Finding Maximum Likelihood Indel Scenarios

Abdoulaye Baniré Diallo[1,2], Vladimir Makarenkov[2], and Mathieu Blanchette[1]

[1] McGill Centre for Bioinformatics and School of Computer Science, McGill University, 3775 University Street, Montréal, Québec, H3A 2B4, Canada
[2] Département d'informatique, Université du Québec à Montréal, C.P. 8888, Succ. Centre-Ville, Montréal (Québec), H3C 3P8, Canada

Abstract. Given a multiple alignment of orthologous DNA sequences and a phylogenetic tree for these sequences, we investigate the problem of reconstructing the most likely scenario of insertions and deletions capable of explaining the gaps observed in the alignment. This problem, that we called the Indel Maximum Likelihood Problem (IMLP), is an important step toward the reconstruction of ancestral genomics sequences, and is important for studying evolutionary processes and genome function. We solve the IMLP using a new type of tree hidden Markov model whose states correspond to single-based evolutionary scenarios and transitions model dependencies between neighboring columns. The standard Viterbi and Forward-backward algorithms are optimized to produce the most likely ancestral reconstruction and to compute the level of confidence associated to specific regions of the reconstruction. The method is illustrated on a set of 85kb sequences from eight mammals.

KEYWORDS: Ancestral genome reconstruction; Insertions and deletions; Tree-HMM; Ancestral mammalian genomes

1 Introduction

It has recently been shown that the phylogeny of eutherian mammals is such that an accurate reconstruction of the genome of an early ancestral mammal is possible [1]. The ancestral genome reconstruction procedure involves several difficult steps, including the identification of orthologous regions in different extant species, ordering of syntenic blocks, multiple alignment of orthologous sequences within each syntenic block, and reconstruction of ancestral sequences for each aligned block. This last step involves the inference of the set of substitutions, insertions, and deletions that have may have produced a given set of multiply-aligned extant sequences. While the problem of reconstructing substitutions scenarios has been well studied (starting with Felsenstein (1981) [7]), the inference of insertions and deletions scenarios has received less attention (but see the seminal contribution of Thorne, Kishino and Felsenstein [19]). The difficulty of the problem is due in large part to the fact that insertions and deletions (indels) often affect several consecutive nucleotides, so the columns of the alignment cannot be treated independently, as opposed to the maximum likelihood

problem for substitutions [7]. The reconstruction of the most parsimonious scenario of indels required to explain a given multiple sequence alignment has been shown to be NP-Complete by Chindelevitch [5] *et al.*, but good heuristics have been developed by Blanchette *et al.* [1], Chindelevitch *et al.* [5], and Fredslund *et al.* [9].

A maximum likelihood reconstruction would be preferable to a most parsimonious reconstruction because it would be more accurate and would allow to estimate the uncertainty related to certain aspects of the reconstruction. Similarly to statistical alignment approaches [13] (which unfortunately remain too slow for genome-wide reconstructions), we seek to gain a richer insight into ancestral sequences and evolutionary processes. In this paper, we thus focus on the problem we call the *Indels Maximum Likelihood Problem (IMLP)*. It consists of inferring the set of insertions and deletions that has the maximal likelihood, according to some fixed evolutionary parameters, and that could explain the gaps observed in a given alignment. An example of the input and output of this problem are shown in Figure 1. Indel evolutionary scenarios are useful for several other problems such as annotating functional regions of extant genomes, including protein-coding regions [17], RNA genes [15], and other types of functional regions [16].

Here, we start by giving a formal definition of the Indel Maximum Likelihood Problem. To solve the problem, we use a special type of tree Hidden Markov Model, which is a combination of a standard Hidden Markov Model and a phylogenetic tree. We show how the most likely path through the tree-HMM leads to the most likely indel scenario and how a variant of the standard Viterbi algorithm can solve the problem. Although the size of the HMM is exponential in the number of extant species considered, we show how the knowledge given by the phylogenetic tree and the aligned sequences allows the state space of the HMM to be considerably reduced, resulting in a practical, yet exact, algorithm. Our implementation is able to solve large problems on a simple desktop computer and allows for an easy parallelization. Finally, we assess the complexity and accuracy of our algorithm on a multiple alignment of eight orthologous mammalian genomic sequences of ∼50kb each.

2 The Indel Maximum Likelihood Problem

In this section we will give a precise definition for the Indel Maximum Likelihood Problem (IMLP). Consider a rooted binary phylogenetic tree $T = (V_T, E_T)$ with branch lengths $\lambda : V_T \to \mathbb{R}^+$. If n is the number of leaves of T, there are $n - 1$ internal nodes and $2n - 2$ edges.

