. An improved algorithm for statistical alignment of sequences related by a star tree. *Bul. Math. Biol.*, 64:771–779, 2002.
[20] I. Miklós, G. A. Lunter, and I. Holmes. A "long indel" model for evolutionary sequence alignment. In preparation.
[21] O. G. Pybus, A. J. Drummond, T. Nakano, B. H. Robertson, and A. Rambaut. The epidemiology and iatrogenic transmission of hepatitis c virus in Egypt: a Bayesian coalescent approach. *Mol Biol Evol*, 20(3):381–387, 2003.
[22] O. G. Pybus, A. Rambaut, and P. H. Harvey. An integrated framework for the inference of viral population history from reconstructed genealogies. *Genetics*, 155(3):1429–1437, 2000.
[23] M. Steel and J. Hein. Applying the Thorne-Kishino-Felsenstein model to sequence evolution on a star-shaped tree. *Appl. Math. Let.*, 14:679–684, 2001.
[24] M. Stephens and P. Donnelly. Inference in molecular population genetics. *J. of the Royal Stat. Soc. B*, 62:605–655, 2000.
[25] D. Swofford. Paup* 4.0. Sinauer Associates, 2001.
[26] J. L. Thorne, H. Kishino, and J. Felsenstein. An evolutionary model for maximum likelihood alignment of DNA sequences. *J. Mol. Evol.*, 33:114–124, 1991.
[27] S. Whelan, P. Lió, and N. Goldman. Molecular phylogenetics: state-of-the-art methods for looking into the past. *Trends in Gen.*, 17:262–272, 2001.
[28] I. J. Wilson and D. J. Balding. Genealogical inference from microsatellite data. *Genetics*, 1998.