Consider a multiple alignment A of n orthologous sequences corresponding to the leaves of the tree T. Since the only evolutionary events of interest here are insertions and deletions, A can be transformed into a binary matrix, where gaps are replaced by 0's and nucleotides by 1's. Let A_x be the row of the binarized alignment corresponding to the sequence at leaf x of T, and let $A_x[i]$ be the binary character at the i-th position of A_x. Assume that the alignment A con-

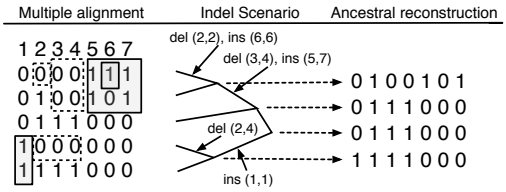

Fig. 1. Example of an input and output to the Indel Maximum Likelihood Problem. The input consists of a multiple alignment (shown on the left in binary format) and the topology and branch lengths of the phylogenetic tree. The output consists of a set of insertions and deletions, placed along the edges of the tree, explaining the gaps (zeros) in the alignment. The dashed boxes in the alignment indicate the deletions and the shaded boxes indicate the insertions of the scenario shown on the right. This set of operations yields the ancestral reconstruction shown on the right.

tains L columns, we add for convenience two extra columns, $A[0]$ and $A[L+1]$, consisting exclusively of 1's.

Definition 1 (Ancestral reconstruction). *Given a multiple alignment A of n extant sequences assigned to the leaves of a tree T, an* ancestral reconstruction *A^* is an extension of A that assigns a sequence $A^*_u \in \{0,1\}^{L+2}$ to each node u of T, and where $A^*_u = A_u$ whenever u is a leaf.*

An ancestral reconstruction thus specifies, for each ancestral node of T, what positions were occupied by a nucleotide and what positions had a gap (see Figure 1 for an example). The following restriction on the set of possible ancestral reconstructions is necessary in some contexts.

Definition 2 (Phylogenetically correct ancestral reconstruction). *An ancestral reconstruction A^* is* phylogenetically correct *if, for any $u, v, w \in V_T$ such that w is located on the path between u and v in T, we have $A^*_u[i] = A^*_v[i] = 1 \implies A^*_w[i] = 1$.*

Requiring an ancestral reconstruction to be phylogenetically correct corresponds to assuming that any two nucleotides that are aligned in A have to be derived from a common ancestor, and thus that all the ancestral nodes between them have to have been a nucleotide. This prohibits aligned nucleotides to be the result of two independent insertions. Assuming that this property holds perfectly for a given alignment A is somewhat unrealistic, but, for mammalian sequences, good alignment heuristics have been developed (e.g. TBA [2], MAVID [3], MLAGAN [4]) and have been shown to be very accurate [2]. In the future, we plan to relax this assumption, but, for now, we will concentrate only on finding phylogenetically correct ancestral reconstructions.

Since we are considering insertions and deletions affecting several consecutive characters, we delimit each operation by the positions s and e in the aligned

sequences where it starts and ends. Let x and y be two nodes of the tree, where x is the parent of y. The alignment consisting of rows A_x^* and A_y^* is divided into a set of regions defined as follows (see Figure 2).

Definition 3 (Deletions, Insertions, Conservations, and Length).

- *The region* (s, e) *is a* deletion *if (a) for all* $i \in \{s, \ldots, e\}, A_y^*[i] = 0$*, (b)* $A_x^*[s] = A_x^*[e] = 1$*, and (c) no region* $(s', e') \supset (s, e)$ *is a deletion (i.e. we only consider regions that are maximal).*
- *The region* (s, e) *is an* insertion *if (a) for all* $i \in \{s, \ldots, e\}, A_x^*[i] = 0$*, (b)* $A_y^*[s] = A_y^*[e] = 1$*, and (c) no region* $(s', e') \supset (s, e)$ *is an insertion.*
- *The region* (s, e) *is a conservation if (a) for all* $i \in \{s, \ldots, e\}, A_x^*[i] = A_y^*[i]$ *and (b) no region* $(s', e') \supset (s, e)$ *is a conservation.*
- *The* length *of region* (s, e) *is the number of non-trivial positions it contains:* $l(s, e) = |\{s \leq i \leq e | A_x^*[i] \neq 0 \text{ or } A_y^*[i] \neq 0\}|$.

A pair of binary alignment rows A_x^* and A_y^* can thus be partitioned into a set of non-overlapping insertions, deletions, and conservations.

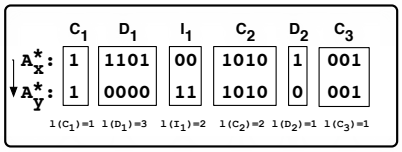

Fig. 2. Example of the partition of a paiwise alignment of A_x^* and A_y^*, where x is the parent of y. The length of each region is given below the region.

Definition 4 (Indel scenario). *The indel scenario defined by an ancestral reconstruction* A^* *is the set of insertions and deletions that occurred between the ancestral reconstructions at adjacent nodes in* T*.*

All that remains is to define an optimization criterion on A^*. Two main choices are possible: a parsimony criterion or a likelihood criterion.

2.1 The Indel Parsimony Problem

The parsimony approach for the indel reconstruction problem has been introduced by Fredslund *et al.* [9] and Blanchette *et al.* [1]. In its simplest version, it attempts to find the phylogenetically correct ancestral reconstruction A^* that minimizes the total number of insertions and deletions defined by A^*:

$$indelParsimony(A^*) = \sum_{u,v:(u,v)\in E} |\{(s, e) : (s, e) \text{ is a deletion or an insertion from } A_u^* \text{ to } A_v^*\}|$$

The Indel Parsimony Problem is NP-Hard [5]. Most authors have studied a weighted version of the IPP where the cost of indels depends linearly on their length (affine gap penalty). Blanchette *et al.* [1] proposed a greedy algorithm, and good exact heuristics have been developed [5, 9]. The limitation of these approaches is that they only give a single solution as output, and provide no measure of uncertainty of the various parts of the reconstruction. In contrast, a likelihood-based approach has the potential of providing a more accurate solution and a richer description of set of possible solutions.

2.2 Indel Maximum Likelihood Problem

In this section, we define the indel reconstruction problem in a probabilistic framework and similar to the Thorne-Kishino-Felsenstein model [18]. To this end, we need to define the probability of transition between an alignment row A_x^* and its descendant row A_y^*. This probability will be defined as a function of the probability of the insertions, deletions, and conservations that happened from A_x^* to A_y^*.

Let $P_{DelStart}(\lambda(b))$ be the probability that a deletion starts at a given position in the sequence, along a branch b of length $\lambda(b)$, and $P_{InsStart}(\lambda(b))$ is defined similarly. We assume that these probabilities only depend on the length $\lambda(b)$ of the branch b along which they occur, but not on the position where the indel occurs. A reasonable choice is $P_{DelStart}(\lambda(b)) = 1 - e^{-\psi_D \lambda(b)}$ and $P_{InsStart}(\lambda(b)) = 1 - e^{-\psi_I \lambda(b)}$, for some deletion and insertion rate parameters ψ_D and ψ_I, but our algorithm allows for any other choice of these probabilities. Thus, the probability that none of the two events happens at a given position, which we call the probability of a conservation, is given by $P_{Cons}(\lambda(b)) = e^{-(\psi_D+\psi_I)\lambda(b)}$. We assume that the length of a deletion follows a geometric distribution, where the probability of a deletion of length k is $\alpha_D^{k-1}(1-\alpha_D)$ and the probability of an insertion of length k is $\alpha_I^{k-1}(1-\alpha_I)$. One can thus see α_D (resp. α_I) as the probability of extending a deletion (resp. insertion). This assumption, necessary to design a fast algorithm, holds relatively well for short indels, but fails for longer ones [12]. Our algorithm allows the parameters α_D and α_I to depend on the branch b, but the results reported in Section 5 $correspondtothecasewherealpha_D$ and α_I were held constant across the tree. The probability that alignment row A_x^* was transformed into alignment row A_y^* along branch b can be defined as follows:

$$\begin{aligned}\Pr(A_y^*|A_x^*, b) = &\prod_{(s,e)\text{: deletion from } A_x^* \text{ to } A_y^*} P_{DelStart}(\lambda(b)) \cdot (\alpha_D^{l(s,e)-1}(1-\alpha_D)) \cdot \\ &\prod_{(s,e)\text{: insertion from } A_x^* \text{ to } A_y^*} P_{InsStart}(\lambda(b)) \cdot (\alpha_I^{l(s,e)-1}(1-\alpha_I)) \cdot \\ &\prod_{(s,e)\text{:conservation from } A_x^* \text{ to } A_y^*} (P_{Cons}(\lambda(b))^{l(s,e)}\end{aligned}$$

This allows us to formulate precisely the problem addressed in this paper:
INDEL MAXIMUM LIKELIHOOD PROBLEM (IMLP):
Given: A multiple sequence alignment A of n orthologous sequences related by a phylogenetic tree T with branch lengths λ, a probability model for insertions and deletions specifying the values of ψ_D, ψ_I, α_D, and α_I.
Find: A maximum likelihood phylogenetically correct ancestral reconstruction A^* for A, where the likelihood of A^* is:

$$L(A^*) = \prod_{b=(x,y)\in E_T} \Pr[A^*_y | A^*_x, b]$$

3 A Tree-Hidden Markov Model

In this section, we describe the tree-based hidden Markov model that is used to solve the IMLP. A tree-Hidden Markov Model (tree-HMM) is a probabilistic model that allows two processes to occur, one in time (related to the sequence history in a given column of A), and one in space (related to the changes toward the neighboring columns). Tree HMMs were introduced by Felsenstein and Churchill [8] and Yang [20] to improve the phylogenetic models that allows for variation among sites in the rate of substitution, and have since then been used for several other purposes (detecting conserved regions [16] and predicting genes [17]). Just as any standard HMM [6], a tree-HMM is defined by three components: the set of states, the set of emission probabilities, and the set of transition probabilities.

3.1 States

Intuitively, each state corresponds to a different single-column indel scenario (although additional complication are described below). Given a rooted binary tree $T = (V_T, E_T)$ with n leaves, each state corresponds to a different labeling of the edges E_T with one of three possible events: I (for insertion), D (for deletion), or C (for conservation). The set $\mathcal{S}$ of possible states of the HMM would then be $\mathcal{S} = \{I, D, C\}^{2n-2}$. However, this definition is not sufficient to model certain biological situations (see Figure 3). We will use the '*' symbol to indicate that, along a certain branch $b = (x, y)$, no event happened because there was a base neither at node x nor at node y. This will happen in two situations: when edge b is a descendant of edge b' that was labeled with D (i.e. the base was deleted higher up the tree), and when there exists an edge b' that is not between b and the root and that is labeled with I (i.e. an insertion happened elsewhere in the tree). The fact that these extraneous events can potentially interrupt ongoing events along branch b means that the HMM needs to have a way to remember what event was actually going on along that branch. This transmission of memory from column to column is achieved by three special labels: $I^*, D*$, and $C*$, depending on whether the * regions is interrupting an insertion, deletion, or conservation. Thus, we have $\mathcal{S} \subseteq \{I, D, C, I^*, D^*, C^*\}^{2n-2}$. Although this state space appears

prohibitively large (6^{2n-2}), the reality is that a number of these states cannot represent actual indel scenarios, and can thus be ignored. The following set of rules specify what states are valid.

Definition 5 (Valid states). *Given a tree $T = (V_T, E_T)$, a state s assigning a label $s(b) \in \{I, D, C, I^*, D^*, C^*\}$ to each branch $b \in E_T$ is* valid *if the two following conditions hold.*

- *(Phylogenetic correctness condition) There must be at most one branch b such that $s(b) = I$.*
- *(Star condition) Let $b \in E_T$, and let $anc(b) \subset E_T$ be the set of branches on the path from the root to b. Then $s(b) \in \{I^*, D^*, C^*\}$ if and only if $\exists b' \in anc(b)$ such that $s(b') = D$ or $\exists b' \in (E_T \setminus anc(b))$ such that $s(b') = I$.*

The number of valid states on a complete balanced phylogenetic tree with n leaves is $O(n \cdot 3^{2n})$ (the number is dominated by states that have a 'I' on a branch leading to a leaf, which leaves all other $2n - 3$ edges free to be labeled with either C^*, D^*, or I^*). Although this number remains exponential, it is significantly better than the 6^{2n-2} valid and invalid states.

3.2 Emission probabilities

In an HMM, each state emits one symbol, according a certain emission probability distribution. In our tree-HMMs, each state emits a collection of symbols, corresponding to the set of characters obtained at the leaves of T when indel scenario s occurs. Intuitively, we can think of a state as emitting an alignment column. The following definition formalizes this.

Definition 6. *Let s be a valid state for tree $T = (V_T, E_T)$ with root r. Then, we define the output of state s as a function $O_s : V_T \to \{0, 1\}$ with the following recursive properties:*

1. $O_s(root) = \begin{cases} 0, & \text{if } \exists x \in V_T \text{ such that } s(x) = I \\ 1, & \text{otherwise} \end{cases}$.
2. *Let $e = (x, y) \in E_T$, with x being the parent of y. Then,*

$$O_s(y) = \begin{cases} 0, & \text{if } s(e) = D \\ 1, & \text{if } s(e) = I \\ O_s(x), & \text{otherwise} \end{cases}$$

Let C be an alignment column (i.e. an assignment of 0 or 1 to each leaf in T). We then have the following degenerate emission probability for state s:

$$Pr_e[C|s] = \begin{cases} 1 \text{ if } O_s(x) = C(x) \text{ for all } x \in leaves(T) \\ 0 \text{ otherwise} \end{cases}$$

Thus, each state s can emit a single alignment column C. However, many different states can emit the same column.

3.3 Transition probabilities

The last component to be defined is the set of transition probabilities of the tree-HMM. The probability of transition from state s to state s', $\Pr_t[s'|s]$ is a function of set of events that occurred along each edge of T. Intuitively, $\Pr_t[s'|s]$ describes the probability of the single-column indel scenario s', given that scenario s occurred at the previous column. This transition probability is a function of insertions and deletions that started between the two columns, of those that were extended going from one column to the next. Specifically, we have $\Pr_t[s'|s] = \prod_{b \in E_T} \rho[s'(e)|s(e), b]$, where ρ is given in Table 1.

$s(e) \setminus s(e)'$	C	D	I	C^*	D^*	I^*
C	$P_{Cons}(\lambda(b))$	$P_{DelStart}(\lambda(b))$	$P_{InsStart}(\lambda(b))$	1	0	0
D	$(1-\alpha_D)P_{Cons}(\lambda(b))$	α_D	$(1-\alpha_D)P_{InsStart}(\lambda(b))$	0	1	0
I	$(1-\alpha_I)P_{Cons}(\lambda(b))$	$(1-\alpha_I)P_{DelStart}(\lambda(b))$	α_I	0	0	1
C^*	$P_{Cons}(\lambda(b))$	$P_{DelStart}(\lambda(b))$	$P_{InsStart}(\lambda(b))$	1	0	0
D^*	$(1-\alpha_D)P_{Cons}(\lambda(b))$	α_D	$(1-\alpha_D)P_{InsStart}(\lambda(b))$	0	1	0
I^*	$(1-\alpha_I)P_{Cons}(\lambda(b))$	$(1-\alpha_I)P_{DelStart}(\lambda(b))$	α_I	0	0	1

Table 1. Edges transition table $\rho[s'(e)|s(e), b]$. Notice that ρ is not a transition probability matrix, since its rows sum to more than one.

3.4 Tree-HMM paths and ancestral reconstruction

We now show how the tree-HMM described above allows us to solve the IMLP. Consider a multiple alignment A of length L on a tree T. A path π in the tree-HMM is a sequence of states $\pi = \pi_0, \pi_1, ..., \pi_L, \pi_{L+1}$. Based on standard HMM theory, we get:

$$Pr[\pi, A] = Pr[\pi_0, A_0] \prod_{i=1}^{L+1} Pr_e[A[i]|\pi_i] \cdot Pr_t[\pi_i|\pi_{i-1}]$$

Figure 3 gives an example of an alignment with some of the non-zero probability paths associated.

Theorem 1. *Consider an alignment A on tree T. Then $\pi^* = argmax_\pi \Pr[\pi, A]$ yields the most likely indel scenario for A, and a maximum likelihood ancestral reconstruction A^* is obtained by setting $A^*_u[i] = O_{\pi^*_i}(u)$.*

Proof. It is simple to show that for any ancestral reconstruction $\hat{A}$ for A, we have $L(\hat{A}) = \Pr[\pi, A]$, where π is the path corresponding to $\hat{A}$. Thus, maximizing $\Pr[\pi, A]$ maximizes $L(\hat{A})$.

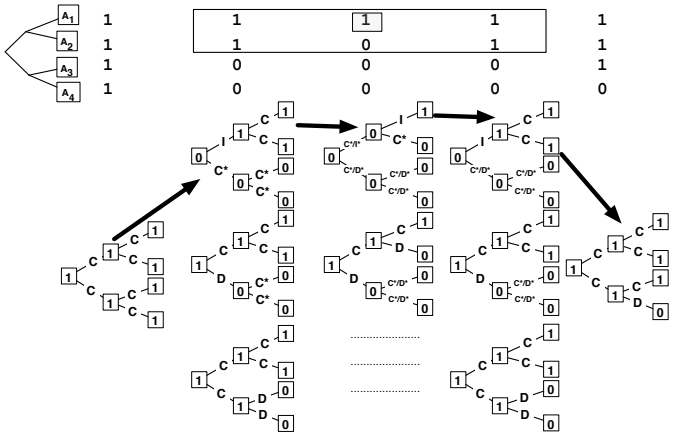

Fig. 3. The set of valid, non-zero probability states associated to the multiple alignment given at the top of the figure. When edges are labeled with more than one character (e.g. C^*, D^*), the tree represents several possible states. For the third column, not all possible states are shown. Arrows indicate one possible path through the tree-HMM. This path corresponds to two interleaved insertions, shown by two boxes in the alignment, illustrating the need for the I^* character.

4 Computing the most likely path

To compute the most likely path π^* through a tree-HMM, we adapted the standard Viterbi dynamic programming algorithm [6]. Let

$$X(i,k) = \max_{\substack{\pi = \pi_0, \pi_1, ..., \pi_i \\ \text{such that } \pi_i = k}} \Pr[\pi, A[1...i]]$$

be the likelihood of the most probable path ending at state k for the i first columns of the alignment. Let $c \in \mathcal{S}$ be the state made of C's on all edges of T. Since the dummy column $A[0]$ consists exclusively of 1's, c is the only possible initial state. For any i between 0 and $L+1$ and for any valid state $s \in \mathcal{S}$, we can compute $X(i,s)$ as follows:

$$X(i,s) = \begin{cases} 1, & \text{if } i = 0 \text{ and } s = c \\ 0, & \text{if } i = 0 \text{ and } s \neq c \\ Pr_e[A[i]|s] \cdot \max_{s' \in \mathcal{S}}(X(i-1,s') \cdot Pr_t[s|s']), & \text{if } i > 0 \end{cases}$$

Finally, π^* is obtained by tracing back the dynamic programming, starting from entry $X(L+1,c)$. The running time of a naive implementation of the Viterbi algorithm is $O(|\mathcal{S}|^2 L)$, which quickly becomes impractical as the size of the tree T grows. In the next section, we show how to make this computation practical for moderately large trees and for long sequences.

4.1 Viterbi optimizations

The previous implementation of the Viterbi algorithm cannot be run for large sequences and number of taxa that is greater to 8 due the number of states that is $O(n3^{2n})$. Even though the number of states is exponential, most of alignment columns can only be generated with non-zero probability by a much more manageable number of states. Given an alignment A, it is possible to compute, for each column $A[i]$, the set S_i of valid states that can emit $A[i]$ with non-zero probability. For instance, an alignment column with only 1's will lead to only one possible state, independently of the number taxa n. To compute only the valid states for a given column of the alignment, we used a divide and conquer approach that is presented in Algorithm 1. The idea behind this algorithm is to compute partial valid states for subtrees and to merge these subtrees while keeping only valid merged states. The process is done in a bottom up fashion until the root of the tree is reached.

Although the sets of possible states $S_0, ..., S_{L+1}$ obtained from this algorithm are generally relatively small, more improvements are possible. This is because the transition probability between most pairs of states is zero. We can thus remove from S_i any state s such that the transition to s from any state in S_{i-1} has probability zero. Proceeding from left to right, we get $S'_0 = S_0$, and $S'_i = \{s \in S_i | \exists t \in S'_{i-1} \text{ s.t. } Pr_t[s|t] > 0\}$. For instance, if, in all states of S_{i-1}, an edge e is labeled by deletion D, then none of the states in S_i can have edge e labeled with $C*$ or $I*$. This yields a huge improvement for alignment regions consisting of a number of adjacent positions with a base in only one of the n species and ensures that the algorithm will be practical for relatively large number of sequences (see Section 5).

Algorithm 1 buildValidState(node *root*, C)

Require: *root*: a tree node, C: an alignment column.
Ensure: Set of valid, non-zero probability states for C.

```
1: if root is a leaf then
2:     return list of possible operations according to the character at that leaf
3: else
4:     leftList = buildValidState(root.left, C)
5:     rightList = buildValidState(root.right, C)
6:     return mergeSubtrees(leftList, rightList, root)
7: end if
```

4.2 Forward-backward algorithm

A significant advantage of the maximum likelihood approach over the parsimony approach is that it allows evaluating the uncertainty related to certain aspects of the reconstruction. For example, it is useful to be able to compute the probability that a base was present at a given position i of a given ancestral

Algorithm 2 mergeSubtrees(StateList $leftList$, StateList $rightList$, node $root$)

Require: $leftList$ and $rightList$: the lists of partial states, $root$: a tree node.
Ensure: Set of valid, non-zero probability states combining elements in $leftList$ and $rightList$.

```
for all partial states l in leftList do
  for all partial states r in rightList do
    if compatible(l, r) == true        #merging those two partial states yields a valid partial state then
      m = merge(l, r)
      if root == initialroot then
        mergedList.add(m)
      else
        for all operations on a branch op do
          if isPossibleUpstream(m,op)        #Checks if op can legally be added to m then
            mergedList.add(addAncestorBranch(m,op))
          end if
        end for
      end if
    end if
  end for
end for
return mergedList
```

node u: $\Pr[A^*_u[i] = 1|A] = \sum_{s \in \mathcal{S}: O_s(u)=1} \Pr[\pi_i = s|A]$. This allows the computation of the probability of making an incorrect prediction at a given position of a given ancestor. The forward-backward is a standard HMM algorithm to compute $\Pr[\pi_i = s|A]$ (see [6] for details):

$$F(i,s) = \begin{cases} 1, & \text{if } i = 0 \text{ and } s = c \\ 0, & \text{if } i = 0 \text{ and } s \neq c \\ Pr_e[A[i]|s] \cdot \sum_{s' \in \mathcal{S}'_{i-1}} (F(i-1,s') \cdot Pr_t[s|s']), & \text{if } i > 0 \end{cases}$$

$$B(i,l) = \begin{cases} 1, & \text{if } i = L+1 \text{ and } l = c \\ 0, & \text{if } i = L+1 \text{ and } l \neq c \\ \sum\limits_{s' \in \mathcal{S}'_{i+1}} Pr_e[A[i+1]|s'] \cdot F(i+1,s') \cdot Pr_t[s'|s]), & \text{if } i < L+1 \end{cases}$$

$$\Pr[\pi_i = s|A] = \frac{F(i,s)B(i,s)}{\sum_{s' \in S'_i} F(i,s')B(i,s')}$$

The optimizations developed for the Viterbi algorithm can be also directly applied to the Forward-Backward algorithm.

5 Results

Our tree-HMM algorithm was implemented as a C program that is available upon request. The program was applied to a 50kb region of chromosome 13 of human,

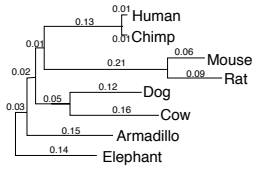

Fig. 4. Phylogenetic tree for the eight species studied in this paper.

Ancestor	**% of agreement**
Cow + Dog	99.38
Mouse + Rat	97.83
Human + Chimp	99.83
Human + Chimp + Mouse + Rat	99.33
Human + Chimp + Mouse + Rat + Cow + Dog	98.41
Human + Chimp + Mouse + Rat + Cow + Dog + Armadillo	94.13
Human + Chimp + Mouse + Rat + Cow + Dog + Armadillo + Elephant	89.01

Table 2. Percentage of alignment columns where there is agreement between the ancestor reconstructed by the greedy algorithm of Blanchette *et al.* [1] and that predicted by our maximum-likelihood algorithm.

together with orthologous regions in 7 other species of mammals: chimp, mouse, rat, cow, dog, armadillo, and elephant[3] [14]. This region is representative of the whole genome, and contains coding, intergenic regions, and intronic regions. The multiple alignment of these regions, computed using TBA [2], contains 85,000 columns. The phylogenetic tree used for the alignment and for the reconstruction is shown in Figure 4. The branch lengths are based on rates of substitution estimated on a genome-wide basis. The parameters of the indel model were set as follows: $\psi_D = 0.05, \psi_I = 0.05, \alpha_D = 0.9$ and $\alpha_I = 0.9$.

We first compared the maximum likelihood ancestral reconstruction found using our Viterbi algorithm to the ancestors inferred using the greedy algorithm of Blanchette *et al.* [1]. Table 2 shows the degree of agreement between the two reconstructed ancestors, for each ancestral node. We observe that both methods agree to a very large degree. The most disagreement concerns the ancestor at the root of the eutherian tree, which, in the absence of an outgroup, cannot be reliably predicted by any method. We expect that in most cases of disagreement, the maximum likelihood is the most likely to be correct, although the opposite may be true in case of gross model violations [11].

The main strength of the likelihood-based method is its ability to measure uncertainty, using the forward-backward algorithm, which something that no previous method allowed. The probability that the maximum posterior probability reconstruction is correct is simply given by $\max\{\Pr[A_u^*[i] = 1|A], 1 - \Pr[A_u^*[i] =$

[3] In the case of cow, armadillo, and elephant, the sequence is incomplete and a small fraction of the bases are missing.

$1|A]\}$. For example, if $\Pr[A_u^*[i] = 1|A] = 0.3$, then the maximum posterior probability reconstruction would predict $A_u^*[i] = 0$, and would be right with probability 0.7. Figure 5 shows the distribution of this probability of correctness, for each ancestral node in the tree. We observe, for example, that 97.7% of the positions in the Boreoeutherian ancestor (the human+chimp+mouse+rat+cow+dog ancestor, living approximately 70 million years ago), are reconstructed with a confidence level above 99% [4]. The ancestor that is the easiest to reconstruct confidently is obvious the human-chimp ancestor, where less than 0.5% of the columns have a confidence level below 99%. Again, the root of the tree is the node that is the most difficult to be reconstructed confidently, because of the absence of an outgroup. Overall, this shows that most positions of most ancestral nodes can be reconstructed very accurately, and that we can identify the few positions where the reconstruction is uncertain.

A potential drawback of the tree-HMM method is that it's running time is, in the worst case, exponential on the number of sequences being compared. However, the optimizations described in this paper greatly reduce this number, so the algorithm remains quite fast. Our optimized Viterbi algorithm produced its maximum likelihood ancestral predictions on the 8-species alignment of 85,000 columns in two hours in an intel Pentium IV machine (3.2 Ghz), while the forward-backward algorithm produced an output after approximately four hours. Figure 6 shows the distribution of the number of states that were actually considered, per alignment column, in the case of the 8-species alignment of 85,000 columns. Most alignment column are actually associated to less than 50 states. However, a small number of columns are associated to a very large number of states (8 columns have more than 21,000 states). Fortunately, these columns are rarely consecutive, so the incurred running time is not catastrophic.

6 Discussion and Future Work

The method developed here allows predicting maximum likelihood indel scenarios and their resulting ancestral sequences for reasonably large alignments. Furthermore, it allows the estimation of the probability of error in any part of the prediction, using the forward-backward algorithm. Integrated into the pipeline for whole-genome ancestral reconstruction, it will improve the quality of the predictions and allow richer analyses. The main weakness of our approach is that it assumes that a correct phylogenetic alignment is given as input. While many existing multiple alignment programs have been shown to be quite accurate on mammalian genomic sequences (including non-functional or repetitive regions) [2], it has also been shown that a sizeable fraction of reconstruction errors is due to incorrect alignments [1]. Ideally, one would include the optimization of the alignment directly in the indel reconstruction problem, as originally suggested by Hein [10]. However, with the exception of statistical alignment approaches

[4] We need to to keep in mind, though, that these numbers assume the correctness of the multiple alignment, as well as that of the branch lengths and indel probability model, so that they do not reflect the true correctness of the reconstructed ancestor.

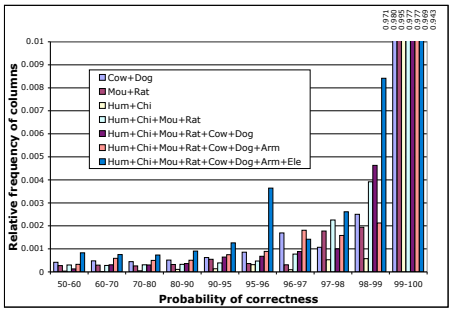

Fig. 5. Distribution of the confidence levels, over all 85,000 columns, for each ancestor. The vast majority of the ancestral positions are reconstructed with a probability of correctness above 99% (assuming the correctness of the alignment).

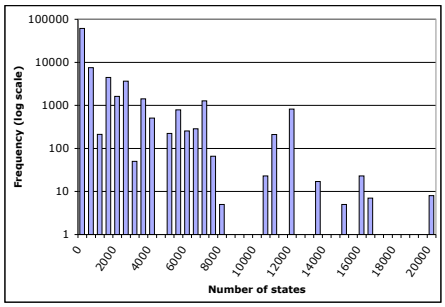

Fig. 6. Distribution of the number of states considered ($|S'_i|$), over all 85,000 positions.

[13] (which remains too slow to be applicable on a genome-wide scale), genomic multiple alignment methods do not treat indels in a probabilistic framework. We are thus investigating the possibility of using the method proposed here to detect certain types of small-scale alignment errors, and to suggest corrections.

When predicting ancestral genomic sequences, it is very important to be able to quantify the uncertainty with respect to certain aspects of the reconstruction. Our forward-backward algorithm calculates this probability of error for each position of each ancestral species. However, errors in adjacent columns are not independent: if position i is incorrectly reconstructed, it is very likely that position $i+1$ will be wrong too. We are currently working on models to represent this type of correlated uncertainties. This new type of representation will play an important role in the analysis and visualization of ancestral reconstructions.

Finally, to be applicable to complete genomes, and to scale up to the * *20 mammalian genomes that will soon be available, our algorithm will require further optimizations. These will probably require us to move away from an exact algorithm toward approximation algorithms.

7 Aknowledgements

A.B.D. is an NSERC fellow. We thank Éric Gaul, Éric Blais, Adam Siepel, and the group of participants to the First Barbados Workshop on Paleogenomics for their useful comments. We thank Webb Miller and David Haussler for providing us with the sequence alignment data.

References

1. M. Blanchette, E. D. Green, W. Miller, and D. Haussler. Reconstructing large regions of an ancestral mammalian genome in silico. *Genome Res*, 14(12):2412–

2423, Dec 2004.
2. M. Blanchette, W. J. Kent, C. Riemer, L. Elnitski, A. F. A. Smit, K. M. Roskin, R. Baertsch, K. Rosenbloom, H. Clawson, E. D. Green, D. Haussler, and W. Miller. Aligning multiple genomic sequences with the threaded blockset aligner. *Genome Research*, 14(4):708–715, Apr 2004.
3. N. Bray and L. Pachter. MAVID: constrained ancestral alignment of multiple sequences. *Genome Research*, 14(4):693–699, Apr 2004.
4. M. Brudno, C. B. Do, G. M. Cooper, M. F. Kim, E. Davydov, E. D. Green, A. Sidow, and S. Batzoglou. LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA. *Genome Research*, 13(4):721–731, Apr 2003.
5. L. Chindelevitch, Z. Li, E. Blais, and M. Blanchette. On the inference of parsimonious indel evolutionary scenarios. *Journal of Bioinformatics and Computational Biology*, 0:In press, 2006.
6. R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis*. Cambridge University Press, 1998.
7. J. Felsenstein. Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of Molecular Evolution*, 17:368–376, 1981.
8. J. Felsenstein and G. Churchill. A hidden markov model approach to variation among sites in rate of evolution. *Mol. Biol. Evol.*, 13:93–104, 1996.
9. J. Fredslund, J. Hein, and T. Scharling. A large version of the small parsimony problem. In *Proceedings of the 4th Workshop on Algorithms in Bioinformatics (WABI)*, 2004.
10. J. Hein. A method that simultaneously aligns, finds the phylogeny and reconstructs ancestral sequences for any number of ancestral sequences. *Molecular Biology and Evolution*, 6(6):649–668, 1989.
11. A. Hudek and D. Brown. Ancestral sequence alignment under optimal conditions. *BMC Bioinformatics*, 6:273:1–14, 2005.
12. W. J. Kent, R. Baertsch, A. Hinrichs, W. Miller, and D. Haussler. Evolution's cauldron: duplication, deletion, and rearrangement in the mouse and human genomes. *Proc Natl Acad Sci U S A*, 100(20):11484–11489, Sep 2003.
13. G. Lunter, I. Miklos, Y. Song, and J. Hein. An efficient algorithm for statistical multiple alignment on arbitrary phylogenetic trees. *J Computational Biology*, 10(6):869–89, 2003.
14. W. Miller. Personal communication.
15. E. Rivas. Evolutionary models for insertions and deletions in a probabilistic modeling framework. *BMC Bioinformatics*, 6(1):63, 2005.
16. A. Siepel, G. Bejerano, J. S. Pedersen, A. S. Hinrichs, M. Hou, K. Rosenbloom, H. Clawson, J. Spieth, L. W. Hillier, S. Richards, G. M. Weinstock, R. K. Wilson, R. A. Gibbs, W. J. Kent, W. Miller, and D. Haussler. Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome Res*, 15(8):1034–1050, Aug 2005.
17. A. Siepel and D. Haussler. Combining phylogenetic and hidden markov models in biosequence analysis. *J Comput Biology*, 11(2-3):413–28, 2004.
18. J. Thorne, H. Kishino, and J. Felsenstein. Inching toward reality: an improved likelihood model of sequence evolution. *J. Mol. Evol.*, 34:3–16, 1992.
19. J. L. Thorne, H. Kishino, and J. Felsenstein. An evolutionary model for maximum likelihood alignment of DNA sequences. *J Mol Evol*, 33(2):114–124, Aug 1991.
20. Z. Yang. Among-site rate variation and its impact on phylogenetic analysis, 1996.